Practical 2

Emilia Löscher

27-9-2022

First, load the packages:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(magrittr)
```

```
##
## Attache Paket: 'magrittr'
```

```
## Das folgende Objekt ist maskiert 'package:purrr':
##
##     set_names
```

```
## Das folgende Objekt ist maskiert 'package:tidyr':
##
##     extract
```

```
library(mice)
```

```
## Warning: Paket 'mice' wurde unter R Version 4.1.3 erstellt
```

```
##
## Attache Paket: 'mice'
```

```
## Das folgende Objekt ist maskiert 'package:stats':
##
##     filter
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     cbind, rbind
```

```
library(DAAG)
```

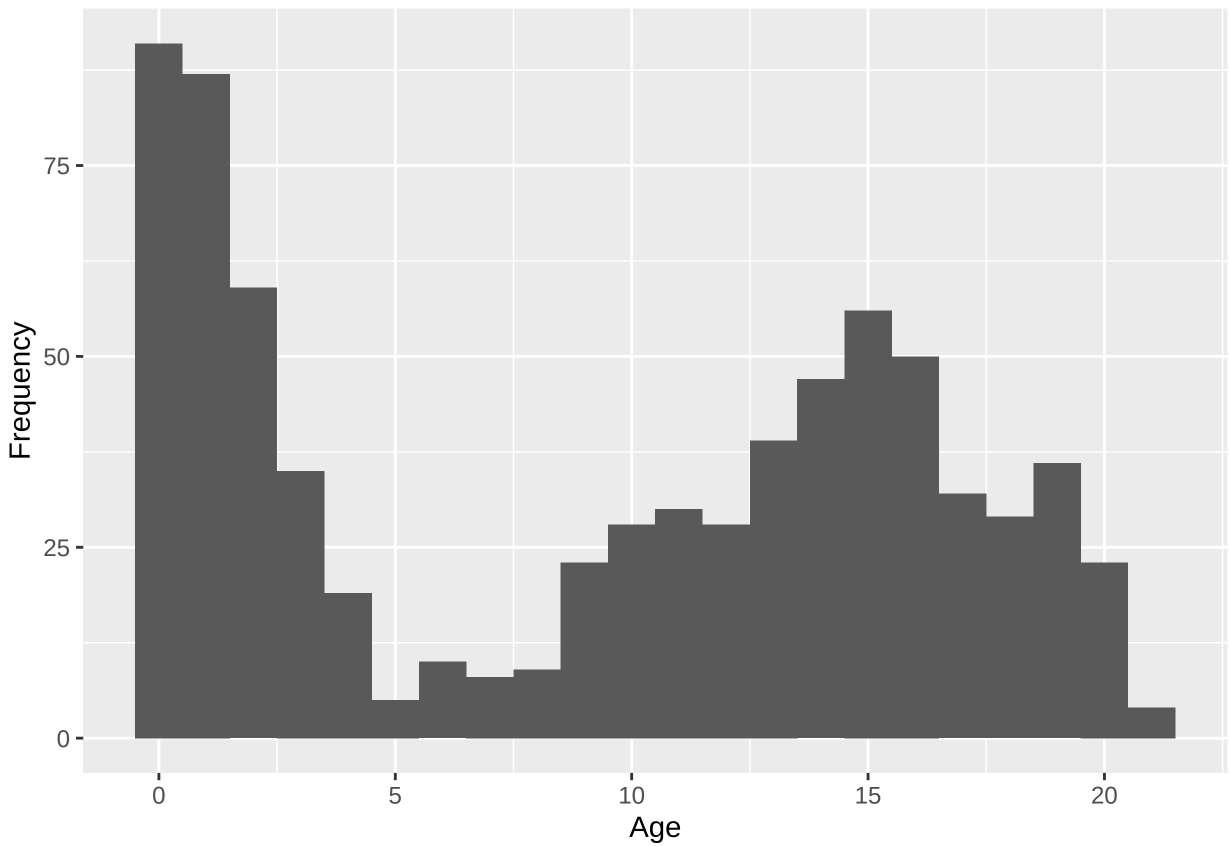
```
## Warning: Paket 'DAAG' wurde unter R Version 4.1.3 erstellt
```

```
library(ggplot2)
```

1.

Create a histogram of the variable age using the function geom_histogram().

```
ggplot(boys, aes(x = age)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Age", y = "Frequency")
```

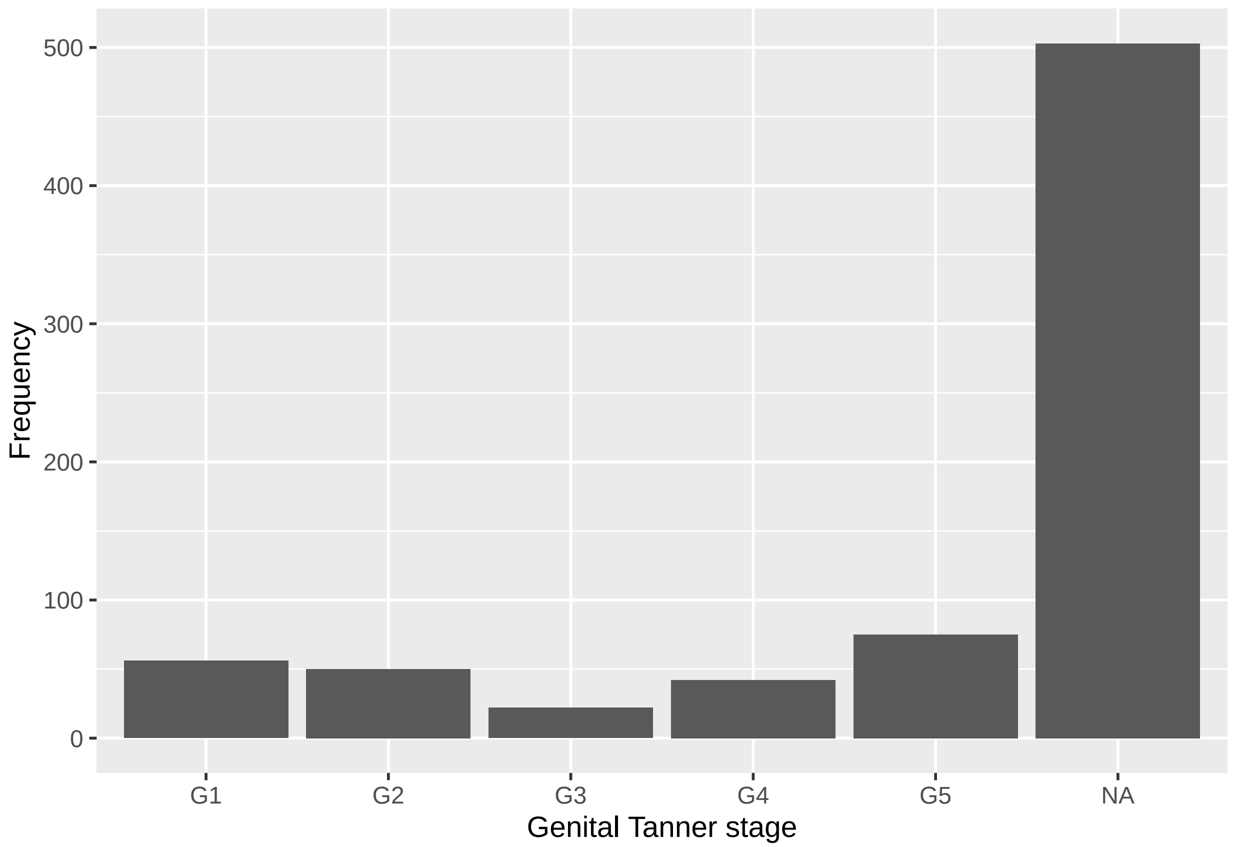


There are only few boys in the data set who are between 5 and 8 years old or older than 20. Most boys are aged 0, 1 or 2.

2.

Create a bar chart of the variable gen using the function geom_bar().

```r
ggplot(boys, aes(x = gen)) +
  geom_bar() +
  labs(x = "Genital Tanner stage", y = "Frequency")
```

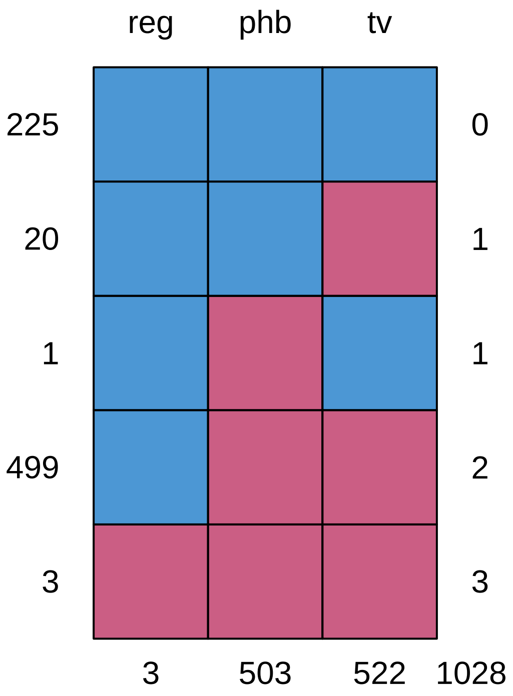


There are a lot of missing values on the gen variable. Most boys are in Genital Tanner stage 5.

3.

Create a missingness indicator for the variables gen, phb and tv.

```
md.pattern(boys[,7:9])
```



```
##      reg phb  tv     
## 225    1   1   1    0
## 20     1   1   0    1
## 1      1   0   1    1
## 499    1   0   0    2
## 3      0   0   0    3
##        3 503 522 1028
```

```
boys_na <- boys %>% mutate(gen_na = is.na(gen), phb_na = is.na(phb), tv_na = is.na(tv))
```

4.

Assess whether missingness in the variables gen, phb and tv is related to someones age.

```
boys_na %>% group_by(gen_na) %>% summarize(age = mean(age))
```

```
## # A tibble: 2 x 2
##   gen_na   age
##   <lgl>  <dbl>
## 1 FALSE  14.0
## 2 TRUE    6.79
```

```
boys_na %>% group_by(phb_na) %>% summarize(age = mean(age))
```

```
## # A tibble: 2 x 2
##   phb_na   age
##   <lgl>  <dbl>
## 1 FALSE  14.0
## 2 TRUE    6.79
```

```
boys_na %>% group_by(tv_na) %>% summarize(age = mean(age))
```

```
## # A tibble: 2 x 2
##   tv_na   age
##   <lgl> <dbl>
## 1 FALSE 14.1
## 2 TRUE   7.02
```

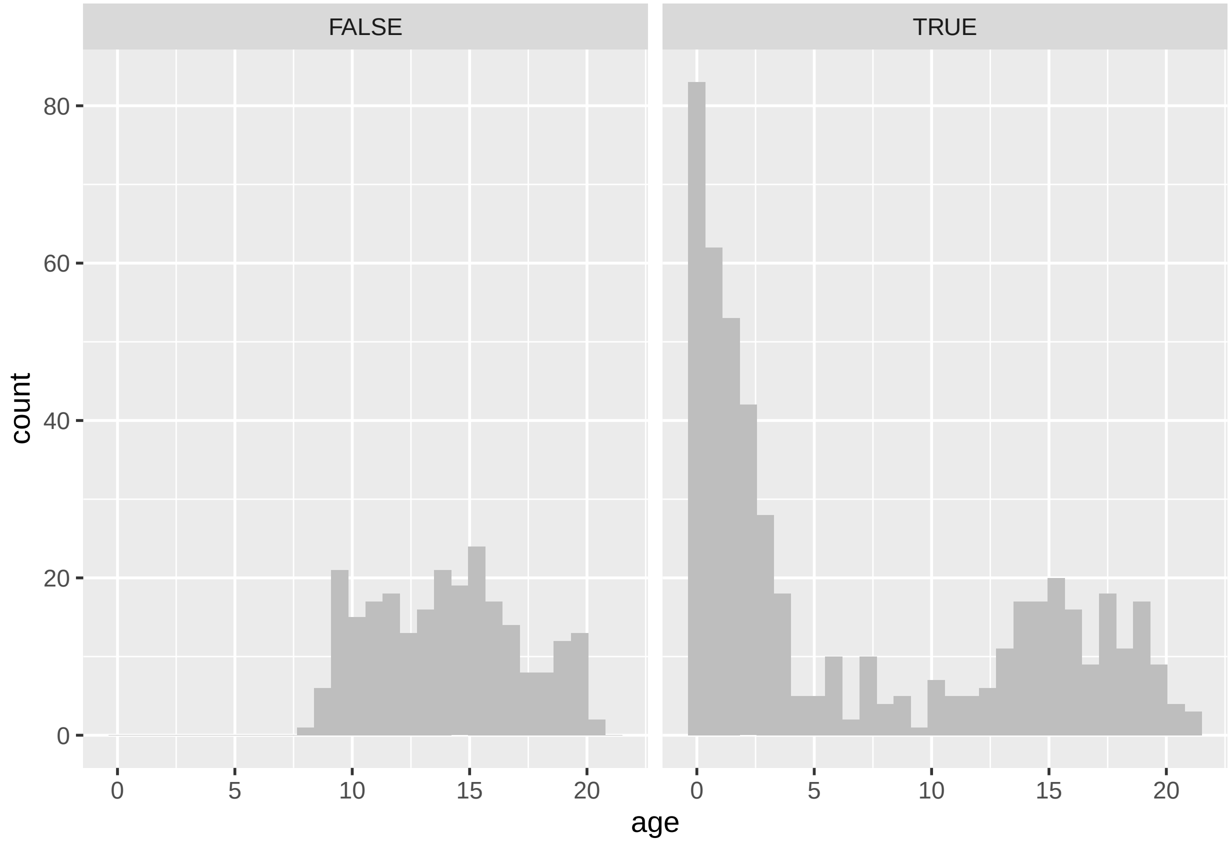
The average age of the boys with missingness on the three variables is lower than of the boys without missing values.

5.

Create a histogram for the variable age, faceted by whether or not someone has a missing value on gen.

```
ggplot(boys_na, aes(x = age)) +
  geom_histogram(fill = "grey") +
  facet_wrap(~gen_na)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



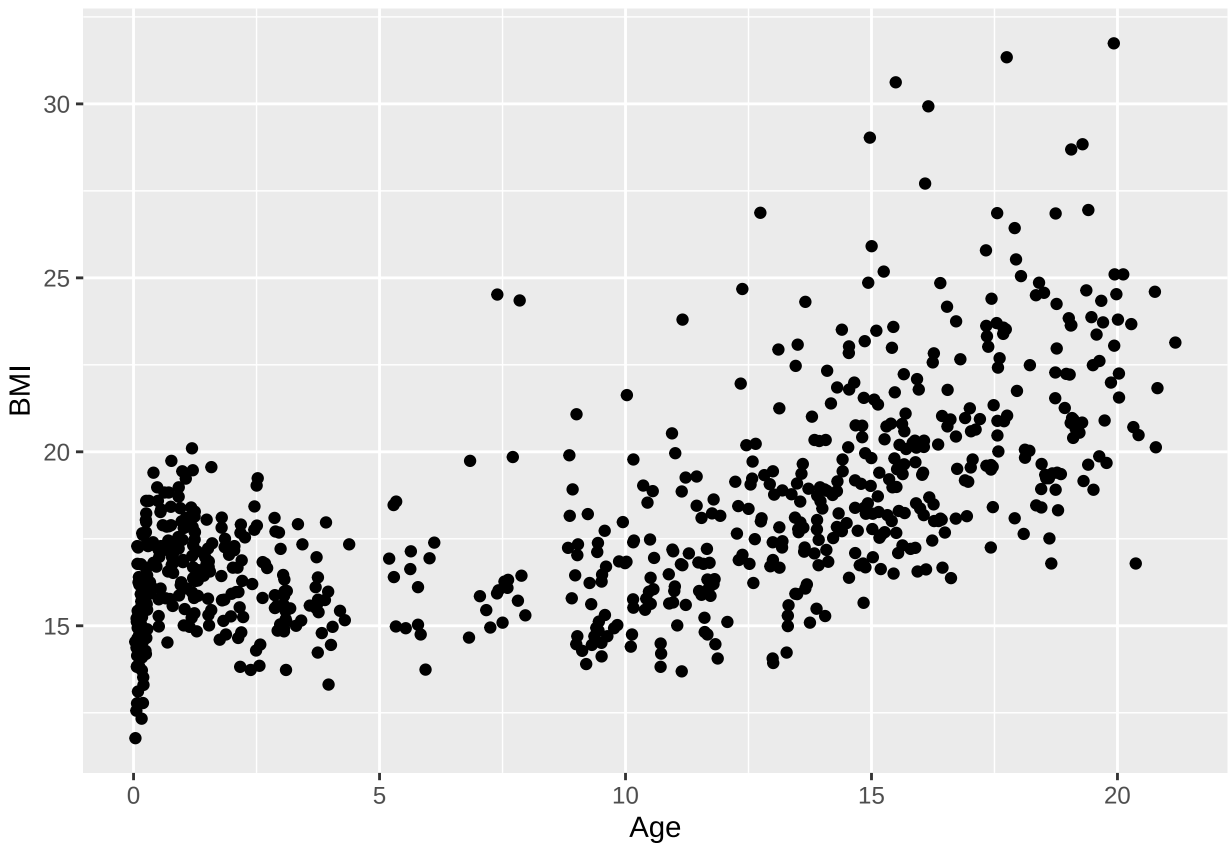
It can be seen that the Genital Tanner stage is only available for some of the boys aged between 8 and 21 and also not for every boy from that age group.

6.

Create a scatterplot with age on the x-axis and bmi on the y-axis, using the function geom_point().

```
ggplot(boys, aes(x = age, y = bmi)) +
  geom_point() +
  labs(x = "Age", y = "BMI")
```

```
## Warning: Removed 21 rows containing missing values (geom_point).
```



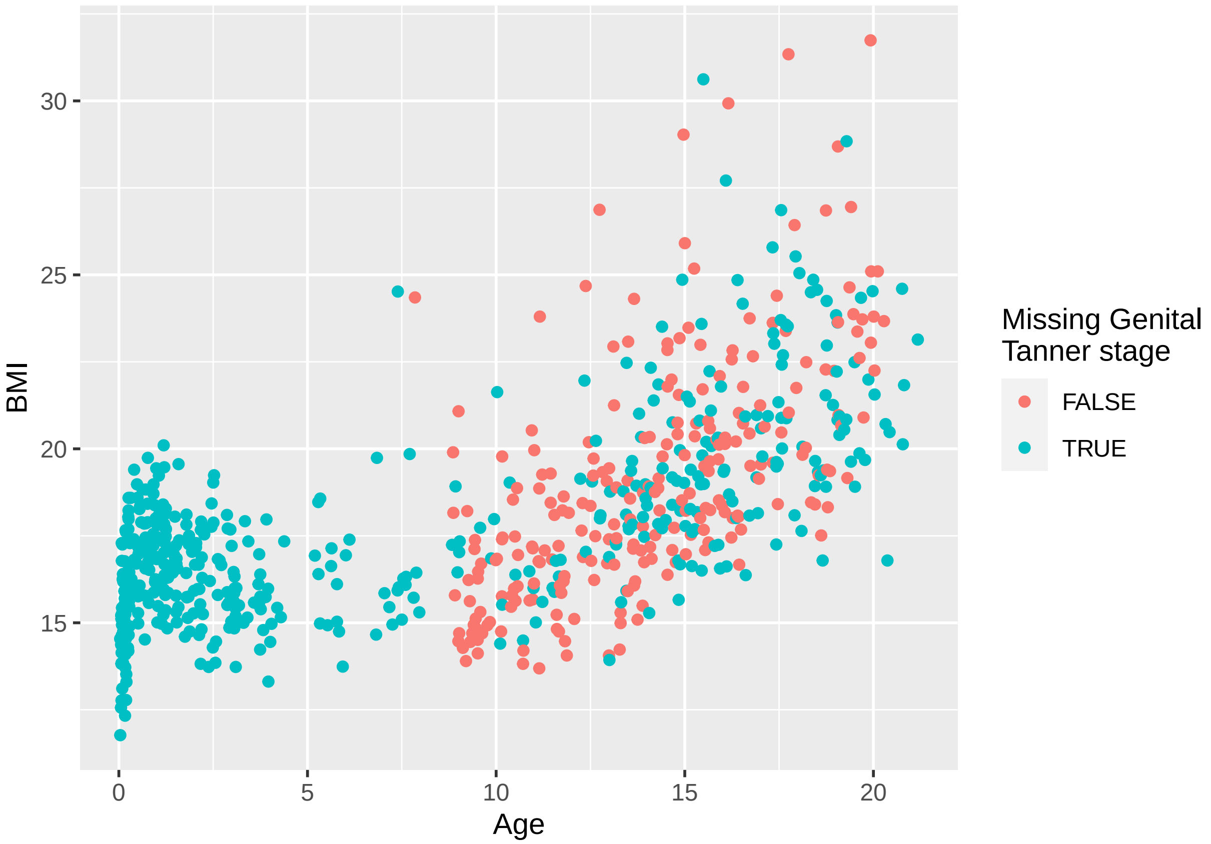
21 observations were removed due to missingness. It can be seen that bmi is increasing for higher age. Also, the variance in bmi is larger for older boys.

7.

Add a colour aesthetic to the previous plot using the missingness indicator of the variable gen.

```
ggplot(boys, aes(x = age, y = bmi, col = is.na(gen))) +
  geom_point() +
  labs(x = "Age", y = "BMI", col= "Missing Genital \nTanner stage")
```

```
## Warning: Removed 21 rows containing missing values (geom_point).
```

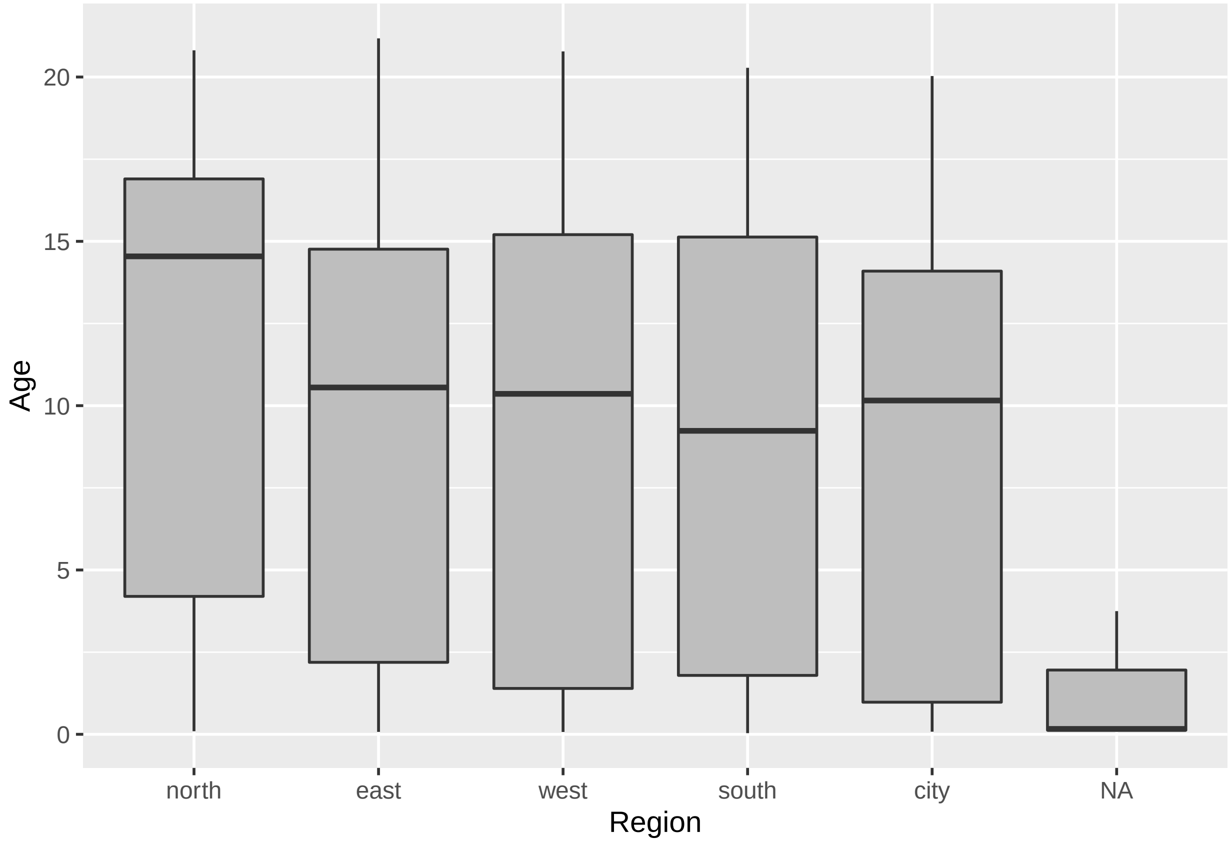


21 observations were removed due to missingness. It can be seen again that values for gen are missing for all boys under 7 and some boys between 7 and 22.

8.

Visualize the relationship between reg (region) and age using a boxplot.

```
ggplot(boys, aes(x = reg, y = age)) +
  geom_boxplot(fill = "grey") +
  labs(x = "Region", y = "Age")
```

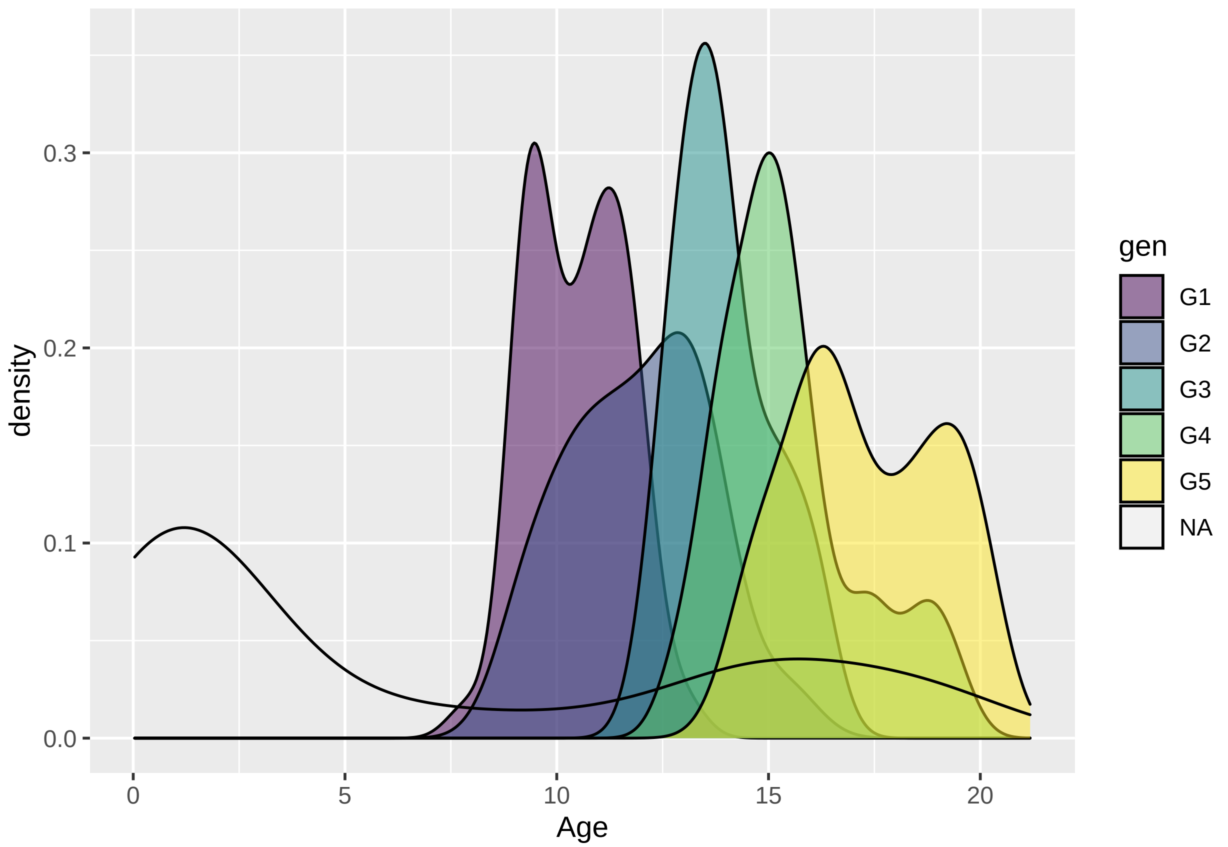


The average age of boys for whom the region is missing, are very young. The boys from the northern region are the slightly older than the boys from the other regions. There are only minor differences between the other regions.

9.

Create a density plot of age, splitting the densities by gen using the fill aesthetic.

```
ggplot(boys, aes(x = age, fill = gen)) +
  geom_density(alpha = 0.5) +
  labs( x = "Age")
```

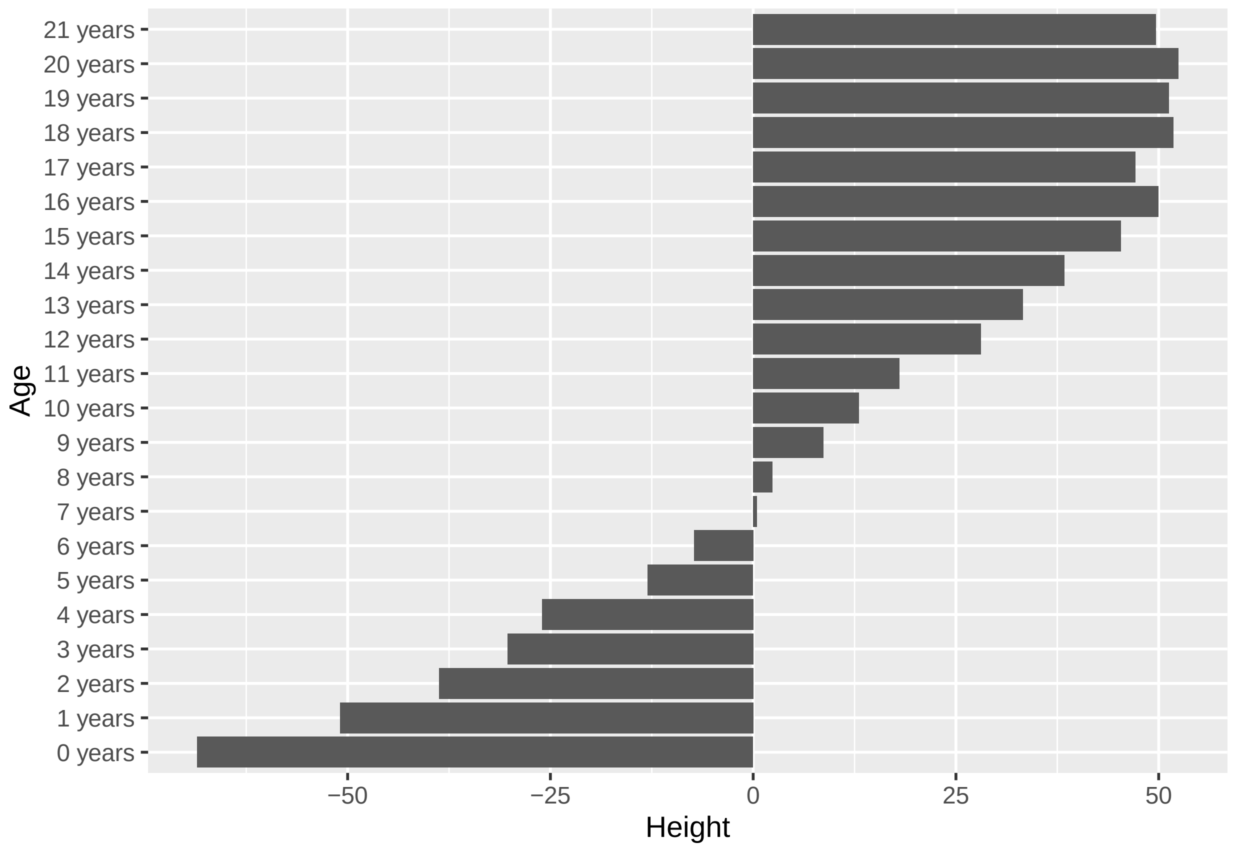


It can be seen that as boys get older they move from G1 to G5. We also see again that there are most missing values for boys aged between 0 and 6.

10.

Create a diverging bar chart for hgt in the boys data set, that displays for every age year that year's mean height in deviations from the overall average hgt.

```
boys %>% mutate(hgt_dev = hgt - mean(hgt, na.rm = TRUE), age_cat = cut(age, 0:22, labels
  summarize(height = mean(hgt_dev, na.rm = TRUE)) %>%
ggplot(aes(y = age_cat, x = height)) +
  geom_bar(stat = "identity") +
  labs( y = "Age", x = "Height")
```



Boys of about 7 years on average have the average height across the whole data set.

11.

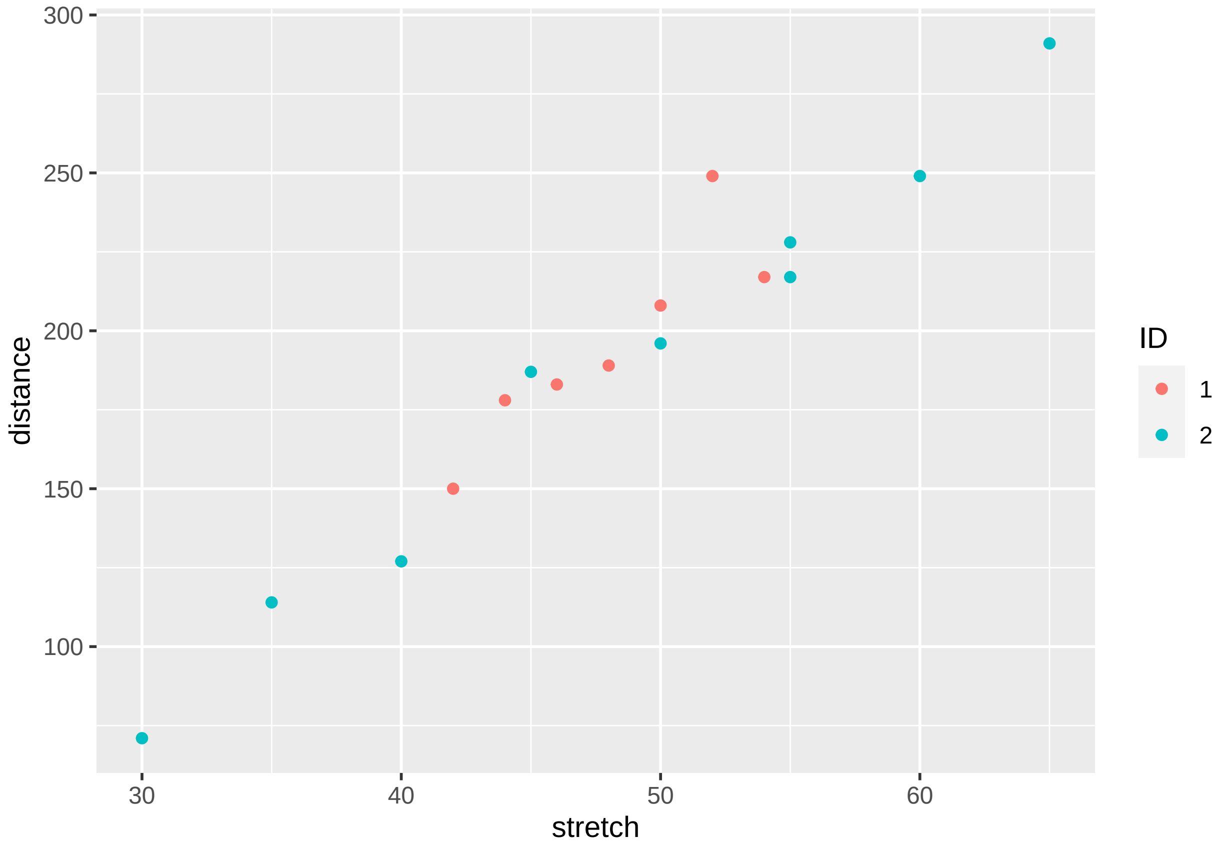
Load the data elastic1 and elastic2 and bind the data frames together using the function bind_rows() and add a grouping variable indicating whether an observation comes from elastic1 or from elastic2.

```
elastic <- bind_rows(elastic1, elastic2, .id = "ID")
```

12.

Create a scatterplot mapping stretch on the x-axis and distance on the y-axis, and map the just created group indicator as the color aesthetic.

```
ggplot(elastic, aes(x= stretch, y = distance, col = ID))+
  geom_point()
```



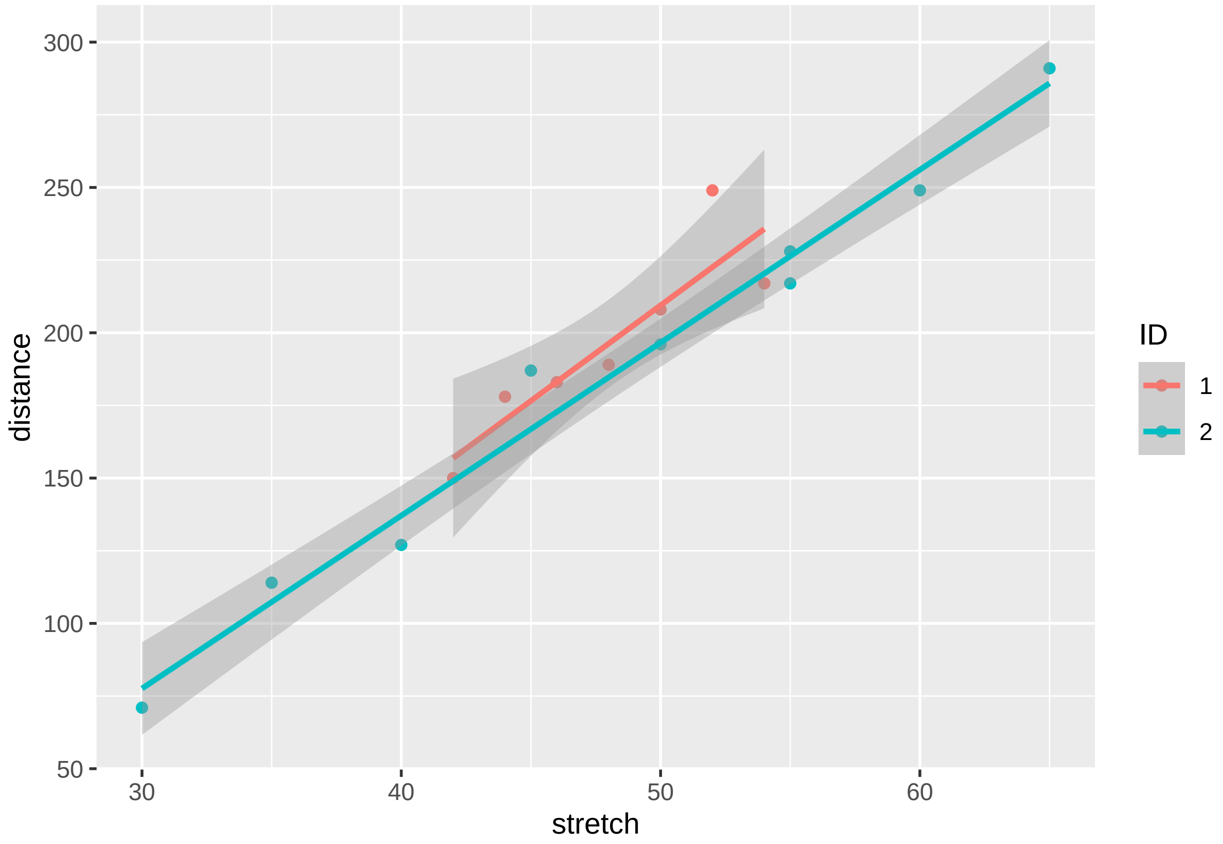
This plot shows which oberservations originated from which original data set.

13.

Recreate the previous plot, but now assess whether the results of the two data sets appear consistent by adding a linear regression line.

```
ggplot(elastic, aes(x= stretch, y = distance, col = ID))+
  geom_point()+
  geom_smooth(method = "lm")
```

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



The results of the two data sets appear to be consistent considering the linear regression lines being parallel to each other and not far apart.

14.

For each of the data sets elastic1 and elastic2, fit a regression model with y = distance on x = stretch using lm(y ~ x, data).

```
lm_el1 <- lm(distance ~ stretch, data= elastic1)

lm_el2 <- lm(distance ~stretch, data =elastic2)
```

15.

For both of the previously created fitted models, determine the fitted values and the standard errors of the fitted values, and the proportion explained variance R^2.

```
pred_el1 <- predict(lm_el1, se.fit = TRUE)
pred_el1
```

```
## $fit
##        1        2        3        4        5        6        7
## 183.1429 235.7143 196.2857 209.4286 170.0000 156.8571 222.5714
##
## $se.fit
## [1]  6.586938 10.621119  5.891537  6.586938  8.331891 10.621119  8.331891
##
## $df
## [1] 5
##
## $residual.scale
## [1] 15.58754
```

```
summary(lm_el1)
```

```
##
## Call:
## lm(formula = distance ~ stretch, data = elastic1)
##
## Residuals:
##        1        2        3        4        5        6        7
##  -0.1429 -18.7143  -7.2857  -1.4286   8.0000  -6.8571  26.4286
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -119.143     70.943  -1.679  0.15391
## stretch        6.571      1.473   4.462  0.00663 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.59 on 5 degrees of freedom
## Multiple R-squared:  0.7992, Adjusted R-squared:  0.7591
## F-statistic: 19.91 on 1 and 5 DF,  p-value: 0.006631
```

```
#Residual standard error: 15.59
#R-squared:  0.7992

pred_el2 <- predict(lm_el2, se.fit = TRUE)
pred_el2
```

```
## $fit
##         1         2         3         4         5         6         7         8 
##  77.58333 196.58333 137.08333 166.83333 256.08333 226.33333 107.33333 226.33333 
##         9 
## 285.83333 
## 
## $se.fit
## [1] 6.740293 3.520003 4.358744 3.635444 5.060323 4.064550 5.453165 4.064550
## [9] 6.296773
## 
## $df
## [1] 7
## 
## $residual.scale
## [1] 10.44202
```

```
summary(lm_el2)
```

```
## 
## Call:
## lm(formula = distance ~ stretch, data = elastic2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -10.0833  -7.0833  -0.5833   5.1667  20.1667 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -100.9167    15.6102  -6.465 0.000345 ***
## stretch        5.9500     0.3148  18.899 2.89e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 10.44 on 7 degrees of freedom
## Multiple R-squared:  0.9808, Adjusted R-squared:  0.978 
## F-statistic: 357.2 on 1 and 7 DF,  p-value: 2.888e-07
```

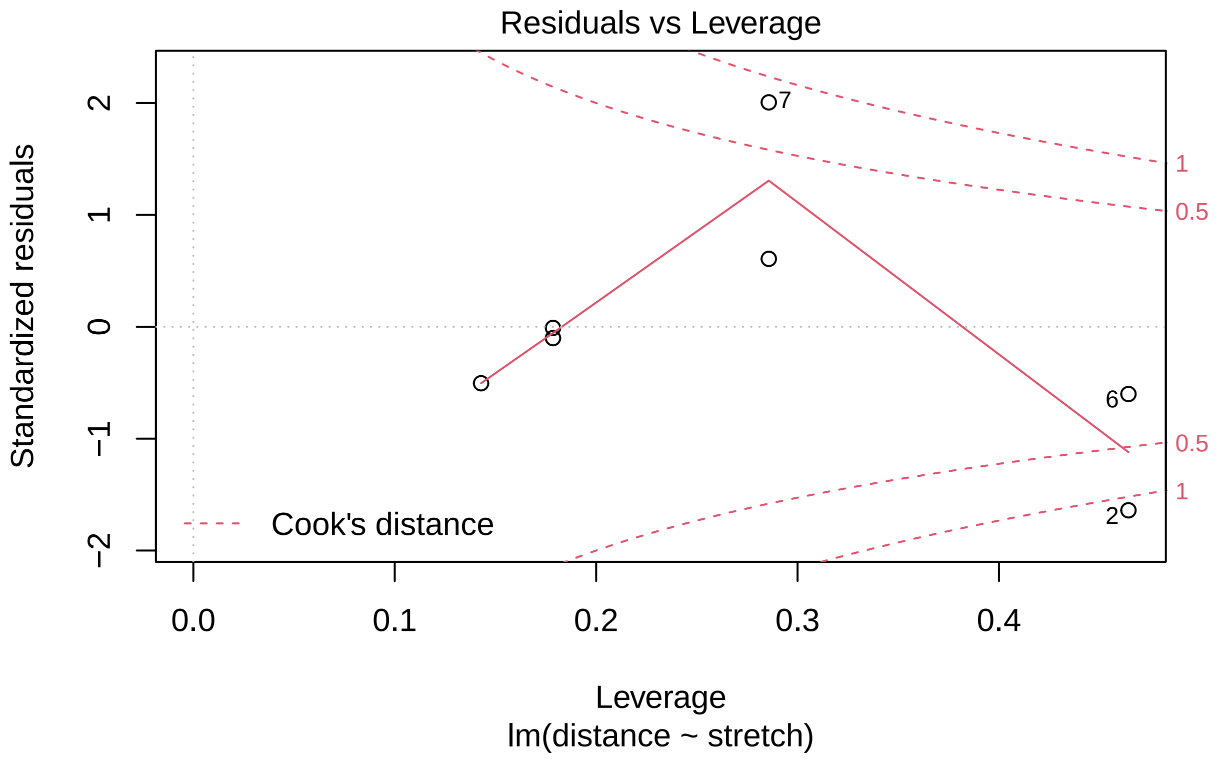
```
#Residual standard error: 10.44
#R-squared:  0.9808
```

The second model has smaller residual standard error and a higher R-squared.

16.

Study the residual versus leverage plots for both models.

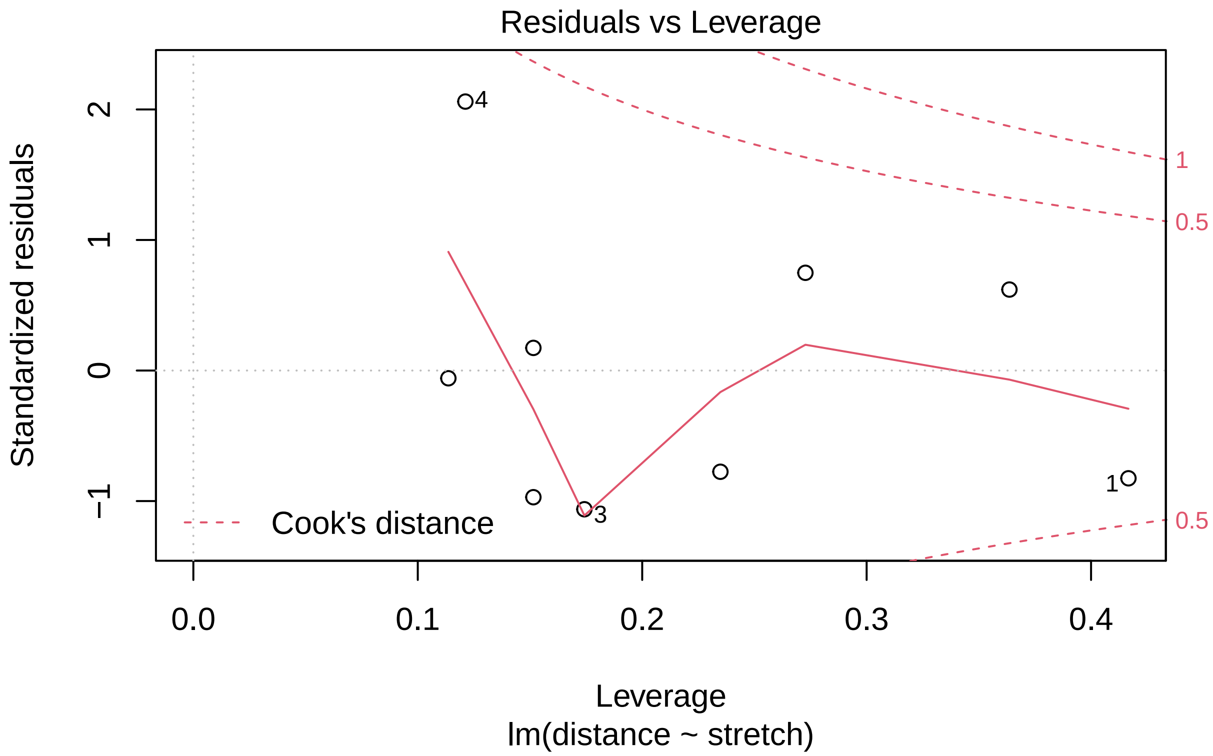
```
plot(lm_el1, which= 5)
```



```
lm_el1$residuals
```

```
##           1           2           3           4           5           6
##  -0.1428571 -18.7142857  -7.2857143  -1.4285714   8.0000000  -6.8571429
##           7
##  26.4285714
```

```r
plot(lm_el2, which = 5)
```



```r
lm_el2$residuals
```

```
##          1          2          3          4          5          6 
## -6.5833333 -0.5833333 -10.0833333 20.1666667 -7.0833333 -9.3333333 
##          7          8          9 
##  6.6666667  1.6666667  5.1666667
```

There are two cases with a large influence (2 and 7) in elastic1. There are no such cases for elastic2.

17.

Use the elastic2 variable stretch to obtain predictions on the model fitted on elastic1.

```
elastic2 <- elastic2 %>% mutate(pred=  predict(lm_el1, newdata = elastic2 ))
```

18.

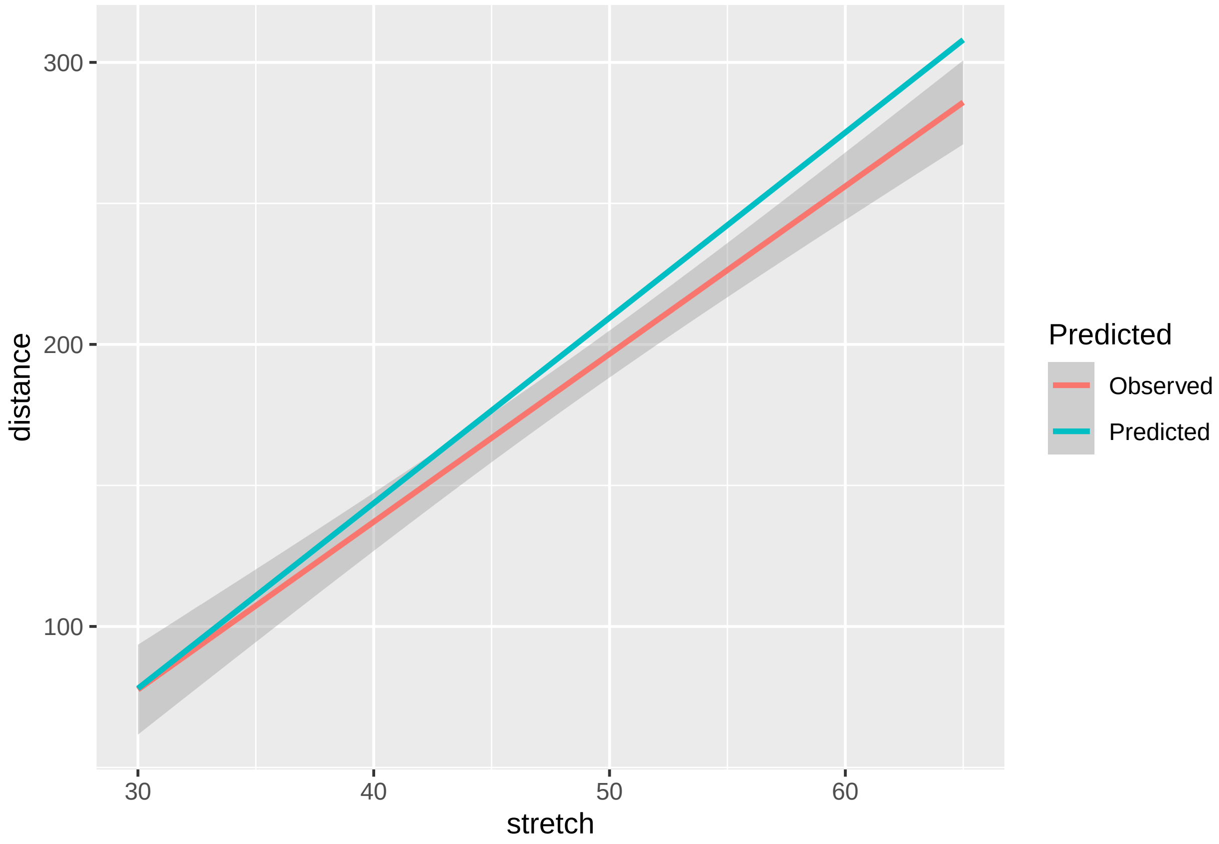
Now make a scatterplot to investigate similarity between the predicted values and the observed values for elastic2.

LOOK AT THIS AGAIN!

```
elastic_pred <- data.frame(distance = elastic2$pred, stretch = elastic2$stretch) %>%
  bind_rows(Predicted = ., Observed = elastic2, .id = "Predicted")

ggplot(elastic_pred, aes(x= stretch, y=distance, col = Predicted )) +
  geom_smooth(method = lm)
```

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



The predicted values are slightly higher than the observed values, but in general very similar. (Note that we used the model from elastic1 to predict the values for elastic2).