

# Lab 3: Modeling correlation and regression

Practice session covering topics  
discussed in Lecture 3

M. Chiara Mimmi, Ph.D. | Università degli Studi di Pavia

July 26, 2024

# GOAL OF TODAY'S PRACTICE SESSION

- Review the basic questions we can ask about ASSOCIATION between any two variables:
  - does it exist?
  - how strong is it?
  - what is its direction?
- Introduce a widely used analytical tool: REGRESSION

The examples and code from this lab session follow very closely the open access book:

- Vu, J., & Harrington, D. (2021). **Introductory Statistics for the Life and Biomedical Sciences.**  
<https://www.openintro.org/book/biostat/>

# Topics discussed in Lecture # 3

## Lecture 3: topics

- Testing and summarizing relationship between 2 variables (**correlation**)
  - Pearson's  $r$  analysis (param)
  - Spearman test (no param)
- Measures of **association**
  - Chi-Square test of independence
  - Fisher's Exact Test
    - alternative to the Chi-Square Test of Independence
- From correlation/association to **prediction/causation**
  - The purpose of observational and experimental studies
- Widely used analytical tools
  - Simple linear regression models
  - Multiple Linear Regression models
- Shifting the emphasis on **empirical prediction**
  - Introduction to Machine Learning (ML)
  - Distinction between Supervised & Unsupervised algorithms

# R ENVIRONMENT SET UP & DATA

# Needed R Packages

- We will use functions from packages **base**, **utils**, and **stats** (pre-installed and pre-loaded)
- We will also use the packages below (specifying **package::function** for clarity).

```
1 # Load pckgs for this R session
2
3 # -- General
4 library(fs)      # file/directory interactions
5 library(here)    # tools find your project's files, based on working directory
6 library(paint)   # paint data.frames summaries in colour
7 library(janitor) # tools for examining and cleaning data
8 library(dplyr)   # {tidyverse} tools for manipulating and summarizing tidy data
9 library(forcats) # {tidyverse} tool for handling factors
10 library(openxlsx) # Read, Write and Edit xlsx Files
11 library(flextable) # Functions for Tabular Reporting
12 # -- Statistics
13 library(rstatix) # Pipe-Friendly Framework for Basic Statistical Tests
14 library(lmtest)  # Testing Linear Regression Models
15 library(broom)   # Convert Statistical Objects into Tidy Tibbles
16 #library(tidymodels) # not installed on this machine
17 library(performance) # Assessment of Regression Models Performance
18 # -- Plotting
19 library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
```

# DATASETS FOR TODAY

We will use examples (with adapted datasets) from real clinical studies, provided among the learning materials of the open access books:

- Vu, J., & Harrington, D. (2021). **Introductory Statistics for the Life and Biomedical Sciences.** <https://www.openintro.org/book/biostat/>
- Çetinkaya-Rundel, M., & Hardim, J. (2023). **Introduction to Modern Statistics (1st Ed).** <https://openintro-ims.netlify.app/>

# Importing Dataset 1 (NHANES)

**Name:** NHANES (National Health and Nutrition Examination Survey) combines interviews and physical examinations to assess the health and nutritional status of adults and children in the United States. Started in the 1960s, it became a continuous program in 1999.

**Documentation:** [dataset1](#)

**Sampling details:** Here we use a sample of 500 adults from NHANES 2009-2010 & 2011-2012 ([nhanes.samp.adult.500](#) in the R [oibiostat](#) package, which has been adjusted so that it can be viewed as a random sample of the US population)

```
1 # Check my working directory location
2 # here::here()
3
4 # Use `here` in specifying all the subfolders AFTER the working directory
5 nhanes_samp <- read.csv(file = here::here("practice", "data_input", "03_datasets",
6                                         "nhanes.samp.csv"),
7                         header = TRUE, # 1st line is the name of the variables
8                         sep = ",", # which is the field separator character.
9                         na.strings = c("?", "NA" ), # specific MISSING values
10                        row.names = NULL)
```

- Adapting the function [here](#) to match your own folder structure

# NHANES Variables and their description

[EXCERPT: see complete file in Input Data Folder]

Variable	Type	Description
X	int	xxxx
ID	int	xxxxx
SurveyYr	chr	yyyy_mm. Ex. 2011_12
Gender	chr	Gender (sex) of study participant coded as male or female
Age	int	##
AgeDecade	chr	yy-yy es 20-29
Education	chr	[>= 20 yro]. Ex. 8thGrade, 9-11thGrade, HighSchool, SomeCollege, or CollegeGrad.
Weight	dbl	Weight in kg
Height	dbl	Standing height in cm. Reported for participants aged 2 years or older.
BMI	dbl	Body mass index (weight/height <sup>2</sup> in kg/m <sup>2</sup> ). Reported for participants aged 2 years or older
Pulse	int	60 second pulse rate
DirectChol	dbl	Direct HDL cholesterol in mmol/L. Reported for participants aged 6 years or older
TotChol	dbl	Total HDL cholesterol in mmol/L. Reported for participants aged 6 years or older
Diabetes	chr	Study participant told by a doctor or health professional that they have diabetes
DiabetesAge	int	Age of study participant when first told they had diabetes
HealthGen	chr	Self-reported rating of health: Excellent, Vgood, Good, Fair, or Poor Fair
Alcohol12PlusYr	chr	Participant has consumed at least 12 drinks of any type of alcoholic beverage in any one year
...	...	...

# Importing Dataset 2 (PREVEND)

**Name:** PREVEND (**P**revention of **R**enal and **V**ascular **E**ND-stage **D**isease) is a study which took place in the Netherlands starting in the 1990s, with subsequent follow-ups throughout the 2000s. This dataset is from the third survey, which participants completed in 2003-2006; data is provided for 4,095 individuals who completed cognitive testing.

**Documentation:** [dataset2](#) and sample dataset variables' [codebook](#)

**Sampling details:** Here we use a sample of 500 adults taken from 4,095 individuals who completed cognitive testing (i.e. the `prevend.samp` dataset in the R `obiostat` package)

```
1 # Check my working directory location
2 # here::here()
3
4 # Use `here` in specifying all the subfolders AFTER the working directory
5 prevend_samp <- read.csv(file = here::here("practice", "data_input", "03_datasets",
6                               "prevend.samp.csv"),
7                           header = TRUE, # 1st line is the name of the variables
8                           sep = ",", # which is the field separator character.
9                           na.strings = c("?", "NA" ), # specific MISSING values
10                          row.names = NULL)
```

# **PREVEND Variables and their description**

[EXCERPT: see complete file in Input Data Folder]

Variable	Type	Description
X	int	Patient ID
Age	int	Age in years
Gender	int	Expressed as: 0 = males; 1 = females
RFFT	int	Performance on the Ruff Figural Fluency Test. Scores range from 0 (worst) to 175 (best)
VAT	int	Visual Association Test score. Scores may range from 0 (worst) to 12 (best)
Chol	dbl	Total cholesterol, in mmol/L.
HDL	dbl	HDL cholesterol, in mmol/L.
Statin	int	Statin use at enrollment. Numeric vector: 0 = No; 1 = Yes.
CVD	int	History of cardiovascular event. Numeric vector: 0 = No; 1 = Yes
DM	int	Diabetes mellitus status at enrollment. Numeric vector: 0 = No; 1 = Yes
Education	int	Highest level of education. Numeric: 0 primary school; 1 = lower secondary education; 3 = university
Smoking	int	Smoking at enrollment. numeric vector: 0 = No; 1 = Yes
Hypertension	int	Status of hypertension at enrollment. Numeric vector: 0 = No; 1 = Yes
Ethnicity	int	Expressed as: 0 = Western European; 1 = African; 2 = Asian; 3 = Other
...	...	...

# Importing Dataset 3 (FAMuSS)

**Name:** FAMuSS (Functional SNPs Associated with Muscle Size and Strength) examine the association of demographic, physiological and genetic characteristics with muscle strength – including data on race and genotype at a specific locus on the ACTN3 gene (the “sports gene”).

**Documentation:** [dataset3](#)

**Sampling details:** the DATASET includes 595 observations on 9 variables ([famuss](#) in the R [obiostat](#) package)

```
1 # Check my working directory location
2 # here::here()
3
4 # Use `here` in specifying all the subfolders AFTER the working directory
5 famuss <- read.csv(file = here::here("practice", "data_input", "03_datasets",
6                               "famuss.csv"),
7                     header = TRUE, # 1st line is the name of the variables
8                     sep = ",", # which is the field separator character.
9                     na.strings = c("?", "NA" ), # specific MISSING values
10                    row.names = NULL)
```

# **FAMuSS Variables and their description**

[See complete file in Input Data Folder]

Variable	Description
X	id
ndrm.ch	Percent change in strength in the non-dominant arm
drm.ch	Percent change in strength in the dominant arm
sex	Sex of the participant
age	Age in years
race	Recorded as African Am (African American), Caucasian, Asian, Hispanic, Other
height	Height in inches
weight	Weight in pounds
actn3.r577x	Genotype at the location r577x in the ACTN3 gene.
bmi	Body Mass Index

# CORRELATION

[Using NHANES and FAMuSS datasets]

# Explore relationships between two variables

Approaches for summarizing relationships between two variables vary depending on variable types...

- Two **numerical** variables
- Two **categorical** variables
- One **numerical** variable and one **categorical** variable

Two variables x and y are

- *positively associated* if y increases as x increases.
- *negatively associated* if y decreases as x increases.

# **TWO NUMERICAL VARIABLES (NHANES)**

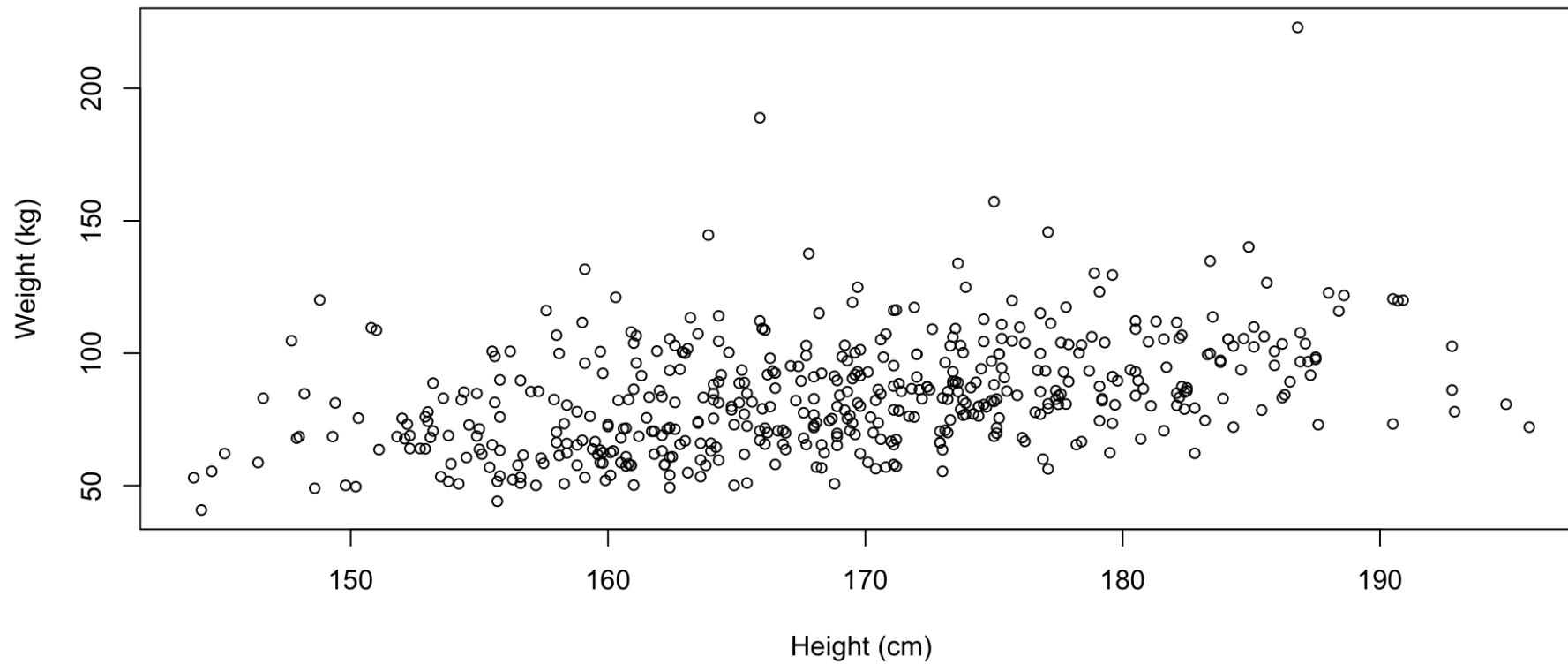
# Two numerical variables (plot)

**Height** and **weight** (taken from the `nhanes_samp` dataset) are positively associated.

- notice we can also use the generic base R function `plot` for a quick scatter plot

```
1 # rename for convenience
2 nhanes <- nhanes_samp %>%
3   janitor::clean_names()
4
5 # basis plot
6 plot(nhanes$height, nhanes$weight,
7       xlab = "Height (cm)", ylab = "Weight (kg)", cex = 0.8)
```

# Two numerical variables (plot)



## Two numerical variables: correlation (with `stats::cor`)

**Correlation** is a numerical summary that measures the strength of a linear relationship between two variables.

- The correlation coefficient  $r$  takes on values between  $-1$  and  $1$ .
- The closer  $r$  is to  $\pm 1$ , the stronger the linear association.
- Here we compute the **Pearson rho (parametric)**, with base R function `stats::cor`
  - the `use` argument let us choose how to deal with missing values (in this case only using **all complete pairs**)

```
1 is.numeric(nhanes$height)
[1] TRUE

1 is.numeric(nhanes$weight)
[1] TRUE

1 # using `stats` package
2 stats::cor(x = nhanes$height, y = nhanes$weight,
3             # argument for dealing with missing values
4             use = "pairwise.complete.obs",
5             method = "pearson")

[1] 0.4102269
```

## Two numerical variables: correlation (with `stats::cor.test`)

- Here we compute the **Pearson rho (parametric)**, with the function `cor.test` (the same we used for testing paired samples)
  - implicitly takes care on `NAs`

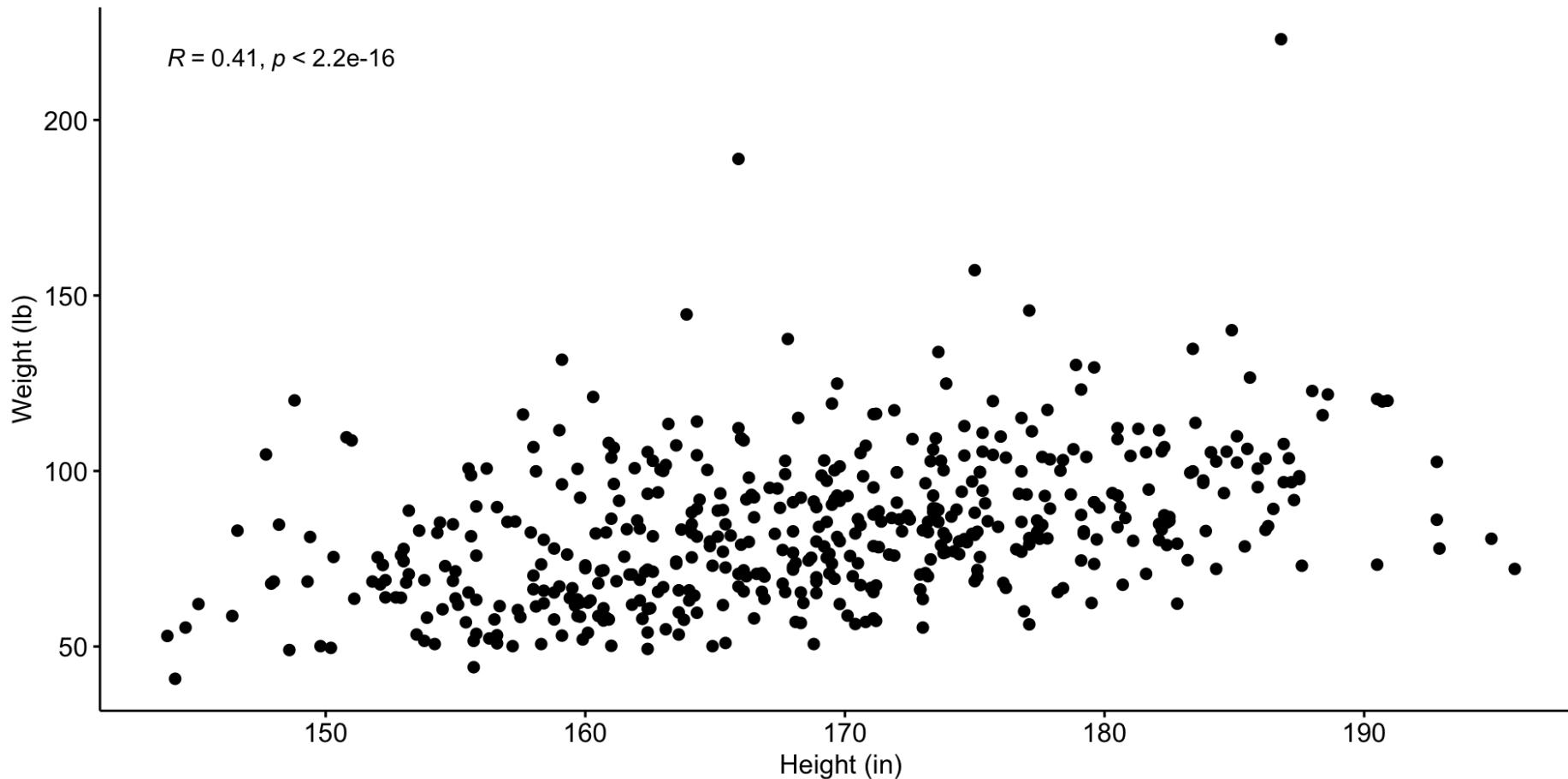
```
1 # using `stats` package
2 cor_test_result <- cor.test(x = nhanes$height, y = nhanes$weight,
3                             method = "pearson")
4
5 # looking at the cor estimate
6 cor_test_result[["estimate"]][["cor"]]
```

```
[1] 0.4102269
```

- The function `ggpubr::ggscatter` gives us all in one (scatter plot + r ("R"))! 😱

```
1 library("ggpubr") # 'ggplot2' Based Publication Ready Plots
2 ggpubr::ggscatter(nhanes, x = "height", y = "weight",
3                   cor.coef = TRUE, cor.method = "pearson", #cor.coef.coord = 2,
4                   xlab = "Height (in)", ylab = "Weight (lb)")
```

## Two numerical variables: correlation (with `stats::cor.test`)



# Spearman rank-order correlation

The **Spearman's rank-order correlation** is the **nonparametric version** of the **Pearson** correlation.

Spearman's correlation coefficient, ( $\rho$ , also signified by  $r_s$ ) measures the strength and direction of association between two ranked variables.

- used when 2 variables have a **non-linear** relationship
- excellent for **ordinal** data (when Pearson's is not appropriate), i.e. Likert scale items

To compute it, we simply calculate Pearson's correlation of the **rankings** of the raw data (instead of the data).

# Spearman rank-order correlation (example)

Let's say we want to get Spearman's correlation with ordinal factors **Education** and **HealthGen** in the **NHANES** sample.

- We have to convert them to their underlying numeric code, to compare rankings.

```
1 tabyl(nhanes$education)

nhanes$education    n percent valid_percent
    8th Grade   32    0.064    0.06412826
  9 - 11th Grade  68    0.136    0.13627255
 College Grad  157    0.314    0.31462926
 High School   94    0.188    0.18837675
 Some College  148    0.296    0.29659319
 <NA>          1    0.002      NA

1 tabyl(nhanes$health_gen)

nhanes$health_gen    n percent valid_percent
 Excellent    47    0.094    0.10444444
 Fair        53    0.106    0.11777778
 Good       177    0.354    0.39333333
 Poor        11    0.022    0.02444444
 Vgood      162    0.324    0.36000000
 <NA>        50    0.100      NA

1 nhanes <- nhanes %>%
2   # reorder education
3   mutate (edu_ord = factor (education,
4                           levels = c("8th Grade", "9 - 11th Grade",
5                                     "High School", "Some College",
6                                     "College Grad" , NA))) %>%
7   # create edu_rank
8   mutate (edu_rank = as.numeric(edu_ord)) %>%
9   # reorder health education
10  mutate (health_ord = factor (health_gen,
11                           levels = c( NA, "Poor", "Fair",
12                                     "Good", "Vgood",
13                                     "Excellent")))) %>%
14  # create health_rank
15  mutate (health_rank = as.numeric(health_ord))
```

# Spearman rank-order correlation (example), cont.

- Let's check out the `...._rank` version of the 2 categorical variables of interest:
  - `education` from `edu_ord` to `edu_rank`

```
1 table(nhanes$edu_ord, useNA = "ifany" )
```

8th Grade	9 - 11th Grade	High School	Some College	College Grad
32	68	94	148	157
<NA>				
1				

```
1 table(nhanes$edu_rank, useNA = "ifany" )
```

1	2	3	4	5 <NA>
32	68	94	148	157
				1

- `general health` from `health_ord` to `health_rank`

```
1 table(nhanes$health_ord, useNA = "ifany" )
```

Poor	Fair	Good	Vgood	Excellent	<NA>
11	53	177	162	47	50

```
1 table(nhanes$health_rank, useNA = "ifany" )
```

1	2	3	4	5 <NA>
11	53	177	162	47
				50

# Spearman rank-order correlation (example cont.)

After setting up the variables in the correct (numerical rank) format, now we can actually compute it: + same function call **stats::cor.test** + but specifying argument **method = "spearman"**

```
1 # -- using `stats` package
2 cor_test_result_sp <- cor.test(x = nhanes$edu_rank,
3                                y = nhanes$health_rank,
4                                method = "spearman",
5                                exact = FALSE) # removes the Ties message warning
6 # looking at the cor estimate
7 cor_test_result_sp
```

```
Spearman's rank correlation rho

data: nhanes$edu_rank and nhanes$health_rank
S = 10641203, p-value = 1.915e-10
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.2946493
```

```
1 # -- only print Spearman rho
2 #cor_test_result_sp[["estimate"]][["rho"]]
```

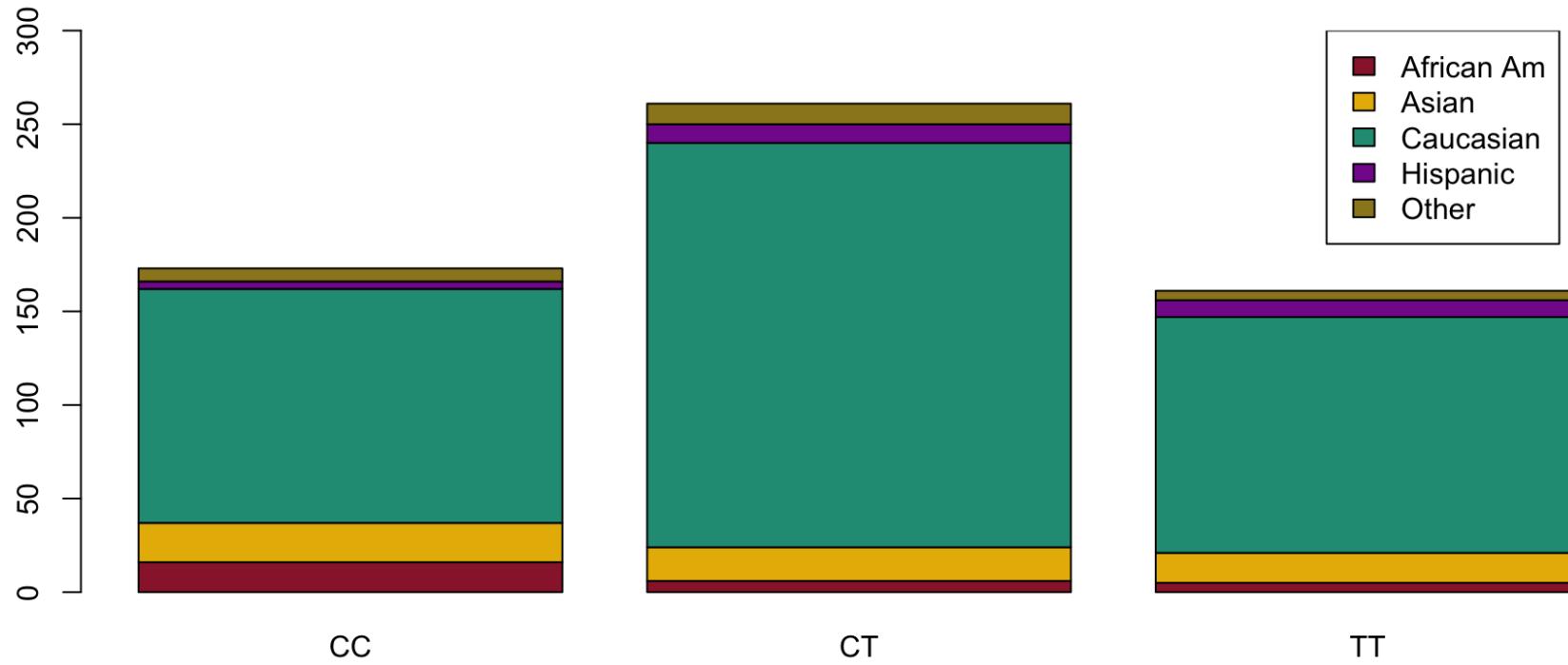
# **TWO CATEGORICAL VARIABLES (FAMuSS)**

# Two categorical variables (plot)

In the `famuss` dataset, the variables `race`, and `actn3.r577x` are categorical variables.

- we can use the generic base R function `graphics::barplot`

```
1 mycolors_contrast <- c("#9b2339", "#E7B800", "#239b85", "#85239b", "#9b8523", "#23399b", "#d8e600", "#0084e6", "#399B23"
2
3 ## genotypes as columns
4 genotype.race = matrix(table(famuss$actn3.r577x, famuss$race), ncol=3, byrow=T)
5 colnames(genotype.race) = c("CC", "CT", "TT")
6 rownames(genotype.race) = c("African Am", "Asian", "Caucasian", "Hispanic", "Other")
7
8 # using generic base::barplot
9 graphics::barplot(genotype.race, col = mycolors_contrast[1:5], ylim=c(0,300), width=2)
10 legend("topright", inset=c(.05, 0), fill=mycolors_contrast[1:5],
11        legend=rownames(genotype.race))
```



# Two categorical variables (contingency table)

Specifically, the variable `actn3.r577x` takes on three possible levels (`CC`, `CT`, or `TT`) which indicate the distribution of genotype at location `r577x` on the `ACTN3` gene for the FAMuSS study participants.

A **contingency table** summarizes data for two categorical variables.

- the function `stats:::addmargins` puts arbitrary *Margins* on multidimensional tables
  - The extra column & row "`Sum`" provide the *marginal totals* across each row and each column, respectively

```
1 # levels of actn3.r577x
2 table(famuss$actn3.r577x)
```

```
CC  CT  TT
173 261 161
```

```
1 # contingency table to summarize race and actn3.r577x
2 addmargins(table(famuss$race, famuss$actn3.r577x))
```

	CC	CT	TT	Sum
African Am	16	6	5	27
Asian	21	18	16	55
Caucasian	125	216	126	467
Hispanic	4	10	9	23
Other	7	11	5	23
Sum	173	261	161	595

# Two categorical variables (contingency table prop)

Contingency tables can also be converted to show *proportions*. Since there are 2 variables, it is necessary to specify whether the proportions are calculated according to the row variable or the column variable.

- using the `margin =` argument in the `base::prop.table` function (1 indicates rows, 2 indicates columns)

```
1 # adding row proportions  
2 addmargins(prop.table(table(famuss$race, famuss$actn3.r577x), margin = 1))
```

	CC	CT	TT	Sum
African Am	0.5925926	0.2222222	0.1851852	1.0000000
Asian	0.3818182	0.3272727	0.2909091	1.0000000
Caucasian	0.2676660	0.4625268	0.2698073	1.0000000
Hispanic	0.1739130	0.4347826	0.3913043	1.0000000
Other	0.3043478	0.4782609	0.2173913	1.0000000
Sum	1.7203376	1.9250652	1.3545972	5.0000000

```
1 # adding column proportions  
2 addmargins(prop.table(table(famuss$race, famuss$actn3.r577x), margin = 2))
```

	CC	CT	TT	Sum
African Am	0.09248555	0.02298851	0.03105590	0.14652996
Asian	0.12138728	0.06896552	0.09937888	0.28973168
Caucasian	0.72254335	0.82758621	0.78260870	2.33273826
Hispanic	0.02312139	0.03831418	0.05590062	0.11733618
Other	0.04046243	0.04214559	0.03105590	0.11366392
Sum	1.00000000	1.00000000	1.00000000	3.00000000

# Chi Squared test of independence

The **Chi-squared test** is a hypothesis test used to determine whether there is a relationship between **two categorical variables**.

- categorical vars. can have *nominal* or *ordinal* measurement scale
- the *observed* frequencies are compared with the *expected* frequencies and their deviations are examined.

```
1 # Chi-squared test
2 # (Test of association to see if
3 # H0: the 2 cat var (race & actn3.r577x) are independent
4 # H1: the 2 cat var are correlated in __some way__
5
6 tab <- table(famuss$race, famuss$actn3.r577x)
7 test_chi <- chisq.test(tab)
```

the obtained result (**test\_chi**) is a list of objects...

 You try...

...run **View(test\_chi)** to check

# Chi Squared test of independence (cont)

Within `test_chi` results there are:

- Observed frequencies = how often a combination occurs in our sample
- Expected frequencies = what would it be if the 2 vars were PERFECTLY INDEPENDENT

```
1 # Observed frequencies  
2 test_chi$observed
```

	CC	CT	TT
African Am	16	6	5
Asian	21	18	16
Caucasian	125	216	126
Hispanic	4	10	9
Other	7	11	5

```
1 # Expected frequencies  
2 round(test_chi$expected , digits = 1)
```

	CC	CT	TT
African Am	7.9	11.8	7.3
Asian	16.0	24.1	14.9
Caucasian	135.8	204.9	126.4
Hispanic	6.7	10.1	6.2
Other	6.7	10.1	6.2

# Chi Squared test of independence (results)

- Recall that:
  - $H_0$ : the 2 cat. var. are **independent**
  - $H_1$ : the 2 cat. var. are **correlated** in some way
- The result of Chi-Square test represents a comparison of the above two tables (*observed v. expected*):
  - p-value = 0.01286 smaller than  $\alpha = 0.05$  so **we REJECT the null hypothesis** (i.e. there's likely an association between race and ACTN3 gene)

```
1 test_chi
```

```
Pearson's Chi-squared test

data: tab
X-squared = 19.4, df = 8, p-value = 0.01286
```

# Computing Cramér's V after test of independence

Recall that **Crammer's V** allows to measure the *effect size* of the test of independence (i.e. the **strength of association** between two nominal variables)

- $V$  ranges from [0 1] (the smaller  $V$ , the lower the correlation)

$$V = \sqrt{\frac{\chi^2}{n(k - 1)}}$$

where:

- $V$  denotes Cramér's V
- $\chi^2$  is the Pearson chi-square statistic from the prior test
- $n$  is the sample size involved in the test
- $k$  is the lesser number of categories of either variable

# Computing Cramer's V after test of independence (2 ways)

-  “By hand” first to see the steps

```
1 # Compute Cramer's V by hand
2
3 # inputs
4 chi_calc <- test_chi$statistic
5 n <- nrow(famuss) # N of obs
6 n_r <- nrow(test_chi$observed) # number of rows in the contingency table
7 n_c <- ncol(test_chi$observed) # number of columns in the contingency table
8
9 # Cramer's V
10 sqrt(chi_calc / (n*min(n_r -1, n_c -1)))
```

X-squared  
0.1276816

-  Using an R function `rstatix::cramer_v`

```
1 # Cramer's V with rstatix
2 rstatix::cramer_v(test_chi$observed)
```

[1] 0.1276816

**Cramer's V = 0.12**, which indicates a relatively weak association between the two categorical variables. It suggests that while there may be some relationship between the variables, it is not particularly strong.

# Chi Squared test of goodness of fit

In some cases the Chi-square test examines whether or not an observed frequency distribution matches an expected theoretical distribution.

Here, we are conducting a type of Chi-square Goodness of Fit Test which:

- serves to test whether the observed distribution of a categorical variable differs from your expectations
- interprets the statistic based on the discrepancies between observed and expected counts

# Chi Squared test of goodness of fit (example)

Since the participants of the **FAMuSS study** where *volunteers* at a university, they did not come from a “representative” sample of the US population, we can use the  $\chi^2$  goodness of fit test to test against:

- $H_0$ : the study participants (1st row below) are racially representative of the general population (2nd row below)

Race	African.American	Asian	Caucasian	Other	Total
FAMuSS (Observed)	27	55	467	46	595
US Census (Expected)	76.16	5.95	478.38	34.51	595

We use the formula

$$\chi^2 = \sum_k \frac{(Observed - Expected)^2}{Expected}$$

Under  $H_0$ , the sample proportions should equal the population proportions.

# Chi Squared test of goodness of fit (example)

```
1 # Subset the vectors of frequencies from the 2 rows
2 observed <- c(27, 55, 467, 46)
3 expected <- c(76.2, 5.95, 478.38, 34.51)
4
5 # Calculate Chi-Square statistic manually
6 chi_sq_statistic <- sum((observed - expected)^2 / expected)
7 df <- length(observed) - 1
8 p_value <- 1 - pchisq(chi_sq_statistic, df)
9
10 # Print results
11 chi_sq_statistic
```

```
[1] 440.2166
```

```
1 df
```

```
[1] 3
```

```
1 p_value
```

```
[1] 0
```

The calculated  $\chi^2$  statistic is very large, and the **p\_value** is close to 0. Hence, there is more than sufficient evidence to **reject the null hypothesis** that the sample is representative of the general population.

Comparing the observed and expected values (or the residuals), we find the **largest discrepancy with the over-representation of Asian study participants**.

# SIMPLE LINEAR REGRESSION

[Using NHANES dataset]

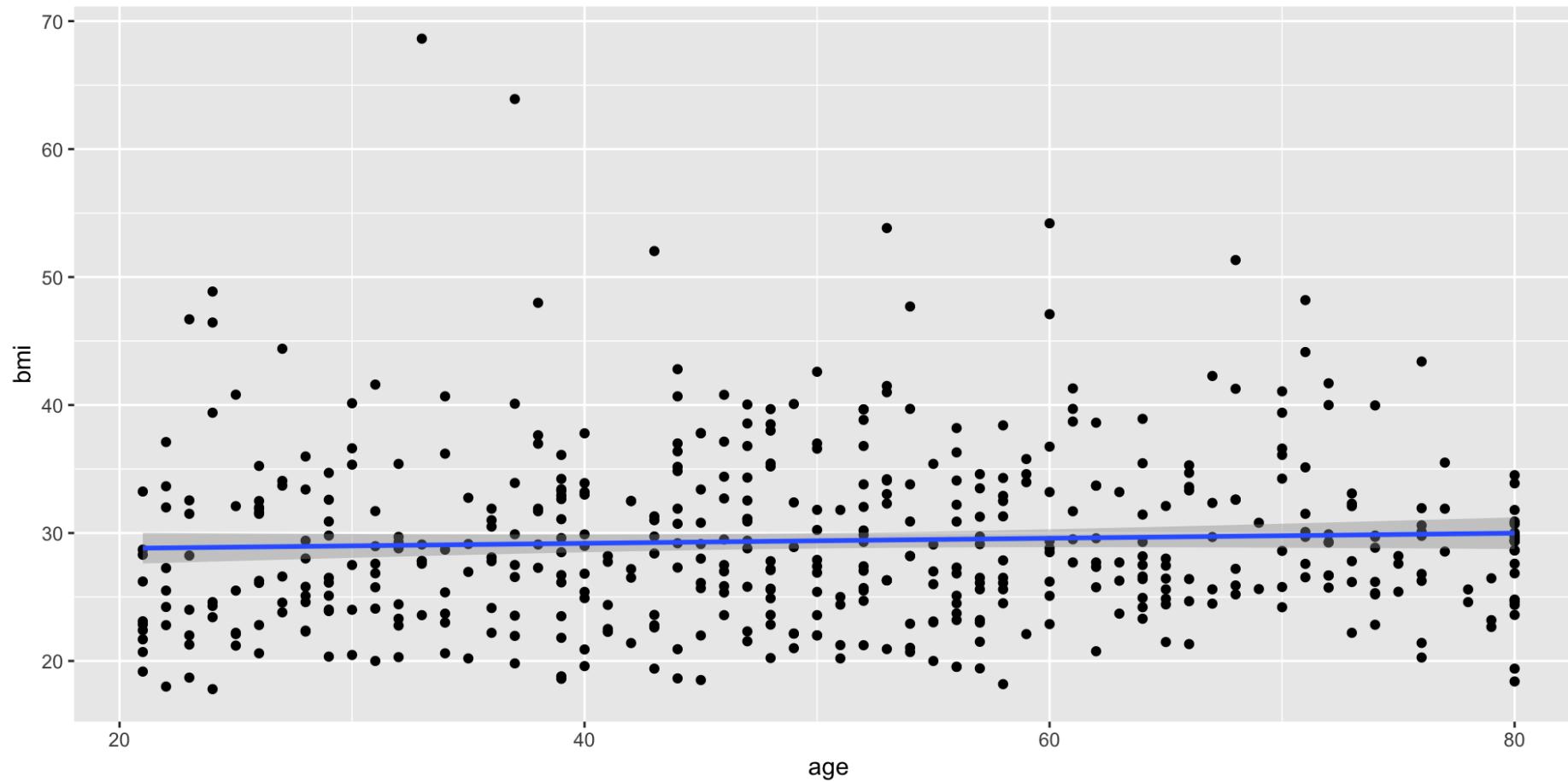
# Visualize the data: BMI and age

We are mainly looking for a “vaguely” linear shape here

- `ggplot2` gives us a visual confirmation with `geom_point()`
- Essentially, `geom_smooth()` adds a trend line over an existing plot
  - inside the function, we have different options with the `method` argument (default is LOESS (locally estimated scatterplot smoothing))
  - with `method = lm` we get the linear best fit (the **least squares regression line**) & its 95% CI

```
1 ggplot(nhanes, aes (x = age,
2                         y = bmi)) +
3   geom_point() +
4   geom_smooth(method = lm,
5               #se = FALSE
6               )
```

# Visualize the data: BMI and age



# Linear regression model

The `lm()` function is used to fit linear models has the following generic structure:

```
1 lm(y ~ x, data)
```

where:

- the 1st argument `y ~ x` specifies the variables used in the model (here the model regresses a **response variable** `y` against an **explanatory variable** `x`).
- The 2nd argument `data` is used only when the dataframe name is not already specified in the first argument.

# Linear regression models syntax

The following example shows fitting a linear model that predicts **BMI** from **age (in years)** using data from **nhanes** adult sample (individuals 21 years of age or older from the NHANES data).

```
1 # fitting linear model  
2 lm(nhanes$bmi ~ nhanes$age)  
  
1 # or equivalently...  
2 lm(bmi ~ age, data = nhanes)
```

Call:

```
lm(formula = bmi ~ age, data = nhanes)
```

Coefficients:

(Intercept)	age
28.40113	0.01982

- Running the function creates an *object* (of class **lm**) that contains several components (model coefficients, etc), either directly displayed or accessible with **summary()** notation or specific functions.

# Linear regression models syntax

We can save the model and then extract individual output elements from it using the `$` syntax

```
1 # name the model object  
2 lr_model <- lm(bmi ~ age, data = nhanes)  
3  
4 # extract model output elements  
5 lr_model$coefficients  
6 lr_model$residuals  
7 lr_model$fitted.values
```

The command `summary` returns these elements

- **Call**: reminds the equation used for this regression model
- **Residuals**: a 5 number summary of the distribution of residuals from the regression model
- **Coefficients**: displays the estimated coefficients of the regression model and relative hypothesis testing, given for:
  - intercept
  - explanatory variable(s) slope

# Linear regression models interpretation: coefficients

- The model tests the null hypothesis  $H_0$  that a coefficient is 0
- **coefficients** outputs are: **estimate**, **std. error**, **t-statistic**, and **p-value** correspondent to the t-statistic for:
  - *intercept*
  - *explanatory variable(s)* slope
- In regression, the population **parameter of interest** is typically the **slope** parameter
  - in this model, **age** doesn't appear significantly  $\neq 0$

```
1 summary(lr_model)$coefficients
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	28.40112932	0.96172389	29.531480	2.851707e-111
age	0.01981675	0.01824641	1.086063	2.779797e-01

## Linear regression models interpretation: Coefficients 2

For the estimated coefficients of the regression model, we get:

- **Estimate** = the average increase in the response variable associated with a one unit increase in the predictor variable, (assuming all other predictor variables are held constant).
- **Std. Error** = a measure of the uncertainty in our estimate of the coefficient.
- **t value** = the t-statistic for the predictor variable, calculated as (Estimate) / (Std. Error).
- **Pr(>|t|)** = the p-value that corresponds to the t-statistic. If less than some alpha level (e.g. 0.05). the predictor variable is said to be *statistically significant*.

# Linear regression models outputs: fitted values

Here we see  $\hat{y}_i$ , i.e. the **fitted y value for the i-th individual**

```
1 fit_val <- lr_model$fitted.values  
2  
3 # print the first 6 elements  
4 head(fit_val)
```

```
1           2           3           4           5           6  
29.39197 29.33252 29.31270 28.95600 29.39197 29.17398
```

# Linear regression models outputs: residuals

Here we see  $e_i = y_i - \hat{y}_i$ , i.e. the **residual value for the i-th individual**

```
1 resid_val <- lr_model$residuals  
2  
3 # print the first 6 elements  
4 head(resid_val)
```

```
1           2           3           4           5           6  
-1.49196704  0.06748322 -3.96270002 -3.15599844 -2.49196704  3.75601726
```

# Linear regression model's fit: Residual standard error

- The **Residual standard error** (an estimate of the parameter  $\sigma$ ) tells the average distance that the observed values fall from the regression line (we are assuming constant variance).
  - The smaller it is, the better the model fits the dataset!*

We can compute it manually as:

$$SE_{\text{resid}} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{df_{\text{resid}}}}$$

```
1 # Residual Standard error (Like Standard Deviation)
2
3 # --- inputs
4 # sample size
5 n =length(lr_model$residuals)
6 # n of parameters in the model
7 k = length(lr_model$coefficients)-1 #Subtract one to ignore intercept
8 # degrees of freedom of the the residuals
9 df_resid = n-k-1
10 # Squared Sum of Errors
11 SSE =sum(lr_model$residuals^2) # 22991.19
12
13 # --- Residual Standard Error
14 ResStdErr <- sqrt(SSE/df_resid) # 6.815192
15 ResStdErr
```

[1] 6.815192

# Linear regression model's fit: : $R^2$ and Adj. $R^2$

The  $R^2$  tells us the **proportion of the variance in the response variable** that can be explained by the predictor variable(s).

- if  $R^2$  close to 0 -> data more spread
- if  $R^2$  close to 1 -> data more tight around the regression line

```
1 # --- R^2  
2 summary(lr_model)$r.squared
```

```
[1] 0.00237723
```

The Adj.  $R^2$  is a **modified version of  $R^2$**  that has been adjusted for the number of predictors in the model.

- It is always lower than the R-squared
- It can be useful for comparing the fit of different regression models that use different numbers of predictor variables.

```
1 # --- Adj. R^2  
2 summary(lr_model)$adj.r.squared
```

```
[1] 0.0003618303
```

# Linear regression model's fit: : F statistic

The **F-statistic** indicates whether the regression model provides a better fit to the data than a model that contains no independent variables. In essence, it tests if the regression model as a whole is useful.

```
1 # extract only F statistic
2 summary(lr_model)$fstatistic

value      numdf      dendf
1.179533  1.000000 495.000000

1 # define function to extract overall p-value of model
2 overall_p <- function(my_model) {
3   f <- summary(my_model)$fstatistic
4   p <- pf(f[1],f[2],f[3],lower.tail=F)
5   attributes(p) <- NULL
6   return(p)
7 }
8
9 # extract overall p-value of model
10 overall_p(lr_model)

[1] 0.2779797
```

Given the **p-value is > 0.05**, this indicate that *the predictor variable is not useful for predicting the value of the response variable*.

# DIAGNOSTIC PLOTS

The following plots help us checking if (most of) the assumptions of linear regression are met!

(the **Independence** assumption is more linked to the study design than to the data used in modeling)

# Linear regression diagnostic plots: residuals 1/4

**ASSUMPTION 1:** there exists a linear relationship between the independent variable,  $x$ , and the dependent variable,  $y$

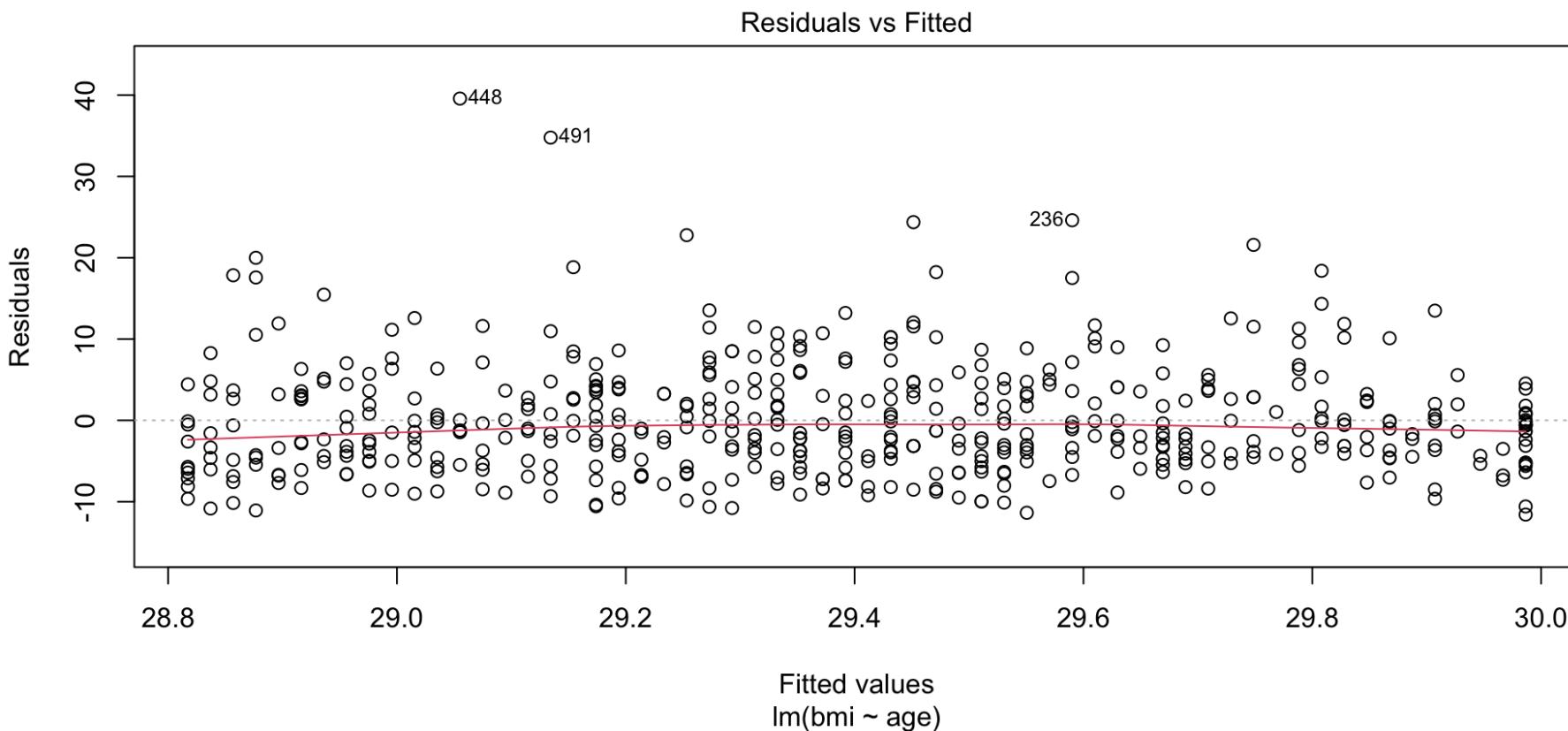
For an observation  $(x_i, y_i)$ , where  $\hat{y}_i$  is the **predicted value** according to the line  $\hat{y} = b_0 + b_1 x$ , the **residual** is the value  $e_i = y_i - \hat{y}_i$

- A linear (e.g. `lr_model`) is a particularly good fit for the data when the residual plot shows random scatter above and below the horizontal line.
  - (In this R plot, we look for a red line that is fairly straight)

```
1 # residual plot
2 plot(lr_model, which = 1 )
```

- We use the argument **which** in the function **plot** so we see the plots one at a time.

# Linear regression diagnostic plots: residuals 1/4



## Linear regression diagnostic plots: normality of residuals 2/4

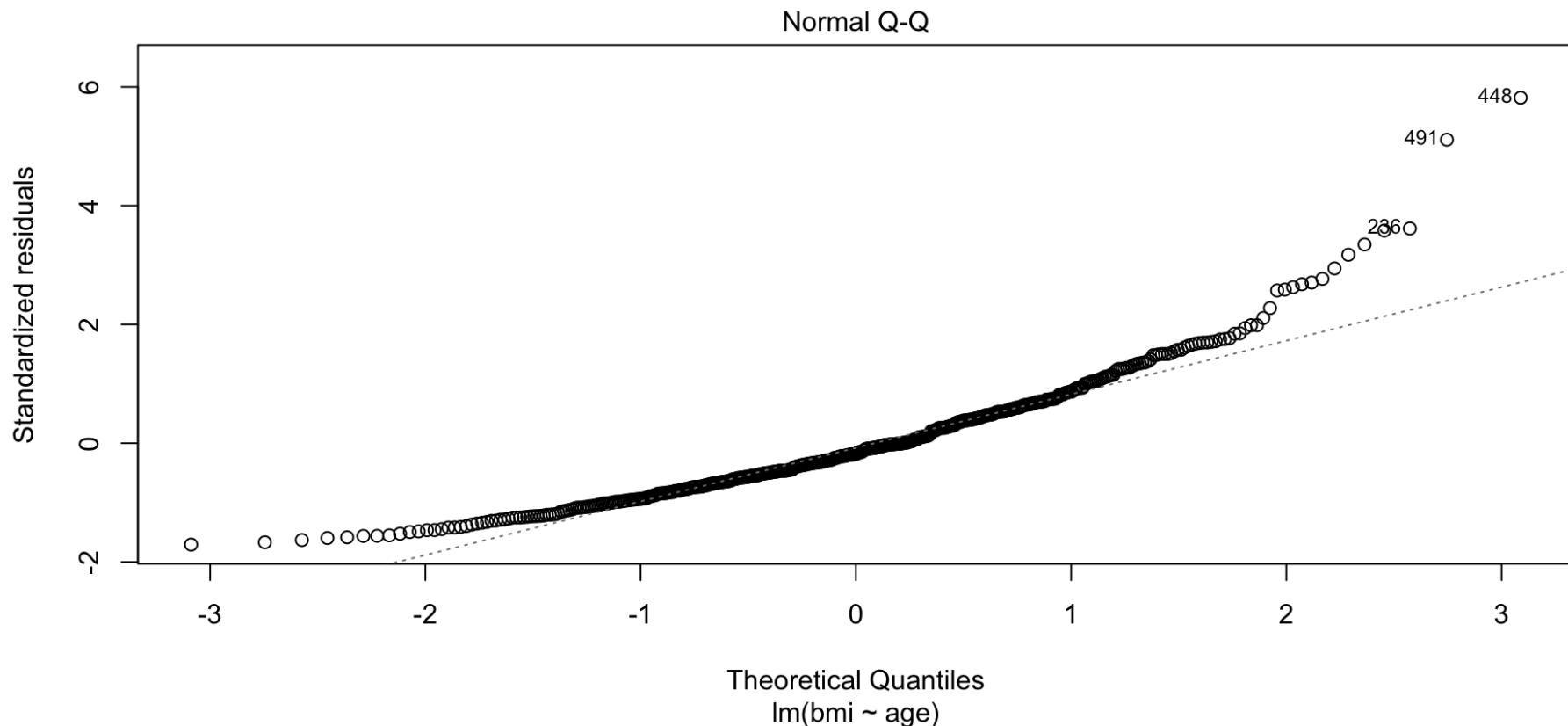
**ASSUMPTION 2:** The residuals of the model are normally distributed

With the quantile-quantile plot (Q-Q) we can checking normality of the residuals.

```
1 # quantile-quantile plot  
2 plot(lr_model, which = 2 )
```

## Linear regression diagnostic plots: normality of residuals 2/4

The data appear roughly normal, but there are deviations from normality in the tails, particularly the upper tail.



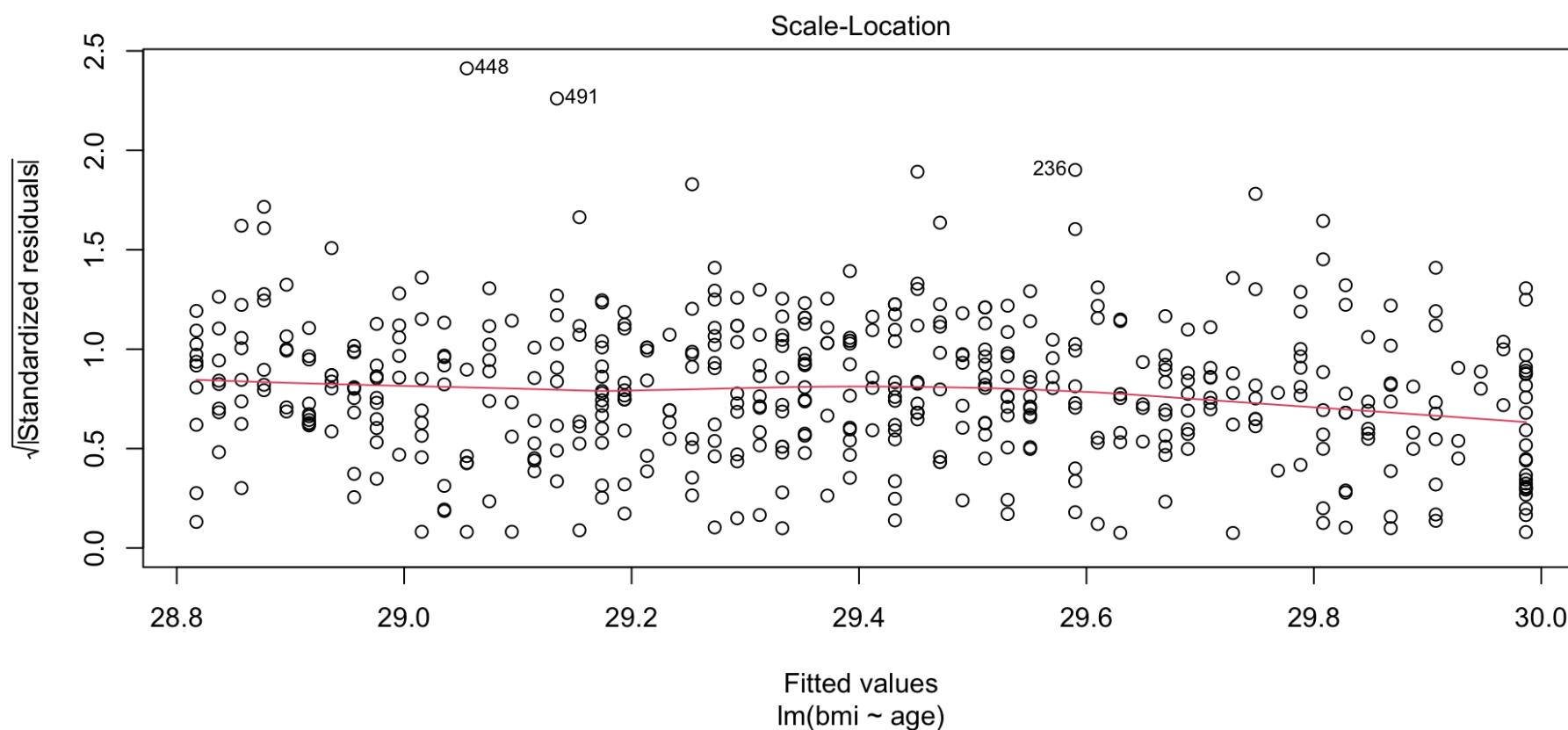
## Linear regression diagnostic plots: Homoscedasticity 3/4

**ASSUMPTION 3:** The residuals have constant variance at every level of x (“*homoscedasticity*”)

This one is called a **Spread-location plot**: shows if residuals are spread equally along the ranges of predictors

```
1 # Spread-location plot  
2 plot(lr_model, which = 3 )
```

## Linear regression diagnostic plots: Homoscedasticity 3/4



# Test for Homoscedasticity

Besides visual check, we can perform the **Breusch–Pagan test** to verify the assumption of homoscedasticity. In this case:

- $H_0$ : residuals are distributed with **equal variance**
- $H_1$ : residuals are distributed with **UNequal variance**
- we use **bptest** function from the **lmtest** package

```
1 # Breusch-Pagan test against heteroskedasticity
2 lmtest::bptest(lr_model)
```

```
studentized Breusch-Pagan test

data: lr_model
BP = 2.7548, df = 1, p-value = 0.09696
```

Because the test statistic (BP) is small and the p-value is not significant ( $p\text{-value} > 0.05$ ): **WE DO NOT REJECT THE NULL HYPOTHESIS** (i.e. we can assume equal variance)

# Linear regression diagnostic plots: leverage 4/4

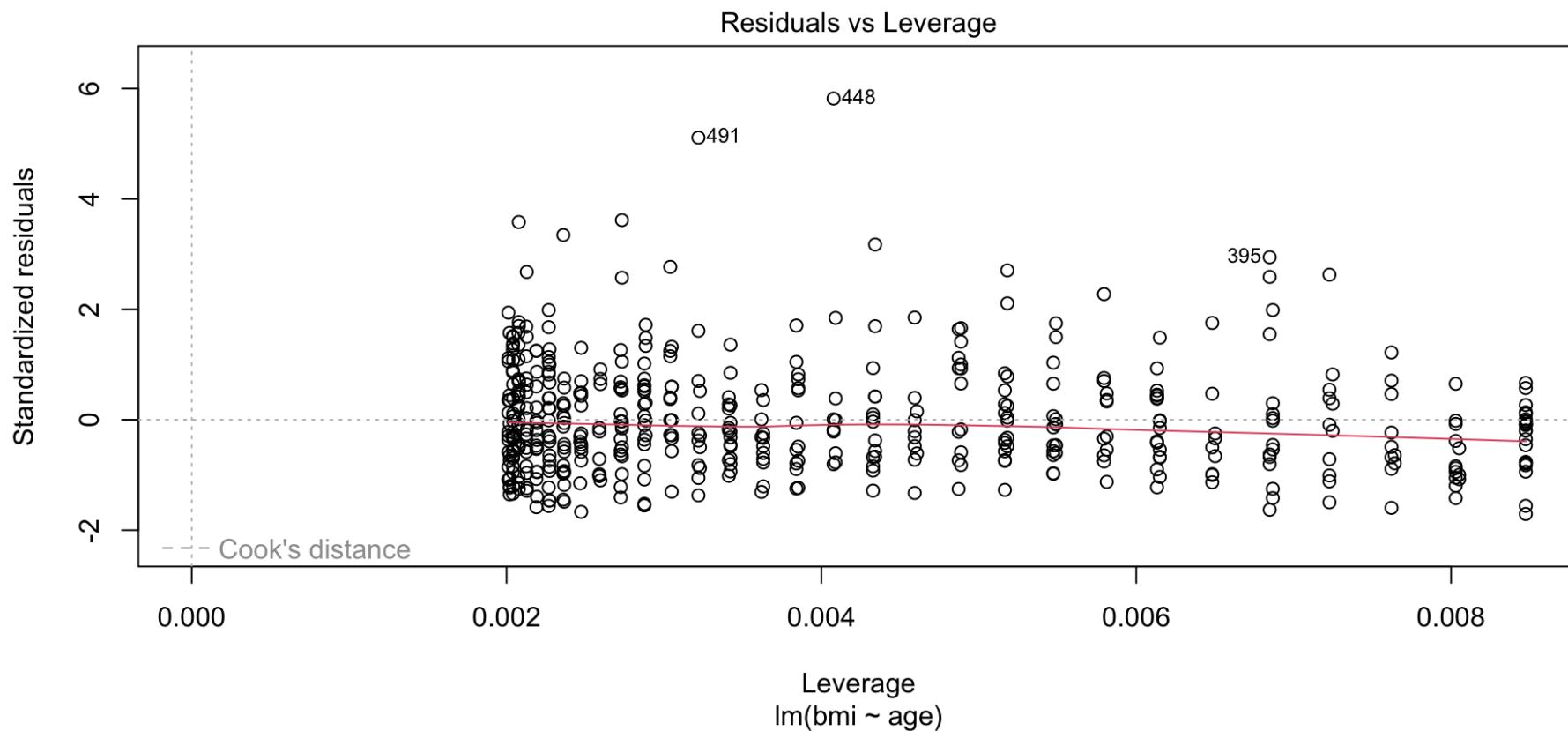
This last diagnostic plot has to do with **outliers**:

- A **residuals vs. leverage plot** allows us to identify *influential observations* in a regression model
  - The x-axis shows the “leverage” of each point and the y-axis shows the “**standardized residual of each point**”, i.e. “*How much would the coefficients in the regression model would change if a particular observation was removed from the dataset?*”
  - **Cook's distance lines** (red dashed lines) – not visible here – should appear on the corners of the plot when there are influential cases

```
1 plot(lr_model, which = 5 )
```

# Linear regression diagnostic plots: leverage 4/4

In this particular case, there is no influential case, or cases



# (Digression on the **broom** package)

- The **broom** package introduces the ***tidy approach*** to regression modeling code and outputs, allowing to convert/save them in the form of **tibbles**
- The function **tidy** will turn an object into a tidy tibble
- The function **glance** will construct a single row summary “glance” of a model, fit, or other object
- The function **augment** will show a lot of results for the model attached to each observation
  - this is very useful for further use of such objects, like **ggplot2** etc.

```
1 # render model as a dataframe
2 broom::tidy(lr_model)
3
4 # see overall performance
5 broom::glance(lr_model)
6
7 # save an object with all the model output elements
8 model_aug <- broom::augment(lr_model)
```

## You try...

Run these functions and then run **View(model\_aug)** to check out the output

# MULTIPLE LINEAR REGRESSION

[Using PREVEND dataset: a sample of 500 obs]

# Visualize the data: Statin use and cognitive function

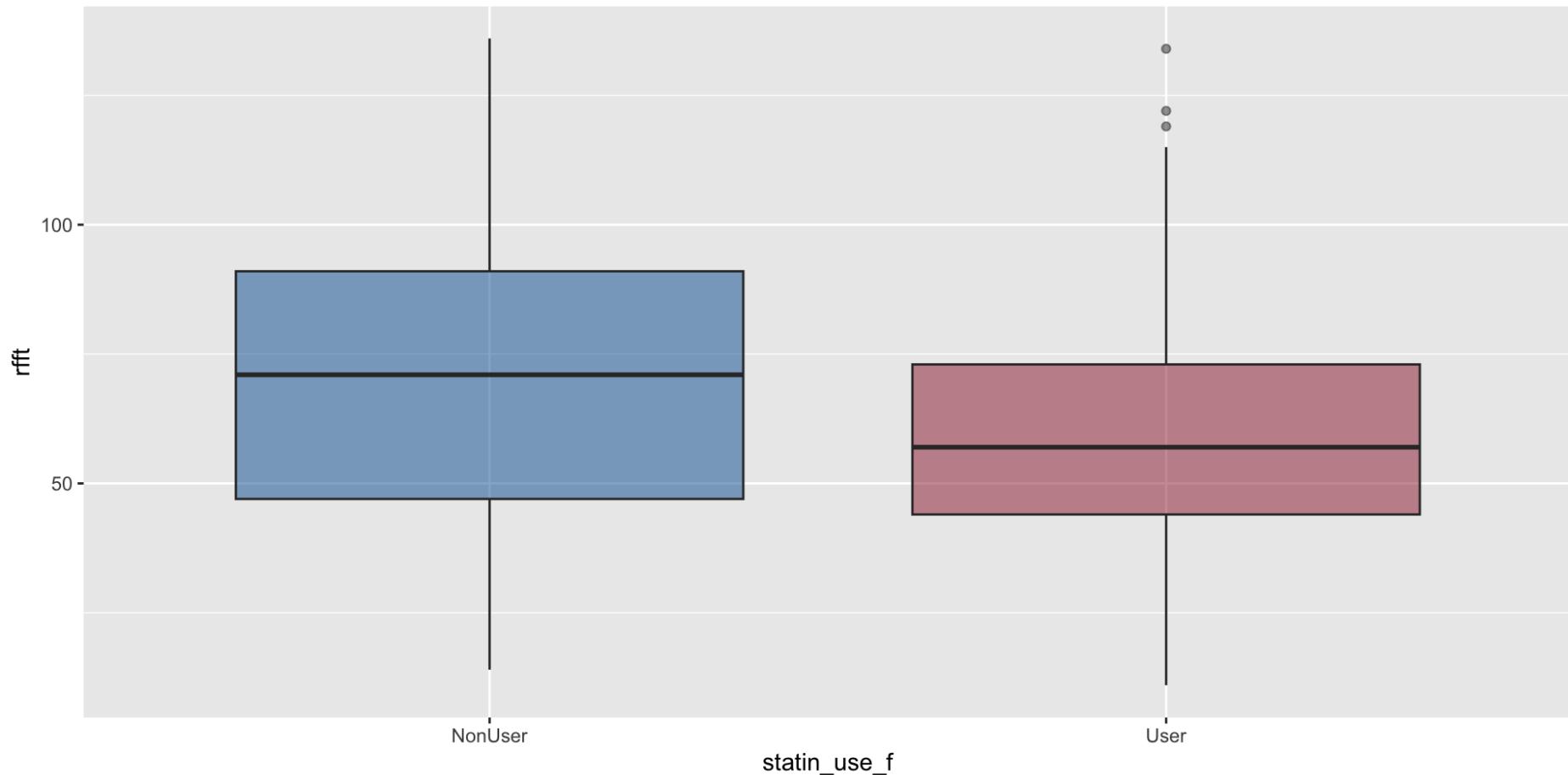
**Statins** are a class of drugs widely used to lower **cholesterol** (recent guidelines would lead to statin use in almost half of Americans between 40 - 75 years of age and nearly all men over 60). But a few small studies have suggested that statins may be associated with lower **cognitive ability**.

- From this sample of the PREVEND study, we can observe the relationship between **statin use** (`statin_use`) and **cognitive ability** (`rfft`).

```
1 # rename for convenience
2 prevend <- prevend_samp %>% janitor::clean_names() %>%
3   #create statin.use logical + factor
4   mutate(statin_use = as.logical(statin)) %>%
5   mutate(statin_use_f = factor(statin, levels = c(0,1), labels = c("NonUser", "User")))
6
7 # box plot
8 ggplot(prevend,
9       aes (x = statin_use_f, y = rfft, fill = statin_use_f)) +
10  geom_boxplot(alpha=0.5) +
11  scale_fill_manual(values=c("#005ca1","#9b2339" ))
12 # drop legend and Y-axis title
13 theme(legend.position = "none")
```

# Visualize the data: Statin use and cognitive function

The boxplot suggests that statin user (red) present lower cognitive ability score, on average



## Confirm visual intuition with independent sample t-test

We could use an independent t-test to confirm what the boxplot shows

```
1 t_test_w <- t.test(prevend$rfft[prevend$statin == 1],  
2                      prevend$rfft[prevend$statin == 0],  
3                      # here we specify the situation  
4                      var.equal = TRUE,  
5                      paired = FALSE, alternative = "two.sided")  
6  
7 t_test_w
```

```
Two Sample t-test  
  
data: prevend$rfft[prevend$statin == 1] and prevend$rfft[prevend$statin == 0]  
t = -3.4917, df = 498, p-value = 0.0005226  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -15.710276 -4.396556  
sample estimates:  
mean of x mean of y  
 60.66087 70.71429
```

...statistically significant difference in means (Statin use: yes and no) do exist

## Consider Simple Linear regression: Statin use and cognitive function

... and build a simple linear regression model like so:

$$E(RFFT) = b_0 + b_{\text{statin}}(\text{Statin use})$$

```
1 #fit the linear model
2 model_1 <- lm(rfft ~ statin, data=prevend)
3 summary(model_1)
```

```
Call:
lm(formula = rfft ~ statin, data = prevend)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.714 -22.714   0.286  18.299  73.339 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  70.714     1.381  51.212 < 2e-16 ***
statin       -10.053    2.879  -3.492 0.000523 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27.09 on 498 degrees of freedom
Multiple R-squared:  0.0239,    Adjusted R-squared:  0.02194 
F-statistic: 12.19 on 1 and 498 DF,  p-value: 0.0005226
```

- This preliminary model shows that, on average, **statin** users score approximately 10 points lower on the RFFT cognitive test (and the statin coefficient is **highly significant!**)

## Visualize the data: Statin use and cognitive function + age

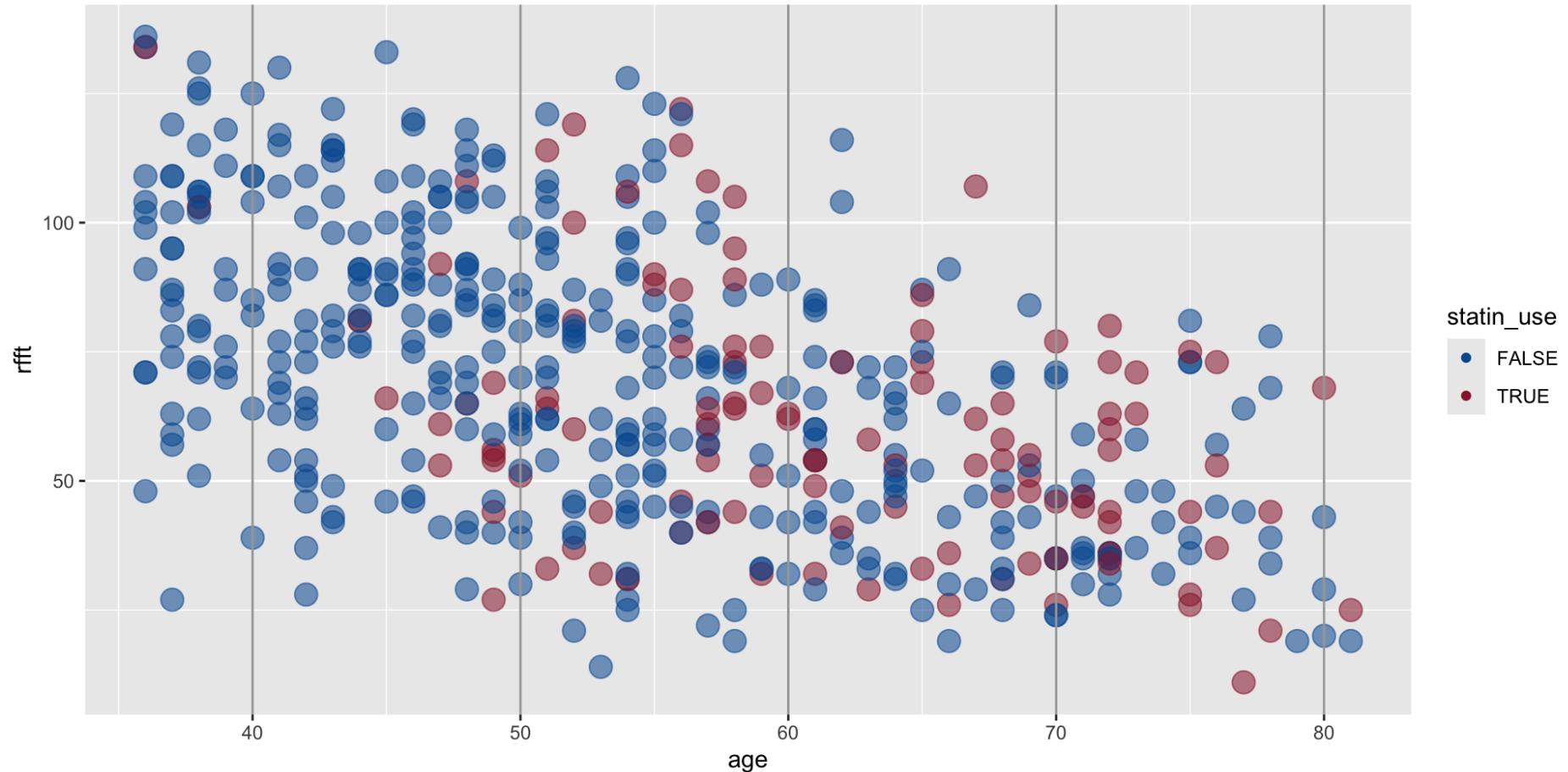
However, following the literature, this preliminary model might be misleading (**biased**) because it does not account for the underlying relationship between age and statin

- hence **age** could be a **confounder** within the **statin -> RFFT** relationship

```
1 ggplot(prevend,
2       aes (x = age, y = rfft, group = statin_use)) +
3     geom_point (aes(color = statin_use , size=.01, alpha = 0.75),
4                  show.legend = c(size = F, alpha = F) )+
5     scale_color_manual(values=c("#005ca1","#9b2339" )) +
6     # decades line separators
7     geom_vline(xintercept = 40, color = "#A6A6A6")+
8     geom_vline(xintercept = 50, color = "#A6A6A6")+
9     geom_vline(xintercept = 60, color = "#A6A6A6")+
10    geom_vline(xintercept = 70, color = "#A6A6A6")+
11    geom_vline(xintercept = 80, color = "#A6A6A6")
```

# Visualize the data: Statin use and cognitive function + age

Statin users are represented with red points; participants not using statins are shown as blue points



# Multiple linear regression model

Multiple regression allows for a (richer) model that incorporates both statin use and age:

$$E(RFFT) = b_0 + b_{\text{statin}}(\text{Statin use}) + b_{\text{age}}(\text{Age})$$

- or (*in statistical terms*) the association between **RFFT** and **Statin use** is being estimated **after adjusting** for **Age**

The R syntax is very easy: simply use **+** to add covariates

```
1 # fit the (multiple) linear model  
2 model_2 <- lm(rfft ~ statin + age , data=prevend)
```

# RFFT vs. statin use & age...

Although the use of statins appeared to be associated with lower RFFT scores when no adjustment was made for possible confounders, **statin use is not significantly associated with RFFT score in a regression model that adjusts for age.**

```
1 summary(model_2)
```

Call:

```
lm(formula = rfft ~ statin + age, data = prevend)
```

Residuals:

Min	1Q	Median	3Q	Max
-63.855	-16.860	-1.178	15.730	58.751

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	137.8822	5.1221	26.919	<2e-16 ***
statin	0.8509	2.5957	0.328	0.743
age	-1.2710	0.0943	-13.478	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.21 on 497 degrees of freedom

Multiple R-squared: 0.2852, Adjusted R-squared: 0.2823

F-statistic: 99.13 on 2 and 497 DF, p-value: < 2.2e-16

# Evaluating a multiple regression model

# Assumptions for multiple regression

Similar to those of simple linear regression...

1. **Linearity**: For each predictor variable  $x_j$ , change in the predictor is linearly related to change in the response variable when the value of all other predictors is held constant.
2. **Constant variability**: The residuals have approximately constant variance.
3. **Normality of residuals**: The residuals are approximately normally distributed.
4. **Independent observations**: Each set of observations  $(y, x_1, x_2, \dots, x_p)$  is independent.
5. **No multicollinearity**: i.e. no situations when there is a strong linear correlation between the independent variables, conditional on the other variables in the model

# Using residual plots to assess LINEARITY: age

**ASSUMPTION 1:** there exists a linear relationship between the independent variables,  $(x_1, x_2, \dots, x_p)$ , and the dependent variable,  $y$

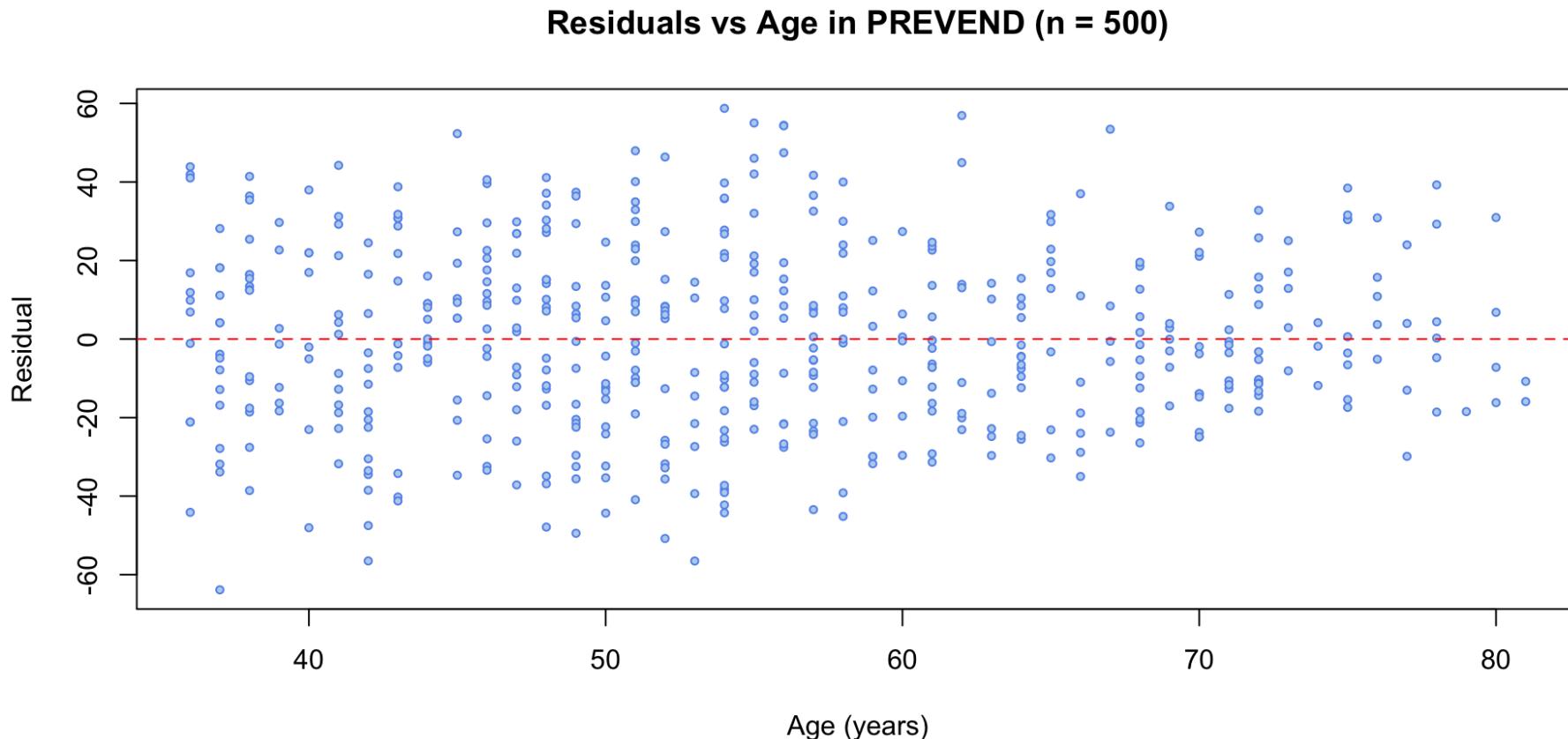
It is not possible to make a scatterplot of a response against several simultaneous predictors. Instead, use a **modified residual plot** to assess linearity:

- For **each** (numerical) predictor, plot the residuals on the y-axis and the predictor values on the x-axis.
- Patterns/curvature are indicative of non-linearity.

```
1 # recall
2 model_2 <- lm(rfft ~ statin + age , data=prevend)
3
4 # assess linearity
5 plot(residuals(model_2) ~ prevend$age,
6      main = "Residuals vs Age in PREVEND (n = 500)",
7      xlab = "Age (years)", ylab = "Residual",
8      pch = 21, col = "cornflowerblue", bg = "slategray2",
9      cex = 0.60)
10 abline(h = 0, col = "red", lty = 2)
```

# Using residual plots to assess LINEARITY: age

There are no apparent trends; the data scatter evenly above and below the horizontal line. There does not seem to be remaining nonlinearity with respect to age after the model is fit.



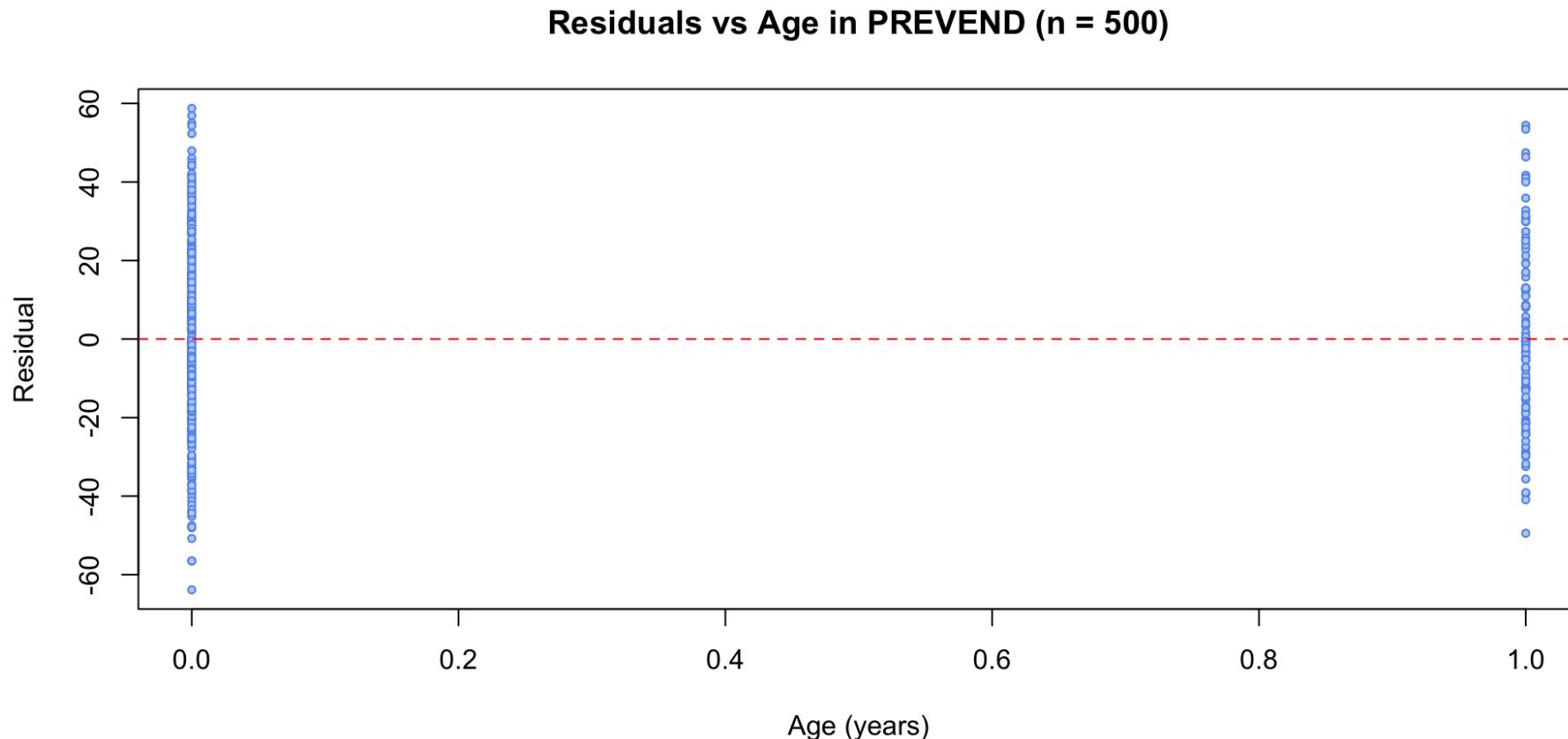
## Using residual plots to assess LINEARITY: statin use

Should we be testing linearity of residuals also against a **categorical variable (statin use)**? (not really, because not meaningful)

```
1 # recall
2 model_2 <- lm(rfft ~ statin + age , data=prevend)
3
4 #assess linearity
5 plot(residuals(model_2) ~ prevend$statin,
6       main = "Residuals vs Age in PREVEND (n = 500)",
7       xlab = "Age (years)", ylab = "Residual",
8       pch = 21, col = "cornflowerblue", bg = "slategray2",
9       cex = 0.60)
10 abline(h = 0, col = "red", lty = 2)
```

## Using residual plots to assess LINEARITY: statin use

It is not necessary to assess linearity with respect to statin use since statin use is measured as a categorical variable. A line drawn through two points (that is, the mean of the two groups defined by a binary variable) is necessarily linear



# Using residual plots to assess CONSTANT VARIABILITY

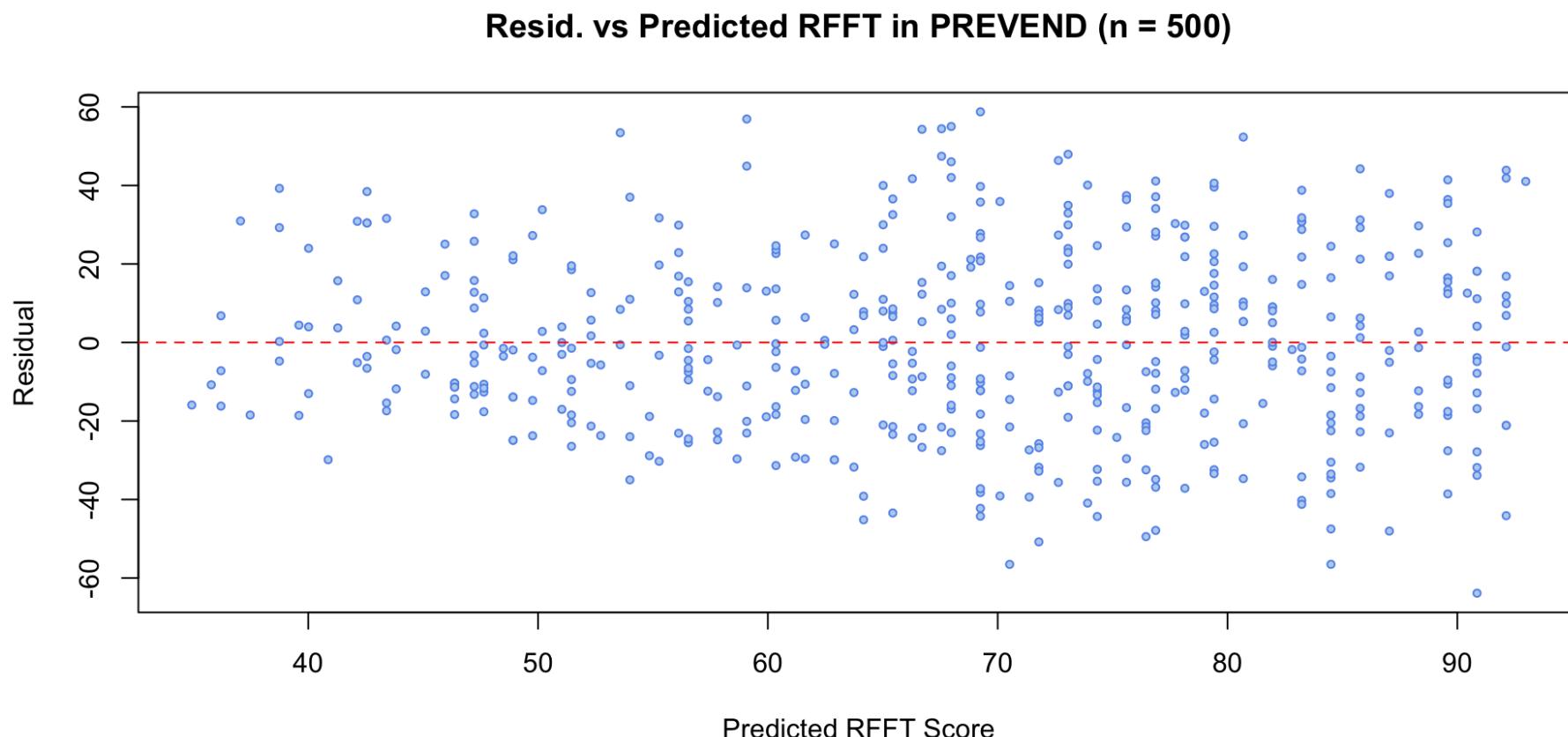
**ASSUMPTION 2:** The residuals have constant variance at every level of x (“*homoscedasticity*”)

- Constant variability: plot the residual values on the y-axis and the predicted values on the x-axis

```
1 #assess constant variance of residuals
2 plot(residuals(model_2) ~ fitted(model_2),
3       main = "Resid. vs Predicted RFFT in PREVEND (n = 500)",
4       xlab = "Predicted RFFT Score", ylab = "Residual",
5       pch = 21, col = "cornflowerblue", bg = "slategray2",
6       cex = 0.60)
7 abline(h = 0, col = "red", lty = 2)
```

## Using residual plots to assess CONSTANT VARIABILITY

The variance of the residuals is somewhat smaller for lower predicted values of RFFT score, but this may simply be an artifact from observing few individuals with relatively low predicted scores. It seems reasonable to assume approximately constant variance.



# Using residual plots to assess NORMALITY of residuals

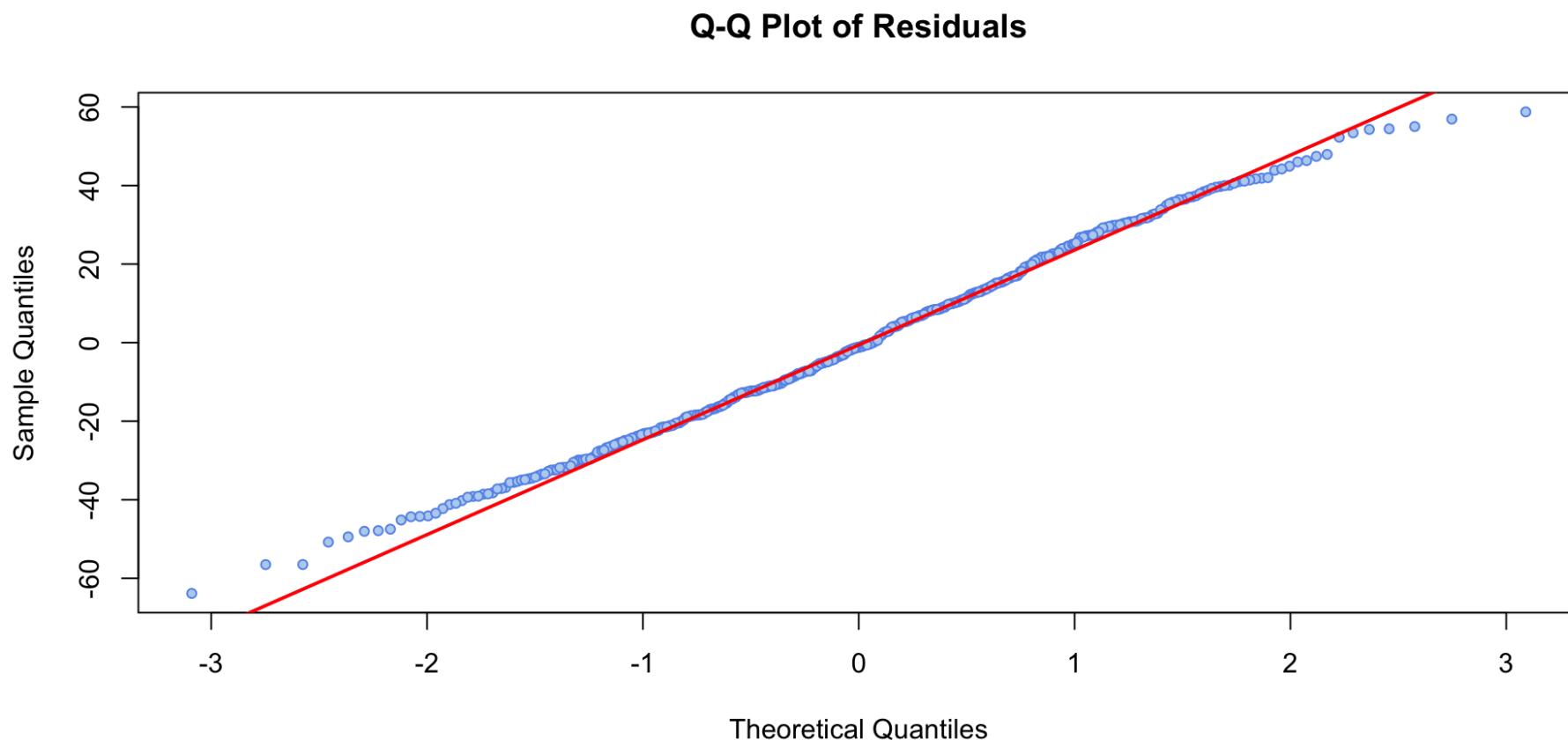
**ASSUMPTION 3:** The residuals of the model are normally distributed -  
Normality of residuals: use Q-Q plots

```
1 #assess normality of residuals
2 qqnorm(resid(model_2),
3         pch = 21, col = "cornflowerblue", bg = "slategray2", cex = 0.75,
4         main = "Q-Q Plot of Residuals")
5 qqline(resid(model_2), col = "red", lwd = 2)
```

In our example, we see that most data points are OK, except some observations at the tails. However, if all other plots indicate no violation of assumptions, some deviation of normality, particularly at the tails, can be less critical.

# Using residual plots to assess NORMALITY of residuals

The residuals are reasonably normally distributed, with only slight departures from normality in the tails.



# Assumption of INDEPENDENCE of observations

**ASSUMPTION 4:** Each set of observations  $(y, x_1, x_2, \dots, x_p)$  is independent.

Is it reasonable to assume that each set of observations is independent of the others?

Using the PREVEND data, it is reasonable to assume that the observations in this dataset are independent. The participants were recruited from a large city in the Netherlands for a study focusing on factors associated with renal and cardiovascular disease.

# Assumption of NO MULTICOLLINEARITY

**ASSUMPTION 5:** Each set of observations  $(y, x_1, x_2, \dots, x_p)$  is independent.

The R package **performance** actually provides a very helpful function **check\_model()** which tests these assumptions all at the same time

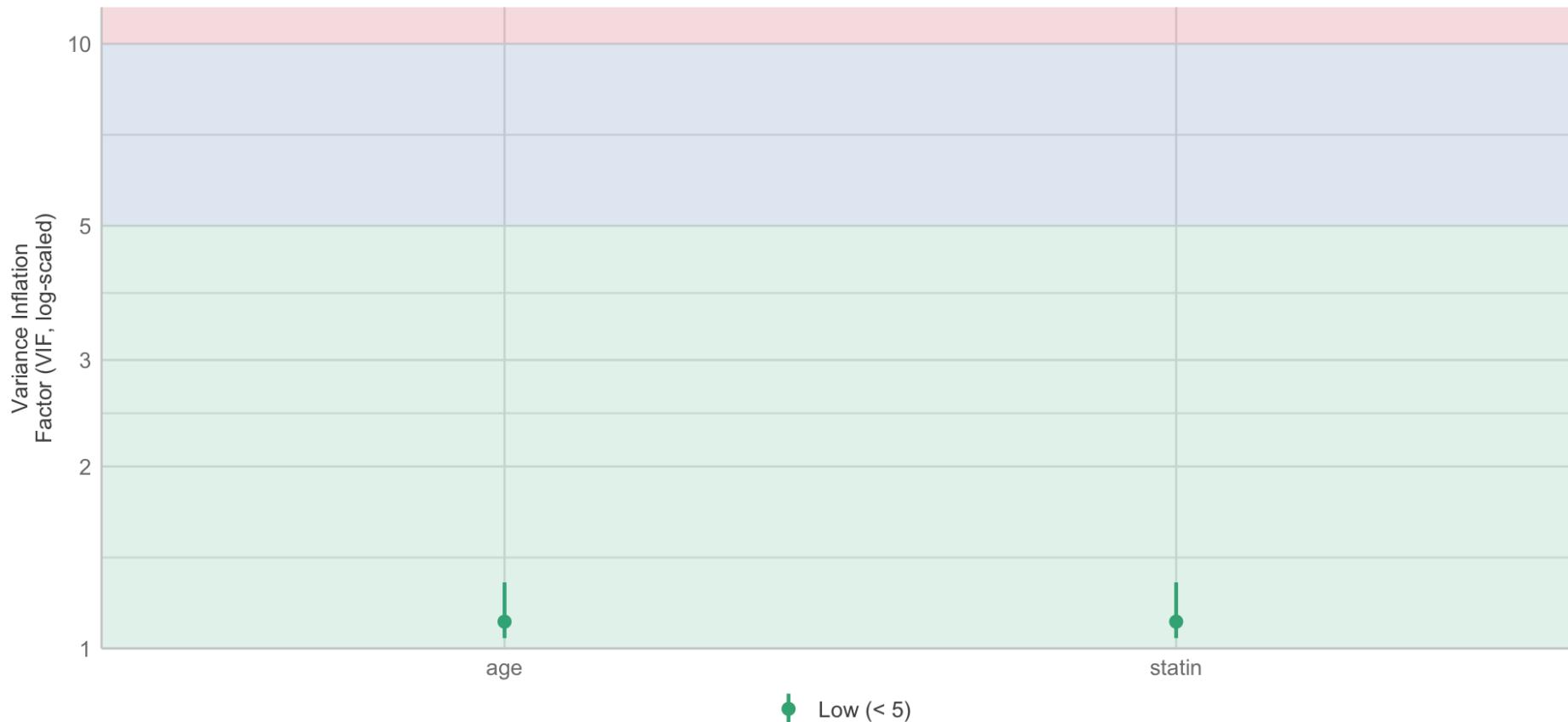
- **Multicollinearity** is not an issue (based on a general threshold of 10 for VIF, all of them are below 10)

```
1 # return and store a list of single plots
2 diagnostic_plots <- plot(performance::check_model(model_2, panel = FALSE))
3
4 # see multicollinearity plot
5 diagnostic_plots[[5]]
```

# Assumption of NO MULTICOLLINEARITY

## Collinearity

High collinearity (VIF) may inflate parameter uncertainty



# Checking out the performance R package

- Find more info on the helpful **performance** R package [here](#) for verifying assumptions and model's quality and goodness of fit.

## » You try...

Run also the following commands

- Diagnostic plot of linearity `diagnostic_plots[[2]]`
- Diagnostic plot of influential observations - outliers `diagnostic_plots[[4]]`
- Diagnostic plot of normally distributed residuals `diagnostic_plots[[6]]`

# $R^2$ with multiple regression

As in simple regression,  $R^2$  represents the proportion of variability in the response variable explained by the model.

- As variables are added,  $R^2$  always increases.

In the `summary(lm( ))` output, **Multiple R-squared** is  $R^2$ .

```
1 #extract R^2 of a model  
2 summary(model_2)$r.squared  
  
[1] 0.2851629
```

The  $R^2$  is 0.285; **the model explains 28.5% of the observed variation in RFFT score.** The moderately low  $R^2$  suggests that the model is missing other predictors of RFFT score.

# Adjusted R<sup>2</sup> as a tool for model assessment

The **adjusted R<sup>2</sup>** is computed as:

$$R_{adj}^2 = 1 - \left( \frac{\text{Var}(e_i)}{\text{Var}(y_i)} \times \frac{n - 1}{n - p - 1} \right)$$

- where n is the number of cases and p is the number of predictor variables.

Adjusted R<sup>2</sup> incorporates a penalty for including predictors that do not contribute much towards explaining observed variation in the response variable.

- It is often used to balance predictive ability with model complexity.
- Unlike R<sup>2</sup>, R<sup>2</sup><sub>adj</sub> does not have an inherent interpretation.

```
1 #extract adjusted R^2 of a model  
2 summary(model_2)$adj.r.squared
```

```
[1] 0.2822863
```

# **INTRODUCING SPECIAL KINDS OF PREDICTORS**

# Categorical predictor in regression - (example)

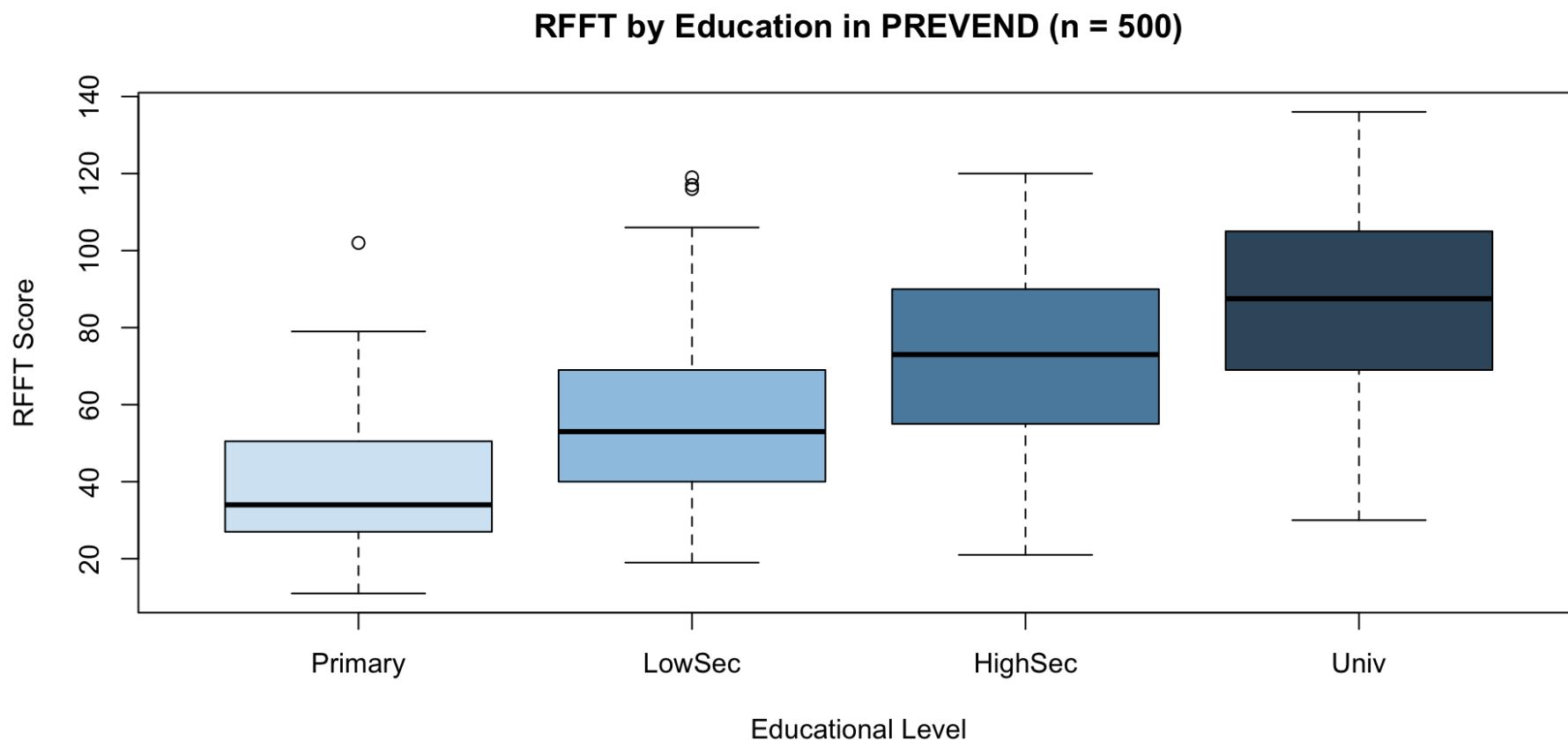
Is RFFT score associated with **education**? The variable **Education** in the **PREVEND** dataset indicates the highest level of education an individual completed in the Dutch educational system:

- 0: primary school
  - 1: lower secondary school
  - 2: higher secondary education
  - 3: university education

```
1 # convert Education to a factor
2 prevend <- prevend %>%
3   mutate(educ_f = factor(education,
4                         levels = c(0, 1, 2, 3),
5                         labels = c("Primary", "LowerSecond",
6                                   "HigherSecond", "Univ")))
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
797
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
948
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1007
1008
1009
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2170
2171
2171
2172
2172
2173
2173
2174
2174
2175
2175
2176
2176
2177
2177
2178
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2270
2271
2271
2272
2272
2273
2273
2274
2274
2275
2275
2276
2276
2277
2277
2278
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
234
```

# Categorical predictor in regression - (example)

A very clear association seems to exist between education level and average RFFT score in the sample



# Categorical predictor in regression - model

Calculate the average RFFT score in the sample across education levels

```
1 # calculate group means
2 prevend %>%
3   group_by(educ_f) %>%
4   summarise(avg_RFFT_score = mean(rfft))

# A tibble: 4 × 2
  educ_f      avg_RFFT_score
  <fct>        <dbl>
1 Primary       40.9
2 LowerSecond   55.7
3 HigherSecond  73.1
4 Univ          85.9
```

Fitting a model with **education** as a predictor

```
1 # fit a model
2 model_cat <- lm(rfft ~ educ_f, data = prevend)
3 model_cat$coefficients

(Intercept)  educ_fLowerSecond educ_fHigherSecond      educ_fUniv
  40.94118           14.77857            32.13345          44.96389
```

- Notice how **Primary** level of **educ\_f** does NOT appear as a coefficient

# Categorical predictor in regression - model interpretation

```
Call:  
lm(formula = rfft ~ educ_f, data = prevend)
```

Residuals:

Min	1Q	Median	3Q	Max
-55.905	-15.975	-0.905	16.068	63.280

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	40.941	3.203	12.783	< 2e-16 ***
educ_fLowerSecond	14.779	3.686	4.009	7.04e-05 ***
educ_fHigherSecond	32.133	3.763	8.539	< 2e-16 ***
educ_fUniv	44.964	3.684	12.207	< 2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.87 on 496 degrees of freedom  
Multiple R-squared: 0.3072, Adjusted R-squared: 0.303  
F-statistic: 73.3 on 3 and 496 DF, p-value: < 2.2e-16

The baseline category represents individuals who at most completed primary school **Education = 0**.  
The coefficients represent the change in estimated average RFFT relative to the baseline category.

- **(Intercept)** is the sample mean RFFT score for these individuals, 40.94 points
- An increase of 14.78 points is predicted for **LowerSecond** level,  $40.94 + 14.78 = 55.72$  points
- An increase of 32.13 points is predicted for **HigherSecond** level,  $40.94 + 32.13 = 73.07$  points
- An increase of 44.96 points is predicted for **Univ** level,  $40.94 + 44.96 = 85.90$  points

## Interaction in regression - (example) - NHANES

Let's go back to the **NHANES** dataset and consider a linear model that predicts **total cholesterol level (mmol/L)** from **age (yrs.)** and **diabetes status**.

The multiple regression model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$$

assumes that when one of the predictors  $x_j$  is changed by 1 unit and the values of the other variables remain constant, the predicted response changes by  $\beta_j$ , *regardless of the values of the other variables*.

- With statistical **interaction**, this assumption is not true, such that *the effect of one explanatory variable  $x_j$  on the y depends on the particular value(s) of one or more other explanatory variables*.

# Interaction in regression - visual

Fitting a model with **age** and **diabetes** as independent predictors (i.e. WITHOUT interaction terms)

```
1 # fit a model
2 model_NOinterac <- lm(tot_chol ~ age + diabetes, data = nhanes)
3 model_NOinterac$coefficients
```

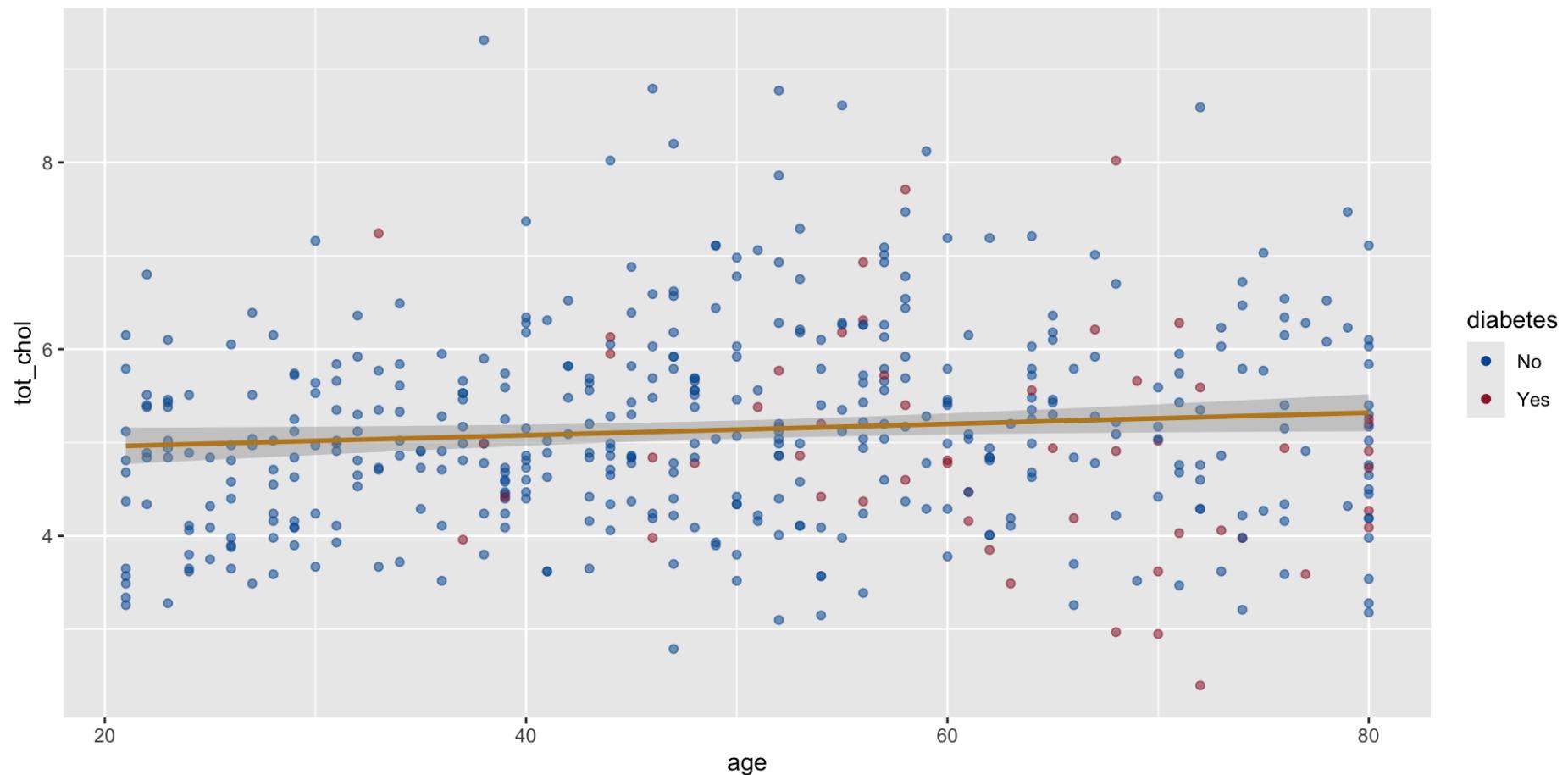
	(Intercept)	age	diabetesYes
	4.800011340	0.007491805	-0.317665963

- Using **geom\_smooth** for a visual intuition of a linear relationship
  - ! here I consider sample DATA **as a whole** for plotting a smooth line

```
1 ggplot(nhanes,
2         aes (x = age, y = tot_chol)) +
3     # For POINTS I split by category (category)
4     geom_point (aes(color = diabetes,
5                      alpha = 0.75),
6                  show.legend = c(size = F, alpha = F) )+
7     scale_color_manual(values=c("#005ca1","#9b2339" )) +
8     # For SMOOTHED LINES I take ALL data
9     geom_smooth(colour="#BD8723", method = lm)
```

# Interaction in regression - visual

Users in two categories are represented points; linear relationship is represented by ONE golden line for ALL SAMPLE



# Interaction in regression - visual (RETHINKING)

Suppose two separate models were fit for the relationship between total cholesterol and age; one in diabetic individuals and one in non-diabetic individuals.

- Using `geom_smooth` for a visual intuition of a linear relationship
  - ⚠ here I consider sample DATA **as 2 separate groups** for plotting a smooth line

```
1 ggplot(nhanes,
2         # For *both POINTS & LINES* I split by category (category)
3         aes (x = age, y = tot_chol, color = diabetes)) +
4         geom_point (aes(alpha = 0.75),
5                     show.legend = c(size = F, alpha = F) )+
6         geom_smooth(method = lm) +
7         scale_color_manual(values=c( "#005ca1", "#9b2339" ))
```

# Interaction in regression - visual (RETHINKING)

Users in two categories are represented points; linear relationship is represented by 2 respective line according to diabetes status... the association has DIFFERENT DIRECTION!



# Interaction in regression - adding term in model

Let's rethink the model and consider this new *specification*:

$$E(\text{TotChol}) = \beta_0 + \beta_1(\text{Age}) + \beta_2(\text{Diabetes}) + \beta_3(\text{Diabetes} \times \text{Age}).$$

Where: + the term  $(\text{Diabetes} \times \text{Age})$  is the **interaction term** between **diabetes** status and **age**, and  $\beta_3$  is the coefficient of such interaction term.

- notice the use of **...\*...** in the model syntax

```
1 #fit a model
2 model_interac2 <- lm(tot_chol ~ age*diabetes, data = nhanes)
3 model_interac2$coefficients
```

	(Intercept)	age	diabetesYes	age:diabetesYes
	4.695702513	0.009638183	1.718704342	-0.033451562

# Interaction in regression - prediction model

We obtained this predictive model:

$$\widehat{\text{TotChol}} = 4.70 + 0.0096(\text{Age}) + 0.172(\text{Diabetes}) - 0.033(\text{Age} \times \text{Diabetes})$$

```
1 summary(model_interac2)
```

```
Call:  
lm(formula = tot_chol ~ age * diabetes, data = nhanes)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-2.3587 -0.7448 -0.0845  0.6307  4.2480  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 4.695703  0.159691 29.405 < 2e-16 ***  
age          0.009638  0.003108  3.101  0.00205 **  
diabetesYes 1.718704  0.763905  2.250  0.02492 *  
age:diabetesYes -0.033452  0.012272 -2.726  0.00665 **  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 1.061 on 469 degrees of freedom  
(27 observations deleted due to missingness)  
Multiple R-squared:  0.03229,   Adjusted R-squared:  0.0261  
F-statistic: 5.216 on 3 and 469 DF,  p-value: 0.001498
```

## Interaction in regression - interactive term interpretation

Given:

$$\widehat{\text{TotChol}} = 4.70 + 0.0096(\text{Age}) + 0.172(\text{Diabetes}) - 0.033(\text{Age} \times \text{Diabetes})$$

For diabetics (**DiabetesYes = 1**), the model equation is:

$$\begin{aligned}\text{TotChol}_{\text{diab}} &= 4.70 + 0.0096(\text{Age}) + 1.72(1) - 0.034(\text{Age})(1) \text{ i.e.} \\ \text{TotChol}_{\text{diab}} &= 6.42 - 0.024(\text{Age})\end{aligned}$$

For non-diabetics (**DiabetesYes = 0**), the model equation is:

$$\begin{aligned}\text{TotChol}_{\text{NOdiab}} &= 4.70 + 0.0096(\text{Age}) + 1.72(0) - 0.034(\text{Age})(0) \text{ i.e.} \\ \text{TotChol}_{\text{NOdiab}} &= 4.70 + 0.0096(\text{Age})\end{aligned}$$

# Final thoughts/recommendations

- The analyses proposed in this Lab are very similar to the process we go through in real life. The following steps are always included:
  - Thorough **understanding of the input data** and the data collection process
  - Bivariate **analysis of correlation / association** to form an intuition of which explanatory variable(s) may or may not affect the response variable
  - **Diagnostic plots** to verify if the necessary assumptions are met for a linear model to be suitable
  - Upon verifying the assumptions, we **fit data** to hypothesized (linear) model
  - **Assessment of the model performance** ( $R^2$ , Adj.  $R^2$ , F – Statistic, etc.)
- As we saw with hypothesis testing, the **assumptions** we make (and require) for regression are of utter importance
- Clearly, we only scratched the surface in terms of all the possible predictive models, but we got a hang of the **fundamental steps** and some **useful tools** that might serve us also in more advanced analysis
  - e.g. **broom** (within **tidymodels**), **performace rstatix**, **lmtest**