UNIVERZA V LJUBLJANI

EKONOMSKA FAKULTETA

MAGISTRSKO DELO

VIZUALIZACIJA VELIKIH PODATKOVNIH ZBIRK Z OMEJENIMI VIRI

Ljubljana, avgust 2017 MARKO SKUBIC

IZJAVA O AVTORSTVU

Podpisani Marko Skubic, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Vizualizacija velikih podatkovnih zbirk z omejenimi viri, pripravljenega v sodelovanju s svetovalcem prof. dr. Jurijem Jakličem,

IZJAVLJAM,

1. da sem predloženo delo pripravil samostojno;

2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;

3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;

4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;

5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;

6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu ter jih v njem jasno označil;

7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;

8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;

9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;

10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne 20. 7. 2018 Podpis študenta:

KAZALO

KAZALO SLIK

SEZNAM KRATIC

ang. – angleško

IT – informacijska tehnologija

MS – Microsoft

MOBA – velika spletna bojna arena (ang. Massive online battle arena)

OLAP – omrežna analitična obdelava (ang. On-Line analitical processing)

UVOD

Poslovni svet je že od vsega začetka temeljil na poznavanju trga in prodaji izdelkov, ki jih trg želi. Za spoznavanje trga so bili vedno potrebni podatki o njem in potrebi ter željah potrošnikov, ki so jih trgovci morali nekako zbrati. Primitivni načini zbiranja podatkov so se začeli med industrijsko revolucijo vse bolj razvijati, največji pospešek pa je razvoju dala doba računalnikov in sorodne informacijske tehnologije (v nadaljevanju IT). V 21. stoletju so podatki bistven del vsakega poslovanja. A čeprav je zbiranje podatkov danes trivialno, so se ravno zaradi tega odprli novi izzivi strokovne obravnave tega področja (Larsen, 2017).

Res je, da je danes podatkov na pretek, ampak preveč podatkov je pogosto težava in ne rešitev, saj vse mase podatkov ne moremo obdelati oz. to naredimo težko (Larsen, 2017; Marr, 2017). S tem se ukvarja veja IT in statistike, ki jo imenujemo podatkovna znanost. Podatkovna znanost se ukvarja z iskanjem vrednosti v velikih količinah podatkov, ki so v nekem časovnem obdobju zbrani v podatkovnih zbirkah. V ta namen podatkovni znanstveniki uporabljajo veliko različnih metod za obdelavo, katerih namen je iz omenjenih zbirk izluščiti koristne informacije.

Podatkovna znanost temelji na treh stebrih – vsak se osredotoča na posamezno fazo v iskanju uporabnih informacij. Prvi steber je zbiranje in shranjevanje podatkov, ki se osredotoča na to, da so podatki zbrani, shranjeni in dokumentirani. Zbiranje in shranjevanje sta samoumevni, pomembno pa je tudi dokumentiranje, saj so opisi podatkov, ki jih imenujemo metapodatki, prav tako pomembni pri nadaljnji obdelavi. Naslednji steber je rudarjenje podatkov, zraven pa lahko štejemo tudi osnovno analitiko, se pravi bolj ali manj avtomatizirane metode statistične analize, prirejene za raziskovanje podatkovnih zbirk. Tretji steber je vizualizacija, ki se ukvarja s tem, kako čim bolj intuitivno in učinkovito predstaviti pridobljene informacije vsem, ki jim informacije lahko koristijo (Chang, Yang & Procopio, 2016). V delu bomo definirali vse tri stebre, saj le redko delujejo neodvisno, osredotočali pa se bomo predvsem na vizualizacijo.

Obdelovanje podatkov ima kar nekaj izzivov, ki jih je treba rešiti, če želimo iz njih dobiti kakovostno informacijo. Bistvena izziva sta kakovost podatkov (Sherice, b. d.), predimenzioniranost (Chan, Correa & Ma, 2014). Kakovost podatkov moramo čim bolje oceniti, saj nam nekakovostni podatki ne bodo dali pravilnih oz. reprezentativnih informacij, kljub temu da smo analitiko izvedli brezhibno. Če je ocena kakovosti slaba, moramo podatke skušati popraviti, s čimer preprečimo napake v kasnejši analizi. Predimenzioniranost, tako navpična (kjer gre za veliko število vnosov) kot vodoravna (kjer gre za veliko število atributov ali spremenljivk), zahteva, da metode ustrezno priredimo, če se želimo izogniti ozkim grlom ali nepreglednosti. Predvsem vodoravna predimenzioniranost je v primeru vizualizacije resen izziv.

Zaradi pomembnosti podatkovne znanosti tako v znanstvenem kot v poslovnem svetu je bilo v zvezi z obdelavo podatkov že veliko napisanega, narejenega. Toda ahilova peta te literature, programske opreme je tako v strokovnosti kot v zaprtosti. Članki so pogosto zelo strokovno zastavljeni (na primer Cottam, Lumsdaine & Wang, 2013; Pahins, Stephens, Scheidegger & Comba, 2017; Wang, Ferreira, Wei, Bhaskar & Scheidegger, 2017), poleg tega pa so osredotočeni na koncepte rešitev in manj na njihovo izvedbo, predvsem z omejenimi viri. Drugi članki so osredotočeni na zelo specifične niše (na primer Godfrey, Gryz & Lasek, 2016; Goguelin, Flynn, Essink & Dhokia, 2017), kar ponovno onemogoči njihovo splošno uporabo. Bolj splošna literatura pa je pogosto zelo teoretične narave v obliki različnih učbenikov. Na koncu imamo še veliko število različnih spletnih člankov in blogov, ki pa so običajno presplošni in pogosto tudi delno namenjeni oglaševanju neke programske rešitve ali izvajalca storitve. Namen magistrskega dela je torej ugotoviti, ali je z viri, do katerih ima dostop povprečno manjše podjetje, mogoče izvesti razumljivo vizualizacijo velike podatkovne zbirke.

V delu se bomo zato osredotočili na podatkovno analitiko in predvsem vizualizacijo večje podatkovne zbirke z uporabo orodij, ki jih lahko pričakujemo v skoraj vsakem podjetju ali podobni organizaciji, to je paketom Microsoft Office (v nadaljevanju MS Office). Paket vsebuje mnogo uporabnih in zmogljivih orodij, tako za obdelavo dokumentov kot za upravljanje z razpredelnicami, statistiko in tudi preprostimi podatkovnimi bazami. Kot najbolj razširjen paket take vrste je idealen za to raziskavo. Tu namreč ne obdelujemo podatkovne zbirke, saj to ni bistvenega pomena za cilj dela. Naš cilj je ugotoviti, če lahko s splošno razširjenimi orodji izvedemo analizo in vizualizacijo zahtevnejše podatkovne zbirke in s kakšnimi izzivi se bomo pri tem srečevali. Na ta način želimo ugotoviti, ali lahko manjša podjetja in podobne organizacije same izvedejo preproste metode podatkovne znanosti, ne da bi morale najeti ali kupiti drago programsko opremo oz. zaposliti ali najeti ustrezne strokovnjake.

Metodološko se bomo najprej osredotočili na pregled literature in iz nje potegnili bistvene podatke. Glede na to bomo nato eksperimentalno preverili, kako težko je z omejenimi viri izvesti vizualizacijo velike podatkovne zbirke.

V delu bomo najprej pregledali, kaj je bilo že narejeno. Orisali bomo teoretično ozadje podatkovne znanosti in izzive, s katerimi se lahko srečamo. Nato bomo definirali faze obdelovanja podatkov in metode, s katerimi faze izvajamo. Metode bomo nato ovrednotili, če so smiselne za uporabo na našem primeru. V drugem delu bomo predstavili praktičen primer podatkovne zbirke. Opisali bomo njeno ozadje (igro League of Legends na najvišji tekmovalni ravni), saj je razumevanje igre pomembno za razumevanje podatkovne zbirke. Podatkovno zbirko bomo nato prepeljali skozi faze obdelave podatkov – najbolj se bomo osredotočili na vizualizacijo rešitve. S tem bomo skušali ugotoviti, ali je tako podatkovno zbirko sploh mogoče dobro vizualizirati le z uporabo preprostih orodij in kakšne težave lahko pri tem nastanejo.

1 PODATKOVNA ZNANOST

1.1 Opis podatkovne znanosti

Podatkovna znanost je relativno sodobna veda, ki jo uvrščamo med veje informatike. Veda se, kot nam ime pove, ukvarja z zbiranjem in obdelavo podatkov, pridobivanjem novih informacij iz obstoječih podatkov in predstavitvijo podatkov za namen uspešnejšega odločanja. Jedro podatkovne znanosti temelji na iskanju odgovorov, ki na prvi pogled niso zapisani v podatkovni zbirki. To dosežemo z raziskovanjem podatkov in iskanjem trendov, vzorcev, agregacij. To zahteva analitično in kreativno miselnost ter pristop, saj odgovori niso vedno takoj jasni ali enoznačni. Če so vprašanja kompleksnejša, kar pa je dandanes pogostejša zahteva, moramo iti globlje, poiskati abstraktnejše odgovore in jih nato logično interpretirati (Lo, b. d.).

Podatkovna znanost je dobila zlasti velik zagon v zadnjih desetletjih, prav zaradi povečanja zmogljivosti računalnikov. Ker zaradi kapacitete in procesorske moči lahko zberemo veliko več podatkov (in jih tudi trajno shranimo), je bilo treba poiskati načine, kako iz nepregledne množice izluščiti ustrezno informacijo – odgovore na naša vprašanja, potrebna za odločanje na strateški ali dnevni ravni. Podatki za večino organizacij nimajo večje vrednosti, razen za statistike in raziskovalce. Bistven za uspešnost projekta je podatkovni proizvod. To je lahko Amazonov predlog nakupa, predvidevanje obnašanja strank ali pa kakovosten graf, ki bo vodilnim pomagal pri odločanju. Vsi ti proizvodi so posledica razvoja in dela podatkovnih inženirjev (Schutt & O'Neil, 2013).

Preden naredimo podrobnejšo analizo podatkovne znanosti in spoznamo, da predstavlja pomemben del poslovnega sveta, moramo najprej opredeliti razliko med podatkovno znanostjo ter njenima sorodnima disciplinama: velikimi podatki (ang. big data) in podatkovno analitiko. Področje velikih podatkov se ukvarja predvsem z zbiranjem in shranjevanjem masivnih količin podatkov ter z njihovo pripravo za nadaljnjo uporabo. Podatkovna analitika se ukvarja z analizo podatkov, se pravi s povprečji, odkloni, razponi in povezavo med različnimi podatkovnimi zbirkami. Podatkovna znanost se na koncu ukvarja s semantično obdelavo pripravljenih podatkov in pridobivanjem informacij, ki so tam le implicitne – drugače povedano: analitika se ukvarja s tem, kaj nam podatki pravijo, podatkovna znanost pa s tem, kaj bi nam poleg tega še lahko sporočili (Monnappa, 2016b).

Če torej povzamemo definiciji (tudi ostale so približno take), ugotovimo, da se podatkovna znanost nanaša na postopke in metode, ki iz že obstoječih podatkov pridobijo dodatno vrednost oz. informacijo, ki pomeni dodatno kakovost in ki v podatkih eksplicitno ni zapisana. V tem se podatkovna znanost tudi razlikuje od klasične podatkovne analitike, ki le agregira in razlaga podatke. Ker v primeru podatkovne znanosti iščemo dodatno informacijo, se kompetence podatkovnega znanstvenika razlikujejo od analitikov. Podatkovni znanstveniki morajo imeti poleg poznavanja metod obdelave podatkov tudi občutek in voljo

do raziskovanja ter eksperimentiranja, ne le analitičnih sposobnosti. Zaradi tega je lahko podatkovna znanost tudi konjiček, po drugi strani pa to pomeni, da lahko v kateri koli gospodarski družbi ali drugi instituciji, ki je lahko tudi javnopravne narave, za iskanje skritih informacij uporabimo obstoječe zaposlene, ki so voljni in sposobni iskati podrobnosti ter skrite povezave, neopazne navzven.

Chang, Yang in Procopio (2016) v članku opredelijo tri stebre podatkovne znanosti: podatkovne baze (oz. napredke v njihovi tehnologiji), podatkovno rudarjenje in vizualizacijo. V svojem članku se nato osredotočijo na vizualizacijo, o čemer bo več napisano v nadaljevanju magistrskega dela. Podatkovna znanost torej temelji na teh treh stebrih; vsi lahko glede na osnovno opredelitev dajo podatkom neko dodano vrednost. Podatkovne baze jih zadržijo in kategorizirajo, podatkovno rudarjenje išče prej skrite povezave in vzorce, vizualizacija pa omogoči predstavitev podatkov v ljudem razumljivi obliki.

To sicer ne pomeni, da posamezni stebri ne morejo biti samostojni. Posamezni steber lahko apliciramo nepovezano z ostalimi, a bomo v tem primeru zelo omejeni pri tem, kaj lahko s podatki sploh naredimo. V dokument zapisano podatkovno zbirko lahko sicer obdelujemo, a jo moramo za natančnejšo obdelavo vsaj zapisati v tabelo, lokalna podatkovna baza pa je pogosto še boljša rešitev. Enako lahko obdelujemo podatke brez rudarjenja, a bomo z vizualizacijo uspeli dobiti le njihovo grafično predstavitev, kar pa tudi ni nujno slabo. Iz dobre grafične predstavitve je moč izluščiti informacijo, ki je računalnik z orodji za podatkovno rudarjenje in raziskovanje podatkov ne bi mogel. Kljub vsemu iz podatkov največ pridobimo, če vsi trije stebri delujejo v sinergiji.

Trije stebri podatkovne znanosti vseeno zahtevajo različna znanja, ki jih mora dober podatkovni znanstvenik obvladati (ali vsaj poznati). To lahko pripelje do težave, če si podjetja, predvsem manjša in srednja, ne morejo privoščiti izvedencev v tej stroki. Na trgu sicer obstaja veliko orodij, ki jih lahko uporabimo (mySQL server, Microsoft Access za baze, Microsoft Visual studio za rudarjenje ter Excel in PowerBI, ki omogočata vizualizacijo poleg veliko poslovnih orodij), a brez znanja o podatkovni znanosti nam bo vedno manjkala neka dodana vrednost, tisto nekaj več, kar na trgu lahko pripomore k večji konkurenčni prednosti. V tem primeru se moramo izobraziti vsaj v osnovah podatkovne znanosti, če pa imamo raziskovalno žilico, toliko bolje. Pomembno je, da se pravilno odločimo, kaj pravzaprav potrebujemo.

Vizualizacija zahteva največ mehkih sposobnosti, saj je končni člen med podatki in ljudmi. Podatkovna tehnologija in informacije, pridobljene z rudarjenjem, niso veliko vredne, če jih ne znamo predstaviti. Zato se je pogosto najbolje osredotočiti na dobro vizualizacijo.

1.2 Pomen podatkovne znanosti v poslovnem okolju

Podatkovna znanost se kot večina ostalih podpornih dejavnosti v poslovnem svetu pojavlja v dveh osnovnih oblikah. To sta osnovna dejavnost, ki jo podjetje prodaja na trgu, in podporna dejavnost, ki podjetju pomaga pri izvajanju in izboljšavi osnovnih dejavnosti. Ker se v delu navezujemo predvsem na situacije, ko ima organizacija omejene vire za izvedbo metod, ki spadajo v področje podatkovne znanosti, se bomo osredotočili predvsem na drugo obliko. Podjetja, ki podatkovno znanost tržijo kot svojo osnovno dejavnost (na primer v smislu poslovnega svetovanja in analiz trga), bodo v ta namen zaposlila podatkovne inženirje, saj so njihov najpomembnejši kader (Columbus, 2018). Po drugi strani podjetja, ki se s podatkovno znanostjo ne ukvarjajo redno, večinoma nimajo za to šolanih ali drugače ustrezno usposobljenih strokovnjakov (Morgan, 2016). Takim podjetjem tako ostane na voljo uporabiti obstoječi kader, zaposliti ustreznega strokovnjaka ali pa se odločiti, da bodo globinsko analizo podatkov prepustili zunanjim izvajalcem. Vse tri možnosti imajo svoje prednosti in slabosti, zato se morajo podjetja ustrezno odločiti, katero bodo izbrali (o načinih izbire tukaj ne bomo govorili, saj to ni v sklopu tematike dela) (Provost & Fawcett, 2013).

Se pa moramo zato najprej vprašati, zakaj bi se podjetja sploh ukvarjala s podatkovno znanostjo oz. boljše rečeno: z globinsko analizo in bolj intuitivno predstavitvijo podatkov (ki večinoma zahteva vizualizacijo).

V sodobni informacijski družbi imajo podjetja na razpolago ogromne količine podatkov. Podatki navadno prispejo iz različnih virov, so različne kakovosti, pogosto niso standardizirani ali poenoteni, poleg tega pa nekateri ničesar ne prispevajo k odgovoru na vprašanja, ki so pomembna za poslovne odločitve (in tu se pogosto pojavi težava, predvsem pri manj izkušenih odločevalcih). Ravno tukaj lahko največ pripomorejo podatkovni znanstveniki, saj njihova znanja temeljijo na pretvorbi suhoparnih in različnih podatkov v uporabne informacije, s katerimi si nato vodilni pomagajo pri strateških odločitvah (Techlabs, 2017).

Definiramo lahko več področij, kjer podatkovna znanost pripomore k povečanju poslovne vrednosti (Monnappa, 2016a):

- izboljšanje sposobnosti odločanja pri managementu,
- usmerjanje dejanj glede na trende in definicije ciljev,
- usmerjanje zaposlenih k dobrim praksam in merjenje rezultatov,
- definiranje bistvenih področij in iskanje izboljšav,
- iskanje novih priložnosti,
- odločanje s pomočjo merljivih, podatkovno podprtih virov,
- preverjanje učinkovitosti rešitev,
- iskanje in analizo ciljnih skupin,
- zaposlovanje ustreznih ljudi za podjetje – kadrovski management.

Podatkovna znanost torej pripomore k izboljšanju delovanja podjetja prek dveh primarnih vektorjev. Prvi vektor je strateško načrtovanje (torej srednje- in dolgoročne odločitve), pri katerih podatkovna znanost odstrani velik del negotovosti. Negotovosti sicer ne moremo popolnoma odstraniti, saj podatkovna znanost za delovanje še vedno potrebuje podatke, ki pa so navadno lahko le zgodovinski. Trendi in napovedi nam lahko pomagajo oceniti optimalno pot razvoja podjetja v prihodnosti, ampak le, če imamo ustrezne in točne podatke; če so podatki slabi ali neustrezni, lahko preveliko zanašanje na podatkovno znanost povzroči slabše rezultate kot zanašanje le na osnovno statistiko in intuicijo.

V poslovnem svetu podatkovna znanost sicer ni novost, so pa novi pristopi, ki zbrane podatke pretvorijo v uporabno informacijo. V sodobnem poslovanju je količina zbranih podatkov preprosto prevelika, da bi omogočala ročno obdelavo, zato je v moderni poslovni podatkovni znanosti poznavanje orodij prav tako pomembno kot sposobnost raziskovanja in iskanja alternativnih vzorcev v podatkih.

Predvsem odločanje na osnovi boljših informacij in predvidevanj je element, ki podjetjem, ki vlagajo v poslovno inteligenco in podatkovno znanost, doda konkurenčno prednost in izboljša njihov poslovni rezultat. Iskanje podobnosti z zgodovinskimi dogodki lahko omogoči, da se na kasnejše predvidljive dogodke bolje pripravimo, s tem pa prehitimo konkurenco, ki na situacijo ne bo pripravljena. Na tak način lahko izkoristimo občasne, a predvidljive dogodke in z njimi ustvarimo dodaten dobiček.

Obdelava podatkov pogosto omogoči tudi boljše trženje izdelkov ciljnim skupinam in njihovo prepoznavo. Spremljanje nakupovalnih navad strank na primer omogoči njihovo razvrstitev v ustrezne skupine, za katere lahko kasneje izdelamo ustrezno prirejeno trženjsko strategijo. Poleg tega omogoči spremljanje poslovnega prihodka po skupinah in identifikacijo najboljših ter ugotavljanje težav pri slabših (Provost & Fawcett, 2013).

Ena izmed težav, s katerimi se lahko sreča poslovna podatkovna znanost v bližnji prihodnosti, je vedno večja ozaveščenost o zasebnosti. Velik del podatkovne znanosti se zanaša na velike količine podatkov, ki pa jih je najprej treba zbrati. Marsikaterega podatka se za potrebe poslovne analitike ne da popolnoma anonimizirati, kar je potencialno lahko poseg v posameznikovo zasebnost. V primeru zaostritve zakonov o varovanju zasebnosti in osebnih podatkov se lahko zgodi, da bo vsaj ta del podatkovne znanosti postal nekoliko manj učinkovit. To sicer ne zmanjša pretirano pomena podatkovne znanosti, saj je, kot meni Monnappa (2016a), velik del podatkovne znanosti oprt na zaprte in/ali anonimizirane podatke.

2 OBDELAVA PODATKOV V PRIMERIH VELIKIH PODATKOVNIH ZBIRK

2.1 Zbiranje, tipi in kakovost podatkov

Vsi podatki, ki jih analitiki obdelujejo, morajo biti najprej nekje zbrani. Zbiranje podatkov je tako faza, brez katere dejansko ne moremo izvesti kasnejših operacij. Ker pa kakovost podatkov, ki jih zberemo kasneje, zelo vpliva na kakovost informacij (tu gre za princip »Garbage in – garbage out« oz. smeti noter – smeti ven, ki pomeni, da iz slabih podatkov lahko dobimo le slabe informacije), moramo pri zbiranju paziti, da ne dobimo slabih podatkov.

Podatki za obdelavo (govorimo predvsem o bolj komercialni uporabi) najpogosteje pride iz podjetju lastnih podatkovnih baz in skladišč. A vseh podatkov pogosto ne moremo dobiti iz lastnih virov. V tem primeru jih moramo pridobiti od zunaj.

Zunanji viri podatkov nam lahko pri marsikateri analizi zelo pomagajo, saj tam dobimo pogosto podatke, ki jih sami ne moremo ali pa ne smemo zbirati. Če so podatki kakovostni in nam pokažejo tisto, česar sami nimamo, lahko s tako zbirko dodamo vrednost prvotni analizi. Vendar pa moramo biti pri zunanjih zbirkah zelo previdni in dosledno upoštevati parametre ter metapodatke. Zunanje zbirke niso narejene za nas, zato se lahko pogosto zgodi, da kakšen parameter odstopa od optimalnih, kar moramo prav tako upoštevati.

Stvar postane nekoliko drugačna, če iščemo zbirko podatkov za lastno ali akademsko uporabo. V tem primeru namesto tega, da iščemo podatke, ki ustrezajo vprašanju, raje poiščimo podatkovno zbirko, ki nam je všeč, in vprašanja zastavimo glede na podatke, ki jih imamo na voljo. Na spletu obstaja več virov, kjer lahko dobimo brezplačne, javno dostopne zbirke, na marsikateri strani pa obstajajo tudi spletni forumi in klepetalnice o tematiki podatkovne znanosti. V primeru, da se odločimo za tako raziskavo, je prav, da vemo, kaj je pravzaprav cilj naše obravnave, s čimer se tudi odločimo, kakšno zbirko želimo. Za vizualizacije so najboljše zbirke, ki so dovolj zanimive in urejene za dobre grafe ter podobne grafične prikaze. Za procesiranje podatkov (statistične analize) so najprimernejše čiste in urejene zbirke, ki morajo biti dovolj velike za kakovostno statistično analizo, poleg tega pa morajo imeti dobro definirane metapodatke. Za podatkovno rudarjenje so dobre zbirke, kjer lahko izvedemo dobre napovedi, stolpci pa niso preveč neodvisni (strojno učenje namreč skuša iz skupine vrednosti več spremenljivk izvesti napoved, za kar pa je obvezna logična medsebojna povezanost). Če pa želimo izvesti čiščenje in pripravo podatkov, je najbolje pridobiti zbirko, kjer podatki niso najbolj kakovostni (Paruchuri, 2016). Primeri na takih (imenujmo jih vadbenih) podatkovnih zbirkah nam tudi pomagajo pri prepoznavanju, kateri podatki oz. podatkovne zbirke so primerne za posamezne faze ali metode podatkovne znanosti, in nas opozorijo na pasti, s katerimi se lahko začetnik sreča.

Nekakšen vmesni način je t. i. metoda luščenja spletnih podatkov (ang. *website scraping*), ki s spletišča postrga podatke in jih vpiše v strukturirano tabelo. Metoda žal ni univerzalno uporabna, saj zahteva strukturiranost spletišča, kar pomeni, da spletišče vsebuje veliko relativno podobnih spletnih strani. Primerna spletišča za to metodo so na primer spletne trgovine, strani z malimi oglasi, spletni komentarji, strani s športnimi rezultati in podobne.

Ne glede na to, kako pridobimo podatke, moramo paziti, da poleg njih pridobimo čim več metapodatkov. Metapodatki so nepogrešljivi predvsem pri pripravi zbirke, kasneje pa tudi pri ocenjevanju zanesljivosti in reprezentativnosti (reprezentativnost pomeni, kako zanesljivo podatki predstavljajo stanje v realnem svetu, kar je še posebej pomembno pri analizah, kjer imamo manjše vzorce, na primer pri analizi javnega mnenja). Metapodatki vsebujejo informacije o podatkih oz. natančneje: metapodatki povejo, kakšnega tipa so podatki, kaj točno pomenijo, kako so bili zbrani in ostalo o metodologiji. Ko zbiramo metapodatke, moramo najprej upoštevati vir osnovnih podatkov. Če gre za podatke iz lastnih zbirk, baz in analiz, je že v začetku pametno upoštevati njihovo pomembnost in zato vse pri pridobivanju ustrezno dokumentirati. Če pa podatke pridobimo iz zunanjih virov, je pomembno, da skušamo o zbirki pridobiti čim več informacij, predvsem v zvezi z metodologijo zbiranja. Tipe in način zapisa podatkov namreč lahko ugotovimo s pregledom zbirke (kar bomo v prvi fazi obdelave tako ali tako storili), ne moremo pa ugotoviti kakovosti, tako posameznih elementov kot podatkovne zbirke v celoti.

Kakovost podatkov je sicer relativno težko opredeliti. Tako kot za podatke tudi za parametre kakovosti (merljive parametre kakovosti moramo definirati, saj brez njih lahko zbirko ocenimo le »čez prst«) ni neke univerzalne formule. Pri parametrih moramo prav tako upoštevati, kdo jih je definiral in zakaj ter za katere namene so bili postavljeni. Poleg tega moramo pri parametrih določiti tudi raven sprejemljivosti. Ti elementi se pogosto razlikujejo glede na namen analize (akademska/raziskovalna ali poslovna) pa tudi glede na tip analize, kar smo omenili že prej. Poleg tega moramo definirati tudi, kakšno odstopanje od optimuma (če ga sploh lahko dobro definiramo) je za nas še sprejemljivo. Še en pomemben element pri definiranju parametrov kakovosti je tudi vir podatkov, saj lahko pri lastnih podatkih bolj nadziramo, ali je zbirka še v sprejemljivih vrednostih parametrov.

Podatkovno profiliranje je eden izmed načinov, kako lahko začnemo z ocenjevanjem kakovosti. Pri tej metodi gre za to, da zbirko (ali podatkovno bazo) pregledamo za točnost in popolnost podatkov. V tej fazi lahko tudi ugotovimo, ali so določeni podatki slabo zapisani oz. zapisani v napačnem podatkovnem tipu (na primer telefonske številke zapisane kot celo število ali rojstni podatki zapisani na način leto-mesec-dan ali obratno). Ko smo ocenili kakovost in našli potencialne napake, se navadno tudi odločimo, kaj s takimi napakami narediti. Ker bodo napake verjetno različnih tipov, se lahko za posamezni tip odločimo, kako ga bomo obravnavali. V vsakem primeru nas potencial za napake ne sme odvrniti od tega, da bi se lotili obdelave podatkovne zbirke. Zbirk brez napak skorajda ni, zato moramo vedno sprejeti kompromis (Sherice, b. d.).

Kljub temu lahko pri ocenjevanju kakovosti podatkov definiramo dimenzije, s katerimi ocenimo kakovost podatkovne zbirke. Askham et al. (2013) v študiji definirajo šest dimenzij:

- popolnost,
- edinstvenost,
- pravočasnost,
- veljavnost,
- natančnost,
- konsistentnost.

Popolnost je definirana v odstotku izpolnjenih podatkovnih polj glede na poslovna pravila, ki jim podatkovna zbirka sledi. Drugače povedano, 100-% popolnost je dosežena, ko so vsi bistveni podatki v podatkovni zbirki izpolnjeni, kar pomeni, da v bistvenih stolpcih ni praznih oz. neizpolnjenih polj. Odstotek, ki ga dimenzija popolnosti doseže, je izračunan glede na razmerje med vsemi zahtevanimi in dejanskimi podatki. Pri stopnji popolnosti pod 100 % moramo definirati, kaj narediti z manjkajočimi podatki.

Edinstvenost je naslednja dimenzija, ki jo definira DAMA UK. Definirana je kot edinstvenost vrstice v razmerju z njenim elementom v resničnem svetu. To pomeni, da vsak element (vrstica) ustreza eni entiteti. Če se vnosi ponavljajo, to pomeni, da je neki element zapisan v bazo večkrat, zato je edinstvenost manjša od 100 %. Dimenzija se izračuna glede na razmerje med številom vrstic v tabeli in dejanskim številom elementov v resničnem svetu. Če lahko ponavljajoče se vrstice ustrezno grupiramo, jih najlažje očistimo tako, da dvojnike združimo ali preprosto izbrišemo.

Pravočasnost ali veljavnost je dimenzija, ki definira, ali podatki v zbirki časovno ustrezajo obdobju, za katero pravzaprav delamo analizo. Če ne ustrezajo, so bodisi zastareli bodisi preveč novi (v primeru, da izdelujemo zgodovinske analize). Ko analiziramo pravočasnost, moramo pogosto upoštevati poslovna pravila, kamor sodi tudi veljavni pravni red, saj je veljavnost podatkov pogosto pogojena z njimi. Prav tako moramo že vnaprej določiti časovne okvire, s katerimi ocenimo podatkovno zbirko. Ustrezne časovne okvire moramo glede na analizo in potrebe ugotoviti za vsak primer (ali celo podatkovni tip) posebej.

Dimenzija veljavnosti je sorodna popolnosti, le da tu opazujemo globlje. Če pri popolnosti iščemo manjkajoče podatke, pri pregledu veljavnosti iščemo vnose, ki ne ustrezajo zahtevanemu tipu podatka. Za analizo veljavnosti so zelo pomembni metapodatki, saj definirajo pravila, ki jih mora posamezni vnos upoštevati. Podatki, ki pravilom ne ustrezajo, so označeni za neveljavne, stopnja veljavnosti pa je odstotek neveljavnih zapisov glede na vse zapise. Pomembno je, da pri analizi veljavnosti za neveljavne označimo le zapise, ki ne ustrezajo splošnim pravilom zapisa (na primer telefonsko številko, kjer vnos vsebuje črke, ali poštno številko za kraj v Sloveniji, ki ima šest znakov), ne pa tistih, ki sicer ustrezajo

pravilom, ampak so napačni. Taki zapisi so veljavni, ampak ne ustrezajo dimenziji natančnosti.

Natančnost ali zanesljivost je naslednja dimenzija, ki jo analiziramo. Če smo pri analizi veljavnosti ugotavljali, kateri podatki so neveljavni, saj ne ustrezajo pravilom zapisa podatka (tudi če je podatek pravilen), pri pregledu natančnosti skušamo ugotoviti, če so podatki, zapisani v zbirki, dejansko pravilni. Ker po navadi nimamo popolnoma zanesljive reference, ki bi jo lahko uporabili za pregled vseh podatkov, moramo najprej ugotoviti, katere podatke je sploh smiselno analizirati v dimenziji natančnosti. Pri tej analizi si najpogosteje pomagamo z zanesljivimi viri, če so na voljo in javno dostopni. Če virov ni, bomo analizo natančnosti zelo težko izvedli. Analiza natančnosti je sicer lahko večstopenjska, saj lahko analiziramo zbirko kot celoto, lahko pa tudi vsak vnos (ali del vnosa) posebej – odvisno od naših potreb in razpoložljivih virov. Že zelo osnovna analiza natančnosti lahko prikaže potencialne napake v zbirki (na primer podatkovna zbirka krajanov nekega kraja, ki je večja od dejanskega števila prebivalcev, ne more biti najbolj zanesljiva).

Zadnja dimenzija je konsistenca, ki jo po navadi analiziramo na ravni metapodatkov. V primeru konsistence gre za to, da so podatki (pogosteje podatkovni tipi) enakomerni glede na poslovna pravila, po katerih deluje podatkovna zbirka (na primer vsi datumi so ali niso zapisani v enakem formatu). V samostojnih zbirkah konsistenca navadno ni problematična, pod pogojem, da je podatkovna baza oz. drugi vir podatkov kakovostno postavljen. Analiza konsistence postane bolj pomembna, če združujemo več različnih virov oz. zbirk, še posebej iz različnih organizacij, saj se poslovna pravila lahko razlikujejo (na primer zapis datuma ali merske enote). Prav tako je konsistenca pomembna pri podatkovnih tipih, ki zaradi specifičnih lastnosti nimajo definirane rigidne strukture, še posebej, če podatke v sistem vnaša več oseb (na primer dnevna poročila).

Ocena kakovosti podatkov, hkrati pa tudi analiza strukture podatkovne zbirke, je pomembna ne glede na njeno velikost. Tako velike kot majhne podatkovne zbirke lahko vsebujejo napake ali nekakovostne podatke, ki bi pri kasnejši obdelavi lahko pripeljale do nereprezentativnih rezultatov. Pri velikih podatkovnih zbirkah, predvsem tistih, ki imajo veliko število vnosov, je sicer praviloma nemogoče pregledati vse vnose, zato se ta faza avtomatizira. Težava nastane, če avtomatizacija ni mogoča. V tem primeru imamo na voljo možnost, da najbolj kritične podatkovne vnose selekcioniramo in kritična področja pregledamo ročno. Če to ni mogoče, moramo sprejeti neki potencialen odstotek neizogibne napake.

2.2 Priprava podatkov

Priprava podatkov je druga pomembna faza, ki se ji pravzaprav ne moremo izogniti, ne glede na to, kakšne metode bomo za obdelovanje podatkov potrebovali kasneje. Meja med fazo ocenjevanja kakovosti in analizo metapodatkov ter fazo priprave podatkov je pogosto

zabrisana. Harris (2018) v blogu kot pripravo podatkov definira pet D (ang. *discover, detain, distil, document* in *deliver* oz. po slovensko odkrij, zadrži, prečisti, dokumentiraj in dostavi). Med petimi elementi se eden nanaša na pridobivanje (ang. *discover*), dva na kakovost (ang. *detain* in *document*) in le eden na transformacijo podatkov v obliko, ki je optimizirana za obdelavo (ang. *distil*). Vendar pa Harris poudarja pomembnost cikličnosti in dolgoročnega razmišljanja. Po njegovem ni dovolj, da v poslovnem svetu najdemo podatke, ki so potrebni le tukaj in zdaj, temveč moramo najti tudi podatke, ki bodo potrebni kasneje. Ker pa vseh potencialnih zahtev ali priložnosti ne moremo napovedati vnaprej, je ponavljanje faz priprave podatkov (po Harris, 2018) stalen organski proces, ki ga dobri podatkovni znanstveniki stalno izvajajo v ozadju.

Ker smo o fazi ocenjevanja kakovosti in strukture podatkov že pisali, se bomo tu osredotočili predvsem na naslednjo fazo. V tem smislu je priprava podatkov proces, kjer podatke, zapisane na način, ki ni najbolj primeren za obsežnejšo obdelavo (je pa morda boljši za prenašanje ali dnevno uporabo), pretvorimo v strukturo, ki nam bo omogočala metode obdelave. Hkrati jih v tej fazi tudi prečistimo (tu bi lahko fazo priprave podatkov razdelili na podfazi: pretvorbo v primeren zapis in čiščenje podatkov). Čiščenje podatkov je proces, kjer podatke, ki smo jih prej ocenili kot slabe ali neprimerne, pretvorimo tako, da ne bodo škodili kasnejši obdelavi.

Glede na vir podatkov so zbirke zapisane na več različnih načinov. V poslovnem svetu je primarni vir pogosto domača podatkovna baza. Te so običajno optimizirane za vsakodnevno uporabo (na primer pogoste preproste poizvedbe ali redno dodajanje), niso pa najbolj primerne za obsežnejšo obdelavo. Če želimo iz podatkov, na primer letnega prometa podjetja, pridobiti manj očitne informacije, je pogosto treba dobiti sliko podatkovne baze, saj marsikatero orodje za strojno učenje ne podpira neposredne povezave z bazo. Nekatera druga orodja (na primer MS Visual Studio) povezovanje podpirajo, vendar pa je zadeva nekoliko okorna (še posebej, če se povezujemo prek interneta), zato je bolje ustvariti sliko baze ali pa naložiti podatke v pomnilnik (ali oblak). Tak sistem na primer uporablja sistem Online analitical processing (v nadaljevanju OLAP). OLAP omogoča večdimenzionalno obdelavo, uporabno za analizo in vizualizacije.

Drugi pogosti vir podatkov so javne podatkovne zbirke, ki jih pridobimo na spletnih straneh. Navadno so zapisane v formatu, ki omogoča preprost prenos, branje in pretvorbo, niso pa najprimernejše za neposredno obdelavo. Zelo pogost tip datoteke je .csv (kar pomeni comma separated values oz. z vejico ločene vrednosti), kjer so vrednosti ločene z vejico, vrstice pa v novem odstavku (datoteka .csv ima še druga pravila, a o njih kasneje). Ta tip datoteke je eden najbolj razširjenih, a ga je za kompleksnejšo obdelavo treba pretvoriti v primernejši tip datoteke, ki pa se lahko razlikuje glede na tip obdelave (iz datoteke lahko tudi zgradimo podatkovno bazo, če bomo delali neposredno z bazo).

Drugi del priprave podatkov sestoji iz t. i. čiščenja podatkov, procesa, ki naj bi odpravil napake, ki smo jih ugotovili v fazi preverjanja kakovosti. Tukaj so zelo pomembni metapodatki, saj prek njih ugotovimo tipologijo podatkovne zbirke in podatkov v njih, kar omogoči, da identificiramo potencialno slabe vnose, jih osamimo in ugotovimo, za kakšne težave gre. Tu lahko govorimo o manjkajočih, zastarelih, okvarjenih (to pomeni, da je zapis izven ustrezne maske) ali napačnih podatkih. Ko smo problematične podatke analizirali, se tudi lažje odločimo za način čiščenja.

Pri čiščenju podatkov moramo najprej izdelati načrt. Načeloma temelji na obstoječem stanju zbirke, na pridobljenih metapodatkih, poleg tega pa moramo zraven vključiti tudi svoj namen analize. Brez tega se lahko zgodi, da bomo čiščenje izvedli preveč površno (kar bo pomenilo nereprezentativen rezultat) ali pa ga bomo izvedli preveč temeljito in s tem porabili preveč časa ter virov, ki so v poslovnem okolju pogosto omejeni. Ko načrtujemo čiščenje podatkov, lahko poleg tega tudi standardiziramo ali agregiramo določene stolpce, če lahko to izboljša ali olajša našo raziskavo.

V skladu z načrtom nato določimo pravila, kaj narediti z neustreznimi podatki. Marsikaj lahko sicer brez večjih težav avtomatiziramo, elemente, ki jih ne moremo popraviti z avtomatizacijo, pa bomo morali vsaj pregledati ročno. Če je takih primerov malo in niso statistični otok, jih običajno izbrišemo. V izbris smo pogosto prisiljeni tudi, če napačnemu podatku ne moremo določiti pravilne vrednosti. Če jo lahko, podatek raje popravimo (Selot, 2012).

Metod, ki jih lahko uporabimo za čiščenje podatkov, je več. Glede na podatke lahko marsikatero avtomatiziramo (pravzaprav jih je veliko že vgrajenih v sisteme podatkovnih baz, mi jim le podamo ustrezne parametre). Po Rapid Insight (2015) obstajajo naslednje:

- Agregacija – to je združevanje več različnih vnosov v enega glede na specifičen podatek. Ko agregiramo, moramo vedno definirati tudi število združenih vnosov, čeprav ga v analizi morda ne bomo potrebovali.
- Filtriranje – to je operacija, ki glede na določen filter izbriše vrstice. Navadno v filter vstavimo neki parameter (vrednost), kot rezultat pa dobimo vrstice, ki mu ustrezajo. Metoda je uporabna za eliminacijo vnosov, kjer so zahtevani podatki manjkajoči ali neprimerni, lahko pa jo uporabimo tudi, če se osredotočamo le na del populacije v podatkovni zbirki.
- Spajanje, ki ga pri tehnologiji podatkovnih baz imenujemo tudi stik – to je metoda, ki iz dveh tabel ustvari eno, ki združuje podatke ene in druge tabele. Tabele morajo imeti vsaj enega izmed stolpcev identičnega; podatki so združeni na osnovi skupne vrednosti v tem stolpcu. Tu je treba definirati, kaj narediti z osamelci, to so tisti elementi, ki obstajajo le v eni izmed tabel. Lahko jih sprejmemo ali zavržemo.

- Dodajanje – to je operacija, sorodna spajanju, a v nasprotni dimenziji. Po navadi je potrebno, kadar moramo združiti več identičnih (ali vsaj podobnih) zbirk. Dodajanje združi dve tabeli s podobnimi podatki v eno večjo tabelo.
- Odstranjevanje dvojnikov – to je metoda, ki podobno kot agregacija več vrstic združi v eno. Primarna razlika je v vsebini: agregacija namreč sešteje različne vnose in jih združi v en primaren vnos, kar je predvsem vsebinske narave. Odstranjevanje dvojnikov je predvsem operativne narave, saj dvojniki načeloma v zbirki podatkov, kot vemo, ne smejo obstajati. Navadno se pojavijo pri dodajanju (če je en element zapisan v obeh zbirkah) ali pa pri neustrezno izdelanih oz. vodenih podatkovnih bazah.
- Transformacija – to je naprednejša možnost glede na odstranjevanje dvojnikov. Uporabimo jo za spreminjanje stolpcev. Glavna načina sta združevanje več stolpcev v enega ali razdružitev enega v več stolpcev. Načeloma so bistveni podatki v bazah atomarni, to pa ni nujno za podatke, ki med razvojem baze niso bili definirani kot atomarni. Prav tako se neatomarni podatki pojavijo v običajnih podatkovnih tabelah, ki ne sledijo normalnim oblikam podatkovnih baz. Pri transformaciji lahko glede na vrednost stolpcev novim stolpcem uvedemo čisto novo zalogo vrednosti.
- Čiščenje podatkov oz. standardizacija – to je predvsem pri strojnem učenju zelo pomembna metoda. Pri njej gre za to, da različno zapisane podatke, ki pa so vsebinsko identični, zapišemo v standardizirani obliki.

2.3 Analiza

Analiza podatkov je področje vsebinske obdelave podatkov (prejšnji fazi sta bolj podporne narave), ki zavzema največ različnih pristopov in metod. Podatkovno rudarjenje in vizualizacija sta, kljub veliko pristopom in metodam, relativno ozko definirani področji obdelave s splošno sprejetimi okvirji in cilji. Analiza kot taka je zelo fluiden pojem, ki je skoraj v vsakem viru definiran nekoliko drugače.

V splošnem gre pri analizi podatkov za to, da iz surovih podatkov pridobimo uporabno informacijo. Ker pa je pojem uporabne informacije (da je uporabna, mora odgovoriti na vprašanje, različnih vprašanj pa je nešteto) zelo širok, je tudi število metod analize podatkov zelo veliko. Pomembno je poudariti, da je analiza podatkov del skoraj vsake analize. Brez osnovne statistike bomo namreč težko definirali okvirje za sistem podatkovnega rudarjenja, vizualizirati pa moramo neki urejen sistem in ne le surovih podatkov (surove podatke redko sicer lahko vizualiziramo kot take, a to so zelo specifični primeri).

Pri analizi podatkov je torej bistveno vprašanje, kaj pravzaprav hočemo vedeti. Brez vprašanj si podatke lahko ogledujemo, ne moremo pa jih analizirati. Zaradi tega je pri analizi podatkov priprava za delo prav tako pomembna kot delo – v dobi računalnikov, ki brez težav izvedejo kompleksne statistične analize, celo še bolj. Hkrati je pomembno, da se skušamo čim bolj distancirati od predpostavk, ki jih v podatkih ni. V teoretičnem delu je takih predpostavk malo, pogoste postanejo v poslovnem svetu, kjer si marsikateri deležnik po

svoje razlaga situacijo in navadno želi, da rezultati analize njegovo razlago potrdijo (Milton, 2009).

Ko smo definirali problem in vzpostavili vprašanja, je na vrsti statistična analiza podatkov. Glede na tip podatkov, ki jih obdelujemo, in vprašanje moramo najti ustrezno statistično funkcijo ter z njo analizirati posamezno spremenljivko. V kompleksnejših primerih uporabimo multivariatno analizo (faktorizacijo, metodo glavnih komponent, razvrščanje v skupine, diskriminanto analizo), ki pa mora biti opisno utemeljena. V tem smislu je analiza podatkov pravzaprav klasična statistika (statistike brez podatkov sploh ne moremo izvajati). Pogosto se zgodi, da je preprosta statistična funkcija izvedena nad pravilno identificiranimi podatki pravzaprav boljša kot kompleksni algoritmi velikih podatkov. Prav tako se pogosto zgodi, da je v primeru kompleksnih algoritmov izgubljena rdeča nit. Tako pravzaprav dobivamo agregirane podatke iz velikih vhodnih zbirk, ne vemo pa, kako smo do njih prišli in če so sploh pravilni (Violino, 2017).

Kje je torej optimalna točka, ko se nam še splača razviti kompleksnejše algoritme namesto preprostejših, a morda dolgotrajnejših analiz? Vprašanje pravzaprav nima enotnega odgovora, saj je zelo odvisno od dejanske situacije, v kateri delamo. Kompleksnejše algoritme je primerneje razviti, kadar imamo vprašanja, na katera moramo odgovarjati redno, ali celo, ko moramo dobivati informacije v realnem času. Takšni sistemi so na primer v sistemih poslovne inteligence, ki stalno analizirajo situacijo podjetja (in v bolj naprednih sistemih tudi okolja) in na nadzorno ploščo (ang. dashboard) napišejo oceno situacije. Pogosta težava takih sistemov je ravno prej omenjena nepreglednost, ki se pogosto pojavi, ko uporabniki niti ne poznajo analitičnih algoritmov v ozadju, tako da se morajo odločiti, kako zanesljive so dane informacije.

Ne glede na to, kako avtomatizirano se bomo lotili analize, se moramo odločiti tudi, na kakšen način jo bomo izvedli. Tu govorimo izključno o načinu izvedbe, to je programski opremi, ki jo uporabimo za analizo. Analizo podatkov lahko izvedemo z veliko različnimi orodji, ki jih grobo ločimo v doma razvita orodja oz. skripte, ki so napisane v nekem programskem jeziku, in že razvita orodja, ki so lahko osebne ali komercialne narave.

Prednost razvoja lastnih skript je velika prilagodljivost. Lastne skripte lahko vedno prilagajamo trenutnim potrebam, in če situacija tako zahteva, napišemo popolnoma nove skripte. Primarna težava tega pristopa je, da zahteva poznavanje programskega jezika in osnov programiranja, prirejenega podatkovni znanosti, kar pa je težko osvojiti v kratkem času (tu je mišljeno nekaj ur do nekaj dni; večina ljudi, ki so vešči IT, osvoji osnove v nekaj tednih). V področju podatkovne znanosti se sicer uporabljajo manjše skripte, ne večja programska orodja. Navadno so pisane v jezikih R in Python, saj sta sintaktično nezahtevna in strukturirana primerno uporabi funkcij. Ni pa nujno, da se omejimo le na R in Python, saj so vsi programski jeziki v teoriji primerni za obdelavo podatkov (McKinney, 2017).

Komercialni in malo manj komercialni programi (tudi GPL odprtokodni programi) nam po drugi strani omogočijo hitrejšo obdelavo, ki zahteva manj specifičnih znanj, saj imamo na voljo celoten uporabniški vmesnik, poleg tega pa zna večina teh programov napake identificirati in predlagati način odprave. Orodja za podatkovno analitiko so razpeta v širok spekter zmogljivosti in zahtevnosti, poleg tega pa so na voljo tudi po različnih cenah. Za zahtevne analize moramo pogosto uporabiti drage komercialne programe, medtem ko lahko preprostejše analize opravimo tudi s preprostejšimi orodji, ki so običajno del programskega paketa v podjetju (na primer MS Access, Excel in sistemi SQL).

Kot smo že omenili, sta analitika in vizualizacija med seboj tesno povezani. Vizualizacija sicer lahko predstavi določene podatke v vizualni obliki (včasih je preprosta vizualizacija dejansko boljša od analitike), a večinoma se uporablja za kakovostno predstavitev agregiranih podatkov. Zaradi tega večina vizualizacij vsebuje neki del podatkovne analitike, ne glede na to, ali smo analitiko prvotno uporabili ali ne (vsaj na izvedbeni ravni). Zaradi povezanosti se bomo v nadaljevanju ukvarjali tudi z uporabo preprostejših orodij za podatkovno analitiko.

2.4 Podatkovno rudarjenje

Podatkovno rudarjenje je med stebri podatkovne znanosti pravzaprav relativno mlada disciplina, ki je postala bolj razširjena šele z razvojem računalništva, predvsem v smislu hitre izmenjave informacij (k čemur je pripomogel internet) in napredka v zmogljivosti strojne opreme, ki je omogočila napredne algoritme. Podatkovno rudarjenje je pogosto povezano z drugima dvema metodama obdelave podatkov, to sta strojno učenje in umetna inteligenca. Pri vseh treh metodah gre za visoko avtomatizirano obdelavo, ki je namenjena odkrivanju informacij, ki s klasičnimi metodami niso opazne. To pa ne pomeni, da so metode črna škatla. Marsikatera metoda podatkovnega rudarjenja in strojnega učenja pravzaprav temelji na multivariatni analizi. Za te metode vemo, da so matematično zelo zahtevne (saj moramo pogosto ponoviti na stotine operacij), kar pa za sodobni računalnik ni pretirano težavna naloga.

Podatkovno rudarjenje omogoča predvsem avtomatizacijo dveh analiz, ki sta v sklopu podatkovne analitike sicer možni (kar je logično, glede na to, da so algoritmi podatkovnega rudarjenja predvsem modeli, izpeljani iz kompleksnih zaporedij operacij podatkovne analitike), in sicer predvidevanje trendov in obnašanja ter iskanje prej neznanih vzorcev. Obe analizi sta za klasično analitiko pogosto preveč zamudni, tudi kadar je podprta z IT. Predvidevanje vzorcev in obnašanja je namreč izredno kompleksna multivariatna analiza, ki za vzorec uporabi običajno tudi po nekaj tisoč (pa tudi do nekaj milijonov) vrstic, ki bi jih bilo ročno pretežko obdelati, nato pa glede na izkušnje aplicira sestavljen model na nove vnose. Iskanje novih vzorcev je podobno kompleksna analiza, le da gre tu za izredno široko obsegajoče razvrščanje v skupine, ki prikažejo vzorce obnašanja, do katerih bi s klasično analitiko le stežka prišli (Thearling, 2008).

Povezava podatkovnega rudarjenja in vizualizacije je podobna kot pri podatkovni analitiki. Gre za to, da je, ne glede na kakovost obdelave, rezultat še vedno neko število oz. tabela, ki na prvi pogled običajno ne da bistvenih informacij. Zaradi tega pogosto pridobljene podatke še vizualiziramo, s čimer poskrbimo, da bo bistvena informacija opazna hitreje.

Ker pa se podatkovno rudarjenje ukvarja predvsem z izdelavo avtomatiziranih modelov, ki kasneje vrnejo le bistvene podatke, so vizualizacije pogosto veliko manj kompleksne kot pri analitiki. Pogosto gre za to, da z barvo poudarimo pridobljeni rezultat, ki je že izpisan na zaslonu. Če pa so pridobljeni podatki kompleksnejše narave (na primer pri iskanju novih vzorcev), lahko z dobro vizualizacijo poenostavimo večdimenzionalen pogled, ki bi bil s preprosto tabelo neintuitiven. Kot pri ostalih stebrih podatkovne znanosti je tudi tu malo stvari »vklesanih v kamen« – namesto tega se moramo odločati na osnovi posameznega primera.

2.5 Podatkovna znanost in neprofesionalci

Do sedaj smo govorili predvsem o podatkovni znanosti v očeh strokovnjakov, se pravi ljudi, ki poklicno delajo s podatki. Predvsem v manjših organizacijah, tu so mišljena predvsem manjša in srednja podjetja, takih strokovnjakov nimajo, zato se z analizo podatkov ukvarjajo predvsem ljudje, ki so zadolženi za druge naloge. To so po navadi zaposleni v vodstvu ali upravi (poslovni podpori, administraciji), ki v takih podjetjih šteje le nekaj ljudi. Ker niso izvedenci za naloge podatkovne znanosti, so običajno dovzetni za napake, ki bi se jim strokovnjaki hitro izognili.

To poglavje bo opredelilo predvsem pogoste napake v podatkovni znanosti in vizualizaciji, zato bo temeljilo na manj formalnih virih, del pa tudi na naših izkušnjah.

Ko se začnemo ukvarjati s podatki, se večina zaveda, da so podatki dejstva. Zaradi tega so dejstva tudi statistične agregacije, ki jih izpeljemo iz teh podatkov. To pa žal ne pomeni, da so dejstva reprezentativna za celotno populacijo, ki jo obravnavamo – nasprotno: pogosto se preveč omejujemo na to, kar imamo, in zato ne vidimo širše slike. Težava je pogosta pri nestrokovnjakih in pripelje do tega, da preveč fokusa usmerimo na skupino, ki jo pravzaprav že dobro poznamo. Čeprav tudi izkušeni podatkovni znanstveniki niso imuni za to past, je njihova ocena reprezentativnosti in pomembnosti podatkov navadno bolj natančna. V poslovnem (in tudi političnem) svetu past pogosto pripelje do t. i. prepričevanja prepričanih, namesto da bi skušali pridobiti nove stranke.

Naslednja past, ki jo strokovnjaki pogosto poudarjajo, čeprav se pogosto tudi sami ujamejo vanjo, je tunelski vid. To je pravzaprav skupina več pasti, vsem pa je skupno to, da se osredotočajo na raziskovanje podatkov namesto na procese podatkovne znanosti kot podporne procese primarni dejavnosti. Zelo pogosta težava se pojavi, ko analitik začne izdelovati analize, ne da bi imel v ozadju dobro definiran načrt ali vprašanja. Načrt je, kot smo že omenili, bistvenega pomena za kakovostno analizo, saj definira parametre, v katerih

mora analiza delovati. Brez njega so analize pogosto brez repa in glave, poslovni procesi pa nimajo večje vrednosti. Drugi tip pasti v sklopu tunelskega vida je pretirano osredotočanje le na podatke, ki jih imamo, in zanemarjanje ostalih elementov. To se pogosto kaže na dva glavna načina: precenjevanje in podcenjevanje podpornih podatkov. V prvem primeru skušamo napačno združiti velike količine podatkov iz mnogo različnih virov, ki so le šibko povezani s stvarjo, ki jo raziskujemo, v drugem pa se omejujemo le na podatke z neposredno povezavo. Pametneje je, da vire podatkov razvrstimo po relevantnosti za raziskovalni problem in analiziramo le tiste, za katere lahko logično opredelimo pomembnost (ostale dodamo le v primeru, da prva analiza ne da zadovoljivih rezultatov) (Shah, 2016; van Cauwenberge, 2015).

Tretja past, zelo pogosta med nestrokovnjaki, je potrditvena pristranskost. Tu gre za miselno zmoto, kjer skušamo s podatki potrditi svoje mišljenje. V teh primerih pogosto nismo pripravljeni sprejeti rezultatov, ki temu nasprotujejo, zato skušamo prirediti analizo (ali celo podatke), da bi dobili zase ugoden ali pričakovan rezultat. Temu se moramo nujno izogniti, vendar pa moramo vseeno paziti, da ne pademo v past slepega zaupanja v podatke, ki smo jo že definirali (van Cauwenberge, 2015).

Ker je vizualizacija del podatkovne znanosti, ki se najbolj približa ljudem, velja omeniti še nekaj bistvenih napak, ki se jim moramo izogniti, ko vizualiziramo pridobljene podatke. To ne pomeni, da nam ni treba upoštevati vodil, ki smo jih že opredelili, temveč so še dodatne stvari, na katere moramo biti pozorni.

Vizualizacije morajo vedno čim bolje povzeti podatke in njihov vpliv na realne situacije. Kot pri analizi moramo zato dobro definirati načrt in vprašanja ter s tem zakoreniniti način in obseg vizualizacije. Vizualizacije ne smejo biti preveč omejene, saj tako podamo nepopolno ali celo napačno informacijo, hkrati pa moramo biti pozorni na to, da niso preveč obsežne in kompleksne, saj je večina lahko pozorna le na omejeno število elementov. To pomeni, da moramo iz celotne analize izločiti bistvena vprašanja in poskrbeti, da za njih podamo izčrpne odgovore s karseda malo balasta. Vizualizacije pa so slabe, saj so pogosto preveč obsežne, s preveč hkrati podanimi informacijami, kar opazovalce po navadi le zmede. Prav tako moramo paziti, da so vizualizacije pregledne in jasne. Preveč podrobnosti na omejenem prostoru povzroči nepreglednosti in zelo verjetno je, da taka vizualizacija ne bo v korist nikomur.

Eksperimentiranje z različnimi vizualizacijami je, dokler ne pretiravamo, lahko zelo pozitivno sprejeto. A tu moramo spet paziti, da vizualizacija ne začne služiti sama sebi namesto kakovostnemu prenosu informacij. Zato moramo najprej poznati svoje tarče in vizualizacije prilagoditi nanje. Če je smiselno, je dobro eksperimentirati (Perez, 2012), dokler alternative služijo našim ciljem. Če pa ocenimo, da so primerne (ali celo primernejše) standardne metode vizualizacije, jih uporabimo (Marr, b. d.; Perez, 2012).

3 VIZUALIZACIJA

3.1 Pomen vizualizacije

Zadnji in pogosto tudi najpomembnejši steber podatkovne znanosti, vsaj s strani uporabnika, je vizualizacija. Tukaj gre v najmanjši meri za prirejanje podatkov v agregirane vrednosti – v večji meri gre za predstavitev podatkov v obliki, ki je na prvi pogled bolj razločna kot številke ali tabele. Ker je čas pogosto vir, ki je omejen, je najbolje, da podatke predstavimo v obliki, ki je čim hitreje razumljiva. Ker je danes dostop do velikih količin podatkov vedno na dosegu roke in ker so sistemi za zbiranje ter shranjevanje napredovali do točke, kjer je podatkov neobvladljivo veliko, lahko podatki postanejo bolj ovira kot rešitev. Pomembno je torej, da podatke predstavimo razumljivo, pravim ljudem ob pravem času. Le na tak način bomo iz njih pridobili najvišjo vrednost. Vizualizacija nam torej omogoča predstaviti podatke v intuitivni in učinkoviti obliki (Marr, 2017).

Vizualizacija podatkov, predvsem števil, je pravzaprav zelo star koncept, ki ga lahko zasledimo že v davnih dobah. Pravzaprav se je koncept števil in s tem kvantitativnih podatkov razvil iz preprostih vizualnih sistemov, na primer vozličkov, ki so jih stare civilizacije (Inki) uporabile za štetje. Vrsta vizualizacije je tako na primer tudi abak, računalo, ki za štetje uporablja kroglice. Način štetja s snopiči, kjer za vsak element narišemo črtico, pet ali deset pa jih »povežemo« v snop, se je ohranil do današnjih dni, saj nam omogoča učinkovito štetje, hkrati pa na ta način po koncu štetja zlahka razberemo števila. To sicer ni vizualizacija podatkov, saj ne pretvarjamo kompleksnih podatkov v razumljivejšo grafično obliko, v širšem pogledu pa gre za relativno podoben koncept (Friendly, 2008).

Če torej povzamemo idejo vizualizacije, gre za dejavnost (lahko bi rekli tudi metodo), s katero surove podatke pretvorimo v obliko, iz katere razberemo uporabno informacijo. V tem smislu vizualizacija podatke agregira in pretvori v manj obsežno ter razumljivejšo obliko. Pretvorba je tudi del podatkovne analitike, o kateri smo pisali že prej. V tem smislu sta vizualizacija in podatkovna analitika med seboj tesno povezani, po drugi strani pa zelo različni. Povezava je zelo odvisna od primera, zato bi težko našli neko strogo metodologijo, ki bi definirala, kdaj uporabiti eno, kdaj drugo in kdaj obe metodi.

Analitika je lahko v določenih primerih bolj uporabna od vizualizacije. V drugih primerih je vizualizacija boljša kot analitične metode. V tretjih primerih je najbolje uporabiti obe. Optimalna metoda je pogosto odvisna od vprašanja, ki smo si ga zastavili, in manj od tega, kakšne podatke imamo. Vzemimo na primer majhno zbirko podatkov, ki vsebuje štiri različna števila, in predpostavimo, da moramo vse operacije izvesti ročno, saj programska oprema za obdelavo podatkov pogosto avtomatizira marsikatero vmesno funkcijo. Če nas zanima aritmetična sredina števil (na primer povprečna vrednost nakupa), je analitika boljša rešitev, saj z vizualizacijo težko predstavimo povprečno vrednost. Lahko sicer izrišemo

stolpčni diagram in tam ocenimo povprečje, toda analitika nam omogoči natančnejši rezultat, ne vpliva pa preveč na dojemanje vsebine.

Enak primer z drugim vprašanjem situacijo obrne. Denimo, da nas ne zanima povprečje, ampak razmerja med posameznimi vrednostmi. Razmerja sicer analitično brez večjih težav izračunamo (kar tudi moramo narediti, če želimo natančne razlike), ampak težava pristopa je, da je rezultat (vsaj en nov stolpec v tabeli z novimi številčnimi podatki) nekoliko neintuitiven – na prvi pogled težko dobimo predstavo o dejanski situaciji. Po drugi strani lahko števila vizualiziramo s stolpčnim diagramom. Ta nam predstavi vrednosti števil z višinami stolpcev, kar omogoči, da že na prvi pogled ugotovimo razmerja med velikostmi števil. Tak diagram lahko narišemo tudi ročno, na primer če predpostavimo, da je vrednost števila višina stolpca v milimetrih.

V tretjem primeru lahko uporabimo analitično metodo hkrati z vizualizacijo. Spet lahko vzamemo primer štirih števil, vendar nas zdaj zanima njihov delež v celoti, ki naj bo seštevek vseh. Deleže lahko izračunamo analitično, s čimer dobimo vrednost v odstotkih (ali decimalni vrednosti, odvisno od potreb), ali pa izrišemo stolpce, le da jih tu združimo le v en stolpec z jasno izrisanimi mejami. Toda najbolj intuitivna metoda za razmerja je pogosto t. i. tortni diagram, ki deleže definira kot deleže kroga. Ročna izdelava tortnega diagrama zahteva nekaj znanja geometrije, sicer pa je relativno preprosta. Deleže v odstotkih pomnožimo s 360 in nato v krog izrišemo kote v velikosti dobljenih vrednosti. Tako razdelimo krog na sorazmerne deleže.

Glede na primere vidimo, da gre pri vizualizaciji za bolj intuitivno predstavitev podatkov. Slabost je, da jih brez zapisa dejanskih vrednosti le težko točno razberemo, a to pravzaprav ni ideja vizualizacij. Namen je, da pridobimo uporabno informacijo, kjer pa je nekoliko manjša natančnost podatkov lahko zanemarljiva. Prav tako ni nujno, da je vizualizacija predstavitev podatkov z diagramom. Ena najpreprostejših vizualizacij v poslovnem svetu je zapisovanje negativnih finančnih vrednosti (izgube ali dolga) z rdečimi številkami, saj so na prvi pogled veliko bolj prepoznavne kot majhen simbol »–« (minus) pred posamezno vrednostjo. Na podoben način lahko poudarimo tudi druge podrobnosti in pomembne elemente v podatkovni zbirki ali poročilu. V vseh primerih gre za vizualizacijo.

Poleg predstavitve podatkov v bolj intuitivni obliki, ki omogoči, da hitreje razberemo informacijo, je vizualizacija (sicer redkeje) uporabljena tudi za dejansko iskanje podatkov. Primer, kjer jo lahko uporabimo na tak način, je regresijska analiza. Računanje regresije in korelacijskega koeficienta je nezahtevno, a zamudno opravilo. Nekoliko manj natančna, a podobno zanesljiva metoda je uporaba točkovnega diagrama, ki na koordinatni sistem izriše točke v koordinatah vrednosti njihovih spremenljivk. Ker je korelacija dejansko povezanost obeh spremenljivk, lahko iz takega grafa hitro ocenimo koreliranost obeh spremenljivk, saj so pri visokem korelacijskem koeficientu zgoščene ob regresijski premici, medtem ko so pri nizkem koeficientu razporejene naključno. Kot vedno tudi tu žrtvujemo nekaj natančnosti.

Še bolj je taka vizualizacija učinkovita za iskanje osamelcev, ki je analitično zelo zamudno opravilo, s pomočjo točkovnega diagrama pa lahko brez večjih težav poiščemo točke, ki so daleč stran od navidezne regresijske premice (Rogelj & Marinšek, 2014).

Kot smo omenili, je vizualizacija zadnji steber podatkovne znanosti. Vizualizacija ustvari most med avtomatizirano podatkovno analitiko na eni strani in končnimi uporabniki na drugi. Vendar pa, kot smo tudi že poudarili, vizualizacija običajno izgubi na natančnosti, še posebej, ko gre za majhne razlike. Tu govorimo predvsem o natančnosti človekove zaznave, saj naše oči ne morejo razločiti majhnih razlik, če so te postavljene v veliki skali. Težava se pojavlja tudi v primeru, ko je podatkov preprosto preveč, da bi jih lahko razločno definirali na površini. Pri vizualizaciji smo vedno omejeni na neko specifično razločnost, ki jo lahko sicer povečamo, s čimer lahko zajamemo večjo količino podatkov, a je ne moremo povečevati v nedogled. Naše oči na neki točki ne bodo več sposobne opaziti dejanske razlike. Sodobne vizualizacije se s to težavo pogosto soočijo v primeru vizualizacije velikih podatkov (ang. big data), ki so zaradi razvoja sledljivih komunikacij in zmogljivih podatkovnih medijev vedno pogostejši (Chang, Yang & Procopio, 2016).

Po drugi strani moramo upoštevati tudi podatkovne zbirke, ki so ustvarjene avtomatsko, pogosto z orodji za snovanje in načrtovanje. Inženirji orodja za računalniško podprto načrtovanje in izdelavo uporabljajo že vrsto let, a je uporaba pogosto kompleksna, saj v orodjih obstaja ločnica med tehničnimi in funkcionalnimi zahtevami, definiranimi s specificiranimi pravili in vizualnim generiranjem. Če generiranje načrtov oz. prototipov avtomatiziramo, lahko dosežemo tudi delno avtomatsko preverjanje doseganja zahtev, vendar pa vizualna tridimenzionalna reprezentacija ni intuitivno povezljiva z načrtovalsko statistiko. Goguelin et al. (2017) definirajo predlog za nadzorno ploščo s serijo vizualizacij, ki vizualizirajo tudi vmesne stopnje pri avtomatiziranem razvoju, s čimer lahko elemente v tridimenzionalnem virtualnem prototipu povežemo z doseganjem funkcionalnih zahtev in vmesnih izračunov. Ta in podobni sistemi so sicer izredno zmogljivi, vendar pa kompleksni za razvoj in ozko strokovno usmerjeni, kar njihov razvoj in prilagajanje naredi nedostopno nestrokovnjakom.

V zadnjem desetletju se vedno pogosteje uporablja tudi t. i. interaktivna in realnočasovna vizualizacija. Pri obeh konceptih gre za sodobno uporabo zmogljivih osebnih računalnikov (in komunikacijskih kanalov), ki avtomatizirajo in racionalizirajo obdelavo podatkov. Analitiki ugotavljajo, da je predstavitev velikih količin podatkov (se pravi povprečna zbirka velikih podatkov) pravzaprav v prvih fazah boljša z neposredno vizualizacijo glede na parametre, ki so v posamezni branži najpogostejši. Interaktivne vizualizacije omogočijo kanaliziranje poizvedb skozi grafične predstavitve (kaj točno to pomeni, sledi v nadaljevanju), kar laikom omogoči hitro in preprosto pregledovanje podrobnosti. Realnočasovne vizualizacije omogočajo spremljanje podatkov v realnem času oz. bolje rečeno: s posodobitvami v enotnih časovnih enotah (ki so lahko od sekunde do nekaj minut, glede na potrebe) (Godfrey, Gryz & Lasek, 2016). Oba koncepta sta kompleksnejša za

izvedbo, zato ju navadno razvijajo specializirana podjetja, ki se ukvarjajo s podatkovno, predvsem poslovno, analitiko. Zaradi tega se večina podjetij z razvojem takih orodij ne ukvarja, ampak raje kupijo obstoječe rešitve. Večina programskih paketov za poslovno inteligenco temelji na nadzorni plošči (ang. dashboard), ki navadno vsebuje vizualizirane standardne poslovne izkaze. Naprednejše rešitve omogočajo tudi orodja za preprosto prilagajanje in izdelavo vizualizacij.

3.2 Metode vizualizacije

Kot smo že pisali, je podatkovna znanost, vključno z vizualizacijami, tako kot večina poslovne IT zelo odvisna od situacije. Zaradi tega tudi ne moremo definirati najboljše metode za vizualizacijo. Namesto tega moramo poiskati metodo, ki najbolje predstavi idejo vprašanja, na katerega želimo odgovoriti. Predvsem se moramo vprašati, kaj je ideja vprašanja, saj je glavno načelo vizualizacije čim bolj racionalna in učinkovita predaja bistvene informacije, ne pa kompleksen in izredno podroben zapis. Ker bomo v nadaljevanju primer obdelovali z orodji programskega paketa MS Office, bomo preprostejše primere tudi tu delno navezali na ta paket. Ne bomo šli v podrobna navodila, kako se metode izvedejo, temveč le v grob oris, s čimer bomo predstavili koncept posamezne metode vizualizacije.

Kot definirajo Chang, Yang in Procopio (2016), gre pri vizualizaciji za predstavitev podatkov z različnimi intuitivnimi elementi, ki jih zaznamo na prvi pogled, na primer barve, velikosti ali oblike. Večina vizualizacij podatkov, predvsem tiste, ki so namenjene predstavitvi že raziskanih podatkov, temelji na takih metodah.

Prva metoda, ki jo velja definirati in je tudi najpreprostejša, je označevanje besedila ali drugačnega zapisanega medija, kar pritegne pozornost na bistvene elemente. V nekaterih primerih s krepko, poševno ali podčrtano pisavo poudarimo pomemben podatek v sicer večji celoti. Tu računalnik ali podobna IT niti ni pomembna. Za boljše zaznavanje je dovolj, da v zapiskih nekaj besed poudarimo s krepkejšo pisavo ali pa podčrtamo bistveni stavek v učbeniku. Tudi avtorji lahko na tak način poudarijo bistvene elemente.

Ta način je zelo osnoven in pravzaprav ne pomaga pri pridobivanju dodatnih informacij, temveč le pritegne pozornost na bistvene podatke. Soroden sistem, ki dejansko omogoči pridobivanje informacije, je pogojno oblikovanje besedila. Tu gre za to, da neki del besedila ali polja v tabeli obarvamo z ustrezno barvo, glede na prej definirani ključ. Najbolj znan zapis, narejen na ta način, je označevanje negativnih števil (predvsem v poslovnih in računovodskih strokah) z rdečimi številkami. Na ta način na primer že takoj ugotovimo, ali je letni dobiček pozitiven ali negativen, ne da bi dejansko morali prebrati številko.

Z razvojem računalništva in obdelavo podatkov v realnem času, še posebej pa s sposobnostjo virtualnega zapisa, se je uveljavilo tudi sprotno posodabljanje in pogojno oblikovanje. Oba koncepta izvirata iz metode, opisane v prejšnjem odstavku, po drugi strani pa obe uporabljata edinstvene specifike, ki so neločljivo povezane z računalnikom. Ker računalniki zapisujejo

podatke v elektronski obliki, je podatke veliko lažje sproti spreminjati ali posodabljati, kot če so zapisani fizično (denimo na papirju). Zaradi tega lahko zapis primarnega podatka spremenimo vsakič, ko se v zbirki zgodi sprememba (na primer prihodke podjetja povečamo vsakič, ko stranka opravi nov nakup). Na ta način lahko sproti spremljamo spremembe v zbirki, kar nam daje bolj tekoč pregled nad dogajanjem, vendar moramo biti pozorni na to, da spremembe niso prepogoste. Če so, je najbolje, da izpis (se pravi tisto, kar vidi končni uporabnik) sistem posodobi v standardnih časovnih presledkih, na primer na eno ali pet minut (Cottam, Lumsdaine & Wang, 2013; Kh, 2016).

Zaradi dinamike IT lahko na podoben način uporabljamo tudi modele za obdelavo relativno standardiziranih podatkov. V teh primerih ustrezne podatke (ki morajo biti poenoteni glede na model) vnesemo v model, ki nato avtomatsko izvede analitiko.

V obeh primerih je zelo uporabna metoda pogojnega oblikovanja zapisa, ki je v tem primeru dejansko primarni element vizualizacije. Pri pogojnem oblikovanju gre za to, da za polja, ki jih pogojno oblikujemo, postavimo model (to je skupina pravil in definiranih dejanj), ki zapis v teh poljih oblikuje glede na definirana pravila. Ta so lahko zelo raznolika – navadno temeljijo na operatorjih, kot so »večje«, »manjše«, »enako«, »pozitivno-negativno« in podobno. Za vsak pogoj postavimo pravila oblikovanja glede na vrednost polja, programska oprema pa nato avtomatsko uporabi obliko zapisa glede na pogoj. Uporaba te sicer relativno preproste vizualizacije je zelo razširjena, čeprav je mogoče niti ne opazimo. Kot smo že omenili, gre lahko za zapis črnih ali rdečih številk pri letnem poročilu, lahko pa tudi z več barvami (odtenki med zeleno in rdečo) definiramo rast oz. padec obsega poslovanja. Podobno se lahko polja v vmesniku obarvajo rdeče, ko policist v poizvedbo vnese poteklo registracijo (to je sicer vizualizacija na ravni uporabniškega vmesnika, v kar se ne bomo poglabljali in kar pravzaprav ni pogojno oblikovanje, gre pa za podoben koncept) (Harkins, 2012).

Te metode so sicer v določenih primerih relativno zmogljive, niso pa med najbolj razširjenimi. Ne glede na vse imamo na koncu še vedno tabele s podatki, ki jih moramo prebrati posamično, četudi nam uspe označiti bistvene podatke. Zaradi tega so na področju vizualizacije uporabljeni vsi štirje elementi: barve, oblike, smeri in velikosti. Na tak način lahko veliko lažje predstavimo večjo količino podatkov v intuitivnem in učinkovitem sistemu. Najpopularnejše metode vizualizacije podatkov so tako diagrami, grafi in grafikoni (Kirk, 2016).

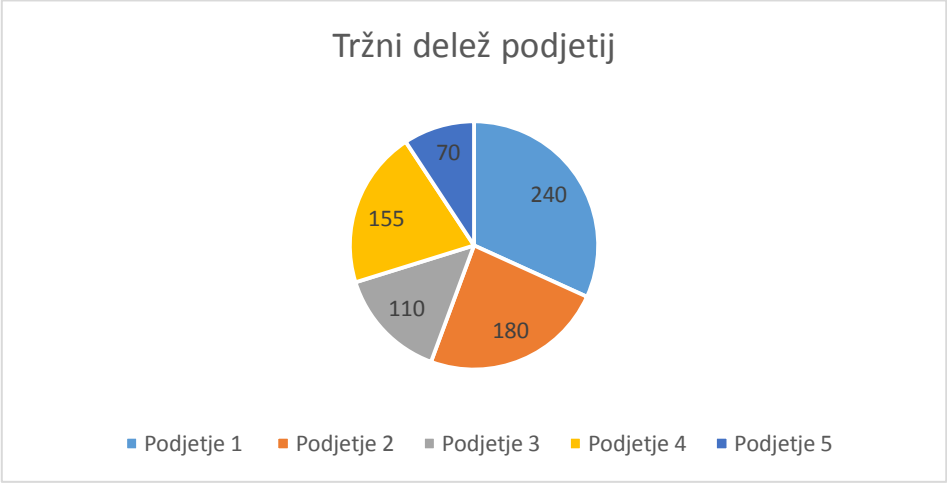
Marsikateremu laiku pomenijo ti trije izrazi eno in isto stvar, vendar gre v strokovnem izrazoslovju za tri različne stvari. Diagrami po navadi prikazujejo strukture ali poteke in imajo v specifičnih metodologijah, kjer se uporabljajo, predpisane elemente in legendo. Graf pomeni skupino povezanih točk (predvsem v matematiki), grafikoni pa so grafične ponazoritve relacij, razmerij in velikosti. Pri vizualizacijah podatkovnih zbirk se torej predvsem uporabljajo grafikoni in včasih grafi.

V podatkovni znanosti se za vizualizacijo uporabljajo predvsem grafikoni, ki so v sodobnem času vse pogosteje interaktivni, če je to smiselno. Diagrami se bolj uporabljajo v razvoju in načrtovanju. Zaradi tega se bomo osredotočili na grafikone.

Pravil za oblikovanje grafikonov je po eni strani veliko in kot marsikaj v podatkovni znanosti so zelo prirejena posamezni zahtevi. Po drugi strani je oblikovanje grafikonov lahko tudi relativno svobodno in dober razvijalec lahko za specifične potrebe razvije čisto svoj grafikon s specializiranimi pravili. Ta metoda je sicer uporabna, ampak jo je težko definirati z univerzalnimi smernicami, zato je ne bomo natančneje opisali (poleg tega je metoda napredne narave in ni ravno primerna za laike).

Univerzalnih grafikonov je veliko in vsak je bil razvit za določene zahteve. Po navadi grafikoni niso popolnoma zamenljivi med seboj, ne drži pa to vedno. Po drugi strani je glede na vprašanje, ki ga obravnavamo, skoraj vedno eden izmed tipov grafikonov najbolj primeren. Le redko se zgodi, da bi bila dva različna grafikona enako primerna (vrstični in stolpčni grafikon bomo obravnavali kot en sam tip, saj gre dejansko za isto stvar, le obrnjeno za 90 stopinj). Definirali bomo najuporabnejše grafikone in njihova pravila, potem pa še nekaj naprednejših.

Slika 1: Primer tortnega grafikona



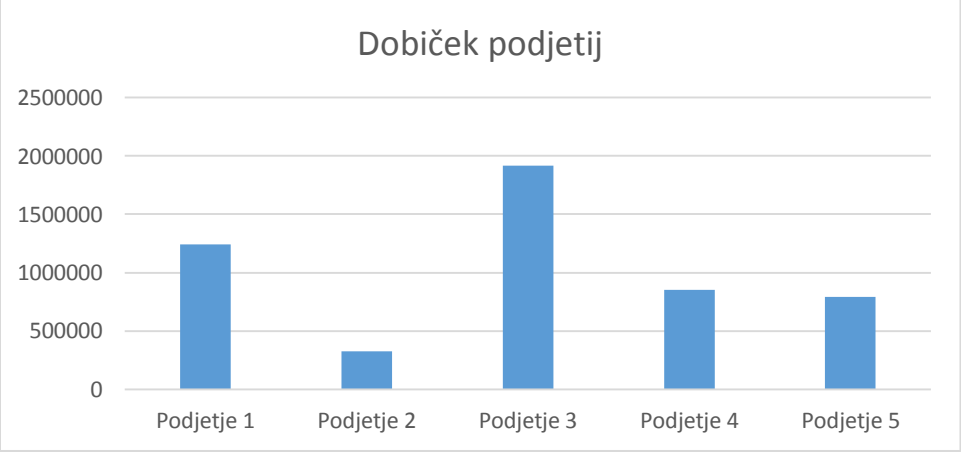
Najosnovnejši in eden najpogostejših grafikonov je t. i. tortni grafikon (Slika 1). Variacija tega grafikona je kolutni grafikon, ki navadno uporablja ista pravila, le da je votel. Tortni grafikon izraža deleže posameznih spremenljivk v skupni celoti, kar pomeni, da spremenljivke med seboj niso popolnoma neodvisne. Pri tortnem grafikonu izrišemo obroč in ga razdelimo na deleže, ki jih mora biti toliko, kot je spremenljivk. Glede na njen delež pripada posamezni spremenljivki sorazmeren izsek iz kroga. Tortni grafikon na tak način intuitivno predstavi deleže posamezne spremenljivke v skupni celoti in medsebojna sorazmerja. Slabost tortnega grafikona (poleg relativno omejene uporabnosti) je v obliki. Izseki kroga so ljudem hitro razumljivi, ne moremo pa jih postaviti drugega ob drugega, da bi lahko primerjali manjše razlike. Zato v primeru, da imamo spremenljivke, kjer so razlike

majhne, tortni grafikon ni najbolj intuitiven (tu gre za izgubo natančnosti, o kateri smo že pisali).

Stolpčni oz. vrstični grafikon (Slika 2) je drugi splošno uporabljeni grafikon, ki je namenjen predstavitvi spremenljivk, kadar so razlike med njimi bolj pomembne kot deleži v celoti (včasih je seštevek vseh spremenljivk tudi nesmiseln – v tem primeru so nesmiselni tudi deleži). Stolpčni grafikon vsebuje določeno število stolpcev, katerih višina je odvisna od vrednosti spremenljivke: tem višja je vrednost, tem višji je stolpec. S tem grafikon intuitivno predstavi razmerja med velikostmi spremenljivk, saj so (v primeru, da je grafikon dobro zasnovan) razlike opazne prek razlike v višinah (glej sliko 2). Stolpčni grafikon je pogosto tudi napačno uporabljen; pogosta napaka je njegova uporaba v primeru, ko so spremenljivke zaporedno povezane – v tem primeru je primernejši črtni grafikon. Druga pogosta napaka (ki mogoče niti ni napaka, ampak namerno manipuliranje) je zanemarjanje ničle. Grafikon lahko narišemo tako, da zanemarimo vrednost nič in za izhodišče uporabimo višje število, kar sicer lahko izboljša natančnost (pri velikih številkah z majhnimi razlikami), po drugi strani pa lahko preceni razlike, ki so tako videti večje, kot dejansko so. Zaradi tega je navadno bolje, da začnemo izhodišče pri nič ali pa na višje postavljeno izhodišče vsaj opozorimo. Vrstični grafikon je pravzaprav ista stvar, le da so bloki, ki ponazarjajo vrednosti, postavljeni vodoravno. Uporaba te izvedbe je primernejša pri večjem številu spremenljivk.

Tortni in stolpčni grafikon lahko združimo. Če so spremenljivke, ponazorjene v stolpčnem grafikonu, sestavljene in so njihovi elementi pomembni, lahko uporabimo t. i. segmentirani stolpčni grafikon. V tem primeru so stolpci grafikona razdeljeni v barvne elemente, kjer je velikost posameznega elementa odvisna od sorazmernega deleža elementa v posamezni spremenljivki.

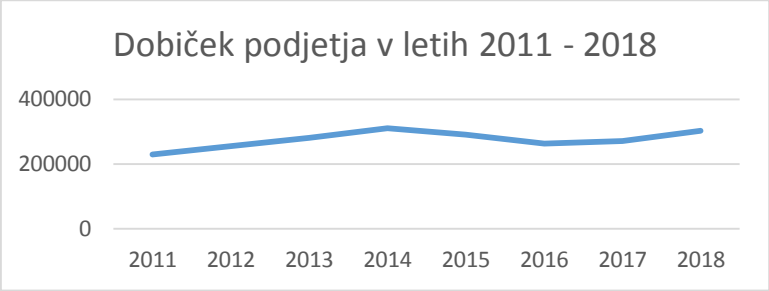
Slika 2: Primer stolpčnega grafikona



Kot smo že omenili, je v primeru, da med spremenljivkami obstaja neko logično zaporedje (na primer časovna vrsta), običajno bolj smiselno uporabiti linijski ali črtni grafikon (Slika

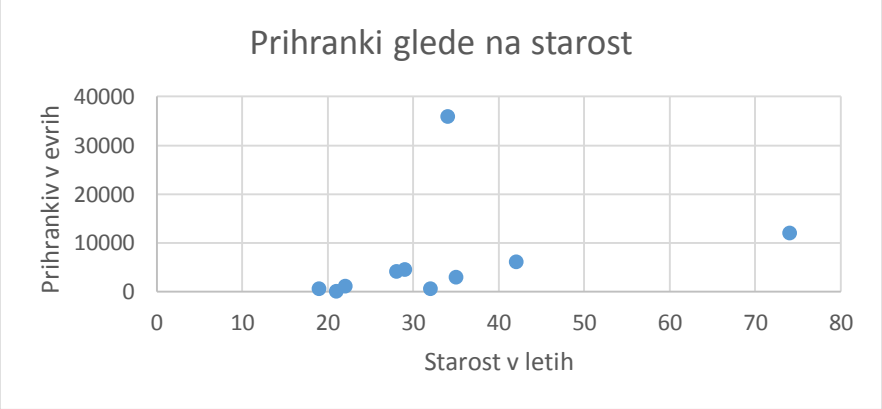
3). Gre za bližnjega sorodnika stolpčnemu grafikonu, saj gre za skupino spremenljivk, ki so med seboj logično povezane (v tem primeru si sledijo v zaporedju). V tem smislu gre za podatkovni zbirki, ki sta oblikovno pravzaprav enaki, šele pri metapodatkih (ko ugotovimo kontekst) pa lahko izberemo črtni grafikon. Črtni grafikon ne izriše stolpcev, ampak le točke na višini osi y, ki ustreza vrednosti posamezne spremenljivke, nato pa točke zaporedno poveže z linijo. Tak grafikon se uporablja v primeru, ko moramo slediti spremembam, ki so posledica določenega zaporedja, in ne neodvisnim spremenljivkam. V primeru, da se ne moremo odločiti med stolpčnim in črtnim grafikonom, je bolj smiselno uporabiti prvega, saj moramo imeti za logičen črtni grafikon logično zaporedje med spremenljivkami.

Slika 3: Primer črtnega grafikona



Črtni in stolpčni grafikon sta sicer vizualno ter pomensko različna, vendar oba temeljita na relativno enaki zbirki podatkov. Po drugi strani je raztreseni grafikon (Slika 4) vizualno podoben črtnemu, čeprav gre dejansko za dva pomensko zelo različna grafikona.

Slika 4: Primer raztresenega grafikona

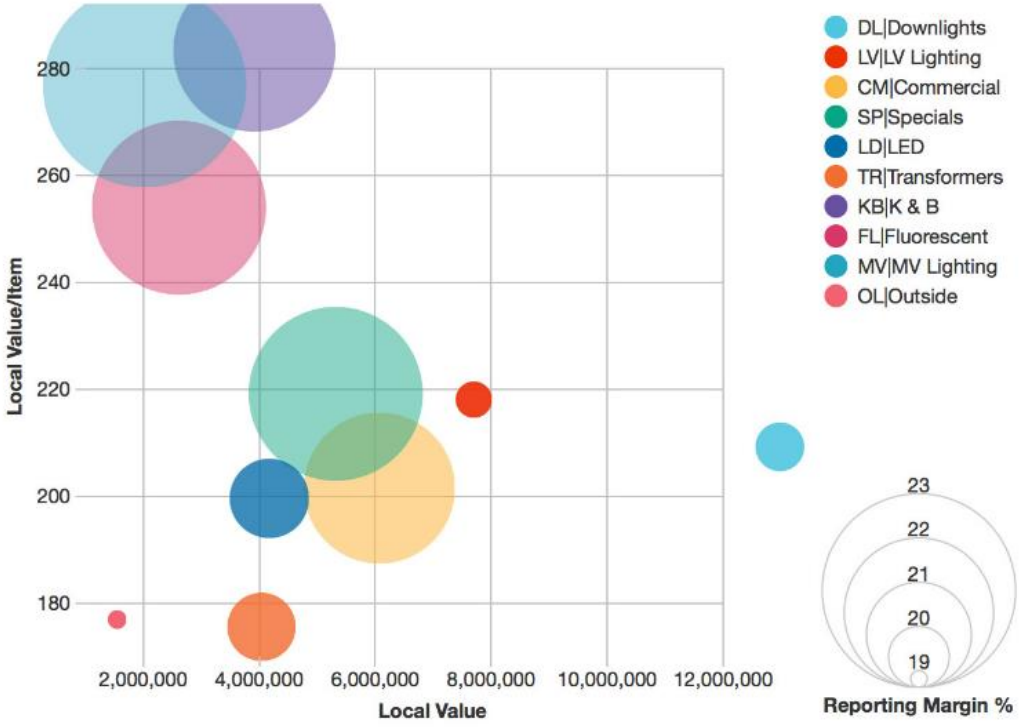


Raztreseni grafikon namreč vizualizira pare spremenljivk, ki so običajno neodvisne. V primeru, da je spremenljivk malo in med njimi obstaja močna korelacija, bi te točke lahko pogosto povezali med seboj in tako dobili črtni grafikon, vendar pa bi bila to s strokovnega vidika napaka. Pri črtnem diagramu si točke na osi x sledijo v logičnem zaporedju (na primer po letih), kar hkrati pomeni, da za vrednost na osi x ne obstajata dve različni spremenljivki. Pri raztresenem grafikonu pa to pravilo ne obstaja, zato imamo lahko poljubno število spremenljivk z enako vrednostjo x. Tak tip grafikona je uporaben vedno, kadar iščemo (ali

predstavljamo) korelacijo ali njeno odsotnost, saj skupina točk, ki so izrisane glede na vrednosti obeh spremenljivk, lahko dobro predstavi sledenje ali odstopanje od premice regresijske funkcije.

Včasih želimo uporabiti raztreseni grafikon, imamo pa tri bistvene spremenljivke. V tem primeru lahko uporabimo mehurčni grafikon (Slika 5). Gre za relativno podobno idejo. Točke razporedimo po grafikonu glede na vrednosti x in z, vrednost z pa vizualiziramo tako, da namesto točk izrišemo ustrezno velike mehurčke (obarvane kroge). Ta grafikon pravzaprav ni najbolj praktičen, saj v večini primerov obseg kroga vpliva na dojemanje vrednosti x in z, kar pa ni vedno zaželeno. V tem primeru moramo razmisliti, ali z uporabo mehurčnega grafikona pravzaprav ne ustvarjamo še dodatne zmede. Če ne moremo najti logične povezave med spremenljivko z in spremenljivkama x ter z, je bolje, da se mehurčnemu grafikonu izognemo, in raje naredimo več grafikonov.

Slika 5: Primer mehurčnega grafikona

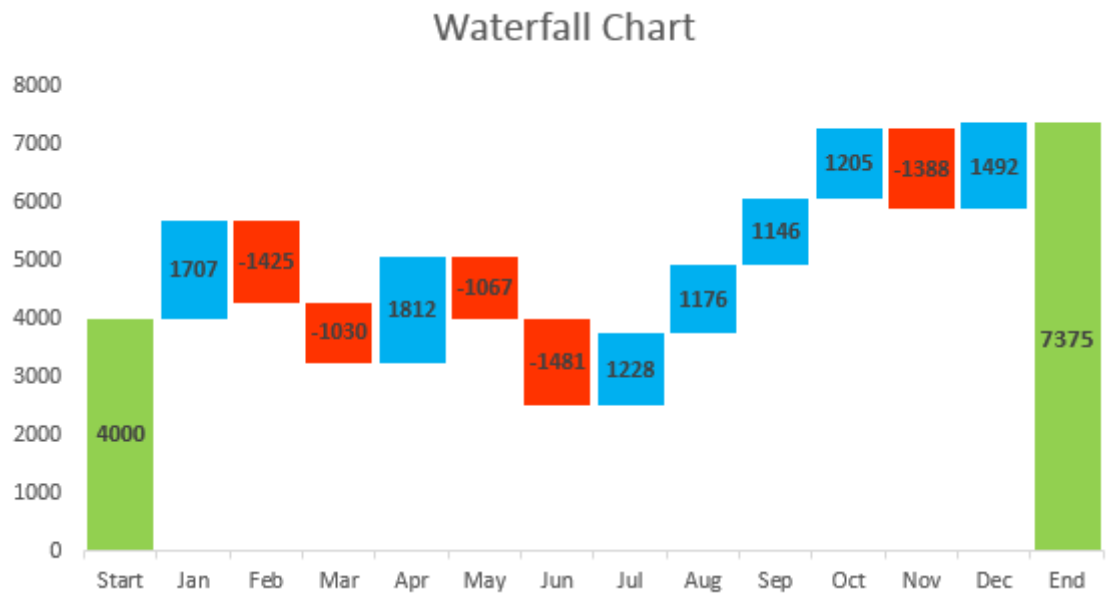


Vir: H. Gosper, Bubble chart - Documentation - Phocas Documentation, 2018.

Kot zadnjega naj omenimo grafikon slap (Slika 6), ki je vizualno sicer podoben stolpčnemu grafikonu, pomensko pa linijskemu in tortnemu. Njegov namen je prikazati elemente skupne vrednosti in vpliv posameznih elementov na skupno vrednost. Grafikon izdelamo tako, da skupno vrednost na začetku izrišemo kot stolpec, nato pa zaporedoma odsekane stolpce, ki zajemajo vrednosti od predhodne, potem pa proporcionalno glede na vrednost elementa (glej sliko 6. Na tak način vizualiziramo spreminjanje vrednosti glede na dodajanje ali odvzemanje posameznih elementov. Ta tip grafikona se pogosto uporablja v računovodstvu

za analizo in segmentiranje prihodkov ter odhodkov (definicije grafikonov povzete po Kirk (2016); Oetting (2018)).

Slika 6: Primer grafikona slap



Vir: E. Bespalaya, How to create waterfall chart in Excel (bridge chart), 2014.

Vsi grafikoni so jedro vizualizacije, ki pa je pogosto učinkovita le toliko, kolikor so učinkovite agregacije in poizvedbe podatkov, ki ležijo v ozadju. Ker grafikoni pogosto zajamejo le eno dimenzijo podatkov, ostale pa agregirajo, se lahko zgodi, da vsebujejo premalo informacij za kritične elemente ali pa preveč manjših elementov za kakovostno sliko. Zaradi tega se vzporedno s sistemi OLAP razvijajo orodja interaktivne vizualizacije. Interaktivna vizualizacija je namreč problematična, še posebej v primeru velikih zbirk, saj na osnovni ravni agregirane elemente vizualizacije ponovno pošljemo v poizvedbo, kar lahko v primerih počasnejšega delovanja podatkovne baze ustvari ozko grlo. Zaradi tega je razvoj usmerjen k vizualiziranju podatkov, naloženih v spomin, kot so kocke OLAP. Te pogosto podpirajo agregacijo in redukcijo posameznih elementov, kar se pri dobro razvitem vmesniku za vizualizacije brž pretvori v hitro odzivno delovanje (Chang, Yang & Procopio, 2016). Pomembnost avtomatskega razvrščanja poudarijo tudi Battle, Stonebraker in Chang (2013), ki opišejo sistem avtomatskega razvrščanja, ki so ga razvili sami. Ta namreč oceni največjo smiselnost vizualizacije, nato pa s statističnimi metodami agregira, vzorči ali pa filtrira rezultate na bistvene. Na tak način lahko sistem kakovostno vizualno predstavi ogromne količine podatkov brez izgube koristne informacije.

3.3 Preprosta orodja za vizualizacijo

Ko definiramo preprosta orodja za vizualizacijo, je predvsem pomembno, da definiramo pomen pogoja preprost. Ta namreč lahko, glede na kontekst, tudi tukaj pomeni več različnih

stvari. Preprosto orodje ne pomeni nujno, da je preprosto za uporabo. Vizualizacije namreč lahko ustvarimo praktično brez uporabniškega vmesnika, le z uporabo vrstičnih vnosov in nekaj programerskega znanja. Spisati program, ki izpiše nekaj vrstic (glede na število spremenljivk), pri katerih vsaka vsebuje določeno število znakov (lahko tudi drug, poljuben znak) glede na vrednost posamezne spremenljivke, je že preprosta vizualizacija. To ne pomeni, da je preprosta za uporabo, ampak je preprosta v smislu neodvisnosti od programske opreme.

Kot drugi element preprostosti lahko definiramo relativno malo zahtevanega predznanja za uporabo orodja. Tudi tu ni nujno, da dobimo kot odgovor uporabno orodje. Marsikatero komercialno orodje je zelo preprosto za uporabo, po drugi strani pa so v visokem cenovnem razredu, kar pomeni, da so za manjša podjetja, ki bi jih uporabljala le občasno, nemogoč strošek. Podobno lahko v to skupino štejemo tudi visoko specializirana orodja, ki so prirejena uporabi za točno določeno dejavnost. Za uporabo so preprosta, niso pa prilagodljiva ali skalabilna, kar pomeni, da so popolnoma neuporabna za organizacije, ki se ne ukvarjajo z dejavnostjo, za katero so programska orodja razvita.

Definirajmo torej preprosta orodja kot orodja, ki so dostopna in uporabna vsem ali pa vsaj večini organizacij, predvsem podjetjem. Najbolj razširjeno orodje je programski paket Microsoft Office in odprtokodni ekvivalent LibreOffice. Oba paketa vsebujeta programsko opremo za upravljanje in nadzor podatkovne baze ter orodje za obdelavo razpredelnic. Hkrati oba vsebujeta solidna orodja za izvedbo kakovostnih vizualizacij, pod pogojem, da uporabnik pozna osnove podatkovne znanosti in koncept vizualizacij. Naprednejših vizualizacij orodja ne podpirajo, zato je v takih primerih potrebno nekaj improvizacije.

Če se osredotočimo na orodja, ki so namenjena zgolj vizualizaciji, ostali elementi pa so le podporne narave, velja omeniti Tableau in Qlikview. V obeh primerih gre za visoko zmogljivi orodji in relativno preprosto uporabo, poleg tega pa nista nedostopni. Poleg njiju je Forbes v letu 2017 na vrh postavil še FusionCharts, Highcharts, Datawrapper, Plotly in Sisense (Marr, 2017). Orodij je še veliko več in skoraj nemogoče je našteti vsa, še posebej zato, ker imajo skoraj vsa analitična orodja tudi nekaj vizualizacijske podpore. Tako nam pravzaprav ne preostane nič drugega, kot da ocenimo svoje zahteve in sposobnosti (tako s stališča človeških virov kot financ) in poiščemo programsko opremo, ki nam najbolj ustreza.

3.4 Vizualizacija velikih podatkov

V sodobnem času je količina podatkov, ki jih zbere povprečna organizacija, veliko večja kot pred 20 leti ali še dlje. Zato moramo tudi pri področju vizualizacije ugotoviti in definirati slabosti, ki lahko nastanejo, če želimo obdelati preprosto preveč podatkov. Vizualizacija je namreč omejena s človekovo zaznavo in maksimalno natančnostjo zaslona oz. tiskalnika. To je trda omejitev vizualizacije, ki jo je treba rešiti na alternativne načine. V zadnjem času se je veliko raziskav ukvarjalo ravno s tem področjem, se pravi, kako preobsežne podatke

vizualizirati na način, ki je ljudem še opazen in razumljiv. Tu moramo spet razdeliti velikost na navpično (kadar imamo mnogo vnosov) in vodoravno (kadar imamo veliko atributov).

Največ raziskav in razvoja se ukvarja z vizualizacijo velikih količin podatkov, ki so veliki v višino. Razlog za to je, da se vizualizacija širokih podatkovnih zbirk lahko razdeli na več vizualizacij, kar je pogosto tudi boljši način, ne moremo pa tega narediti za zbirke, ki so velike v dolžino, kakršna je večina.

Raziskave bi lahko na področju visokih zbirk razdelili na reduciranje velikosti, ki ga pogosto izvedemo z različnimi metodami čiščenja in reduciranja (Wickham, 2013). Wickham predlaga metodo bin-summarise-smoth, kar bi lahko prevedli v poenostavi-agregiraj-zgladi. Trije koraki pravzaprav niso del vizualizacije, ampak priprave podatkov nanjo. Kot pravi Wickham, lahko metodo uporabimo za poenostavitev podatkovne zbirke, tudi če ne izvedemo vizualizacije. Metoda najprej poenostavi vnose s tem, da jih vse poenostavi na omejeno število celoštevilskih vrednosti, ki je po navadi vnaprej fiksno določeno. Nato v koraku »agregiraj« izračuna bistvene statistike iz poenostavljenih podatkov glede na poizvedbo. V fazi glajenja odpravimo variabilnost, ki nastane v prejšnjih dveh korakih, s čimer iz vizualizacije odstranimo moteče artefakte, ki navadno nimajo uporabne vrednosti. S to metodo skuša Wickham odpraviti ozka grla, ki nastanejo zaradi omejenosti vizualnega medija oz. tehnologije, na kateri temelji. Podobno razmišljajo Battle, Stonebraker in Chang (2013), ki prav tako uporabijo metode za reduciranje podatkov na raven, ki jo tehnologije za vizualne medije še lahko predstavijo. Njihov sistem temelji na agregaciji, vzorčenju in filtriranju, s čimer veliko podatkov reduciramo na raven, ki jo lahko predstavimo. Bistvo sistema, ki so ga razvili Battle, Stonebraker in Chang pa je, da je zasnovan za dinamično reduciranje, kar pomeni, da se sproti prilagaja širini poizvedbe. Tako vse tri metode pri vsaki spremembi ocenijo obseg reduciranja glede na optimalno natančnost.

Druge raziskave se ukvarjajo z zniževanjem zahtevnosti vizualizacij, ki nastanejo zaradi velike količine podatkov. Včasih se ne soočimo s težavo omejene resolucije, saj so podatki že na tak ali drugačen način agregirani – dober primer so toplotne mape (ang. *heatmaps*) (Pahins, Stephens, Scheidegger & Comba, 2017). V večini takih primerov nastane težava, saj moramo za vse spremembe v poizvedbi poslati zahtevo bazi. Sistem, ki se temu izogne, temelji na sistemu OLAP, kjer so podatki (oz. agregirana statistika) naloženi v spominu računalnika, kar drastično poveča hitrost. Pahnis definira zgoščene kocke (ang. *hashedcubes*), ki podatke definirajo v drevesih, kjer višje dimenzije definirajo veje kot posamezni element te dimenzije. Na podoben način Wang, Ferreira, Wei, Bhaskar in Scheidegger (2017) razvijejo gaussove kocke, kjer klasično statistiko nadomestijo optimalne multivariatne gaussove vrednosti glede na specifičen model. Ta sistem drastično zmanjša računsko zahtevnost, njegova slabost pa je omejen nabor modelov, saj so ti osnovani na standardnih agregacijah.

Drugi del raziskav se ukvarja z obvladovanjem podatkovnih zbirk, ki so velike v širino, kar pomeni, da imajo veliko število atributov, kakršen je obravnavan v nadaljevanju. Zaradi tega je izziv pri vizualizaciji bistveno večji. Vizualizacije so namreč pogosto omejene glede na število dimenzij, ki jih lahko predstavijo. Večina lahko predstavi dve ali tri (redko štiri) dimenzije, ki so definirane kot tipi spremenljivk ali bolje rečeno atributi v podatkovni zbirki.

Zaradi te omejitve moramo za vizualizacijo pet ali več spremenljivk združiti več različnih grafikonov skupaj. To lahko naredimo z večstopenjsko interaktivno vizualizacijo (ni pa nujno, da je interaktivna). Drugi način je, da več vizualizacij ali grafikonov logično razvrstimo na virtualno delovno površino (Goguelin et al., 2017), tretji pa, da jih skušamo standardizirati na eno spremenljivko in nato agregirati glede na njo. S tem ustvarimo skupino standardiziranih diagramov, po enega za vsak atribut. Metoda se imenuje tableplot (Tennekes, de Jonge & Daas, 2013). Slabost metode je, da zahteva agregacije glede na specifičen atribut, kar je včasih zaradi narave elementov v bazi težko izvesti.

Ne glede na to, katero metodo izberemo, pa moramo najprej definirati vprašanje. Vizualno iskanje vzorcev v širokih podatkovnih bazah je mogoče, a glede na splošna načela podatkovne znanosti ne najbolj zaželeno. Poleg tega lahko iskanje vzorcev tudi avtomatiziramo z metodami podatkovnega rudarjenja.

V primeru bolj raztresenih in neenakomernih podatkov, ki pa so vseeno enakovredno bistveni za pregled nad celotno situacijo, je ena boljših metod uporaba nadzorne plošče (ang. dashboard). Nadzorna plošča niti ni enotna vizualizacija. Pri tem sistemu gre za to, da na zaslon razvrstimo čim več podatkov, združenih v logične skupine, ki so nato po potrebi vizualizirani z grafikoni ali diagrami. Ko načrtujemo tako nadzorno ploščo, je poleg posamezne vizualizacije potreben tudi pregled nad celoto, drugače nas lahko plošča zmede in dosežemo ravno nasproten učinek (Rasmussen, Bansal & Chen 2009).

Če se že odločimo za vizualiziranje široke podatkovne baze v bolj poslovnem svetu, je mogoče celo najbolje, da vizualiziramo le atribute, ki so za nas pomembni, ostale pa skušamo smiselno agregirati (Kirk, 2016). V dveh metodah od treh (večstopenjska vizualizacija in nadzorna plošča) namreč naredijo točno to, na koncu pa vse grafikone le združijo v celoto. Edino tableplot standardizira in zgosti vse atribute hkrati.

4 PREDSTAVITEV IZBRANEGA PRIMERA

4.1 Metodologija vizualizacije predstavitvenega primera

Preden se lotimo praktičnega dela raziskave, bomo najprej definirali načrt poteka in metodologijo dela, saj je dobro definirana metodologija bistvena za dobro raziskavo.

V prvi fazi bomo z metodami, definiranimi v poglavju Zbiranje, tipi in kakovost podatkov, ovrednotili in opisali podatkovno zbirko. Ne glede na to, da je zbirka predvsem

demonstrativne narave, moramo poznati njeno kakovost, hkrati pa je to tudi dobra vaja za ocenjevanje zbirk v praksi. Hkrati s tem bomo opredelili osnovne metapodatke o zbirki. V naslednji fazi, še vedno analizi metapodatkov, bomo opredelili vsak podatkovni tip (stolpec) posebej, definirali njegov način zapisa vrednosti in določili njegovo potencialno uporabnost za različne analize. Zaradi velikega števila relativno podobnih podatkovnih tipov bodo stolpci združeni v logične skupine, še posebej tisti, ki definirajo podatke o posameznih igralcih.

V naslednji fazi praktičnega dela raziskave bomo opredelili hipotetičen poslovni primer in definirali vprašanja, ki bi jih bilo v takem poslovnem primeru primerno in dobro analizirati ter vizualizirati. Tu bomo opredelili načine vizualizacije velikih zbirk in ugotovili, kateri so primerni za našo zbirko in kateri niso (v tem primeru bomo tudi na kratko opredelili razloge, zakaj ne).

V četrti fazi bomo predstavili pripravo podatkovne zbirke, ki smo jo pridobili na spletnem viru Kaggle, saj je v formatu .csv, kar pomeni, da za natančnejšo obdelavo z orodji MS Office ni najbolj primerna, saj ta ne omogočajo dodatnega razčlenjevanja (Excel sicer odpre datoteko .csv, razčlenjeno v polja, ampak za široke raznolike zbirke je to premalo). Iz tako pripravljene baze bomo nato izluščili podatke za vizualizacijo in jih pripravili, da bodo zanjo primerni. Podatke bomo nato glede na poslovna vprašanja vizualizirali, in sicer z uporabo smiselnih metod, ki smo jih definirali v prvi fazi. S tem bomo odgovorili na vprašanje, ali so orodja v paketu MS Office dovolj zmogljiva za izdelavo takih vizualizacij.

Ker na splošno želimo ugotoviti, ali je vizualizacija z omejenimi viri mogoča, bomo v fazi eksperimenta prosili sodelavca, če lahko izvede vse tehnične faze od podatkovne baze naprej. Drugi del raziskovanega vprašanja je namreč, kako vešči uporabe orodij so ljudje, ki sicer imajo izkušnje z rabo orodij MS Office, niso pa izkušeni v uporabi metod podatkovne analitike in vizualizacij. Tu ne bo šlo za zasnovo vizualizacij (predpostavljamo, da se bo poslovni kolektiv na idejnem sestanku odločil, kaj želi v grafikonih), ampak za njihovo tehnično izvedbo glede na obseg orodij, ki jih omogoča paket MS Office. Delo sodelavca in težave, s katerimi se bo soočil, bomo popisali in predstavili v podpoglavju.

V zadnji fazi bomo pridobili še enega sodelavca, tokrat rednega igralca igre League of Legends (sicer ne na profesionalni ravni), in skupaj sestavili vsebinsko oceno kakovosti vizualizacij glede na podatke, ki so zapisani v podatkovni zbirki. Na ta način bomo ocenili, kako dobro smo z uporabo vizualizacij predali uporabne informacije glede na podatke, dostopne v zbirki.

4.2 Predstavitev zbirke podatkov

Zbirka podatkov, ki jo bomo uporabili za demonstrativni primer, je bila izbrana zaradi dveh glavnih razlogov. Prvi je primernost zbirke glede na obravnavani primer. Zbirka je bogata s podatki, ki pa so pogosto težko obvladljivi, predvsem zaradi velikega števila. Hkrati je več

atributov segmentiranih, kar pomeni, da so sestavljeni iz še več podatkov (kateri so to, je opisano v predstavitvi posameznih atributov). Drugi primarni razlog za izbiro zbirke je naše poznavanje obravnavane tematike, kar omogoči boljše predstavitve obravnavanega primera. Ideja namreč je, da vizualiziramo podatke o posameznih igrah, ki jih potem lahko taktiki in igralci v ekipah predelajo ter s tem izboljšajo svoj način igranja. To se pravzaprav sklada s podobnimi situacijami v podjetjih, saj v primeru primerjave ekipe in podjetja oboji analizirajo svojo primarno dejavnost z namenom njenega izboljšanja.

Zbirka je bila pridobljena na spletnem portalu Kaggle. Portal je združenje ljubiteljev in strokovnjakov podatkovne znanosti in v ta namen gosti večje število podatkovnih zbirk. Uporabniki lahko zbirke prosto obdelujejo in se o njih pogovarjajo z ostalimi uporabniki spletišča. Prav tako lahko z drugimi delijo svoje skripte, ki pa temeljijo na programskih jezikih R in Python, tako da v tem delu ne bodo v pretirano pomoč, razen kot morebitna idejna podlaga.

Zbirka je bila pridobljena z uporabo prepisovanja spletne strani oz. tako imenovane metode luščenja spletnih podatkov, ki smo jo opisali v teoretičnem delu. Podobno kot pri marsikaterem drugem športu so tudi za igre na tekmovalni ravni na voljo zelo podrobni podatki o poteku igre. Čeprav zbirka ne vsebuje vseh možnih podatkov, ki bi jih bilo o poteku dobro vedeti, nam da zelo dobro predstavo o tem, kako je posamezna igra potekala. Poleg tega Riot games, avtor igre League of Legends, ponuja tudi uradni API (vmesnik za namensko programiranje), ki omogoča realnočasovno zbiranje iger v poteku. Na tak način zberejo tudi podatke uradnih tekmovanj, ki so nato javno objavljeni.

Podatkovna zbirka je velika predvsem v širino. Vsebuje namreč 57 stolpcev, ki definirajo posamezne vidike igre. V zbirki je zabeleženih 7620 vrstic, a zaradi posodobitev z dodajanjem novejših podatkov tekmovanj se številka lahko tudi spremeni. Poleg osnovnih podatkov vsebujejo nekateri stolpci tudi strukturirane podatke, ki vsebujejo več podatkovnih elementov hkrati (vse primere bomo definirali v analizi metapodatkov). Če bi si želeli poenostaviti delo, bi lahko že razbite pridobili na spletišču kot podporno zbirko, ampak zaradi specifike dela bomo razbitje opravili sami.

Pomemben element, ki ga moramo upoštevati pri analizi podatkov in je v našem primeru lahko bistvenega pomena, še posebej, če analiziramo agregacije in ne posamezne vrstice, je konsistentnost podatkov v času. Ni nujno, da gre v primeru nekonsistentnosti podatkov za napake, gre pa za spremembe v specifiki opazovanega objekta, v tem primeru igre. Za razliko od marsikaterih drugih športov ali iger se v e-športih pravila spreminjajo veliko pogosteje (pa tudi v ostalih športih lahko nastanejo take anomalije, denimo v primeru »pravila zlatega gola«, če analiziramo nogometne ali hokejske tekme v zadnjih 30 oz. 20 letih). Zaradi relativno pogostih sprememb lahko v analizi daljšega obdobja nastanejo anomalije (tudi zaradi drugih razlogov, na primer ustanovitve ali razpustitve ekipe). V primeru takih anomalij, ki so sicer bolj nevarne v primeru podatkovnega rudarjenja, je smiselno opazovati

tudi zgodovino sprememb. Tako se na primer junak Ekko ne bo nikoli pojavil v zapisih pred 28. 5. 2015, saj je bil šele takrat dodan v igro (Emptylord, 2015). Podobno se zapisu, ki označuje posamezne zmage nad zmaji, spremeni struktura v obdobju, ko je Riot igri dodal različne tipe zmajev, in sicer tako, da poleg ekipe in časovne označbe doda še tip zmaja, ki ga je ekipa premagala.

4.3 Igra League of Legends

V praktičnem delu bomo obravnavali podatkovno zbirko, ki vsebuje statistiko posameznih iger na tekmovalni ravni računalniške igre League of Legends. Omenili smo, da je poznavanje strokovnega področja vsebine podatkovne zbirke pomembno pri analizi, saj nam omogoča razumevanje problematike. Ker se ne ukvarjamo z vsebinsko analizo, ampak uporabljamo podatkovno zbirko kot preizkusni koncept analize in vizualizacije z omejenimi viri, lahko problematiko predstavimo le na kratko. Po drugi strani moramo poznati osnove igre, saj le tako lahko pravilno ocenimo metapodatke, ki so nujni za kakovostno oceno zbirke, kasneje pa tudi analize.

Igra League of Legends je trenutno (verjetno) najbolj igrana računalniška igra na svetu. Razvila jo je razvijalska hiša Riot games na osnovi modifikacije Defense of the Ancients za Warcraft 3. Koncept igre se imenuje MOBA (ang. Massive multiplayer battle arena oz. velika večigralska bojna arena). Temelji na variaciji realnočasovne strategije, pri čemer za razliko od klasičnih realnočasovnih strateških iger vsak izmed soigralcev nadzira le en (t. i. junaški) lik. Ostali so računalniško vodeni z zelo preprosto umetno inteligenco. Igra je predvsem zaradi brezplačnosti postala v zelo kratkem času zelo popularna, zaradi visokega sposobnostnega stropa pa je hitro postala popularna tudi na profesionalni tekmovalni ravni. Svetovni pokal v igri League of Legends je danes primerljiv z marsikaterim drugim pomembnim športnim dogodkom, čeprav ga množični mediji (še) ne tako spremljajo.

Koncept igre je relativno preprost. Nasproti si stojita dve ekipi, v vsaki je pet igralcev (ekipe so na amaterski ravni sestavljene naključno ali pa gre za skupino prijateljev, medtem ko imajo profesionalne ekipe zaprte skupine in stroge pogoje za vstop). Naloga posamezne ekipe je, da podre nasprotnikovo jedro (v izvirniku se imenuje Nexus), ki stoji na sredini nasprotne postojanke. Za dosego tega se morajo prebiti mimo računalniško vodenih vojščakov, varovalnih stolpov in nasprotne ekipe. Ekipa zmaga, ko podre nasprotnikov Nexus.

Na začetku si vsak igralec izbere junaka, ki ga bo nadzoroval med igro. Igralci imajo nadzor le nad junakom, vse ostalo je vodeno s strani relativno preproste umetne inteligence. Junaki na grobo spadajo v pet kategorij, vendar lahko posamezen junak pokriva več kategorij. Kategorije so grobo korelirane s področjem delovanja v prvih fazah igre. Najbolj vzdržljivi junaki začnejo na gornji poti, junaki z visokim potencialom za močne, hitre napade začnejo na vmesni liniji, na spodnji liniji pa v paru delujeta podporni junak in napadalec, ki se

osredotoča na konstantne napade. Peti igralec nima določene poti, temveč igra v džungli, to je področje med tremi potmi, in se osredotoča na priložnostne napade, ki imajo učinek presenečenja. V poznih stadijih igre se ekipe navadno strnejo in skušajo napasti nasprotnikove šibke točke, da bi čim prej prišle do jedra.

Junaki med igro dobivajo izkušnje in zlato. Čim bolj je igralec aktiven, tem hitreje akumulira zlato in izkušnje. Dodatno zlato na primer dobi, ko premaga nasprotnikovega vojaka – junaka ali podre stolp. Z izkušnjami lahko igralci izboljšajo sposobnosti svojih likov, vendar le za posamezno igro. Zlato uporabijo za nakup predmetov, ki so lahko varovala, ki igralca obvestijo v primeru, da se jim približa nasprotnik, zdravilni napoji pa tudi predmeti, ki še bolj izboljšajo moč lika, hkrati pa mu lahko dajo tudi določene dodatne pasivne sposobnosti. T. i. gradnja junakov je eden jedrnih konceptov igre, v katero pa se ne bomo spuščali, saj v podatkovni zbirki niti ni zabeležena. Dovolj je, da poudarimo, da ima ekipa, ki akumulira več zlata, prednost pred nasprotnikom. Kot bistven element igre (ki je tudi zabeležen v podatkovni zbirki) moramo omeniti tudi nevtralna sovražnika. Prvi je zmaj, ki se prvič prikaže po poteku dveh minut in pol, vsak naslednji pa po šestih minutah od trenutka, ko je prejšnji premagan. Ekipa, ki ga premaga, dobi trajno izboljšavo, ki se z vsakim dodatnim premaganim zmajem izboljšuje, zato je napad na zmaja pogosto pomemben element, še posebej v začetnih fazah igre (v žargonu temu rečemo prvi in drugi zmaj). Drugi nevtralni sovražnik je baron Nashor. Baron je veliko nevarnejši od zmaja, po drugi strani pa se prikaže veliko kasneje, šele po 20 minutah igre, nato pa vsakih sedem minut od trenutka, ko je prejšnji premagan. Izboljšava, ki jo dobi ekipa, ko premaga barona, je veliko bolj intenzivna od tiste, ki bi jo pridobili z zmajem, a traja le 210 sekund (tri minute in pol), zato morajo ekipe načrtovati napad na barona, da mu sledi pritisk na nasprotnika, preden izboljšava mine. Oba sovražnika sta sicer pasivna in bosta ignorirala igralce, dokler jih ne napadejo. Zmago si prišteje ekipa, ki zmaju oz. baronu zada končni udarec, zato mora ekipa, ki se spopade z zmajem, paziti na nasprotnike, saj lahko v boj vrinejo končni udarec. Taktika se imenuje kraja zmaja (ang. steal the dragon) in je zelo popularna predvsem v nižje rangiranih igrah, medtem ko je pri profesionalcih redka, saj jo je relativno lahko onemogočiti.

Bistveni element razumevanja podatkovne zbirke je tudi proces izbire junakov v tipu igre Draft (kar bi lahko smiselno prevedli kot vpoklic). Način izbire junakov na tekmovalni ravni je nekoliko drugačen od standardne igre, a ker naša zbirka temelji na tekmovalnih igrah, se bomo osredotočili prav nanj. Vsaka ekipa dobi na začetku pet prepovedi. Kapetana ekip v prvi fazi porabita vsak po tri. To pomeni, da kapetana ekip izmenično izbereta junaka (v igri je trenutno 140 različnih junakov, verjetno pa se bo, ko bo magistrsko delo zaključeno, številka povzpela na 141), ki ga v fazi izbora ne bo smela izbrati nobena ekipa. Ko je šest junakov prepovedanih, se začne druga faza. V njej si prvi trije igralci izberejo svojega junaka. Prvi izbor ima eden izmed igralcev ekipe, ki ni imela prvega izbora v prvi fazi (imenujmo jo ekipa 1). Temu sledita dva izbora nasprotne ekipe (ekipa 2), dva izbora ekipe 1 in na koncu tretji izbor ekipe 2. Po tej fazi je šest junakov prepovedanih, trije igralci v vsaki ekipi pa so si izbrali svoje junake. Tretja faza je ponovno faza prepovedi. Kapetana

ekipe zdaj izmenično porabita še ostali dve prepovedi. Prepovedi v tej fazi so bolj taktične narave, glede na pozicije, ki jih morata ekipi še zapolniti. Četrta faza je namenjena izboru junakov za preostale štiri igralce. Ekipa 1 začne z izborom, nato ima ekipa 2 dva zaporedna izbora. Kot zadnji si izbere svojega junaka peti igralec ekipe 1.

5 ANALIZA IN VIZUALIZACIJA ZBIRKE

5.1 Opis metapodatkov, strukture in ocena kakovosti zbirke

Preden začnemo podatke analizirati in predstavljati, moramo najprej preveriti, kako so sestavljeni, njihovo strukturo, razmerja in kakovost. Brez tega ne moremo biti prepričani, ali so podatki uporabni in reprezentativni. Čeprav bomo obravnavali le demonstrativni primer, moramo priprave vseeno opraviti, vsaj zato, ker glede na strukturo podatkov nato definiramo tipe analize in vizualizacije. Poleg tega demonstrativni primer temelji na situaciji, ki je logična tudi v realnem življenju, zato je dobro, da podatke dobro povežemo s problematiko, opisano v prejšnjem poglavju.

Podatkovna zbirka, ki jo obravnavamo, ima 57 stolpcev. Nekateri izmed njih so sestavljeni podatki, pridobljeni s spletne strani za analitiko posameznih iger. Vrstice, ki jih bomo podrobneje opisali kasneje, lahko grobo razdelimo v štiri skupine. Prva skupina so podatki o igri in zajema splošne podatke, kot so vir, področje, leto, sezona in podobno. Druga skupina podatkov definira ekipe (ime, oznako) in njihove igralce. V tretjo skupino lahko uvrstimo podatke o igralcih in junakih, ki so jih igrali. V četrti skupini so časovni preseki količine zlata in ostali podatki o poteku igre (zmaji, baron, stolpi in podobno).

Večina podatkov je zapisanih v podatkovnem tipu string (niz znakov), kar pomeni, da jih bomo sprva težko aritmetično obdelali. Nekateri podatki so zapisani v numeričnem tipu, vendar so za aritmetično obdelavo nesmiselni. Poleg tega imamo en tip podatka, ki je niz numeričnih elementov – tu gre za večino zapisov časovnega poteka. Pri tem tipu podatka gre za niz zaporednih numeričnih podatkov, ki je smiselno uporaben za kvantitativno analizo pod pogojem, da ga ustrezno pretvorimo v zaporedje numeričnih elementov.

Na spletni strani Kaggle (2018) lahko v razdelku Column metrics pridobimo nekaj podpornih podatkov, ki nam omogočijo lažjo analizo kakovosti podatkovne zbirke. Kot metodo za oceno kakovosti bomo izbrali študijo Askhama et al. (2013), ki smo jo opisali v teoretičnem delu. V ta namen bomo ovrednotili svojo podatkovno zbirko glede na šest kriterijev, ki jih študija definira, čeprav glede na metapodatke lahko grobo ocenimo, da je kakovost podatkovne zbirke dobra.

Prvi kriterij je popolnost zbirke, kar pomeni, da morajo biti vsa polja v njej zapolnjena. To lahko preverimo v podatkovni bazi (ki smo jo ustvarili z uvozom datoteke .csv v programsko orodje MS Access) z uporabo funkcije »count NULL«, lahko pa tudi uporabimo razdelek Column metrics, kjer nam stolpec »Null« pove, da je odstotek praznih polj v vseh stolpcih

nič. To pomeni, so vsa polja zapolnjena s podatki. V kriteriju popolnost torej dobi podatkovna zbirka najvišjo oceno, to je 100% popolnost.

Drugi kriterij je edinstvenost, ki v splošnem pomeni, da v podatkovni zbirki ni dvojnikov. To moramo načeloma omogočiti z dobrim pristopom pridobivanja podatkov, ki je bil v tem primeru korekten. Hkrati lahko primerjamo število vnosov edinstvenega podatka (v tem primeru gre za podatek Address, ki definira naslov spletne strani, kjer so objavljeni podatki specifične igre). Število edinstvenih vnosov je 7620, ki se sklada s številom vrstic (tudi 7620). To pomeni, da vsaka vrstica definira edinstven osebek (v tem primeru posamezno igro). Hkrati smo med uvozom podatkovne zbirke v MS Access definirali stolpec Address kot primarni ključ in uvozili brez napak, kar pomeni, da so vrstice res edinstvene.

Pravočasnost je element, ki definira konsistentnost podatkov glede na časovne okvirje in njihovo ustreznost glede na analizo, ki jo želimo opraviti. Tu se pojavijo prve težave. Kot smo že omenili, razvijalci igre redno dopolnjujejo igro in spreminjajo manjše vidike, kot so ravnotežja med junaki. Zaradi tega so starejši podatki v določenih primerih lahko nezanesljivi, predvsem kar se tiče izbire junakov. Poleg tega so se v obdobju zbiranja podatkov spremenila tudi določena pravila igre (na primer uvedba različnih tipov zmajev), kar moramo upoštevati. V naši analizi to ne bo predstavljalo težav, saj se bomo osredotočili na analizo posamezne igre (to pomeni, da bomo analizirali le eno vrstico). Spremembe bi morali upoštevati, če bi se na primer odločili analizirati navade in tehnike posameznega igralca ali specifiko posameznega junaka.

Kriterij veljavnosti in natančnosti bomo združili, saj se bomo osredotočili predvsem na analizo z uporabo že obstoječih podpornih zbirk in manj na lastno analizo (ki bi bila v tem primeru zamudna in izven koncepta vsebine). Kot veljavnost definiramo ustrezanje zapisov poslovnim pravilom. Ker tu poslovna pravila pravzaprav niso uradno definirana, bi težko rekli, da posamezen zapis ne ustreza pravilom. Poleg tega gre v marsikaterem podatkovnem tipu, ki ne definira časovnih presekov, za posredno gledano prost vnos, ki mu težko določimo fiksno strukturo. Po drugi strani bi lahko ocenili veljavnost časovnih presekov, a le, ko jih razbijemo na posamezne časovne elemente, pa še tam bomo morali biti previdni, saj se lahko razlikujejo glede na število podelementov, potek in trajanje igre. Po drugi strani lahko s pregledom vira ugotovimo, da obstajajo podporne zbirke, ki so razbitje že opravile, in z njihovim pregledom ugotovimo, da so segmentirani podatki veljavno zapisani (tu predpostavimo, da med razbitjem avtorji niso odpravljali napak ad hoc). Natančnost kot soroden element bi lahko preverili tako, da bi z izvirnega vira sami prepisali (avtomatsko ali celo ročno) nekaj testnih iger in jih primerjali z dejanskimi vrednostmi v podatkovni zbirki. Kot bolj agresivno metodo bi lahko opredelili preverjanje vseh vrstic z izvornim virom, kar bi potrdilo (ali ovrglo) natančnost zbirke, dodatna prednost pa bi bila, da bi lahko vse napačne vnose kar avtomatsko popravili na pravo vrednost. Ker je primer le demonstrativne narave in nima dejanske vsebinske vrednosti, je kriterij natančnosti podatkov v tem primeru pravzaprav nekoliko brezpredmeten.

Zadnji kriterij, ki ga definiramo, je konsistentnost. Ker gre za eno samo podatkovno zbirko, pridobljeno s prepisovanjem strukturiranih informacij, lahko že takoj na grobo ocenimo, da je konsistentnost visoka. Kot dodaten dokaz lahko definiramo pregled podpornih zbirk, ki so vse enakomerno strukturirane, in možnost, da lahko vse strukturirane stolpce razbijemo na enak način, kjer je to smiselno ali potrebno.

5.2 Opis metapodatkov posameznih atributov

Metapodatke, ki se tičejo celotne zbirke, smo že analizirali, a za kakovostno poznavanje moramo poznati tudi metapodatke posameznih atributov oz. stolpcev. Brez tega namreč težko izvedemo kakršno koli smiselno analizo, ki je podlaga za vse vizualizacije.

Pregledali bomo vsak atribut posebej ter jim opredelili strukturo in pomen glede na obravnavano problematiko. Nekatere atribute bomo združili, saj so izredno podobni in bi bilo nesmiselno definirati vsakega posebej (združili jih bomo le v opisu, ne pa dejansko v zbirki oz. bazi). Atributov je 57, zato jih bomo za boljši pregled grupirali v tri splošne skupine.

V prvi skupini so atributi, ki definirajo osnovne podatke o igri. Ti podatki so pomembni predvsem, če želimo igre agregirati in analizirati po posameznih osnovnih parametrih, kot so liga, obdobje ali tip tekmovanja. Agregacije lahko nato tudi med seboj primerjamo. Atributi, ki definirajo osnovne podatke o igri, so:

- Naslov (ang. Address) je spletni naslov strani, od koder je bila vrstica prepisana. Vsaka spletna stran ima zaradi specifike infrastrukture svoj edinstveni naslov, ki lahko edinstveno definira vrstico. Ker je to tudi edini atribut, ki je edinstven v vsakem primeru (imamo tudi druge atribute, ki so v praksi edinstveni, ni pa to nujno), smo ga uporabili kot primarni ključ. Ker gre tu le za naslov strani, ga v analizi iger ni smiselno uporabljati.
- Liga (ang. League) definira, v kateri ligi (označeno s kratico lige) je bila odigrana igra. Lige so lahko regionalna prvenstva, svetovno prvenstvo ali pa specifični dogodki. Polje je pomembno, če želimo analizirati samo eno specifično ligo (predvsem uporabno pri regionalnih ligah) ali pri primerjavi posameznih specifik lig.
- Leto (ang. Year) definira leto, v katerem je bila igra odigrana. Zapisano je kot število (na primer 2015). Skupaj z atributom Sezona je pomemben, ker le na ta način lahko ugotovimo verzijo igre, ki so jo igrali igralci, kar je lahko pomembno v kasnejši analizi.
- Sezona (ang. Season) definira sezono, v kateri je bila igra odigrana. Sezona ima lahko le dve vrednosti: Summer (poletna) in Spring (spomladanska). Sezona, podobno kot leto, definira obdobje, kdaj je bila igra odigrana, in je predvsem pomemben element, če moramo paziti na verzijo igre.
- Tip (ang. Type) definira tip tekmovanja. Tipov je pet in so zapisani z besedo (ang. Season, Playoffs, International, Promotional, Regional). Podatek je bistven v primeru,

da primerjamo različna tekmovanja med seboj ali pa če se želimo omejiti na le en tip tekmovanj.

V drugo skupino lahko uvrstimo podatke o poteku iger na ravni skupine. Ti so pogosto tesno povezani (tako elementarno kot vsebinsko) s podatki v tretji skupini, to so podatki o igri posameznih igralcev. Zaradi tega smo uvrstili v to skupino le podatke, ki definirajo podatke o poteku igre, ki se tičejo celotne ekipe. To so:

- Oznaka (modre/rdeče) ekipe (blue/redTeamTag) sta dva različna stolpca, eden za rdečo in eden za modro ekipo. Tag je tričrkovna oznaka ekipe, ki je v tisti igri igrala na posamezni strani. Ekipe imajo večinoma enotne skupine igralcev, zato je atribut zelo uporaben pri marsikateri analizi, od načina igre do prehodov igralcev med ekipami.
- Moder/rdeč rezultat (b/rResult) sta stolpca, ki z vrednostma ena ali nič definirata zmago oz. poraz ekipe, kjer ena pomeni zmago, nič pa poraz pri ustrezni barvi. Podatek je bistven, saj definira rezultat igre.
- Trajanje igre (game length) definira trajanje igre v minutah. Podatek je vsestransko uporaben, hkrati pa iz njega lahko pridobimo velikost nizov Gold.
- Razlika v zlatu (goldDiff) je vrednost, ki definira razliko v pridobljenem zlatu med modro in rdečo ekipo (po funkciji goldBlue – goldRed). To je prvi izmed časovnih nizov števil. Stolpec namreč vsebuje niz števil, za vsako minuto trajanje igre definira razliko v zlatu.
- (Modre/rdeče) zmage (b/rKills) je kompleksen podatek, sestavljen iz kar devetih različnih elementov. To so čas igre, premaganec, zmagovalec, assistence (od ena do štiri, saj lahko v teoriji asistirajo vsi soigralci) in lokacija (zapisana s koordinatama x in z). V kompleksnejših analizah s podatkovnim rudarjenjem bi tu lahko pridobili veliko podatkov o dinamiki igre in sodelovanju med igralci, a ker se osredotočamo na vizualizacije, bomo namesto tega stolpca uporabili podobne stolpce, ki definirajo posamezne igralce.
- (Moder/rdeč) stolp (b/rTowers) definira, kdaj je ekipa podrla sovražnikov stolp. Zapis je sestavljen iz časa v igri in lokacije stolpa (lokacije stolpov so sicer fiksne). V kompleksnejših analizah lahko s tem podatkom analiziramo igralno taktiko ekipe.
- (Moder/rdeč) inhibitor (b/rInhib) definira, kdaj je ekipa podrla nasprotnikov inhibitor. Stolpca delujeta na enak način kot prejšnji primer, le da, kot sledi iz imena, opredeljujeta napade na inhibitorje.
- (Moder/rdeč) zmaj (b/rDragons) definira, kdaj je ekipa premagala zmaja. Podobno kot v prejšnjih primerih je podatek sestavljen iz vseh zmag nad zmaji, ki so zapisane s časom igre in tipom zmaja (v primeru starejših verzij pred uvedbo tipov zmajev je zapisan kot splošen »zmaj«). Ker zmaga nad zmajem naredi igralce trajno močnejše, je pregled nad zmaji bistven element za analizo igre.

- (Moder/rdeč) baron (b/rBarons) deluje na enak način kot prejšnji stolpec, le da popisuje dogodek v igri, ko je ena izmed skupin premagala barona. Tu je zapisana le časovna značka, saj je samo en tip barona.
- (Moder/rdeč) Rift herald (b/rHeralds) je tretji stolpec, ki opisuje dogodke tega tipa v igri. Kot v primeru barona je tudi tu zapisana le časovna značka, saj je v igri le en tip znanilca.
- (Modre/rdeče) prepovedi (blue/redBans) definira, katere junake sta kapetana ekip že pred igro prepovedala v fazi izbire junakov. Ta stolpec vedno vsebuje niz petih imen junakov. V analizi igre je stolpec še posebej pomemben, saj lahko z njim ocenimo, katere junake bo nasprotna ekipa verjetno prepovedala, če nad njo izvedemo dobro agregacijo. Po drugi strani je lahko varljiv, saj so prepovedi včasih odvisne od nasprotnikovih preferenc.

V zadnjo skupino uvrščamo podatke o posameznih igralcih. Ta skupina ima največ stolpcev, več kot polovico, kar je pravzaprav logično. Igralcev je namreč deset, pet v vsaki skupini, in če želimo za vsakega posebej zapisovati podatke poteka igre, bomo hitro dobili zelo široko tabelo. Da bi skrajšali opis posameznih stolpcev na smiselno velikost, bomo kar tu omenili, da je vsak izmed naštetih stolpcev pravzaprav en par, po en stolpec za vsako (modro in rdečo) ekipo. Drugi element, ki je skupen, je pozicija, na kateri igralec igra – teh je pet (ang. top, middle, support, ADC, jungle). Ime stolpca tako vsebuje barvo ekipe in pozicijo igralca v imenu, prav tako pa naziv opazovane spremenljivke. Stolpec goldBlueTop na primer pomeni, da stolpec zapisuje zlato po minutah za igralca modre ekipe, ki igra na zgornji poziciji (top):

- Igralec definira ime igralca, to pomeni, kateri član ekipe pravzaprav igra na tisti poziciji. To je edina skupina stolpcev, kjer definicija spremenljivke ni zapisana v naslovu stolpca. Ta skupina stolpcev je zelo pomembna za večino analiz, saj je le tu zapisano, kdo pravzaprav igra za neko ekipo.
- Junak (Champ) zapisuje, katerega junaka je izbral igralec na tej poziciji. Podatek bi lahko uporabili za opazovanje dinamike med junaki, lahko pa tudi za analizo pogosto igranih junakov za nekega igralca.
- Zlato (Gold) – podobno kot v stolpcih, ki zlato zapisujejo na ravni ekipe, je tudi ta skupina stolpcev pravzaprav dolg niz števil, saj zapisuje količino zlata, ki ga ima posamezni igralec v vsaki minuti igre. Ker je zlato eden bistvenih elementov igre, je stolpec pomemben predvsem, ko analiziramo potek posamezne igre, na primer, če želimo izvedeti, kdaj se je ena ekipa točkovno oddaljila od druge, in v primeru podatkovnega rudarjenja ter kompleksnejših analiz, kdaj lahko predvidevamo zmago.

5.3 Definicija predstavitvenega problema

V opisu igre smo že povedali, da se igra League of Legends igra tudi na profesionalni ravni in da se ekipe, ki igrajo na najvišje rangiranih tekmovanjih, lahko po prizadevnosti primerjajo z ostalimi profesionalnimi športniki. Zato ni dovolj, da le vadijo, ampak morajo tudi natančno analizirati že odigrane igre; tako svoje, kjer ugotovijo priložnosti za

izboljšavo, kot nasprotnikove, da ugotovijo njihove pomanjkljivosti in kompetence. Za to se najpogosteje uporablja videoanaliza posnetka igre. Ker pa so relativno dolgi, je včasih treba poiskati tudi bližnjice. Zato bomo skušali razviti vizualizacijo, ki bo dobro predstavila potek igre in v njem vse bistvene lastnosti, s čimer lahko ugotovimo kritične trenutke in s tem veliko lažje izvedemo videoanalizo.

Obdelovali bomo torej eno samo vrstico, ki pa bo vsebovala veliko stolpcev, torej bo velika v širino, ne toliko v višino. Razlog za to izbiro je logičen. Večina raziskav velikih baz temelji na višini, kar pomeni, da je to področje bolje raziskano, hkrati pa je, vsaj na najvišjem vidiku (to je z vidika poslovnega uporabnika), relativno nezanimivo. Če namreč že kot uporabniki obdelujemo veliko število vrstic, ki jih je treba agregirati in pregledati, bomo po vsej verjetnosti uporabili komercialen program. Razvoj algoritmov za optimizacijo analize velikih podatkov, kot so jih opisali Chan, Correa in Ma (2014), Pahins et al. (2017) ter ostali, je namreč kompleksen tudi s stališča informatikov, torej je nesmiselno razmišljati, ali jih laik lahko razvije. Po drugi strani take algoritme razvijamo od dna navzgor, se pravi, da začnemo že pri podporni statistiki in jo nato implementiramo v program, ki smo ga sami razvili, česar pa ne moremo razviti, če se omejimo na paket MS Office (programsko opremo bi lahko razvili v okolju MS Visual Basic ali kakšnem drugem razvojnem okolju, ki pa jih metodologija ne predvideva). Dodatno gre pri vseh teh algoritmih za optimizacije, ki pospešijo obdelavo, posebej pri interaktivnih vizualizacijah – torej ne gre toliko za razvoj kot za pospešitev že obstoječih metod na inovativne načine.

Za vizualizacijo veliko podatkovnih tipov hkrati smo v teoretičnem delu opisali bistveni metodi. Prva metoda je tableplot (Tennekes, de Jonge & Daas, 2013), ki temelji na združitvi več vrstic v skupine in nato njihovi razvrstitvi po določenem ključu na osnovi primarnega atributa. Za naš primer ta sistem ni primeren. Kot smo že zapisali, tabelarni grafikon ni najbolj optimalen, če ima lahko določen stolpec veliko različnih vrednosti, saj v tem primeru ne moremo izvesti smiselnega grupiranja. Iz istega razloga je pogoj tabelarnega grafikona, da so vsi podatkovni tipi atomarni. V našem primeru nobeden od dveh pogojev ni izpolnjen, saj imamo vsaj en zvezni tip (trajanje igre, ki bi ga sicer lahko diskretizirali), poleg tega pa imamo veliko neatomarnih atributov, ki jih bomo vsekakor morali razbiti. Atributi so sploh problematični, saj zaradi svoje narave nimajo enakega števila elementov. Dodatna neprimernost tabelarnega grafikona za naš primer je zanašanje na le eno vrstico, kar popolnoma ovrže smisel takšne vizualizacije. Tabelarni grafikon torej za naš primer ni primeren.

Druga osnovna metoda, ki smo jo omenjali kot možnost za vizualizacijo veliko različnih podatkov, je nadzorna plošča, ki se zelo pogosto uporablja v poslovne namene. Tu gre za to, da podatkov ne skušamo stisniti v en sam grafikon, ampak namesto tega za logične skupine atributov izrišemo različne grafikone, ki jih nato logično razvrstimo na delovno ploščo (Rasmussen, Bansal & Chen, 2009). Uporaba nadzorne plošče ima še dodatno prednost, in sicer da nam atributov ni treba prisiliti na neko enotno formo – namesto tega lahko grafikone

strukturno priredimo posamezni skupini podatkov, glede na njene specifične lastnosti. Dokler so posamezni grafikoni smiselno razporejeni in dobro označeni, strukturna enotnost podatkov niti ni pogoj. Zaradi tega je nadzorna plošča dobra metoda za vizualizacijo našega primera, kar pomeni, da jo bomo uporabili. Nadzorna plošča je relativno preprosta za razvoj, kar izpolnjuje pogoj, da jo lahko razvijemo le z uporabo programske opreme paketa MS Office, za razvoj pa potrebujemo le dober načrt, vizualizacijsko znanje in logiko ter obvladovanje podatkovnih orodij v programu Excel. Slabost tega pristopa je, da smo omejeni v dodatni funkcionalnosti, kot je interaktivnost, ampak glede na idejo raziskave moramo kompromis pač sprejeti.

Za vizualizacijo velike podatkovne zbirke je torej najprimernejša zasnova nadzorne plošče, vsaj v primeru, ko se osredotočamo na različne podatke, ki so med seboj lahko odvisni ali pa tudi ne. Pristop prav tako lahko reši vsaj osnovno idejo obravnavanja z omejenimi viri, saj lahko nadzorno ploščo logično razbijemo v več neodvisnih grafikonov. Drugače povedano: za idejno zasnovo nadzorne plošče ne potrebujemo izkušenih izvedencev (ki bodo sicer naredili veliko boljšo, ampak bodo za organizacijo tudi večji strošek), temveč jo lahko sestavijo nestrokovnjaki, če le analizirajo, kaj imajo na voljo in kaj morajo vedeti.

5.4 Zasnova nadzorne plošče

Preden začnemo pripravljati podatke in izdelovati grafikone, moramo zasnovati razporeditev in načrt nadzorne plošče, ki naj bo logično povezana in naj združuje elemente v smiselne skupine.

Nadzorna plošča bo grobo razporejena v štiri četrtine. V levi zgornji četrtini bomo preprosto definirali osnovne podatke o igri, ki so pomembni za analizo, vendar pa jih na ravni ene igre ne moremo smiselno vizualizirati. Zato se bomo osredotočili na čim bolj intuitivno tabelo in jo obarvali z barvami ekipe. Zapisi bodo obarvani modro ali rdeče. Intenzivnost barve bo hkrati definirala zmago ali poraz. Bleda barva označuje poraz. Če je torej igralec z imenom Wulthar (ime je izmišljeno, vse povezave z resničnimi igralci so le naključne) zapisan v tabeli, obarvani bledo rdeče, to pomeni, da je igral v rdeči ekipi, ki je bila v igri poražena. Na podoben način bomo definirali tudi prepovedi in igrane junake (natančen načrt bo opisan v nadaljevanju).

V desni zgornji četrtini bomo definirali količino zlata glede na trajanje igre. Za to bomo uporabili črtni diagram z 12 linijami (po eno za vsakega igralca in dve za skupno zlato ekipe). V primeru nepreglednosti bomo grafikon razdelili na dva neodvisna grafikona, enega za igralce in enega za skupno zlato, s čimer bomo preprečili, da bi bile linije zaradi razlike med vrednostmi, ki označujejo trende posameznih igralcev, preveč stisnjene.

V spodnjem desnem delu (desni del je nasploh namenjen zlatu, saj je bistveno pri gradnji junakov, to pa je bistven del igre) bomo definirali razmerje zlata med igralci z uporabo

tortnega grafikona, ki nam bo pokazal, koliko skupnega zlata je ob koncu igre imel posamezni igralec.

V levem spodnjem področju bomo definirali časovnice. Naredili jih bomo s pomočjo razsevnega grafikona, s čimer bomo prikazali potek igre in trenutke pomembnih dogodkov, denimo zmajev, baronov, stolpov in inhibitorjev, oz. bolje rečeno: trenutke, ko je ena izmed ekip premagala enega izmed teh objektov. V primeru, da gre za zmaje ali barona, bomo z ustrezno barvo označili tudi, za katero ekipo gre.

5.5 Uvoz in priprava podatkov

Uvoz in priprava podatkov je bila prva faza v praktičnem delu. Kot smo omenili v teoretičnem delu, je dobra priprava podatkov za analizo in vizualizacije bistvenega pomena, saj drastično zmanjša možnost za potencialne težave in količino dela, ki ga moramo opraviti kasneje (Harris, 2018). Ko gre za vizualizacijo z orodjem Excel, dobro pripravljene podatke le označimo, orodje pa nam izriše v večini primerov ustrezen grafikon (z neustreznimi podatki niti ne moremo delati).

Podatkovno zbirko smo najprej sneli s spletne strani Kaggle (Kaggle, 2018). Vzeli smo le glavno zbirko, podpornim zbirkam pa smo se izognili, čeprav bi bile za določene vizualizacije boljše/primernejše. A ker take podporne zbirke niso vedno na voljo, smo se odločili, da bomo raje vse podatke pretvorili iz osnovne zbirke.

Zbirka je bila prvotno v formatu .csv, kar označuje »vrednosti, ločene z vejico« (ang. *comma separated values*), kjer so, kot ime pove, strukturirano zapisane vrednosti, ločene z vejico. Večina programske opreme za obdelavo podatkov zna tak tip datoteke pravilno razčleniti, ni pa nujno, da to naredi brez težav. Naše zbirke MS Access ni znal uvoziti pravilno, zato smo uporabili zelo pogost obvod, in sicer smo datoteko .csv najprej pretvorili v Excelovo razpredelnico, s katerimi ima Access veliko manj težav. Ta pristop ima še eno prednost, in sicer da lahko podatkovno zbirko hitro preletimo, da vidimo, če je dejansko tista, ki jo želimo uvoziti. Izbrali smo možnost, da vodilna vrstica definira naslove stolpcev, saj v podatkovni zbirki dejansko jih.

Razlog, da smo se odločili za uvoz v podatkovno bazo Access, je, da se v poslovnem svetu pogosto dela z bazami, saj jih je veliko lažje posodabljati. Datoteke, kot so .csv, omogočajo veliko lažji prenos velikih količin strukturiranih podatkov, zato se uporabljajo za ad hoc obdelavo.

Naslednji korak je bil kreiranje nove Excelove datoteke, saj bo nadzorna plošča sestavljena v tem okolju. Za boljšo urejenost smo kreirali štiri različne strani (ang. *sheet*), kamor bomo vnesli surove podatke. S tem omogočimo, da v nadzorni plošči ni nepotrebne navlake. Na prvi strani bomo hranili podatke o igri, v drugi podatke o dogodkih, ki jih bomo uporabili za časovnico, v tretji pa podatke o zlatu.

Priprava podatkov na prvi strani je bila preprosta. Ker so podatki o igralcih in igri atomarni, jih niti ni bilo treba posebej pripravljati. Zaradi tega smo v podatkovni bazi izbrali ustrezno igro, ki bo uporabljena kot primer, in preprosto prenesli podatke v ustrezna mesta v Excelovi razpredelnici. Ker se ukvarjamo le s posamezno igro, teh podatkov ne moremo neposredno analizirati, vseeno pa služijo kot bistvena informacija.

Za stran zlato smo začeli enako. Vzeli smo 12 vrstic in eno pomožno, v katero smo vpisali minute igre. Nato smo v vsako ustrezno vrstico prenesli niz, ki označuje zlato po posameznem kriteriju, mimogrede pa odstranili oglate oklepaje. Tu nastane težava, saj kopiranje v Excel obravnava celoten niz kot eno celico, mi pa želimo imeti vrednost za vsako minuto posebej. Tu moramo zato procesirati podatke. Excel ima za to ustrezno orodje: temu je namenjena operacija »tekst v vrstice« (ang. *Text to Columns*). V operaciji smo izbrali vejico kot razmejitev (ker se za razmejevanje v podatkih uporablja vejica). Z operacijo smo dolg niz pretvorili v 46 celic (za vsako minuto igre ena). Število celic je v tem primeru dinamično, saj igre trajajo različno dolgo.

Na koncu nam pri pripravi podatkov ostane le še stran z dogodki. Podobno kot pri strani o zlatu tukaj pretvarjamo podatke, ki so v zbirki shranjeni kot niz. Vendar gre za razliko od enakomernega niza pri podatkih o zlatu tu za kompleksnejšo hierarhijo. Podatek o času je prvi, sledi mu podatek, na kateri poti je podrti stolp, in na koncu še, na katerem mestu je (na vsaki poti so namreč trije stolpi). Da bi uredili tabelo, smo nizu najprej odstranili oglate oklepaje, saj jih Excel brez zunanjih vtičnikov ne zna pravilno procesirati (funkcija je sicer trivialna in bi jo vsak izvedenec IT stroke verjetno lahko spisal v integriranem razvojnem okolju Visual Basic, a ker se osredotočamo na osebe, ki na tem področju niso strokovnjaki, tega ne bomo naredili), vejice pa vseeno lepo ločujejo elemente. Nato smo s funkcijo »pase« transponirali podatke iz vodoravne v navpično lego. Sledilo je razvrščanje. V trenutni situaciji so bili podatki, ki bi morali biti zapisani v tristolpčno tabelo, zapisani zaporedno v enem stolpcu. To smo popravili tako, da smo polje druge vrstice zapisali eno višje in en stolpec v levo, pri čemer smo uporabili preprosto funkcijo, na primer »=C2« (v primeru, da kopiramo iz polja B3). Funkcije smo prenesli navzdol za vse vrstice. Podobno smo naredili še za tretji podatek, ki smo ga prav tako prenesli dve polji desno in navzgor. Nato nam je ostalo le še, da izbrišemo neveljavne vrstice. Za konec smo očistili podatke, da bodo v vizualizaciji preglednejši.

Na enak način smo vnesli tudi ostale podatke na strani dogodkov, kar pa je bilo pravzaprav še lažje, saj so vsebovali manj elementov. Ker je bila igra odigrana pred uvedbo tipov zmajev, smo lahko ta podatek preprosto zanemarili. Podatek o baronih prav tako nima tipa. Na koncu je še podatek o inhibitorjih, ki nosi le zapis o poti, ne pa o poziciji na njej (za razliko od stolpov je na posamezni poti le en inhibitor).

Podatek, ki ga nismo uporabili, je sicer še podatek b- in rKills, ki se nanaša na boje med posameznimi igralci. Težava tega niza (podatek je dejansko niz) je v tem, da nima enotne

strukture. V igri se registrirajo tudi asistence (kjer eden izmed igralcev pomaga soigralcu premagati nasprotnika), katerih število je odvisno od tega, koliko igralcev je pomagalo (med enim in štirimi igralci). Excel žal nima funkcije, ki bi znala pametno razčleniti strukturo gnezdenih oglatih oklepajev (to smo prej že omenili), zato si brez veliko ročnega dela ne moremo veliko pomagati.

5.6 Izgradnja nadzorne plošče

Za izgradnjo nadzorne plošče smo se najprej premaknili na novo stran, ki poskrbi, da v področju nadzorne plošče ne bo nepotrebnih podatkov. Izgradnjo smo začeli v levem zgornjem kotu, s tabelami o osnovnih podatkih o igri. Na vrhu smo neposredno vnesli vir (v primeru, da bi uporabljali povezave na bazo, bi tako lahko vnesli igro neposredno). Nekoliko nižje smo vnesli podatke o igri. Podatki niso vneseni neposredno, ampak so preneseni s strani o igralcih. Podatek o sezoni je združen s podatkom o letu z uporabo funkcije »Concatenate«. Isto funkcijo smo uporabili v polju »trajanje igre« (ang. *game length*), kjer smo dodali oznako za minute. Zadnje polje v tabeli definira zmagovalca, zato je ustvarjeno s pogojem: v primeru, da je podatek bResult enak ena (kar pomeni zmago modre ekipe), bomo zapisali vrednost »Blue« (modra ekipa), v primeru, da je vrednost nič, pa »Red« (rdeča ekipa). Podatek rResult je tu redundanten, saj neodločenih iger ni. Za podatke, ki se nanašajo na ekipe, smo uporabili pogojno oblikovanje, zato je polje »zmagovalec« (ang. *winner*) obarvano z barvo zmagovalne ekipe.

Slika 7: Osnovni podatki na nadzorni plošči

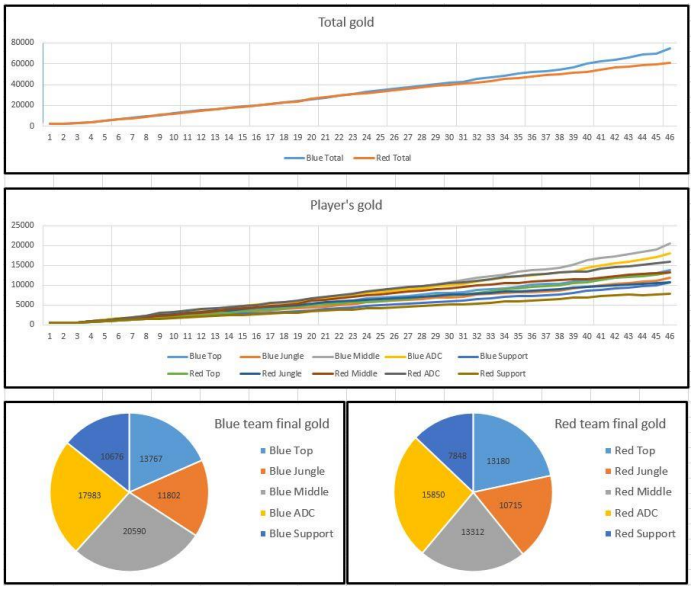
Source	http://matchhistory.na.leagueoflegends.com/en/#match-details/TREU/1000560042?gameHash=a52541e75b0b762a			
Match Info				
Season	Spring 2015		Game length	46 min
Game type	Promotion		Winner	Blue
	Blue team	H2K	Red team	FAC
Top	Odoamne	Lissandra	Xaxus	Jax
Jungle	loulex	JarvanIV	Obvious	Elise
Middle	Febiven	Zed	SozPurefect	Ahri
ADC	Hjarnan	Corki	Sedrion	Graves
Support	Voidle	Thresh	MounTain	Janna
	Blue bans		Red bans	
	Syndra		Xerath	
	Leblanc		Jayce	
	Irelia		LeeSin	

Večji tabeli sta uporabljeni za podatke o prepovedih in igralcih. Tudi te tabele podatke črpajo s strani o igralcih. Tukaj smo prav tako uporabili pogojno oblikovanje, vendar nekoliko drugače. Za dobro preglednost so polja, ki se nanašajo na modro ekipo, vedno obarvana modro, polja rdeče ekipe pa rdeče. Za še boljšo preglednost smo dodali pogoj, kjer se polja zmagovalne ekipe obarvajo v močno, polja poražene ekipe pa bledo barvo. Pogoj smo dodali tudi za tabeli »prepovedi« (ang. bans) obeh skupin (slika 7).

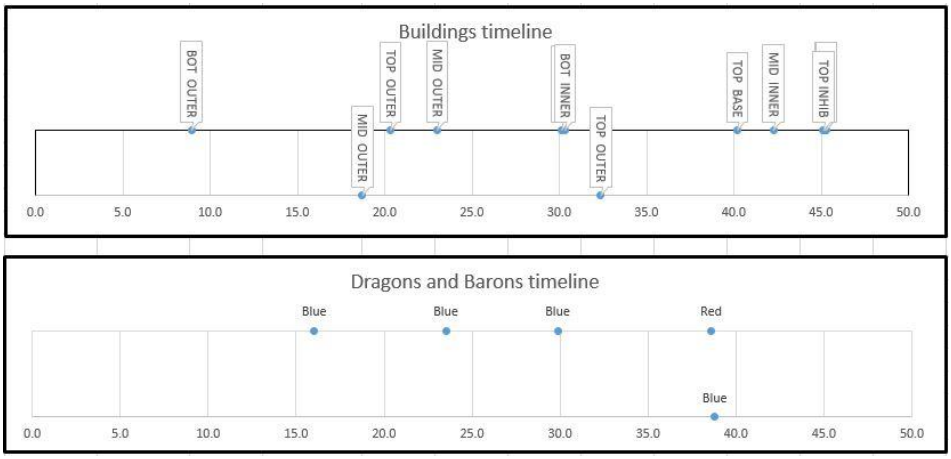
Nadaljevali smo z drugo polovico plošče, kjer smo planirali podatke o zlatu. Tja smo postavili črtna grafikona, ki definirata količino zlata glede na trajanje igre (na vsako minuto).

Najprej smo poskusili z enim, a je bil preveč nepregleden zaradi večje razlike med količino zlata vseh igralcev skupaj in količino zlata posameznih igralcev, zato smo grafikona ločili. Za boljši pregled razmerja smo dodali še dva tortna grafikona, ki prikazujeta količino zlata igralcev ob koncu igre. Vse podatke smo črpali s strani »zlato« (slika 8).

Slika 8: Področje podatkov o zlatu



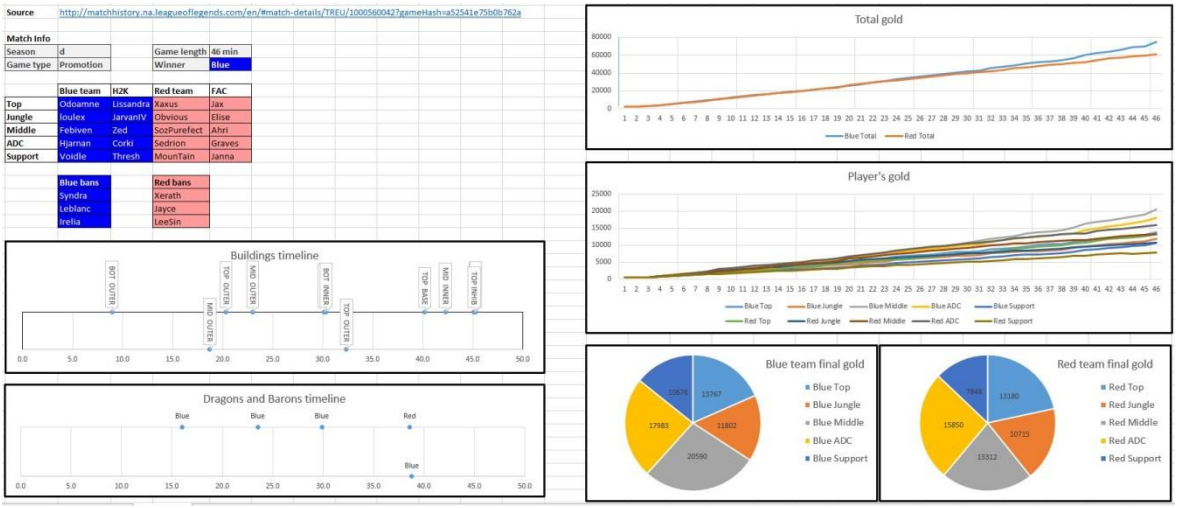
Slika 9: Časovnica



Na podoben način deluje druga časovnica, le da se na osi z nanaša tip dogodka (zmaji in baroni), oznaka točke pa nam pove, katera ekipa ga je premagala. Tu smo morali sprejeti kompromis, saj se časovnici glede na definicije osi z razlikujeta. To smo naredili zaradi boljšega pregleda nad bistvenimi podatki glede na posamezno časovnico (slika 9).

Skupna nadzorna plošča (slika 10) torej omogoča pregled nad večino bistvenih podatkov. Eden izmed podatkov, ki zaradi težavne obdelave ustreznih podatkov ni predstavljen, je interakcija med igralci. Tudi tu bi se sicer dalo narediti nekaj vizualizacij, a bi najprej potrebovali dobro razčlenitev nizov, ki pa je Excel brez vtičnikov ne omogoča.

Slika 10: Končana nadzorna plošča



5.7 Eksperiment s ponovno izgradnjo nadzorne plošče

Do zdaj smo ugotovili, da orodje Excel omogoča izgradnjo preprostejših nadzornih plošč tudi brez vtičnikov in kompleksnih funkcij. V naslednji fazi se moramo vprašati, kako težavno je to izvesti, še posebej za nekoga, ki ni najbolj vešč Excelovih orodij za obdelavo podatkov in grafikonov. Včasih je potrebno tudi nekaj improvizacije, kjer pa moramo prav tako vedeti, kako nekatere funkcije delujejo (tu velja omeniti čiščenje nepotrebnega balasta z uporabo orodja »poišči in zamenjaj« (ang. *Find and Replace*), kjer nepotrebne znake zamenjamo z vrednostjo null, se pravi s praznim poljem).

Da bi to preverili, smo zasnovali eksperiment, kjer smo sodelavca prosili, naj izvede nekaj nalog, ki temeljijo na bistvenih korakih pri gradnji nadzorne plošče. Lahko bi ga preprosto prosili, naj zgradi nadzorno ploščo od začetka, ampak v tem primeru bi morali bolj podrobno opazovati, poleg tega pa bi bili podatki manj natančni. Hkrati bi vse skupaj po nepotrebnem trajalo dlje.

V magistrskem delu obravnavamo načeloma dve povezani, a ne popolnoma odvisni vprašanji. Prvo, ali je mogoče veliko zbirko vizualizirati, smo že definirali v poglavju o zasnovi nadzorne plošče, še več pa smo opisali v povzetku literature v poglavju Vizualizacija velikih podatkov. V tem poglavju moramo analizirati, kako se uporabnik znajde pri vizualizaciji.

V nalogah smo preverjali bistvene operacije, ki smo jih uporabili pri pripravi podatkov in izdelavi nadzorne plošče. Nismo sicer preverjali uvoza datoteke .csv, saj smo predpostavili, da bo v dejanskih primerih organizacija (gospodarska družba, druga ustanova) že uporabljala podatkovno bazo (v takšni ali drugačni obliki). Prav tako nismo preizkušali idejne zasnove nadzorne plošče, saj predpostavljamo, da so vodilni v administraciji podjetja zasnovali idejo, kako naj je stvar videti. Tako preverjamo le izgradnjo.

Eksperiment je bil razdeljen na sedem povezanih nalog, ki jih je sodelavec opravljal drugo za drugo. Med opravljanjem smo ves čas sledili sodelavcu, opazovali, kako uspešen je, analizirali težave in v njihovem primeru pomagali z namigi. Na velike podatke se nanaša predvsem prvi del, kjer govorimo o pripravi podatkov. Drugi del analizira bolj splošno sposobnost povprečnega uporabnika za pripravo vizualizacij in poznavanje orodij, ki jih pri tem uporabljamo. Primarno vprašanje, ki analizira vizualizacijo velikih podatkov, je sicer pri dobro pripravljenih podatkih irelevantno, saj, kot smo ugotovili v prejšnjem poglavju, že preprosta orodja avtomatsko ustvarijo kakovostne vizualizacije, če jih le znamo uporabiti. V ta namen bomo torej preizkusili poznavanje uporabe teh orodij.

V tem primeru je sodelavka, univerzitetna diplomirana pravnica, ki uporablja računalnik kot pripomoček od leta 1986, posebnega izobraževanja, razen nekaj tečajev (za Word, Word Perfect in še starejše, ki so bili v rabi konec osemdesetih let), pa nima. Kot večina drugih, ki so začeli uporabljati računalnik v tistem času, je v glavnem samouk. Na svojem službenem računalniku ima vse programe Office v slovenščini, vendar ji zaradi dobrega znanja angleščine program ne dela težav. Obvlada branje grafikonov (vse vrste poslovnih podatkov in informacij), zaradi svoje vsebine dela pa jih še ni kreirala, ker to naredijo na podlagi njenih podatkov ustrezne strokovne službe pri njenem delodajalcu.

V prvi nalogi smo sodelavki naročili, naj iz podatkovne baze prenese podatke v Excelovo razpredelnico. Kot že v izdelavi naše glavne nadzorne plošče se nismo ukvarjali s povezavami neposredno na bazo, ampak smo podatke preprosto prenesli ročno. Ta del sodelavki ni povzročal večjih težav, nekoliko se je zataknilo le na začetku, saj ni poznala obravnavane problematike in je poznala le slabo strukturo baze, kar je stvar nekoliko upočasnilo. Za tem se je hitro privadila, in prenašanje ostalih podatkov v Excel je teklo brez nadaljnjih težav.

V naslednji nalogi smo se osredotočili na pripravo podatkov. Ker so bili v podatkovni zbirki stolpci, ki so dejansko vsebovali niz podatkov, smo preverjali, kako bi znala sodelavka pretvoriti podatke iz niza v tabelo. Tu se je zataknilo, saj sodelavka ni imela izkušenj s tako obdelavo, zato smo pomagali z nasveti. Namignili smo, da je ustrezna operacija v zavihku »Data«. Sodelavka je našla operacijo »Text to columns« in ustrezno nastavila nastavitve v čarovniku, s čimer so se podatki iz niza pretvorili v vrstico.

V naslednji nalogi smo obravnavali čiščenje podatkov. Sodelavka je najprej skušala poiskati funkcijo, kar je sicer tudi možnost, ampak v našem primeru nekoliko preveč kompleksna. Ker je ni našla, smo pomagali z namigom, da lahko nepotrebne znake zamenjamo z ničimer, kar jih v praksi odstrani. Ker sodelavka obvlada Word, je hitro prišla do uspešne uporabe operacije »poišči in zamenjaj« in uspešno odstranila nepotrebne priveske v skladu z navodili.

Največ težav ji je povzročila uporaba filtra, saj tudi s tem doslej ni imela izkušenj. Tu je bila potrebna pomoč ves čas, saj sodelavka niti ni poznala operacije niti se ni znašla v seznamih možnosti za uporabo filtrov.

Nato smo se lotili izgradnje nadzorne plošče. Prva naloga je bila uporaba funkcij »Concatenate« in »If«. Sodelavka ni poznala nobene od njih, zato smo ji jih namignili. Pri uporabi funkcije Concatenate je hitro razumela delovanje z uporabo vgrajenega namiga, ki ga ponuja Excel. Več težav je nastalo pri uporabi funkcije If, saj sodelavka tudi pri tem nima nobenih izkušenj v programiranju, zato ni najbolje razumela ideje logičnih preizkusov.

Podobno težavo je imela pri uporabi pogojnega oblikovanja, saj tudi to uporablja logične funkcije, s katerimi nato ustrezno oblikuje celice ali besedilo. Težave so se pojavile pri postavitvi logičnih poskusov, nekaj zmedenosti pa je bilo pri obravnavanju podatkov (rešitev, ki jo je sodelavka predlagala, je bila sicer smiselna, ampak nekoliko redundantna, kar bi v redkih primerih, če podatki niso pravilno zapisani, lahko pripeljalo do zmede).

Na koncu smo preverili še izdelavo grafikonov. Sodelavki smo namignili, kako naj označi obravnavane podatke, nato pa je večino opravila brez težav. Nekaj manjših zapletov pri navigaciji po nastavitvah je sicer bilo, ampak niso zahtevali dodatnih pojasnil.

Glede na opazovano bi lahko zaključili, da povprečen uporabnik ni ravno najbolj vešč naprednejših funkcij, po drugi strani pa, glede na namige, ki smo jih morali dati sodelavki, in njena vprašanja v zvezi s tem, ni naletela na nobeno težavo, ki je ne bi mogli rešiti s hitro poizvedbo generičnega vprašanja na kakšnem popularnem spletnem iskalniku.

5.8 Ocena nadzorne plošče

Kot so napisali Rasmussen, Bansal in Chen (2009), je kakovost nadzorne plošče odvisna od količine in kakovosti informacije, ki jo plošča preda uporabniku. Zato smo se odločili, da bomo za mnenje prosili izkušenega igralca igre League of Legends (sicer ne na tekmovalni ravni), naj pregleda nadzorno ploščo in oceni njeno kakovost.

Ocena nadzorne plošče je bila pozitivna. Ocenjevalec je bil navdušen predvsem nad grafikonom za zlato, s katerim lahko po njegovem mnenju dobimo zavidljivo količino informacij. V študijskem primeru se sicer ne vidi tako dobro (saj gre le za profesionalne igralce), ampak če bi v začetni fazi eden izmed igralcev dobil manj zlata od pričakovanega, bi lahko hitro ugotovili, da nekaj ni prav. Prav tako bi lahko ugotovili, kaj se zgodi, če eden izmed igralcev v skupinski fazi (se pravi, ko je skupina v enem kosu) dobiva preveč zlata – to pomeni, da verjetno ne igra kot del skupine, ampak samostojno »melje« vojščake. Z grafikonom za spremljanje zlata po igralcih in znanjem o igri lahko torej hitro identificiramo anomalije.

Na podoben način lahko igro analiziramo s časovnicami. Tam lahko vidimo, da je skupina zaostajala za povprečno igro, in identificiramo, zakaj na primer ekipa ni podrla stolpa. Ker vemo, kdaj pride prvi zmaj, prav tako lahko ugotovimo, da je ekipa zamujala na boj z zmajem. V novih pravilih lahko tudi vidimo, kdaj je ena izmed ekip premagala znanilca (ang. Rift herald), ki je prav tako pomemben za dober pritisk na nasprotno ekipo. S časovnicami tako identificiramo večje ofenzivne premike ali pa ugotovimo, da jih ni bilo tam, kjer bi jih pričakovali.

V osnovi je ocenjevalec ocenil nadzorno ploščo kot zelo solidno analitično orodje za pregled poteka igre in prosil, če lahko sistem vzame za dejanske analize, saj bi mu tako orodje prišlo zelo prav. V tej fazi smo ocenjevalca vprašali tudi za potencialne izboljšave.

Ena izmed predlaganih izboljšav je agregacija vseh iger ene ekipe v prvem kvadrantu, ki ga lahko razširimo v še eno ploščo. V njej bi lahko analizirali svoje ali nasprotne igralce, tako da vidimo, kakšne so njihove preference do junakov in pozicij. Na podlagi tega lahko naši igralci analizirajo tistega, ki jim bo verjetno nasproti. Prav tako lahko za taktične prepovedi analiziramo, kateri junaki so priljubljeni pri nasprotnikih (nekaj prepovedi gre vedno najmočnejšim junakom, za katere v slengu pravimo, da so »pokvarjeni«, ang. *broken*). Prav tako lahko predvidimo, katere junake po navadi potencialni nasprotnik prepove, in se ustrezno pripravimo.

Naslednja predlagana dopolnitev je časovnica spopadov med posameznimi igralci in njihov neoperativni čas. Če je igralec namreč premagan, traja nekaj časa, preden lahko spet vstopi v igro, tako da ima nasprotna ekipa takrat prednost. Najnevarnejše je, če ena ekipa premaga vse nasprotnikove igralce, saj v končnici to pomeni, da imajo med 10 in 30 sekund neovirane ofenzive. V časovnici bi torej definirali, kdaj so bili igralci premagani in obdobje do njihovega ponovnega vstopa v igro. Prav tako zmaga nad nasprotnikom prinese večji kos zlata, kar da igralcu še dodatno prednost. Tako bi lahko naredili časovnico in nanjo nanesli zmage ter poraze, pri porazih pa še čas neaktivnosti.

Tretji predlog se je nanašal na statičnost vstopnih podatkov (kompromis, ki smo ga sprejeli, ker smo ocenili, da je izdelava konektorjev na zunanji vir preveč kompleksna za povprečnega uporabnika). Ocenjevalec je predlagal, da bi na strani s podatki vnašali podatke avtomatizirano s konektorjem na bazo ali pa kar specifičnim aplikacijskim orodjem.

SKLEP

Sodoben poslovni svet temelji na sposobnosti upravljanja in razumevanja podatkov, saj se le tako lahko prilagaja nenehnim spremembam, s katerimi se srečuje. Zaradi tega uporablja stare in nove metode prilagajanja predstavitve podatkov. Če želijo biti podjetja na trgu konkurenčna, morajo obvladati tako zajem in shranjevanje podatkov kot statistično obdelavo in na koncu vizualizacijo. Ti elementi so znanstveno združeni v vedo, ki ji pravimo podatkovna znanost.

Podatkovna znanost je kompleksna in široka veda, zato je predvsem manjša in srednja podjetja (in podobne organizacije) ne morejo izvajati sama. Zaradi tega je na trgu večje število orodij in organizacij, ki podatkovno znanost naredijo izvedljivo za tiste, ki si ustreznih strokovnjakov ne morejo privoščiti. Toda ta orodja so draga, zato skuša marsikatero podjetje tako ali drugače improvizirati.

Ker nobeno podjetje nima neomejenih finančnih ali kadrovskih virov, morajo sklepati kompromise. Obdelava podatkov zato pade predvsem na kupljena orodja ali zunanje sodelavce, ni pa nujno, da vsa. Za primer smo izbrali vizualizacijo, saj je zadnja v verigi obdelave podatkov, poleg tega pa kot člen med podatki in človekom (ki naj iz teh podatkov pridobi informacijo) tudi tista, ki zahteva največ prilagajanja. Ker tako prilagajanje veliko stane, smo sklenili, da je vizualizacija najboljši element podatkovne znanosti za raziskavo.

Za odgovor na vprašanje, ali je vizualizacija podatkov z omejenimi viri sploh mogoča, smo morali najprej definirati omejene vire. Razdelili smo jih v orodja in znanja. Kot orodja smo za omejene vire opredelili vire, ki jih večina administracij redno uporablja pri delu in niso specifično namenjeni vizualizacijam, čeprav te možnosti imajo. V raziskavi smo uporabili MS Excel in Access. V primeru znanja smo vzeli primer uslužbenca, ki se je z orodji že srečal, nima pa izkušenj z njihovimi bolj strokovnimi funkcijami.

V raziskavi smo ugotovili, da orodja, ki jih lahko pričakujemo v vsakem povprečnem malem podjetju, zadostujejo za izdelavo preprostejših vizualizacij, ki jih z istimi orodji lahko združimo v nadzorno ploščo. Ni pa nujno, da imajo povprečni uporabniki tudi znanja, da to dejansko izvedejo.

Glede na raziskavo lahko torej povzamemo, da nestrokovnjaki lahko naredijo preproste vizualizacije, še posebej, če so podatki dobro pripravljeni. Pri pripravi podatkov lahko laiki naletijo na težave, za katere pa bi težko rekli, da so nepremostljive. Glede na namige, ki smo jih med preizkusom dali sodelavcu, ko je prišlo do težav, smo ugotovili, da nobena težava ni bila tako kompleksna, da je uporabnik ne bi mogel rešiti s kratkim pregledom programske dokumentacije ali interneta. Če podatki niso dobro pripravljeni, so orodja nekoliko okorna, v primeru dobrih (predvsem atomarnih) podatkov pa zadovoljijo večino parametrov. Tudi vizualizacije, ki jih naredimo s pomočjo teh orodjih, ne odstopajo od bolj specifičnih orodij.

Glede na namen bi lahko izvedli tudi bolj kompleksne raziskave. Namesto uporabe enega sodelavca bi eksperiment lahko opravili na večjem vzorcu, kjer bi bolj zanesljivo ugotovili, ali povprečni uporabniki znajo izvesti takšne vizualizacije. Raziskava z le enim sodelujočim namreč prikaže določene težave, s katerimi se lahko srečamo, ni pa nujno, da odraža povprečje. Taka raziskava bi imela tudi uporabno vrednost za prihodnji razvoj kadrovanja podjetij glede na vedno večje zahteve obdelave podatkov, saj primanjkuje kakovostnih raziskav v tej smeri.

Lahko bi šli tudi v kompleksnejšo raziskavo, kaj pravzaprav uporabljena orodja zmorejo. Ena izmed omejitev je bila ravno v osredotočanju na povprečnega uporabnika, saj nam to ni dopustilo uporabe kompleksnejših funkcij.

V splošnem lahko sklenemo, da smo dosegli cilj, saj nam je uspelo dokazati, da je vizualizacije, čeprav preproste, mogoče izdelati z omejenimi viri, tudi če so podatkovne zbirke velike. V ta namen bi lahko predlagali vsem, ki se znajdejo v tej situaciji, da vlagajo znanja v poznavanje orodij, s katerimi delajo, saj se lahko z njimi marsikaj doseže. Prav tako velja znanja prenesti na vse zaposlene, vsaj na osnovni ravni.

LITERATURA IN VIRI

1. Askham, N., Denise, C., Doyle, M., Fereday, H., Gibson, M., Landbeck, U., Lee, R., Maynard, C., Palmer, G. & Schwarzenbach, J. (2013). *The six primary dimensions for data quality assessment*. London: DAMA UK.
2. Battle, L., Stonebraker, M. & Chang, R. (2013). Dynamic reduction of query result sets for interactive visualizaton. *Big Data, 2013 IEEE International Conference on* (str. 1–8). IEEE.
3. Bespalaya, E. (2014, julij 25). *How to create waterfall chart in Excel (bridge chart)*. Najdeno 27. junija 2018, na spletnem naslovu https://www.ablebits.com/office-addins-blog/2014/07/25/waterfall-chart-in-excel/
4. Chan, Y.-H., Correa, C. D. & Ma, K.-L. (2014). Regression cube: a technique for multidimensional visual exploration and interactive pattern finding. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *4*(1), 7.
5. Chang, R., Yang, F. & Procopio, M. (2016). From Vision Science to Data Science: Applying Perception to Problems in Big Data. *Electronic Imaging*, *2016*(16), 1–7.
6. Columbus, L. (2018, januar 29). Data Scientist Is the Best Job In America According Glassdoor's 2018 Rankings. *Forbes.* Najdeno 24. junija 2018, na spletnem naslovu https://www.forbes.com/sites/louiscolumbus/2018/01/29/data-scientist-is-the-best-job-in-america-according-glassdoors-2018-rankings/#f6f4f4755357
7. Cottam, J. A., Lumsdaine, A. & Wang, P. (2013). Abstract rendering: Out-of-core rendering for information visualization. V *IS&T/SPIE Electronic Imaging* (str. 90170K-90170K – 13). International Society for Optics and Photonics.
8. Emptylord. (2015, maj 28). *V5.10 patch notes*. Najdeno 5. junija 2018, na spletnem naslovu http://leagueoflegends.wikia.com/wiki/V5.10
9. Friendly, M. (2008). A brief history of data visualization. V *Handbook of data visualization* (str. 15–56). Berlin: Springer.
10. Godfrey, P., Gryz, J. & Lasek, P. (2016). Interactive visualization of large data sets. *IEEE Transactions on Knowledge and Data Engineering*, *28*(8), 2142–2157.
11. Goguelin, S., Flynn, J. M., Essink, W. P. & Dhokia, V. (2017). A Data Visualization Dashboard for Exploring the Additive Manufacturing Solution Space. *Procedia CIRP*, *60*, 193–198.
12. Gosper, H. (2018, maj 30). *Bubble chart - Documentation - Phocas Documentation.* Najdeno 26. junija 2018, na spletnem naslovu https://help.phocassoftware.com/display/userdoc/Bubble+chart
13. Harkins, S. (2012, april 11). 10 cool ways to use Excel's conditional formatting feature. *Techrepublic.* Najdeno 24. junija 2018, na spletnem naslovu https://www.techrepublic.com/blog/10-things/10-cool-ways-to-use-excels-conditional-formatting-feature/

14. Harris, J. (2018, januar 24). *The five D's of data preparation*. Najdeno 26. marca 2018, na spletnem naslovu https://www.sas.com/en_us/insights/articles/data-management/the-five-d-s-of-data-preparation.html
15. Kaggle. (2018, februar). *League of Legends*. Najdeno 7. junija 2018, na spletnem naslovu https://www.kaggle.com/chuckephron/leagueoflegends
16. Kh, R. (2016, januar 18). *The Advantages of Data Visualization for Business*. Najdeno 24. junija 2018, na spletnem naslovu https://www.smartdatacollective.com/advantages-data-visualization-business/
17. Kirk, A. (2016). Data Visualisation: A Handbook for Data Driven Design. London: SAGE.
18. Larsen, A. (2017, november 14). *Why Is Data Important for Your Business?* Najdeno 17. maja 2018, na spletnem naslovu https://blog.grow.com/data-important-business/
19. Lo, F. (b. d.). *What is Data Science?* Najdeno 8. februarja 2017, na spletnem naslovu https://datajobs.com/what-is-data-science
20. Marr, B. (2017, julij 20). The 7 Best Data Visualization Tools In 2017. *Forbes*. Najdeno 22. junija 2018, na spletnem naslovu https://www.forbes.com/sites/bernardmarr/2017/07/20/the-7-best-data-visualization-tools-in-2017/#2337e2946c30
21. Marr, B. (b. d.). *The 5 Biggest Data Visualization Mistakes Everyone Can Easily Avoid*. Najdeno 9. maja 2018, na spletnem naslovu https://blog.qlik.com/the-5-biggest-data-visualization-mistakes-everyone-can-easily-avoid
22. McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. Sebastopol: O'Reilly Media, Inc.
23. Milton, M. (2009). Head First Data Analysis: *A Learner's Guide to Big Numbers, Statistics, and Good Decisions*. Sebastopol: O'Reilly Media, Inc.
24. Monnappa, A. (2016a, januar 29). *Why Data Science Matters And How It Powers Business Value?* Najdeno 15. januarja 2018, na spletnem naslovu https://www.simplilearn.com/why-and-how-data-science-matters-to-business-article
25. Monnappa, A. (2016b, april 5). *Data Science vs. Big Data vs. Data Analytics.* Najdeno 20. avgusta 2017, na spletnem naslovu https://www.simplilearn.com/data-science-vs-big-data-vs-data-analytics-article
26. Morgan, L. (2016, februar 22). *Outsourcing Data Science: What You Need To Know.* Najdeno 24. junija 2018, na spletnem naslovu https://www.informationweek.com/big-data/big-data-analytics/outsourcing-data-science-what-you-need-to-know/d/d-id/1324291
27. Oetting, J. (2018, april 29). *Data Visualization 101: How to Choose the Right Chart or Graph for Your Data.* Najdeno 28. aprila 2018, na spletnem naslovu https://blog.hubspot.com/marketing/types-of-graphs-for-data-visualization
28. Pahins, C. A. L., Stephens, S. A., Scheidegger, C. & Comba, J. L. D. (2017). Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data.

IEEE Transactions on Visualization and Computer Graphics, *23*(1), 671–680. https://doi.org/10.1109/TVCG.2016.2598624.

29. Paruchuri, V. (2016, september 13). 18 places to find data sets for data science projects. *Dataquest*. Najdeno 9. marca 2018, na spletnem naslovu http://www.dataquest.io/blog/free-datasets-for-projects/
30. Perez, R. (2012, april 10). *Four Easy Visualization Mistakes to Avoid | Visually Blog*. Najdeno 9. maja 2018, na spletnem naslovu https://visual.ly/blog/data-visualization-mistakes-to-avoid/
31. Provost, F. & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. Sebastopol: O'Reilly Media, Inc.
32. Rapid Insight. (2015, september 2). 7 Data Cleanup Terms Explained Visually - Rapid Insight Inc. *Rapid Insight.* Najdeno 28. marca 2018, na spletnem naslovu http://www.rapidinsightinc.com/7-data-cleanup-terms-explained-visually/
33. Rasmussen, N. H., Bansal, M. & Chen, C. Y. (2009). *Business Dashboards: A Visual Catalog for Design and Deployment*. Hoboken: John Wiley & Sons.
34. Rogelj, R. & Marinšek, D. (2014). *Statistična analiza : zbirka rešenih primerov s komentarji*. Ljubljana: Ekonomska fakulteta.
35. Schutt, R. & O'Neil, C. (2013). *Doing Data Science: Straight Talk from the Frontline*. Sebastopol: O'Reilly Media, Inc.
36. Selot, M. (2012, september 12). *Keeping it Clean: The Five Step Data Cleansing Process.* Najdeno 28. marca 2018., na spletnem naslovu http://www.salesify.com/keeping-it-clean-the-five-step-data-cleansing-process/
37. Shah, K. (2016, januar). *7 Common Data Science Mistakes and How to Avoid Them.* Najdeno 9. maja 2018., na spletnem naslovu https://www.kdnuggets.com/2016/01/7-common-data-science-mistakes.html
38. Sherice, J. (b. d.). *What is Data Quality and How Do You Measure It for Best Results?* Najdeno 9. marca 2018., na spletnem naslovu https://blog.kissmetrics.com/data-quality/
39. Techlabs, M. (2017, julij 31). *8 Ways you can grow your Business using Data Science.* Najdeno 24. junija 2018, na spletnem naslovu https://medium.com/the-mission/8-ways-you-can-grow-your-business-using-data-science-2bfbc7d893f3
40. Tennekes, M., de Jonge, E. & Daas, P. J. H. (2013). Visualizing and Inspecting Large Datasets with Tableplots. *Journal of Data Science*, *11*.
41. Thearling, K. (2008, julij 17). *An Introduction to Data Mining.* Najdeno 11. aprila 2018, na spletnem naslovu http://www.thearling.com/text/dmwhite/dmwhite.htm
42. van Cauwenberge, L. (2015, september 3). *22 easy-to-fix worst mistakes for data scientists.* Najdeno 9. maja 2018, na spletnem naslovu https://www.datasciencecentral.com/profiles/blogs/10-worst-mistakes-for-data-scientists

43. Violino, B. (2017, november 27). *12 myths of data analytics debunked.* Najdeno 1. aprila 2018, na spletnem naslovu https://www.cio.com/article/3238088/analytics/data-analytics-myths-debunked.html
44. Wang, Z., Ferreira, N., Wei, Y., Bhaskar, A. S. & Scheidegger, C. (2017). Gaussian Cubes: Real-Time Modeling for Visual Exploration of Large Multidimensional Datasets. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 681–690. https://doi.org/10.1109/TVCG.2016.2598694.
45. Wickham, H. (2013). *Bin-summarise-smooth: a framework for visualising large data*. had.co.nz.