

Metod i kriminologi

Kriminolog grupp vid Umeå universitet

2024-11-17

Contents

1	Introduktion	5
2	Set up	7
2.1	Packages	7
3	Univariat analysis	9
3.1	Ladda in data till R	9
3.2	Data överblick.	9
3.3	Descriptive statistics	10
3.4	Visualisering	11
4	Bivariat analys	17
4.1	Krosstabler: Två kategorisk variabler	17
4.2	Korrelationer: Två kontinuerlig variabel	20
4.3	Jämförelsen av medelvärde: Kontinuerlig beroende variabel och kategorisk oberoende variabel	24

Chapter 1

Introduktion

Välkommen till **Kriminologi A**.

Vi bygger upp innehållet i den här boken över tid och uppdaterar den allt eftersom för att ge den bästa möjliga introduktionen till kvantitativa metoder inom kriminologi som vi kan. Vi har börjat använda R (jämfört med andra som SPSS, STATA eller Jamovi) i dessa kriminologikurser av flera anledningar. För det första tror vi att R är mer mångsidigt, vilket gör det möjligt för studenter att lära sig att använda olika statistiska verktyg som finns tillgängliga i andra program, samtidigt som de har möjlighet att tillåta andra typer av analyser som är relevanta för kriminologi (till exempel rumsliga analyser eller nätverksanalyser) som inte är tillgängliga i andra program. För det andra används R i större utsträckning inom både akademisk forskning och andra yrkesområden av intresse för studenter som studerar kriminologi, t.ex. arbete i lokala och nationella myndigheter. För det tredje är R gratis och open access, vilket innebär att det inte kommer att kosta dem pengar i framtiden att ge studenterna en introduktion till detta verktyg. För det fjärde tror vi att det finns pedagogiska fördelar som studenter får när de tvingas tänka på data och analyser med kod istället för peka-och-klicka-program. Ett exempel på detta är att kommandoraden för en linjär regressionsmodell i R är mycket lik den faktiska formeln, vilket innebär att eleverna är mer benägna att komma ihåg varje komponent i formeln tillsammans med deras betydelse. Den tänkta användningen av denna bok är att paras ihop med kriminologikurserna vid Umeå universitet och de exempel vi använder är från föreläsningarna och workshopparna i den kursen. Där det är möjligt försöker vi tillhandahålla textinformation för att förklara begreppen och hur de tillämpas i R-miljön utan att boken blir för “texttung”.

Chapter 2

Set up

Här kommer instruktioner till studenter om hur att uplägga R. Följ länken och ladda när R sen ladda när Rstudio. Se till att du ladda när versionen som passar ditt dator (windows eller mac).

<https://posit.co/download/rstudio-desktop/>

Efter du har laddat ner båda öppna R studio gå till 'file' och tryck på 'ny projekt'.

2.1 Packages

När man först installera R eller när man uppdatera R man behöver installera paketen (packages) som vi använder att hjälpa bearbeta visualisera samt analysera data. Att installera dessa packeter kopiera och klistra in den följande kod och tryck 'enter'. Det finns hur många som helst paketen i R som byggdes upp att göra många olika saker. I allmänhet om du vill göra någonting i R förmodligen det finns en paket för det.

"tidyverse", "rio", "ggmap", "knitr", "psych", "emmeans", "sjPlot"

Chapter 3

Univariat analysis

Detta kapitel handlar om *univariat analys* som vi diskuterade i klassrummet.

Här ska vi prata om olika skal nivåer och hur de olika typer ser ut i data formen. Sen börja med data data behandling och grunläggande visualisering.

3.1 Ladda in data till R

Först vi behöver ladda in data kopiera och och klistra in data. Kan du ser datafilen i R studio? Om inte se till att du had lagt filen i samma plats som ditt R projekt. När du har gjort den och kan se filen i R studio då du kan trycka den gröna plus tecken högst upp i vänsterhörnet. Kopiera och klistra in den följande kod till dina skript och kör koden.

```
library(rio)
df<-import('NTU 2017-2021.RData')
```

```
## Warning: Missing `trust` will be set to FALSE by default for RData in 2.0.0.
```

Nu du borde se en ny objekt i enironment fönstret som heter 'df' som inkludera 375790 observationer och 6 variabler. Om du vill titta på datan du kan trycka på den och en ny fönster ska öppna med datavy.

3.2 Data överblick.

Här kommer två sätt att får samma information

Table 3.1: Here is a nice table!

alder_4	Brottsoffer_hot	Fortroende_index	Radsla_oro	Radsla_beteende	S
65-84	Nej	8	Orolig	Sällan eller aldrig avstått aktivitet	I
65-84	Nej	20	Orolig	Sällan eller aldrig avstått aktivitet	S
65-84	NA	7	Inte orolig	Sällan eller aldrig avstått aktivitet	I
65-84	Nej	NA	Inte orolig	Sällan eller aldrig avstått aktivitet	I
65-84	Nej	19	Orolig	Ofta avstått aktivitet	I
65-84	Nej	15	Orolig	Ofta avstått aktivitet	I
65-84	Nej	NA	Orolig	Sällan eller aldrig avstått aktivitet	S
65-84	Nej	6	Inte orolig	Sällan eller aldrig avstått aktivitet	I
65-84	Nej	14	Orolig	Sällan eller aldrig avstått aktivitet	I
65-84	Nej	19	Orolig	Sällan eller aldrig avstått aktivitet	S

```
head(df)
```

```
##   alder_4 Brottsoffer_hot Fortroende_index Radsla_oro
## 1   65-84              Nej              8      Orolig
## 2   65-84              Nej             20      Orolig
## 3   65-84             <NA>              7  Inte orolig
## 4   65-84              Nej             NA  Inte orolig
## 5   65-84              Nej             19      Orolig
## 6   65-84              Nej             15      Orolig
##                                Radsla_beteende Storstad
## 1 Sällan eller aldrig avstått aktivitet  Inte storstad
## 2 Sällan eller aldrig avstått aktivitet      Storstad
## 3 Sällan eller aldrig avstått aktivitet  Inte storstad
## 4 Sällan eller aldrig avstått aktivitet  Inte storstad
## 5                                Ofta avstått aktivitet  Inte storstad
## 6                                Ofta avstått aktivitet  Inte storstad
```

```
knitr::kable(
  head(df, 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

3.3 Descriptive statistics

Vi måste alltid komma ihåg skal nivå av variablerna när vi kör alla slags analysen. Även i univariatanalys t.ex. vad betyder det att räkna ut medelvärdet av en dikatom variabel som ‘storstad’?

```
library(psych)
knitr::kable(describe(df))
```

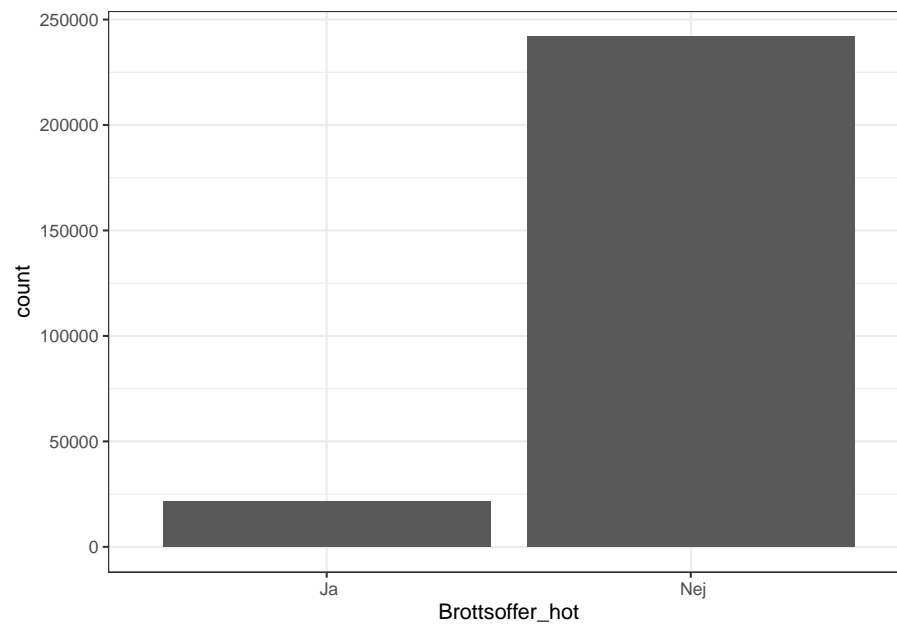
	vars	n	mean	sd	median	trimmed	mad	min	max	range
alder_4*	1	375590	2.841875	1.0327887	3	2.927344	1.4826	1	4	3
Brottsoffer_hot*	2	371042	1.925895	0.2619417	2	2.000000	0.0000	1	2	1
Fortroende_index	3	269696	11.408805	4.5050388	12	11.625914	4.4478	0	20	20
Radsla_oro*	4	366123	1.451627	0.4976552	1	1.439534	0.0000	1	2	1
Radsla_beteende*	5	373014	1.117682	0.3222315	1	1.022104	0.0000	1	2	1
Storstad*	6	375590	1.289012	0.4533041	1	1.236265	0.0000	1	2	1

3.4 Visualisering

```
#Här är en plot av variabeln 'Brottsoffer_hot'.
library(tidyverse)
```

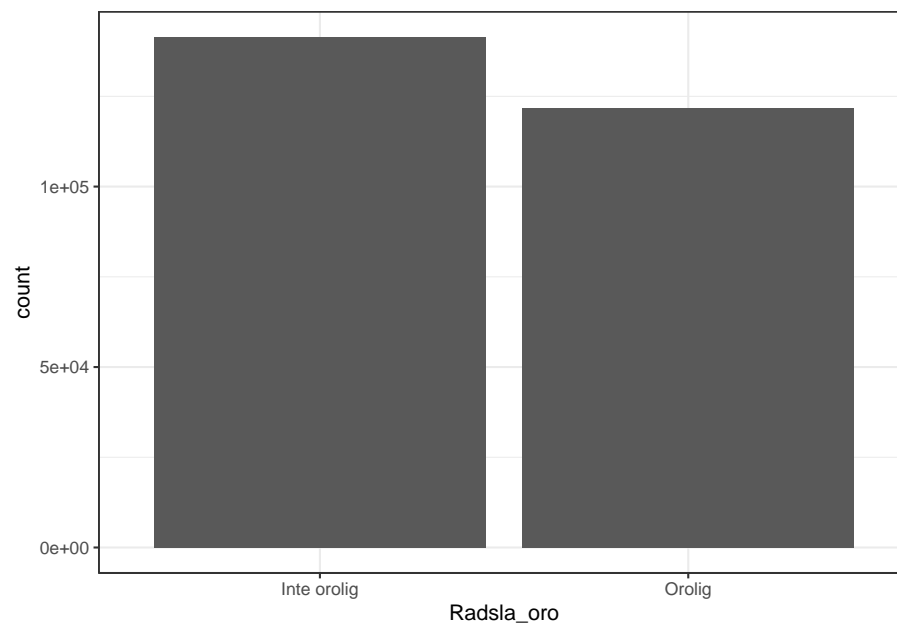
```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.5.1      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x ggplot2::%>%() masks psych::%>%()
## x ggplot2::alpha() masks psych::alpha()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
df %>% drop_na() %>% #Det har ta bort alla NA annars det får sin egen kolumn
ggplot()+ geom_bar(aes(x=Brottsoffer_hot))+theme_bw()
```



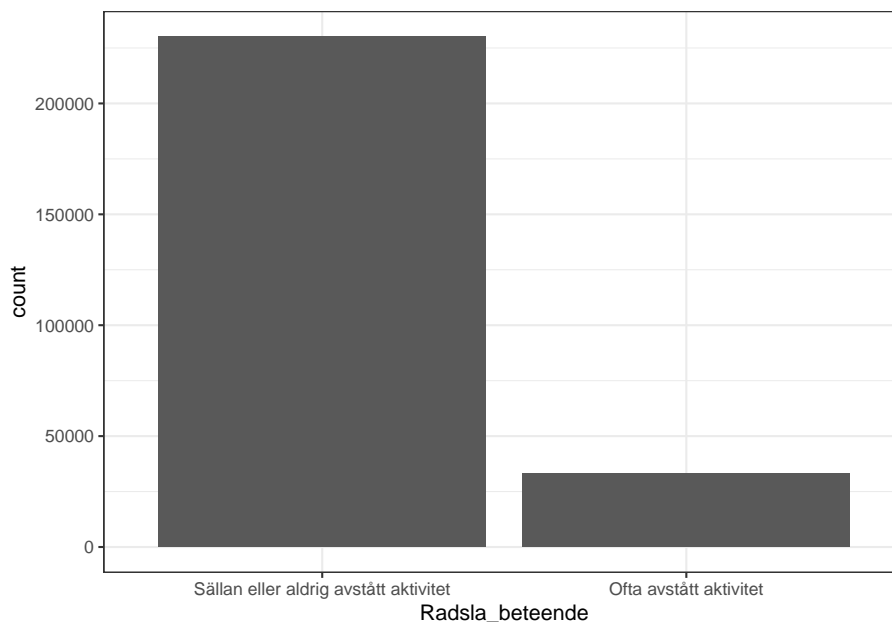
#Här är en plot av variabeln 'Radsla_oro'.

```
df %>% drop_na() %>% #Det har ta bort alla NA annars det får sin egen kolumn  
ggplot()+ geom_bar(aes(x=Radsla_oro))+theme_bw()
```



#Här är en plot av variabeln 'Radsla_beteende'.

```
df %>% drop_na() %>% #Det har ta bort alla NA annars det får sin egen kolumn
  ggplot()+ geom_bar(aes(x=Radsla_beteende))+theme_bw()
```



De tre variablerna var alla dikotomiserad i.e. de varierar bara mellan två kategorier. Men vi kan också skapa en 'index' av många variabler

I denna fall vi har skapat ett additivt index (en sammanslagning av fem frågor) som mäter om respondenten har stort eller litet förtroende för rättsväsendet. Kontinuerlig: 0 (lågt förtroende) till 20 (högt förtroende) (Ingenåsikt/vetinte=bortfall)

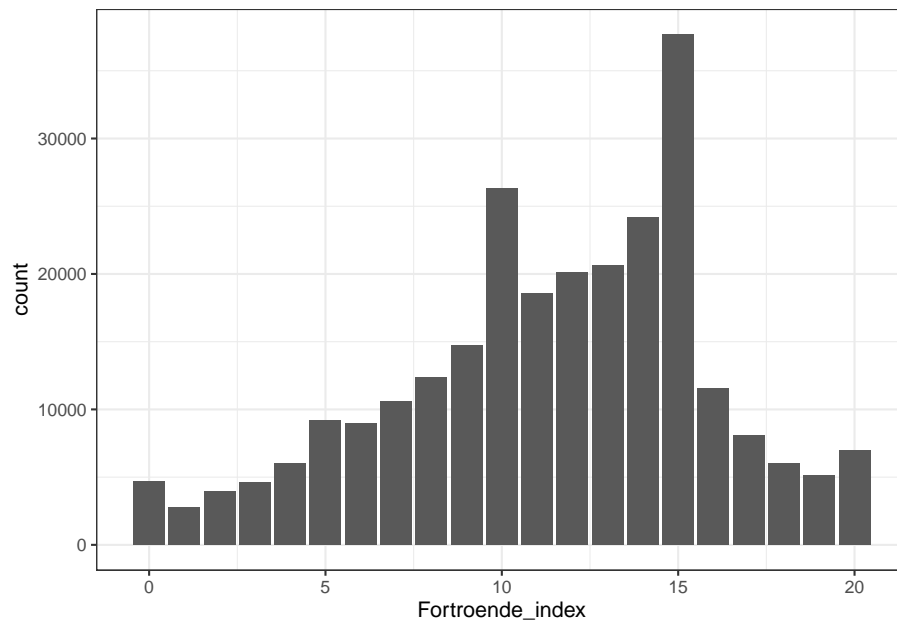
Vi kan se de beskrivande statistiker för bara denna variabel om vi lägga till variabelns namn efter datasetets namn och en '\$' tecken

```
knitr::kable(describe(df$Förtroende_index, skew = FALSE))
```

	vars	n	mean	sd	median	min	max	range	se
X1	1	269696	11.4088	4.505039	12	0	20	20	0.0086748

#Här är en plot av variabeln 'Förtroende_index'.

```
df %>% drop_na() %>% #Det har ta bort alla NA annars det får sin egen kolumn
  ggplot()+ geom_bar(aes(x=Förtroende_index))+theme_bw()
```



Sen vi kan även ladda upp en till data set om vi vill. Lada ner den 'shootings' excel fil från canvas och spara den i samma map som du har R projektet. När du har gjort den det borde dycker upp i fönstret igen. Sen kör denna kod att ladda in data till R och nämna den 'shootings'

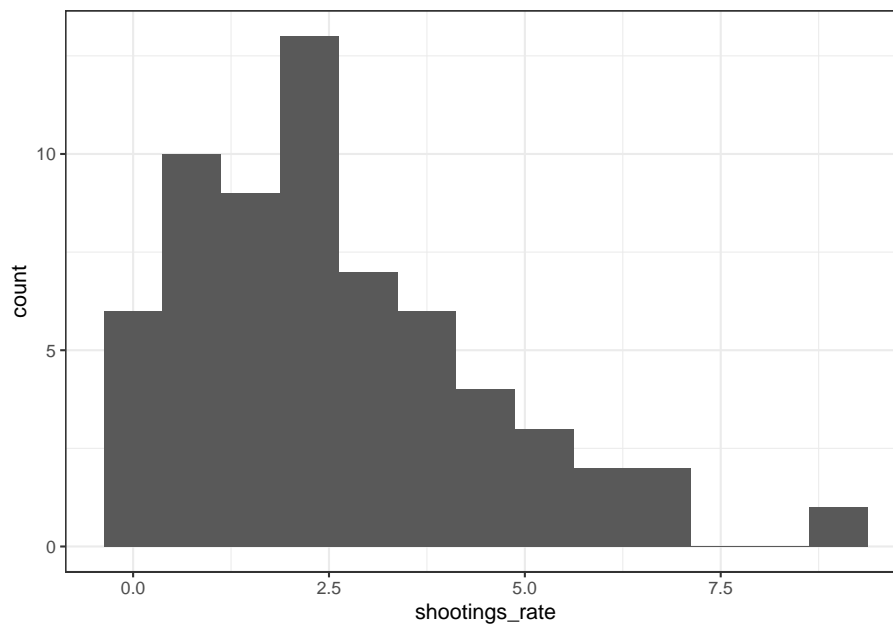
```
shootings<- import('shootings.csv')
#Sen kolla hur det ser ut
head(shootings)
```

```
##   year      region shootings_rate trust_police_average education_average
## 1 2018 västragötaland      2.982781          2.277397          2.252655
## 2 2018      blekinge      1.252474          2.430769          2.112299
## 3 2018     jönköping      2.494284          2.264463          2.113990
## 4 2018 västernorrland      0.407410          2.447059          2.180451
## 5 2018     västmanland      1.825290          2.169014          2.216000
## 6 2018     södermanland      4.411341          2.272727          2.148036
##   income_average
## 1      3.501167
## 2      3.200000
## 3      3.400552
## 4      3.315175
## 5      3.502146
## 6      3.370253
```

```
knitr::kable(describe(shootings$shootings_rate, skew = FALSE))
```

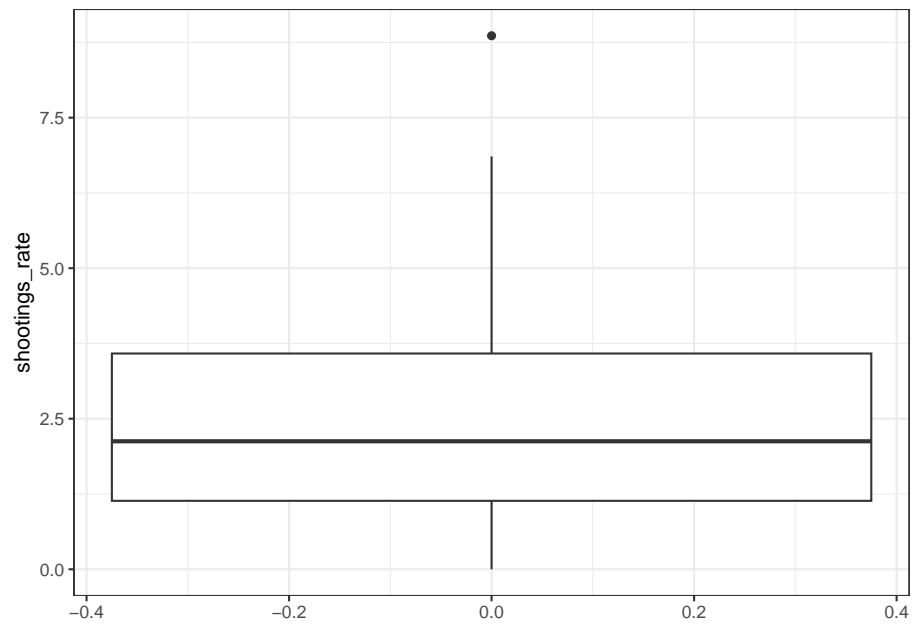
	vars	n	mean	sd	median	min	max	range	se
X1	1	63	2.557833	1.898535	2.125386	0	8.860789	8.860789	0.2391929

```
shootings %>% drop_na() %>%
  ggplot(aes(shootings_rate)) + geom_histogram(binwidth= .75) +
  theme_bw()
```



Vi kan också skapa en boxplot av variabeln

```
shootings %>% drop_na() %>%
  ggplot(aes(x= shootings_rate)) + geom_boxplot() + coord_flip() + theme_bw()
```



Chapter 4

Bivariat analys

4.1 Krosstabler: Två kategorisk variabler

Precis som alla andra saker i R det finns många olika sätt att skapa kross tabell i R. Här er den enklaste exempel som beräkna bara hur många fall finns det i varje kategori.

```
table(df$Radsla_oro, df$Storstad)
```

```
##
##           Inte storstad Storstad
## Inte orolig      140012      60760
## Orolig           119997      45354
```

Men vi kan också ändra den kod att räkna ut procent i varje låda

```
prop.table(table(df$Radsla_oro, df$Storstad),2)
```

```
##
##           Inte storstad Storstad
## Inte orolig      0.5384891 0.5725917
## Orolig           0.4615109 0.4274083
```

Och vi kan även räkna ut de så att de ser ut som procent

```
round(100* prop.table(table(df$Radsla_oro, df$Storstad),2))
```

```
##
##           Inte storstad Storstad
##   Inte orolig           54      57
##   Orolig              46      43
```

Men nu att räkna ut χ^2 vi behöver först räkna ut förväntade värdena om de fanns ingen skillnad mellan grupperna. Formel att räkna ut varje cell i en tabell är $e = \frac{radtotal * kolttotal}{total}$ sen från tabellen ovanför det skulle ser ut så här

```
(200772*260009)/366123
```

```
## [1] 142581.9
```

```
(200772*106114)/366123
```

```
## [1] 58190.06
```

```
(165351*260009)/366123
```

```
## [1] 117427.1
```

```
(165351*106114)/366123
```

```
## [1] 47923.94
```

Sen med förväntade värdena vi kan få χ^2 genom formeln

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Och vi kan använda R att räkna ut χ^2 för oss så här:

```
chisq.test(df$Storstad, df$Radsla_oro)
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  df$Storstad and df$Radsla_oro
## X-squared = 353.74, df = 1, p-value < 2.2e-16
```

Men vi kan också använda en paket i R som heter sjPlot att visa alla information i samma plats samt kör χ^2 prov.

```
library(sjPlot)
```

```
## Warning: package 'sjPlot' was built under R version 4.4.2
```

```
tab_xtab(
  var.row = df$Radsla_oro,
  var.col = df$Storstad,
  show.summary = TRUE,
  show.col.prc = TRUE,
  use.viewer = TRUE
)
```

Radsla_oro

Storstad

Total

Inte storstad

Storstad

Inte orolig

14001253.8 %

6076057.3 %

20077254.8 %

Orolig

11999746.2 %

4535442.7 %

16535145.2 %

Total

260009100 %

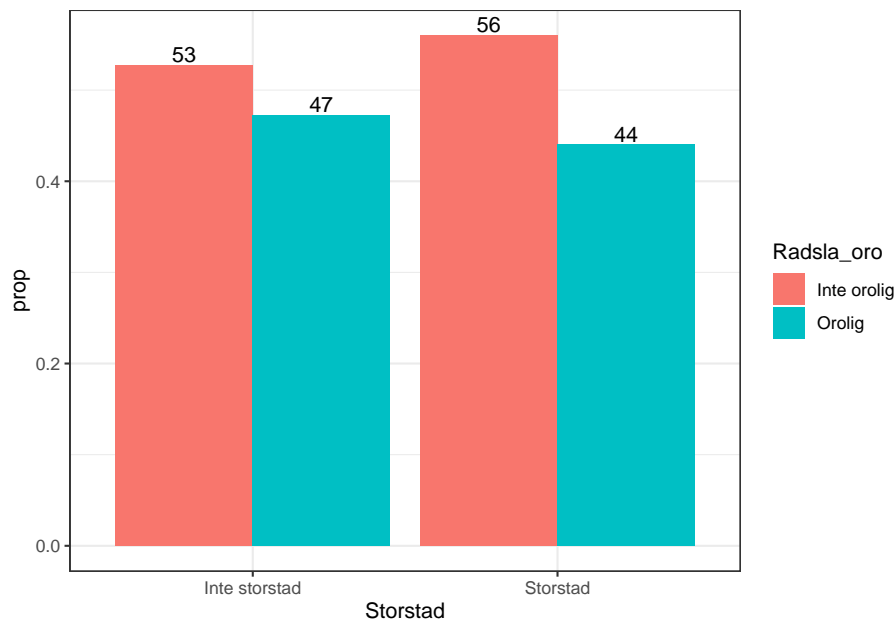
106114100 %

366123100 %

2=353.742 · df=1 · φ=0.031 · p=0.000

Men då vi kan också skapa en figur av skillnader andel de som är orolig baserad på stads storlek.

```
df %>%
  drop_na() %>%
  count(Storstad, Radsla_oro) %>%
  group_by(Storstad) %>%
  mutate(prop = n / sum(n), na.rm=TRUE) %>%
  ggplot(aes(x = Storstad, y = prop, fill = Radsla_oro)) +
  geom_col(position = position_dodge()) +
  geom_text(aes(label = round(100 * prop)),
            position = position_dodge(.9), vjust = -.2)
  ) +
  theme_bw()
```



4.2 Korrelationer: Två kontinuerlig variabel

Bara två variabler från datasetet:

```
shootings %>%
  select(shootings_rate, education_average) %>% #Första välja variablerna att inkludera
  tab_corr(corr.method='pearson', triangle = 'lower')
```

shootings_rate

education__average
shootings__rate

education__average
0.381**

Computed correlation used pearson-method with listwise-deletion.

Sen vi kan kolla på hur alla fyra variabler korrelera med vandra:

```
shootings %>%
  select(shootings_rate, education_average, trust_police_average, income_average) %>% #Första vä
  tab_corr(corr.method='pearson', triangle = 'lower')
```

shootings__rate
education__average
trust__police__average
income__average
shootings__rate

education__average
0.381**

trust__police__average
-0.151
-0.066

```
income_average
0.562***
0.565***
-0.184
```

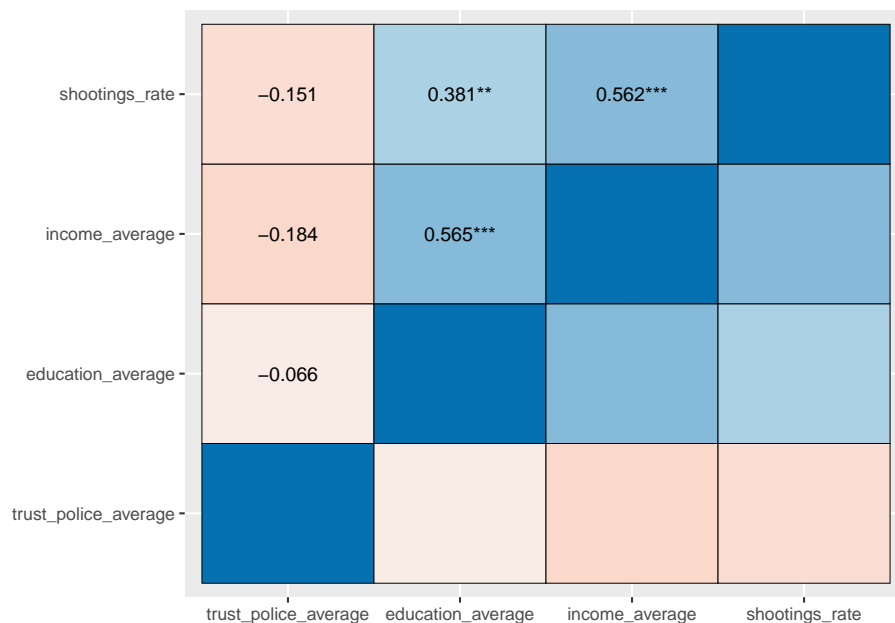
Computed correlation used pearson-method with listwise-deletion.

```
shootings %>%
  select(shootings_rate, education_average, trust_police_average, income_average) %>%
  sjp.corr(corr.method = 'pearson')

## Warning in sjp.corr(., corr.method = "pearson"): 'sjp.corr' is deprecated.
## Please use 'correlation::correlation()' and its related plot()-method.

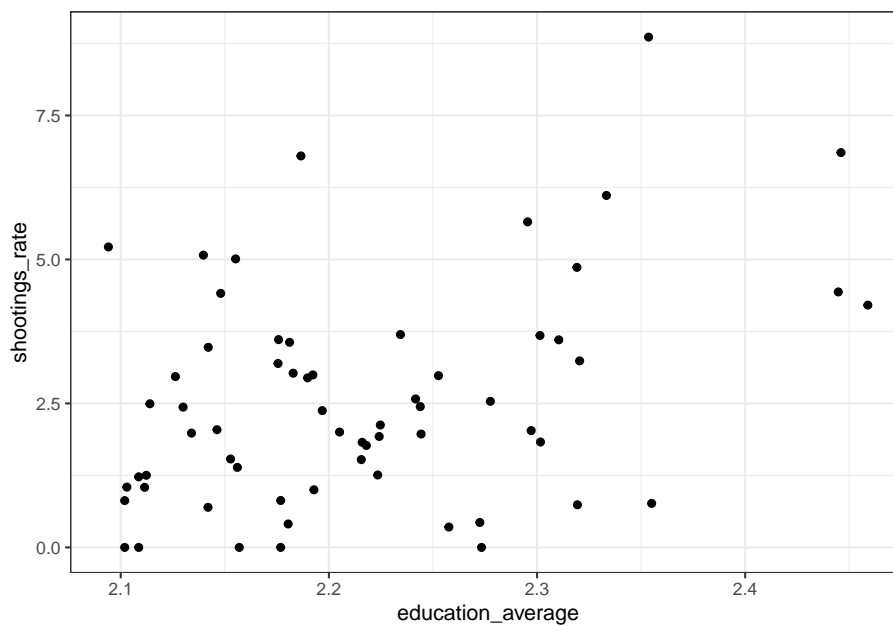
## Computing correlation using pearson-method with listwise-deletion...

## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_text()`).
```



Sen vi kan visualisera sambandet tillsammans med iakttag punkterna med ggplot. Första vi kan bygga en scatterplot av sambandet

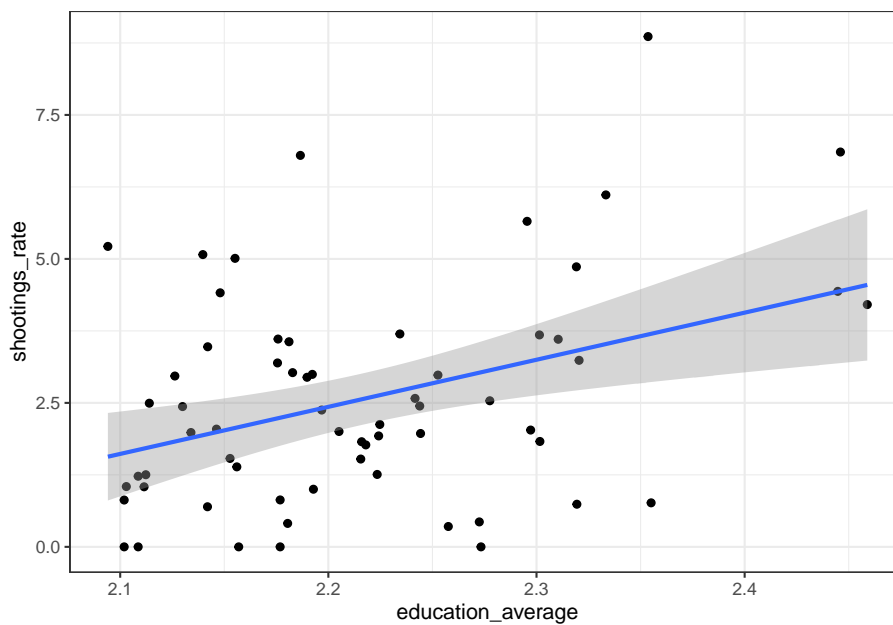
```
shootings %>%  
  ggplot(aes(education_average, shootings_rate))+  
  geom_point()+  
  theme_bw()
```



Sen vi kan lägga till en regressions linje

```
shootings %>%  
  ggplot(aes(education_average, shootings_rate))+  
  geom_point()+  
  geom_smooth(method= 'lm')+  
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



4.3 Jämförelsen av medelvärde: Kontinuerlig beroende variabel och kategorisk oberoende variabel

```
aov<-aov(shootings_rate~ year, data=shootings)
summary(aov)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## year      1   2.97   2.965    0.82  0.369
## Residuals 61 220.51   3.615
```

Sen vi får medelvärdena och konfidensintervaller men emmeans paketet

```
library(emmeans)
```

```
## Warning: package 'emmeans' was built under R version 4.4.2
```

```
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
```


4.3. JÄMFÖRELSEN AV MEDELVÄRDET: KONTINUERLIG BEROENDE VARAIBEL OCH KATEGORISK OB

```
emmeans(aov, spec = 'year')
```

```
## year emmean SE df lower.CL upper.CL
## 2019 2.56 0.24 61 2.08 3.04
##
## Confidence level used: 0.95
```

Sen vi kan visualisera sambandet mellan skottfrekvens och år.

```
shootings %>%
  ggplot(aes(year, shootings_rate))+
  geom_jitter(position=position_jitter(.05))+
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar",
    width = 0.1, position=position_nudge(x = 0.15)) +
  stat_summary(fun.y = mean, geom = "point",
    size = 3, position=position_nudge(x = 0.15))+
  theme_bw()
```

```
## Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
## i Please use the `fun` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

