Čiščenje podatkov

Datoteke iz statističnega urada

- http://www.stat.si/
- Zavihek `Podatki`, `Izobraževanje`, kliknemo link `Izobraževanje` na podnaslovu nižje na strani `Več tabel v podatkovni bazi SI-STAT`
- Izberemo `Terciarno izobraževanje`, `Vpisani v visokošolsko izobraževanje`
- Izberemo kako od tabel tam. Mi bomo uporabili eno staro CSV datoteko.
- Opomba: ta tabela mogoče ni več dosegljiva preko direktnega brskanja.
- Nastavimo filtre, način prikaza tabele (izvoza) kot `Prikaz tabele na zaslonu`, za vsak tip podatka izberemo vse možnosti, razen tistih, ki so sumarne (v imenu je `SKUPAJ`)
- Gumb `Izpis podatkov`
- Podatki se prikažejo kot tabela na zaslonu

Tabele s hierarhičnimi indeksi

- Privzet prikaz tabele je s hieararhičnimi indeksi

Študenti visokošolskega študija na univerzah in samostojnih visokošolskih zavodih po: VISOKOŠOLSKI ZA SPOL				2004		2005		2006		2007		2008		
				Moški	Ženske	Moški	Ženske	Moški	Ženske	Moški	Ženske	Moški	Ženske	M
Samostojni visokošolski zavodi	Visokošolsko strokovno (prejšnje)	1. letnik	Redni	122	152	141	218	95	163	-	1	6	5	
			Izredni	191	231	218	323	147	259	-	-	-	7	
		2. letnik	Redni	130	101	148	166	80	176	23	43	-	4	
			Izredni	206	205	374	547	236	502	174	321	5	4	
		3. letnik	Redni	67	109	100	80	81	119	51	135	22	45	
			Izredni	198	351	285	470	301	588	300	607	170	352	
		4. letnik	Redni	-	-	4	7	2	3	-	-	-	6	
			Izredni	5	8	1	-	-	1	-	-	1	3	

- S hierarhičnimi indeksi je težko delati
- R nima dobre podpore (npr. Python Pandas ima)

“Navadne tabele”

- Bolj smiselno bi bilo imeti podatke v obliki tabele s stolpci:
  - VISOKOŠOLSKI ZAVOD
  - VRSTA IZOBRAŽEVANJA
  - LETNIK
  - NAČIN ŠTUDIJA
  - SPOL
  - ŠTUDIJSKO LETO
  - ST_STUDENTOV
- Na takih tabelah obstaja t.i. “Relacijska algebra”, ki je osnova za delo s tabelami v podatkovnih bazah
- Dobra podpora teorije in funkcij za izvajanje operacij

Obdelava in pridobivanje podatkov

- Ročno “zavrtimo” tabelo
- Izvozimo v obliki “Razmejena datoteka z glavo .csv”
- Ogledamo si format CSV v tekstovni datoteki
- Ogledamo si še datoteko v Excelu
- Prve 4 vrstice so nepomembne
- Tabela nima glave
- Vsebina v prvih nekaj stolpcih je še vedno podana hierarhično
- Nimamo glave tabele
- Prazna polja so označena kot “-”
- Imena stolpcev niso definirana

Branje tabele

- Uporabili bom paket `readr`
- RStudio CheatSheets - https://www.rstudio.com/resources/cheatsheets/
- Pričakuje se, da na koncu predavanj obvladate cheatsheets-e Data Import, Data Transformation, RMarkdown, RStudio, Shiny, Data Visualization (skoraj vse :)
- Poskusimo naivno prebrati CSV datoteko

```
uvoz <- read_csv2("0955201ss.csv")
```

- Problem: kodna tabela

```
uvoz <- read_csv2("0955201ss.csv",
                  locale=locale(encoding="cp1250"))
```

Branje tabele

- Problem? Pogledamo:

```
problems(uvoz)
```

- Vizualno pregledamo:

```
View(uvoz)
```

Branje tabele

- V uvozu imamo samo en stolpec!
- Definirajmo stolpce

```
stolpci <- c("VISOKOSOLSKI_ZAVOD", "VRSTA_IZOBRAZEVANJA",
"LETNIK", "NACIN_STUDIJA", "SPOL" ,
"STUDIJSKO_LETO", "ST_STUDENTOV")
```

```
uvoz <- read_csv2("0955201ss.csv",
                  locale=locale(encoding="cp1250"),
                  col_names=stolpci)
```

```
problems(uvoz)
```

```
View(uvoz)
```

Branje tabele

- Izpustiti moramo prve 4 vrstice

```
uvoz <- read_csv2("0955201ss.csv",
                  locale=locale(encoding="cp1250"),
                  col_names=stolpci,
                  skip=4)
```

- Zadnja prebrana vrstica (po branju), ki nas še zanima je 7162

```
uvoz <- read_csv2("0955201ss.csv",
                  locale=locale(encoding="cp1250"),
                  col_names=stolpci,
                  skip=4,
                  n_max=7162)
```

Branje tabele

- Znak "-" pri številu študentov bi radi interpretirali kot `NA`

```
uvoz <- read_csv2("0955201ss.csv",
                  locale=locale(encoding="cp1250"),
                  col_names=stolpci,
                  skip=4,
                  n_max=7162,
                  na=c("", " ", "-"))
```

Obdelava podatkov

- Zaradi 'hierarhičnega uvoza' bi radi, da se vsi dimenzijski stolpci ponavljajo v vrsticah do naslednjega vnosa
- Izjema je stolpec `STUDIJSKO_LETO`

```
podatki <- uvoz %>% fill(1:5) %>% drop_na(STUDIJSKO_LETO)
```

- PAZI! Operator %>% iz knjižnice `dplyr` pomeni: Rezultat izraza pred operatorjem uporabi kot prvi argument naslednje funkcije
- Princip kodiranja s pomočjo operatorja %>% imenujemo *veriženje* (ang. chaining)

Obdelava podatkov

- Ko se navadimo postane koda zelo berljiva (začnemo s tem, potem najprej naredimo to, potem to, potem to . . .)

```
podatki <- uvoz %>%
           fill(1:5) %>%
           drop_na(STUDIJSKO_LETO)
```

- Alternativa je gnezdenje klicev funkcij -> NEPREGLEDNO

```
podatki <- drop_na(fill(uvoz, 1:5), STUDIJSKO_LETO))
```

Obdelava podatkov

- Preverimo tipe stolpcev

```
sapply(zdruzena, class)
```

- Tip stolpca `ST_STUDENTOV` bi moral biti `integer`.
- Uporabimo funkcije iz paketa `readr` za pretvorbo stolpcev iz nizov

```
zdruzena$ST_STUDENTOV <-
  parse_integer(zdruzena$ST_STUDENTOV)
```

Relacijski model

- Podatke organiziramo kot množico večih tabel (`data.frame`-ov)
- Model je v osnovi star več kot 35 let
- Uporablja se v večini večjih poslovnih sistemov, relacijskih podatkovnih bazah
- Enostaven za razumevanje, pregleden
- Omogoča zmogljive poizvedbe v standardiziranem jeziku SQL (na podatkovnih bazah) in podobno zmogljive poizvedbe/operacije v R
- Podpira učinkovite implementacije (pri podatkovnih bazah in v R)

Relacijska algebra

- Podatke organiziramo kot množico večih tabel (`data.frame`-ov)
- *Relacija* = tabela
- *Relacijska algebra* je matematični opis operacij nad relacijami (tabelami).
- Operatorji so operacije, ki sprejmejo relacije (tabele) in vrnejo (nove) relacije (tabele).
- *Shema relacije* = definicija tabele (imena + tipi).

Operatorji relacijske algebre

- Operatorji so odvisni od shem relacij nad katerimi jih izvajamo.
- $\sigma_p(R)$ - izberi vrstice v relaciji R, ki ustrezajo pogoju p.
  - Pogoj je lahko logični izraz.
  - Shema vrnjene tabele je ista.
  - Primer: operator `[pogoj,]` v kombinaciji z logičnim indeksom
- $\pi_{a_1,a_2,\ldots,a_n}(R)$ - izberi stolpce z imeni $a_1, a_2, \ldots, a_n$ relacije R in vrni novo tabelo s shemo, ki jo določajo definicije teh stolpcev.
  - Število vrstic ostane nespremenjeno.
  - Primer: operator `[vektor_imen]`
- $\rho_{a/b}(R)$ - spremeni ime stolpcu a v b. Vrni enako tabelo (glede vrstic), le z drugo shemo.
  - Primer: preimenovanje stolpca preko `names(df)[[ime]] <- novo_ime`

Operatorji relacijske algebre

- $R \cup S$ - vrni relacijo z unijo vrstic, če imata relaciji R in S enaki shemi.
- $R \setminus S$ - vrni relacijo z razliko vrstic, če imata relaciji R in S enaki shemi.
- $R \times S$ - vrni kartezični produkt relacij (vsaka vrstica R z vsako vrstico S).
  - Shema rezultata sta združeni shemi.
  - Ni tako uporabna operacija, so pa uporabne ustrezne učinkovite izvedbe s filtriranjem (združitve)

Operacija JOIN

$$R \bowtie S = \pi_{shema(R) \cup shema(S)}(\sigma_{R.a_1 = S.a_1 \wedge R.a_2 = R.a_2 \wedge \ldots}(R \times S))$$

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Manager
Finance	George
Sales	Harriet
Production	Charles

Employee* ⋈ *Dept

Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet

Vir: Wikipedia.

Paket *dplyr*

- Podpira operacije iz relacijske algebre
- Učinkovita implementacija
- Alternativa: uporaba paketa `data.table`
- https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html
- https://cran.r-project.org/web/packages/dplyr/vignettes/two-table.html

Operatorji v *dplyr*

- Osnovni relacijski operatorji:
  - `filter(p)` - $\sigma_p(R)$
  - `select(a_1, a_2, ..., a_n)` - $\pi_{a_1,a_2,\ldots,a_n}(R)$
  - `rename()` - $\rho_{a/b}(R)$
  - `union(x, y)` - $R \cup S$
  - `setdiff(x, y)` - $R \setminus S$
  - `inner_join(x, y)` - združitev po skupnih stolpcih
- Dodatni praktično uporabni operatorji
  - `arrange(...)` - urejanje vrstic glede na izbrane stolpce
  - `mutate(...)` - preimenovanje stolpcev in dodajanje novih, ki so funkcije obstoječih.
  - `distinct()` - ohranjanje enoličnih vrstic
  - `summarize(...)` - uporaba združevalne funkcije na nekem stolpcu
  - `group_by(...)` - združevanje po vrsticah glede na enake vrednosti v stolpcih

Popravljanje stolpca LETNIK

- V stolpcu LETNIK bi želeli krajši zapis
- Poglejmo kaj imamo v stolpcu (histogram)

```
table(podatki$LETNIK)
```

- Obstoječa imena v stolpcu LETNIK

```
> imena <- c("1. letnik", "2. letnik", "3. letnik",
"4. letnik", "5. letnik", "6. letnik", "Absolventi")
```

- Radi bi jih poenostavili imena v:

```
> letniki <- c("1","2","3","4","5","6","Abs")
```

Popravljanje stolpca LETNIK

- Ustvarimo "relacijo" s temi dvemi stolpci

```
> tab2 <- data.frame(letnik=letniki, ime=imena)
> tab2$ime <- as.character(tab2$ime)
```

- Uporabimo operacijo inner_join po stolpcih LETNIK IN ime

```
> require(dplyr)
> zdruzena <- podatki %>% inner_join(tab2, c("LETNIK"="ime"
```

Poizvedbe

- Uporabi operacije iz relacijske algebre pravimo *poizvedba*
- Ime izhaja iz relacijskih podatkovnih baz in jezika SQL, ki izvaja operacije in iz obstoječih tabel (relacij) preko operacij proizvajajo nove tabele (relacije)
- Vrni tabelo z vrsticami, ki pripadajo ženskam

```
> filter(zdruzena, SPOL=="Ženske")
```

- Ekvivalentno opratorju `[pogoj,]`
- Bolj po dplyr-jevsko

```
> zdruzena %>% filter(SPOL=="Ženske")
```

Poizvedbe

- Vse vrstice, v katerih so ženske vpisane po letu 2011

```
> zdruzena %>% filter(SPOL=="Ženske" &
    STUDIJSKO_LETO > 2011)
```

- Namesto operatorja "&" lahko pogoje ločimo z vejico

```
> zdruzena %>% filter(SPOL=="Ženske",
    STUDIJSKO_LETO > 2011)
```

Operacija arrange

- Uredi po stolpcu ST_STUDENTOV

```
> zdruzena %>% arrange(ST_STUDENTOV)
```

- Uredi po stolpcih STUDIJSKO_LETO in potem po stolpcu ST_STUDENTOV, in sicer padajoče

```
> zdruzena %>% arrange(STUDIJSKO_LETO, desc(ST_STUDENTOV))
```

Operaciji `select` in `rename`

- Izberi samo stolpce `STUDIJSKO_LETO`, `ST_STUDENTOV` in `SPOL`

```
> zdruzena %>% select(STUDIJSKO_LETO, ST_STUDENTOV, SPOL)
```

- Ob tem še preimenuj stolpec `STUDIJSKO_LETO` v `LETO`.

```
> zdruzena %>%
    select(LETO=STUDIJSKO_LETO, ST_STUDENTOV, SPOL)
```

- Preimenuj stolpec `STUDIJSKO_LETO` v `LETO`

```
> zdruzena %>% rename(LETO=STUDIJSKO_LETO)
```

Združevanje po vrsticah

- Za katera leta imamo podatke?

```
> zdruzena %>% select(STUDIJSKO_LETO) %>% distinct()
```

- Operator %>% iz paketa `dplyr` nam omogoča "pythonovske" klice kot za metode in s tem veriženje operacij (poveča preglednost kode)
- Koliko študentov je bilo vpisanih vsako leto?

```
> zdruzena %>%
    group_by(STUDIJSKO_LETO) %>%
    summarize(VPIS=sum(ST_STUDENTOV, na.rm=TRUE))
```

Združevanje po vrsticah

- Najprej smo združili vrstice po istih vrednostih v stolpcu `STUDIJSKO_LETO`, potem pa uporabili združevalno funkcijo na nekem od preostalih stolpcev.
- V rezultatu so le smiselni stolpci.
- Združevalne funkcije: `min(x)`, `max(x)`, `mean(x)`, `sum(x)`, `sd(x)`, `median(x)`, `IQR(x)`, `n(x)`, `n_distinct(x)`, `first(x)`, `last(x)` in `nth(x, n)`

Združevanje

- Koliko je bilo vpisanih po spolih za posamezna leta?

```
> zdruzena %>%
    group_by(SPOL, STUDIJSKO_LETO) %>%
    summarize(VPIS=sum(ST_STUDENTOV, na.rm=TRUE))
```

- Koliko žensk in koliko moških je bilo vpisanih na posameznih vrstah študija na univerzi in kakšni so njihovi deleži?

```
> zdruzena %>%
  filter(VISOKOSOLSKI_ZAVOD == "Univerze - SKUPAJ") %>%
  select(VRSTA_IZOBRAZEVANJA, SPOL, ST_STUDENTOV) %>%
  group_by(VRSTA_IZOBRAZEVANJA, SPOL) %>%
  summarize(STEVILO=sum(ST_STUDENTOV, na.rm=TRUE)) %>%
  spread(SPOL, STEVILO) %>%
  arrange(VRSTA_IZOBRAZEVANJA) %>%
  mutate(
    deležMoški=round(Moški/(Moški + Ženske), 2),
    deležŽenske=round(Ženske/(Moški + Ženske), 2)
  )
```

Organizacija podatkov

- Podatke si skušamo organizirati v obliko, ki se ji reče Tidy data
- Paketa `dplyr` in `ggplot2` (za vizualizacijo) sta še posebej prilagojena za to vrstno obliko podatkov
- Taka organizacija podatkov je povezana s t.i. *normalizacijo relacij* pri relacijskih podatkovnih bazah in v relacijski algebri

Normalizacija

- *Normalizacija* je proces v katerem sistematično pregledamo relacije (tabele) in anomalije. Ko identificiramo anomalijo relacijo razbijemo na dve novi.
- Med procesom normalizacije ponavadi dobimo še globji vpogled, kakšna bo interakcija med podatki v različnih tabelah.
- Normalizacija nam pomaga odstraniti redundantnost zapisa podatkov.
- Ampak zato moramo morda delati več `join`-ov.
- Včasih se zaradi učinkovitosti namerno odločimo, da ne izvedemo nekega koraka normalizacije (npr. za namene določenih hitrih analiz).

Funkcijska odvisnost

- *Funkcijska odvisnost* opisuje odnos med stolpci znotraj iste relacije (tabele).
- Stolpec je funkcijsko odvisen od drugega, če lahko s pomočjo vrednosti prvega stolpca v neki vrstici impliciramo vrednost drugega stolpca v isti vrstici.
- Primer: Številka študenta implicira študij študenta.
- Za nakazovanje funkcijske odvisnosti uporabimo simbol `->`
- Stolpec je lahko funkcijsko odvisen od kombinacije večih stolpcev.
- Primer: `Solsko_leto, Predmet -> Predavatelj`

Ključ

- Ključ: eden ali več stolpcev, ki enolično določajo vrstico.
- Izbor ključev temelji na konkretni aplikaciji baze. Kaj je ključ izvemo velikokrat iz konteksta in od uporabnikov.
- Za ključ vedno velja: `Ključ -> vsi ostali stolpci`.
- Obstajajo lahko funkcijske odvisnosti, ki na levi strani nimajo (samo) ključev.
- Kaj z njimi?

Vrste normalizacij

- Vrste:
  - Prva normalizirana oblika (1NF)
  - Druga normalizirana oblika (2NF)
  - Tretja normalizirana oblika (3NF)
  - Boyce-Codd normalizirana (BCNF)
  - Četrta normalizirana oblika (4NF)
  - Peta normalizirana oblika (5NF)
- Vsaka naslednja oblika vsebuje prejšnjo.

Normalizacija

Normalizacija v ustrezno obliko poteka na naslednji način:

- Določimo ključe vsake relacije (tabele).
- Določimo funkcijske odvisnosti.
- Preverimo ali so kršene zahteve ustrezne definicije.
- Če pride do kršitve v neki relaciji, potem to relacijo (tabelo) razdelimo na dve relaciji.
- Ponovno preverimo pogoje za izbrano obliko.
- http://holowczak.com/database-normalization/

1NF = predpostavke za relacijo

- Pogoji:
  - Vsaka vrstica ima za določen stolpec samo eno vrednost.
  - Podatki v stolpcu so istega tipa.
  - Isto ime stolpca se lahko pojavi le enkrat v relaciji.
  - Vrstni red stolpcev ni važen.
  - Nobeni dve vrstici ne smeta biti enaki.
  - Vrstni red vrstic ni važen.

2NF

- Ključ (ang. superkey): katera koli skupina stolpcev, za katere ne obstajata dve vrstici z istima vrednostima v teh stolpcih. Vsi drugi stolpci so funkcijsko odvisni od stolpcev, ki določajo ključ.
- 1NF - vsi stolpci skupaj določajo nek ključ
- Minimalni ključ (ang. candidate key): ključ, za katerega nobena stroga podmnožica ne predstavlja ključa.
- Primarni ključ: izbrani minimalni ključ
- Neključni stolpec: stolpec, ki ni v nobenem minimalnem ključu
- Pogoj za 2NF: 1NF + nobena stroga podmnožica kakega minimalnega ključa ne funkcijsko določa kak neključni stolpec

Employees' Skills

Employee	Skill	Current Work Location
Brown	Light Cleaning	73 Industrial Way
Brown	Typing	73 Industrial Way
Harrison	Light Cleaning	73 Industrial Way
Jones	Shorthand	114 Main Street
Jones	Typing	114 Main Street
Jones	Whittling	114 Main Street

Employees

Employee	Current Work Location
Brown	73 Industrial Way
Harrison	73 Industrial Way
Jones	114 Main Street

Employees' Skills

Employee	Skill
Brown	Light Cleaning
Brown	Typing
Harrison	Light Cleaning
Jones	Shorthand
Jones	Typing
Jones	Whittling

3NF

- Pogoj: relacija je v 2NF in nimamo tranzitivnih funkcionalnih odvisnosti.
- Tranzitivna funkcionalna odvisnost:
  - iz `A -> B`, `B -> C` sledi `A -> C`.

Tournament Winners

Tournament	Year	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Tournament Winners

Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner Dates of Birth

Winner	Date of Birth
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

BCNF

- Pogoj: za vsako funkcionalno odvisnost oblike:
  - `A1, ..., An -> B` velja,
  - da stolpci `A1, ..., An` predstavljajo primarni ključ.

“Statistična” definicija “tidy data”

- Stolpci lahko predstavljajo spremenljivke ali meritve
- Spremenljivke opisujejo parametre pri katerih je izvedene meritev (“dimenzije”)
- Definicija “Tidy data”
  - vsaka spremenljivka tvori stolpec
  - za vsako meritev imamo eno vrstico
  - vsak tip meritve je v ločeni tabeli
- Ekvivalentno: podatki so v 3NF

Najbolj pogosti problemi

- Imena stolpcev so vrednosti, ne pa imena spremenljivk
- V enem stolpcu hranimo več spremenljivk
- Spremenljivke hranimo tako v vrsticah kot v stolpcih
- Več vrst meritev (podatkov) v eni tabeli
- Več istovrstnih meritev v večih tabelah
- Hadley Wickham, Tidy Data, Journal of Statistical Software, August 2014, Volume 59, Issue 10

Imena stolpcev so vrednosti

religion	<$10k	$10–20k	$20–30k	$30–40k	$40–50k	$50–75k
Agnostic	27	34	60	81	76	137
Atheist	12	27	37	52	35	70
Buddhist	27	21	30	34	33	58
Catholic	418	617	732	670	638	1116
Don't know/refused	15	14	15	11	10	35
Evangelical Prot	575	869	1064	982	881	1486
Hindu	1	9	7	9	11	34
Historically Black Prot	228	244	236	238	197	223
Jehovah's Witness	20	27	24	24	21	30
Jewish	19	19	25	25	30	95

row	a	b	c
A	1	4	7
B	2	5	8
C	3	6	9

(a) Raw data

row	column	value
A	a	1
B	a	2
C	a	3
A	b	4
B	b	5
C	b	6
A	c	7
B	c	8
C	c	9

(b) Molten data

Operacija "gather"

- paket `tidyr`

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
airquality %>% gather(MERITEV, VREDNOST, -Month, -Day) %>% head
```

```
##   Month Day MERITEV VREDNOST
## 1     5   1   Ozone        41
## 2     5   2   Ozone        36
## 3     5   3   Ozone        12
## 4     5   4   Ozone        18
## 5     5   5   Ozone        NA
## 6     5   6   Ozone        28
```

Alternativa - operacija "melt"

- paket `reshape2`

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
airquality %>% melt(id.vars = c("Month", "Day"),
                    variable.names=c("MERITEV"),
                    value.name = "VREDNOST")
```

```
##     Month Day variable VREDNOST
## 1       5   1    Ozone     41.0
## 2       5   2    Ozone     36.0
## 3       5   3    Ozone     12.0
## 4       5   4    Ozone     18.0
## 5       5   5    Ozone       NA
```

En stolpec več spremenljivk

country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

(a) Molten data

country	year	sex	age	cases
AD	2000	m	0–14	0
AD	2000	m	15–24	0
AD	2000	m	25–34	1
AD	2000	m	35–44	0
AD	2000	m	45–54	0
AD	2000	m	55–64	0
AD	2000	m	65+	0
AE	2000	m	0–14	2
AE	2000	m	15–24	4
AE	2000	m	25–34	4
AE	2000	m	35–44	6
AE	2000	m	45–54	5
AE	2000	m	55–64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

(b) Tidy data

- Obdelavo nizov si bomo ogledali kasneje (regularni izrazi)

Spremenljivke v vrsticah in stolpcih

id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

(a) Molten data

id	date	tmax	tmin
MX17004	2010-01-30	27.8	14.5
MX17004	2010-02-02	27.3	14.4
MX17004	2010-02-03	24.1	14.4
MX17004	2010-02-11	29.7	13.4
MX17004	2010-02-23	29.9	10.7
MX17004	2010-03-05	32.1	14.2
MX17004	2010-03-10	34.5	16.8
MX17004	2010-03-16	31.1	17.6
MX17004	2010-04-27	36.3	16.7
MX17004	2010-05-27	33.2	18.2

(b) Tidy data

Več vrst meritev v eni tabeli

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

- Normalizacija

id	artist	track	time
1	2 Pac	Baby Don't Cry	4:22
2	2Ge+her	The Hardest Part Of ...	3:15
3	3 Doors Down	Kryptonite	3:53
4	3 Doors Down	Loser	4:24
5	504 Boyz	Wobble Wobble	3:35
6	98^0	Give Me Just One Nig...	3:24
7	A*Teens	Dancing Queen	3:44
8	Aaliyah	I Don't Wanna	4:15
9	Aaliyah	Try Again	4:03
10	Adams, Yolanda	Open My Heart	5:30
11	Adkins, Trace	More	3:05
12	Aguilera, Christina	Come On Over Baby	3:38
13	Aguilera, Christina	I Turn To You	4:00
14	Aguilera, Christina	What A Girl Wants	3:18
15	Alice Deejay	Better Off Alone	6:50

id	date	rank
1	2000-02-26	87
1	2000-03-04	82
1	2000-03-11	72
1	2000-03-18	77
1	2000-03-25	87
1	2000-04-01	94
1	2000-04-08	99
2	2000-09-02	91
2	2000-09-09	87
2	2000-09-16	92
3	2000-04-08	81
3	2000-04-15	70
3	2000-04-22	68
3	2000-04-29	67
3	2000-05-06	66

Več istovrstnih meritev v večih tabelah

- Npr. meritve za vsako leto, po osebah, ...
- Po potrebi dodamo stolpce, ki odražajo delitev in združimo v eno tabelo
- Npr. delitev po letih: dodamo stolpec leto