Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Samo Remec

Merjenje inflacije v Sloveniji s ceniki spletnih trgovin

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Damjan Vavpotič

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge obravnavajte problem merjenja inflacije na podlagi spremljanja cenikov v slovenskih spletnih trgovinah. Pri tem se zgledujte po projektu „The Billion Prices Project“, ki poteka v okviru MIT in Harvard ter ugotovite v kolikšni meri je mogoče njihov pristop prenesti tudi v Slovenijo. Delo naj najprej predstavi problem izračunavanja inflacije na splošno, nato pa naj se posveti specifikam izračunavanja na podlagi cenikov v spletnih trgovinah. Pridobljene ugotovitve naj bodo osnova za izdelavo ustrezne informacijske rešitve, ki omogočala spremljanje gibanja indeksov cen različnih kategorij artiklov v spletnih trgovinah. Indekse cen izračunane z vašo rešitvijo primerjajte z uradno izračunanimi indeksi cen, ki jih pripravlja Statistični urad RS. Kritično ovrednotite rezultate dela in jih, kjer je to smiselno, primerjajte z rezultati projekta „The Billion Prices Project“.

Zahvaljujem se profesorju in mentorju Damjanu Vavpotiču, staršem, prijateljem in vsem, ki so mi pomagali pri pisanju diplomske naloge.

Svojim prijateljem.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
CPI	Consumer Price Index	Indeks cen življenjskih potrebščin
ICŽP	-	Indeks cen življenjskih potrebščin
NAS	Network-attached storage	Trdi disk na omrežju
ECOICOP	European Classification of Individual Consumption according to Purpose	Evropska klasifikacija individualne potrošnje glede na namen
SURS	Statistical Office of the Republic of Slovenia	Statistični urad Republike Slovenije
HTTP	HyperText Transfer Protocol	-
HTML	Hypertext Markup Language	-
AJAX	Asynchronous JavaScript	-
JSON	JavaScript Object Notation	-
BLS	Bureau of Labor Statistics	Urad za statistiko dela
JEP	Journal of Economic Perspectives	Revija ekonomskih perspektiv
NBER	National Bureau of Economic Research	Nacionalni urad za ekonomska raziskovanja

Povzetek

Naslov: Merjenje inflacije v Sloveniji s ceniki spletnih trgovin

Avtor: Samo Remec

V tej diplomski nalogi smo merili inflacijo preko cenikov spletnih trgovin. Najprej razložimo, kaj je inflacija in kako se jo meri v praksi. Nato povzamemo delo trenutnih projektov, ki jo merijo na omenjen način. V drugem delu razložimo naš pristop k zbiranju podatkov. Opišemo vse tehnologije, ki smo jih uporabil in na težave, na katere smo naleteli. V zadnjem delu primerjamo pridobljene podatke z uradnimi merami inflacije. Približno polovica kategorij se dobro ujema z uradnim indeksom, druga polovica pa ne. Skupni indeks inflacije se v opazovanem obdobju lepo ujema. Ugotovili smo tudi, da trgovci na drobno spremenijo med 0,8 % in 1,1 % cen vsak dan. Implicitno trajanje cene je tako med 91 in 125 dni, kar je bistveno daljše, kot rezultati obstoječih raziskav iz Evrope, ZDA in držav Južne Amerike. Povprečna sprememba cene v istih trgovinah je bila velika (-21,2 % in 26,5 %) in približno enaka, kot v ZDA.

Ključne besede: inflacija, ekonomija, statistika cen.

Abstract

Title: Measuring inflation in Slovenia using online prices

Author: Samo Remec

In this thesis we measure price inflation with prices from online retailers. First, we explain what inflation is and how it is measured in practice. Then we summarise the work of current projects which use this technique. In the second part we explain our approach to collecting data. We describe all the technology used and all the problems we encountered. In the last part we compare our results with official inflation measures. About half of the individual categories match nicely with the official price index, the other half not so much. In the observed period the total price index seems to match nicely. We also found that retailers change between 0.8 % and 1.1 % prices every day. Implicit price spell duration is therefore between 91 and 125 days, which is considerably longer than results from existing studies in Europe, USA and South America. Average price change from the same retailers was large (-21.2 % and 26.5 %) and about the same as in the USA.

Keywords: inflation, economics, price statistics.

Poglavje 1

Uvod

V tržnem gospodarstvu se cene dobrin stalno spreminjajo. Nekatere grejo gor, druge dol. Ko se splošen nivo cen zviša, temu pravimo inflacija. Evropska centralna banka skrbi, da je inflacija v Evroobmočju nizka in stabilna[7]. Merjenje statistike cen je ključnega pomena za uspešno izvajanje monetarne politike.

Tradicionalni pristopi uporabljajo reprezentativno košarico dobrin, za katero se vsak mesec zbirajo cene. Večino se jih zbere na terenu, nekaj pa centralno, preko telefona in interneta. Napredek informacijske tehnologije v zadnjem desetletju pa je omogočil razvoj novega pristopa z merjenjem preko cenikov, objavljenih na spletu.

MIT in Harvard v sklopu projekta The Billion Prices Project[5, 18] že od leta 2007 spremljata inflacijo v več kot 20 državah. Njihov program vsak dan prebere cenike v več tisočih spletnih trgovinah. S pomočjo uradnih podatkov o strukturi potrošnje tako dnevno objavijo svoj indeks cen za te države. Bili so prvi, ki so se s tem ukvarjali, imajo objavljenih vrsto člankov v uglednih ekonomskih publikacijah (JEP, AER, QJE, NBER in drugi).

Za razliko od tradicionalnih metod zbiranja podatkov je ta način mnogo cenejši in hitrejši. Dodatna prednost je, da lahko izhaja dnevno, skoraj v realnem času, medtem ko rabimo za uradnega čakati en mesec.

Iz zbranih podatkov lahko delamo tudi bolj podrobno statistiko cen, ker

imamo zbranih mnogo več podatkov. Obstaja več raziskav[1], ki so to naredile, vendar nobena v Sloveniji. Izračun prvega spletnega indeksa inflacije v Sloveniji je tema te diplomske naloge.

Poglavje 2

Obstoječi pristopi za merjenje inflacije

2.1 Kaj je inflacija?

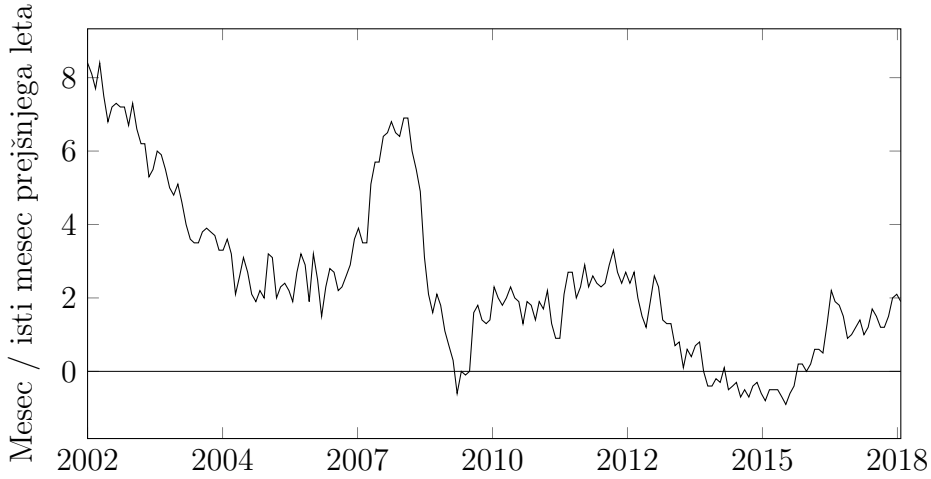
Inflacija je definirana kot splošna rast cen življenjskih potrebščin v državi. V obdobju inflacije sčasoma vse stvari postanejo dražje. To pomeni, da 1 € skozi čas kupi vedno manj stvari. Inflacija je razlog, da je bila hrana 10 let nazaj 30 % cenejša[1].

Inflacija je najpogosteje izražena kot odstotna podražitev splošnega nivoja cen na letni ravni. Drug način prikaza je cenovni indeks (angl. price index), ki prikazuje kupno moč valute glede na določeno leto. Grafa 2.1 in 2.2 prikazujeta razliko.

Meriti želimo, koliko se cene v agregatu višajo ali nižajo (angl. cost of goods and services price index). Eksplicitno ne merimo stroškov življenja (angl. cost-of-living index) – tudi taki indeksi sicer obstajajo, ampak so fundamentalno drugače zgrajeni in ne merijo vrednosti valute.

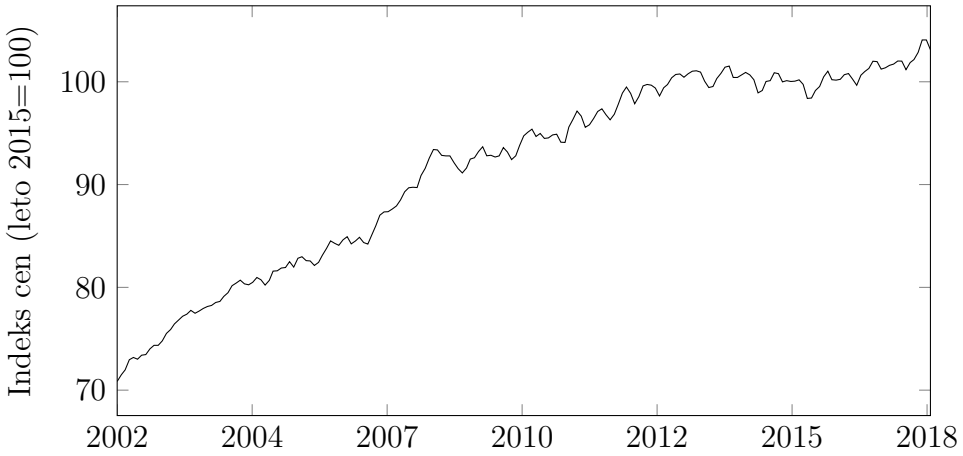
[1]Podatki iz SURS

Grafa prikazujeta gibanje cen v Sloveniji. Leta 2007 smo prevzeli Evro. Leta 2009, 2012 in 2013 smo preživeli v recesiji[2].



Slika 2.1: Inflacija v Sloveniji (mesec glede na isti mesec prejšnje leto).

Če želimo ugotoviti, koliko so se v določenem obdobju spremenile cene, delimo končni in začetni indeks. Primer: julija 2018 je bil indeks cen v Sloveniji 103,12, julija 2010 pa 94,68. Cene so se v tem času torej zvišale za 8,9 %.



Slika 2.2: Indeks cen v Sloveniji.

[2]Recesija pomeni dve zaporedni četrtletji negativne rasti BDP.

2.2 Različni indeksi inflacije

Poznamo več indeksov, ki inflacijo merijo na različne načine. Najbolj splošno uporabljen je CPI (angl. consumer price index), ki meri inflacijo na strani končne potrošnje gospodinjstev. Za CPI lahko upoštevamo samo jedrno inflacijo, ki izključuje volatilne dobrine, kot so hrana in energija (to so dobrine, katerih cena se hitro in močno spreminja). Uporablja se pa še recimo PPI (angl. producer price index), ki cene meri na strani podjetij in proizvodnje. Če pa želimo izračunati BDP države, moramo upoštevati samo dobrine, proizvedene v njej. Indeksu cen teh dobrin pravimo BDP deflator.

Ko se odločimo, kaj bomo merili, se moramo odločiti kako. Celotno potrošnjo je potrebno razdeliti na kategorije. Treba je izbrati košarico dobrin in formule za agregacijo. V Evropi se uporablja HICP (angl. Harmonised Index of Consumer Prices), standardiziran indeks, ki je enak za vse države. Brez standardov indeksov cen med državami ne bi morali neposredno primerjati. V Sloveniji je temu indeksu ime HICŽP (harmonizirani indeks cen življenjskih potrebščin).

Indeks, ki ga pa v Sloveniji najpogosteje uporabljamo je ICŽP (indeks cen življenjskih potrebščin). Po sestavi in košarici je enak HICŽP, razlikuje se le po utežeh[14]. To je tudi indeks, ki ga bom v sklopu tega projekta meril. Uporabil bom iste uteži in formule za agregacijo.

2.3 Košarica dobrin

Za namene ICŽP merimo inflacijo blaga in storitev na strani gospodinjstev, ki se nahajajo v Sloveniji. Upoštevati moramo samo cene, ki jih potrošniki vidijo s strani podjetij. To vključuje tudi dobrine, ki jih neprofitne organizacije kupujejo v imenu končnih potrošnikov. To so v ZDA recimo cene zdravstva in cene zdravstvenih zavarovanj, plačane preko delodajalca.

Upoštevati moramo samo neto prenose denarja v gospodinjstvih. To pomeni, da nakupi med njimi ne štejejo. Če prvo gospodinjstvo nekaj kupi, je drugo moralo to dobrino prodati. To pomeni, da recimo nakup rabljenega

avtomobila med gospodinjstvi ni vštet v inflacijo. Všteti so pa seveda nakupi rabljenih avtomobilov od podjetij. Zato ne smemo upoštevati cen na spletnih straneh, kot so bolha.com.

Ali je nakup neke dobrine v državi legalen ali ne, nima vpliva na njeno obravnavanje. V teoriji bi morali spremljati tudi cene na črnem trgu (npr. cene prepovedanih drog), v praksi se pa to pokaže za pretežko in tega ne merimo.

Idealno bi bilo imeti podatke cen vseh dobrin, ki se v državi prodajajo. V praksi je to seveda nemogoče, zato se izbere neko omejeno, reprezentativno košarico dobrin, katerih cene se spremlja. V Sloveniji z letom 2018 spremljamo cene 725 dobrin. Za vsako se vsak mesec zbere cene na več lokacijah, skupaj je pa tako zbranih okoli 11.000 cen[16].

2.3.1 ECOICOP kategorije potrošnje

Potrošnja gospodinjstev je kategorizirana v svetovni standard, imenovan COICOP (Classification of Individual COnsumption by Purpose). Evropa ima svojo razširitev, ECOICOP, ki je še en nivo globlja in bolj specifična. Slovenija ima uradno klasificiranih več kot 200 kategorij. Klasifikacija je zelo specifična, obstaja recimo kategorija za umetne noge in steklena očesa. Vključena je vsaka posamezna dobrina, ki predstavlja vsaj 0,1 % potrošnje gospodinjstev[16]. Sama navodila za klasifikacijo so dostopa na spletni strani Eurostata, so zelo podrobna in dolga več kot 20.000 besed.

Primer ECOICOP klasifikacije za pice in pite:

- **Code** 01.1.1.5
- **Description** Pizza and quiche
- **This item includes**
  - farinaceous-based (flour based) products prepared with meat, fish, seafood, cheese, vegetables or fruit

- **This item excludes**
  - meat pies (01.1.2.8)
  - fish pies (01.1.3.6)

- **Caselaw**
  Includes:
  - pizza-like rolls, onion filled rolls, or buns with salty or sweet stuffing
  - filled pancakes, except for pancakes filled with meat (1.1.2.8) or fish (1.1.3.6).

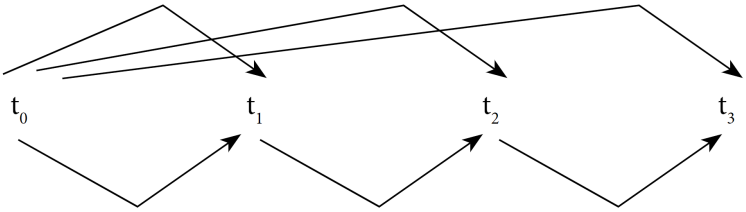
2.4 Kako se inflacija meri

ICŽP je sestavljen iz košarice dobrin, ki je razdeljena na več kot 200 kategorij, recimo kruh, kava, električna energija, pnevmatike in glasbeni instrumenti. Najprej izračunamo spremembo znotraj teh kategorij, za končni indeks pa vzamemo uteženo povprečje vseh.

2.4.1 Veriženje indeksov

Struktura potrošnje se stalno spreminja. Zato je potrebno tudi uteži kategorij in košarico stalno posodabljati. V Sloveniji in v večini držav se ankete o potrošnji izvajajo enkrat letno. Indeks se potem znotraj leta gleda kot cena fiksne košarice glede na december prejšnjega leta. Uteži in struktura košarice se nato decembra spet spremeni in košarica ostane do naslednjega leta nespremenjena. Ker se vsebina košarice spremeni, je treba takrat indekse posameznih let verižiti.

Slika 2.3 kaže razliko pri veriženju. Zgornji indeks se vedno navezuje na cene in košarico iz obdobja t_0, spodnji pa v vsakem obdobju meri spremembo glede na prejšnjega. V Sloveniji se vedno merijo cene glede na december prejšnjega leta. To pomeni, da je decembra vsako leto potrebno zbrati cene za košarico tekočega in naslednjega leta. Ker se vsako leto spremeni le nekaj elementov v košarici, to običajno ni problem.



Slika 2.3: Verižen in neverižen indeks (vir: Eurostat[9]).

2.4.2 Popravki za spremembo kvalitete

Če kakšen artikel v procesu veriženja indeksov zamenjamo, recimo da vzamemo drug televizor za referenco, je seveda možno, da ne stane enako kot prejšnji, niti ne vemo, koliko je stal, preden smo ga začeli spremljati. To spremembo lahko ignoriramo in rečemo, da je celotna sprememba cene samo zato, ker je to drug artikel (celotna sprememba cene je zaradi spremembi kakovosti). Možno je, da so novejši televizorji res cenejši. Za primer vzemimo lanski televizor, ki je črno-bel in nov televizor, ki je barvni. Pričakovali bi, da je barvni model dražji, ker je boljši. Če pa staneta enako, to pomeni, da lahko za isti denar kupimo več – to je pa deflacija. Če upoštevamo to implicitno spremembo cene, temu rečemo hedonični popravek[6].

Računanje hedoničnih popravkov je dokaj nov pristop, recimo v Angliji se ga uporablja od leta 2003, tako da obstajajo države, ki take spremembe še vedno ignorirajo (recimo Danska, Finska in Nizozemska). Količinsko je pa takih dobrin presenetljivo malo (v Angliji predstavljajo le 0,1 % potrošnje)[22].

2.5 Zgodovina statistike cen v Sloveniji

Od leta 1952 do 2005 je SURS računal indeks cen na drobno[8]. To ni bil indeks inflacije, zato so leta 1998 začeli računati še pravi indeks, ki mu danes pravimo ICŽP. Do leta 2017 so bile dobrine klasificirane v COICOP klasifikacijo, po tem letu pa so začeli uporabljati razširjeno verzijo, ECOICOP[20]. Z letom 2018 so začeli računati inflacijo prvih dveh kategorij po ECOICOP klasifikaciji (01 Hrana in brezalkoholne pijače ter 02 Alkoholne pijače in to-

bak) preko podatkovnih baz trgovcev (skenirani podatki). Ostale kategorije se še vedno merijo po tradicionalnih metodah[3].[14]. Uporabljajo tudi avtomatske metode za branje cen na internetu[2], vendar ne računajo nobenega dnevnega indeksa cen.

Skenirani podatki (angl. scanner data)

Danes je Slovenija ena od vodilnih držav na področju statistike cen, saj smo eni redkih, ki za računanje statistike cen uporabljamo podatkovne baze trgovcev. Leta 2017 jih je uporabljala le petina držav v EU[23].

SURS od večjih trgovcev na koncu vsakega tedna prejme podatke o vseh njihovih transakcijah. Tako pridobijo povprečno prodajno ceno vseh prodanih artiklov (upoštevani so vsi popusti) in njihovo količino. Indeksi cen so tako lahko dosti bolj natančni, saj zajemajo dosti več artiklov in upoštevajo strukturo potrošnje bolj podrobno, kot to lahko ugotovimo iz letnih anket o potrošnji. Pomembno je le omeniti, da SURS za izračune v resnici upošteva le prva dva polna tedna v mesecu[14].

2.6 Različne formule za izračun indeksov

2.6.1 Jevonsovova formula

Najprej je potrebno izračunati indeks cen znotraj posamezne kategorije. V Evropi in Sloveniji se to računa kot geometrijsko povprečje razmerij cen vseh dobrin v košarici[13]. Idealno bi bilo, če bi lahko še vsako dobrino utežili glede na njen delež potrošnje. Ker pa teh podatkov nimamo, ne moremo uporabiti uteženega povprečja.

$$P_J = \left(\prod \frac{p_t}{p_0} \right)^{1/n} \tag{2.1}$$

, kjer je p_t trenutna cena dobrine, p_0 cena decembra na začetku obdobja in n število dobrin v kategoriji. V principu bi lahko uporabili navadno aritmetično

[3]Preko telefona, e-pošte, fizično v trgovini ipd.

povprečje cen, ni razloga, da bi to bilo napačno. Se pa izkaže, da če se en izdelek v kategoriji poceni, ga bodo ljudje kupovali več, relativno glede na ostale. Če se pa podraži, ga bodo ljudje kupovali manj. To dinamiko lahko upoštevamo tako, da namesto aritmetičnega povprečja uporabimo geometrijsko povprečje cen (oziroma povprečje relativnih sprememb).

Relativne spremembe cen med sabo pomnožimo in nato izračunamo n-ti koren tega produkta. To je tudi formula, ki se od leta 2017 naprej uporablja v Sloveniji za izračun ICŽP[16].

Rezultat je recimo številka 0.9, kar pomeni, da je ta kategorija dobrin 10 % cenejša glede na december prejšnjega leta.

2.6.2 Laspeyresova formula

Ko imamo izračunane indekse za posamezne kategorije, moramo izračunati še končni indeks. Laspeyresova formula je uteženo aritmetično povprečje cen dobrin v košarici in se v ICŽP uporablja za agregacijo v končni indeks. Splošna formula je:

$$P_L = \frac{\sum(p_t \cdot w_0)}{\sum(p_o \cdot w_o)} \tag{2.2}$$

, kjer je p_t je trenutna cena artikla, p_0 je cena v prvem obdobju, w_0 je pa utež artikla v košarici in se skozi leto ne spreminja.

V primeru ICŽP je p_t že indeks kategorije glede na prvo obdobje (december prejšnjega leta), tako da je $p_0 = 1$. Formula se pokrajša v:

$$P_L = \sum(p_t \cdot w_0) \tag{2.3}$$

2.6.3 Fisherjev idealni indeks

Laspeyresova formula upošteva samo uteži v prvem obdobju. Ker pa potrošniki trošijo manj dobrin, ki se dražijo, bi bilo bolje, če bi upoštevali utež v zadnjem obdobju. To bi seveda podcenilo inflacijo, ker ne bi upoštevali, da so ljudje spremenili okus prav zaradi višanja cen. Fisherjev "idealni"indeks je zgrajen tako, da zračunamo en indeks z začetnimi utežmi in enega s končnimi,

potem pa vzamemo njuno geometrijsko povprečje[10]. To je formula, ki jo uporablja The Billion Prices Project[5]:

$$P_F = \sqrt{P_L \cdot P_P} \tag{2.4}$$

, kjer je P_L indeks, zračunan z utežmi iz prvega obdobja, P_P pa iz zadnjega.

V praksi pa žal nimamo uteži za trenutno obdobje, ker rezultate anket o potrošnji dobimo šele nekaj mesecev kasneje, zato se takih indeksov ne uporablja – vsaj ne za ocene inflacije, ki izhajajo mesečno (angl. flash estimate). Kasneje, ko so te uteži na voljo, pa jih seveda lahko uporabimo, kot to recimo v ZDA dela BLS z indeksom C-CPI (chained CPI). Uporabljajo pa malo drugačno, Törnqvistovo formulo[3].

Poglavje 3

Naš pristop za merjenje inflacije

3.1 Zbiranje podatkov

Cene zbiramo vsak dan na devetih straneh:

- Mercator spletna trgovina
- Mercator M Tehnika
- Mimovrste
- Big Bang
- Spar
- Lastra (glasbeni inštrumenti)
- Varcuj24 (elektrika)
- promet.si (cene goriva)
- moje-lece.si

Mercator in Spar smo izbrali, ker sta edini trgovini na drobno v Sloveniji s cenikom na spletu. Ostale trgovine so dokaj specifične (recimo Lastra) in so dodane samo zato, da lahko pokrijemo večji delež potrošnje. Moje-lece.si

je v času opazovanja spremenila samo eno ceno, zato je v izračunih nismo upoštevali.

Morda zveni dobro, da bi cene zbirali na eni centralni spletni strani, kot je to ceneje.si. Vendar je v samem principu računanja inflacije pomembno, da merimo le ceno dobrin, ki jih ljudje kupujejo. Mimovrste ima recimo po podatkih AJPES-a veliko prometa[1], portal ceneje.si pa vključuje mnogo majhnih lokalnih trgovin, ki so sicer morda cenejše, ampak imajo malo prometa.

Nasploh smo se pa želeli izogniti zanašanju na sisteme tretjih oseb, ker tako težko zagotovimo zanesljivost podatkov. Omenjeni portal med drugim nima datoteke sitemap.xml, kar rahlo oteži delo spletnega pajka.

Najpomembnejši razlog, da ne uporabljamo spletnega portala ceneje.si je pa, ker presenetljivo ne pokrije dosti končne potrošnje, le približno 20 %. Spletna trgovina Mercator, ki je ceneje.si ne spremlja, pokrije več kot 30 % potrošnje gospodinjstev.

3.2 Agregiranje podatkov

Artikli so najprej razvrščeni v ECOICOP kategorije, vsak v eno. Ob času pisanja je bilo kategoriziranih 50 % vseh dobrin, ki jih spremljamo.

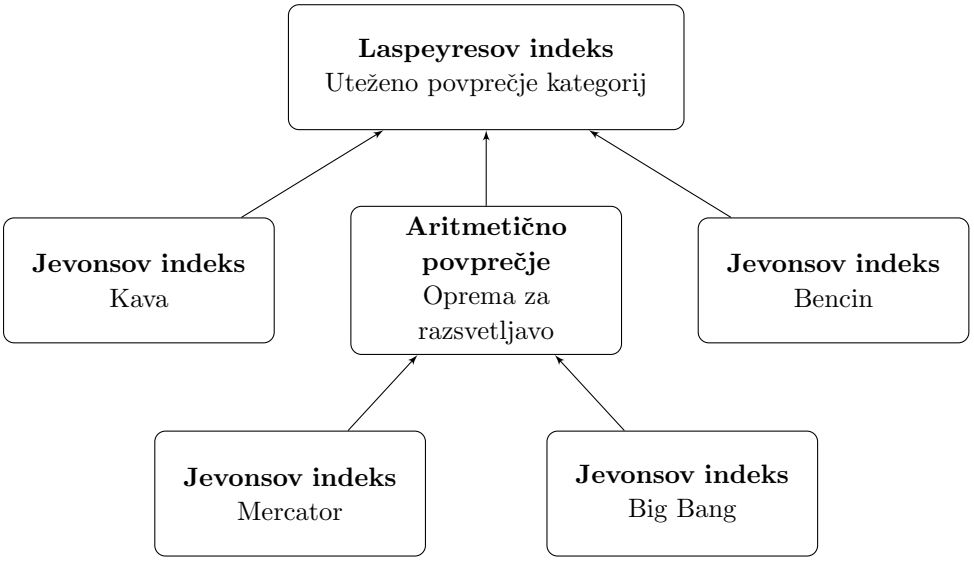
Za vsako kategorijo se najprej zračuna indeks cen glede na prvo obdobje z Jevonsovo formulo (geometrijsko povprečje razmerij cen). Če dve trgovini prodajata artikle v isti kategoriji, se izračuna Jevonsov indeks za vsako posebej in se za končni rezultat vzame navadno neuteženo aritmetično povprečje obeh.

V izračunu se upoštevajo vse dobrine, ki so naprodaj v trenutnem obdobju. Tiste, ki trenutno recimo niso na zalogi, se v izračunu ne upoštevajo. Tako kot The Billion Prices Project, tudi mi ignoriramo vse podražitve, ki so večje kot 200 % ali pocenitve za več kot 70 %. S tem izločimo približno 1 % vseh meritev. Take spremembe so bolj verjetno napake v ceniku kot realno

[1] Čisti prihodki od prodaje leta 2016 so znašali slabih 44 milijonov €

stanje.

Za končni rezultat izračunamo Laspeyresov indeks kategorij (uteženo aritmetično povprečje). Diagram 3.1 prikazuje poenostavljeno shemo agregacije.



Slika 3.1: Agregacija cen.

3.3 Branje cenikov

Večina strani ima na naslovu /robots.txt napisan naslov datoteke sitemap.xml. V njej je napisan seznam vseh URL naslovov na tej domeni, ki so namenjeni za spletne pajke. Njegov naslov smo ročno vpisali v program. V večini primerov njegovo vsebino razčlenjujemo kar z regularnimi izrazi. Primer za razčlenjevanje URL naslovov za Mercator spletno trgovino:

```
<loc>(.*/izdelek/.*)</loc>
```

Nekatere strani, recimo Mimovrste, imajo sitemap.xml strukturiran v več nivojih. Prvi dokument kaže na 1000 drugih, v katerih so zapisani URL-ji artiklov. Zaradi preprostosti je algoritem za razčlenjevanje napisan za vsako spletno stran posebej.

Najbolj izstopa Spar, ker nima nikjer objavljenega seznama artiklov. V robots.txt imajo še zapisano, naj spletni roboti izvedejo le en HTTP zahtevek na 10 sekund. Za pregled celotne strani bi zato potrebovali skoraj tri dni. Stran je zelo neprijazna za pajke, saj se vsi zahtevki izvajajo preko AJAX-a. To pomeni, da pajek ne more preprosto slediti vsem HTML značkam `<a href>`, ker se vse izvaja preko JavaScripta. Za indeksiranje te spletne strani je bilo potrebno ročno ugotoviti, kako se AJAX zahtevki izvajajo, in ročno razčlenjevati JSON, ki ga stran vrne. Na srečo stran vrača podatke po 12 artiklov hkrati, tako da tudi če upoštevamo zahtevo enega zahtevka na 10 sekund, uspemo indeksirati celotno stran v približno 8 urah.

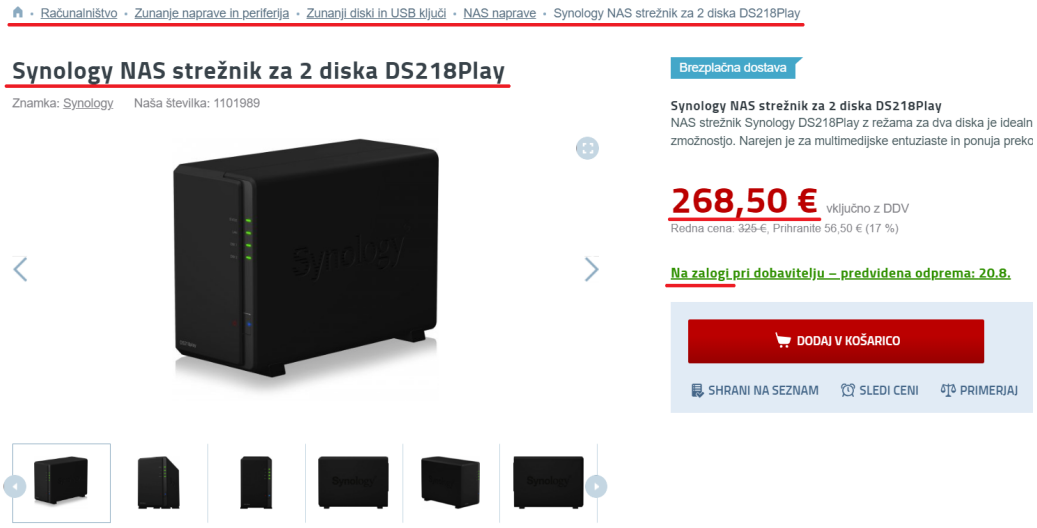
Sliki 3.2 in 3.3 prikazujeta podatke, ki se na spletni strani berejo. Včasih so podatki na strani spletnim pajkom še bolj prijazno napisani v izvorni kodi strani (slika 3.4).

3.3.1 Hitrost branja cen

Pomembno je, da cene iz spleta beremo dovolj hitro, da jih lahko preberemo v enem dnevu, ne smemo pa pošiljati toliko zahtevkov hkrati na spletno stran, da bi s tem vplivali na druge uporabnike. Večina portalov ima dovolj malo URL naslovov, da je dovolj, če beremo samo eno ceno hkrati. Mercator recimo ima približno 20.000 unikatnih spletnih naslovov (slika 3.5), kar pomeni, da če pošljemo en zahtevek na sekundo, bomo v manj kot 6 urah prebrali vse cene. Če se skripta zažene ob 4.00 zjutraj, bodo do 10.00 vse cene že prebrane.

Mimovrste

Daleč največ artiklov prodajajo Mimovrste, vsak dan okoli 160.000 (od tega jih je na prodaj oz. na zalogi približno 90.000). Skripta je zato sposobna večnitnosti že na nivoju procesiranja posameznega portala. Za Mimovrste se izkaže, da je dovolj, če tečeta 2 niti hkrati. Sam spletni portal deluje zelo hitro, tako da že 2 niti pomenita 4 zahtevke na sekundo. Za vsak slučaj,

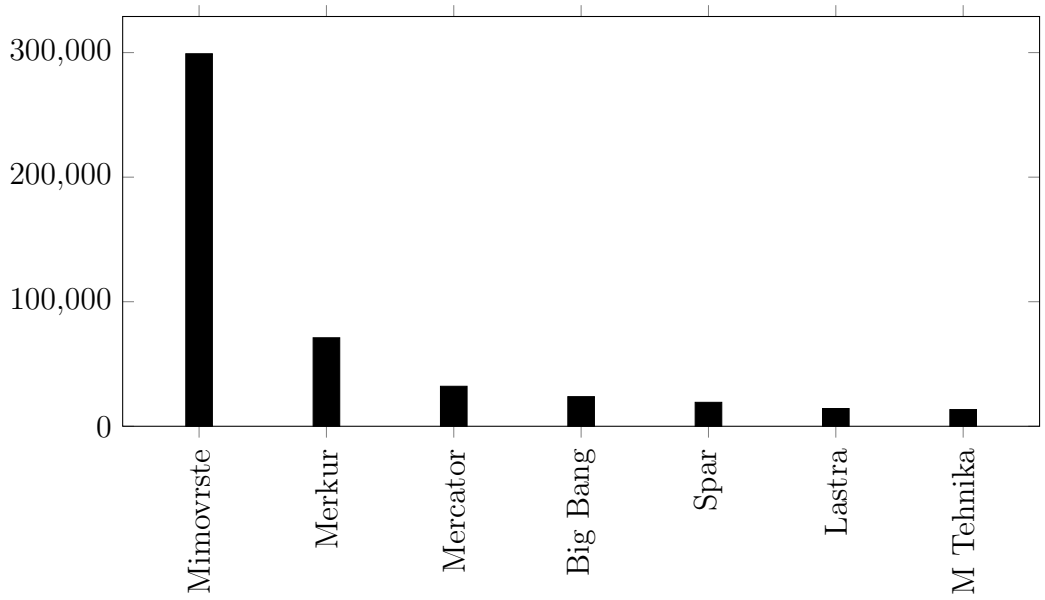


Slika 3.2: Spletna stran Mimovrste.

Slika 3.3: Spletna stran Mercator.

```
<div class="lay-none" itemprop="offers" itemscope itemtype="http://schema.org/Offer">
  <span itemprop="priceCurrency" content="EUR"></span>
  <span itemprop="price">268.5</span>
</div>
```

Slika 3.4: Izvorna koda trgovine Mimovrste.



Slika 3.5: Število unikatnih URL naslovov.

v primeru, da spletna stran postane neodzivna za nekaj ur (kar se je zgodilo najmanj enkrat), pa uporabljamo 3 niti, kar pomeni približno 6 do 8 zahtevkov na sekundo.

Največ, kar smo poizkusili, je bilo 7 vzporednih niti. To je bilo skupaj več kot 30 zahtevkov na sekundo in več kot 10 Mbps prometa na mrežni kartici. Opazili pa nismo nobenega znaka upočasnjevanja delovanja skripte ali pa spletne strani.

Strani z manj unikatnimi naslovi

Na drugi strani imamo M Tehniko, ki ima le približno 10.000 unikatnih URL naslovov. Ena nit hkrati konča s procesiranjem v približno dveh urah. Ker smo želeli še dodatno razbremeniti spletno stran, smo med vsakim zahtevkom dodali še eno sekundo zakasnitve.

Največ zakasnitve med zahtevki (10 sekund) ima Spar, saj so tako željo izrazili v datoteki robots.txt.

Merkur

Merkur spletna trgovina je bila žal premalo odzivna in je s tremi vzporednimi nitmi komaj končala procesiranje v enem dnevu. Ko smo preučevali dnevniške datoteke, smo videli, da je spletna stran čez dan večkrat postala popolnoma neodzivna. Za to je bila skoraj gotovo kriva naša skripta, zato smo se odločili, da te strani ne bomo spremljali.

Skupaj ustvarjen mrežni promet

Celoten promet, ustvarjen zaradi izvajanja skripte je le nekaj Mbps. Kljub temu, da ob največji hitrosti (ob zagonu) pošilja več kot 10 zahtevkov na sekundo, je ta številka še vedno razumljiva, saj je edina stvar, ki se prenaša HTML izvorna koda, brez slik, JavaScript-a in CSS oblikovnih datotek. S temi informacijami lahko sklepamo, da bi se skripta lahko brez problema izvajala tudi na počasnejših internetnih povezavah.

3.3.2 Večnitnost in vzporedno procesiranje

Ogromno število unikatnih spletnih naslovov, ki jih je potrebno vsak dan obiskati, nujno zahteva večnitno procesiranje. Ena sama nit bi za branje potrebovala več dni, preden bi se ustavila. Program je bil tako že v osnovi zasnovan, da podpira večnitno procesiranje, ne samo na nivoju posameznih spletnih portalov, ampak tudi večnitno pošiljanje zahtevkov na posamezno spletno stran.

3.4 Izbira SUPB-ja

SUPB, ki ga uporabljamo je MariaDB. SQLite je zagotovo premalo zmogljiv za potrebe programa, saj nima niti podpore hkratnemu pisanju in vzporednemu izvajanju transakcij. Drugi, večji sistemi so ali plačljivi, ali pa niso na voljo na sistemu, kjer se skripta izvaja (Synology NAS).

Vrsta tabele, ki jo uporabljamo, je InnoDB, ker je med drugim tudi priporočena s strani avtorjev sistema MariaDB. MyISAM ne bi bila primerna, saj zapise zaklepa na nivoju tabele, namesto na posameznih zapisih; to bi močno upočasnilo izvajanje programa in zagotovo vodilo v napake zaradi čakanja (angl. deadlock). Tudi tujih ključev ne podpira, ki so nujno potrebni za zagotovitev integritete podatkov.

3.4.1 Vzporedni dostop do podatkovne baze

Kljub skrbnemu načrtovanju vzporednega dostopa, se včasih zgodi napaka, podobna:

```
MySQLTransactionRollbackException: Lock wait timeout
exceeded; try restarting transaction
```

Ta napaka je pričakovana in ne pomeni, da smo naredili napako pri programiranju. Včasih se pač zgodi, da se tabele zaklenejo ravno ob napačnem trenutku. Nič slabega se ni zgodilo, le ponovno moramo zagnati transakcijo in verjetno se bo tokrat uspešno izvedla. Lahko se jim pa probamo izogibati, tako da se tabele (in vrstice) pri pisanju zaklepajo vedno v istem vrstnem redu. Če programiramo s funkcijami in ne podvajamo kode, ki je povsod malo drugače napisana, smo večino dela že naredili. Druga stvar, ki jo lahko naredimo je, da so transakcije čim krajše. Hitreje, kot se izvedejo, manjša je verjetnost, da se bosta izvajali dve hkrati. V primeru, da se pa ta napaka le zgodi, skripta na kritičnih točkah (recimo zagon in shranjevanje cene) v zanki for večkrat poizkusi izvesti transakcijo.

V najhujšem primeru, da se celotna skripta konča z napako, tudi ni nič hudega, ker se bo jutri ponovno zagnala, prebrala cene in kljub napaki naslednji dan pravilno izračunala indeks cen.

3.4.2 Indeksi

Indeksirani so privzeto že seveda vsi tuji ključi. Potreben je bil še en unikaten indeks na URL naslovih v tabeli artiklov. Ta atribut je bil že od začetka tipa

VARCHAR(255). To je bilo ravno dovolj, ker je bil najdaljši zabeležen URL dolg 249 znakov. V primeru, da bi morali polje podaljšati, MariaDB podpira velikost VARCHAR do 65.535 znakov, oziroma 21.844 za utf8 polja[21].

UTF8 in velikost indeksov

Kodna tabela, ki smo jo na začetku uporabili, je bila pa utf8. Ta zapis uporabi enega do tri bajte na znak, kar pomeni, da je lahko največja dolžina atributa na disku v resnici trikrat daljša. Navadnega ASCII kodiranja nismo mogli uporabiti, ker ima Mimovrste približno 20 artiklov, ki imajo v URL naslovu šumnike.

Žal indeksi v InnoDB nimajo dinamične dolžine in zasedejo kar maksimalno, trikratno, dolžino (plus dva bajta), torej 767 bajtov na zapis[17]. Pri 400.000 artiklih to pomeni, da je samo indeks za URL naslove velik več kot 300 MB. To za moderne računalnike z veliko spomina sploh ni problem, mi smo pa program poganjali na majhnem NAS strežniku z 1 GB RAM-a. Ker SUPB indeksa ni mogel zadržati v glavnem spominu, je skripta sčasoma začela delovati občutno počasneje. Kodno tabelo za URL naslove smo zato spremenil v ASCII. Tistih 20 artiklov s šumniki sedaj ni možno zapisati v podatkovno bazo, se je pa velikost indeksov na tej tabeli zmanjšala na 85 MB.

Alternativne oblike indeksov

Trenutna rešitev ima dva potencialna problema:

1. Večina URL naslovov ima prvih 35 znakov enakih, kar vpliva na hitrost iskanja.

2. Če bi se odločili povečati dolžino polja za URL naslove, bi bilo predolgo, da bi se na njem lahko ustvaril indeks.

Preprosta rešitev, ki se lahko uporabi je indeks na pomožnem polju, ki je MD5 izvleček originalnega. Nov indeks bi bil tako dolg le 32 znakov, njegova vsebina bi bila pa enakomerno razpršena po celotnem prostoru.

3.5 Poraba pomnilnika in čiščenje smeti

Manjši problem predstavlja dejstvo, da program v spominu stalno shranjuje vsebino spletnih strani, jih razčleni in potem zavrže. Velikost modernih spletnih strani se meri v MB, kar pomeni, da poraba pomnilnika narašča s hitrostjo več kot 10 MB na sekundo. Pomnilnik se tako polni od približno 140 MB do 250 MB, potem se zažene čiščenje (angl. garbage collection), nakar se proces ponovi.

Analiza s programskim orodjem Eclipse Memory Analyzer potrdi, da večino pomnilnika porablja knjižnica za razčlenjevanje HTML-ja, JSoup. Žal nismo našli nobenega načina, da bi lahko ta pomnilnik ročno počistili ali ponovno uporabili.

Seznami URL naslovov

Drug velik vir porabe pa predstavljajo seznami URL naslovov v pomnilniku. Hitra ocena potrdi, da približno 40 MB spomina zasedejo samo URL-ji:

$$\begin{aligned} \text{Število URL naslovov v programu} &= 240.000 \\ \text{Povprečna dolžina URL naslova} &= 83 \text{ znakov} \\ \text{Število bajtov na znak} &= 2 \text{ (Java uporablja UTF16)} \\ 240.000 \times 83 \times 2 &\approx 40 \text{ MB} \end{aligned}$$

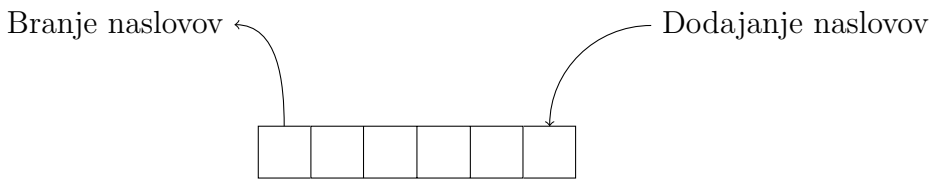
Program uporablja dve vrsti URL naslovov (glavna in pomožna), vendar ti vrsti vsebujeta samo referenci na naslove (objekte String). Tudi če se vsi podvojijo, so to le reference, sam naslov je v spominu zapisan le enkrat.

Nekatere Java implementacije podpirajo uporabo posebne zastavice `-XX:+UseCompressedStrings`, ki samodejno uporabi tabele `byte[]` za objekte String, kjer je to mogoče (besedila, ki uporabljajo samo ASCII znake). Implementacija, ki jo mi uporabljamo, tega ne podpira, tako da bi lahko nekaj takega naredili le ročno.

3.6 Vrste in večnitno procesiranje

Program za svoj glavni del uporablja samo eno glavno vrsto (angl. queue) spletnih naslovov, iz katerega lahko več niti bere sočasno. Implementacija, ki jo uporabljamo je `LinkedBlockingDeque` (slika 3.6). To je linearni dvostransko povezani seznam, ki je narejen za delo v večnitnem okolju. Tako implementacijo potrebujemo, ker se velikost seznama dinamično spreminja, elementi se pa včasih dodajajo na začetek, včasih pa na konec (recimo v primeru napake, ko želimo, da se isti naslov obišče ponovno kasneje).

Poleg glavne uporabljamo še pomožno množico obiskanih URL naslovov, s katerim preprečimo, da bi isti naslov obiskal dvakrat. Iz glavne vrste se namreč elementi sproti brišejo. To je uporabno v primeru, da spletna stran nima datoteke sitemap.xml in želimo narediti pajka, ki preprosto sledi vsem povezavam, ki jih najde. Iskanje obiskanih URL naslovov po linearno povezanem seznamu bi imelo časovno kompleksnost $O(n)$, kar je absolutno prepočasno. Idealen bi bil razred z imenom `ConcurrentHashSet`, ker bi imel časovno kompleksnost $O(1)$, vendar tak razred v standardnih knjižnicah žal ne obstaja. Obstaja pa `ConcurrentSkipListSet`, s kompleksnostjo $O(log\ n)$, kar je pa sprejemljivo.



Slika 3.6: Linearno povezani seznam

Dodajanje elementov v vrsto

Vrsta se že pred začetkom izvajanja napolni z enim ali več URL naslovi. To je lahko samo datoteka sitemap.xml, lahko je naslov več izhodiščnih strani (recimo pri Sparu), v vsakem primeru se pa v vrsto dodajo vsi naslovi, ki so bili ob zadnjem obisku dosegljivi. Tudi če naslova ni več v datoteki site-

map.xml, ga moramo nujno vsaj še enkrat obiskati. Na nekaterih spletnih trgovinah se seznam namreč generira dinamično in je možno, da je več URL naslovov čisto slučajno izpuščenih. Če pa artikel ni več naprodaj, si bomo lahko shranili napako „HTTP 404 Not Found“, ki jo bo stran vrnila.

V primeru drugih napak, recimo ob poteku časovne omejitve (angl. timeout), se element doda na konec vrste, da bo lahko procesiran še enkrat.

Branje elementov iz vrste

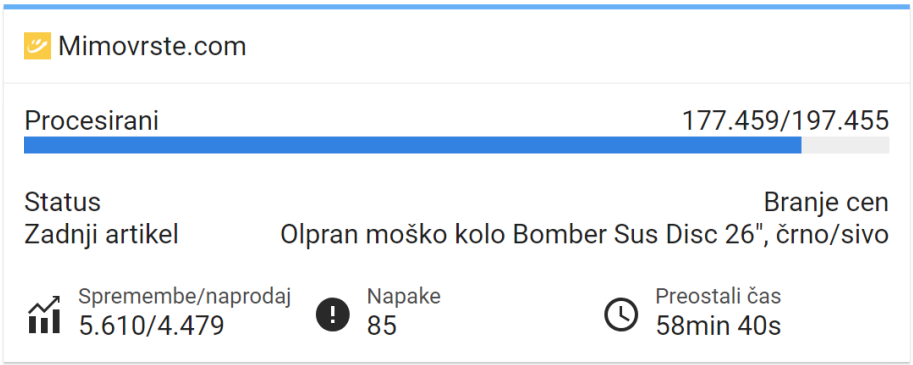
Če posamezna nit pride do konca vrste, to ne pomeni, da je dela konec. Morda se datoteka sitemap.xml še ni procesirala do konca in bo še dodala elemente čez nekaj trenutkov. Razred `LinkedBlockingDeque` to lepo reši s funkcijo `take()`. Njen klic vrne prvi element iz vrste, oz. če je prazna, počaka, da bo element na voljo.

Pomembno je, da se nit na koncu ustavi, da se lahko izvajanje programa nadaljuje. Rešitev je poseben element čisto na koncu vrste, recimo prazen URL naslov, ki ustavi izvajanje trenutne niti (angl. poison pill). To deluje dobro in zanesljivo, vendar ne omogoča dodajanja elementov na konec vrste. Zato uporabljamo funkcijo `poll()`. Ta vrne prvi element oziroma `null`, če je vrsta prazna. V zanki enkrat na sekundo poizkuša dobiti element (angl. polling). Če ugotovi, da vse niti čakajo na nov element, se izvajanje ustavi in nadaljuje z računanjem inflacije.

3.7 Spremljanje stanja procesiranja

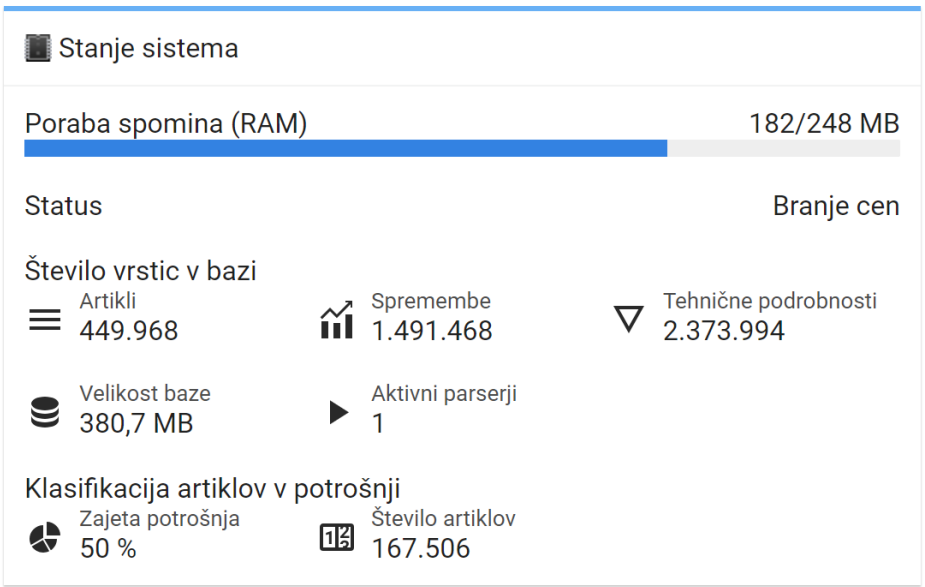
Naredili smo javno dostopno spletno stran, kjer se stanje izvajanja lahko spremlja. Ogledamo si lahko tudi indekse posameznih kategorij (slika 3.9). Modra črta prikazuje naš indeks cen, črna črta je uradni indeks, siva črta in desna os pa prikazujeta število dobrin, vključenih v izračun spletnega indeksa. Padci sive črte jasno prikazujejo začetke meseca in veriženje indeksa, ko se artikli, ki niso več naprodaj, izključijo iz izračuna.

Branje cen za spletno stran Mimovrste se na sliki 3.7 približuje koncu. Druge strani so običajno zdaj že končale in čakajo na nadaljnje procesiranje.

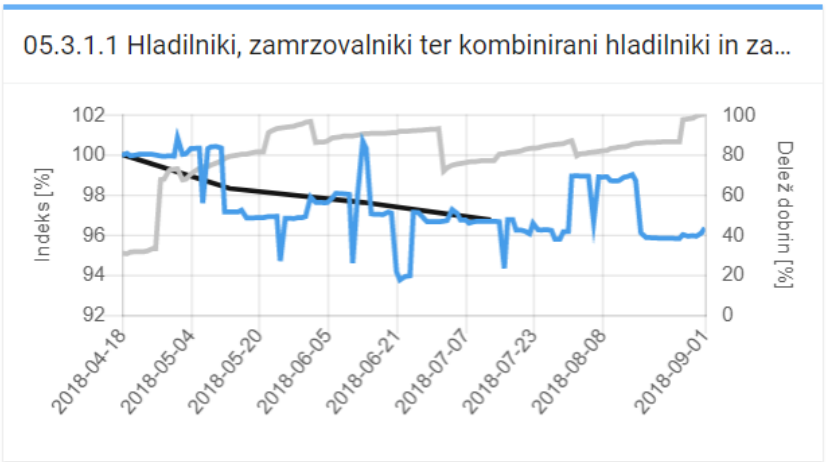


Slika 3.7: Prikaz stanja procesiranja ene strani.

Sistem, na katerem skripta teče ima le 1 GB spomina, zato je ključno, da spremljamo njegovo porabo (slika 3.8). Spremljamo lahko tudi ostale zanimive podatke, kot so število vrstic v izbranih tabelah in koliko dobrin je zajetih v računanju inflacije.



Slika 3.8: Prikaz stanja sistema.



Slika 3.9: Prikaz indeksa cen ene kategorije.

3.7.1 Ocenjevanje preostalega časa procesiranja

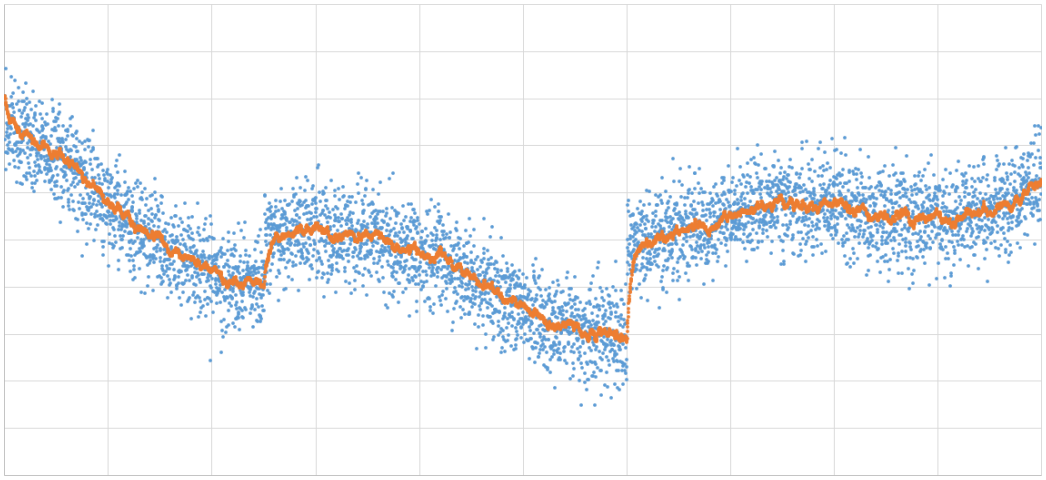
Na spletni strani želimo prikazati, približno koliko časa se bo skripta še izvajala. Najbolj osnovni način za računanje ocene je, da porabljen čas za zadnji procesiran element pomnožimo s številom preostalih elementov:

$$x_t = t \times n \tag{3.1}$$

, kjer je x_t ocena preostalega časa, t porabljen čas zadnjega elementa in n preostalo število elementov. Problem je, da je varianca med posameznimi ocenami tako velika, da je skoraj neuporabna. Lahko bi računali rezultat glede na skupaj porabljen čas za vse že procesirane elemente, vendar bi tako dobili napačno oceno, če bi se procesiranje na koncu močno upočasnilo, kot se to zgodi pri Sparu. Lahko vzamemo tekoče povprečje zadnjih nekaj elementov, vendar je tudi to preveč zahtevno. Obstaja še bolj preprost način – nizkoprepustni filter (angl. low-pass ali high-cut filter):

$$x_t = \alpha x_{t-1} + (1 - \alpha) x_t \tag{3.2}$$

Vsaka nova ocena x_t upošteva nek faktor α prejšnje ocene x_{t-1} in $(1 - \alpha)$ trenutne ocene x_t, izračunane s formulo 3.1. Ker je število spletnih naslovov



Slika 3.10: Primer delovanja nizkoprepustnega filtra.

in varianca zelo velika, uporabljamo faktor $\alpha = 0.003$. To pomeni, da nova ocena vpliva samo 0,3 % na končen rezultat. Formula 3.2 je enostavna, ker si rabimo zapomniti samo prejšnjo oceno preostalega časa.

Na sliki 3.10 lahko vidimo, kako filter ignorira šum, ampak se vseeno hitro prilagodi na spremembe v oceni preostalega časa.

Če si ocene preostalega časa predstavljamo kot signal, nam nizkopasovni filter poreže njegove visoke frekvence. Faktor α spreminjamo, dokler ne dosežemo signala, ki je dovolj stabilen za naš okus.

3.8 Klasifikacija v ECOICOP kategorije

Vse spletne strani, ki jih spremljamo, imajo artikle že kategorizirane tako, da skoraj popolnoma ustrezajo ECOICOP klasifikaciji. To nam je prihranilo mnogo dela, ker smo morali povezati le imena kategorij s klasifikacijo. Če ne bi bilo take klasifikacije, bi lahko uporabili algoritme strojnega učenja, kot to predlaga Eurostat za procesiranje skeniranih podatkov[23]. Naredili smo si orodje, kamor napišemo iskalni niz (recimo mleko) in nam je vrne seznam vseh artiklov s to besedo v imenu. Potem smo ročno pregledali, katere kategorije je smiselno vključiti.

Kategorija Sveže ali ohlajeno sadje tako vključuje vse artikle v Mercatorju, katerih ime kategorije se začne z:

```
SADJE IN ZELENJAVA > SVEŽE SADJE > GROZDJE
SADJE IN ZELENJAVA > SVEŽE SADJE > JABOLKA, HRUŠKE
SADJE IN ZELENJAVA > SVEŽE SADJE > BANANE IN AGRUMI
...
```

Napisanih imamo več kot 750 takih klasifikatorjev.

3.8.1 Izvedba klasifikacije

Vsi klasifikatorji za ECOICOP kategorije so napisani v podatkovni bazi, prav tako kot vsi artikli. Algoritem za klasifikacijo je napisan kar v SQL jeziku, ker je tako najenostavnejše in verjetno najhitreje. Podatkovne baze so namreč optimizirane za delo s podatki. V primeru, da za procesiranje ni dovolj prostora v pomnilniku, se podatki avtomatsko zapišejo na trdi disk, kjer se skripta lahko izvaja naprej.

SQL skripta za klasifikacijo artiklov:

```
UPDATE products p
INNER JOIN (
    SELECT * FROM ecoicop_clasificators WHERE store = ?
) c ON(
    p.category LIKE CONCAT(c.category_starts_with, "%")
    AND (c.title_contains IS NULL OR
         p.title LIKE CONCAT("%", c.title_contains, "%"))
)
SET p.ecoicop = c.ecoicop
WHERE p.store = ?
AND c.priority = ?
```

Skripta se izvede večkrat, za vsako trgovino enkrat ali dvakrat, ker so klasifikatorji napisani po prioritetah. Kljub preprosti implementaciji pa celotna klasifikacija za 400.000 artiklov traja približno 20 minut.

3.9 Popravki za spremembo kvalitete

Obstaja nekaj dokazov, da indeksi, ki vsak mesec zamenjajo in verižijo košarico (angl. Monthly Chaining and Resampling) oz. uporabljajo metode mostičenja (angl. Bridged Overlap) dajejo podobne rezultate kot hedonični indeksi, s tem da so še bolj preprosti[11, 12]. To velja za kategorije, kjer se generacije novih in starih produktov prekrivajo (recimo elektronika)[5, 12]. V tem projektu uporabljamo prvo metodo. Prvi dan v vsakem mesecu iz košarice odstranimo vse dobrine, ki niso več naprodaj in mesečne indekse med sabo verižimo.

Problem bo nastal, ko bo podatkov za več kot eno leto. Pojavila se bodo nova oblačila, ki jih ne bo možno direktno povezati s prejšnjo sezono. Zato že zdaj shranjujemo ne samo cene artiklov, ampak tudi vse njihove tehnične podrobnosti (npr. velikost, barva, material, dolžina rokavov), v primeru da so na spletni strani jasno napisane[2]. Iz izbranih podatkov izračunamo koeficiente vpliva posameznih parametrov (recimo majice z dolgimi rokavi so v povprečju 10 % dražje od majic s kratkimi rokavi). Iz vseh zbranih koeficientov in začetne vrednosti (angl. intercept) lahko napovemo ceno majice. Inflacija je nato izračunana kot razlika med dejansko in napovedano ceno.

3.10 Računanje inflacije

Priprava podatkov

Podatki se originalno nahajajo v eni relacijski tabeli. Vsak zapis vsebuje vsaj ID artikla, datum, ceno in status, ali je artikel na zalogi. Pridobiti želimo ceno, ko je bil artikel prvič na zalogi in ceno na današnji datum. Artiklov, ki danes niso na zalogi ne želimo upoštevati.

Na prvi pogled zgleda, da lahko samo uporabimo SQL funkcijo `MIN()`, vendar bomo hitro naleteli na težave. Zanima nas namreč vrednost atributa cena, kjer je atribut čas najmanjši, grupirano po ID-ju zapisa. Funkcija `MIN()`

[2]Znamka oblačil se ne upošteva.

nam vrne le najmanjšo vrednost izbranega atributa, ne pa nekega drugega. Srečali smo se s problemom „greatest-n-per-group“.

Da dobimo želen rezultat, moramo združiti tabelo samo s seboj z vgnezdenim selectom in z ukazom `INNER JOIN`. Združevali bomo po atributih ID dobrine in po času. Notranji `SELECT` nam bo vrnil najmanjši čas za vsak artikel, zunanji pa ceno pri tem času:

```
SELECT c.product, c.price, min_time AS time
FROM readings c
INNER JOIN (
    SELECT product, MIN(time) AS min_time
    FROM readings
    WHERE in_stock = 1
    GROUP BY product
) a ON (c.product=a.product AND c.time=a.min_time)
```

Podobno skripto uporabimo še, da dobimo cene dobrin, ki so danes na zalogi. Oba rezultata shranimo v eno začasno tabelo. Končna tabela vsebuje celotno košarico, z vsako dobrino zabeleženo dvakrat (oziroma enkrat, če je danes ni na zalogi). En zapis ima začetno ceno, drugi končno.

Vrtilne tabele v SQL

Zapisi so zdaj podvojeni, želimo jih grupirati po dobrinah. Končna oblika tabele bo vsebovala ime dobrine, začetno ceno in končno ceno, vse v eni vrstici (glej tabeli 3.1 in 3.2). Navaden `GROUP BY` ne bo dovolj, ker želimo isti atribut (cena) enkrat shraniti v atribut začetna cena, drugič pa v končna cena. Za rešitev uporabimo SQL ukaz `CASE`. V naši tabeli med začetno in končno ceno ločimo po atributu time. Začetna cena ima vpisana datum, kdaj je bila dobrina prvič na zalogi, končna cena pa ima ta atribut nastavljen na `NULL`.

Naziv	Datum	Cena
Pizza	1.5.2018	5 €
Pizza	2.5.2018	6 €
Čokolada	1.5.2018	10 €
Čokolada	2.5.2018	8 €

Tabela 3.1: Tabela pred vrtenjem.

Naziv	Cena 1	Cena 2
Pizza	5 €	6 €
Čokolada	10 €	8 €

Tabela 3.2: Tabela po vrtenju.

Za vrtilno tabelo uporabimo preprost trik: uporabimo `GROUP BY` in seštevamo v enem primeru ceno, drugič pa 0, odvisno od atributa time.

```
SELECT
    product,
    SUM(CASE WHEN time IS NOT NULL THEN price ELSE 0 END)
      AS prev_price,
    SUM(CASE WHEN time IS NULL THEN price ELSE 0 END)
      AS today_price
FROM cene_temp
GROUP BY product
```

Geometrijsko povprečje v SQL

MySQL nima podpore za računanje geometrijskega povprečja, zato ga moramo izpeljati. Produkt več števil lahko zapišemo tudi kot vsoto njihovih logaritmov. To vsoto nato delimo s številom elementov, kar predstavlja n-ti koren števila. Rezultat nato potenciramo s številom e[19].

```
EXP(SUM(LOG(today_price/prev_price)) / COUNT(*))
```

Končni rezultat

Ko ugotovimo spremembe po posameznih kategorijah, za končni rezultat izračunamo aritmetično povprečje posameznih indeksov. Rezultat nam pove, koliko dražja ali cenejša je ista košarica dobrin danes glede na začetno obdobje.

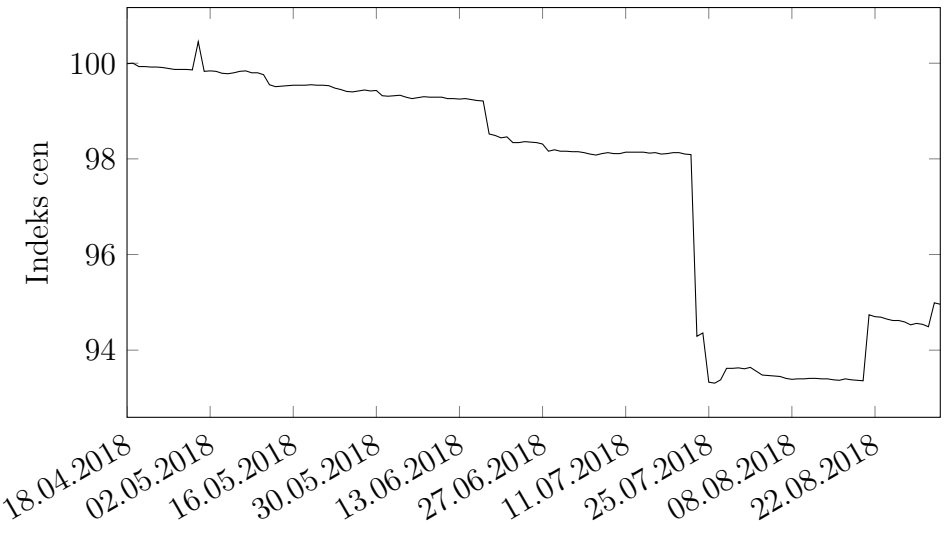
Poglavje 4

Rezultati

4.1 Indeksi posameznih trgovin

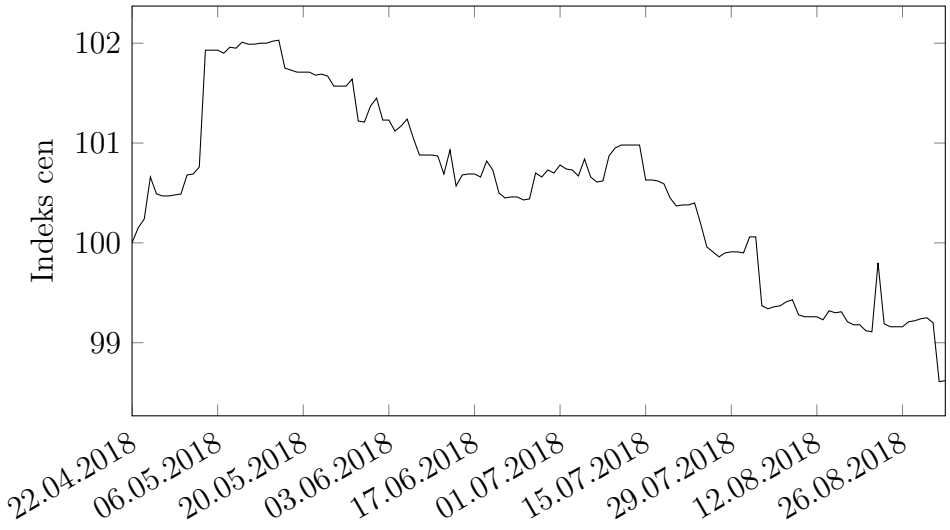
Dobrine, ki jih trgovine prodajajo so različne, zato se tudi indeksi med seboj seveda ne ujemajo. Zato lahko razberemo tudi vzorce gibanj cen posameznih trgovcev.

Padec indeksa konec julija za trgovino Mimovrste (graf 4.1) v celoti pojasnimo s sezonskimi pocenitvami oblačil in obutev.



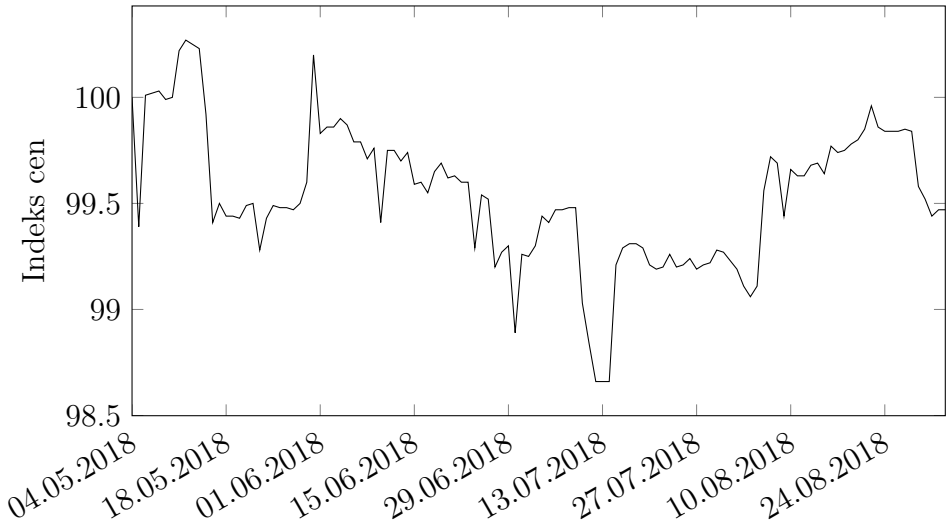
Slika 4.1: Indeks cen za Mimovrste

Na začetku smo opazili močno podražitev cen v trgovini Mercator (graf 4.2). Cene so se nato znižale, tam ostale en mesec in se ponovno znižale.



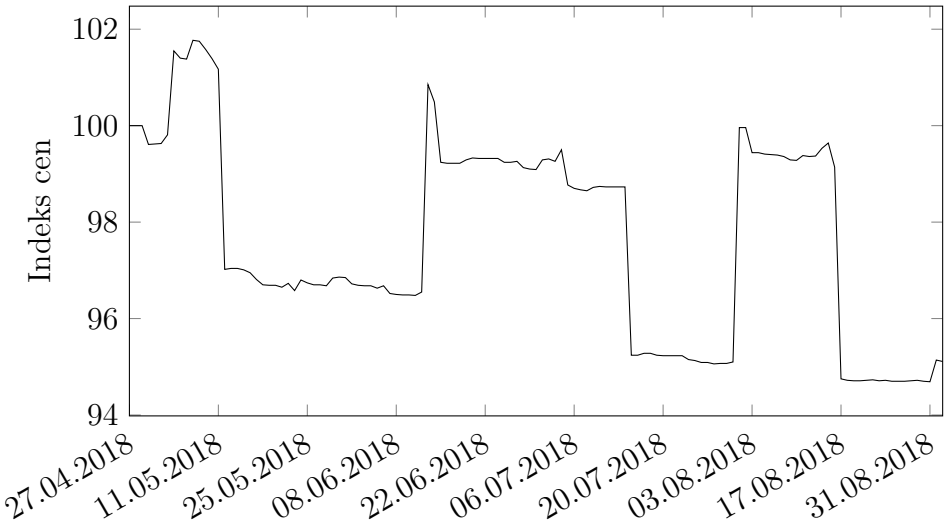
Slika 4.2: Indeks cen za Mercator

Indeks cen za trgovino Spar (graf 4.3) zaradi manjše napake pri branju podatkov zgleda rahlo nazobčan. Končen rezultat je kljub temu pravilen. Videli smo, kako so se cene znižale in potem zvišale nazaj.

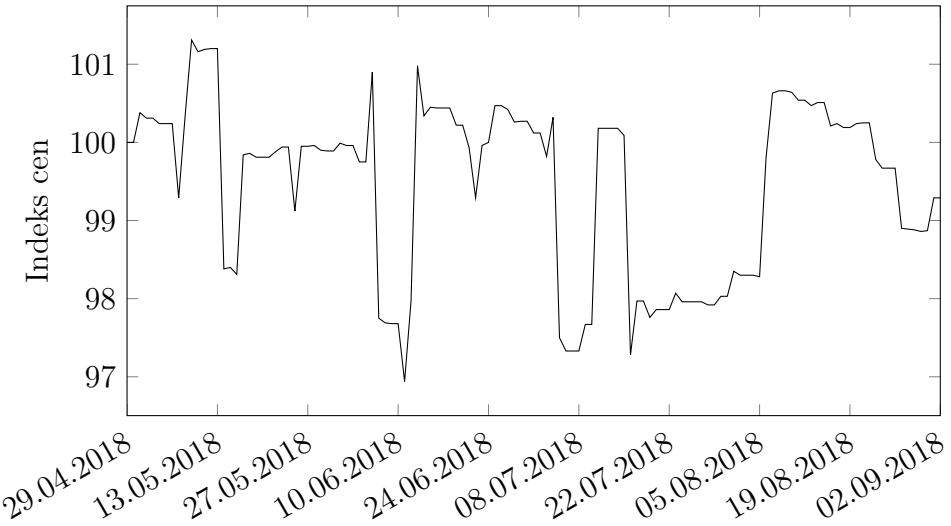


Slika 4.3: Indeks cen za Spar

Pri trgovinah Big Bang in M Tehnika (grafa 4.4 in 4.5) smo opazili zanimiv vzorec, kako večino cen spremenijo v enem dnevu, namesto vsak teden.



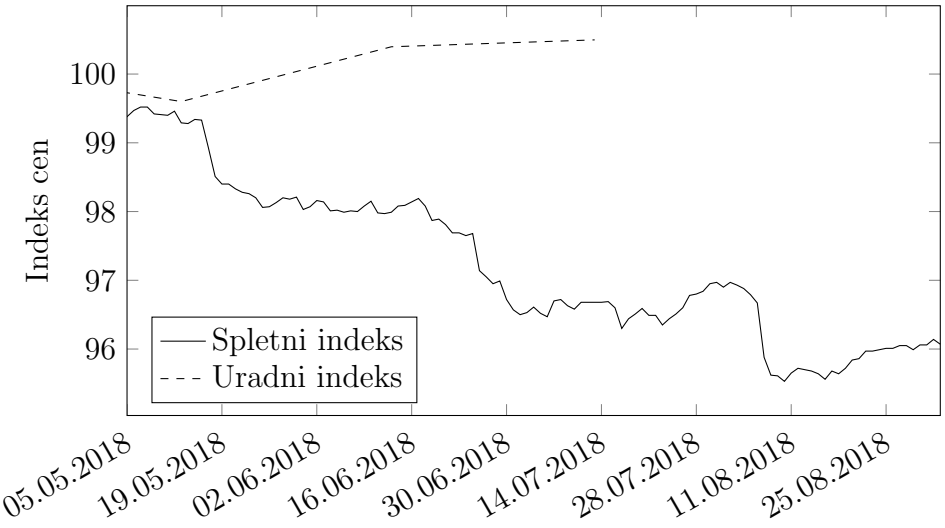
Slika 4.4: Indeks cen za Big Bang



Slika 4.5: Indeks cen za M Tehnika

Cene v trgovini Mimovrste

Cene na spletni trgovini Mimovrste so se spreminjale v zanimivem vzorcu – stalno rahlo navzdol. Približno polovica indeksov posameznih kategorij pri njih je podobna grafu 4.6. Tudi izračunan indeks se ne ujema z uradnim oz. z drugimi trgovinami, ki jim sledimo (niso interno konsistentni). Čisti prihodki od prodaje podjetja so skoraj 45-krat manjši kot prihodki od Spara in Mercatorja skupaj, zato smo se odločili, da te trgovine pri izračunu inflacije preprosto ne bomo upoštevali.



Slika 4.6: Indeks cen za 09.3.4.2 Izdelki za male živali na Mimovrstah

4.2 Delež kategoriziranih dobrin

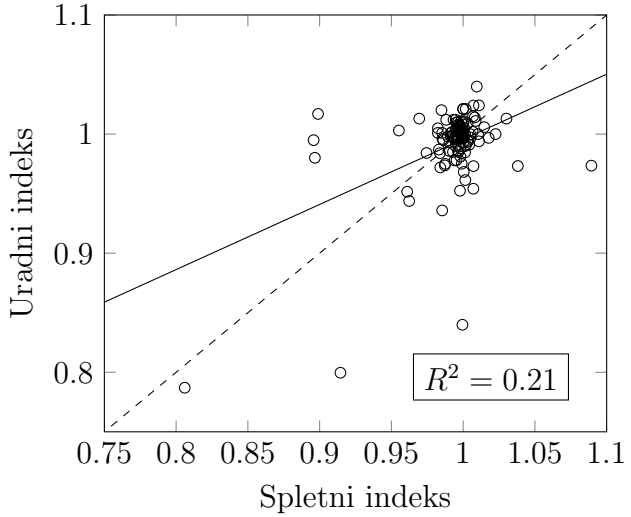
V ECOICOP smo kategorizirali vse večje trgovine (tabela 4.1). Za inflacijo smo upoštevali tudi cene goriv, vode in elektrike. Skupaj smo v indeksu zajeli 44,4 % končne potrošnje. Manjka nam večino storitvenih dejavnosti, za katere je težko dobiti cene preko spleta. Za trgovini Mercator in M Tehnika ni bilo mogoče dobiti podatkov o zalogi. Tam smo konec prodaje upoštevali takrat, ko stran začne vračati napako „HTTP 404 Not Found“.

Trgovina	Nikoli naprodaj	Vsaj 7 dni naprodaj	Kategorizirani in vsaj 7 dni naprodaj
Mercator	-	21.419	75 %
Spar	2.566 (13 %)	16.468	81 %
Mimovrste	91.357 (32 %)	185.830	60 %
Big Bang	1.593 (7 %)	21.573	40 %
M Tehnika	-	5.564	53 %
Lastra	1.852 (13 %)	12.374	82 %

Tabela 4.1: Kategorizirani podatki večjih trgovin

4.3 Primerjava z uradnim indeksom po kategorijah

Vseh 117 kategorij smo primerjali z uradnim indeksom. Gledali smo izmerjeno spremembo v prvih treh mesecih projekta glede na uradno objavljene podatke. Korelacijski koeficient R^2 je 0,21.



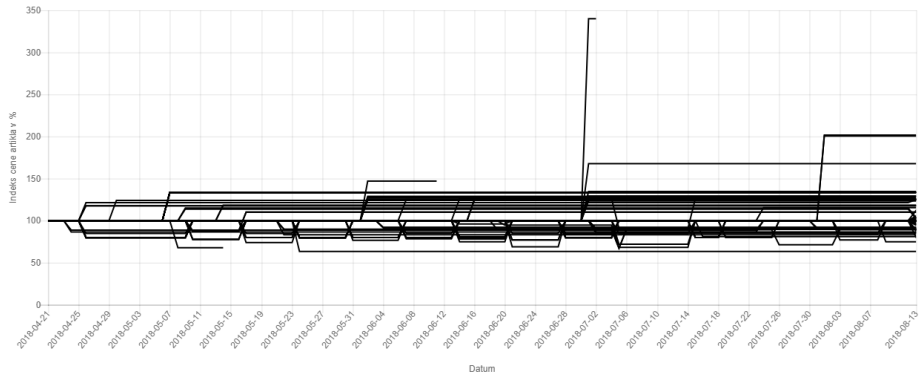
Slika 4.7: Ujemanje spletnega indeksa z uradnim.

Polna črna črta na grafu 4.7 je regresijska premica čez vse kategorije. Črtkana črta je funkcija $y = x$, na kateri bi ležale vse točke, če bi se indeksa popolnoma ujemala.

Spremembo po kategorijah za vsak mesec posebej je težko narediti, ker je težko izbrati začetni in končni datum za primerjavo. Ujemanje indeksov je zelo občutljivo na izbrano obdobje. Uradni indeks ima samo eno številko glede na prejšnji mesec, dnevni indeks jih ima lahko 31[1]. Za našo primerjavo smo preprosto izračunali spremembo našega indeksa od 17.4.2018 (takrat se je projekt začel) do 17.7.2018 glede na uradno spremembo v istih treh mesecih.

4.4 Indeksi posameznih kategorij

Težko je sploh pokazati cene vseh dobrin v eni kategoriji. Slika 4.8 ne prikazuje cen, ampak indeks cen čokolade. Prvi dan imajo vsi izdelki indeks 100, naslednje dni pa vrednost več kot 100 pomeni podražitev, manj kot 100 pa pocenitev.

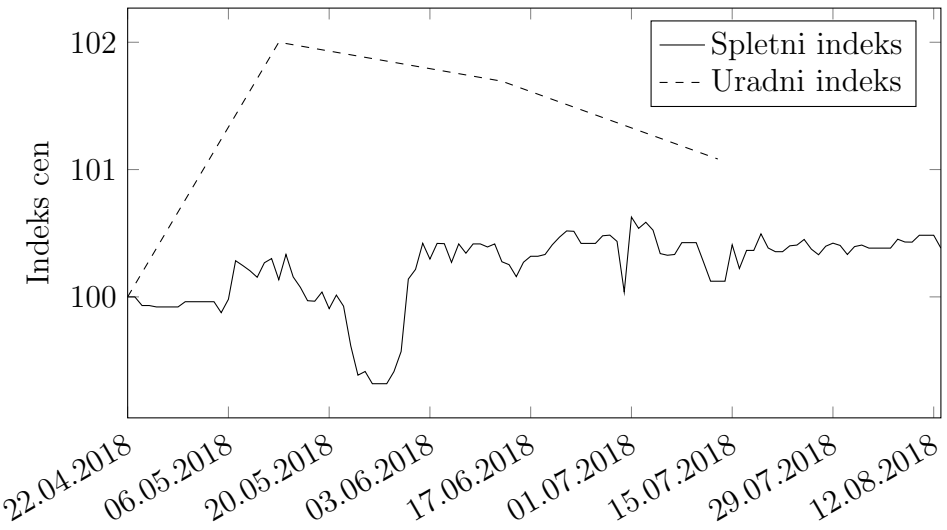


Slika 4.8: Artikli v kategoriji čokolada (n=476).

[1]Lahko si pomagamo z dejstvom, da se uradne cene zbirajo med 1. in 25. v mesecu.

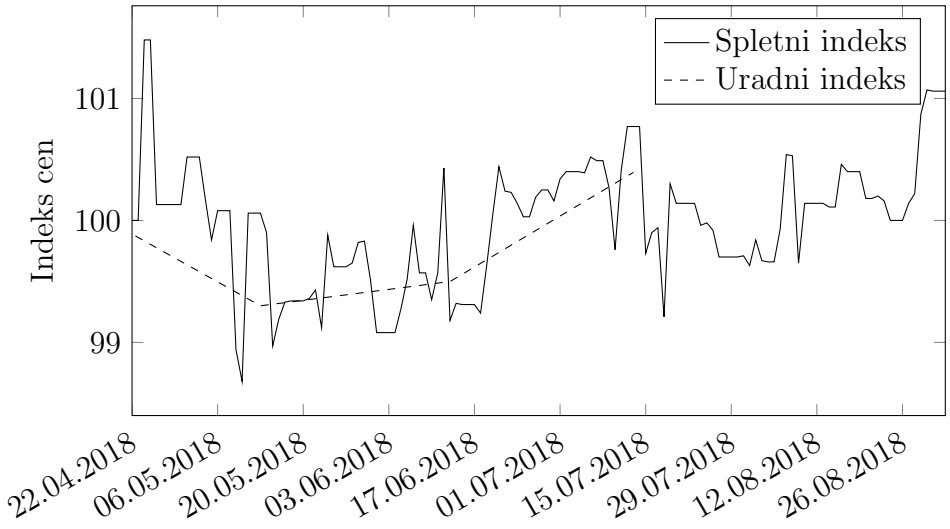
Indeks kategorije je bolj berljiv. Na grafu 4.9 vidimo, kako se je v treh mesecih čokolada podražila za približno 0,4 %. Na sliki 4.8 je en izdelek, ki se je podražil na 350 % začetne cene, potem je pa izginil. V izračunu indeksa je njegova cena upoštevana od začetka prodaje, dokler se ne podraži. Sprememba je nad pragom 200 % in je zato ignorirana.

Kot lahko vidimo na grafih 4.9, 4.10 in 4.11, se spletni indeks včasih ujema z uradnim, včasih pa ne. Nimamo razvite nobene metode za objektivno ocenjevanje ujemanja, na pogled se jih pa ujema približno polovica. Razlog za neujemanje je lahko slaba klasifikacija, vendar smo ročno preverili vse klasificirane dobrine in poskrbeli, da je napačnih klasifikacij minimalno. Bolj verjetno je morda, da spremljamo premalo trgovcev (samo dva večja trgovca na drobno).



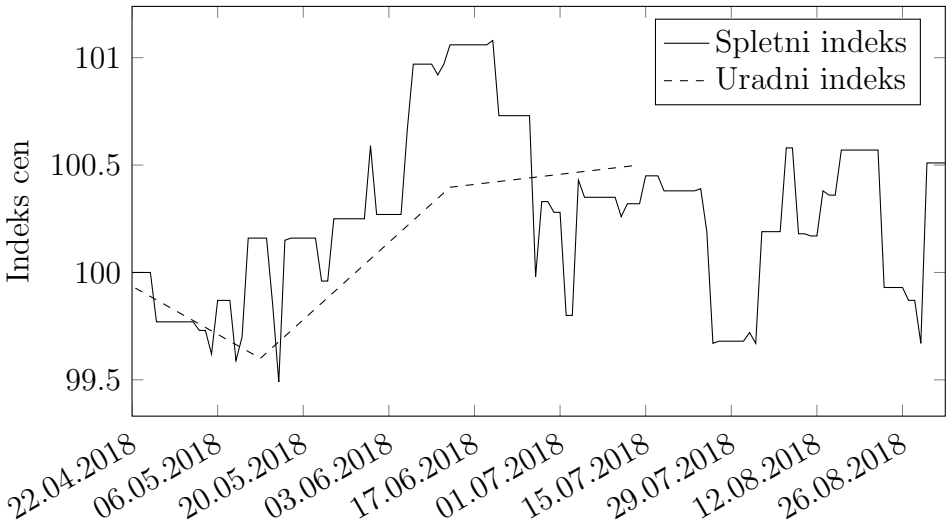
Slika 4.9: Kategorija 01.1.8.3 Čokolada.

Indeksa cen perutnine in izdelkov za male živali (grafa 4.10 in 4.11) se ujemata dosti lepše.



Slika 4.10: Indeks cen za 01.1.2.4 Perutnina

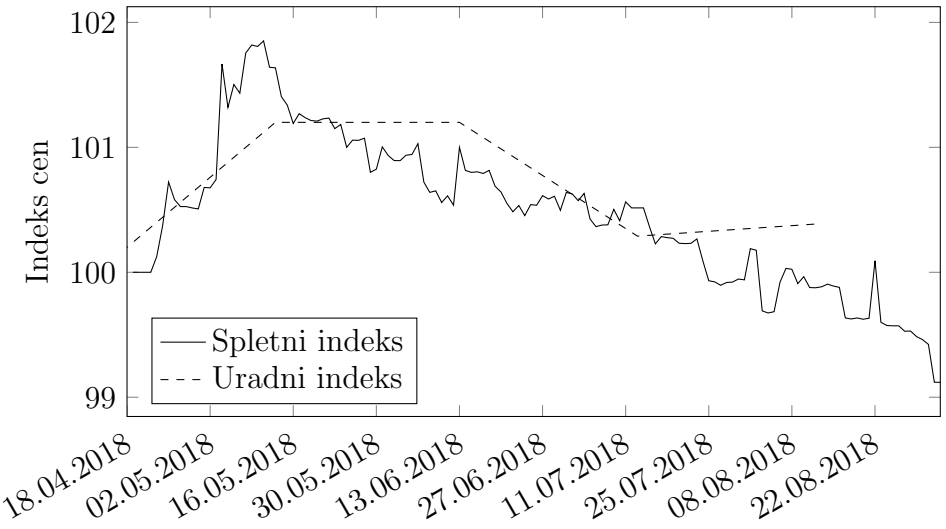
Uradni indeks ne upošteva nobene cene po 25. dnevu v mesecu, zato izgleda, kot da nekaterih visokih cen ne upošteva.



Slika 4.11: Indeks cen za 09.3.4.2 Izdelki za male živali

4.5 Primerjava z uradnim indeksom skupaj

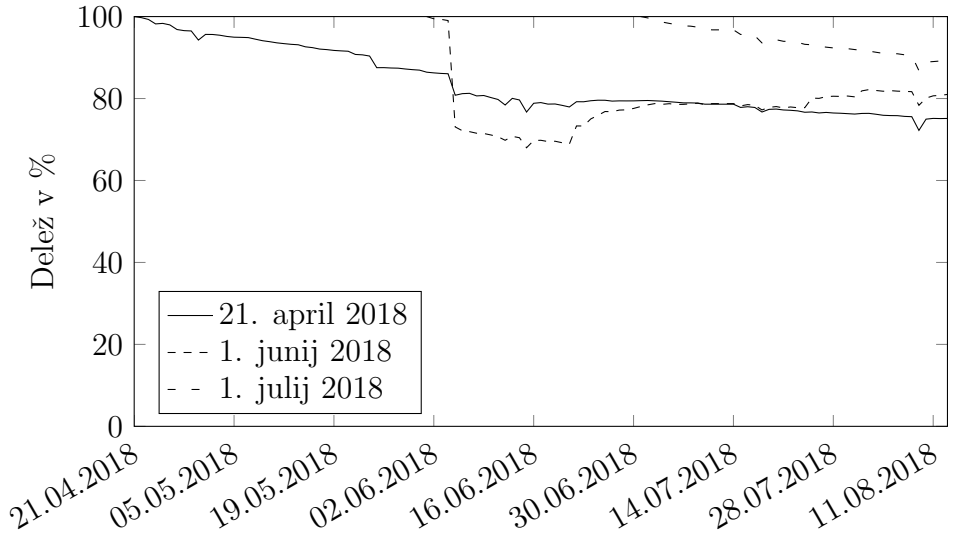
Indeks se v prvih štirih mesecih lepo ujema. Pričakujemo pa lahko, da bo naš indeks konsistentno rahlo podcenil ali precenil inflacijo (angl. drift). The Billion Prices Project jo rahlo podcenjuje glede na ameriški CPI.



Slika 4.12: Primerjava indeksov.

4.6 Delež dobrin, ki jim skripta še sledi

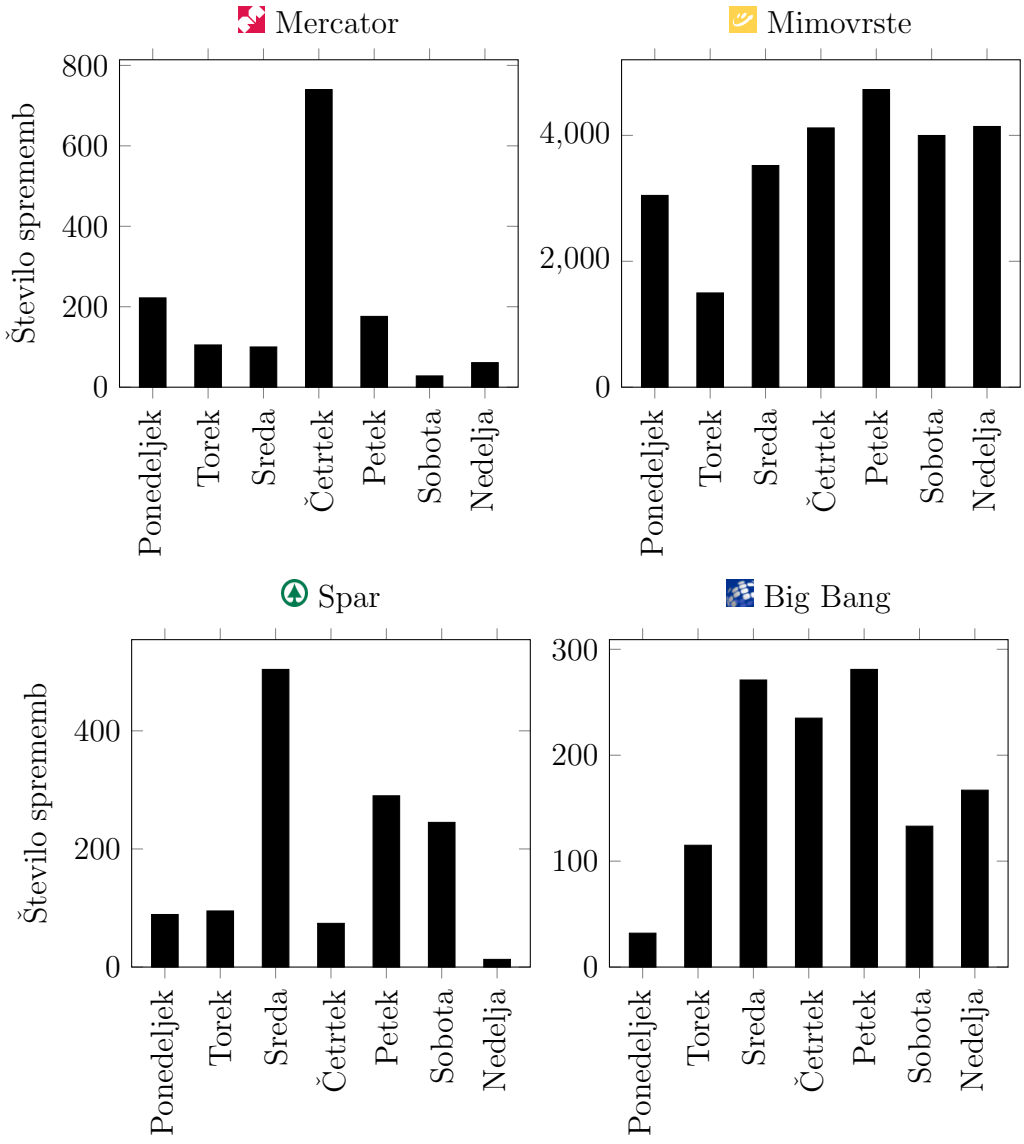
Pomembno je, da v indeksu ohranimo čim več dobrin, da se ne zamenjajo prehitro. Nekaj jih bomo zagotovo izgubili, ker so sezonske oz. se ne prodajajo več. Graf 4.13 prikazuje delež dobrin iz 21. aprila 2018, ki so še upoštevane. Na ta dan smo začeli spremljati cene trgovine Mercator. V prvi polovici vidimo en manjši padec, v drugi pa zgleda delež dokaj stabilen pri 75 %. Prvi dan je skripta sledila 52.125 dobrinam, ki so kategorizirane v ECOICOP in so bile na zalogi.



Slika 4.13: Delež dobrin, ki jim skripta še sledi.

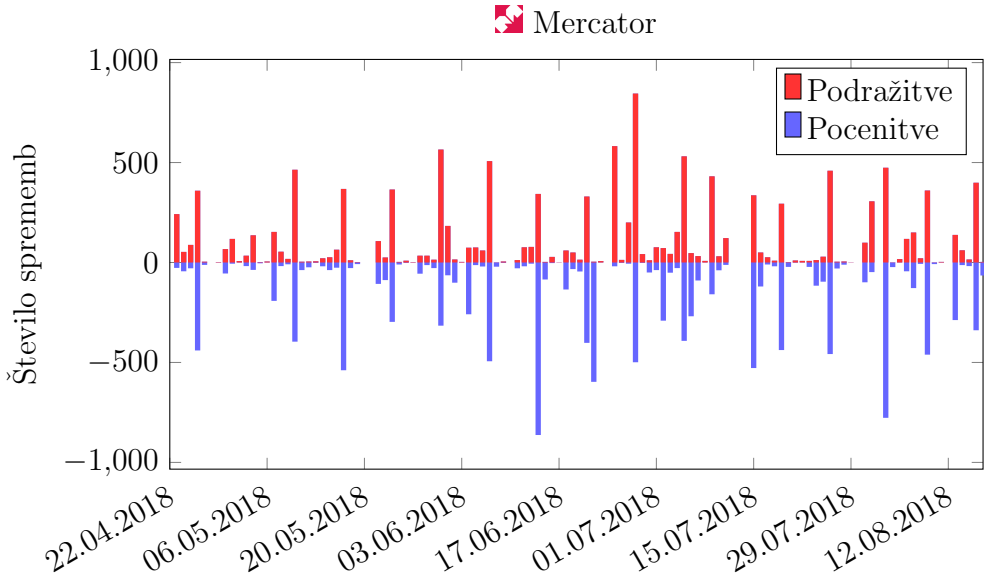
4.7 Spremembe cen po dnevih

Za trgovini Mercator in Spar smo opazili vzorec, da večno cen spremenijo na določen dan v tednu (graf 4.14). Graf za M Tehniko je zelo podoben temu za Mercator. Grafi 4.15, 4.17, 4.16 in 4.18 prikazujejo število sprememb po dnevih. Pri Big Bangu vidimo, da cene spreminjajo bolj poredko, ampak ko jih spremenijo, jih spremenijo veliko.

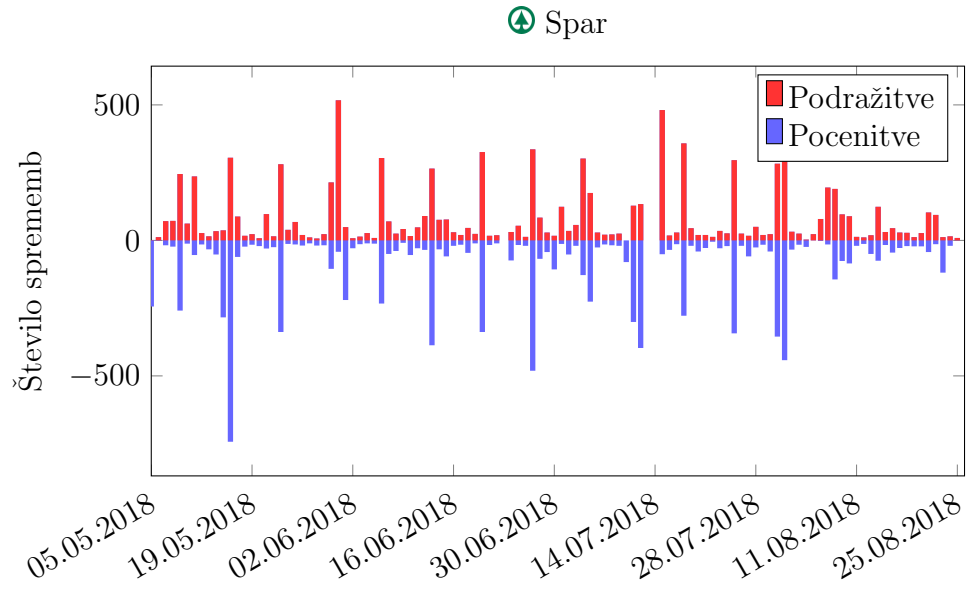


Slika 4.14: Povprečno število sprememb cen na dan.

Trgovca na drobno (grafa 4.15 in 4.16) prikazujeta konsistenten vzorec spreminjanja cen na določen dan v tednu.

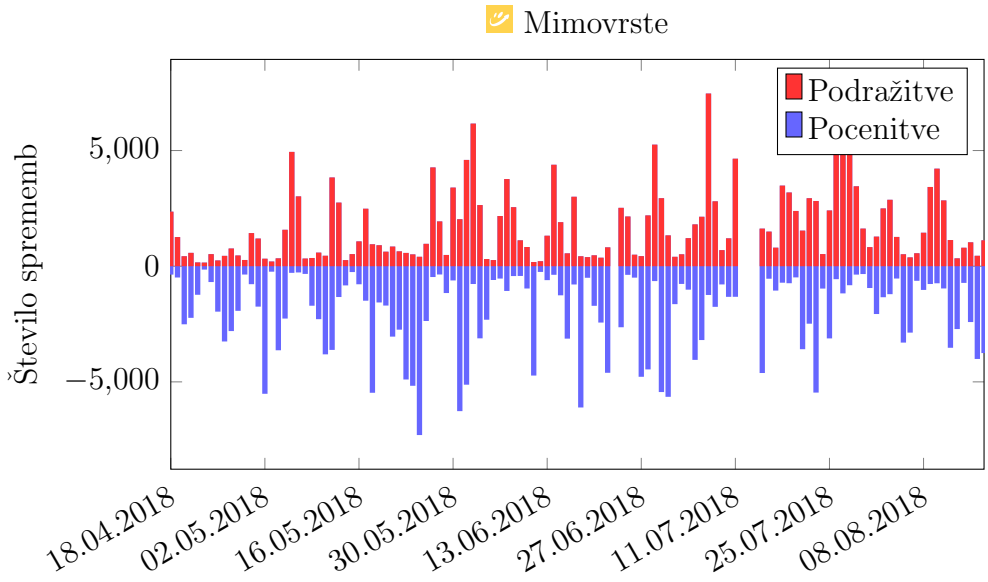


Slika 4.15: Število sprememb cen na dan za Mercator.

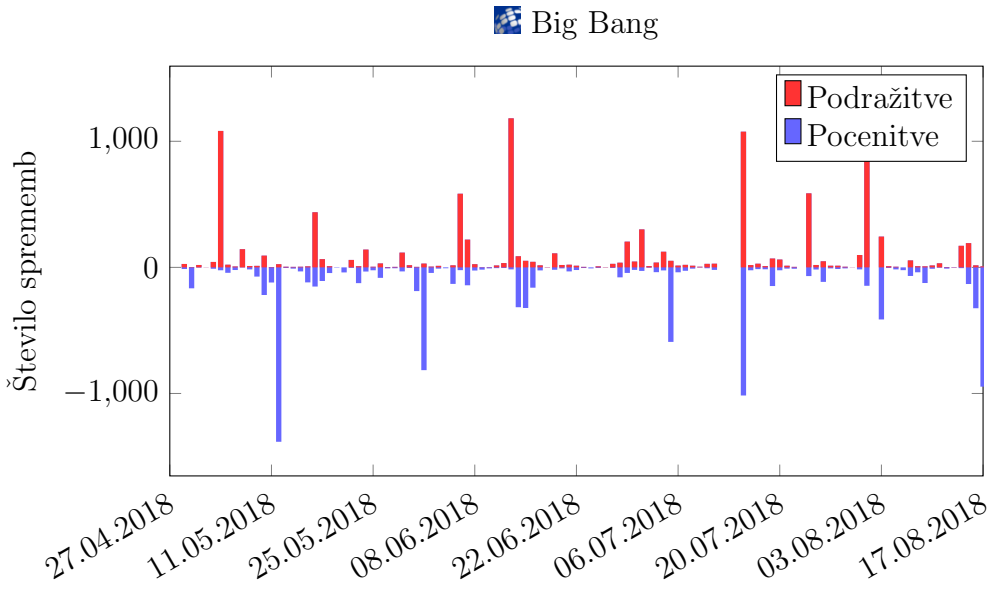


Slika 4.16: Število sprememb cen na dan za Spar.

Zaradi tehničnih težav skripta od 12.4.2018 do 14.4.2018 ni brala cen. Posledica so manjkajoči dnevi v grafih.



Slika 4.17: Število sprememb cen na dan za Mimovrste.



Slika 4.18: Število sprememb cen na dan za Big Bang.

4.8 Povprečno trajanje cene

Ena od ključnih predpostavk v novi keynesianski ekonomski teoriji, ki pojasni poslovni cikel in recesije je, da so cene lepljive oz. navzdol rigidne. To pomeni, da se pomikajo počasneje navzdol kot navzgor. Predhodno se je ta hipoteza testirala na podatkih, ki jih statistični uradi zbirajo za inflacijo. Njihova slabost je, da izhajajo le mesečno, spremljajo pa le nekaj deset tisoč cen. S podatki, zbranimi preko interneta, lahko to hipotezo testiramo dosti bolj natančno[15, 1, 4].

Iz podatkov najprej izračunamo delež cen, ki so se tisti dan spremenile. Upoštevamo samo dobrine, ki so bile na opazovani dan na zalogi in naprodaj. Za maloprodajne trgovine z živili smo izračunali povprečno trajanje cene (angl. price spell) med 56 in 91 dni, kar je bistveno daljše kot rezultati obstoječih raziskav iz Evrope, ZDA in držav Južne Amerike[15, 1, 4]. Opazili smo bistvene razlike med posameznimi ECOICOP kategorijami (graf 4.19).

Trgovina	Delež spremememjenih cen na dan	Implicitno povprečno trajanje cene
Mimovrste	3,1 %	32 dni
Spar	1,1 %	91 dni
Mercator	1,0 %	100 dni
M Tehnika	1,0 %	100 dni
Big Bang	0,8 %	125 dni
Lastra	0,06 %	>4 leta

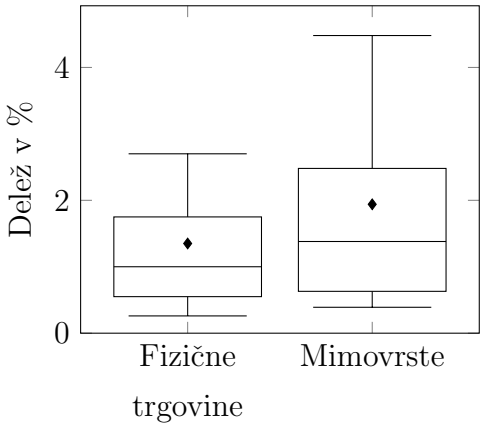
Tabela 4.2: Povprečno trajanje cene.

Ugotovili smo, da trgovine večino cen v času opazovanja niso spremenile (tabela 4.3).

Trgovina	Število dni opazovanja	Delež spremenjenih cen
Lastra	92	5 %
M Tehnika	115	42 %
Mimovrste	125	38 %
Mercator	122	41 %
Big Bang	117	40 %
Spar	109	44 %

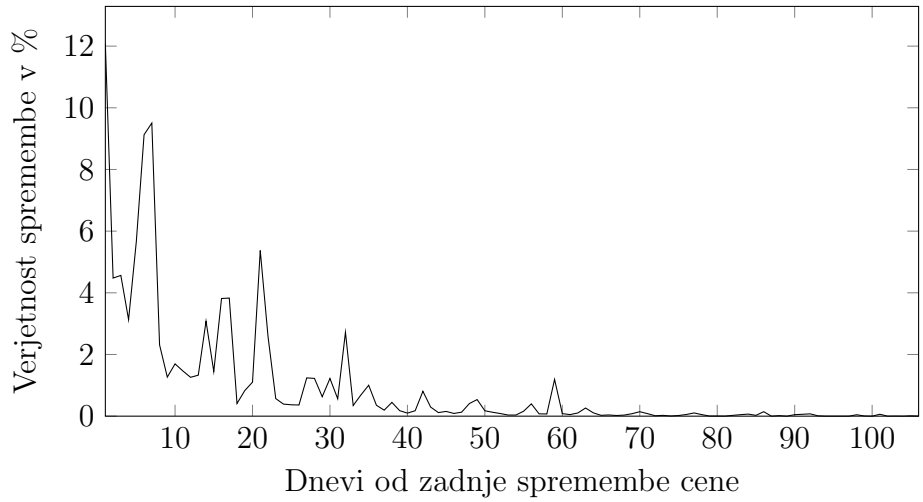
Tabela 4.3: Delež cen, ki jih je trgovina spremenila vsaj enkrat.

Za Mimovrste smo opazili vzorec hitrega menjavanja dobrin, ki so naprodaj (graf 4.21). Polovico stvari je pri njih naprodaj manj kot 76 dni.

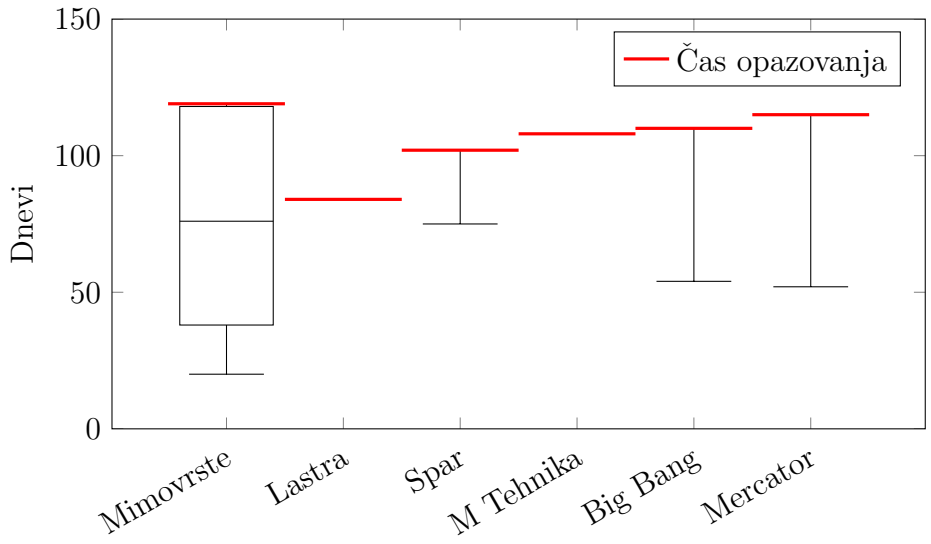


Slika 4.19: Povprečen delež spremenjenih cen na dan po ECOICOP kategorijah.

Graf 4.20 prikazuje verjetnost, da se bo cena spremenila glede na čas od zadnje spremembe (angl. hazard function). Če se je cena spremenila 7 dni nazaj ali manj, ima največjo verjetnost (20,2 %) ponovne spremembe.



Slika 4.20: Verjetnost spremembe cene glede na čas od zadnje spremembe (angl. hazard function)

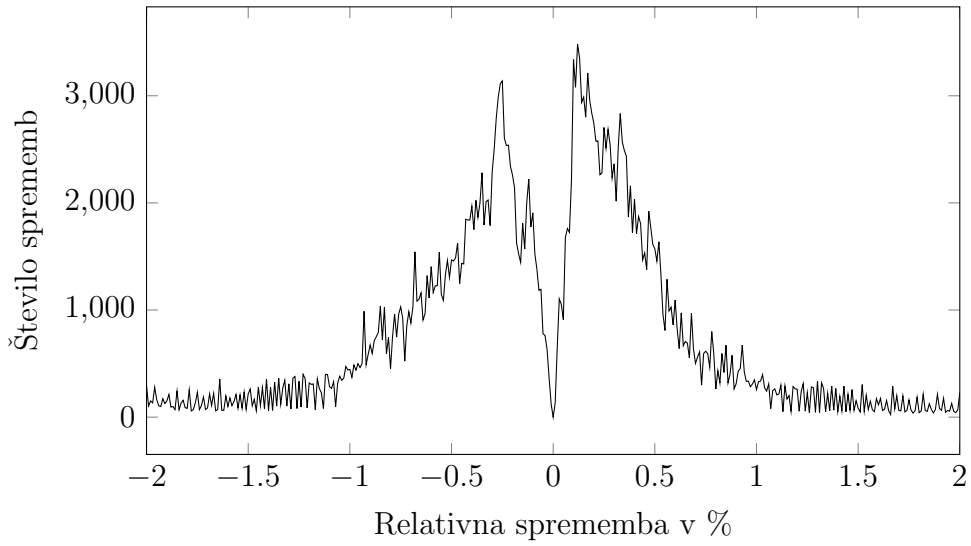


Slika 4.21: Povprečno število dni, ko je dobrina naprodaj.

4.9 Porazdelitev relativnih sprememb cen

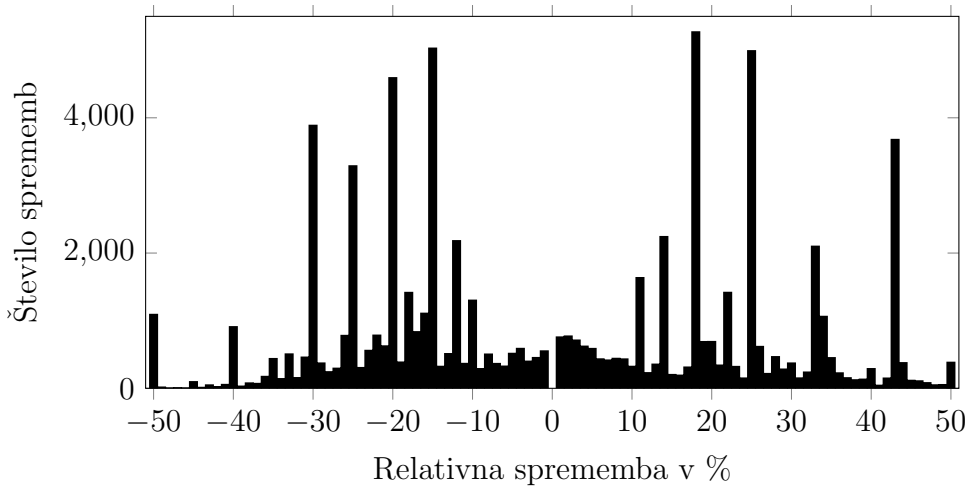
Sprememb cen med -70 % in +100 % je bilo zabeleženih 498.182. Od tega je 260.887 pocenitev in 237.295 podražitev. Delež podražitev je torej 47,63 %.

Večino sprememb je bilo zabeleženih na edini spletni trgovini, Mimovrste, kar lahko zamegli dogajanje v ostalih trgovinah (graf 4.22). Ker nimajo fizične trgovine, kjer bi bilo potrebno ročno menjati nalepke s cenami, jim je mnogo lažje vsak dan spreminjati cenik (angl. menu cost). Če odstranimo vse njihove podatke in spet upoštevamo samo spremembe med -70 % in +100 %, je pocenitev 38.237, podražitev pa 39.071. Enostranski z-test nam pokaže statistično značilno več podražitev ($p = 0.0014$).

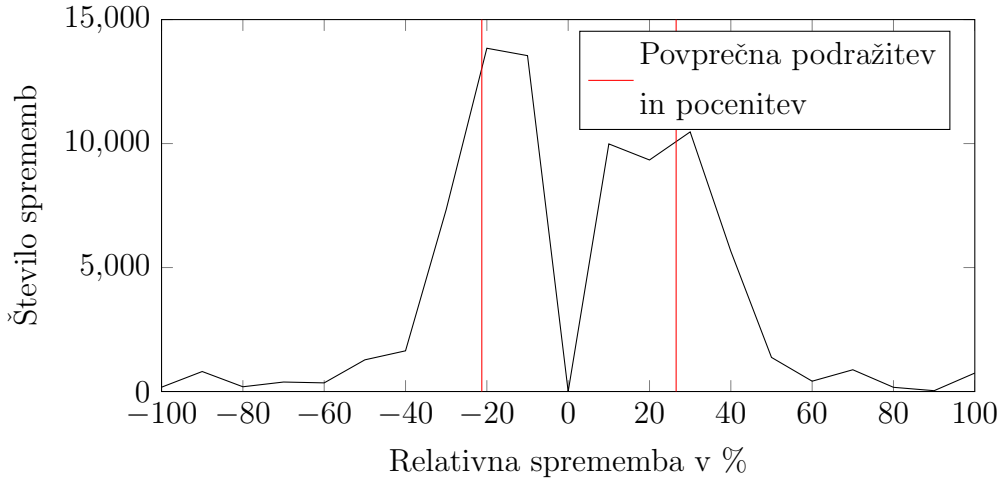


Slika 4.22: Relativne spremembe cen za vse trgovine (med -2 % in +2 %).

Graf 4.23 prikazuje porazdelitev sprememb v večjih trgovinah. Povprečne podražitve in pocenitve so velike (-21,2 % in 26,5 %), vendar približno enake, kot v ZDA (-22 % in 29 %)[4]. Graf 4.24 ima spremembe grupirane v intervalih po 10.



Slika 4.23: Relativne spremembe cen za Mercator, M Tehnika, Spar in Big Bang (n=79.357).



Slika 4.24: Relativne spremembe cen v intervalih po 10.

Poglavje 5

Zaključek

Statistika spletnih cen je dokaj novo področje, saj je mlajše kot deset let. Ponuja še mnogo priložnosti za raziskovanje. Tudi v našem primeru je možna še vrsta izboljšav. Spremljali bi lahko še več trgovcev in zajeli večji delež potrošnje. Kljub temu pa menimo, da je imel projekt uspešen začetek in ima obetavno prihodnost. Pričakujemo lahko tudi, da bo vedno več trgovcev svoje cenike objavilo na spletu.

V nalogi smo predstavili dve večji ugotovitvi: (a) trgovci na drobno v Sloveniji spremenijo manjši delež cen vsak dan, kot trgovci v drugih državah in (b) povprečna sprememba cene je približno enaka kot v ZDA. Kot pričakovano smo pokazali tudi, da spletne trgovine cene spreminjajo bolj pogosto, kot fizične.

Zanimiva ideja za nadaljnje delo je povezovanje istih artiklov pri različnih trgovcih. Tako bi lahko ugotovili, ali se njihove cene spreminjajo hkrati oz. izmerili, kako velika je zakasnitev. Iz shranjenih tehničnih podrobnosti pa nameravamo ugotovili pomembnost posameznih parametrov in računati hedonične popravke.

Prednost branja spletnih cenikov je zelo malo potrebnega sprotnega dela, saj je branje cen in izračunavanje indeksa v celoti avtomatizirano. Projekt bo tekel še naprej, vsi indeksi cen bodo pa javno objavljeni na naši spletni strani.

Literatura

[1] Luis J Alvarez, Emmanuel Dhyne, Marco Hoeberichts, Claudia Kwapil, Hervé Le Bihan, Patrick Lünnemann, Fernando Martins, Roberto Sabbatini, Harald Stahl, Philip Vermeulen, et al. Sticky prices in the euro area: a summary of new micro-evidence. *Journal of the European Economic association*, 4(2-3):575–584, 2006.

[2] Ingolf Boettcher. Automatic data collection on the internet (web scraping). *Eurostat*, Maj 2015.

[3] Robert Cage, John Greenlees, and Patrick Jackman. Introducing the chained consumer price index. In *International Working Group on Price Indices (Ottawa Group): Proceedings of the Seventh Meeting*, pages 213–246. Paris: INSEE, 2003.

[4] Alberto Cavallo. Scraped data and sticky prices. Working Paper 21490, National Bureau of Economic Research, August 2015.

[5] Alberto Cavallo and Roberto Rigobon. The billion prices project: Using online prices for measurement and research. *Journal of Economic Perspectives*, 30(2):151–78, Spring 2016.

[6] Frequently asked questions about hedonic quality adjustment in the cpi. Dosegljivo: `https://www.bls.gov/cpi/quality-adjustment/questions-and-answers.htm`, 2018. [Dostopano: 25. 5. 2018].

[7] The definition of price stability. Dosegljivo: https://www.ecb.europa.eu/mopo/strategy/pricestab/html/index.en.html, 2018. [Dostopano: 15. 8. 2018].

[8] Vlasta Kohek Ema Mišić. Metodološko pojasnilo - indeksi cen na drobno. *Statistični urad Republike Slovenije*, 2005.

[9] *European Price Statistics - An overview.* Eurostat, 2008.

[10] Glossary:fisher price index. Dosegljivo: http://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Fisher_price_index, 2014. [Dostopano: 25. 5. 2018].

[11] *Handbook on price and volume measures in national accounts.* Eurostat, 2001.

[12] *HICP Methodological Manual.* Eurostat, 2017.

[13] Hicp methodology. Dosegljivo: http://ec.europa.eu/eurostat/statistics-explained/index.php?title=HICP_methodology, 2018. [Dostopano: 25. 5. 2018].

[14] Mojca Maček Kenk. Spremembe v izračunu inflacija z letom 2018. *Statistični urad Republike Slovenije*, 2018.

[15] Patrick Lunnemann and Ladislav Wintr. Are internet prices sticky? *ECB*, 2006.

[16] Tina Vratanar Mojca Zlobec. Metodološko pojasnilo - indeksi cen življenjskih potrebščin in povprečne drobnoprodajne cene. *Statistični urad Republike Slovenije*, 2018.

[17] Mysql varchar index length - stack overflow. Dosegljivo: https://stackoverflow.com/a/16474039/971972, 2018. [Dostopano: 12. 5. 2018].

[18] Pricestats. Dosegljivo: https://www.pricestats.com/, 2017. [Dostopano: 25. 5. 2018].

[19] Statistical functions in mysql - code is poetry. Dosegljivo: https://www.xarg.org/2012/07/statistical-functions-in-mysql/, 2012. [Dostopano: 29. 5. 2018].

[20] *Državna statistika v letu 2017. Poročilo o izvajanju Letnega programa statističnih raziskovanj za 2017.* Statistični urad Republike Slovenije, 2018.

[21] Varchar - mariadb knowledge base. Dosegljivo: https://mariadb.com/kb/en/library/varchar/, 2018. [Dostopano: 25. 5. 2018].

[22] James Wells. Review of hedonic quality adjustment in uk consumer price statistics and internationally. *Office for National Statistics*, 2014.

[23] James Wells. Practical guide for processing supermarket scanner data. *Eurostat*, 2017.