POGLAVLJE 5

Istraživačka analiza podataka

Uvod

U ovom poglavlju naučićete kako da koristite vizuelizaciju i transformisanje da biste sistematski istražili svoje podatke – zadatak koji statističari zovu istraživačka analiza podataka (engl. *exploratory data analysis*, EDA). EDA je iterativni ciklus u kome vi:

1. Postavljate pitanja o podacima.
2. Tražite odgovore tako što vizuelizujete, transformišete i modelujute podatke.
3. Koristite ono što ste saznali da biste doterali svoja pitanja i/ili formulisali nova pitanja.

EDA nije formalni proces sa strogim skupom pravila, već je – pre svega – stanje uma. Tokom početnih faza procesa EDA, trebalo bi da potpuno slobodno istražujete svaku ideju koja vam padne na pamet. Neke od tih ideja biće korisne, dok druge neće voditi nigde. Kako budete nastavljali istraživanje, naići ćete na nekoliko posebno produktivnih oblasti koje ćete na kraju iskoristiti za izveštaj i saopštavanje rezultata drugima.

EDA je važan deo svake analize podataka – čak i ako pitanja dobijete na tanjiru – zato što uvek morate ispitati kvalitet podataka s kojima radite. Pročišćavanje (sređivanje) podataka samo je jedna aktivnost u okviru EDA: postavljate pitanja o tome da li dati podaci ispunjavaju vaša očekivanja ili ne. Da biste pročistili podatke, morate upotrebiti sve alatke istraživačke analize podataka: vizuelizaciju, transformisanje i modelovanje.

Preduslovi

U ovom poglavlju kombinovaćemo ono što ste naučili o paketima **dplyr** i **ggplot2** da bismo interaktivno postavljali pitanja, odgovarali na njih koristeći podatke a zatim postavljali nova pitanja.

```
library(tidyverse)
```

Pitanja

> Ne postoje rutinska statistička pitanja već samo upitne statističke rutine.
>
> – Sir David Cox

> Mnogo je bolji približan odgovor na pravo pitanje – koje je često nejasno – nego tačan odgovor na pogrešno pitanje, koje se uvek može precizno formulisati.
>
> –John Tukey

Tokom istraživačke analize podataka, cilj vam je da razvijete sposobnost da razumete podatke s kojima radite. To ćete najlakše postići tako što ćete koristiti pitanja kao alatke za usmeravanje svoje istrage. Kada postavljate pitanje, ono usmerava vašu pažnju na određen deo ispitivanog skupa podataka i pomaže vam da odlučite koje dijagrame, modele ili transformacije da napravite.

EDA je u osnovi kreativan proces. Kao i u većini kreativnih procesa, ključ postavljanja *kvalitetnih* pitanja jeste formulisati veliku *količinu* pitanja. Teško je postavljati dobra pitanja na početku analize zato što ne znate kakve uvide omogućava vaš skup podataka. S druge strane, svako novo pitanje koje postavite izložiće vas novom aspektu podataka i povećati vam šanse da nešto otkrijete. Brzo možete prodreti u najzanimljivije delove podataka – i razviti skup inspirativnih pitanja – ako svako pitanje pratite novim pitanjem zasnovanom na onome što ste saznali.

Nema pravila o tome koja bi pitanja trebalo da postavite kako bi dalje usmeravala vaše istraživanje. Međutim, dve vrste pitanja uvek će biti korisne za otkrivanje važnih stvari u podacima koje analizirate. Ta pitanja možete formulisati otprilike ovako:

1. Koja se vrsta varijacije javlja unutar mojih promenljivih?
2. Koja se vrsta kovarijacije javlja između mojih promenljivih?

U ostatku ovog poglavlja razmotrićemo ta dva pitanja. Objasnićemo šta su varijacija i kovarijacija, a navešćemo i nekoliko načina za odgovaranje na svako od pomenutih pitanja. Da bismo olakšali diskusiju, definisaćemo neke pojmove:

- *Promenljiva* je kvantitet, kvalitet ili svojstvo koje možete meriti.
- *Vrednost* je stanje promenljive u trenutku merenja. Vrednost promenljive se može menjati od jednog merenja do drugog.
- *Opservacija* – ili slučaj (engl. *case*) – jeste skup merenja obavljenih pod sličnim uslovima (sva merenja u jednoj opservaciji obično obavljate u isto vreme i na istom objektu). Jedna opservacija će sadržati nekoliko vrednosti, od kojih će svaka biti vezana s drugom promenljivom. Ponekad ćemo opservaciju zvati tačka podatka (engl. *data point*).

- *Tabelarni podaci* (engl. *tabular data*) predstavljaju skup vrednosti od kojih je svaka povezana s jednom promenljivom i jednom opservacijom. Tabelarni podaci su *uredni* ako je svaka vrednost smeštena u sopstvenu „ćeliju“ – svaka promenljiva u sopstvenu kolonu, a svaka opservacija u sopstveni red.

Svi podaci koje ste dosad videli u ovoj knjizi bili su uredni (sređeni, pročišćeni). U stvarnosti, većina podataka nije uredna, pa ćemo se ovim temama vratiti u poglavlju 9.

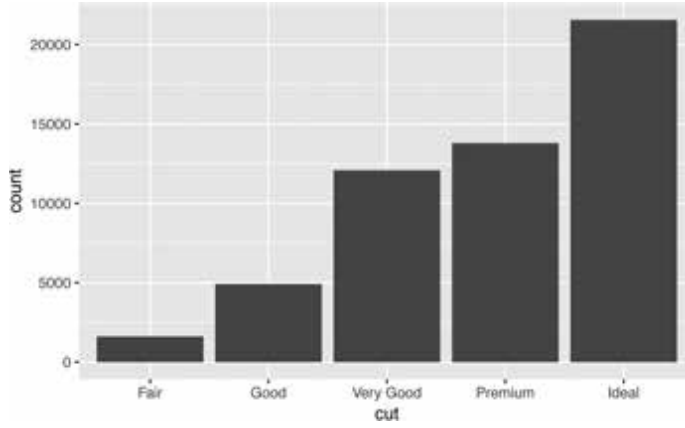
Varijacija

Varijacija je tendencija vrednosti promenljive da se menja od jednog merenja do drugog. Varijaciju možete lako videti u stvarnom životu; ukoliko bilo koju neprekidnu promenljivu izmerite dvaput, dobićete dva različita rezultata. To važi čak i ako merite konstantne količine – recimo, brzinu svetlosti. Svako vaše merenje sadržaće malu grešku koja će varirati od jednog merenja do drugog. Kategorijske promenljive takođe mogu varirati ako ih merite kod različitih subjekata (npr., boja očiju različitih ljudi) ili u različito vreme (npr., nivoi energije elektrona u različitim trenucima). Svaka promenljiva ima sopstveni šablon (engl. *pattern*) varijacije, što može otkriti zanimljive informacije. Taj šablon ćete najbolje razumeti ako grafički prikažete raspodelu vrednosti promenljivih.

Vizuelizacija raspodele

Kako ćete prikazati raspodelu promenljive, zavisiće od toga da li je promenljiva kategorijska ili neprekidna. Promenljiva je kategorijska (engl. *categorical*) ako može da uzme samo jednu od vrednosti iz malog skupa. U R-u, kategorijske promenljive se obično čuvaju kao faktori, tj. znakovni vektori (engl. *character vectors*). Da biste istražili raspodelu kategorijske promenljive, koristite stubičasti dijagram:

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```

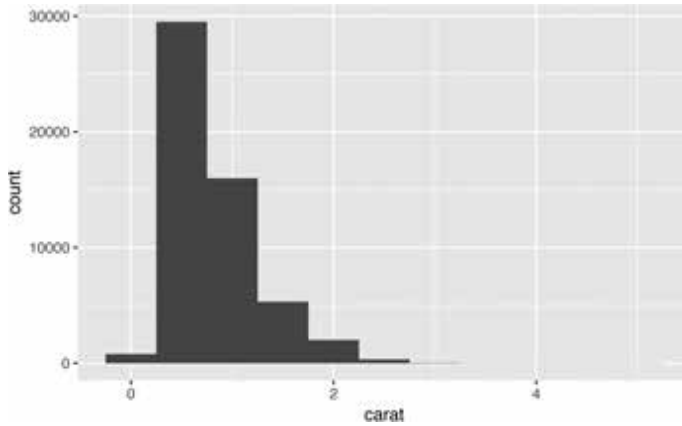


Visina stubića prikazuje koliko ima opservacija za svaku vrednost x. Te vrednosti možete izračunati ručno, pomoću funkcije `dplyr::count()`:

```
diamonds %>%
  count(cut)
#> # Tibl: 5 × 2
#>         cut     n
#>       <ord>  <int>
#> 1      Fair  1610
#> 2      Good  4906
#> 3 Very Good 12082
#> 4   Premium 13791
#> 5     Ideal 21551
```

Promenljiva je neprekidna (engl. *continuous*) ako može uzeti bilo koju vrednosti iz beskonačnog skupa uređenih vrednosti. Brojevi i vrednosti datum-vreme dva su primera neprekidnih promenljivih. Da biste ispitali raspodelu neprekidne promenljive, koristite histogram:

```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```



To možete izračunati i ručno, kombinujući funkcije `dplyr::count()` i `ggplot2::cut_width()`:

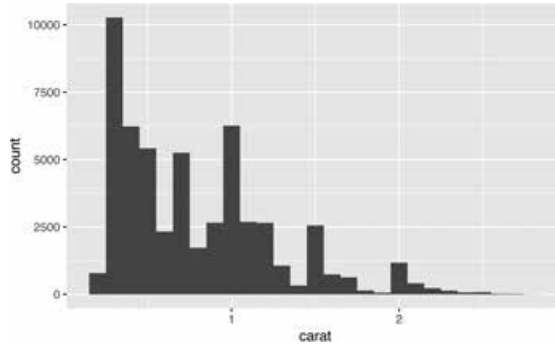
```
diamonds %>%
  count(cut_width(carat, 0.5))
#> # Tibl: 11 × 2
#>   `cut_width(carat, 0.5)`     n
#>                    <fctr> <int>
#> 1           [-0.25,0.25]   785
#> 2            (0.25,0.75] 29498
#> 3            (0.75,1.25] 15977
#> 4            (1.25,1.75]  5313
#> 5            (1.75,2.25]  2002
#> 6            (2.25,2.75]   322
#> # ... i još 5 redova
```

Histogram deli x-osu na intervale (odeljke) iste širine, a zatim koristi visinu svakog stubića da bi se prikazao broj opservacija koje spadaju u svaki interval. Na prethodnom dijagramu, najviši stubić pokazuje da je u gotovo 30.000 opservacija vrednost `carat` između 0,25 i 0,75, što su leva i desna ivica intervala.

Širinu intervala na histogramu možete podesiti pomoću argumenta `bin width`, koji se meri u jedinicama promenljive `x`. Trebalo bi da uvek ispitate razne širine intervala kada radite s histogramima, pošto različite širine mogu otkriti različite šablone raspodele. Na primer, evo kako izgleda prethodni dijagram kada zumiramo samo dijamante manje od tri karata i izaberemo uži interval:

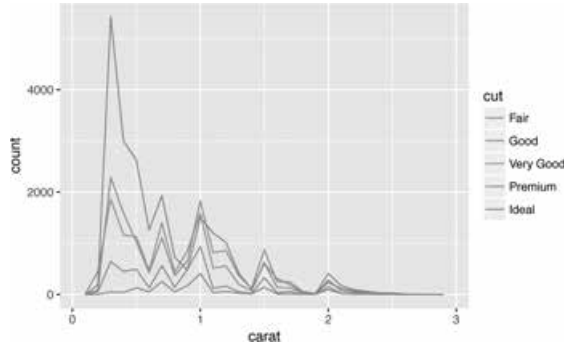
```
smaller <- diamonds %>%
  filter(carat < 3)

ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.1)
```



Ako želite da na isti dijagram postavite više histograma jedne preko drugih, preporučujemo da koristite funkciju `geom_freqpoly()` umesto `geom_histogram()`. Funkcija `geom_freqpoly()` izračunava isto što i `geom_histogram()`, ali – umesto da broj tačaka prikaže stubićima – koristi linije. Mnogo je lakše razumeti preklopljene linije nego stubiće:

```
ggplot(data = smaller, mapping = aes(x = carat, color = cut)) +
  geom_freqpoly(binwidth = 0.1)
```



U BOJI NA KRAJU KNJIGE

Postoji nekoliko problema sa ovom vrstom dijagrama, kojima ćemo se vratiti u odeljku „Kategorijska i neprekidna promenljiva“, na strani 82.

Pošto sada možete da grafički predstavite varijaciju, šta bi trebalo da tražite na svojim dijagramima? I kakva bi naknadna pitanja trebalo da postavljate? U nastavku je data lista najkorisnijih vrsta informacija koje ćete pronaći na dijagramima, zajedno s dodatnim pitanjima za svaki tip informacija. Za postavljanje dobrih naknadnih pitanja najvažnije je da se oslonite na svoju radoznalost (O čemu želite da saznate više?) i na svoj skepticizam (Kako bi ovo moglo da vodi na pogrešan put?).

Uobičajene vrednosti

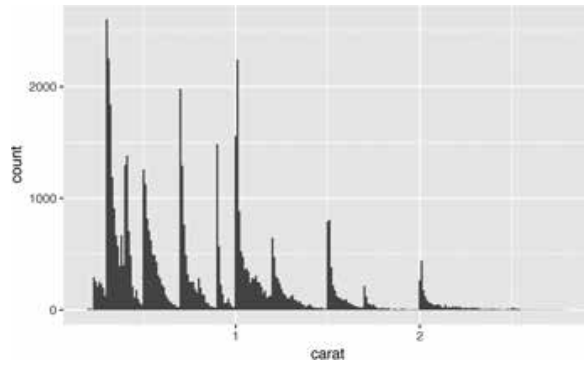
I na stubičastim dijagramima i na histogramima, visoki stubići prikazuju uobičajene (tipične) vrednosti promenljive, a niži stubići prikazuju vrednosti koji se javljaju ređe. Mesta na kojima nema stubića otkrivaju vrednosti koje nisu zapažene u vašim podacima. Da biste te informacije pretvorili u korisna pitanja, tražite bilo šta što je neočekivano:

- Koje vrednosti su najčešće? Zašto?
- Koje vrednosti su retke? Zašto? Da li to odgovara vašim očekivanjima?
- Zapažate li neke neuobičajene šablone? Kako biste ih mogli objasniti?

Na primer, sledeći histogram sugeriše nekoliko zanimljivih pitanja:

- Zašto ima više dijamanata čija je veličina ceo broj karata ili neka uobičajena decimalna vrednost karata?
- Zašto ima više dijamanata malo desno od svakog maksimuma nego onih koji su malo levo od svakog maksimuma?
- Zašto nema dijamanata većih od 3 karata?

```
ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.01)
```

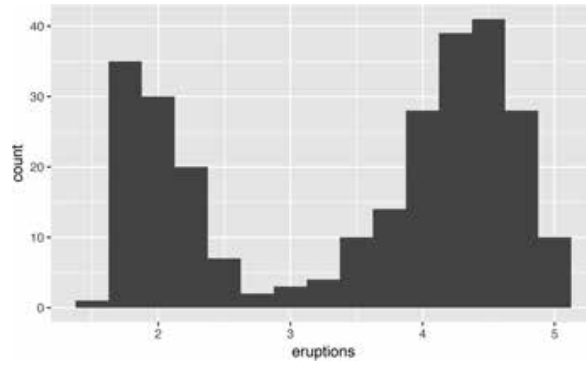


Uopšteno govoreći, klasteri (grozdovi) sličnih vrednosti sugerišu da u ispitivanom skupu podataka postoje podgrupe. Da biste razumeli te podgrupe, postavite sledeća pitanja:

- Po čemu su opservacije unutar svakog klastera slične jedne drugima?
- Po čemu se opservacije u različitim klasterima međusobno razlikuju?
- Kako možete objasniti ili opisati klastere?
- Zašto bi izgled klastera mogao da vara?

Naredni histogram prikazuje trajanje (u minutima) 272 erupcije gejzera Old Faithful u nacionalnom parku Jelouston. Izgleda da trajanja erupcija uglavnom spadaju u dve grupe: kratke erupcije (oko 2 minuta) i dugačke erupcije (4–5 minuta), a malo je onih između:

```
ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_histogram(binwidth = 0.25)
```

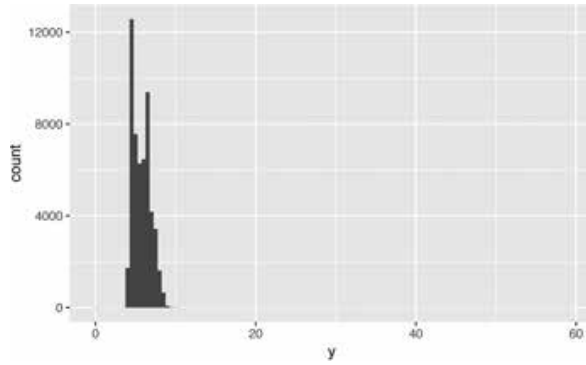


Mnoga od prethodno navedenih pitanja podstaći će vas da istražite vezu *između* promenljivih – na primer, da biste videli mogu li vrednosti jedne promenljive objasniti ponašanje druge promenljive. To ćemo uskoro opisati.

Neuobičajene vrednosti

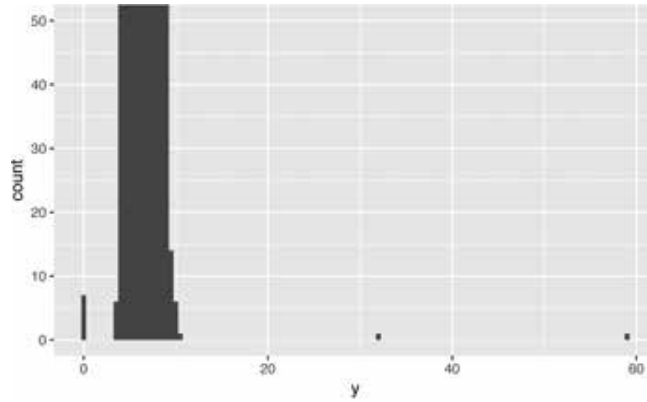
Neuobičajene vrednosti, tj. netipične tačke (engl. *outliers*) jesu opservacije koje su neočekivane – one koje odstupaju od datog šablona. Takve tačke su ponekad posledica grešaka pri unošenju podataka; u drugim slučajevima, one ukazuju na važno novo saznanje. Kada imate mnogo podataka, netipične tačke je često teško videti na histogramu. Na primer, pogledajte raspodelu promenljive y iz skupa podataka o dijamantima. Jedini dokaz postojanja netipičnih tačaka predstavljaju neuobičajeno široke granice na y-osi:

```
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```



Ima toliko mnogo opservacija u očekivanim intervalima da su retki intervali toliko kratki da ih ne možete videti (mada ćete možda videti nešto ako budete intenzivno posmatrali tačku 0). Da bismo lako uočili neuobičajene vrednosti, moramo zumirati deo y-ose s malim vrednostima pomoću funkcije `coord_cartesian()`:

```
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  coord_cartesian(ylim = c(0, 50))
```



(Funkcija `coord_cartesian()` ima i argument `xlim()` za slučaj kada treba da zumirate x-osu. Paket **ggplot2** sadrži funkcije `xlim()` i `ylim()`, koje rade malo drugačije: one odbacuju podatke koji su van granica.)

To nam omogućava da uočimo tri neuobičajene vrednosti: 0, ~30 i ~60. Izbacujemo ih pomoću paketa **dplyr**:

```
unusual <- diamonds %>%
  filter(y < 3 | y > 20) %>%
  select(price, x, y, z) %>%
  arrange(y)
```

```
unusual
#> # A tibble: 9 × 4
#>   price     x     y     z
#>   <int> <dbl> <dbl> <dbl>
#> 1  5139  0.00   0.0  0.00
#> 2  6381  0.00   0.0  0.00
#> 3 12800  0.00   0.0  0.00
#> 4 15686  0.00   0.0  0.00
#> 5 18034  0.00   0.0  0.00
#> 6  2130  0.00   0.0  0.00
#> 7  2130  0.00   0.0  0.00
#> 8  2075  5.15  31.8  5.12
#> 9 12210  8.09  58.9  8.06
```

Promenljiva y meri jednu od tri dimenzije dijamanata, u mm. Znamo da dijamanti ne mogu imati širinu 0 mm, što znači da su dobijene vrednosti netačne. Možemo, takođe, sumnjati u vrednosti 32 mm i 59 mm: ti dijamanti su dugački preko 3 cm, a ne koštaju stotine hiljada dolara!

Dobra je praksa ponoviti analizu s netipičnim tačkama i bez njih. Ako one minimalno utiču na rezultate i ne možete ustanoviti zašto su tamo, razumno je zameniti ih nedostajućim vrednostima i nastaviti rad. Međutim, ukoliko je njihov uticaj na rezultate značajan, ne bi trebalo da ih odbacite bez opravdanja. Moraćete da ustanovite šta ih je uzrokovalo (npr., greška pri unošenju podataka) i da u izveštaju navedete da ste ih uklonili.

Vežbe

1. Istražite raspodelu promenljivih `x`, `y` i `z` u skupu podataka `diamonds`. Šta ste saznali? Kako biste mogli zaključiti koja dimenzija predstavlja dužinu, širinu i dubinu.
2. Istražite raspodelu promenljive `price`. Vidite li nešto neobično ili neočekivano? (Pomoć: pažljivo razmislite o širini intervala (`bin width`) i obavezno isprobajte širok opseg vrednosti.)
3. Koliko ima dijamanata od 0,99 karata? Koliko ih je od 1 karata? Po vašem mišljenju, šta uzrokuje tu razliku?
4. Uporedite `coord_cartesian()` sa `xlim()` ili `ylim()` kada zumirate histogram. Šta se dešava ako ostavite nepodešenu promenljivu `binwidth`? Šta se dešava ukoliko pokušate da zumirate tako da se vidi samo polovina trake histograma?

Nedostajuće vrednosti

Ako ste u ispitivanom skupu podataka naišli na neobične vrednosti i samo želite da nastavite analizu, imate dve mogućnosti:

- Odbacite ceo red s čudnim vrednostima:

```
diamonds2 <- diamonds %>%
  filter(between(y, 3, 20))
```

Ne preporučujemo ovu opciju jer to što jedno merenje nije dobro ne znači da su sva ostala u redu. Osim toga, ako imate nekvalitetne podatke i primenite ovaj pristup na svaku promenljivu, možda vam na kraju neće ostati nijedan podatak!

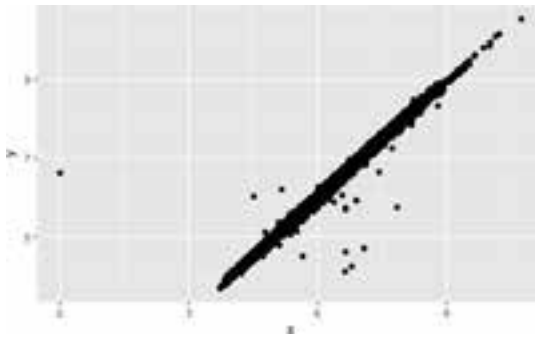
- Umesto toga, preporučujemo da zamenite netipične vrednosti nedostajućim vrednostima. To ćete najlakše uraditi ako upotrebite funkciju `mutate()` da biste datu promenljivu zamenili izmenjenom. Možete koristiti funkciju `ifelse()` da biste neuobičajene vrednosti zamenili sa `NA`:

```
diamonds2 <- diamonds %>%
  mutate(y = ifelse(y < 3 | y > 20, NA, y))
```

Funkcija `ifelse()` ima tri argumenta. Prvi argument, `test`, trebalo bi da bude logički vektor. Rezultat će sadržati vrednost drugog argumenta, `yes`, kada je `test` `TRUE`, a vrednost trećeg argumenta, `no`, kada je `FALSE`.

Kao i R, i paket **ggplot2** vodi se filozofijom da nedostajuće vrednosti nikada ne treba da nestanu neprimetno. Nije očigledno gde bi trebalo da nacrtate nedostajuće vrednosti, pa ih **ggplot2** ne prikazuje na dijagramu, ali vas upozorava da su uklonjene:

```
ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +
  geom_point()
#> Warning: Removed 9 rows containing missing values
#> (geom_point).
```

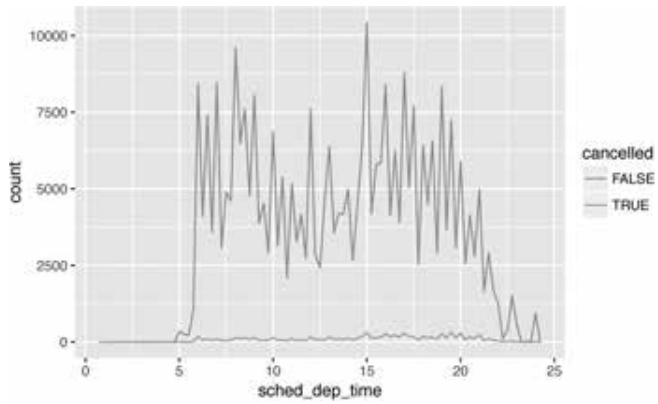


Da se upozorenje ne bi prikazivalo, zadajte `na.rm = TRUE`:

```
ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +
  geom_point(na.rm = TRUE)
```

Ponekad želite da shvatite po čemu se opservacije s nedostajućim vrednostima razlikuju od opservacija sa evidentiranim vrednostima. Na primer, u skupu podataka `nycflights13::flights`, nedostajuće vrednosti promenljive `dep_time` ukazuju na to da je dati let bio otkazan. Zbog toga biste možda želeli da uporedite planirana vremena odlaska za otkazane i neotkazane letove. To možete uraditi tako što ćete napraviti novu promenljivu pomoću funkcije `is.na()`:

```
nycflights13::flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %/% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + sched_min / 60
  ) %>%
  ggplot(mapping = aes(sched_dep_time)) +
    geom_freqpoly(
    mapping = aes(color = cancelled),
    binwidth = 1/4
  )
```



U BOJI NA KRAJU KNJIGE

Međutim, ovaj dijagram nije baš dobar pošto ima više neotkazanih letova nego otkazanih. U narednom odeljku istražićemo neke tehnike za poboljšavanje ovog poređenja.

Vežbe

1. Šta se dešava s nedostajućim vrednostima na histogramu, a šta na stubičastom dijagramu? Zašto postoji razlika?
2. Šta radi naredba `na.rm = TRUE` u funkcijama `mean()` i `sum()`?

Kovarijacija

Ako varijacija opisuje ponašanje *unutar* promenljive, kovarijacija opisuje ponašanje *između* promenljivih. *Kovarijacija* je tendencija da vrednosti dve ili više promenljivih variraju zajedno na načine koji su međusobno povezani. Kovarijaciju ćete najbolje uočiti ako prikažete vezu između dve ili više promenljivih. Kako bi to trebalo da uradite, opet zavisi od tipa promenljivih.