Introdução e Estatística Descritiva

Fernando B. Sabino da Silva

Software

Rstudio

- Faça uma pasta no seu computador onde você deseja manter os arquivos para usar no **Rstudio**.
- Defina o diretório de trabalho nesta pasta: `Session -> Set Working Directory -> Choose Directory` (atalho: Ctrl+Shift+H).
- Torne a alteração permanente definindo o diretório padrão em: `Tools -> Global Options -> Choose Directory`.

R básico

- Cálculos simples:

```
4.6 * (2 + 3)^4
```

```
## [1] 2875
```

- Defina um objeto (escalar) e o imprima:

```
a <- 4
a
```

```
## [1] 4
```

- Defina um objeto (vetor) e o imprima:

```
b <- c(2, 5, 7)
b
```

```
## [1] 2 5 7
```

Extensões do **R**

- O **R** não precisa ser usado apenas como calculadora ou para atribuição de objetos simples. A sua funcionalidade pode ser estendida atráves de bibliotecas ou pacotes (muito similar a utilização de Plug-ins nos navegadores ou baixar aplicativos no google play). Alguns já vem instalados (automaticamente, by default) no **R** e você precisa apenas carregá-los (como fazemos depois que baixamos um aplicativo no celular e queremos usá-lo, por exemplo).
- Para instalar um novo pacote no **Rstudio** você pode usar o menu: `Tools -> Install Packages`
- Você precisa saber o nome do pacote que deseja instalar. Você também pode fazê-lo através do comando install.packages como abaixo:

```
install.packages("mosaic")
```

- Uma vez que o pacote esteja instalado, você pode carregá-lo através do comando `library` (ou `require`):

Ajuda do **R**

- Você pode receber ajuda (help) via `?<command>`:

```
?sum
```

- Procurando por ajuda:

```
help.search("plot")
```

- Você pode encontrar um cheat sheet com funções do **R** que usaremos neste curso aqui. Clique em 'Raw' ou 'View Raw' e abra ('Open') o arquivo.
- Você pode salvar os comandos que você porventura tenha digitado em um arquivo para uso posterior:
  - Selecione o guia History no painel superior direito no **Rstudio** .
  - Marque os comandos que você deseja salvar.
  - Pressione o botão `To Source`.
- Pratique as suas habilidades básicas em: http://tryr.codeschool.com

Data

Data example

- Data: Magazine Ads Readability
- Thirty magazines were ranked by educational level of their readers.
- Three magazines were **randomly** selected from each of the following groups:
  - Group 1: highest educational level
  - Group 2: medium educational level
  - Group 3: lowest educational level.
- Six advertisements were **randomly** selected from each of the following nine selected magazines:
  - Group 1: [1] Scientific American, [2] Fortune, [3] The New Yorker
  - Group 2: [4] Sports Illustrated, [5] Newsweek, [6] People
  - Group 3: [7] National Enquirer, [8] Grit, [9] True Confessions
- So, the data contains information about a total of 54 advertisements.

Data example (continued) - variables and format

- For each advertisement (54 cases), the data below were observed.
- **Variable names**:
  - WDS = number of words in advertisement
  - SEN = number of sentences in advertisement
  - 3SYL = number of 3+ syllable words in advertisement
  - MAG = magazine (1 through 9 as above)
  - GROUP = educational level (1 through 3 as above)
- Take a look at the data from within **Rstudio**:

```
magAds <- read.delim("https://asta.math.aau.dk/datasets?fil
head(magAds)
```

```
##   WDS SEN X3SYL MAG GROUP
## 1 205   9    34   1     1
## 2 203  20    21   1     1
## 3 229  18    37   1     1
## 4 208  16    31   1     1
```

Data types

Quantitative variables

- The measurements have numerical values.
- Quantative data often comes about in one of the following ways:
  - **Continuous variables**: measurements of e.g. waiting times in a queue, revenue, share prices, etc.
  - **Discrete variables**: counts of e.g. words in a text, hits on a webpage, number of arrivals to a queue in one hour, etc.
- Measurements like this have a well-defined scale and in **R** they are stored as the type **numeric**.
- It is important to be able to distinguish between discrete count variables and continuous variables, since this often determines how we describe the uncertainty of a measurement.

Categorical/qualitative variables

- The measurement is one of a set of given categories, e.g. sex

Population and sample

Aim of statistics

- Statistics is all about "saying something" about a population.
- Typically, this is done by taking a random sample from the population.
- The sample is then analysed and a statement about the population can be made.
- The process of making conclusions about a population from analysing a sample is called **statistical inference**.

Selecting **randomly**

- For the magazine data:
  - First we select **randomly** 3 magazines from each group.
  - Then we select **randomly** 6 ads from each magazine.
  - An important detail is that the selection is done completely at **random**, i.e.
    - each magazine within a group have an equal chance of being chosen and
    - each ad within a magazine have an equal chance of being chosen.
- In the following it is a fundamental requirement that the data collection respects this principle of randomness and in this case we use the term **sample**.
- More generally:
  - We have a **population** of objects.
  - We choose completely at random n of these objects, and from the jth object we get the measurement y_j, $j = 1, 2, \ldots, n$.
  - The measurements $y_1, y_2, \ldots, y_n$ are then called a **sample**.
- If we e.g. are measuring the water quality 4 times in a year then it is a bad idea to only collect data in fair weather. The

Variable grouping and frequency tables

Binning

- The function cut will divide the range of a numeric variable in a number of equally sized intervals, and record which interval each observation belongs to. E.g. for the variable X3SYL (the number of words with more than three syllables) in the magazine data:

```
# Before 'cutting':
magAds$X3SYL[1:5]
```

```
## [1] 34 21 37 31 10
```

```
# After 'cutting' into 4 intervals:
syll <- cut(magAds$X3SYL, 4)
syll[1:5]
```

```
## [1] (32.2,43]     (10.8,21.5]   (32.2,43]     (21.5,32.2
## Levels: (-0.043,10.8] (10.8,21.5] (21.5,32.2] (32.2,43]
```

Tables

- To summarize the results we can use the function `tally` from the `mosaic` package (remember the package **must be loaded** via `library(mosaic)` if you did not do so yet):

```
tally( ~ syll, data = magAds)
```

```
## syll
##  few some many lots
##   26   14   10    4
```

  - In percent:

```
tally( ~ syll, data = magAds, format = "percent")
```

```
## syll
##  few some many lots
## 48.1 25.9 18.5  7.4
```

2 factors: Cross tabulation

- To make a table of all combinations of two factors we use `tally` again:

```
tally( ~ syll + GROUP, data = magAds)
```

```
##        GROUP
## syll     1  2  3
##   few    8 11  7
##   some   4  2  8
##   many   3  5  2
##   lots   3  0  1
```

- Relative frequencies (in percent) columnwise:

```
tally( ~ syll | GROUP, data = magAds, format = "percent")
```

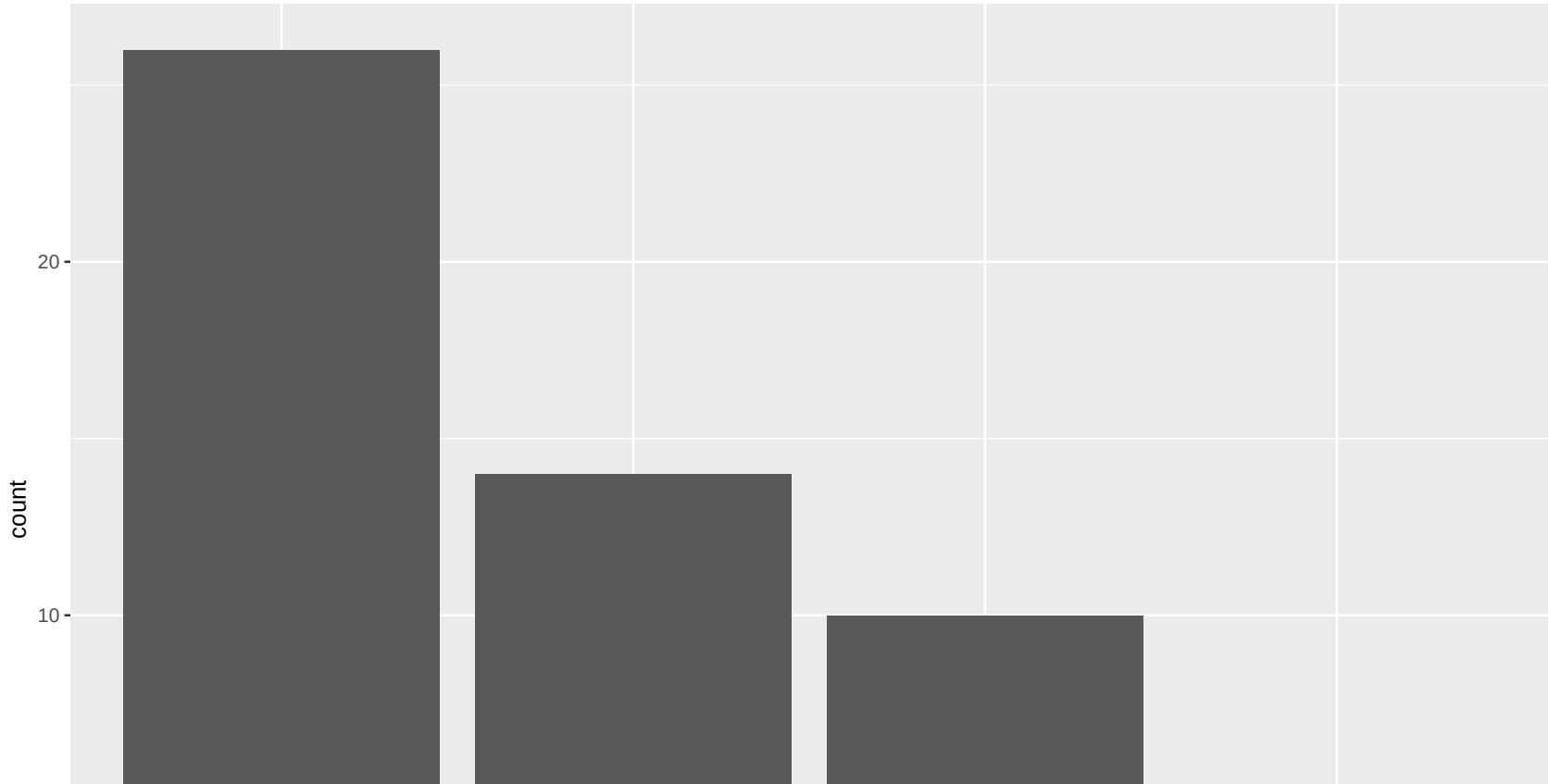
```
##        GROUP
## syll       1    2    3
```

Graphics

Bar graph

- To create a bar graph plot of table data we use the function `gf_bar` from `mosaic`. For each level of the factor a box is drawn with the height proportional to the frequency (count) of the level.

```
gf_bar( ~ syll, data = magAds)
```



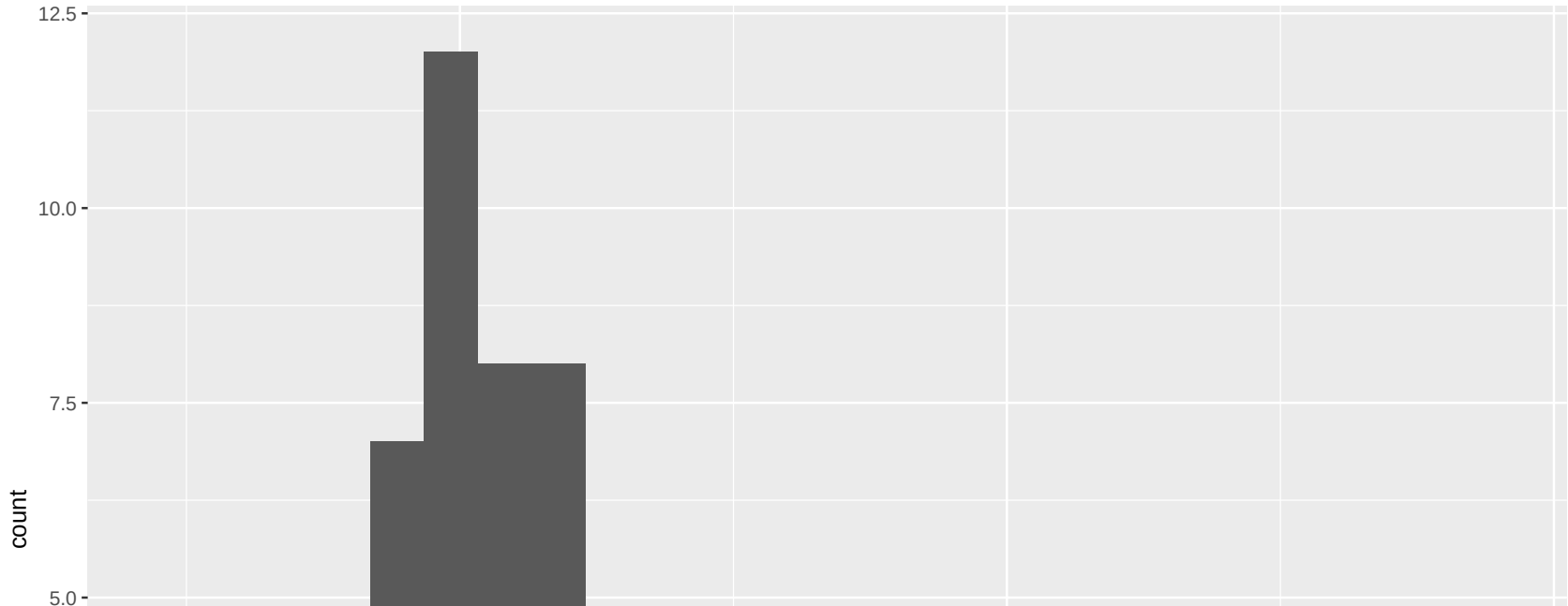
The Ericksen data

- Description of data: Ericksen 1980 U.S. Census Undercount.
- This data contains the following variables:
  - `minority`: Percentage black or Hispanic.
  - `crime`: Rate of serious crimes per 1000 individuals in the population.
  - `poverty`: Percentage poor.
  - `language`: Percentage having difficulty speaking or writing English.
  - `highschool`: Percentage aged 25 or older who had not finished highschool.
  - `housing`: Percentage of housing in small, multiunit buildings.
  - `city`: A factor with levels: `city` (major city) and `state` (state or state-remainder).
  - `conventional`: Percentage of households counted by conventional personal enumeration.
  - `undercount`: Preliminary estimate of percentage undercount.
- The Ericksen data has 66 rows/observations and 9 columns/variables.
- The observations are measured in 16 large cities, the remaining

Histogram (quantitative variables)

- How to make a histogram for some variable `x`:
  - Divide the interval from the minimum value of `x` to the maximum value of `x` in an appropriate number of equal sized sub-intervals.
  - Draw a box over each sub-interval with the height being proportional to the number of observations in the sub-interval.
- Histogram of crime rates for the Ericksen data

```
gf_histogram( ~ crime, data = Ericksen)
```



Summary of quantitative variables

Measures of center of data: Mean and median

- We return to the magazine ads example (`WDS` = number of words in advertisement). A number of numerical summaries for `WDS` can be retrieved using the `favstats` function:

```
favstats( ~ WDS, data = magAds)
```

```
##  min Q1 median  Q3 max mean sd  n missing
##   31 69     96 202 230  123 66 54       0
```

- The observed values of the variable `WDS` are $y_1 = 205$, $y_2 = 203, \ldots, y_n = 208$, where there are a total of $n = 54$ values. As previously defined this constitutes a **sample**.
- **mean** $= 123$ is the **average** of the sample, which is calculated by

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

We may also call $\bar{y}$ the **(empirical) mean** or the **sample mean**.

Measures of variability of data: range, standard deviation and variance

- The **range** is the difference of the largest and smallest observation.
- The **(empirical) variance** is the average of the squared deviations from the mean:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2.$$

- **sd** = **standard deviation** $= s = \sqrt{s^2}$.
- Note: If the observations are measured in meter, the **variance** has unit meter^2 which is hard to interpret. The **standard deviation** on the other hand has the same unit as the observations.
- The standard deviation describes how much data varies around the (empirical) mean.

Calculation of mean, median and standard deviation using R

The mean, median and standard deviation are just some of the summaries that can be read of the `favstats` output (shown on previous page). They may also be calculated separately in the following way:

- Mean of `WDS`:

```
mean( ~ WDS, data = magAds)
```

```
## [1] 123
```

- Median of `WDS`:

```
median( ~ WDS, data = magAds)
```

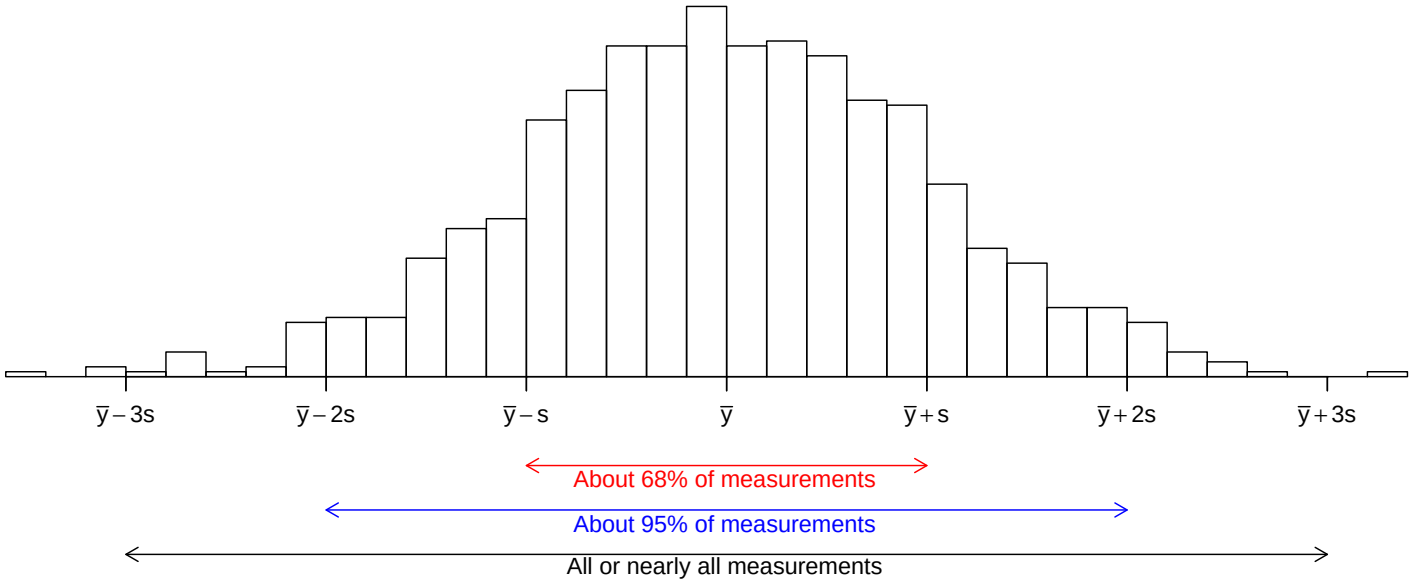
```
## [1] 96
```

- Standard deviation for `WDS`:

A word about terminology

- **Standard deviation**: a measure of variability of a population or a sample.
- **Standard error**: a measure of variability of an estimate. For example, a measure of variability of the sample mean.

The empirical rule



If the histogram of the sample looks like a bell shaped curve, then

- about 68% of the observations lie between $\bar{y} - s$ and $\bar{y} + s$.
- about 95% of the observations lie between $\bar{y} - 2s$ and $\bar{y} + 2s$.
- All or almost all (99.7%) of the observations lie between $\bar{y} - 3s$ and $\bar{y} + 3s$.

Percentiles

- **The pth percentile** is a value such that about $p\%$ of the population (or sample) lies below or at this value and about $(100 - p)\%$ of the population (or sample) lies above it.

Percentile calculation for a sample:

- First, sort data in increasing order. For the `WDS` variable in the magazine data:

$$y_{(1)} = 31, y_{(2)} = 32, y_{(3)} = 34, \ldots, y_{(n)} = 230.$$

  Here the number of observations is $n = 54$.
- Find the 5th percentile (i. e. $p = 5$):
  * The observation number corresponding to the 5-percentile is $N = \frac{n \cdot p}{100} = 2.7$.
    - That is, the 5-percentile must lie between the observations $x_{(k)} = 32$ and $x_{(k+1)} = 34$, where $k = 2 < N < 3$.
    - Let $d = N - k = 0.7$. One of several methods for estimating the 5-percentile:

Median, quartiles and interquartile range

Recall

```
favstats( ~ WDS, data = magAds)
```

```
##  min Q1 median  Q3 max mean sd  n missing
##   31 69     96 202 230  123 66 54       0
```

- 50-percentile = 96 is the **median** and it is a measure of the center of data.
- 0-percentile = 31 is the **minimum** value.
- 25-percentile = 69 is called the **lower quartile** (Q1). Median of lower 50% of data.
- 75-percentile = 201.5 is called the **upper quartile** (Q3). Median of upper 50% of data.
- 100-percentile = 230 is the **maximum** value.
- **Interquartile Range (IQR)**: a measure of variability given by the difference of the upper and lower quartiles: 201.5 - 69 = 132.5.

More graphics

Box-and-whiskers plots (or simply box plots)

How to draw a box-and-whiskers plot:

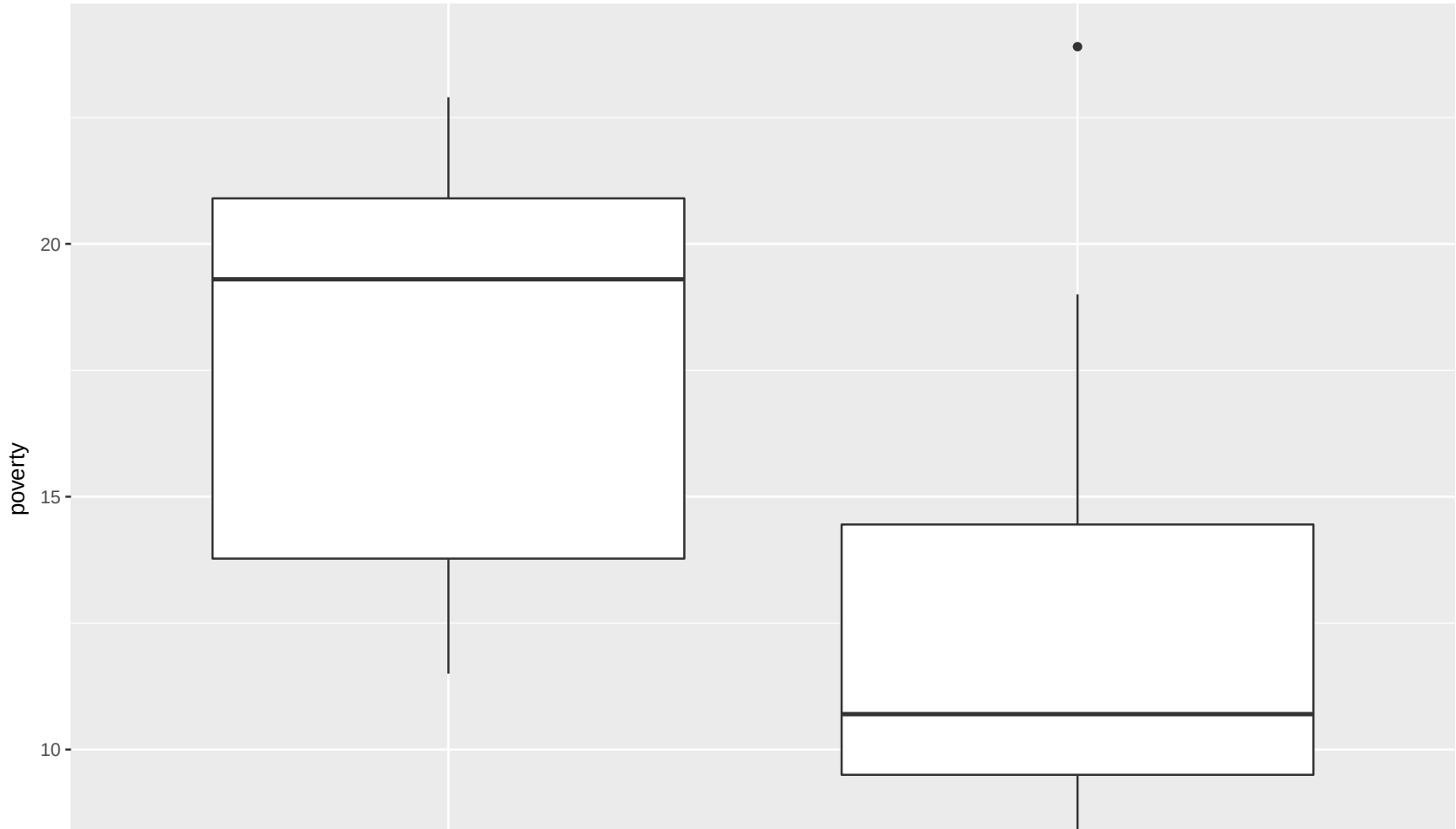
- Box:
  - Calculate the median, lower and upper quartiles.
  - Plot a line by the median and draw a box between the upper and lower quartiles.
- Whiskers:
  - Calculate interquartile range and call it IQR.
  - Calculate the following values:
    - L = lower quartile - 1.5*IQR
    - U = upper quartile + 1.5*IQR
  - Draw a line from lower quartile to the smallest measurement, which is larger than L.
  - Similarly, draw a line from upper quartile to the largest measurement which is smaller than U.
- Outliers: Measurements smaller than L or larger than U are drawn as circles.

Note: Whiskers are minimum and maximum of the observations that are not deemed to be outliers.

Boxplot for Ericksen data

Boxplot of the poverty rates separately for cities and states (variable `city`):

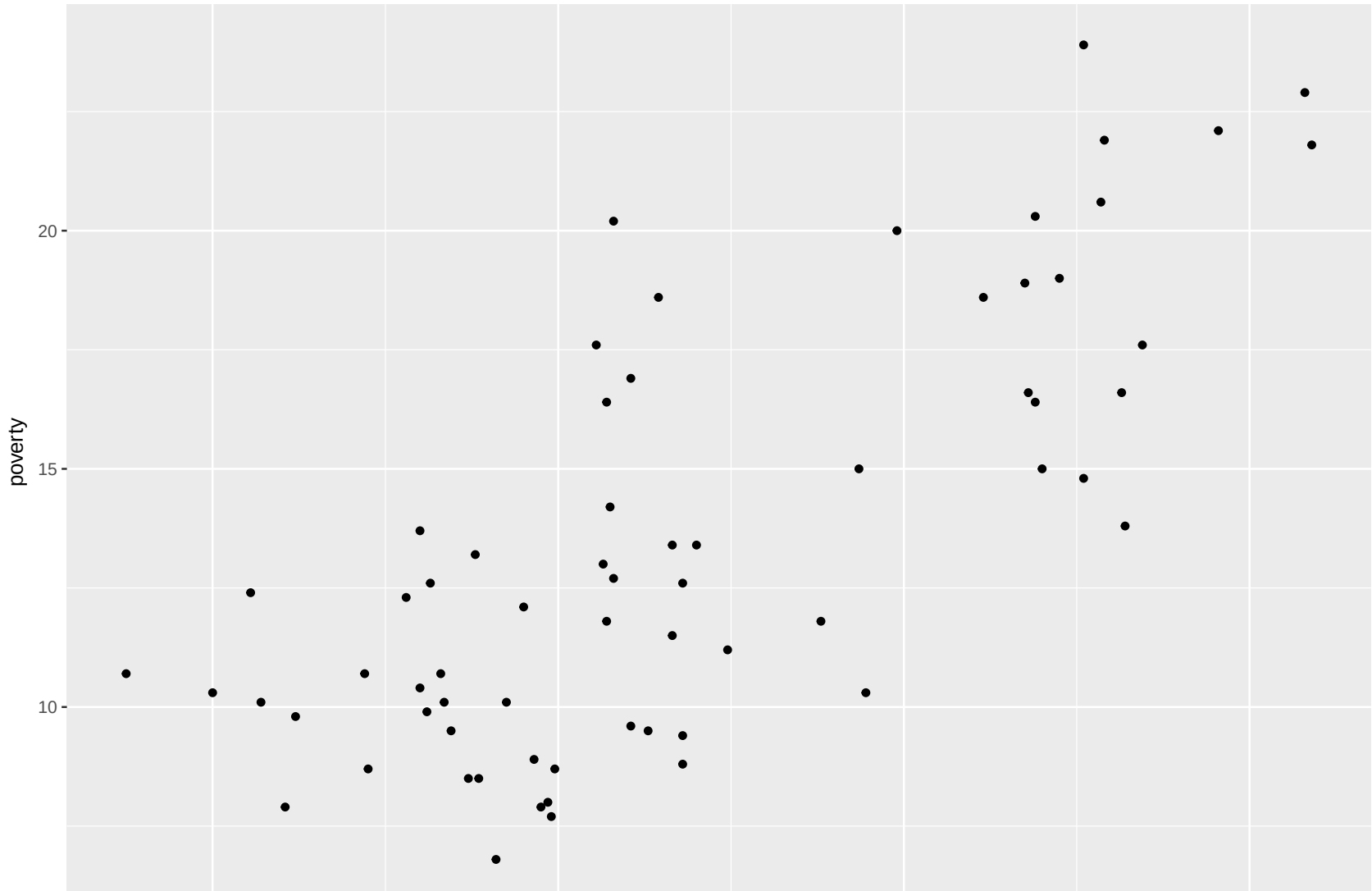
```
gf_boxplot(poverty ~ city, data = Ericksen)
```



2 quantitative variables: Scatter plot

For two quantitative variables the usual graphic is a scatter plot:

```
gf_point(poverty ~ highschool, data = Ericksen)
```



Appendix

Recoding variables

- The function `factor` will directly convert a vector to be of type `factor`. E.g.:

```
head(magAds$GROUP)
```

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

```
f <- factor(magAds$GROUP)
magAds$GROUP <- f
head(magAds$GROUP)
```

```
## [1] 1 1 1 1 1 1
## Levels: 1 2 3
```

- Custom labels for the levels can also be used:

```
f <- factor(magAds$GROUP,
            levels = c("1", "2", "3"),
            labels = c("high", "medium", "low"))
```

Point and click plotting

mplot

- If `mosaic` is loaded and the package `manipulate` is installed you can construct plots using point and click using the function `mplot`.
- You simply use `mplot` on your dataset and answer the question and then you can change things by pressing the settings button (cog wheel) in the top left of the plot window.

```
mplot(Ericksen)
```

- In the end you can press "Show expression" to get the code for the plot.