stats-nutshell

Sebastian Sauer

9/04/2022

Table of contents

Preface

Figure 1: A nutshell of little (statistics) stars

Welcome!

This is an introductory course on statistical modelling. Welcome!

The focus of this course is on how to specify a theoretical idea (possibly vague) in a testable statistical model.

PDF-Version

There's a PDF version of this book available. Note that the HTML is the more recent one.

Course description

Analyzing research data can broadly be classified in three parts: explorative data analysis, modeling (including inference), and visualization. Either part is pivotal in its own right, but it can be argued that modeling is at the core of the scientific endeavor. However, in practice, modeling, visualization, and data exploration is heavily intertwined, so that three parts may be recognized (as individual entities) but not usefully separated from each other. This idea provides the rationale of this course: Data exploration, data visualization and data modeling is discussed as an integrated framework.

The focus is on practical data analysis; theoretical concepts are, where mentioned, second class citizens due to time constraints and the didactic aims of the course.

For example, statistical inference – such as p-values and confidence intervals – are not more than touched briefly, as the instructor believes that modeling, not inference, is of prime importance for the auditorium.

We will use the R environment for all computations (freely available). Please bring your own Laptop with R and RStudio installed (installation guides are provided). Data and R code will be provided.

We're on a crash course

The course is set-up as a "crash course" which indicates that we'll rather try to cover a breadth of steps rather than digging deep at certain particular points. The rationale of this approach is that before digging deep, it is necessary to gain an overview of the territory. In addition, if one particular topic is not of interest to a given student (perhaps to difficult/simple), not much time is lost.

Be warned! Compare this crash course to a dancing crash course right before your wedding: A lot can be achieved by such a course in some instances, or rather, the worst consequences (of not knowing how to dance) may be fenced off, but one should not expect to be a dancing queen (king) thereafter.

More on modelling

Models and modeling are of pivotal importance in many sciences, not only for providing an explanation of nature en miniature (theoretical models), but also for gauging how closely the empirical data at hand match the theoretical model. Translating a theoretical model into statistical language is called statistical modeling and provides the guiding principle in this introductory course. Regression models will be presented as a lingua franca of statistical modeling, and we will learn that many empirical questions can (comfortably) be analyzed using a regression framework. Depending on the background and aims of the participants (and time permitting), we will shed light on some standard topics such as model comparison, classification models, and typical pitfalls. Given a more advanced auditorium, we may want to explore how causal and non-causal associations can be translated and tested using simple linear statistical models. Foundational ideas of statistical modeling will be accompanied by short examples and case studies to facilitate transfer and practical application after the course.

Course prerequisites

Basic computer usage knowledge is needed (downloading materials from the internet, operating a PC, etc). Basic R knowledge is needed. Basic knowledge of statistical concepts (such as descriptive statistics) is needed. Willingness to learn is essential.

Learning objectives

Upon successful completion of this course, students should be able to:

- select the right statistical visualization for a variety of data contexts
- "crunch" or "wrangle" data
- explain what statistical modeling means
- formulate basic statistical models
- differentiate between predictive and explanatory modeling
- apply the methods to own datasets

Course Literature

This course builds on the freely available e-book ModernDive. Each topic is paralleled by an ackompagnying chapter from ModernDive. A hard copy can be purchased here. The book is for sale in print here.

Course logistics

This course can be presented as a one-day seminar or split-up in four blocks.

The course can be held in English or German.

Please *bring your own computer* and *read the notes* regarding course logistics in advance. Note that some *upfront preparation is needed* from the learners.

R and RStudio[1] will be needed throughout the course. Please make sure that the IT is running. In case of technical difficulties with R feel free to use RStudio Cloud; free plans are available.

All learning materials (such as literature, code, data) will be provided in electronic format.

[1] Desktop version, not the server

UPFRONT student preparation

- *Install R and RStudio*, see ModernDive Chap. 1.1. In case you have your R running on your system, please make sure that you're uptodate. If outdated, download and install the most recent versions of the software. Similarly, hit the "Update" button in RStudio's "Packages" tab to update your packages if you have not done so for a couple of months.
- Sign-in at RStudio Cloud. It's super helpful because I as the techer can provide you with an environment where all R stuff is ready to use (packages installed etc).
- Install the necessary R packages as used in the book chapters covered in this course (see the sections on "Needed packages" in each chapter). If in doubt, see here the instructions on how to install R packages. Here's the actual list on the R packages we'll need.
- Students new to R are advised to learn the basics, see ModernDive, Chap 1.2 - 1.5
- Bring your own laptop
- Make sure your internet connection is stable and your loudspeaker/headset is working; a webcam is helpful.
- Students are advised to review the course materials after each session.
- I recommend that you carefully check the course description to make sure the course fits your needs (not too advanced/basic).

Didactic outline

This course can rather be considered a workshop in the sense that the instructor uses a dialogue-based approach to teaching and that there are numerous exercises during the course. Instead of providing long talks to the students, the instructor feels obligated to engage students in back-and-forth conversations. Similarly, the presentation of a large number of Powerpoint slide is avoided. Instead, a thorough course literature is available (free online), so that students will have no barrier in diving deeply into the materials and ideas presented. However, during class it is more important to transmit the pivotal ideas; details need to be read and worked by the students individually after (and before) the course. As an alternative to presenting a lot of text on slides, in this course there will be a (electronic) whiteboard where concepts are developed dynamically and in pace of the teaching conversation thereby adjusting the "dose" of new thoughts to the actual pace of the instruction.

Schedule

Overview on topics covered

- Data Visualization using the grammar of graphics and ggplot2
- Data Wrangling based on the tidyverse in R
- Basic concepts of statistical modelling
- Primer on causal inference
- Introduction to regression analysis

Block 1: Explorative Data Analysis

Visualization

- Data *visualization*, see ModernDive Chap. 2, and get the R code here
  - Exploring common types of statistical diagrams, the "5NG"
  - Discussing when (not) to use diagrams see Anscombe's Quartett, and when to use which one
  - Building elegant graphics in R

Data Wrangling

- Data *wrangling*, see ModernDive Chap. 3, and get the R code here
  - A taxonomy of typical data operations
  - How to perform common data operations with R
  - Summarizing data (aka computing descriptive statistics)

Exercises / Case study

- Exercises
  - Exercises on life expectancy.
  - Case study on the visualization of flight delays
  - Advanced case study on one hit wonders
  - Visualization covid cases
  - Case study on nominal data: Survival on the Titanic
  - Inspiration for own project: Visualize Covid-19 cases from this source.

Block 2: Statistical Modelling: Basic

Theory

- Basics of *modelling*, see ModernDive Chap. 5.0, and get the R code here
  - What is modelling?
  - Basic terminology
  - Prediction vs. explanation
- Some thoughts on *causal inference*, see ModernDive Chap. 5.3.1
- *Regression* with one numerical predictor, see ModernDive Chap. 5.1
- Regression with one categorical predictor, see ModernDive Chap. 5.2
- Assessing *model fit* (using (adjusted) R^2), see ModernDive Chap. 5.3.2
- For some tips and tricks on typical issues, see ModernDive tips and tricks

Case study

- Exercises/Case studies:
  - Prices of Boston houses, first part
  - Modeling movie succes, first part

Block 3: Statistical Modelling: Multiple Regression and interaction

Theory

- Slightly more advanced topics on linear regression such as *multiple regression and interaction*, see ModernDive Chap. 6, and get the R code here
- One numerical and one categorical predictor, see ModernDive Chap. 6.1
- Two numerical predictors, see ModernDive Chap. 6.2
- Simpson's paradox and more on causal inference, see ModernDive Chap. 6.3.3

Case study

- Exercises/Case studies:
  - Prices of Boston houses, second part
  - Modeling movie succes, second part
  - Modeling flight delays

Block 4: Project coaching

- This session is dedicated to work on real projects brought in by the students.
- In addition, open questions regarding the presented concepts are being discussed.

Instructor

Sebastian Sauer works as a professor at Ansbach university, teaching statistics and related stuff. Analyzing data to answer questions related to social phenomena is one of his major interests. He is trying to help raising the methological (and particularly statistical) skills in the sciences (ie., scientists). The programming language "R" is one of his favorite tools. He sees himself as a learner, and is particularly interested learning more on quantitative approaches to understand nature. Open Science is a hot topic to him. He hopes to contribute to pressing social problems such as populism by bringing in his statistical and psychological know-how. He writes a blog which serves as a sketchpad for stuff in his mind (not immune to thought updates) at https://data-se.netlify.app/. Sebastian is the author of "Moderne Datenanalyse mit R" (Sauer 2019). His publication list is available on Google Scholar.

Contact me

Feel free to contact me via email at `sebastiansauer1@gmail.com`.

Assessment and grades

There is no assessment, there are no grades!

Talk to me

It's my goal to make this an excellent course and a stimulating and enjoyable experience for all of us. So that I can find out if this is happening, I encourage feedback—be it positive or negative—on all aspects of the course at any time. For example, if something I'm doing is making it difficult for you to learn, then let me know before it's too late; if you particularly enjoyed something we did in class, say so so that we can do it again.

Course materials

Most of the materials as presented below is made available through the course book Modern-Dive. Please check the relevant chapters of the book before the course to make sure you have all materials available.

Licence

Resources

Recommendations

- RStudio Cheatsheets, particularly on data wrangling, and data vizualization
- Book R for Data Science as a handy reference or a serious text book.
- Tidy Tuesday video series
- Post your open question on Stack Overflow.
- Follow `#rstats` hashtag on Twitter.

For students willing to learn more and go deeper (than the concepts explored in the present course), this book on regression modelling, and this book on statistical learning are recommended. For German folks, check out my book on modern data analysis.

Suggested literature for deepening the analytic skills include Statistical Rethinking. For an introduction to graphical causal models, check out Julia Rohrer's paper. For a more in-depth journey, consider reading this book. While I wholeheartedly recommend such books, we will not be able to discuss many of the ideas presented therein in class (in this course) due to time constraints.

R Packages

All R packages are accessible through the course book; please consult the relevant chapters. Please install all R packages used before the course. Here's a tutorial on how to install R packages.

The most important R packages for this course are:

- tidyverse
- easystats

The following packages are useful for data access (but not strictly mandatory):

- gapminder
- nycflights13
- fivethirtyeight
- skimr
- ISLR

For the Bayes models you'll need some extra software (free, save and stable), but somewhat more hassle to install. Using Bayes in this course is *optional*. You don't miss a lot if you don't use it.

- rstanarm

For the R package `{rstanarm}` to run, you'll need to install RStan. On Windows, this amounts to installing RTools. On Mac, you'll need to install the XCode CLI[2].

In sum, follow the instructions on the RStan website. It's unfortunately a bit complicated.

Data

All data are accessible through the course book; please consult the relevant chapters.

[2]possibly you need also a Fortran compiler, but maybe that's optional

Labs (case studies)

Practical data analysis skills can be practiced using these labs; in addition Chapter 11 provides two cases studies. Note that such content may be used as homework.

There are a lot of case studies scattered on the internet.

Sketching causal models

Dagitty is great tool for sketching causal graphs (DAGs), it can be usd in your browser or as R package. Here's an example of a collider bias. Check out this post for an intuitive explanation.

German introductary course

Readers who speak German may check out this Blitzkurs into data analysis using R.

Where are the slides?

There are none. I feel that slides are not optimal for learning. In class, slides can be detrimental if they are too wordy because that distracts from that the dialogue with the instructor, and I hold this very dialogue as essential. Outside of class, slides are neither helpful. Instead, a good book is much more beneficial, because in a book, there's enough room to patiently explain in sufficient details, an endeavor which is impossible for a slide deck.

To underline my messages to you, dear learners, I will use some sketches on a virtual white-board, some interactive apps, live coding, and some (pre-prepared) diagrams. That's a bit similar to what happens at Khan Academy. You might have noticed that many courses at Coursera follow a similar approach.

I readily confess that this approach is novel to many learners in these days, learners who are accustomed to hundreds of Powerpoint slides. Please be open and I think you will appreciate this didactic style.

Technical Details

Last update: 2022-10-30 12:14:43

```
sessioninfo::session_info()
```

```
- Session info -------------------------------------------------------------
 setting  value
 version  R version 4.2.1 (2022-06-23)
 os       macOS Big Sur ... 10.16
 system   x86_64, darwin17.0
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       Europe/Berlin
 date     2022-09-16
 pandoc   2.19.2 @ /usr/local/bin/ (via rmarkdown)

- Packages -----------------------------------------------------------------
 package     * version date (UTC) lib source
 cli           3.4.0   2022-09-08 [1] CRAN (R 4.2.0)
 colorout    * 1.2-2   2022-06-13 [1] local
 digest        0.6.29  2021-12-01 [1] CRAN (R 4.2.0)
 evaluate      0.16    2022-08-09 [1] CRAN (R 4.2.0)
 fastmap       1.1.0   2021-01-25 [1] CRAN (R 4.2.0)
 htmltools     0.5.3   2022-07-18 [1] CRAN (R 4.2.0)
 jsonlite      1.8.0   2022-02-22 [1] CRAN (R 4.2.0)
 knitr         1.40    2022-08-24 [1] CRAN (R 4.2.0)
 magrittr      2.0.3   2022-03-30 [1] CRAN (R 4.2.0)
 rlang         1.0.5   2022-08-31 [1] CRAN (R 4.2.0)
 rmarkdown     2.16    2022-08-24 [1] CRAN (R 4.2.0)
 rstudioapi    0.14    2022-08-22 [1] CRAN (R 4.2.0)
 sessioninfo   1.2.2   2021-12-06 [1] CRAN (R 4.2.0)
 stringi       1.7.8   2022-07-11 [1] CRAN (R 4.2.0)
 stringr       1.4.1   2022-08-20 [1] CRAN (R 4.2.0)
 xfun          0.33    2022-09-12 [1] CRAN (R 4.2.0)

 [1] /Users/sebastiansaueruser/Rlibs
 [2] /Library/Frameworks/R.framework/Versions/4.2/Resources/library

---------------------------------------------------------------------------
```

1 Goals in statistics

1.1 Overview

Many stories to be told. Here's one, on the goals pursued in statistics (and related fields), see Figure Figure 1.1.

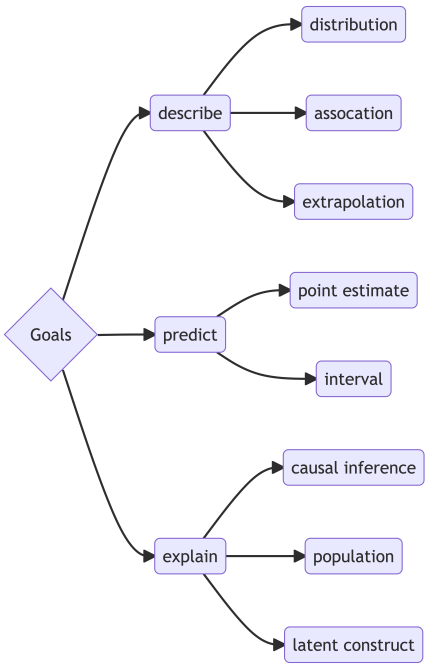


Figure 1.1: A taxonomy of statistical goals

> **Note**
>
> Note that "goals" do not exist in the world. We make them up in our heads. Hence, they have no ontological existence, they are epistemological beasts. This entails that we are free to devise goals as we wish, provided we can convince ourselves and other souls of the utility of our creativity.

1.2 Further reading

Hernán, Hsu, and Healy (2019) distinguish:

Hernán et al. (2019) distinguish:

- *Description*: "How can women aged 60–80 years with stroke history be partitioned in classes defined by their characteristics?"
- *Prediction*: "What is the probability of having a stroke next year for women with certain characteristics?"
- *Causal inference*: "Will starting a statin reduce, on average, the risk of stroke in women with certain characteristics?"

Gelman, Hill, and Vehtari (2021), chap. 1.1 proposes three "challenges" of statistical inference.

1.3 If nothing else helps

Stay calm and behold the infinity.

2 Basics

2.1 A framework for problem solving

2.1.1 PPDAC

The PPDAC Model is a methodological framework (aka a model) for applying the scientific method to any analytical or research question, or at least it is applicable to quite a few (MacKay and Oldford 2000). It is not meant to be a rigid sequence, but rather a cycle that may turn a number of rounds like a spiral. Statistician Chris Wild puts the PPDAC cycle in the following figure, see Figure Figure 2.1. In this short essay, he summaries his ideas on how to use the PPDAC as a tool for data analysis in problem solving.

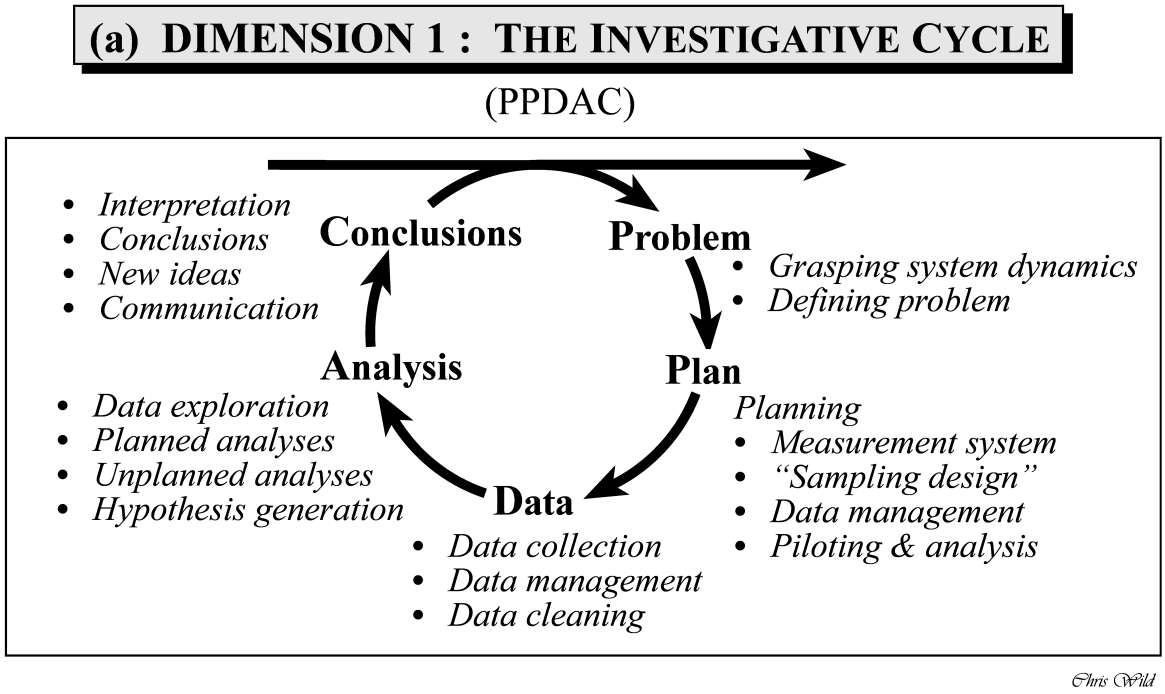


Figure 2.1: PPDAC cycle. Image source: Chris Wild

Wickham and Grolemund (see Figure Figure 3.1 in Section 3.3) provide a suggestion of the parts of the statistical analyses, that is the "Analysis" step in the PPDAC.

2.1.2 Fundamental issues in data analysis

Wild and Pfannkuch (1999) further note that variation is one of the essential characteristics of data. They discern to types of variation however, see Figure Figure 2.2.

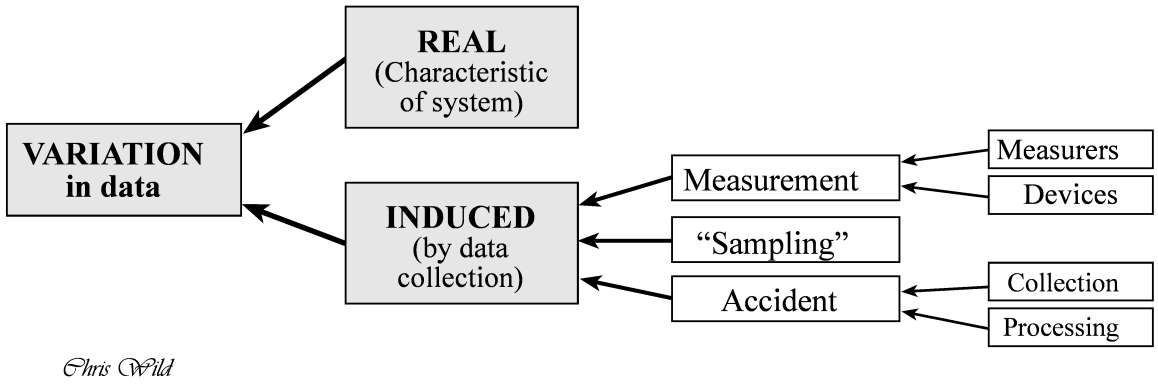


Figure 2.2: Two types of variartion. Image source: Chris Wild

Wild and Pfannkuch (1999) give a more systematic overview on how a quantitative research question - applied or basic - can be tackled and conceived. For example, in their paper the authors enumarate some dispositions that researcher should embrace in order to fruitfully engage in empirical research:

- Scepticism
- Imagination
- Curiosity
- Openness
- A propensity to seek deeper menaing
- Being logical
- Engagement
- Perseverance

2.2 R Basics

Check out chapter 1 in ModernDive for an accessible introduction to getting started with R and RStudio.

Please also note that R and RStudio should be installed before starting (this course).

2.3 Initial quiz

To get an idea whether you have digested some R basics, consider the following quiz.

Exercise 2.1 (Define a variable)**.** Define in R the variable `age` and assign the value `42`.[1]

Exercise 2.2 (Define a variable as a string)**.** Define in R the variable `name` and assign the value `me`.[2]

Exercise 2.3 (Define a variable by another variable)**.** Define in R the variable `name` and assign the *variable* `age`.[3]

Exercise 2.4 (Call a function)**.** Ask R what today's `date()` is, that is, call a function.[4]

Exercise 2.5 (Define a vector)**.** Define in R a vector `x` with the values 1,2,3 .[5]

Exercise 2.6 (Vector wise computation)**.** Square each value in the vector `x`.[6]

Exercise 2.7 (Vector wise computation 2)**.** Square each value in the vector `x` and sum up the values.[7]

[1] `age <- 42`, spaces are optional but useful
[2] `age <- "me"`
[3] `age <- age`
[4] `date()`
[5] `x <- c(1, 2, 3)`
[6] `x^2`
[7] `sum(x^2)`

Exercise 2.8 (Vector wise computation 3)**.** Square each value in the vector `x`, sum up the values, and divide by 3.[8]

Exercise 2.9 (Compute the variance)**.** Compute the variance of `x` using basic arithmetic.[9][10]

Exercise 2.10 (Work with NA)**.** Define the vector `y` with the values 1,2,NA. Compute the mean. Explain the results.[11]

2.4 Data import

Check out chapter 4 in ModernDive on how to import data into RStudio and for some basic concepts about "tidy data".

Spoiler: There's a button in RStudio in the "Environment" Pane saying "Import Dataset". Just click it, and things should work out.

2.5 Blitz start with data

To blitz start with data, type the following in R:

[8] `mean(x^2)`
[9] `sum(x^2)`
[10]

```
x <- c(1, 2, 3)

sum((x - mean(x))^2) / (length(x)-1)
```

```
[1] 1
```

```
# compare:
var(x)
```

```
[1] 1
```

[11] `y <- c(1, 2, NA); mean(y)` NA (not available, ie., missing) is contagious in R: If there's a missing element, R will assume that something has gone wrong and will raise a red flag, i.e, give you a NA back.

```r
data(mtcars)
```

And the data set `mtcars` will be available.

To get help for the data set, type `help(mtcars)`

A bit more advanced, but it's a nice data set, try the Palmer Penguins data set:

```r
d <- read.csv("https://vincentarelbundock.github.io/Rdatasets/csv/palmerpenguins/penguins.

head(d)  # see the first few rows, the "head" of the table
```

```
  X species    island bill_length_mm bill_depth_mm flipper_length_mm
1 1  Adelie Torgersen           39.1          18.7               181
2 2  Adelie Torgersen           39.5          17.4               186
3 3  Adelie Torgersen           40.3          18.0               195
4 4  Adelie Torgersen             NA            NA                NA
5 5  Adelie Torgersen           36.7          19.3               193
6 6  Adelie Torgersen           39.3          20.6               190
  body_mass_g    sex year
1        3750   male 2007
2        3800 female 2007
3        3250 female 2007
4          NA   <NA> 2007
5        3450 female 2007
6        3650   male 2007
```

Here's some documentation (code book) for this data set.

2.6 More data set

Check out this curated list of data sets useful for learning and practicing your data skills.

2.7 Literature

Wild and Pfannkuch (1999) discuss the thougnht processes involved in statistical problem solving seen from a broad perspective. Ismay and Kim (2020) is a helpful start into the first steps in R.

3 Exploratory Data Analysis

3.1 R packages needed for this chapter

```
library(easystats)
library(tidyverse)
library(rstanarm)  # optional!
```

3.2 What's EDA?

Exploratory Data Analysis (EDA) is a procedure to scrutinize a dataset at hand in order learn about it. EDA comprises descriptive statistics, data visualization and data transformation techniques (such as dimension reduction).

It's not so mathematical deep as modelling, but in practice it's really important.

There's this famous saying:

> In Data Science, 80% of time spent prepare data, 20% of time spent complain about the need to prepare data.

EDA can roughly be said to comprise the following parts:

- Importing (and exporting) data
- Data cleansing (such as deal with missing data etc)
- Data transformation or "wrangling" (such as long to wide format)
- Computing desriptive statistics (such as the notorious mean)
- Analyzing distributions (is it normal?)
- Finding patterns in data (aka data mining)
- More complex data transformation techniques (such as factor analysis)

3.3 Data journey

Wickham and Grolemund (2016) present a visual sketch of what could be called the "data journey", i.e., the steps we are taking in order to learn from data, seen from an hands-on angle, see Figure 3.1.

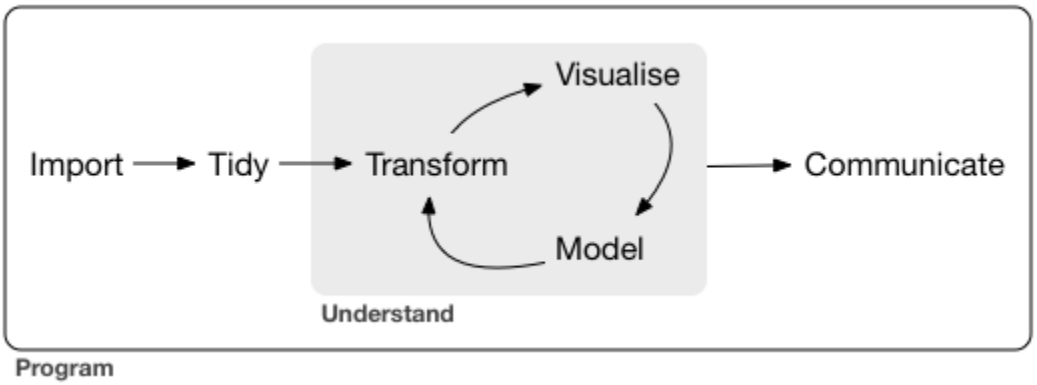


Figure 3.1: The data journey

3.4 Blitz data

See Section 2.5 for some data sets suitable to get going.

3.5 Data cleansing

The R package {janitor} provides some nice stuff for data cleansing. Check out this case study.

3.6 Convenience functions

There a quite a few functions (residing in some packages) that help you doing EDA from a helicoptor point of view. In other words, you do not have to pay attention to nitty-gritty details, the function will do that for you. This is approach is, well, convenient, but of course comes at a price. You will not have a great amount of choice and influence on the way the data is analyzed and presented.

3.6.1 Data Explorer

There are many systems and approaches to explore data. One particular interesting system is the R-package `DataExplorer`.

Figure 3.2: R-package DataExplorer

Check it out on its Github page.

3.6.2 vtree

A bit similar to {DataExplorer}, the R package {vtree} helps to explore visually datasets.

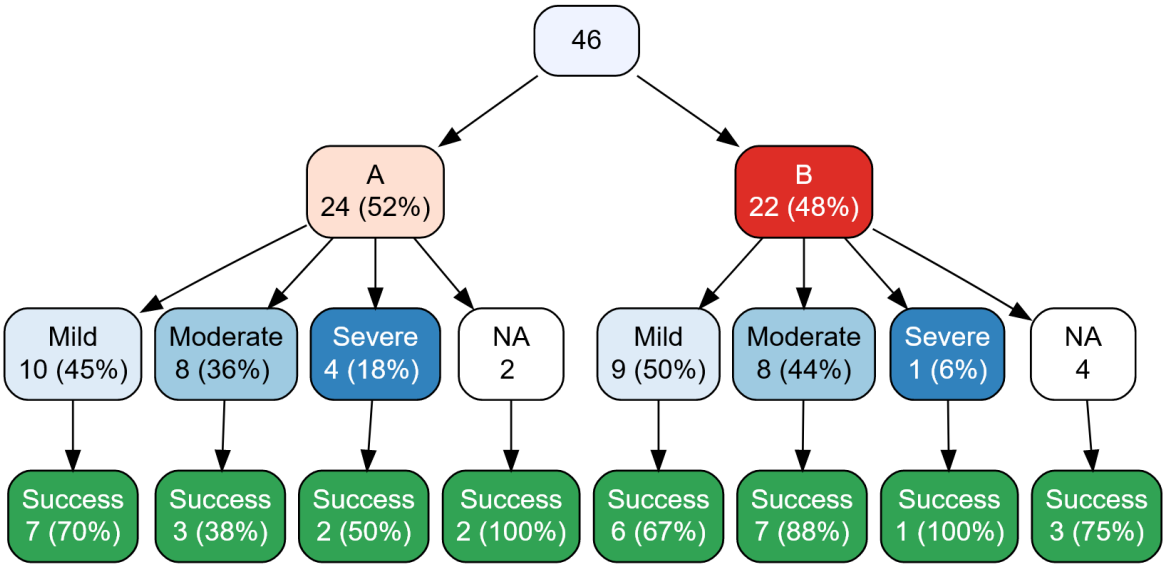


Figure 3.3: vtree is used to generate variable trees, like the one above.

3.6.3 The easystats way

There are some packages, such as `{easystats}`, which provide comfortable access to basic statistics:

```
library(easystats)  # once per session
describe_distribution(mtcars)
```

```
Variable |   Mean |     SD |    IQR |           Range | Skewness | Kurtosis |  n | n_Missing
-------------------------------------------------------------------------------------------
mpg      |  20.09 |   6.03 |   7.53 |  [10.40, 33.90] |     0.67 |    -0.02 | 32 |         0
cyl      |   6.19 |   1.79 |   4.00 |    [4.00, 8.00] |    -0.19 |    -1.76 | 32 |         0
disp     | 230.72 | 123.94 | 221.53 | [71.10, 472.00] |     0.42 |    -1.07 | 32 |         0
hp       | 146.69 |  68.56 |  84.50 | [52.00, 335.00] |     0.80 |     0.28 | 32 |         0
drat     |   3.60 |   0.53 |   0.84 |    [2.76, 4.93] |     0.29 |    -0.45 | 32 |         0
wt       |   3.22 |   0.98 |   1.19 |    [1.51, 5.42] |     0.47 |     0.42 | 32 |         0
qsec     |  17.85 |   1.79 |   2.02 |  [14.50, 22.90] |     0.41 |     0.86 | 32 |         0
vs       |   0.44 |   0.50 |   1.00 |    [0.00, 1.00] |     0.26 |    -2.06 | 32 |         0
am       |   0.41 |   0.50 |   1.00 |    [0.00, 1.00] |     0.40 |    -1.97 | 32 |         0
gear     |   3.69 |   0.74 |   1.00 |    [3.00, 5.00] |     0.58 |    -0.90 | 32 |         0
carb     |   2.81 |   1.62 |   2.00 |    [1.00, 8.00] |     1.16 |     2.02 | 32 |         0
```

`describe_distribution` provides us with an overview on typical descriptive summaries.

For nominal variables, consider `data_tabulate`:

```
data_tabulate(mtcars, select = c("am", "vs"))
```

```
am (am) <numeric>
# total N=32 valid N=32

Value |  N | Raw % | Valid % | Cumulative %
------+----+-------+---------+-------------
0     | 19 | 59.38 |   59.38 |        59.38
1     | 13 | 40.62 |   40.62 |       100.00
<NA>  |  0 |  0.00 |    <NA> |         <NA>

vs (vs) <numeric>
# total N=32 valid N=32

Value |  N | Raw % | Valid % | Cumulative %
```

```
------+----+-------+---------+-------------
0     | 18 | 56.25 |   56.25 |        56.25
1     | 14 | 43.75 |   43.75 |       100.00
<NA>  |  0 |  0.00 |    <NA> |         <NA>
```

We can also get *grouped* tabulations, which amounts to something similar to a contingency table:

```
mtcars %>%
  group_by(am) %>%
  data_tabulate(select = "vs", collapse = TRUE)
```

```
# Frequency Table

Variable |  Group | Value |  N | Raw % | Valid % | Cumulative %
---------+--------+-------+----+-------+---------+-------------
vs       | am (0) |     0 | 12 | 63.16 |   63.16 |        63.16
         |        |     1 |  7 | 36.84 |   36.84 |       100.00
         |        |  <NA> |  0 |  0.00 |    <NA> |         <NA>
---------+--------+-------+----+-------+---------+-------------
vs       | am (1) |     0 |  6 | 46.15 |   46.15 |        46.15
         |        |     1 |  7 | 53.85 |   53.85 |       100.00
         |        |  <NA> |  0 |  0.00 |    <NA> |         <NA>
-----------------------------------------------------------------
```

Checkout the function reference of your favorite package in order to learn what's on the shelf. For example, here's the function reference site of `datawizard`, one of the packages in the `easystats` ecosystem.

3.7 Tidyverse

3.7.1 Intro to the tidyverse

The Tidyverse is probably the R thing with the most publicity. And it's great. It's a philosophy baken into an array of R packages. Perhaps central is the idea that a lot of little lego pieces, if fitting nicely together, provides a simple yet flexibel and thus powerful machinery.

There's a lot of introctory material to the tidyverse around for instance, so I'm not repeating that here.

3.7.2 More advanced tidyverse

3.7.2.1 Repeat a function over many columns

At times, we would like to compute the same functions for many variables, ie columns for tidyverse applications.

Let's load the penguins data for illustration.

```
d <- read_csv("https://vincentarelbundock.github.io/Rdatasets/csv/palmerpenguins/penguins.

head(d)
```

```
# A tibble: 6 x 9
   ...1 species island    bill_length_mm bill_dept~1 flipp~2 body_~3 sex    year
  <dbl> <chr>   <chr>              <dbl>       <dbl>   <dbl>   <dbl> <chr> <dbl>
1     1 Adelie  Torgersen           39.1        18.7     181    3750 male   2007
2     2 Adelie  Torgersen           39.5        17.4     186    3800 fema~  2007
3     3 Adelie  Torgersen           40.3        18       195    3250 fema~  2007
4     4 Adelie  Torgersen           NA          NA        NA      NA <NA>   2007
5     5 Adelie  Torgersen           36.7        19.3     193    3450 fema~  2007
6     6 Adelie  Torgersen           39.3        20.6     190    3650 male   2007
# ... with abbreviated variable names 1: bill_depth_mm, 2: flipper_length_mm,
#   3: body_mass_g
```

Say, we would like to compute the mean value for each numeric variable in the data set:

```
d %>%
  summarise(across(bill_length_mm:body_mass_g, mean, na.rm = TRUE))
```

```
# A tibble: 1 x 4
  bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
           <dbl>         <dbl>             <dbl>       <dbl>
1           43.9          17.2              201.       4202.
```

Synonymously, we could write:

```
d %>%
  summarise(across(where(is.numeric), ~ mean(.x, na.rm = TRUE)))
```

```
# A tibble: 1 x 6
   ...1 bill_length_mm bill_depth_mm flipper_length_mm body_mass_g  year
  <dbl>          <dbl>         <dbl>             <dbl>       <dbl> <dbl>
1  172.           43.9          17.2              201.       4202. 2008.
```

Say, we would like to compute the z-value of each numeric variable.

Addmittedly, `easystats` makes it quite simple:

```
d %>%
  standardise(select = is.numeric) %>%
  head()
```

```
# A tibble: 6 x 9
   ...1 species island    bill_length_mm bill_dept~1 flipp~2 body_~3 sex    year
  <dbl> <chr>   <chr>              <dbl>       <dbl>   <dbl>   <dbl> <chr> <dbl>
1 -1.72 Adelie  Torgersen         -0.883       0.784  -1.42   -0.563 male  -1.26
2 -1.71 Adelie  Torgersen         -0.810       0.126  -1.06   -0.501 fema~ -1.26
3 -1.70 Adelie  Torgersen         -0.663       0.430  -0.421  -1.19  fema~ -1.26
4 -1.69 Adelie  Torgersen         NA          NA      NA      NA     <NA>  -1.26
5 -1.68 Adelie  Torgersen         -1.32        1.09   -0.563  -0.937 fema~ -1.26
6 -1.67 Adelie  Torgersen         -0.847       1.75   -0.776  -0.688 male  -1.26
# ... with abbreviated variable names 1: bill_depth_mm, 2: flipper_length_mm,
#   3: body_mass_g
```

See the help page of `standardise` for mor details on how to select variables and on more options.

But for the purpose of illustration, let's do it with more simple means, i.e. tidyverse only.

```
d %>%
  transmute(across(bill_length_mm:body_mass_g,
                   .fns = ~ {(.x - mean(.x, na.rm = TRUE)) / sd(.x, na.rm = TRUE)},
                   .names = "{.col}_z"))
```

```
# A tibble: 344 x 4
   bill_length_mm_z bill_depth_mm_z flipper_length_mm_z body_mass_g_z
              <dbl>           <dbl>               <dbl>         <dbl>
 1           -0.883           0.784              -1.42         -0.563
 2           -0.810           0.126              -1.06         -0.501
 3           -0.663           0.430              -0.421        -1.19
```

```
 4           NA              NA                  NA          NA
 5           -1.32            1.09               -0.563      -0.937
 6           -0.847           1.75               -0.776      -0.688
 7           -0.920           0.329              -1.42       -0.719
 8           -0.865           1.24               -0.421       0.590
 9           -1.80            0.480              -0.563      -0.906
10           -0.352           1.54               -0.776       0.0602
# ... with 334 more rows
```

It's maybe more succint to put the z-value computation in its function, and then just apply this function:

```
z_stand <- function(x){
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}
```

```
d2 <-
d %>%
  mutate(across(bill_length_mm:body_mass_g,
                .fns = z_stand))

d2 %>%
  glimpse()
```

```
Rows: 344
Columns: 9
$ ...1              <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
$ species           <chr> "Adelie", "Adelie", "Adelie", "Adelie", "Adelie", "A~
$ island            <chr> "Torgersen", "Torgersen", "Torgersen", "Torgersen", ~
$ bill_length_mm    <dbl> -0.8832047, -0.8099390, -0.6634077, NA, -1.3227986, ~
$ bill_depth_mm     <dbl> 0.78430007, 0.12600328, 0.42983257, NA, 1.08812936, ~
$ flipper_length_mm <dbl> -1.4162715, -1.0606961, -0.4206603, NA, -0.5628905, ~
$ body_mass_g       <dbl> -0.563316704, -0.500969030, -1.186793445, NA, -0.937~
$ sex               <chr> "male", "female", "female", NA, "female", "male", "f~
$ year              <dbl> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

3.7.3 Rowwise operations

For technical reasons, it's a bit cumbersome in (base) R to compute rowwise operations. The thing is, R's dataframes are organized as vectors of *columns* so it's much easier to do stuff columnwise.

However, since recently, computing rowwise operations with the tidyverse has become simpler. Consider the following example. Say we would like to know the highest z-value for each variable we just computed, that is the highest values *per individual*, ie., by row in the data frame.

```
d2 %>%
  drop_na() %>%
  rowwise() %>%
  mutate(max_z = max(c(bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g))) %>
  head()
```

```
# A tibble: 6 x 10
# Rowwise:
   ...1 species island    bill_lengt~1 bill_~2 flipp~3 body_~4 sex    year max_z
  <dbl> <chr>   <chr>            <dbl>   <dbl>   <dbl>   <dbl> <chr> <dbl> <dbl>
1     1 Adelie  Torgersen       -0.883   0.784  -1.42   -0.563 male   2007 0.784
2     2 Adelie  Torgersen       -0.810   0.126  -1.06   -0.501 fema~  2007 0.126
3     3 Adelie  Torgersen       -0.663   0.430  -0.421  -1.19  fema~  2007 0.430
4     5 Adelie  Torgersen       -1.32    1.09   -0.563  -0.937 fema~  2007 1.09
5     6 Adelie  Torgersen       -0.847   1.75   -0.776  -0.688 male   2007 1.75
6     7 Adelie  Torgersen       -0.920   0.329  -1.42   -0.719 fema~  2007 0.329
# ... with abbreviated variable names 1: bill_length_mm, 2: bill_depth_mm,
#   3: flipper_length_mm, 4: body_mass_g
```

3.8 Case Study

Figure 3.4: R package/dataset palmerpenguins

Explore the `palmerpenguins` dataset, it's a famous dataset made for learning data analysis.

There's a great interactive course on EDA based on the penguins. Have a look, it's great!

Go penguins! Allez!

3.9 Cheatsheets

There are a number of nice cheat sheets available on an array of topics related to EDA, made available by the folks at RStudio.

Consider this collection:

- {dplyr}: data wrangling
- {tidyr}: data preparation
- {ggplot}: data visualization
- {gtsummary}: publication ready tables

So much great stuff! A bit too much to digest in one go, but definitely worthwhile if you plan to dig deeper in data analysis.

3.10 Literature

Wickham and Grolemund (2016) is an highly recommendable resource in order to get a thorough understanding of data analysis using R. Note that this source is focusing on the "how to", not so much to theoretical foundations. Ismay and Kim (2020) is a gently introduction into many steps on the data journey, including EDA.

4 Inference

4.1 What is it?

Statistical inference, according to Gelman, Hill, and Vehtari (2021), chap. 1.1, faces the challenge of *generalizing* from the particular to the general.

In more details, this amounts to generalizing from …

1. a sample to a population
2. a treatment to a control group (i.e., causal inference)
3. observed measurement to the underlying ("latent") construct of interest

> **!** Important
>
> Statistical inference is concerned with making general claims from particular data using mathematical tools.

4.2 Population and sample

We want to have an estimate of some population value, for example the proportion of `A`.

However, all we have is a subset, a sample of the populuation. Hence, we need to *infer* from the sample to the popluation. We do so by generalizing from the sample to the population, see Figure Figure 4.1.

4.3 What's not inference?

Consider fig. Figure 4.2 which epitomizes the difference between *descriptive* and *inferential* statistics.

(a) Population (b) Sample

Figure 4.1: Population vs. sample (Image credit: Karsten Luebke)

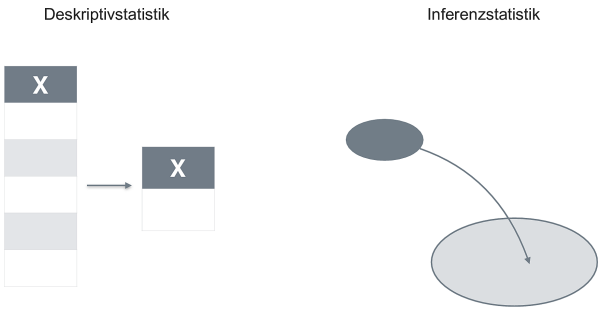


Figure 4.2: The difference between description and inference

4.4 When size helps

Larger samples allow for more precise estimations (ceteris paribus).

4.5 What flavors are available?

Typically, when one hears "inference" one thinks of p-values and null hypothesis testing. Those procedures are examples of the school of *Frequentist statistics*.

However, there's a second flavor of statistics to be mentioned here: *Bayesian statistics*.

4.5.1 Frequentist inference

Frequentism is *not* concerned about the probability of your research hypothesis.

Frequentism is all about controlling the *long-term error*. For illustration, suppose you are the CEO of a factory producing screws, and many of them. As the boss, you are not so much interested if a particular scree is in order (or faulty). Rather you are interested that the overall, long-term error rate of your production is low. One may add that your goal might not the minimize the long-term error, b ut to control it to a certain level - it may be to expensive to produce super high quality screws. Some decent, but cheap screws, might be more profitable.

4.5.2 Bayes inference

Bayes inference is concerned about the probability of your research hypothesis.

It simply redistributes your beliefs based on new data (evidence) you observe, see Figure **?@fig-belief-update**.

In more detail, the posterior belief is formalized as the posterior probability. The Likelihood is the probability of the data given some hypothesis. The normalizing constant serves to give us a number between zero and one.

$$\overbrace{\Pr(H|D)}^{\text{posterior probability}} = \overbrace{Pr(H)}^{\text{prior}} \frac{\overbrace{Pr(D|H)}^{\text{likelihood}}}{\underbrace{Pr(D)}_{\text{normalizing constant}}}$$

In practice, the posterior probability of your hypothesis is, the average of your prior and the Likelihood of your data.

Can you see that the posterior is some average of prior and likelihood?

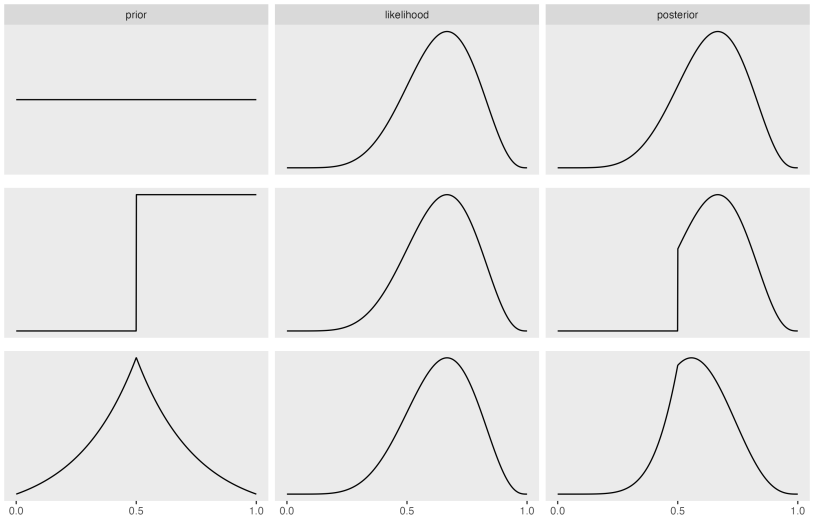


Figure 4.3: Prior-Likelihood-Posterior

Check out this great video on Bayes Theorem by 3b1b.

4.6 But which one should I consume?

PRO Frequentist:

- Your supervisor and reviewers will be more familiar with it
- The technical overhead is simpler compared to Bayes

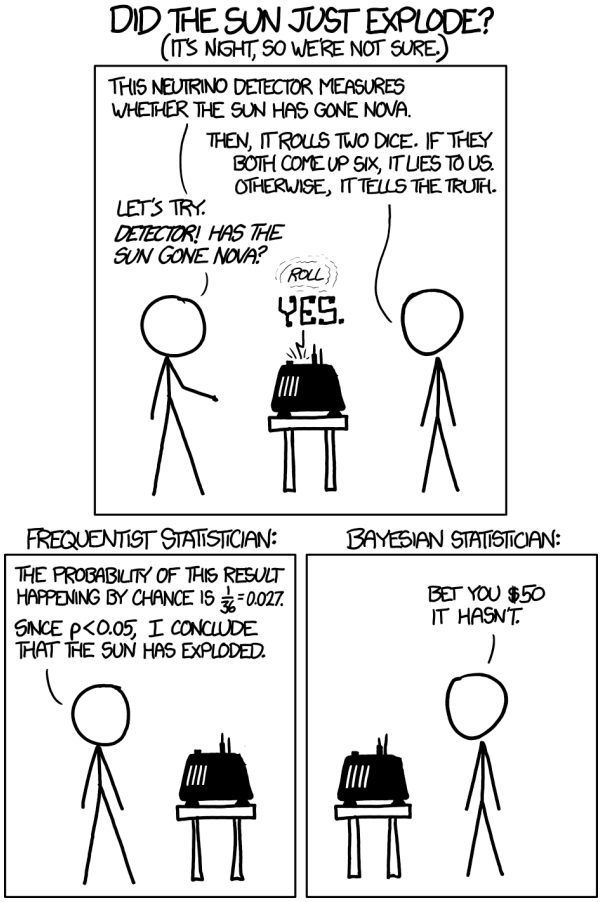
PRO Bayes:

- You'll probably want to have a posterior probability of your hypothesis
- You may appear as a cool kid and an early adoptor of emering statistical methods

> **Tip**
>
> You'll learn that the technical setup used for doing Bayes statistics is quite similar to doing frequentist statistics. Stay tuned.

4.7 Comment from xkcd

Quelle


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY.
DETECTOR! HAS THE SUN GONE NOVA?
ROLL
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS $\frac{1}{36} = 0.027$.
SINCE $p < 0.05$, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

4.8 p-value

The p-value has been used as the pivotal criterion to decide about whether or not a research hypothesis were to be "accepted" (a term forbidden in frequentist and Popperian langauge) or to be rejected. However, more recently, it is advised to use the p-value only as *one* indicator among multiple; see Wasserstein and Lazar (2016) and Wasserstein, Schirm, and Lazar (2019).

> **!** Important
>
> The p-value is defined as the probability of obtaining the observed data (or more extreme) under the assumption of no effect.

Figure Figure 4.4 visualizes the p-value.

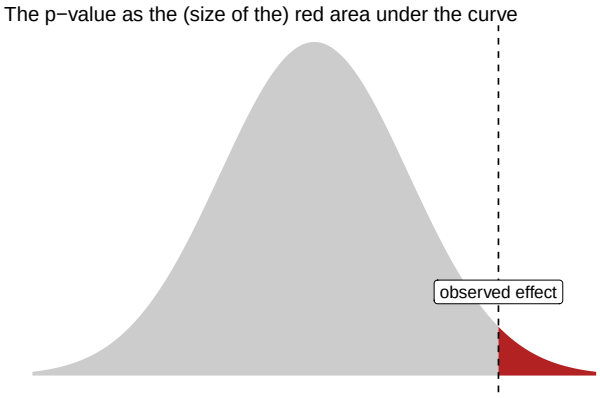


Figure 4.4: Visualization of the p-value

4.9 Some confusion remains about the p-value

Goodman (2008) provides an entertaining overview on typical misconceptions of the p-value full text.

Figure 4.5: Source: from ImgFlip Meme Generator

5 Modelling and regression

5.1 R packages needed for this chapter

```
library(easystats)
library(tidyverse)
library(rstanarm)  # optional!
```

5.2 What's modelling?

Read this great introduction by modelling by Russel Poldrack. Actually, the whole book is nice Poldrack (2022).

An epitome of modelling is this, let's call it the fundamental modelling equation, a bit grandiose but at the point, see Equation 5.1.

The data can be separated in the model's prediction and the rest (the "error"), i.e., what's unaccounted for by the model.

$$\text{data} = \text{model} + \text{error} \tag{5.1}$$

A more visual account of our basic modelling equation is depicted in **?@fig-model1**.

5.3 Regression as the umbrella tool for modelling

Source: Image Flip

Alternatively, venture into the forest of statistical tests as outlined e.g. here, at Uni Muenster.

You may want to ponder on this image of a decision tree of which test to choose, see Figure Figure 5.1.

5.3.1 Common statistical tests are linear models

As Jonas Kristoffer Lindeløv tells us, we can formulate most statistical tests as a linear model, ie., a regression.

5.3.2 How to find the regression line

In the simplest case, regression analyses can be interpreted geometrically as a line in a 2D coordinate system, see Figure Figure 5.3.

Source: Orzetoo, CC-SA, Wikimedia

Put simple, we are looking for the line which is in the "middle of the points". More precisely, we place the line such that the squared distances from the line to the points is minimal, see Figre Figure 5.3.

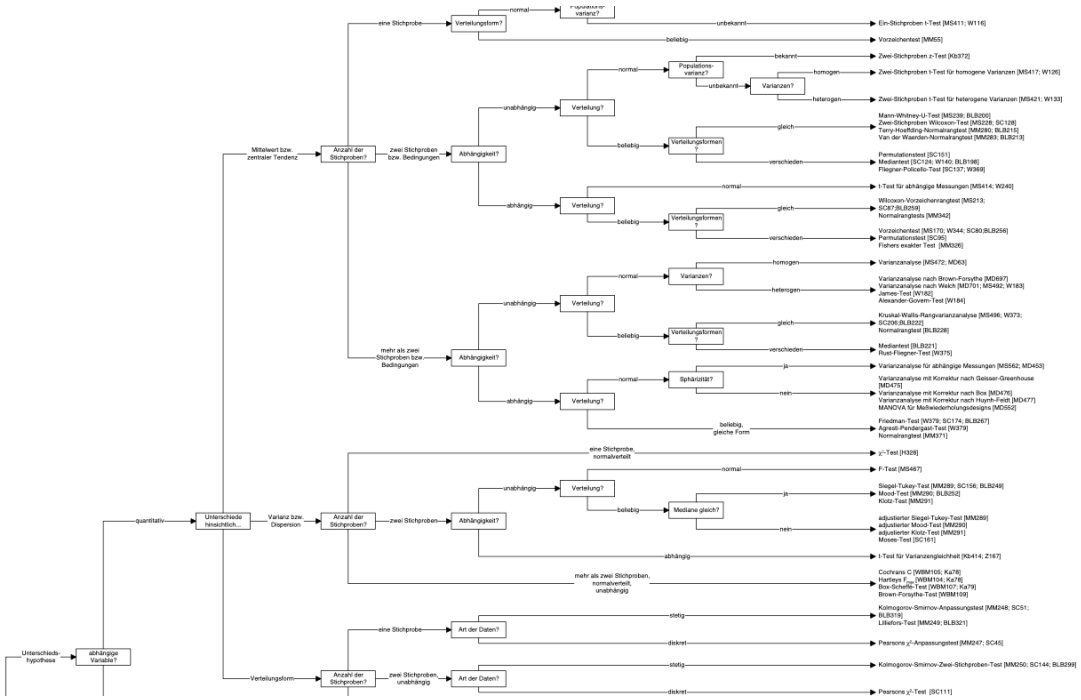


Figure 5.1: Choose your test carefully

Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y₁, y₂, paired=TRUE) wilcox.test(y₁, y₂, paired=TRUE)	lm(y₂ - y₁ ~ 1) lm(signed_rank(y₂ - y₁) ~ 1)	✓ for N >14	One intercept predicts the pairwise **y₂-y₁** differences. - (Same, but it predicts the *signed rank* of **y₂-y₁**.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y₁, y₂, var.equal=TRUE) t.test(y₁, y₂, var.equal=FALSE) wilcox.test(y₁, y₂)	lm(y ~ 1 + G₂)[A] gls(y ~ 1 + G₂, weights=…[B])[A] lm(signed_rank(y) ~ 1 + G₂)[A]	✓ ✓ for N >11	An intercept for **group 1** (plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x₁ + x₂ +…)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G₂ + G₃ +…+ Gₙ)[A] lm(rank(y) ~ 1 + G₂ + G₃ +…+ Gₙ)[A]	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G₂ + G₃ +…+ Gₙ + x)[A]	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G₂ + G₃ + … + Gₙ + S₂ + S₃+ … + Sₙ + G₂*S₂+G₂*S₃+…+Gₙ*Sₙ)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: $G_{2 \text{ to } n}$ is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for $S_{2 \text{ to } n}$ for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G₂ + G₃ + … + Gₙ + S₂ + S₃+ … + Sₙ + G₂*S₂+G₂*S₃+…+Gₙ*Sₙ, family=…)[A]	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments: `glm(model, family=poisson())` As linear-model, the Chi-square test is log(y) = log(N) + log(α) + log(β) + log(αβ) where α and β are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G₂ + G₃ +…+ Gₙ, family=…)[A]	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation y ~ 1 + x is R shorthand for y = 1·b + a·x which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at https://lindeloev.github.io/tests-as-linear.

[A] See the note to the two-way ANOVA for explanation of the notation.
[B] Same model, but with one variance per group: `gls(value ~ 1 + G₂, weights = varIdent(form = ~1|group), method="ML")`.

Jonas Kristoffer Lindeløv
https://lindeloev.net

Figure 5.2: Common statistical tests as linear models

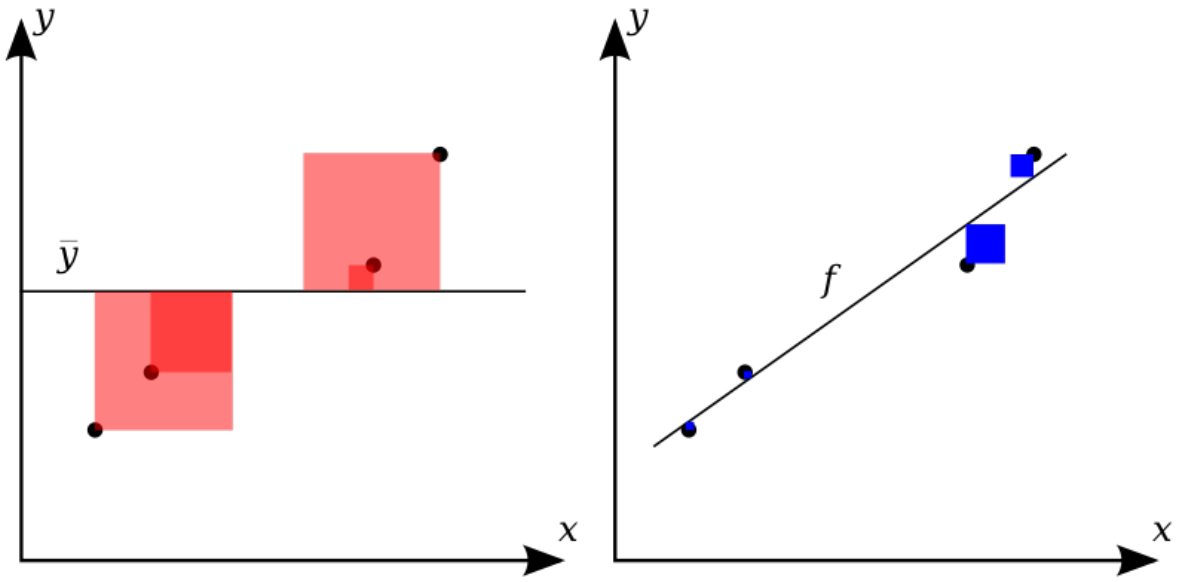


Figure 5.3: Least Square Regression

Consider Figure Figure 5.4, from this source by Roback and Legler (2021). It visualizes not only the notorious regression line, but also sheds light on regression assumptions, particularly on the error distribution.

5.3.3 The linear model

Here's the canonical form of the linear model.

Consider a model with k predictors:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k + \epsilon$$

5.3.4 Algebraic derivation

For the mathematical inclined, check out this derivation of the simple case regression model. Note that the article is written in German, but your browser can effortlessly translate into English. Here's a similar English article from StackExchange.

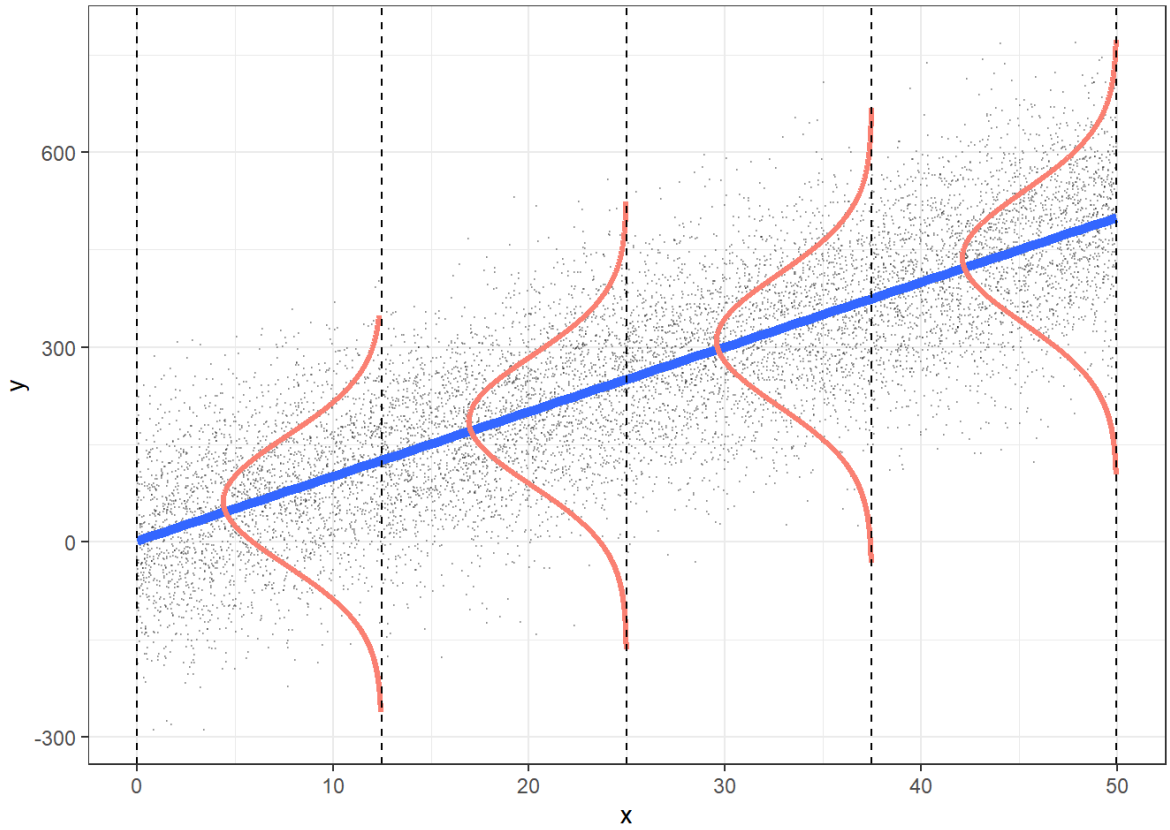
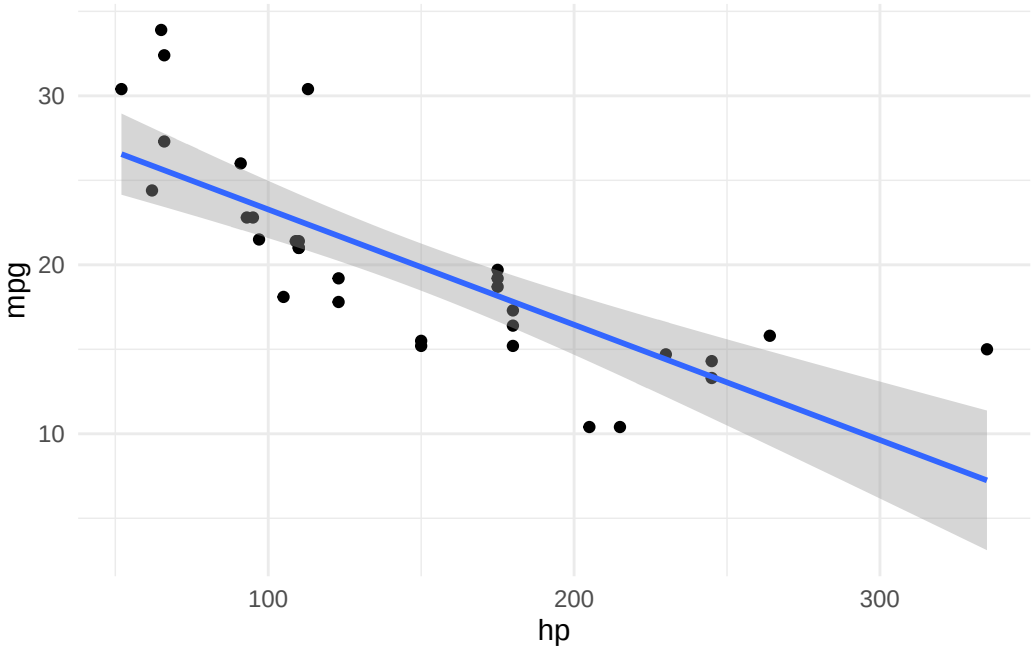


Figure 5.4: Regression and some of its assumptions

5.4 In all its glory



5.5 First model: one metric predictor

First, let's load some data:

```
data(mtcars)
glimpse(mtcars)
```

```
Rows: 32
Columns: 11
$ mpg  <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19.2, 17.8,~
$ cyl  <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 8,~
$ disp <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7, 140.8, 16~
$ hp   <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180, 180, 180~
$ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.92, 3.92,~
$ wt   <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190, 3.150, 3.~
$ qsec <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00, 22.90, 18~
$ vs   <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,~
$ am   <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,~
$ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, 3,~
```

```
$ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2, 1, 1, 2,~
```

5.5.1 Frequentist

Define and fit the model:

```
lm1_freq <- lm(mpg ~ hp, data = mtcars)
```

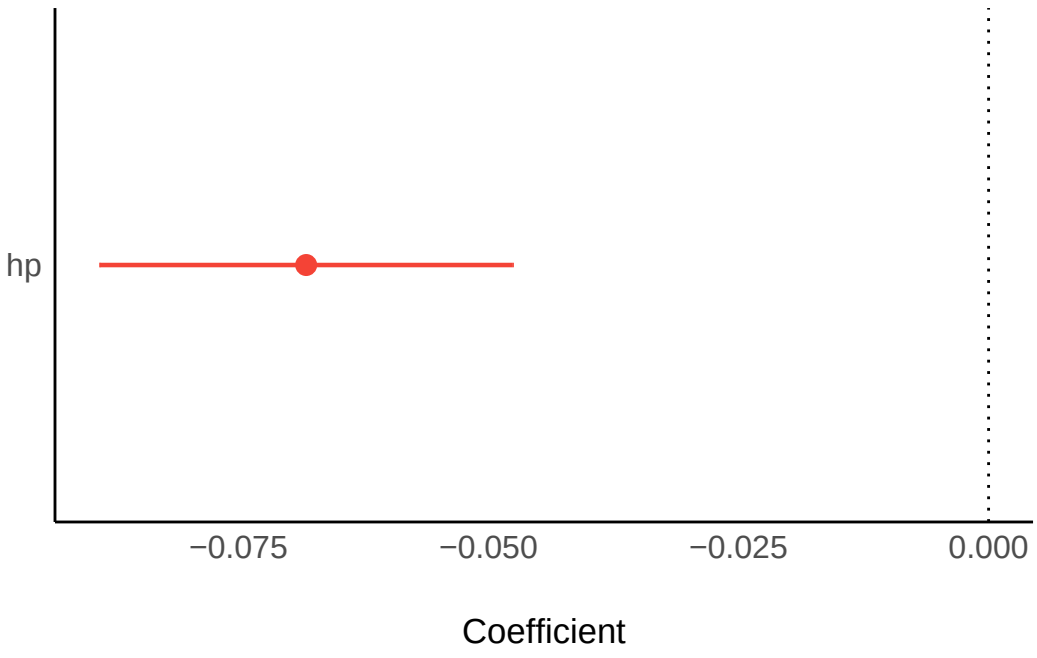
Get the parameter values:

```
parameters(lm1_freq)
```

```
Parameter   | Coefficient |   SE |         95% CI | t(30) |      p
-------------------------------------------------------------------
(Intercept) |       30.10 | 1.63 | [26.76, 33.44] | 18.42 | < .001
hp          |       -0.07 | 0.01 | [-0.09, -0.05] | -6.74 | < .001
```

Plot the model parameters:

```
plot(parameters(lm1_freq))
```



5.5.2 Bayesian

```r
lm1_bayes <- stan_glm(mpg ~ hp, data = mtcars)
```

```
SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000671 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 6.71 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.031964 seconds (Warm-up)
Chain 1:                0.030428 seconds (Sampling)
Chain 1:                0.062392 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1.4e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.14 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.032053 seconds (Warm-up)
Chain 2:                0.030316 seconds (Sampling)
Chain 2:                0.062369 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.7e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.17 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.030275 seconds (Warm-up)
Chain 3:                0.03358 seconds (Sampling)
Chain 3:                0.063855 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.6e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
```

```
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.031825 seconds (Warm-up)
Chain 4:                0.029619 seconds (Sampling)
Chain 4:                0.061444 seconds (Total)
Chain 4:
```

Actually, we want to suppress some overly verbose output of the sampling, so add the argument `refresh = 0`:

```
lm1_bayes <- stan_glm(mpg ~ hp, data = mtcars, refresh = 0)
```

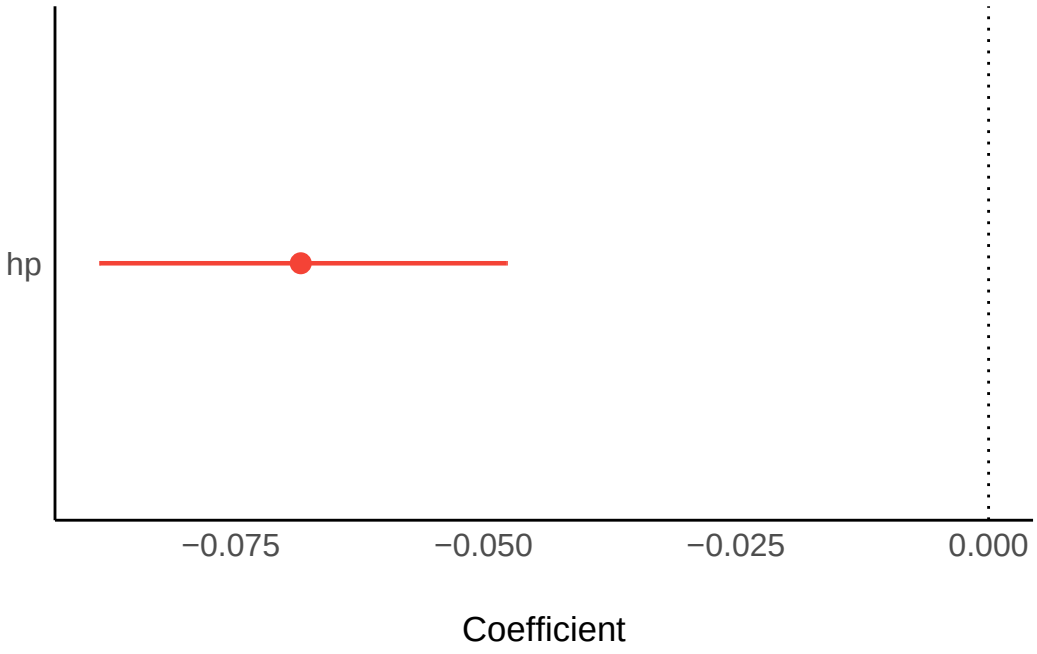
Get the parameter values:

```
parameters(lm1_bayes)
```

```
Parameter   | Median |         95% CI |   pd | % in ROPE |  Rhat |     ESS |
-------------------------------------------------------------------------------------
(Intercept) |  30.12 | [26.85, 33.34] | 100% |        0% | 1.001 | 3257.00 | Normal (20.09 +
hp          |  -0.07 | [-0.09, -0.05] | 100% |      100% | 1.000 | 3387.00 |   Normal (0.00 -
```

Plot the model parameters:

```
plot(parameters(lm1_bayes))
```



5.5.3 Model performance

```
r2(lm1_freq)
```

```
# R2 for Linear Regression
       R2: 0.602
  adj. R2: 0.589
```

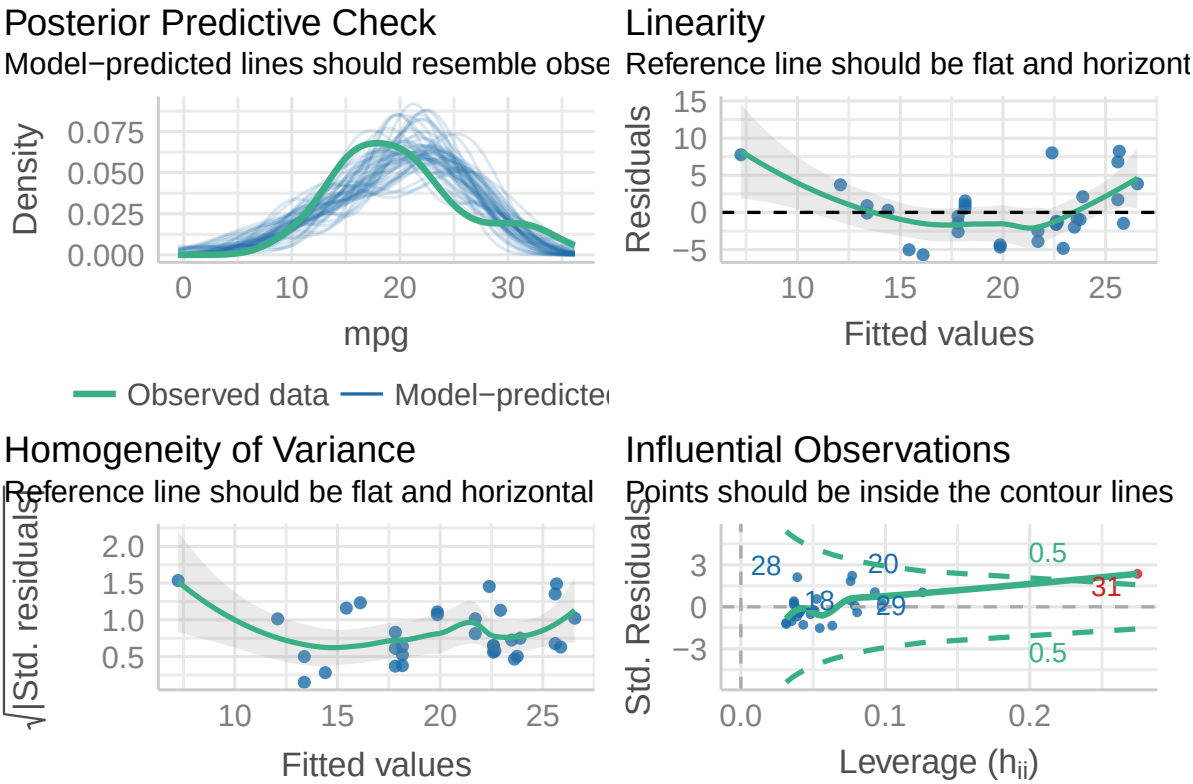
```
r2(lm1_bayes)
```

```
# Bayesian R2 with Compatibility Interval

  Conditional R2: 0.585 (95% CI [0.376, 0.744])
```

5.5.4 Model check

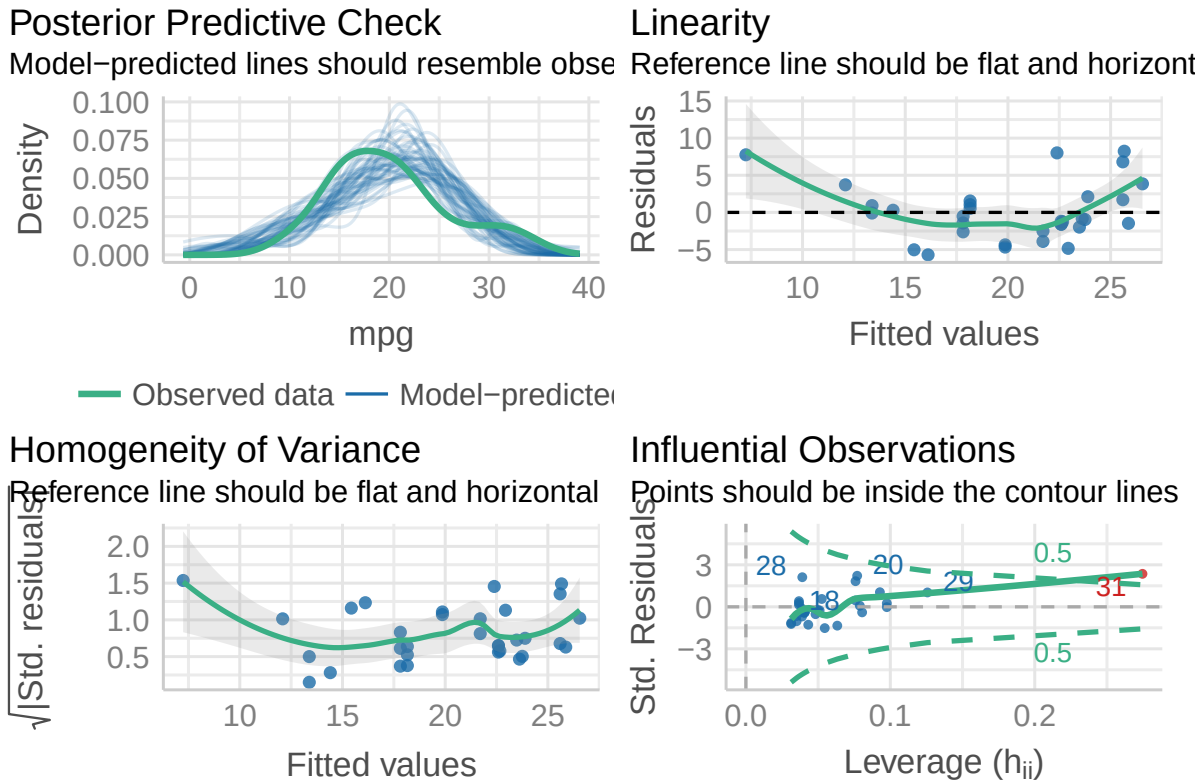
Here's a bunch of typical model checks in the Frequentist sense.

```
check_model(lm1_freq)
```



And here are some Bayesian flavored model checks.

```
check_model(lm1_bayes)
```



5.5.5 Get some predictions

```
lm1_pred <- estimate_relation(lm1_freq)
lm1_pred
```

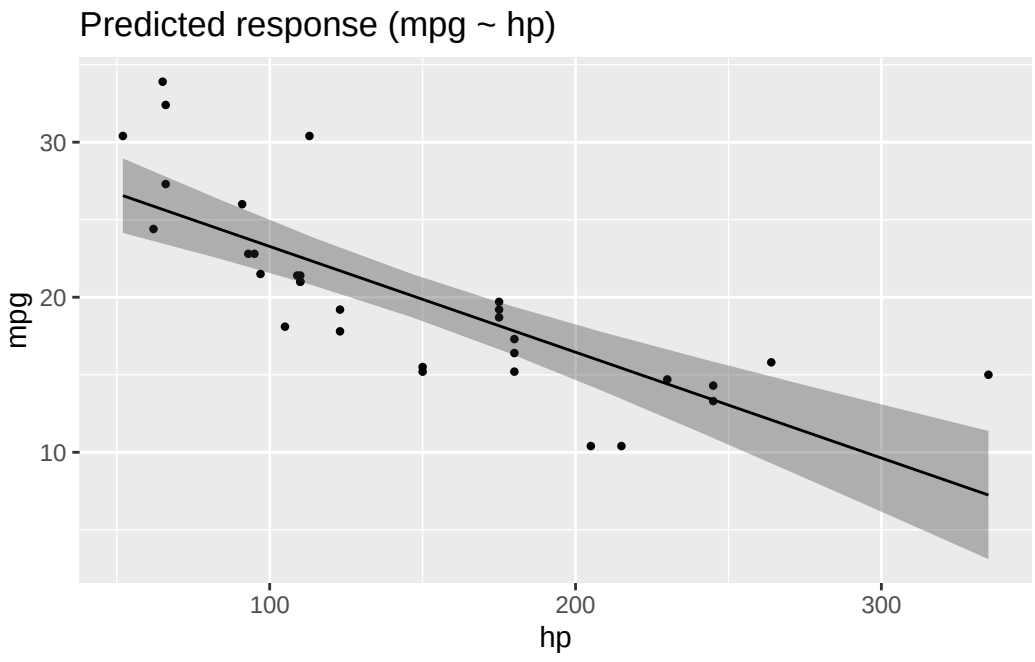
```
Model-based Expectation

hp     | Predicted |   SE |         95% CI
```

```
----------------------------------------
52.00  |     26.55 | 1.18 | [24.15, 28.95]
83.44  |     24.41 | 0.94 | [22.49, 26.32]
114.89 |     22.26 | 0.75 | [20.72, 23.80]
146.33 |     20.11 | 0.68 | [18.72, 21.51]
177.78 |     17.97 | 0.75 | [16.43, 19.50]
209.22 |     15.82 | 0.93 | [13.92, 17.73]
240.67 |     13.68 | 1.17 | [11.29, 16.07]
272.11 |     11.53 | 1.44 | [ 8.59, 14.48]
303.56 |      9.39 | 1.73 | [ 5.86, 12.92]
335.00 |      7.24 | 2.02 | [ 3.11, 11.38]

Variable predicted: mpg
Predictors modulated: hp
```

More details on the above function can be found on the respective page at the easystats site.

5.5.6 Plot the model

```
plot(lm1_pred)
```



5.6 More of this

More technical details for gauging model performance and model quality, can be found on the site of the R package "performance at the easystats site.

5.7 Bayes-members only

Bayes statistics provide a distribution as the result of the analysis, the posterior distribution, which provides us with quite some luxury.

As the posterior distribution manifests itself by a number of samples, we can easily filter and manipulate this sample distribution in order to ask some interesing questions.

See:

```
lm1_bayes %>%
  as_tibble() %>%
  head()
```

```
# A tibble: 6 x 3
  `(Intercept)`      hp sigma
          <dbl>   <dbl> <dbl>
1          27.5 -0.0545  3.86
2          32.4 -0.0812  3.60
3          29.7 -0.0567  3.86
4          29.5 -0.0608  4.47
5          30.7 -0.0664  3.78
6          29.4 -0.0682  3.89
```

5.7.1 Asking for probabilites

What's the probability that the effect of hp is negative?

```
lm1_bayes %>%
  as_tibble() %>%
  count(hp < 0)
```

```
# A tibble: 1 x 2
  `hp < 0`     n
  <lgl>    <int>
```

```
1 TRUE      4000
```

Feel free to ask similar questions!

5.7.2 Asking for quantiles

With a given probability of, say 90%, how large is the effect of hp?

```
lm1_bayes %>%
  as_tibble() %>%
  summarise(q_90 = quantile(hp, .9))
```

```
# A tibble: 1 x 1
     q_90
    <dbl>
1 -0.0550
```

What's the smallest 95% percent interval for the effect of hp?

```
hdi(lm1_bayes)
```

```
Highest Density Interval

Parameter   |        95% HDI
----------------------------
(Intercept) | [26.95, 33.43]
hp          | [-0.09, -0.05]
```

In case you prefer 89% intervals (I do!):

```
hdi(lm1_bayes, ci = .89)
```

```
Highest Density Interval

Parameter   |        89% HDI
----------------------------
(Intercept) | [27.41, 32.68]
hp          | [-0.09, -0.05]
```

5.8 Multiple metric predictors

Assume we have a theory that dictates that fuel economy is a (causal) function of horse power and engine displacement.

```
lm2_freq <- lm(mpg ~ hp + disp, data = mtcars)
parameters(lm2_freq)
```

```
Parameter   | Coefficient |       SE |         95% CI | t(29) |      p
-------------------------------------------------------------------------
(Intercept) |       30.74 |     1.33 | [28.01, 33.46] | 23.08 | < .001
hp          |       -0.02 |     0.01 | [-0.05,  0.00] | -1.86 | 0.074
disp        |       -0.03 | 7.40e-03 | [-0.05, -0.02] | -4.10 | < .001
```

Similarly for Bayes inference:

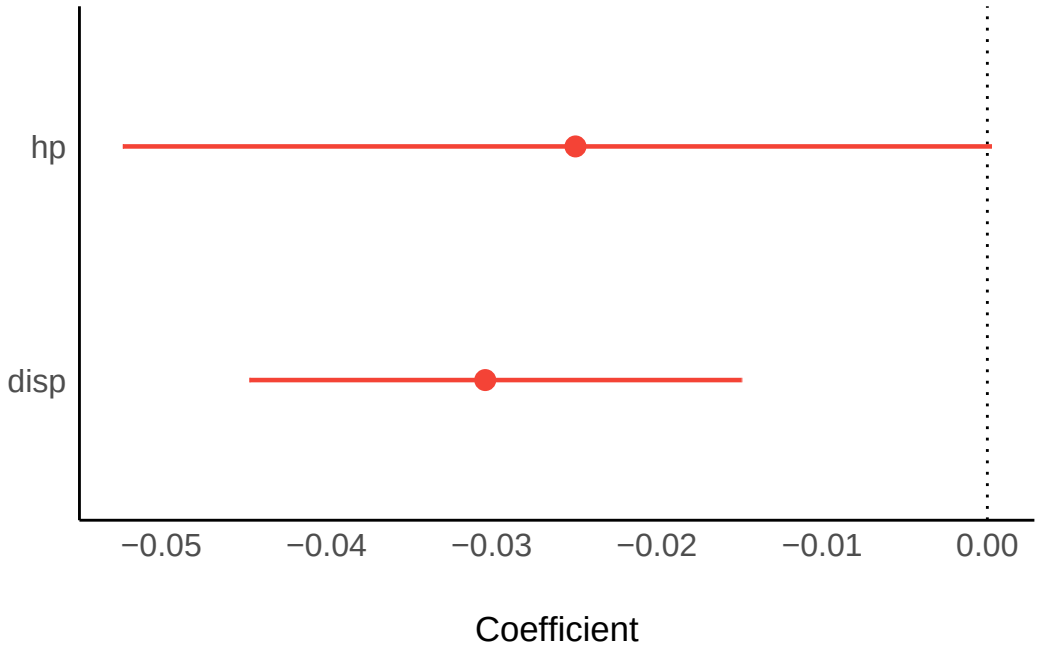
```
lm2_bayes <- stan_glm(mpg ~ hp + disp, data = mtcars)
```

Results

```
parameters(lm2_bayes)
```

```
Parameter   | Median |         95% CI |     pd | % in ROPE |  Rhat |     ESS |
-------------------------------------------------------------------------------------------
(Intercept) |  30.78 | [27.94, 33.57] |   100% |        0% | 0.999 | 5095.00 | Normal (20.09
hp          |  -0.02 | [-0.05,  0.00] | 97.28% |      100% | 1.001 | 2226.00 |   Normal (0.0
disp        |  -0.03 | [-0.04, -0.01] |   100% |      100% | 1.001 | 2150.00 |   Normal (0.0
```

```
plot(parameters(lm2_bayes))
```

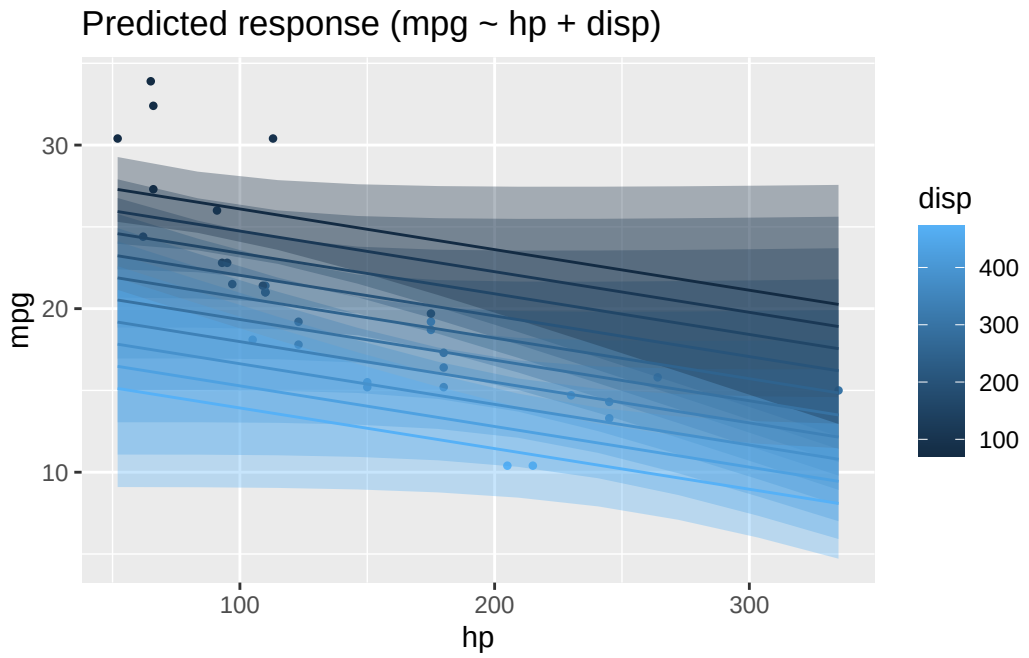



```
r2(lm2_bayes)
```

```
# Bayesian R2 with Compatibility Interval

  Conditional R2: 0.731 (95% CI [0.579, 0.847])
```

Depending on the value of `disp` the prediction of `mpg` from `hp` will vary:

```
lm2_pred <- estimate_relation(lm2_freq)
plot(lm2_pred)
```



5.9 One nominal predictor

```
lm3a <- lm(mpg ~ am, data = mtcars)
parameters(lm3a)
```

```
Parameter   | Coefficient |   SE |         95% CI | t(30) |      p
-------------------------------------------------------------------
(Intercept) |       17.15 | 1.12 | [14.85, 19.44] | 15.25 | < .001
am          |        7.24 | 1.76 | [ 3.64, 10.85] |  4.11 | < .001
```

```
lm3a_means <- estimate_means(lm3a, at = "am = c(0, 1)")
lm3a_means
```

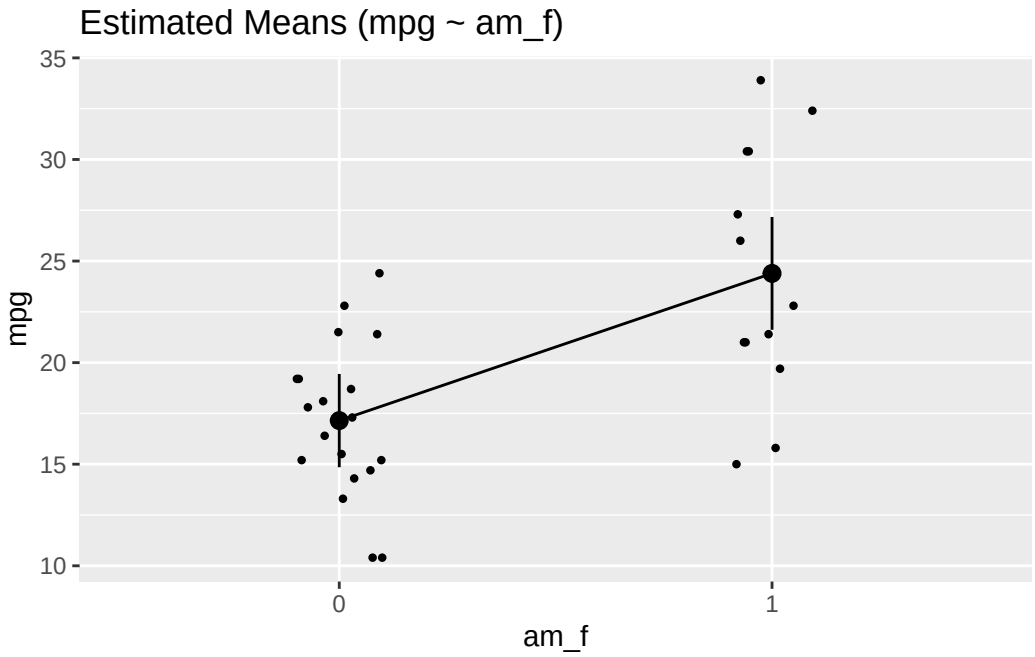
```
Estimated Marginal Means

am   |  Mean |   SE |         95% CI
-----------------------------------
0.00 | 17.15 | 1.12 | [14.85, 19.44]
1.00 | 24.39 | 1.36 | [21.62, 27.17]
```

```
Marginal means estimated at am
```

If we were not to specify the values of `am` which we would like to get predictions for, the default of the function would select 10 values, spreaded across the range of `am`. For numeric variables, this is usually fine. However, for nominal variables - and `am` is in fact a nominally scaled variable - we insist that we want predictions for the levels of the variable only, that is for `0` and `1`.

However, unfortunately, the plot *needs* a nominal variable if we are to compare groups. In our case, `am` is considered a numeric variables, since it consists of numbers only. The plot does not work, malheureusement:

```
plot(lm3a_means)
```



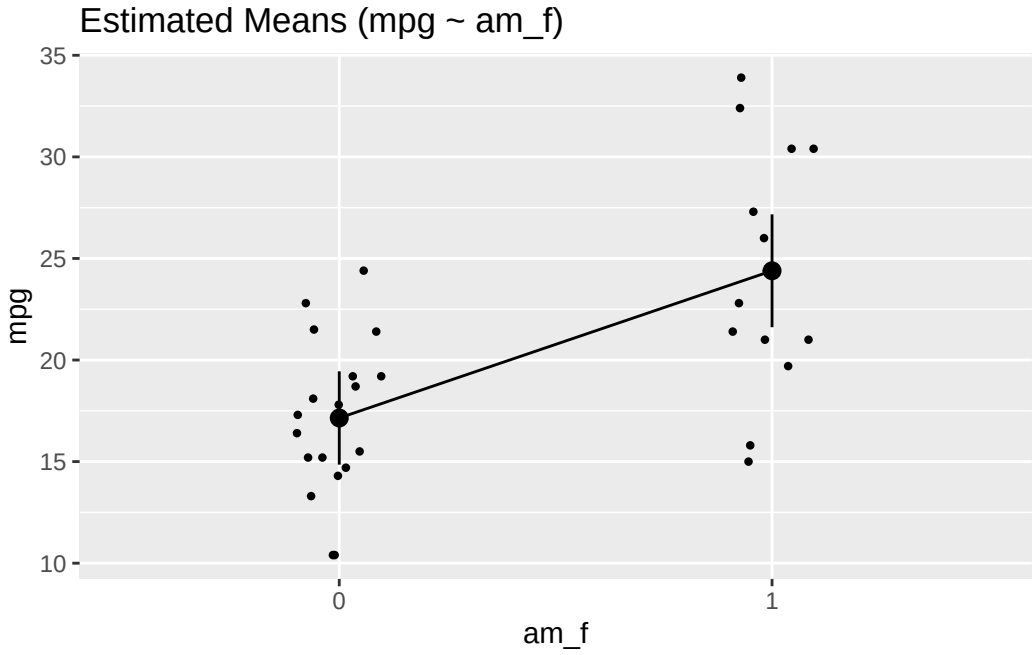
We need to transform `am` to a factor variable. That's something like a string. If we hand over a `factor()` to the plotting function, everything will run smoothly. Computationwise, no big differences:

```
mtcars2 <-
  mtcars %>%
  mutate(am_f = factor(am))

lm3a <- lm(mpg ~ am_f, data = mtcars2)
```

```
parameters(lm3a)
```

```
Parameter   | Coefficient |   SE |         95% CI | t(30) |      p
-------------------------------------------------------------------
(Intercept) |       17.15 | 1.12 | [14.85, 19.44] | 15.25 | < .001
am f [1]    |        7.24 | 1.76 | [ 3.64, 10.85] |  4.11 | < .001
```

```
lm3a_means <- estimate_means(lm3a)
plot(lm3a_means)
```



Note that we should have converted `am` to a factor variable before fitting the model. Otherwise, the plot won't work.

Here's a more hand-crafted version of the last plot, see Fig. Figure 5.5.

```
ggplot(mtcars2) +
  aes(x = am_f, y = mpg) +
  geom_violin() +
  geom_jitter(width = .1, alpha = .5) +
  geom_pointrange(data = lm3a_means,
                  color = "orange",
```

```
                  aes(ymin = CI_low, ymax = CI_high, y = Mean)) +
  geom_line(data = lm3a_means, aes(y = Mean, group = 1))
```

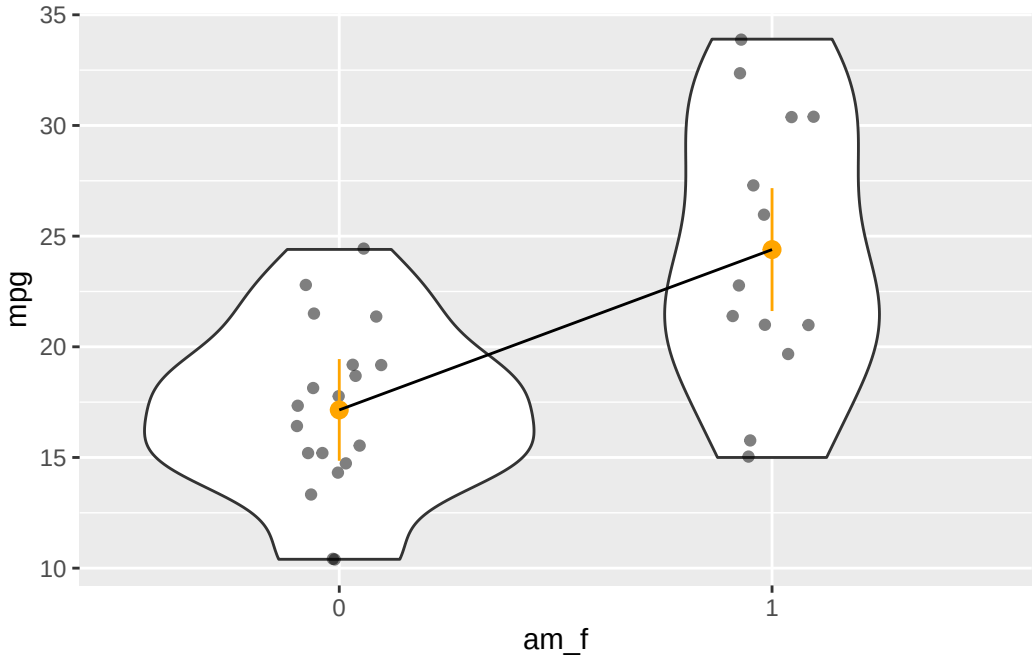


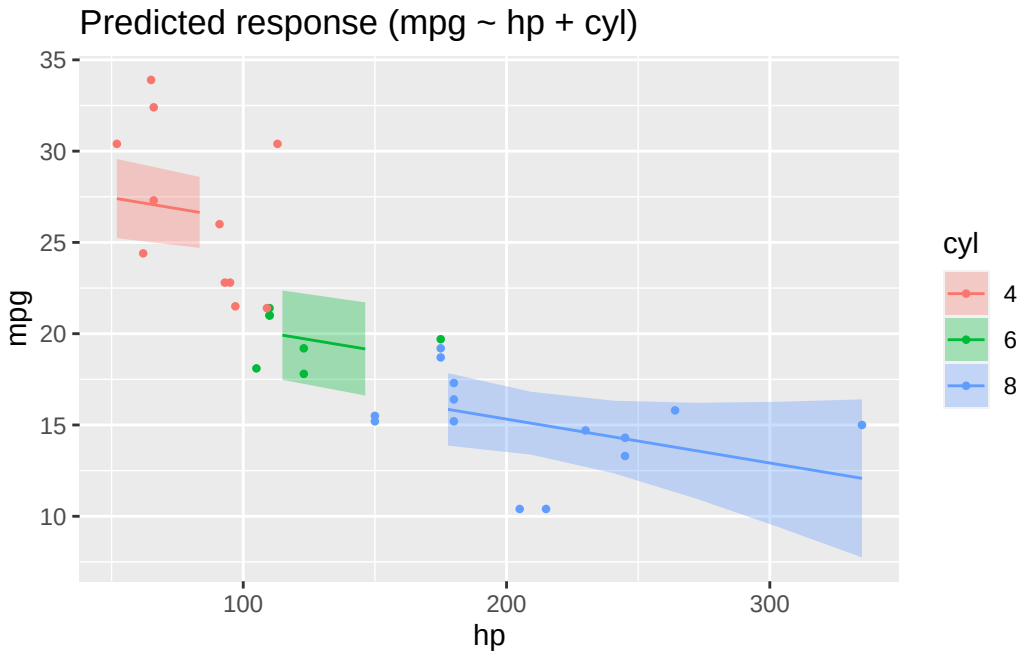
Figure 5.5: **?(caption)**

5.10 One metric and one nominal predictor

```
mtcars2 <-
  mtcars %>%
  mutate(cyl = factor(cyl))

lm4 <- lm(mpg ~ hp + cyl, data = mtcars2)
parameters(lm4)
```

```
Parameter   | Coefficient |   SE |          95% CI | t(28) |      p
-----------------------------------------------------------------
(Intercept) |       28.65 | 1.59 | [ 25.40, 31.90] | 18.04 | < .001
hp          |       -0.02 | 0.02 | [ -0.06,  0.01] | -1.56 | 0.130
cyl [6]     |       -5.97 | 1.64 | [ -9.33, -2.61] | -3.64 | 0.001
cyl [8]     |       -8.52 | 2.33 | [-13.29, -3.76] | -3.66 | 0.001
```

```
lm4_pred <- estimate_relation(lm4)
plot(lm4_pred)
```



5.11 Watch out for Simpson

Beware! Model estimates can swing wildly if you add (or remove) some predictor from your model. See this post for an demonstration.

5.12 What about correlation?

Correlation is really a close cousin to regression. In fact, regression with standardized variables amounts to correlation.

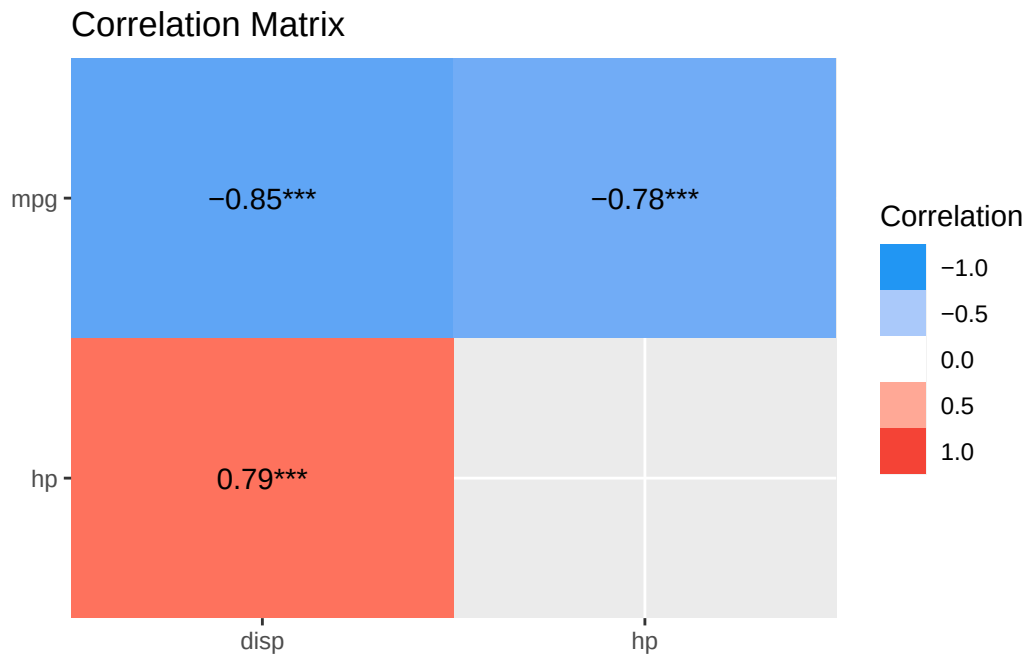
Let's get the correlation matrix of the variables in involved in `lm4`.

```
lm4_corr <-
  mtcars %>%
  select(mpg, hp, disp) %>%
  correlation()
```

```
lm4_corr
```

```
# Correlation Matrix (pearson-method)

Parameter1 | Parameter2 |     r |         95% CI | t(30) |         p
-------------------------------------------------------------------
mpg        |         hp | -0.78 | [-0.89, -0.59] | -6.74 | < .001***
mpg        |       disp | -0.85 | [-0.92, -0.71] | -8.75 | < .001***
hp         |       disp |  0.79 | [ 0.61,  0.89] |  7.08 | < .001***

p-value adjustment method: Holm (1979)
Observations: 32
```

```
plot(summary(lm4_corr))
```



5.13 Exercises

1. mtcars simple 1
2. mtcars simple 2
3. mtcars simple 3

5.14 Lab

Get your own data, and build a simple model reflecting your research hypothesis. If you are lacking data (or hypothesis) get something close to it.

5.15 Literature

An accessible treatment of regression is provided by Ismay and Kim (2020).

Roback and Legler (2021) provide a more than introductory account of regression while being accessible. A recent but already classic book (if this is possible) is the book by Gelman, Hill, and Vehtari (2021). You may also benefit from Poldrack (2022) (open access).

5.16 Debrief

Science!

6 More lineare models

6.1 R-packages needed

6.2 R packages needed for this chapter

```
library(easystats)
library(tidyverse)
```

6.3 Multiplicative associations

6.3.1 The Log-Y model

Consider again the linear model, in a simple form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + ... + b_k x_k +$$

Surprisingly, we can use this *linear* model to describe *multiplicative* assocations:

$\hat{y} = e^{b_0 + b_1 x_1 + b_2 x_2 + ... + b_k x_k}$

(I wrote `b` instead of β just to show that both has its meaning, but are separate things.)

Exponentiate both sides to get:

$log(\hat{y}) = b_0 + b_1 x_1 + b_2 x_2 + ... + b_k x_k$

For simplicity, let's drop the subscripts in the following without loss of generality and keep it short:

$y = e^x$, with $e \approx 2.71...$

Exponentiate both sides to get:

$log(y) = x$

This association is called multiplicative, because if x increases by 1, y increased by a *constant factor*.

> **Note**
>
> The logarithm is not defined for negative (input) values. And $log(0) = -\infty$.

A side-effect of modelling `log_y` instead of `y` is that the distribution shape of the outcome variable changes. This can be useful at times.

Log-Y Regression can usefully be employed for modelling growth, among othrs, see Example 6.1.

Example 6.1 (Bacteria growth)**.** Some bacteria dish grows with at a fixed proportional rate, that is it doubles its population size in a fixed period of time. This is what is called exponential growth. For concreteness, say, the bacteriae double each two days, starting with 1 unit of bacteria.

After about three weeks, we'll have this number (of units) of bacteriae:

```
e <- 2.7178
e^10
```

```
[1] 21987.45
```

6.3.2 Exercise

- Effect of education on income
- Effect of log-y transformation on the distribution, an example

> **Note**
>
> The exercises are written in German Language. Don't fret. Browsers are able to translate websites instantaneously. Alternatively, go to sites such as Google Translate and enter the URL of the website to be translated. Also check out the webstor of your favorite browser to get an extention such as this one for Google Chrome.

6.3.3 Visualizing Log Transformation

Check out this post for an example of a log-y regression visualized.

This post puts some more weight to the argument that a log-y transformation is useful (if you want to model multiplicative relations).

6.3.4 Further reading

Check out this great essay by Kenneth Benoit on different log-variants in regression. Also Gelman, Hill, and Vehtari (2021), chapter 12 (and others), is useful.

6.4 Interaction

6.4.1 Multiple predictors, no interaction

Regression analyses can be used with more than one predictor, see Figure Figure 6.1.

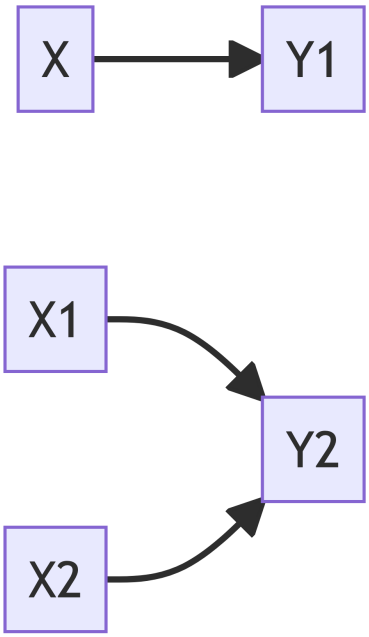


Figure 6.1: One predictor (X) vs. two predictors (X1, X2)

given by Figure **?@fig-3dregr**, where a 3D account of a regression is given. 3D means to input variables, and (which is always the case) one output variable.

> **Note**
>
> Note that the slope in linear in both axis (X1 and X2).

A different perspective is shown here,
where a 3D account of a regression is given. 3D means to input variables, and (which is always the case) one output variable.

> **Important**
>
> If the slope for one predictor is the same for all values of the other predictor, then we say that no interaction is taking place.

Here's a visualization of a 3D regression plane (not line) *without interaction*: constant slope in one axis, see the following figure, **?@fig-3dregr2**.

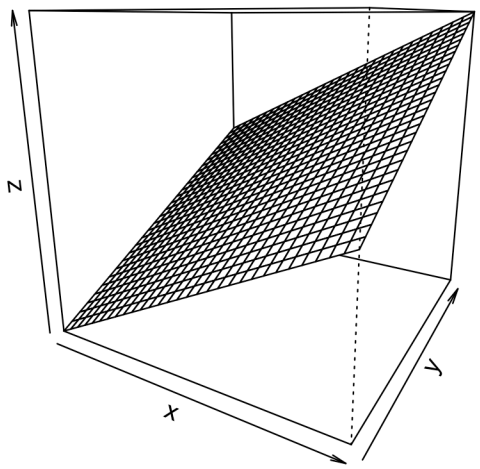


Figure 6.2: 3D regression plane (not line) without interaction

Note that in the above figure, the slope in each predictor axis equals 1, boringly. Hence the according 2D plots are boring, too.

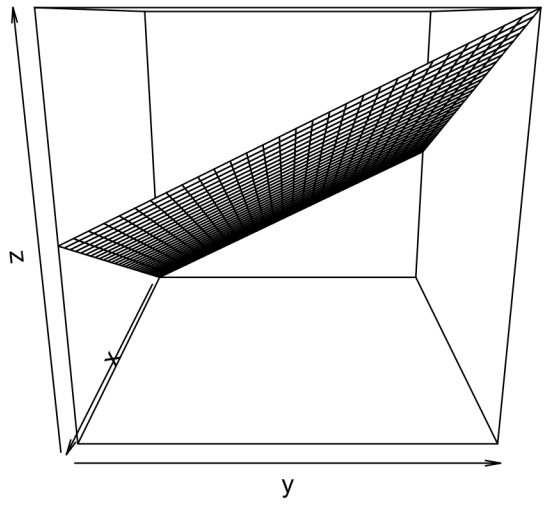


Figure 6.3: 3D regression plane (not line) without interaction

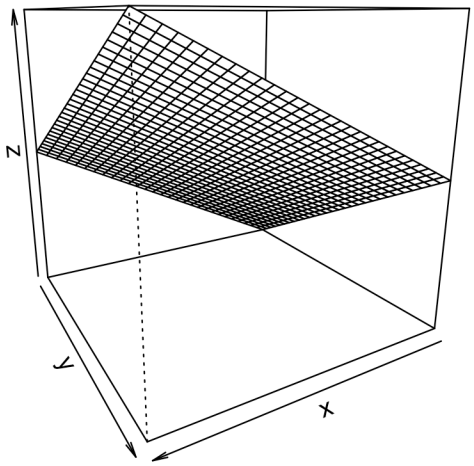


Figure 6.4: 3D regression plane (not line) without interaction

For the sake of an example, consider this linear model:

$mpg \sim hp + disp$

Or, in more regression like terms:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$, where x1 is `hp` and `x2` is `disp` in the `mtcars` dataset.

In R terms:

```
lm3d <- lm(mpg ~ hp + disp, data = mtcars)
```

The 3D plot is shown in Figure Figure 6.5.

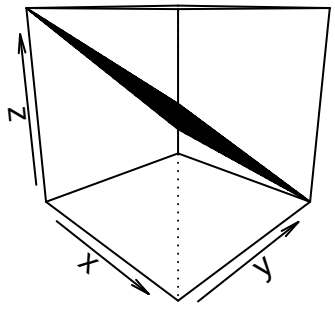
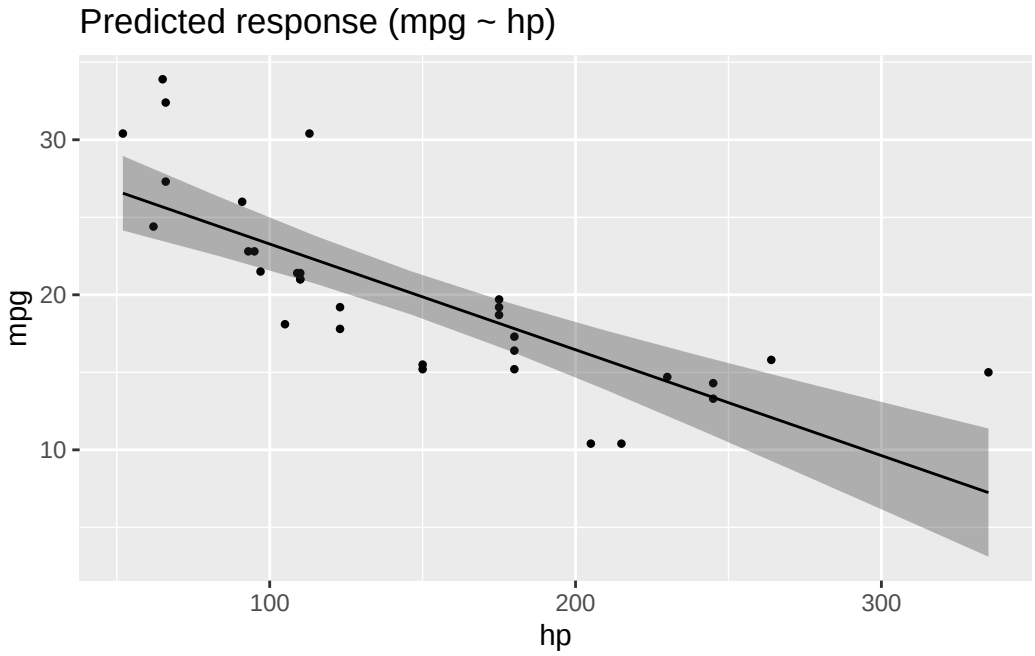


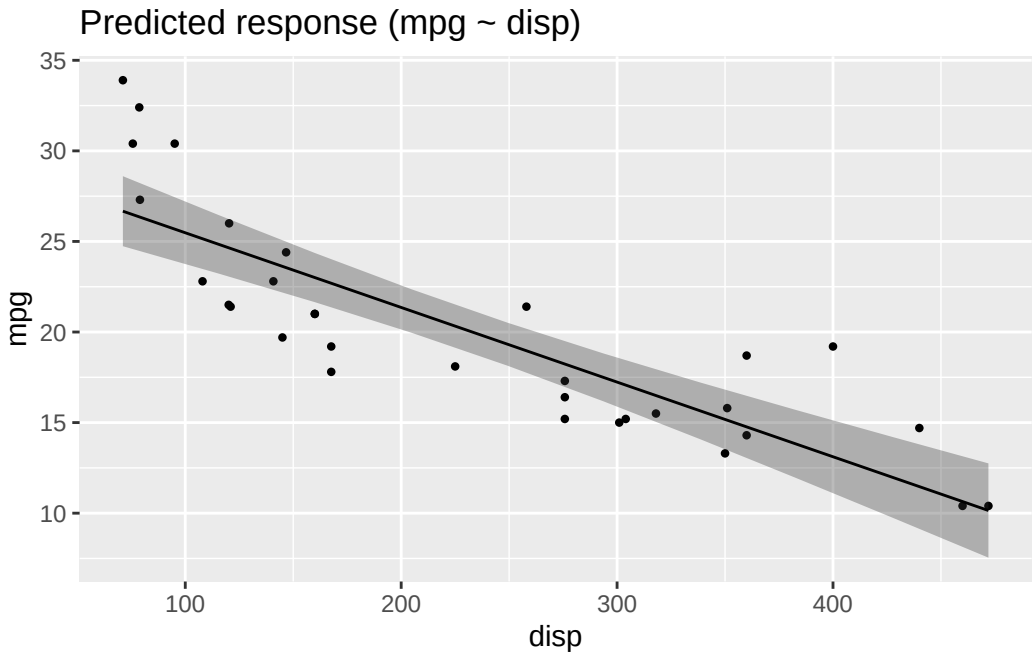
Figure 6.5: mpg ~ hp + disp

Here are the two corresponding 2d (1 predictor) regression models:

```
lm1 <- lm(mpg ~ hp, data = mtcars)
plot(estimate_relation(lm1))
```




```
lm2 <- lm(mpg ~ disp, data = mtcars)
plot(estimate_relation(lm2))
```



Checkout this post for a visually slightly more appealing 3d regression plane.

6.4.2 Interaction

For interaction to happen we relax the assumption that the slope of predictor 1 must be constant for all values of predictor 2.

In R, we specify an interaction model like this:

```
lm3d_interact <- lm(mpg ~ hp + disp + hp:disp, data = mtcars)
```

The symbol `hp:disp` can be read as "the interaction effect of `hp` and `disp`".

Here's a visual account, see Figure Figure 6.6.

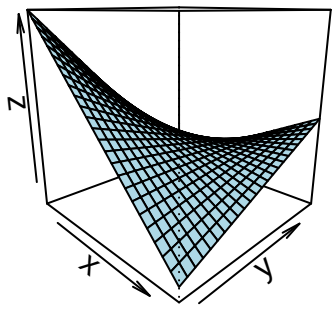


Figure 6.6: mpg ~ hp + disp

Compare Figure 6.6 and Figure 6.5.

In Figure 6.6 you'll see that the lines along the Y axis are not parallel anymore. Similarly, the lines along the X axis are not parallel anymore.

> **!** Important
>
> If the regression lines (indicating different values of one predictor) are *not* parallel, we say that an interaction effect is taking place.

However, the *difference* or *change* between two adjacent values (lines) is constant. This value is the size the regression effect.

6.4.3 Interaction made simple

If you find that two sophisticated, consider the following simple case.

First, we mutate `am` to be a factor variable, in order to make things simpler (without loss of generality).

```
mtcars2 <-
  mtcars %>%
  mutate(am_f = factor(am))
```

Now we use this new variable for a simple regression model:

```
lm_interact_simple <- lm(mpg ~ disp + am_f + disp:am_f, data = mtcars2)
```

Here's the plot, Figure Figure 6.7.

```
plot(estimate_relation(lm_interact_simple))
```

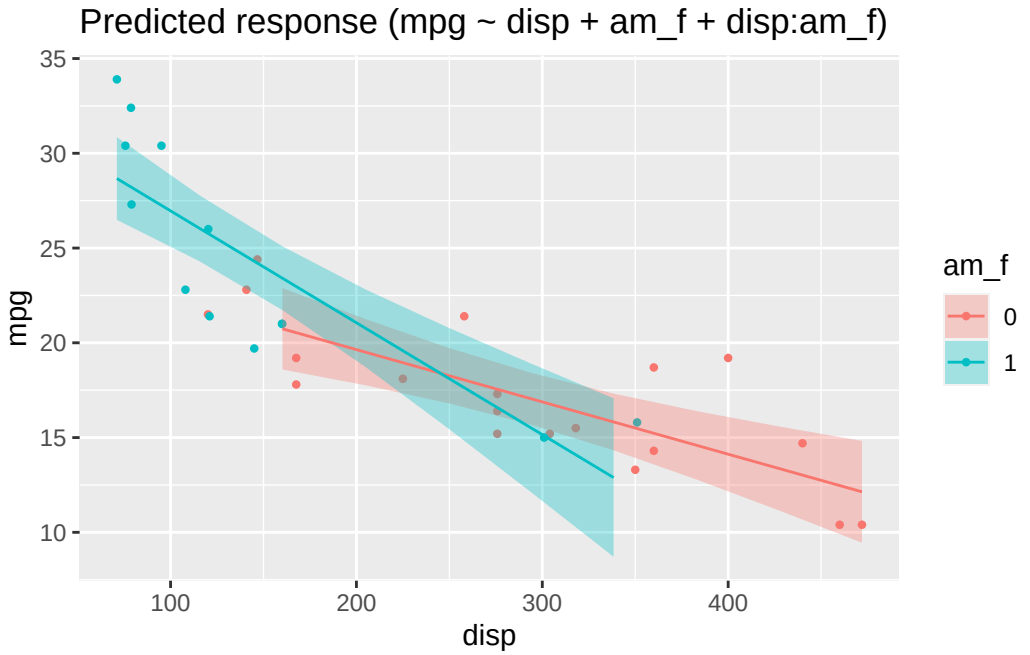


Figure 6.7: A simple interaction model

In this picture, we see that the two regression lines are *not* parallel, and hence there is evidence of an interaction effect.

The interaction effect amounts to the *difference* in slops in Figure Figure 6.7.

One might be inclined to interpret Figure Figure 6.7 as an 3D image, where the one (reddish) line is in the foreground and the blueish line in the background (or vice versa, as you like).

Given a 3D image (and hence 2 predictors), we are where we started further above.

For completeness, here are the parameters of the model.

Parameter	Coefficient	SE	95% CI	t(28)	p
(Intercept)	25.16	1.93	(21.21, 29.10)	13.07	< .001
disp	-0.03	6.22e-03	(-0.04, -0.01)	-4.44	< .001
am f (1)	7.71	2.50	(2.58, 12.84)	3.08	0.005
disp * am f (1)	-0.03	0.01	(-0.05, -7.99e-03)	-2.75	0.010

6.4.4 Centering variables

The effect of of `am_f` must be interpreted when `disp` is zero, which does not make much sense.

Therefore it simplifies the interpretation of regression coefficients to *center* all input variables, by subtrating the mean value ("demeaning" or "centering"):

$$x' = x - \bar{x}$$

In R, this can be achieved e.g,. in this way:

```
mtcars3 <-
mtcars2 %>%
  mutate(disp_c = disp - mean(disp))
```

```
lm_interact_simple2 <- lm(mpg ~ disp_c + am_f + disp_c:am_f, data = mtcars3)
parameters(lm_interact_simple2)
```

```
Parameter         | Coefficient |       SE |         95% CI | t(28) |      p
-------------------------------------------------------------------------------
(Intercept)       |       18.79 |     0.76 | [17.23, 20.36] | 24.63 | < .001
disp c            |       -0.03 | 6.22e-03 | [-0.04, -0.01] | -4.44 | < .001
am f [1]          |        0.45 |     1.39 | [-2.40,  3.30] |  0.32 | 0.748
disp c * am f [1] |       -0.03 |     0.01 | [-0.05, -0.01] | -2.75 | 0.010
```

6.5 Predictor relevance

Given a model, we might want to know which predictor has the strongest association with the outcome?

In order to answer this question, all predictor must have the same scale. Otherwise the importance of a predictor would increase by 1000, if we multiply each of the observations' values by the same factor. However, this multiplication should not change the relevance of a predictor.

A simple solution is to standardize all predictors to the same scale (sd=1).

```
mtcars4 <-
  mtcars %>%
  standardize(select = c("disp", "hp", "cyl"))
```

By the way, "standardizing" centers the variable by default to a mean value of zero (by de-meaning).

See:

```
head(mtcars4$disp)
```

```
[1] -0.57061982 -0.57061982 -0.99018209  0.22009369  1.04308123 -0.04616698
```

```
head(mtcars$disp)
```

```
[1] 160 160 108 258 360 225
```

Here's the SD:

```
sd(mtcars4$disp)
```

```
[1] 1
```

```
sd(mtcars$disp)
```

```
[1] 123.9387
```

And here's the mean value:

```
mean(mtcars4$disp)
```

```
[1] -9.084937e-17
```

```r
mean(mtcars$disp)
```

```
[1] 230.7219
```

Now we are in a position to decide which predictor is more important:

```r
m <- lm(mpg ~ disp + hp + cyl, data = mtcars4)
parameters(m)
```

```
Parameter   | Coefficient |   SE |         95% CI | t(28) |      p
-----------------------------------------------------------------
(Intercept) |       20.09 | 0.54 | [18.98, 21.20] | 37.20 | < .001
disp        |       -2.33 | 1.29 | [-4.98,  0.31] | -1.81 | 0.081
hp          |       -1.01 | 1.00 | [-3.06,  1.05] | -1.00 | 0.325
cyl         |       -2.19 | 1.42 | [-5.11,  0.72] | -1.54 | 0.135
```

6.6 Exercises

- Predictor relevance
- Adjusting
- Adjusting 2
- Interpreting Regression coefficients

6.7 Lab

Get your own data, and build a simple model reflecting your research hypothesis based on the topics covered in this chapter. If you are lacking data (or hypothesis) get something close to it.

6.8 Glimpse on parameter estimation

An elegant yet simple explanation of the math of parameter estimation can be found at "go data driven". A similar approach is presented here.

Consider this geometric interpretation of the least square method in Figure Figure 6.8.

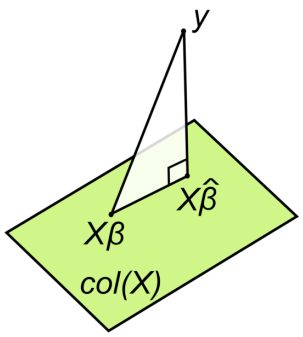


Figure 6.8: Geometric interpretation of the least square method. Source: Oleg Alexandrov on Wikimedia

6.9 Literatur

A recent but already classic book on regression and inference (if this is possible) is the book by Gelman, Hill, and Vehtari (2021). A great textbook on statistical modelling (with a Bayesian flavor) was written by McElreath (2020); it's suitable for PhD level.

Mathematical foundations can be found in Deisenroth, Faisal, and Ong (2020). Here's a collection of online resources tapping into statistics and machine learning.

7 Causality

7.1 R packages needed for this chapter

```
library(tidyverse)
library(ggdag)  # optional
```

7.2 Intro to causality

Check out this talk.

7.3 Literature

Rohrer (2018) provides an accessible introduction to causal inference. Slightly more advanced is the introduction by one of the leading figures of the Field, Judea Pearl, Pearl, Glymour, and Jewell (2016). If your after a text book on modelling that covers causal inference, and if you like Bayesian statistics, than you should definitely check out McElreath (2020).

8 Case studies

8.1 Case studies on explorative data analysis

FALLSTUDIEN - NUR EXPLORATIVE DATENANALYSE

- Datenjudo mit Pinguinen
- Data-Wranglinng-Aufgaben zur Lebenserwartung
- Case study: data vizualization on flight delays using tidyverse tools
- Aufgabe zur Datenvisualisierung des Diamantenpreises
- Fallstudie Flugverspätungen - EDA
- Fallstudie zur EDA: Top-Gear
- Fallstudie zur EDA: OECD-Wellbeing-Studie
- Fallstudie zur EDA: Movie Rating
- Fallstudie zur EDA: Women in Parliament
- Finde den Tag mit den meisten Flugverspätungen, Datensatz 'nycflights13'
- Cleaning and visualizing genomic data: a case study in tidy analysis
- Tidyverse Case Study: Exploring the Billboard Charts
- Analyse einiger RKI-Coronadaten: Eine reproduzierbare Fallstudie
- OpenCaseStudies - Health Expenditure
- Open Case Studies: School Shootings in the United States - includes dashboards
- Open Case Studies: Disparities in Youth Disconnection
- YACSDA Seitensprünge
- The Open Case Study Search provides a nice collection of helpful case studies.
- ifes@FOM Fallstudienseite

8.2 Case studies on linear modesl

FALLSTUDIEN - NUR LINEARE MODELLE

- Beispiel für Prognosemodellierung 1, grundlegender Anspruch, Video
- Beispiel für Ihre Prognosemodellierung 2, mittlerer Anspruch
- Beispiel für Ihre Prognosemodellierung 3, hoher Anspruch
- Fallstudie: Modellierung von Flugverspätungen
- Modelling movie successes: linear regression
- Movies
- Fallstudie Einfache lineare Regression in Base-R, Anfängerniveau, Kaggle-Competition TMDB
- Fallstudie Sprit sparen
- Fallstudie zum Beitrag verschiedener Werbeformate zum Umsatz; eine Fallstudie in Python, aber mit etwas Erfahrung wird man den Code einfach in R umsetzen können (wenn man nicht in Python schreiben will)
- Practical Linear Regression with R: A case study on diamond prices
- Case Study: Italian restaurants in NYC
- Vorhersage-Modellierung des Preises von Diamanten
- Modellierung Diamantenpreis 2

8.3 Case studies on machine learning using tidymodels

FALLSTUDIEN - MASCHINELLES LERNEN MIT TIDYMODELS

- Experimenting with machine learning in R with tidymodels and the Kaggle titanic dataset
- Tutorial on tidymodels for Machine Learning
- Classification with Tidymodels, Workflows and Recipes
- A (mostly!) tidyverse tour of the Titanic
- Personalised Medicine - EDA with tidy R
- Tidy TitaRnic

- Fallstudie Seegurken
- Sehr einfache Fallstudie zur Modellierung einer Regression mit tidymodels
- Fallstudie zur linearen Regression mit Tidymodels
- Analyse zum Verlauf von Covid-Fällen
- Fallstudie zur Modellierung einer logististischen Regression mit tidymodels
- Fallstudie zu Vulkanausbrüchen
- Fallstudie Himalaya
- Fallstudien zu Studiengebühren
- 1. Modell der Fallstudie Hotel Bookings
- Aufgaben zur logistischen Regression, PDF
- Fallstudie Oregon Schools
- Fallstudie Windturbinen
- Fallstudie Churn
- Einfache Durchführung eines Modellierung mit XGBoost
- Fallstudie Oregon Schools
- Fallstudie Churn
- Fallstudie Ikea
- Fallstudie Wasserquellen in Sierra Leone
- Fallstudie Bäume in San Francisco
- Fallstudie Vulkanausbrüche
- Fallstudie Brettspiele mit XGBoost
- Fallstudie Serie The Office
- Fallstudie NBER Papers
- Fallstudie Einfache lineare Regression mit Tidymodels, Kaggle-Competition TMDB
- Fallstudie Einfaches Random-Forest-Modell mit Tidymodels, Kaggle-Competition TMDB
- Fallstudie Workflow-Set mit Tidymodels, Kaggle-Competition TMDB
- Fallstudie Titanic mit Tidymodels bei Kaggle
- Einfache Fallstudie mit Tidymodels bei Kaggle

- Exploring the Star Wars "Prequel Renaissance" Using tidymodels and workflowsets

References

Deisenroth, Marc Peter, A. Aldo Faisal, and Cheng Soon Ong. 2020. *Mathematics for Machine Learning*. Cambridge ; New York, NY: Cambridge University Press.

Gelman, Andrew, Jennifer Hill, and Aki Vehtari. 2021. *Regression and Other Stories*. Analytical Methods for Social Research. Cambridge: Cambridge University Press.

Goodman, Steven. 2008. "A Dirty Dozen: Twelve p-Value Misconceptions." *Seminars in Hematology*, Interpretation of quantitative research, 45 (3): 135–40. https://doi.org/10.1053/j.seminhematol.2008.04.003.

Hernán, Miguel A., John Hsu, and Brian Healy. 2019. "A Second Chance to Get Causal Inference Right: A Classification of Data Science Tasks." *Chance* 32 (1): 42–49. https://doi.org/10.1080/09332480.2019.1579578.

Ismay, Chester, and Albert Young-Sun Kim. 2020. *Statistical Inference via Data Science: A ModernDive into r and the Tidyverse*. Chapman & Hall/CRC the r Series. Boca Raton: CRC Press / Taylor & Francis Group. https://moderndive.com/.

MacKay, R. J., and R. W. Oldford. 2000. "Scientific Method, Statistical Method and the Speed of Light." *Statistical Science* 15 (3): 254–78. https://doi.org/10.1214/ss/1009212817.

McElreath, Richard. 2020. *Statistical Rethinking: A Bayesian Course with Examples in r and Stan*. 2nd ed. CRC Texts in Statistical Science. Boca Raton: Taylor; Francis, CRC Press.

Pearl, Judea, Madelyn Glymour, and Nicholas P. Jewell. 2016. *Causal Inference in Statistics: A Primer*. Chichester, West Sussex: Wiley.

Poldrack, Russell. 2022. *Statistical Thinking for the 21st Century*. https://statsthinking21.github.io/statsthinking21-core-site/index.html.

Roback, Paul, and Julie Legler. 2021. *Beyond Multiple Linear Regression: Applied Generalized Linear Models and Multilevel Models in*. 1st ed. Chapman and Hall Texts in Statistical Science. Boca Raton: CRC Press.

Rohrer, Julia M. 2018. "Thinking Clearly about Correlations and Causation: Graphical Causal Models for Observational Data." *Advances in Methods and Practices in Psychological Science* 1 (1): 27–42. https://doi.org/10.1177/2515245917745629.

Sauer, Sebastian. 2019. *Moderne Datenanalyse Mit r: Daten Einlesen, Aufbereiten, Visualisieren Und Modellieren*. 1. Auflage 2019. FOM-Edition. Wiesbaden: Springer. https://www.springer.com/de/book/9783658215866.

Wasserstein, Ronald L., and Nicole A. Lazar. 2016. "The ASA's Statement on p-Values: Context, Process, and Purpose." *The American Statistician* 70 (2): 129–33. https://doi.org/10.1080/00031305.2016.1154108.

Wasserstein, Ronald L., Allen L. Schirm, and Nicole A. Lazar. 2019. "Moving to a World

Beyond ‘ p</i> p < 0.05’.” *The American Statistician* 73 (March): 1–19. https://doi.org/10.1080/00031305.2019.1583913.

Wickham, Hadley, and Garrett Grolemund. 2016. *R for Data Science: Visualize, Model, Transform, Tidy, and Import Data*. O’Reilly Media. https://r4ds.had.co.nz/index.html.

Wild, Chris J, and Maxine Pfannkuch. 1999. “Statistical Thinking in Empirical Enquiry.” *International Statistical Review* 67 (3): 223–48.