Introduction to R and descriptive statistics

The ASTA team

Contents

1 Introduction to R

1.1 Rstudio

- Make a folder on your computer where you want to keep files to use in **Rstudio**. **Do NOT use Danish characters æ, ø, å** in the folder name (or anywhere in the path to the folder).

- Set the working directory to this folder: `Session -> Set Working Directory -> Choose Directory` (shortcut: Ctrl+Shift+H).
- Make the change permanent by setting the default directory in: `Tools -> Global Options -> Choose Directory`.

1.2 R basics

- Ordinary calculations:

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

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

- Make a (scalar) object and print it:

```
a <- 4
a
```

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

- Make a (vector) object and print it:

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

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

- Make a sequence of numbers and print it:

```
s <- 1:4
s
```

```
## [1] 1 2 3 4
```

- Note: A more flexible command for sequences:

```
s <- seq(1, 4, by = 1)
```

- **R** does elementwise calculations:

```
a * b
```

```
## [1]  8 20 28
```

```
a + b
```

```
## [1]  6  9 11
```

```
b ^ 2
```

```
## [1]  4 25 49
```

- Sum and product of elements:

```
sum(b)
```

```
## [1] 14
```

```
prod(b)
```

```
## [1] 70
```

1.3 R markdown

- The slides and all exercises in R (including the exam questions) are made in the special Rmarkdown format.
- This allows you to combine text and R code.
- You can write formulas using standard LaTeX commands.

1.4 R extensions

- The functionality of **R** can be extended through libraries or packages (much like plugins in browsers etc.). Some are installed by default in **R** and you just need to load them.
- To install a new package in **Rstudio** use the menu: `Tools -> Install Packages`
- You need to know the name of the package you want to install. You can also do it through a command:

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

- When it is installed you can load it through the `library` command:

```
library(mosaic)
```

- This loads the `mosaic` package which has a lot of convenient functions for this course (we will get back to that later). It also prints a lot of info about functions that have been changed by the `mosaic` package, but you can safely ignore that.

1.5 R help

- You get help via `?<command>`:

```
?sum
```

- Use `tab` to make **Rstudio** guess what you have started typing.
- Search for help:

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

- You can find a cheat sheet with the **R** functions we use for this course here.
- Save your commands in a file for later usage:
  - Select history tab in top right pane in **Rstudio** .
  - Mark the commands you want to save.
  - Press `To Source` button.

2 Data in R

2.1 Data example

- Now we will have a look at a data set concerning the 1988 vote in Chile for or against Pinochet to continue as leader. The sample consists of 2700 voters randomly selected from the Chilenean population.
- The data set contains the variables:
  - `region`: The region in Chile where the voter lives
  - `population`: Population of the region.

  - `sex`: The gender of the voter.
  - `age`: The age of the voter.
  - `education`: Education level of the voter.
  - `income`: Monthly income of the voter.
  - `statusquo`: To which degree the voter supports the status quo.
  - `vote`: Should Pinochet continue? `Y` = yes, `N`= no, `U`=undecided, `A`= will abstain from voting.

- More information about the data set may be found here.

```
Chile <- read.delim("https://asta.math.aau.dk/datasets?file=Chile.txt")
head(Chile)
```

```
##   region population sex age education income statusquo vote
## 1      N     175000   M  65         P  35000       1.0    Y
## 2      N     175000   M  29        PS   7500      -1.3    N
## 3      N     175000   F  38         P  15000       1.2    Y
## 4      N     175000   F  49         P  35000      -1.0    N
## 5      N     175000   F  23         S  35000      -1.1    N
## 6      N     175000   F  28         P   7500      -1.0    N
```

2.2 Data types

2.2.1 Quantitative variables

- The measurements have numerical values.
- Quantative data often comes about in one of the following ways:
  - **Continuous variables**: measurements of e.g. speed, temperature, etc.
  - **Discrete variables**: counts of e.g. number of household members, hits on a webpage, cars passing on a road 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.

2.2.2 Categorical/qualitative variables

- The measurement is one of a set of given categories, e.g. sex (male/female), education level, satisfaction score (low/medium/high), etc.
- Factors have two so-called scales:
  - **Nominal scale**: There is no natural ordering of the factor levels, e.g. sex and hair color.
  - **Ordinal scale**: There is a natural ordering of the factor levels, e.g. education level and satisfaction score.
- The measurement is usually stored (which is also recommended) as a **factor** in **R**. The possible categories are called **levels**. Example: the levels of the factor "sex" is male/female. A factor in **R** can have a so-called **attribute** assigned, which tells if it is ordinal.

2.3 Variables in the data set

```
head(Chile)
```

```
##   region population sex age education income statusquo vote
## 1      N     175000   M  65         P  35000       1.0    Y
## 2      N     175000   M  29        PS   7500      -1.3    N
## 3      N     175000   F  38         P  15000       1.2    Y
```

```
## 4      N     175000   F  49         P  35000      -1.0    N
## 5      N     175000   F  23         S  35000      -1.1    N
## 6      N     175000   F  28         P   7500      -1.0    N
```

- Quantitative variables in the `Chile` data set:
  - `population`, `age`, `income`, `statusquo`
- Categorical variables:
  - `region`, `sex`, `education`, `vote`
- All the categorical variables are nominal except `education`, which has three ordered categories (primary, secondary, post-secondary).

3 Descriptive statistics of categorical data

3.1 Tables

- To summarize the the variable `vote` 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( ~ vote, data = Chile)
```

```
## vote
##    A    N    U    Y <NA>
##  187  889  588  868  168
```

- In percent:

```
tally( ~ vote, data = Chile, format = "percent")
```

```
## vote
##    A    N    U    Y <NA>
##  6.9 32.9 21.8 32.1  6.2
```

- Here we use an **R** `formula` (characterized by the "tilde" sign `~`) to indicate that we want this variable from the dataset `Chile` (without the tilde it would look for a global variable called `vote` and use that rather than the one in the dataset).

3.2 2 factors: Cross tabulation

- To get an overview over the relation between two categorical variables, we can make a cross tabulation.
- To make a table of all combinations of the two factors `vote` and `sex`, we use `tally` again:

```
tally( ~ vote + sex, data = Chile)
```

```
##       sex
## vote     F   M
##   A    104  83
##   N    363 526
##   U    362 226
##   Y    480 388
##   <NA>  70  98
```

- We can also get the relative frequencies (in percent) columnwise:

```
tally( ~ vote | sex, data = Chile, format = "percent")
```

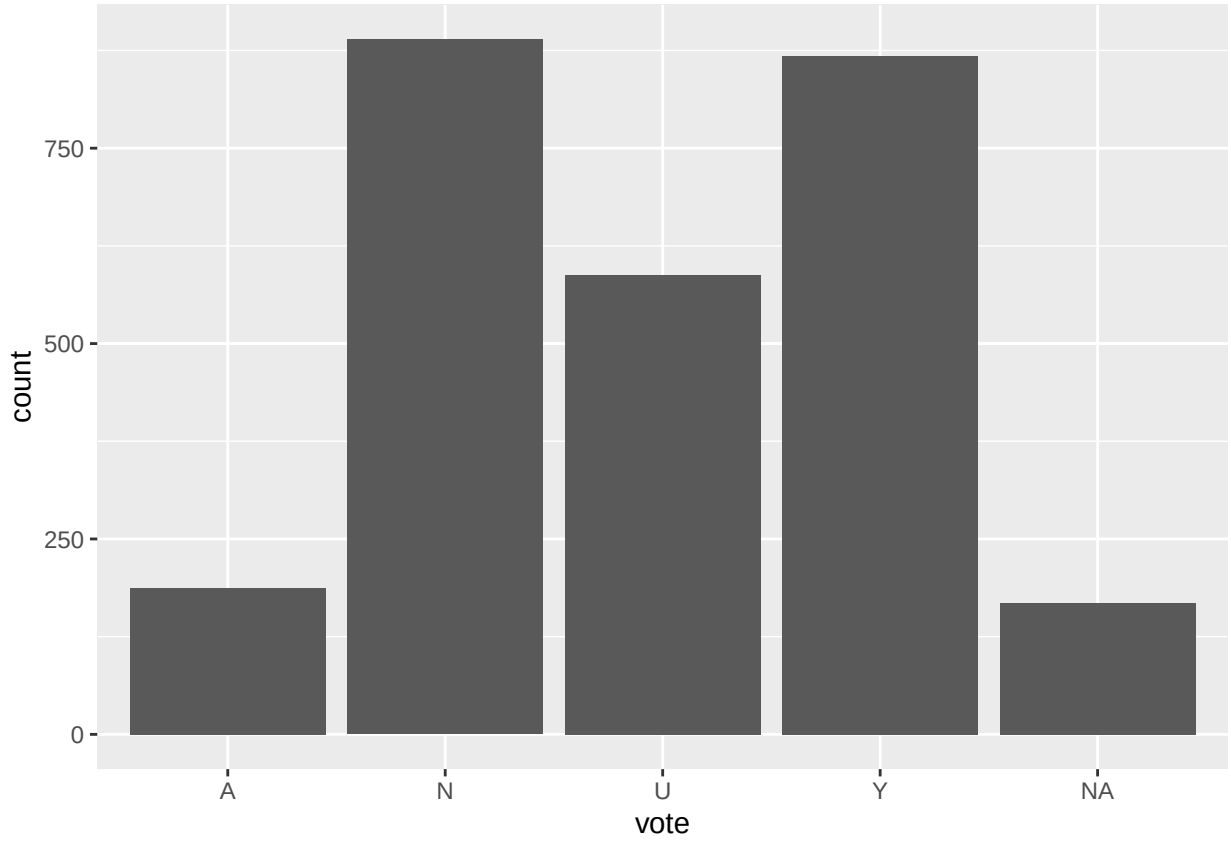
```
##       sex
## vote      F    M
##   A     7.5  6.3
##   N    26.3 39.8
##   U    26.3 17.1
##   Y    34.8 29.4
##   <NA>  5.1  7.4
```

- For instance we see that 34.8% of the women said they would vote yes, while this holds for only 29.4% of the men.

3.3 Visualizing categorical data: Bar graph

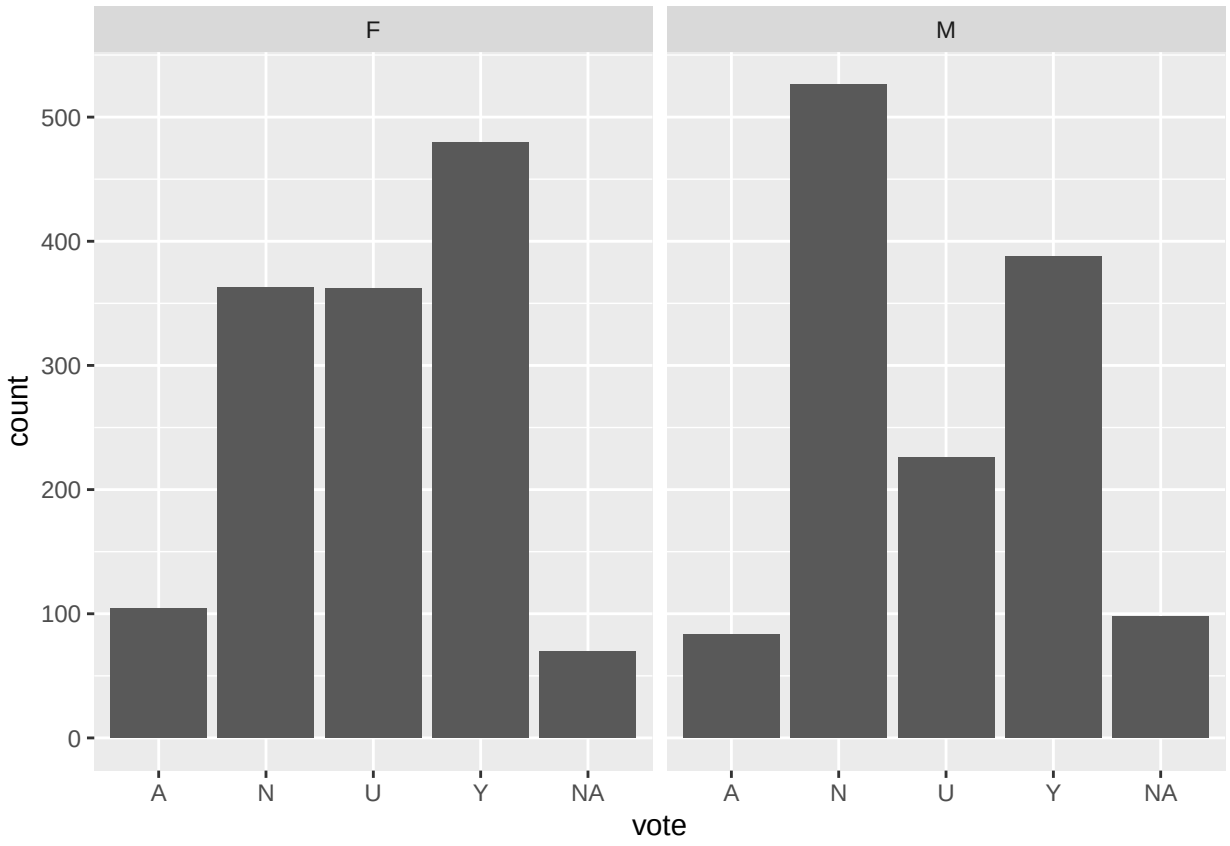
- To create a bar graph plot of table data we use the function `gf_bar` from `mosaic`. For each level of the factor, a box is drawn with the height proportional to the frequency (count) of the level.

```
gf_bar( ~ vote, data = Chile)
```



- The bar graph can also be split by group:

```
gf_bar( ~ vote | sex, data = Chile)
```



4 Descriptive statistics of quantitative variables

4.1 Data example: Fuel consumption of cars

- In this data set, a car magazine tested the fuel consumption of 32 cars. The variable `mpg` gives the fuel consumption in miles pr. gallon (the data set is from 1974).
- The data set is built into **R** under the name `mtcars`, so it does not need to be loaded before use.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat  wt qsec vs am gear carb
## Mazda RX4          21   6  160 110  3.9 2.6   16  0  1    4    4
## Mazda RX4 Wag      21   6  160 110  3.9 2.9   17  0  1    4    4
## Datsun 710         23   4  108  93  3.9 2.3   19  1  1    4    1
## Hornet 4 Drive     21   6  258 110  3.1 3.2   19  1  0    3    1
## Hornet Sportabout  19   8  360 175  3.1 3.4   17  0  0    3    2
## Valiant            18   6  225 105  2.8 3.5   20  1  0    3    1
```

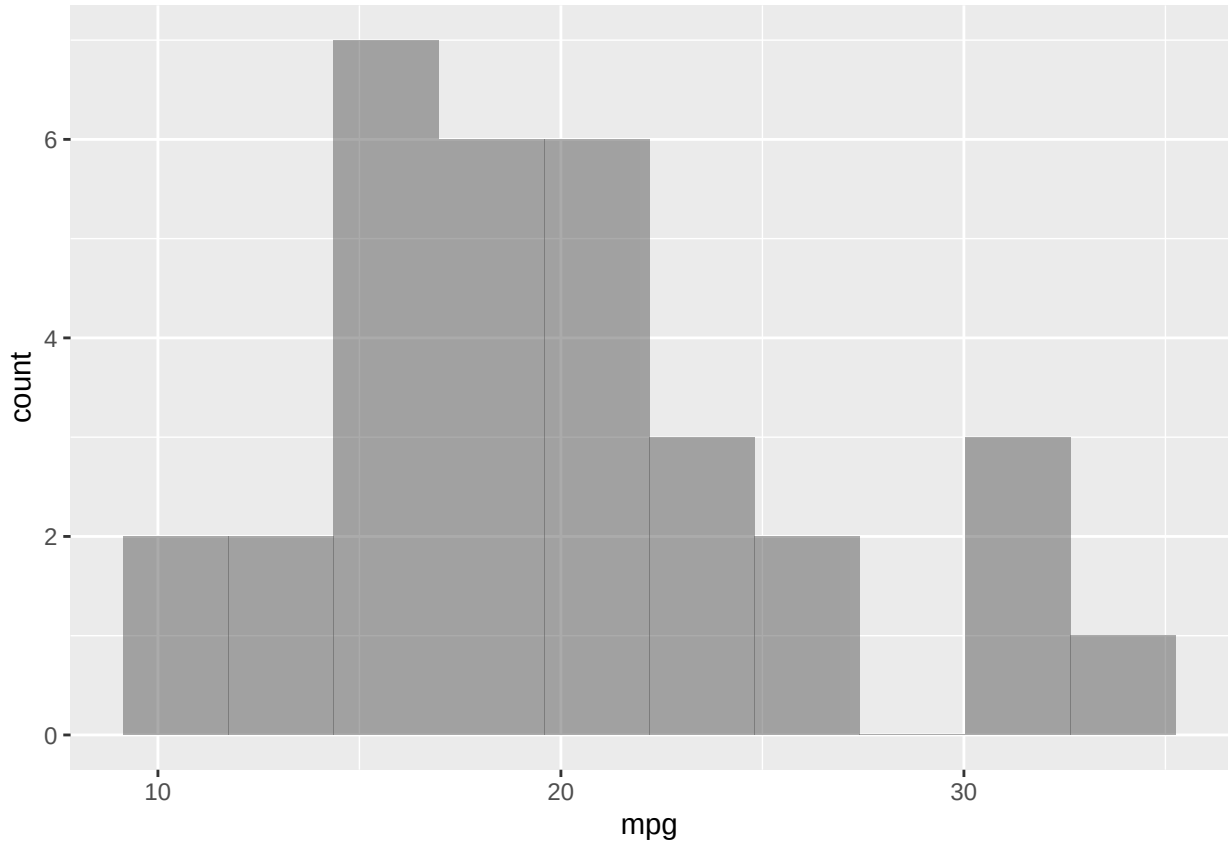
4.2 Visualizing quantitative data: Histogram

- The way to get a first impression of a quantitative variable is to draw a histogram.
- The histogram of a variable `x` is made as follows:
  - 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.

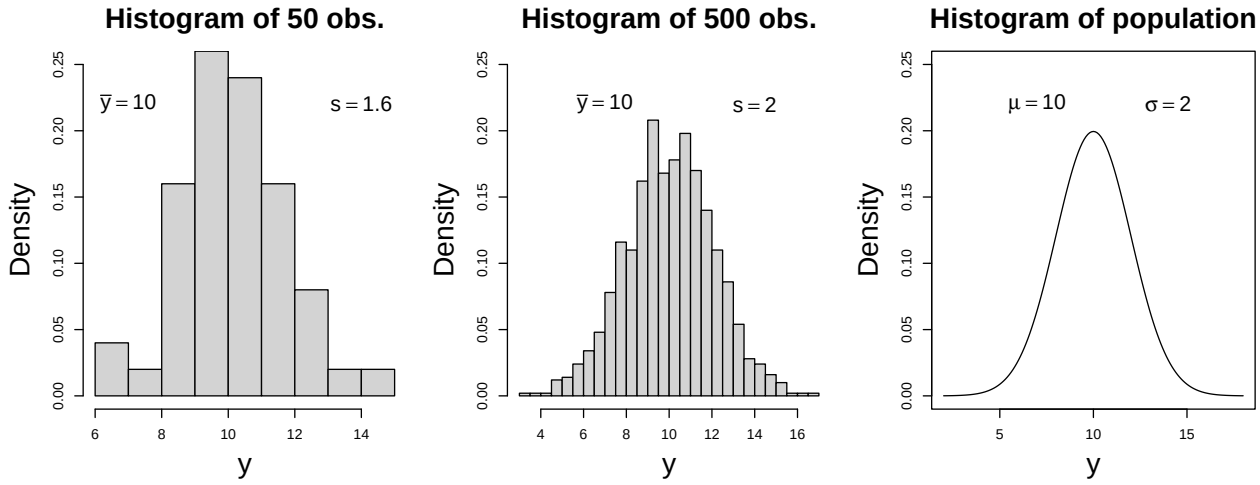
- Histogram of `mpg` for the `mtcars` data. The `bins` option sets the number of subintervals to 10.

```
gf_histogram( ~ mpg, data = mtcars,bins=10)
```



4.3 Relation between histogram and denity function

- Suppose a sample comes from a population having a continuous distribution with density function f.
- Draw a histogram where the y-axis is scaled such that the total area of the bars is 1.
- When the number of observations (the sample size) increases we can make a finer interval division and get a more smooth histogram.
- When the number of observations tends to infinity, we obtain a nice smooth curve, where the area below the curve is 1. This curve is exactly the probability density function f.



- If the histogram looks bell-shaped this may suggest a normal distribution.

4.4 Summary statistics for quantitative data

- We return to the `mtcars` example. A summary of the fuel consumption `mpg` can be retrieved using the `favstats` function:

```
favstats( ~ mpg, data = mtcars)
```

```
##  min Q1 median Q3 max mean sd  n missing
##   10 15     19 23  34   20  6 32       0
```

- The output contains the following information
  - **min** The minimal value in the sample is 10.4.
  - **max** The maximal value in the sample is 33.9.
  - **n** The sample size (number of observations) is 32.
  - **mean** The sample mean is 20.1. Recall that this was the average of all observations $x_1, \ldots, x_n$, i.e.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  - **sd** The sample standard deviation is 6.03. Recall that this was given by

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

  - **missing** There are no missing values.
  - **median** The median (or 50-percentile) is the value such that half of the sample has lower values than the median and half the sample has larger values.
  - **Q1** and **Q3** will be introduced on later slides.
- Both the mean and the median can be considered the center of a distribution. In a symmetric distribution (such as the normal distribution) they are equal, while in a skewed distribution, they tend to be different.

4.5 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:

- Sample size of `mpg`:

```
length(mtcars$mpg)
```

```
## [1] 32
```

- Mean of `mpg`:

```
mean( ~ mpg, data = mtcars)
```

```
## [1] 20
```

- Median of `mpg`:

```
median( ~ mpg, data = mtcars)
```

```
## [1] 19
```

- Standard deviation for `mpg`:

```
sd( ~ mpg, data = mtcars)
```

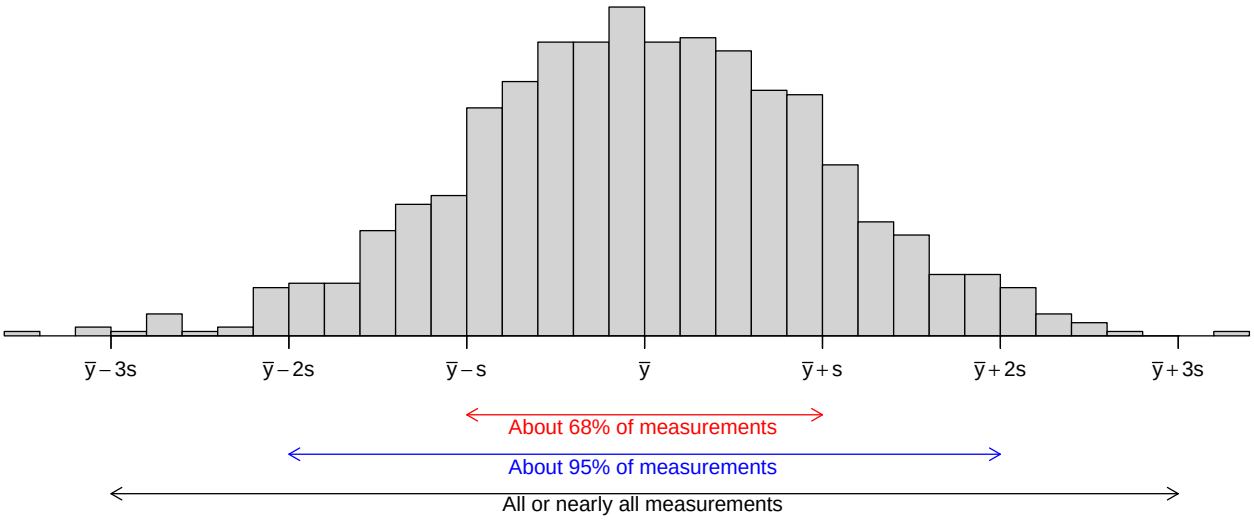
```
## [1] 6
```

- We may also calculate the summaries within groups. For instance, for each engine type (variable `vs`) the sample mean is:

```
mean( ~ mpg | factor(vs), data = mtcars)
```

```
##  0  1
## 17 25
```

4.6 Interpretation of summary statistics: The empirical rule



- If the histogram of the sample looks like a bell shaped curve, then we have
  - 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 $\bar{y} + 3s$.

4.7 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.

4.7.1 Percentile calculation for a sample:

- First, sort data from smallest to largest. For the `mpg` variable:

$$x_{(1)} = 10.4, x_{(2)} = 10.4, x_{(3)} = 13.3, \ldots, x_{(n)} = 33.9.$$

  Here the number of observations is $n = 32$.

- Find the 10th percentile (i. e. $p = 10$):
  - The observation number corresponding to the 10-percentile is $N = \frac{32 \cdot 10}{100} = 3.2$.
  - So the 10-percentile lies between the observations with observation number $k = 3$ and $k + 1 = 4$. That is, its value lies somewhere in the interval between $x_{(3)} = 13.3$ and $x_{(4)} = 14.3$.
  - One of several methods for estimating the 10-percentile from the value of N is defined as:

$$x_{(k)} + (N - k)(x_{(k+1)} - x_{(k)})$$

    which in this case gives

$$x_{(3)} + (3.2 - 3)(x_{(4)} - x_{(3)}) = 13.3 + 0.2 \cdot (14.3 - 13.3) = 13.5.$$

4.8 Median, quartiles and interquartile range

Recall

```
favstats( ~ mpg, data = mtcars)
```

```
##  min Q1 median Q3 max mean sd  n missing
##   10 15     19 23  34   20  6 32       0
```

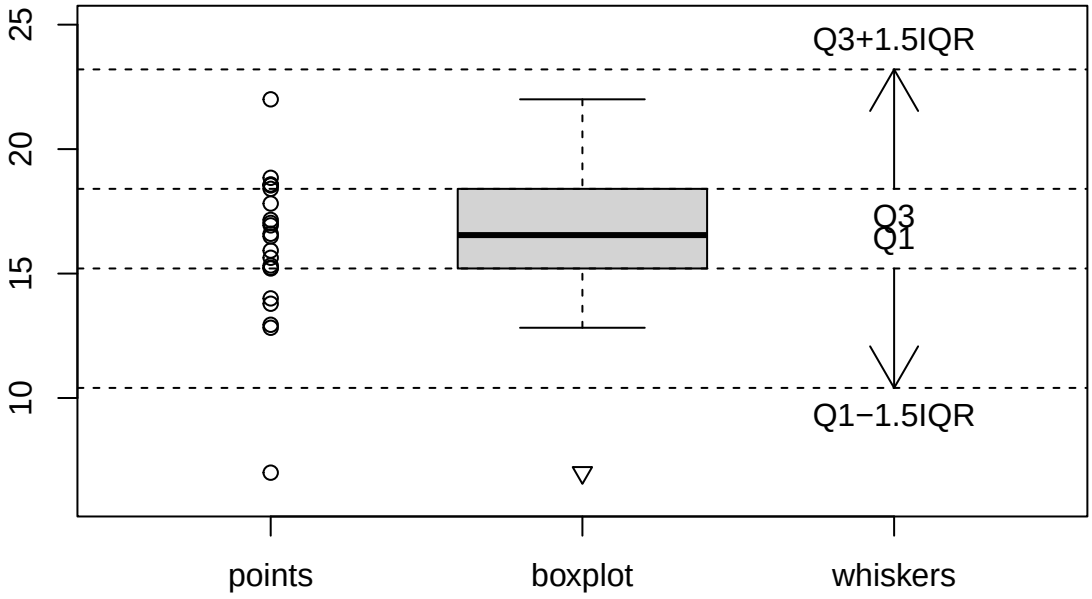
- 0-percentile = 10.4 is the **minimum** value.
- 50-percentile = 20.1 is the **median** and it is a measure of the center of data.
- 25-percentile = 15.4 is called the **lower quartile** (Q1). Median of lower 50% of data.
- 75-percentile = 22.8 is called the **upper quartile** (Q3). Median of upper 50% of data.
- 100-percentile = 33.9 is the **maximum** value.
- **Interquartile Range (IQR)**: a measure of variability given by the difference of the upper and lower quartiles: 23 -15 = 8.

4.9 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.



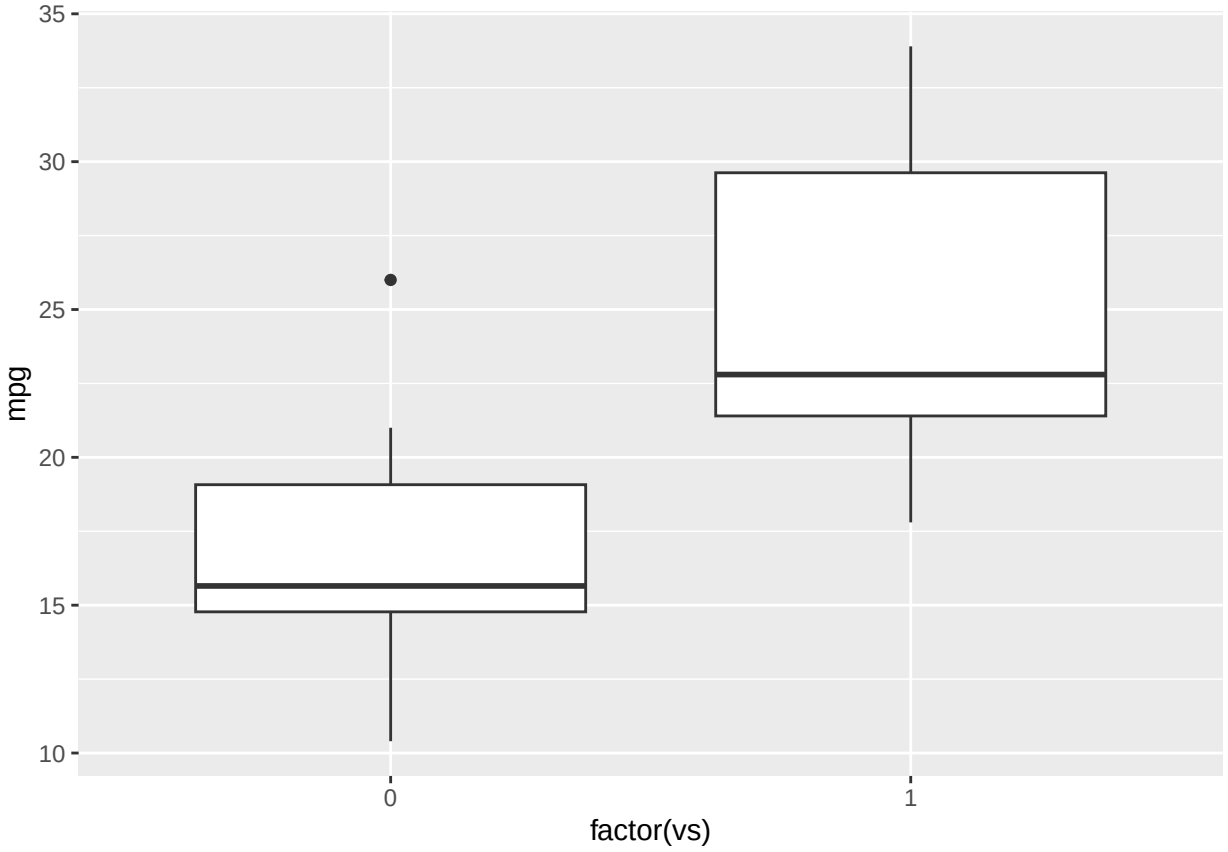
4.10 Boxplot for fuel consumption

- Boxplot of the fuel consumption separately for each engine type:

```
favstats(mpg ~ vs, data = mtcars)
```

```
##   vs min Q1 median Q3 max mean  sd  n missing
## 1  0  10 15     16 19  26   17 3.9 18       0
## 2  1  18 21     23 30  34   25 5.4 14       0
```

```
gf_boxplot(mpg ~ factor(vs), data = mtcars)
```

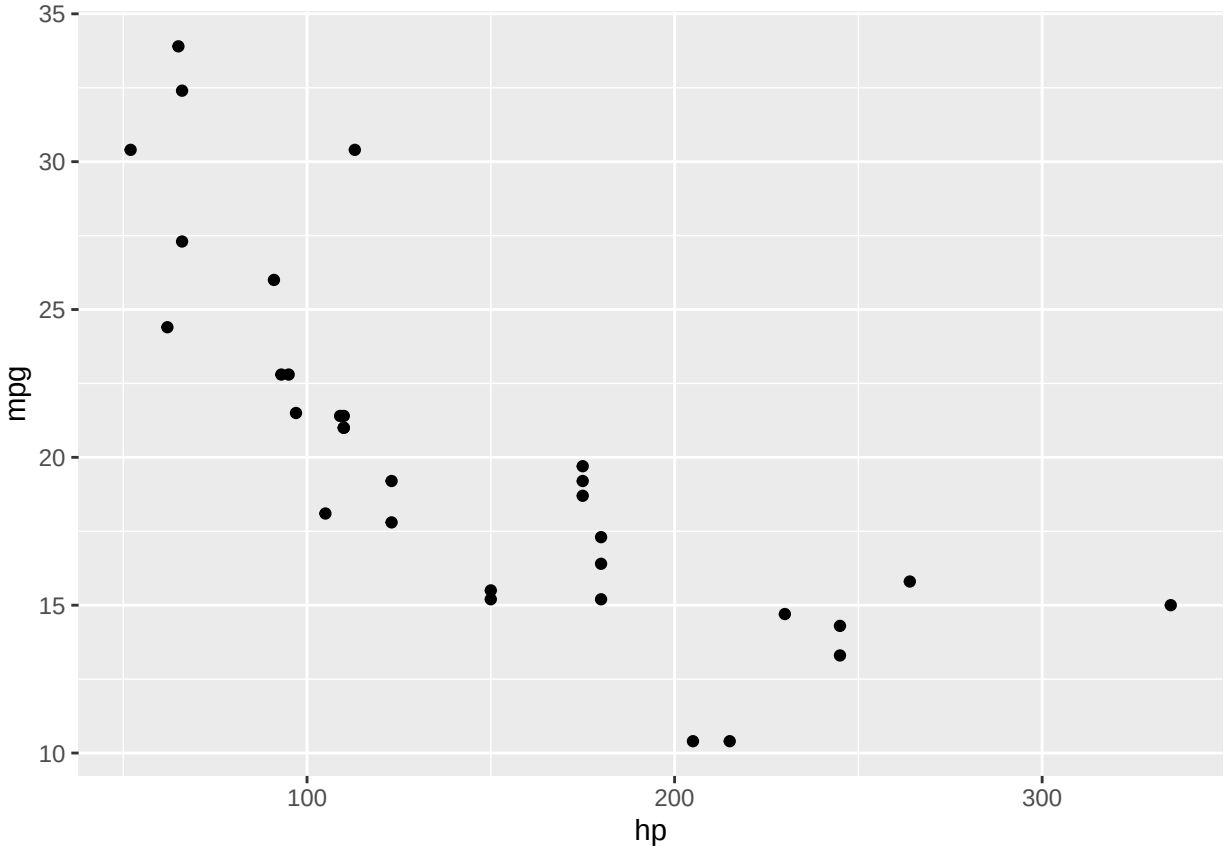


- Cars with engine type 1 seem to use more fuel.
- A single car with engine type 0 differs noticeably from the others with a high fuel consumption.

4.11 2 quantitative variables: Scatter plot

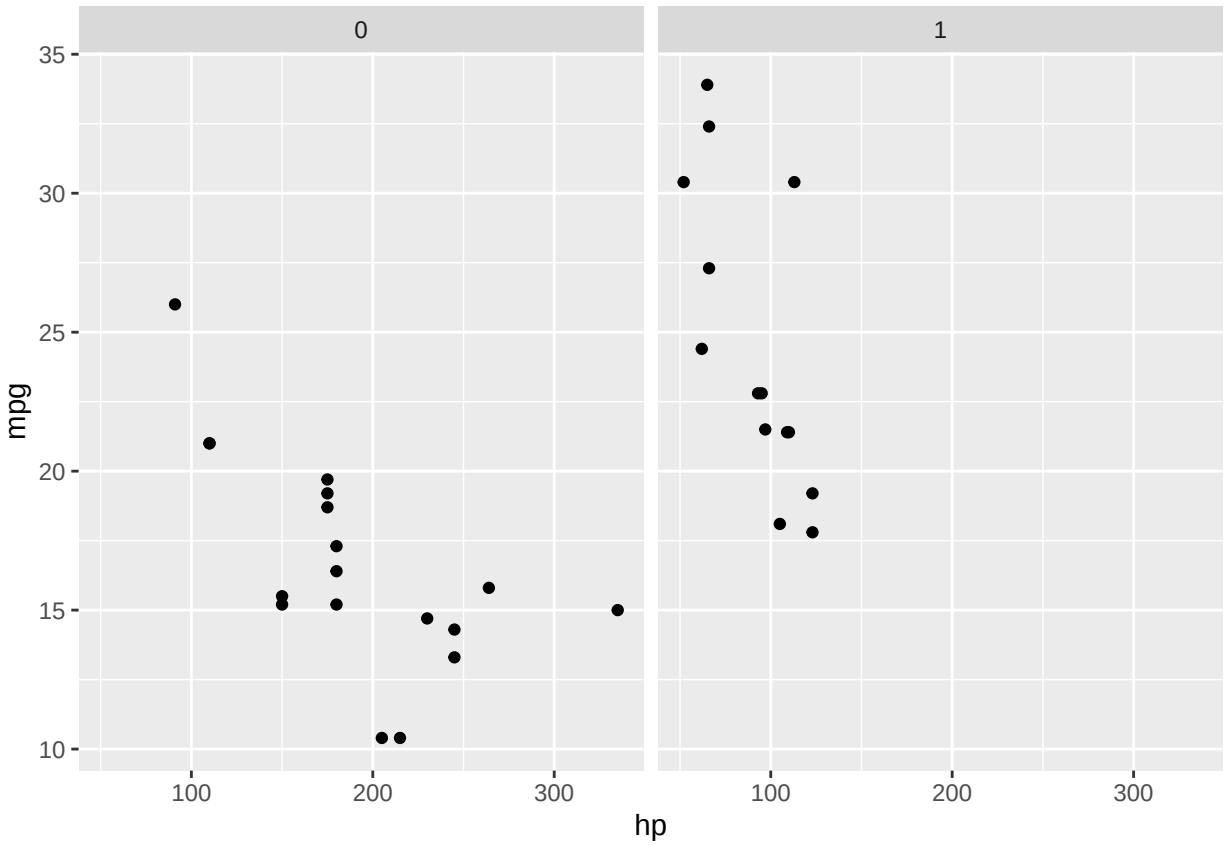
- A **scatter plot** is used to visualize two quantitative variables.
- For instance, we can plot the relation between fuel consumption and horse powers (`hp`) of a car as follows

```
gf_point(mpg ~ hp, data = mtcars)
```

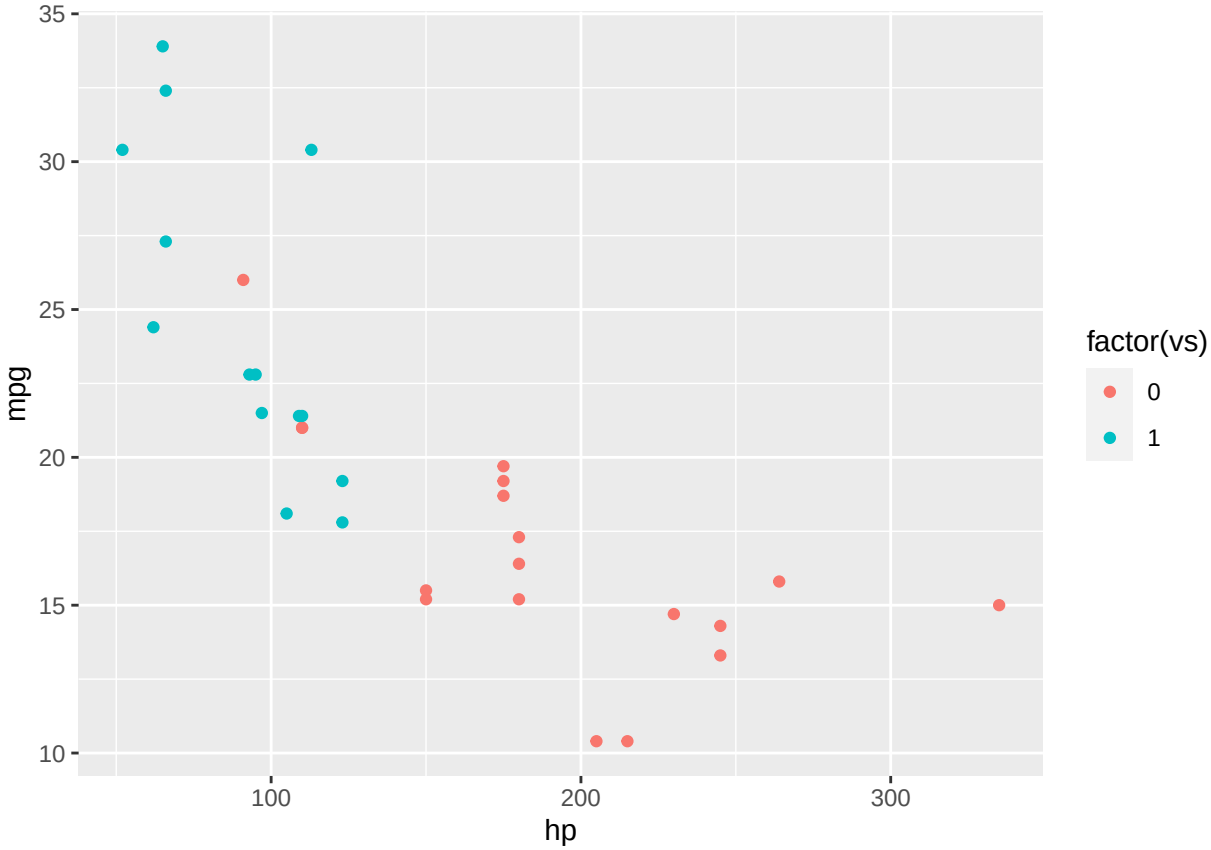


- This can be either split or coloured according to the engine type`vs`:

```
gf_point(mpg ~ hp | factor(vs), data = mtcars)
```

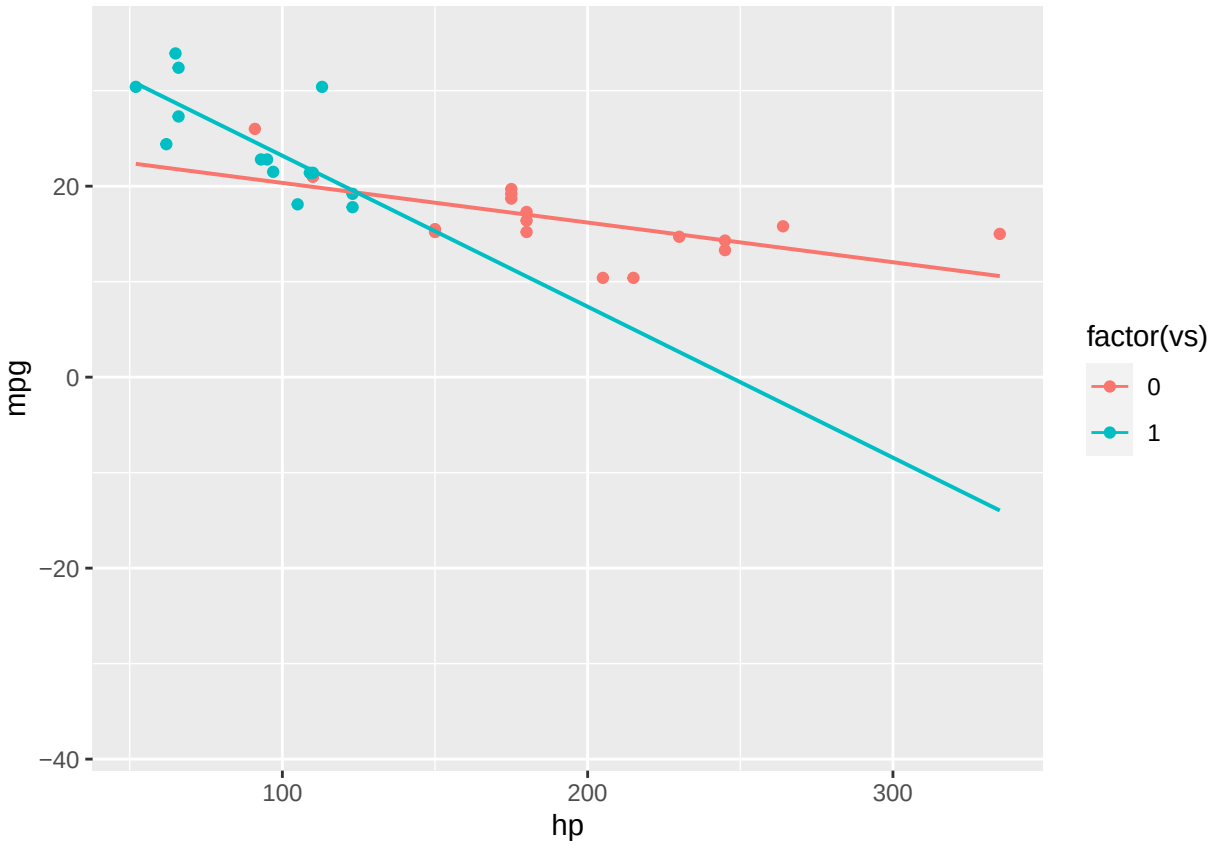



```
gf_point(mpg ~ hp, col = ~factor(vs), data = mtcars)
```



- If we want a regression line along with the points we can do:

```
gf_point(mpg ~ hp, col = ~factor(vs), data = mtcars) %>% gf_lm()
```



5 Quantile plots

5.1 The empirical quantiles

- Recall that the distribution function of a random variable X was defined as:

$$F(x) = P(X \leq x).$$

- The $\frac{i}{n}$ quantiles of a distribution are the points q_i such that $F(q_i) = \frac{i}{n}$, $i = 1, \ldots, n$.
- If we rank the observations in a sample

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)},$$

  we can approximate F at $x_{(i)}$ by:

$$\hat{F}(x_{(i)}) = \frac{i}{n}.$$

- Interpretation: $x_{(i)}$ is approximately the $\frac{i}{n}$ quantile.
- Note: various authors may use slightly different quantiles, e.g. $\frac{i-0.5}{n}$ quantiles.

5.2 Normal quantile-quantile plots

- The quantiles may be used to investigate whether the sample comes from a normal distribution.
- Call the $\frac{i}{n}$th quantile of a standard normal distribution q_i, i.e. $P(Z \leq q_i) = \frac{i}{n}$.

- If $Y \sim \texttt{norm}(\mu, \sigma)$, then this is equivalent to is equivalent to

$$P(Y \leq \mu + \sigma q_i) = \frac{i}{n}.$$

- Suppose the population follows a $\texttt{norm}(\mu, \sigma)$ distribution, then the sample quantiles $x_{(i)}$ should be approximately equal to the population quantiles $\mu + \sigma q_i$.
- If we make a scatter plot of the pair $(q_i, x_{(i)})$, these should lie on a straight line. We call this a **normal Q-Q plot** (or quantile-quantile plot).
  - **Example:** We investigate whether the `mpg` variable in the `mtcars` data set follows a normal distribution:

```
qqnorm(mtcars$mpg)
qqline(mtcars$mpg)
```