Statistik_21

Sebastian Sauer

2017-01-26

Contents


```
source("./source/libs.R")
```

Vorwort

Es gibt noch kein gutes Buch in deutscher Sprache zu den Grundlagen moderner Statistik, auch "Data Science" genannt. Dieses Buch soll helfen, einen Teil dieser Lücke zu füllen. Die Zielgruppe sind Analysatoren mit praktischem, wirtschaftsnahem Hintergrund. Auf mathematische Hintergründe wird großteils verzichtet; Matheliebhaber werden kaum auf ihre Kosten kommen. Im Blick habe ich (hier spricht der Autor) Anwender, die einen Freischwimmer in der modernen Datenanalyse erlernen möchten (oder müssen, liebe Studierende).

Dieses Buch wurde mit dem Paket `bookdown` [bookdown] erstellt, welches wiederum stark auf den Paketen `knitr` (Xie, 2015) und `rmarkdown` (**?**) beruht. Diese Pakete stellen verblüffende Funktionalität zur Verfügung als freie Software (frei wie in Bier und frei wie in Freiheit).

- Worum geht es in diesem Buch
  - Einführung in moderne Verfahren der Statistik
  - Für Praktiker
  - Betonung liegt auf "modern" und "Praktiker"
- Ziel des Buches
  - Intuitives, grundlegendes Verständnis zu zentralen Konzepten
  - Handwerkszeug zum selber Anwenden
- Unterschied zu anderen Büchern
  - Wenig Formeln
  - Keine/weniger "typischen" klassischen Methoden wie ANOVA, Poweranalyse etc.
  - Aufzeigen von Problemen mit klassischen Verfahren
  - Kritik am Status-Quo
- Didaktik
  - Hands-on
  - R
  - Lernfragen
  - Fallstudien
  - Aktuelle Entwicklungen ausgerichtet

```
library(knitr)
```

Einführung

0.1 Rahmen

Der "Rahmen" dieses Buches ist der Überblick über wesentliche Schritte der Datenanalyse (aus meiner Sicht). Es gibt viele Ansätze, mit denen der Ablauf von Datenanalyse dargestellt wird. Der im Moment populärste oder bekannteste ist wohl der von Hadley Wickham und Garret Grolemund (Wickham and Grolemund, 2016). Hadley und Garrett haben einen "technischeren" Fokus als der dieses Buches. Ihr Buch "R for Data Science" ist hervorragend (und frei online verfügbar); nur ist der Schwerpunkt ein anderer; es baut ein viel tieferes Verständnis von R auf. Hier spielen aber statistisch-praktisch und statistisch-philosophische[1] Aspekte eine größere Rolle.

Das Diagramm @ref("Rahmen") stellt den Rahmen dieses Buch dar: Die drei Hauptaspekte sind *Umformen*, *Visualisieren* und *Modellieren*. Dies ist vor dem Hintergrund der *Reproduzierbarkeit* eingebettet. Dieser Rahmen spiegelt das hier vertretene Verständnis von Datenanalyse wieder, wobei es sich nicht unbedingt um eine Abfolge von links nach rechts handeln muss. Wilde Sprünge sind erlaubt und nicht unüblich.

Mit *Umformen* ist gemeint, dass Daten in der Praxis häufig nicht so sind, wie man sie gerne hätte. Mal fehlt eine Variable, die den Mittelwert anderer ausdrückt, oder es gibt unschöne "Löcher", wo starrsinnige Versuchspersonen standhaft keine Antwort geben wollten. Die Zahl an Problemen und (Arten von) Fehlern übersteigt sicherlich die Anzahl der Datensätze. Kurz: Wir sehen uns gezwungen, den Daten einige Einblick abzuringen, und dafür müssen wir sie erst in Form bringen, was man als eine Mischung zwischen Artistik

[1] zumindest bei den meisten Befehlen.

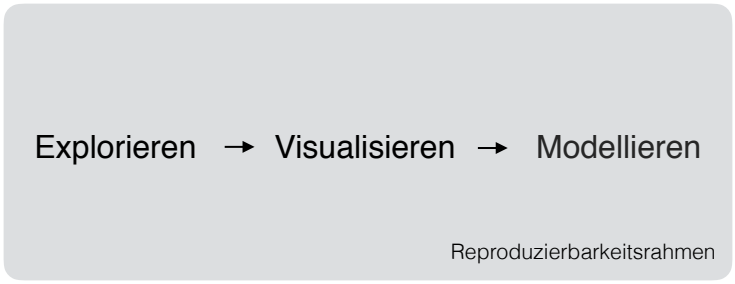


Figure 1: Rahmen

und Judo verstehen kann. Ach ja, die deskriptive Statistik fristet (in diesem Buch) eine untergeordnete Rolle in diesem Schritt.

Dann die *Visualisierung*. Ein Bild sagt mehr als 1000 Worte, weiß der Volksmund. Für die Datenanalyse gilt dies auch. Ein gutes Diagramm vermittelt eine Fülle an Informationen "auf einen Blick" und erzielt damit eine Syntheseleistung, die digitalen Darbietungsformen, sprich: Zahlen, verwehrt bleibt Nebenbei sind Diagramme, mit Geschick erstellt, ein Genuss für das Auge, daher kommt der Visualisierung großen Wert zu.

Als letzten, aber wesentlichen Punkt führen wir das *Modellieren* an. Es gibt mehr Definitionen von "Modell" als ich glauben wollte, aber hier ist damit gemeint, dass wir uns eine Geschichte ausdenken, wie die Daten entstanden sind, oder präziser gesagt: welcher Mechanismus hinter den Daten steht. So könnten wir Klausurnoten und Lernzeit von einigen Studenten[2] anschauen, und verkünden, wer mehr lerne, habe auch bessere Noten (ein typischer Dozentenauspruch). Unser Modell postuliert damit einen (vielleicht linearen) Anstieg des Klausurerfolgs bei steigender Vorbereitungszeit. Das schönste an solchen Modellen ist, dass wir Vohersagen treffen können. Zum Beispiel: "Joachim, du hast 928 Stunden auf die Klausur gelernt; damit solltest du 93% der Punkte erzielen".

Was ist dann mit dem *Reproduzierbarkeitshintergrund* gemeint? Ihre Arbeiten von Umformen, Visualisieren und Modellieren sollten sich nicht ausschließlich im Arbeitsspeicher Ihres Gehirns stattfinden, auch wenn das bei Ihnen, lieber Leser, vielleicht schneller ginge. Stattdessen soll der Mensch sich Mühe machen, seine Gedanken aufzuschreiben, hier insbesondere die Rechnungen bzw. alles, was den Daten angetan wurde, soll protokolliert werden (auch die Ergebnisse, aber wenn der Weg dorthin klar protokolliert ist, kann man die Ergebnisse ja einfach "nachkochen"). Ein Vorteil dieses Vorgehens ist, dass andere (inklusive Ihres zukünftigen Ich) die Ergebnisse bzw. das Vorgehen einfacher nachvollziehen können.

0.2 Was ist Statistik? Wozu ist sie gut?

Zwei Fragen bieten sich sich am Anfang der Beschäftigung mit jedem Thema an: Was ist die Essenz des Themas? Warum ist das Thema (oder die Beschäftigung damit) wichtig?

Was ist Stististik? *Eine* Antwort dazu ist, dass Statistik die Wissenschaft von Sammlung, Analyse, Interpretation und Kommunikation mithilfe mathematischer Verfahren ist und zur Entscheidungshilfe beitragen solle (oxf, 2006; Romeijn, 2016). Damit hätten wir auch den Unterschied zur schnöden Datenanalyse (ein Teil der Statistik) herausgemeiselt. Statistik wird häufig in die zwei Gebiete *deskriptive* und *inferierende* Statistik eingeteilt. Erstere fasst viele Zahlen zusammen, so dass wir den Wald statt vieler Bäume sehen. Letztere verallgemeinert von den vorliegenden (sog. "Stichproben-")Daten auf eine zugrunde liegende Grundmenge (Population). Dabei spielt die Wahrscheinlichkeitsrechnung und Zufallsvariablen eine große Rolle.

Auch wenn die gerade genannte Diskussion die häufigste oder eine typische ist, mehren sich doch Stimmen, die Statstik anders akzentuieren. So schreibt Briggs in einem aktuellen Buch (Briggs, 2016), dass es in der Statistik darum ginge, die Wahrscheinlichkeit zukünftiger Ereignisse vorherzusagen: "Wie wahrscheinlichi ist es, dass - gegeben einem statistischen Modell, allerlei Annahmen und einem Haufen Daten - Kandidat X der neue Präsident wird"[3]? Das schöne an dieser Idee ist, dass das "Endprodukt" etwas sehr Weltliches und damit praktisches ist: Die Wahrscheinlichkeit einer interessanten (und unbekannten) Aussage. Nebenbei ruht diese Idee auf dem sicheren Fundament der Wahrscheinlichkeitstheorie.

Abgesehen von philosophischen Überlegungen zum Wesen der Statistik kann man sagen, dass Vorhersagen von Ereignissen etwas sehr praktisches sind. Sie nehmen daher aus praktischen Überlegungen einen zentralen Platz in diesem Buch an. Die philosophische Relevanz des prädiktiven Ansatzes ist gut bei Briggs (Briggs, 2016, 2008) nachzulesen.

[2] hier und überall sind Frauen und Männer gleichermaßen angesprochen; aus Gründen der Lesbarkeit das generische Maskulinum bevorzugt.

[3] Eine Vorhersage, die bei vielen Vorhersagemodellen komplett in die Grütze ging, wenn man sich die US-Präsidentenwahl 2016 anschaut.

Traditionell ist die Statistik stark daran interessiert, Parameter von Populationen vorherzusagen. Ein Beispiel dazu wäre die mittlere Größe (Parameter) aller Deutschen (Population). Leider sind Populationen häufig ziemlich abstrakt. Nehmen wir als Beispiel an, ein Dozent der FOM (Prof. S.) wie sich der Lernerfolg ändert, wenn die Stoffmenge pro Stunde verdoppelt. Zu seiner Überraschung ist der Lernerfolg geringer als in einem Kontrollkurs. Auf welche Population ist jetzt die Studie bzw. die Daten seiner Stichprobe zu verallgemeinern? Alle Menschen? Alle Studierenden? Alle deutschen Studierenden? Alle Studierenden der FOM? Alle Studierenden aller Zeiten?

- Statistik meint Methoden, die das Ziel haben, Ereignisse präzise vorherzusagen
- Statistik soll sich um Dinge dieser Welt drehen, nicht um Parameter
- Statt einer Frage "ist µ_1 größer als µ_2?" besser "Wie viel Umsatz erwarte ich von diesem Kunden?", "Wie viele Saitensprünge hatte er wohl?", "Wie groß ist die Wahrscheinlichkeit für sie zu überleben?" und dergleichen.
- Der Nutzen von Vorhersagen liegt auf der Hand: Vorhersagen sind praktisch; eine nützliche Angelegenheit (wenn auch schwierig).

0.3 Aufbau des Buches

sdkljf

0.4 Datensätze

Name des Datensatzes | Quelle | Beschreibung `profiles` | {okcupiddata} | Daten von einer Online-Singlebörse

Grundlagen

In diesem Kapitel diskutieren wir einige zentrale Begriffe der Wissenschaft bzw. der quantitativen Methodik der Wissenschaft.

0.5 Wahrscheinlichkeit

Was ist Wahrscheinlichkeit und (warum) ist sie wichtig? Wo wir schon bei den großen Fragen sind, können wir noch eins drauf setzen: Was ist das Ziel von Wissenschaft? Eine einfache Antwort auf diese Frage ist, die Wahrheit von Aussagen zu bestimmen. Zum Beispiel: "Ein Proton besteht aus siebenunzwanzig Dscharbs" oder "Deutsche Frauen verdienen im Schnitt weniger als Männer" oder "Morgen wird es regnen". Leider ist es oft nicht möglich, sichere Aussagen über die Natur zu bekommen. In der Logik oder Mathe ist dies einfacher: "Joachim Z. ist eine Mensch und alle Menschen sind sterblich" (A) erlaubt die Ableitung "Joachim Z. ist sterblich" (B). Wir haben soeben eine wahre Aussage abgeleitet. Die Aussage ist sicher wahr, also zu 100%. Die Verneinung dieser Aussage B ist sicher falsch; wir sind uns zu 100% sicher, dass die Verneinung von falsch ist.

Aussagen wie die vom Regen morgen sind nicht sicher, wir sind nicht zu 100% gewiss, dass es morgen regnet. Der kühnste Wetterfrosch auch nicht. Genauso gilt, dass wir nicht zu 0% sicher sind; dies hieße, dass das ~~Regenteil~~ sicher ist. Wir brauchen also eine Methode, *Ungewissheit* auszudrücken. Das ist die Aufgabe der Wahrscheinlichkeit. Sie erlaubt Gewissheitsgrade zwischen 0% und 100%, etwa "P(Regen morgen | Daten, Modell, Randbedingungen) = 84%". In Worten: "Die Wahrscheinlichkeit, dass es morgen in einem spezifizierten Gebiet regnet, gegeben meine Daten, mein Modell und sonstige Randbedingungen, die man leicht vergisst, die aber auch wichtig sind, liegt bei 84%".

> Wahrscheinlichkeit ist ein *epistemologisches* Konzept. Wahrscheinlichkeit beschreibt keine (physikalischen) Tatsachen über diese Tatsachen. Wahrscheinlichkeit beschreibt unsere Ungewissheit über Behauptungen.

Ein weiteres Beispiel: P(Kopf | Wurf einer fairen Münze) = 1/2. In Worten: "Die Wahrscheinlichkeit, Kopf zu werfen, wenn man eine faire Münze hat, liegt bei 1/2". Genauer gesagt und etwas pedantisch, müsste man hinzufügen, dass wir stillschweigend vorausgesetzt haben, das Ergebnis des Wurfes nicht zu kennen. Denn: P(Kopf | Ich weiß, dass es Kopf ist) = 1. Das Beispiel zeigt, dass die Wahrscheinlichkeit eines Ereignisses von den Prämissen abhängt (das, was nach dem Strich steht). Diese Prämissen können bei Ihnen anders sein als bei; daher ist es plausibel, dass sich unsere Wahrscheinlichkeiten für P(Kopf) unterscheiden. Das ist rational (nicht subjektiv).

Dieses auf der formalen Logik basierende Konzept von Wahrscheinlichkeit (Briggs, 2016) besticht mit einer breiten Anwendungsfeld. Was halten Sie von dieser Aussage: "Wenn Hillary Clinton keine EMail-Affäre gehabt hätte, hätte sie die Wahl gewonnen"? Ist es eine sinnvolle Aussage? Es mag schwer oder unmöglich sein, an diese Aussage eine konkrete Wahrscheinlichkeit anzuheften; das ändert aber nichts daran, dass wir über eine solche Frage nachdenken können (ich glaube, viele Leute haben sich diese Frage gestellt). Fragen dieser Art könnten wir als "Rum-ums-Eck-Fragen" bezeichnen. Ein deutscher Politiker sprach in

dem Zusammenhang von "Hätte-Hätte-Fahrradkette", was zeigt, dass Aussagen *kein empirischer Gehalt* zukommen muss, die Wahrscheinlichkeit dieser Aussagen für Menschen aber von Belang ist.

Definiert man die Wahrscheinlichkeit als relative Häufigkeit, kommt man natürlich bei solchen Rum-ums-Eck-Fragen in die Bredouille, da sie nicht häufig, nämlich überhaupt nicht passiert sind. Die logische Definition hat aber kein Problem mit diesen Fragen an sich.

Oder betrachten Sie dieses Beispiel (angelehnt an Briggs, 2016): "George ist ein Marsianer; 1/3 aller Marsianer lieben französischen Weichkäse (speziell Camembert)". Wie hoch ist die Wahrscheinlichkeit, dass George, ein Marsianer, Weichkäse (speziell Camembert) liebt? Beachten Sie, dass die zu Verfügung stehende Information beachtet werden soll aber sonst keine Information. Die Antwort lautet: P = 1/3.

Ein weiteres Problem mit Wahrscheinlichkeit, die auf eine unendliche Wiederholung eines Ereignisses aufgebaut ist, ist dass Unendlich kompliziert ist. Es ist unendlich schwierig, sich unendlich viele Dinge vorzustellen oder sich ein unendlich großes Ding vorzustellen. Wenn wir uns aber nicht vorstellen können, was mit einer Aussage gemeint ist, was sagt dann diese Aussage?

Daher ist besser, Warscheinlichkeiten als Erweiterung der (formalen) Logik zu begreifen. Wo die Logik sagt, eine Aussage sei richtig oder falsch, gibt die Wahrscheinlichkeit eine Gradierung zwischen diesen beiden Extremen an.

Kurz gesagt: Wahrscheinlichkeit misst den Grad der Gewissheit (oder Ungewisseheit) einer Aussage. Eine Aussage mit einer Wahrscheinlichkeit von 100% ist eine sichere Aussage, eine Aussage, die sicher zutrifft (wahr ist); analog ist eine Aussage mit einer Wahrscheinlichkeit von 0% sicher falsch (nicht zutreffend). Die Grade dazwischen markieren die unterschiedlichen Abstufungen von Gewissheit.

0.6 Hypothesen

Hypothesen sind Aussagen. Aussagen, bei denen wir nicht sicher sind, dass sie richtig oder falsch (d.h. P=0 oder P=1). Man beachte, dass der letzte Satz epistemologisch argumentiert hat (es war eine Aussage über unser Wissen); es war keine Aussage über Tatsachen WIRKLCIH????

deren Wahrheitswert nicht extrem ist - die Wahrscheinlichkeit der Richtigkeit der Behauptung ist also größer als 0 aber kleiner als 1.

Aussagen sind Sätze, deren Wahrheitswert überprüfbar ist, zumindest potenziell. Beispiele für solche Aussagen wären "Webseiten mit Bilder sind einfacher zu lesen", "Power Posing hat einen Effekt auf den Testosteronlevel" und "Es gibt Leben auf den Mars". Das letzte Beispiel ist interessant, weil es im Moment vielleicht noch nicht im Vermögen der Forschung liegt, diesen Satz zu bestätigen oder zu widerlegen (falsifizieren). Überhaupt sind Sätze der Art "Es gibt…" schwierig zu widerlegen (manchmal geht es). Fruchtbarer sind daher Aussagen mit mehr empirischen Gehalt, die "angreifbarer" weil "gewagter" sind.

Hypothesen haben demnach den Charakter von Wahrscheinlichkeitsaussagen.

0.7 Falsifikationismus

Hm.

Trends

- Big Data
- Open Science
- Computerisierung
- Neue Methoden zur numerischen Vorhersage
- Textmining

Unbehagen

In diesem Kapitel finden sich einige Probleme, die einigen Wissenschaftlern Bauchschmerzen oder Unbehagen verursacht.

0.8 Der p-Wert

Der p-Wert ist die heilige Kuh der Forschung. Das ist nicht normativ, sondern deskriptiv gemeint. Der p-Wert entscheidet (häufig) darüber, was publiziert wird, und damit, was als Wissenschaft sichtbar ist - und damit, was Wissenschaft ist (wiederum deskriptiv, nicht normativ gemeint). Kurz: Dem p-Wert wird viel Bedeutung zugemessen.

Allerdings hat der p-Wert seine Probleme. Vor allem: Er wird missverstanden. Jetzt kann man sagen, dass es dem p-Wert (dem armen) nicht anzulasten, dass andere/ einige ihm missverstehen. Auf der anderen Seite finde ich, dass sich Technologien dem Nutzer anpassen sollten (soweit als möglich) und nicht umgekehrt. Die Definition des p-Werts ist aber auch so kompliziert, man kann sie leicht missverstehen:

> Der p-Wert gibt die Wahrscheinlichkeit P unserer Daten D an (und noch extremerer), unter der Annahme, dass die getestete Hypothese H wahr ist (und wenn wir den Versuch unendlich oft wiederholen würden, unter identischen Bedingungen und ansonsten zufällig). p = P(D|H)

Viele Menschen - inkl. Professoren und Statistik-Dozenten - haben Probleme mit dieser Definition (Gigerenzer, 2004). Das ist nicht deren Schuld: Die Definition ist kompliziert. Vielleicht denken viele, der p-Wert sage das, was tatsächlich interessant ist: die Wahrscheinlichkeit der (getesteten) Hypothese, gegeben der Tatsache, dass bestimmte Daten vorliegen. Leider ist das *nicht* die Definition des p-Werts. Also:

$$P(D|H) \neq P(H|D)$$

Der p-Wert ist für weitere Dinge kritisiert worden (Wagenmakers, 2007, Briggs (2016)); z.B. dass die "5%-Hürde" einen zu schwachen Test für die getestete Hypothese bedeutet. Letzterer Kritikpunkt ist aber nicht dem p-Wert anzulasten, denn dieses Kriterium ist beliebig, könnte konservativer gesetzt werden und jegliche mechanisierte Entscheidungsmethode kann ausgenutzt werden. Ähnliches kann man zum Thema "P-Hacking" argumentieren (Head et al., 2015, Wicherts et al. (2016)); andere statistische Verfahren können auch gehackt werden.

Meine Meinung ist, dass der p-Wert problematisch ist und nicht oder weniger benutzt werden sollte (das ist eine normative Aussage). Da der p-Wert aber immer noch der Platzhirsch auf vielen Forschungsauen ist, führt kein Weg um ihn herum. Er muss genau verstanden werden: Was er sagt und - wichtiger noch - was er nicht sagt.

0.9 Wahrscheinlichkeit im Frequentismus

Die Idee von

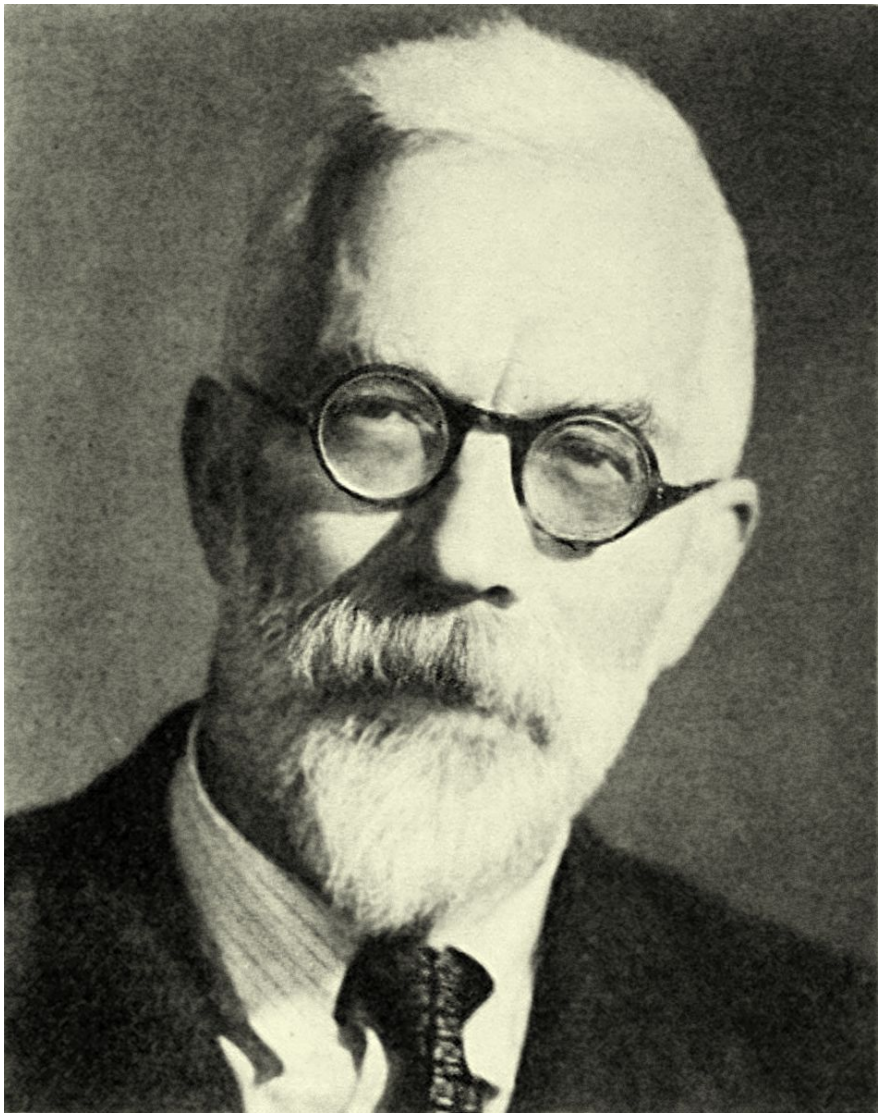


Figure 2: Der größte Statistiker des 20. Jahrhunderts ($p < .05$)

- Theorie der Wahrscheinlichkeit im Frequentismus
- Reproduzierbarkeitskrise
- Parameter
- Kausalität
- Übersicherheit

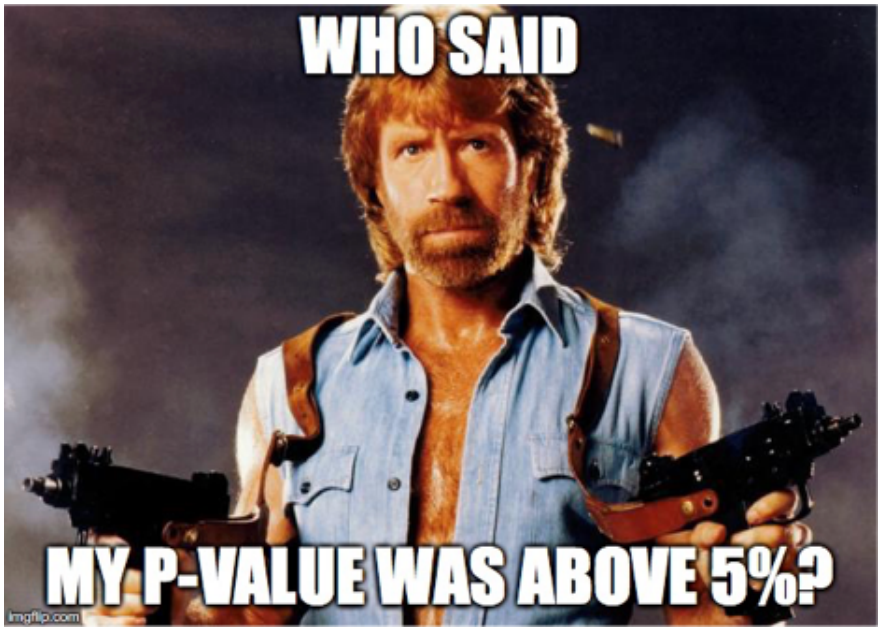


Figure 3: Der p-Wert wird oft als wichtig erachtet

Software

Als Haupt-Analysewerkzeug nutzen wir R; daneben wird uns die sog. "Entwicklungsumgebung" RStudio einiges an komfortabler Funktionalität bescheren. Eine Reihe von R-Paketen (Erweiterungen) werden wir auch nutzen. R ist eine recht alte Sprache; viele Neuerungen finden in Paketen Niederschlag, da der "harte Kern" von R lieber nicht so stark geändert wird. Stellen Sie sich vor: Seit 29 Jahren nutzen Sie eine Befehl, der Ihnen einen Mittelwert ausrechnet, sagen wir die mittlere Anzahl von Tassen Kaffee am Tag. Und auf einmal wird der Mittelwert anders berechnet?! Eine Welt stürzt ein! Naja, vielleicht nicht ganz so tragisch in dem Beispiel, aber grundsätzlich sind Änderungen in viel benutzen Befehlen potenziell problematisch. Das ist wohl ein Grund, warum sich am "R-Kern" nicht so viel ändert. Die Innovationen in R passieren in den Paketen. Und es gibt viele davon; als ich diese Zeilen schreibe, sind es fast schon 10.000! Genauer: 9937 nach dieser Quelle: `https://cran.r-project.org/web/packages/`.

0.10 R and Friends installieren

Setzt natürlich voraus, dass R installiert ist. Sie können R unter `https://cran.r-project.org` herunterladen und installieren (für Windows, Mac oder Linux). RStudio finden Sie auf der gleichnamigen Homepage: `https://www.rstudio.com`; laden Sie die "Desktop-Version" für Ihr Betriebssystem herunter.

0.11 Hilfe! R tut nicht so wie ich das will

> Manntje, Manntje, Timpe Te, Buttje, Buttje inne See, myne Fru de Ilsebill will nich so, as ik wol will. Gebrüder Grimm, Märchen vom Fischer und seiner Frau, `https://de.wikipedia.org/wiki/Vom_Fischer_und_seiner_Frau`

Ihr R startet nicht oder nicht richtig? Die drei wichtigsten Heilmittel sind:

1. Schließen Sie die Augen für eine Minute. Denken Sie gut nach, woran es liegen könnte.
2. Schalten Sie den Rechner aus und probieren Sie es morgen noch einmal.
3. Googeln.

Sorry für die schnottrigen Tipps. Aber: Es passiert allzu leicht, dass man Fehler wie diese macht:

- `install.packages(dplyr)`
- `install.packages("dliar")`
- `install.packages("derpyler")`
- `install.packages("dplyr")  # dependencies vergessen`
- Keine Internet-Verbindung
- `library(dplyr)  # ohne vorher zu installieren`

Wenn R oder RStudio dann immer noch nicht starten oder nicht richtig laufen, probieren Sie dieses:

- Sehen Sie eine Fehlermeldung, die von einem fehlenden Paket spricht (z.B. "Package 'Rcpp' not available") oder davon spricht, dass ein Paket nicht installiert werden konnte (z.B. "Package 'Rcpp' could

not be installed" oder "es gibt kein Paket namens 'Rcpp' " oder "unable to move temporary installation XXX to YYY"), dann tun Sie folgendes:

  - Schließen Sie R und starten Sie es neu.
  - Installieren Sie das oder die angesprochenen Pakete mit `install.packages("name_des_pakets", dependencies = TRUE)` oder mit dem entsprechenden Klick in RStudio.
  - Starten Sie das entsprechende Paket mit `library(paket_name)`.

- Gerade bei Windows 10 scheinen die Schreibrechte für R (und damit RStudio oder RComannder) eingeschränkt zu sein. Ohne Schreibrechte kann R aber nicht die Pakete ("packages") installieren, die Sie für bestimmte R-Funktionen benötigen. Daher schließen Sie R bzw. RStudio und suchen Sie das Icon von R oder wenn Sie RStudio verwenden von RStudio. Rechtsklicken Sie das Icon und wählen Sie "als Administrator ausführen". Damit geben Sie dem Programm Schreibrechte. Jetzt können Sie etwaige fehlende Pakete installieren.
- Ein weiterer Grund, warum R bzw. RStudio die Schreibrechte verwehrt werden könnnten (und damit die Installation von Paketen), ist ein Virenscanner. Der Virenscanner sagt, nicht ganz zu Unrecht: "Moment, einfach hier Software zu installieren, das geht nicht, zu gefährlich". Grundsätzlich gut, in diesem Fall unnötig. Schließen Sie R/RStudio und schalten Sie dann den Virenscanner *komplett* (!) aus. Öffnen Sie dann R/RStudio wieder und versuchen Sie fehlende Pakete zu installieren.
- Läuft der RCommander unter Mac nicht, dann prüfen Sie, ob Sie X11 (synonym: XQuartz) installiert haben. X11 muss installiert sein, damit der RCommander unter Mac läuft.
- Die "app nap" Funktion beim Mac kann den RCommander empfindlich ausbremsen. Schalten Sie diese Funktion aus z.B. im RCommander über `Tools - Manage Mac OS X app nap for R.app`.

0.12 Allgemeine Hinweise zur Denk- und Gefühlswelt von R

- Wenn Sie RStudio starten, startet R automatisch auch. Starten Sie daher, wenn Sie RStudio gestartet haben, *nicht* noch extra R. Damit hätten Sie sonst zwei Instanzen von R laufen, was zu Verwirrungen (bei R und beim Nutzer) führen kann.
- Ein neues R-Skript im RStudio können Sie z.B. öffnen mit `File-New File-R Script`.
- R-Skripte können Sie speichern (`File-Save`) und öffnen.
- R-Skripte sind einfache Textdateien, die jeder Texteditor verarbeiten kann. Nur statt der Endung `txt`, sind R-Skripte stolzer Träger der Endung `R`. Es bleibt aber eine schnöde Textdatei.
- Bei der Installation von Paketen mit `install.packages("name_des_pakets")` sollte stets der Parameter `dependencies = TRUE` angefügt werden. Also `install.packages("name_des_pakets", dependencies = TRUE)`. Hintergrund ist: Falls das zu installierende Paket seinerseits Pakete benötigt, die noch nicht installiert sind (gut möglich), dann werden diese sog. "dependencies" gleich mitinstalliert (wenn Sie `dependencies = TRUE` setzen).
- Hier finden Sie weitere Hinweise zur Installation des RCommanders: `http://socserv.socsci.mcmaster.ca/jfox/Misc/Rcmdr/installation-notes.html`.
- Sie müssen online sein, um Packages zu installieren.
- Die "app nap" Funktion beim Mac kann den RCommander empfindlich ausbremsen. Schalten Sie diese Funktion aus z.B. im RCommander über `Tools - Manage Mac OS X app nap for R.app`.

Verwenden Sie möglichst die neueste Version von R, RStudio und Ihres Betriebssystems. Ältere Versionen führen u.U. zu Problemen; je älter, desto Problem… Updaten Sie Ihre Packages regelmäßig z.B. mit `update.packages()` oder dem Button "Update" bei RStudio (Reiter `Packages`).

R zu lernen kann hart sein. Ich weiß, wovon ich spreche. Wahrscheinlich eine spirituelle Prüfung in Geduld und Hartnäckigkeit… Tolle Gelegenheit, sich in diesen Tugenden zu trainieren :-)

0.13 dplyr und andere Pakete installieren

Ein R-Paket, welches für die praktische Datenanalyse praktisch ist, heißt `dplyr`. Wir werden viel mit diesem Paket arbeiten. Bitte installieren Sie es schon einmal, sofern noch nicht geschehen:

```
install.packages("dplyr", dependencies = TRUE)
```

Übrigens, das `dependencies = TRUE` sagt sinngemäß "Wenn das Funktionieren von dplyr noch von anderen Paketen abhängig ist (es also Abhängigkeiten (dependencies) gibt), dann installiere die gleich mal mit".

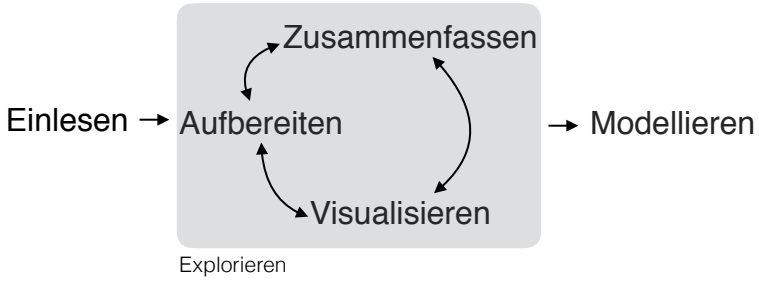
Nicht vergessen: Installieren muss man eine Software *nur einmal*; *starten* muss man sie jedes Mal, wenn man sie vorher geschlossen hat und wieder nutzen möchte:

```
library(dplyr)
```

Das Installieren und Starten anderer Pakete läuft genauso ab.

Daten explorieren

Den Ablauf des Datenexplorierens kann man so darstellen:



Zuerst müssen die Daten für die Analyse(software) verfügbar gemacht werden. Sprich, die Daten müssen *eingelesen* werden. Dann beginnt das eigentliche Explorieren; dieses kann man wiederum in drei Schritte einteilen, die keine Abfolge sind, sondern sich wild abwechseln können. Diese sind: Daten *aufbereiten*, Daten *zusammenfassen* und Daten *visualisieren*.

Unter Daten aufbereiten im engeren Sinne ist gemeint, die Daten einer "Grundreinigung" zu unterziehen, dass sie für weitere Analysen in geeigneter Form sind. Daten zusammenfassen meint die deskriptive Statistik; Daten visualisieren ist das Erstellen von Diagrammen. Im Anschluss kann man die Daten modellieren.

Ist das Explorieren von Daten auch nicht statistisch anspruchsvoll, so ist es trotzdem von großer Bedeutung und häufig recht zeitintensiv, vor allem das Daten aufbereiten. Eine Anekdote zur Relevanz der Exploration, die (so will es die Geschichte) mir an einer Bar nach einer einschlägigen Konferenz erzählt wurde (daher keine Quellenangebe, Sie verstehen…). Eine Computerwissenschaftlerin aus den USA (deutschen Ursprungs) hatte einen beeindruckenden "Track Record" an Siegen in Wettkämpfen der Datenanalyse. Tatsächlich hatte sie keine besonderen, raffinierten Modellierungstechniken eingesetzt; klassische Regression war ihre Methode der Wahl. Bei einem Wettkampf, bei dem es darum ging, Krebsfälle aus Krankendaten vorherzusagen (z.B. Röntgenbildern) fand sie nach langem Datenjudo heraus, dass in die "ID-Variablen" Information gesickert war, die dort nicht hingehörte und die sie nutzen konnte für überraschend (aus Sicht der Mitstreiter) gute Vorhersagen zu Krebsfällen. Wie war das möglich? Die Daten stammten aus mehreren Kliniken, jede Klinik verwendete ein anderes System, um IDs für Patienten zu erstellen. Überall waren die IDs stark genug, um die Anonymität der Patienten sicherzustellen, aber gleich wohl konnte man (nach einigem Judo) unterscheiden, welche ID von welcher Klinik stammte. Was das bringt? Einige Kliniken waren reine Screening-Zentren, die die Normalbevölkerung versorgte. Dort sind wenig Krebsfälle zu erwarten. Andere Kliniken jedoch waren Onkologie-Zentren für bereits bekannte Patienten oder für Patienten mit besonderer Risikolage. Wenig

überraschen, dass man dann höhere Krebsraten vorhersagen kann. Eigentlich ganz einfach; besondere Mathe steht hier (zumindest in dieser Geschichte) nicht dahinter. Und, wenn man den Trick kennt, ganz einfach. Aber wie so oft ist es nicht leicht, den Trick zu finden. Sorgfältiges Datenjudo hat hier den Schlüssel zum Erfolg gebracht.

0.14 Daten einlesen

In R kann man ohne Weiteres verschiedene, gebräuchliche (Excel) oder weniger gebräuchliche (Feather) Datenformate einlesen. In RStudio lässt sich dies z.B. durch einen schnellen Klick auf `Import Dataset` im Reiter `Environment` erledigen. Dabei wird im Hintergrund das Paket `readr` verwendet (`?`) (die entsprechende Syntax wird in der Konsole ausgegeben, so dass man sie sich anschauen und weiterverwenden kann).

Es ist für bestimmte Zwecke sinnvoll, nicht zu klicken, sondern die Syntax einzutippen. Zum Beispiel: Wenn Sie die komplette Analyse als Syntax in einer Datei haben (eine sog. "Skriptdatei"), dann brauchen Sie (in RStudio) nur alles auszuwählen und auf `Run` zu klicken, und die komplette Analyse läuft durch! Die Erfahrung zeigt, dass das ein praktisches Vorgehen ist.

Die gebräuchlichste Form von Daten für statistische Analysen ist wahrscheinlich das CSV-Format. Das ist ein einfahes Format, basierend auf einer Textdatei. Schauen Sie sich mal diesen Auszug aus einer CSV-Datei an.

```
"ID","time","sex","height","shoe_size"
"1","04.10.2016 17:58:51",NA,160.1,40
"2","04.10.2016 17:58:59","woman",171.2,39
"3","04.10.2016 18:00:15","woman",174.2,39
"4","04.10.2016 18:01:17","woman",176.4,40
"5","04.10.2016 18:01:22","man",195.2,46
```

Erkenenn Sie das Muster? Die erste Zeile gibt die "Spaltenköpfe" wieder, also die Namen der Variablen. Hier sind es 5 Spalten; die vierte heißt "shoe_size". Die Spalten sind offenbar durch Komma `,` voneinander getrennt. Dezimalstellen sind in amerikanischer Manier mit einem Punkt `.` dargestellt. Die Daten sind "rechteckig"; alle Spalten haben gleich viele Zeilen und umgekehrt alle Spalten gleich viele Zeilen. Man kann sich diese Tabelle gut als Excel-Tabelle mit Zellen vorstellen, in denen z.B. "ID" (Zelle oben links) oder "46" (Zelle unten rechts) steht.

An einer Stelle steht `NA`. Das ist Errisch für "fehlender Wert". Häufig wird die Zelle auch leer gelassen, um auszudrücken, dass ein Wert hier fehlt (hört sich nicht ganz doof an). Aber man findet alle möglichen Ideen, um fehlende Werte darzustellen. Ich rate von allen anderen ab; führt nur zu Verwirrung.

Lesen wir diese Daten jetzt ein:

```
daten <- read.csv("https://sebastiansauer.github.io/data/wo_men.csv")
head(daten)
#>                  time   sex height shoe_size
#> 1 04.10.2016 17:58:51 woman    160        40
#> 2 04.10.2016 17:58:59 woman    171        39
#> 3 04.10.2016 18:00:15 woman    174        39
#> 4 04.10.2016 18:01:17 woman    176        40
#> 5 04.10.2016 18:01:22   man    195        46
#> 6 04.10.2016 18:01:53 woman    157        37
```

Der Befehl `read.csv` liest eine CSV-Datei, was uns jetzt nicht übermäßig überrascht. Aber Achtung: Wenn Sie aus einem Excel mit deutscher Einstellung eine CSV-Datei exportieren, wird diese CSV-Datei als Trennzeichen `;` (Strichpunkt) und als Dezimaltrennzeichen `,` verwenden. Da der Befehl `read.csv` als Standard mit Komma und Punkt arbeitet, müssen wir die deutschen Sonderlocken explizit angeben, z.B. so:

```
# daten_deutsch <- read.csv("daten_deutsch.csv", sep = ";", dec = ".")
```

Dabei steht `sep` (separator) für das Trennzeichen zwischen den Spalten und `dec` für das Dezimaltrennzeichen.

Übrigens: Wenn Sie keinen Pfad angeben, so geht R davon aus, dass die Daten im aktuellen Verzeichnis liegen. Das aktuelle Verzeichnis kann man mit `getwd()` erfragen und mit `setwd()` einstellen. Komfortabler ist es aber, das aktuelle Verzeichnis per Menü zu ändern. In RStudio: `Session > Set Working Directory > Choose Directory ...` (oder per Shortcut, der dort angezeigt wird).

0.15 Datenjudo (Daten aufbereiten)

Bevor man seine Statistik-Trickkiste so richtig schön aufmachen kann, muss man die Daten häufig erst noch in Form bringen. Das ist nicht schwierig in dem Sinne, dass es um komplizierte Mathe ginge. Allerdings braucht es mitunter recht viel Zeit und ein paar (oder viele) handwerkliche Tricks sind hilfreich. Hier soll das folgende Kapitel helfen.

Mit "Datenjudo" (ein Fachbegriff aus der östlichen Zahlentheorie) ist gemeint, die Daten so "umzuformen", "aufzubereiten", oder "reinigen" , dass sie passend für statistische Analysen sind.

Typische Probleme, die immer wieder auftreten sind:

- Fehlende Werte: Irgend jemand hat auf eine meiner schönen Fragen in der Umfrage nicht geantwortet!
- Unerwartete Daten: Auf die Frage, wie viele Facebook-Freunde er oder sie habe, schrieb die Person "I like you a lot". Was tun???
- Daten müssen umgeformt werden: Für jede der beiden Gruppen seiner Studie hat Joachim einen Google-Forms-Fragebogen aufgesetzt. Jetzt hat er zwei Tabellen, die er "verheiraten" möchte. Geht das?
- Neue Spalten berechnen: Ein Student fragt nach der Anzahl der richtigen Aufgaben in der Statistik-Probeklausur. Wir wollen helfen und im entsprechenden Datensatz eine Spalte erzeugen, in der pro Person die Anzahl der richtig beantworteten Fragen steht.

0.15.1 Überblick

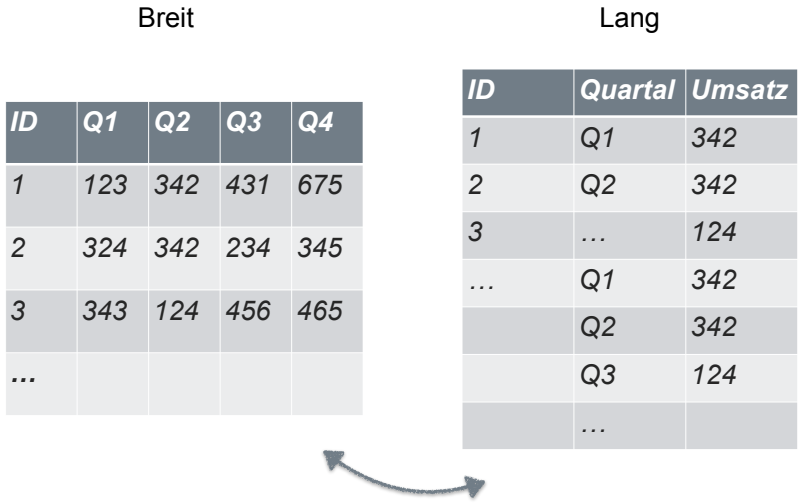
0.15.2 Normalform einer Tabelle

Tabellen in R werden als `data frames` (oder moderner: als `tibble`, kurz für "Table-df") bezeichnet. Tabellen sollten in "Normalform" vorliegen, bevor wir weitere Analysen starten. Unter Normalform verstehen sich folgende Punkte:

- Es handelt sich um einen data frame, also Spalten mit Namen und gleicher Länge; eine Datentabelle in rechteckiger Form
- In jeder Zeile steht eine Beobachtung, in jeder Spalte eine Variable
- Fehlende Werte sollten sich in *leeren* Tabellen niederschlagen
- Daten sollten nicht mit Farkbmarkierungen o.ä. kodiert werden
- keine Leerzeilen, keine Leerspalten
- am besten keine Sonderzeichen verwenden und keine Leerzeichen in Variablennamen und -werten, am besten nur Ziffern und Buchstaben und Unterstriche
- Variablennamen dürfen nicht mit einer Zahl beginnen

Der Punkt "Jede Zeile eine Beobachtung, jede Spalte eine Variable" verdient besondere Beachtung. Betrachen Sie dieses Beispiel:

```
knitr::include_graphics("./images/breit_lang.pdf")
```



ID	Q1	Q2	Q3	Q4
1	123	342	431	675
2	324	342	234	345
3	343	124	456	465
...				

ID	Quartal	Umsatz
1	Q1	342
2	Q2	342
3	...	124
...	Q1	342
	Q2	342
	Q3	124
	...	

In der rechten Tabelle sind die Variablen `Quartal` und `Umsatz` klar getrennt; jede hat ihre eigene Spalte. In der linken Tabelle hingegen sind die beiden Variablen vermischt. Sie haben nicht mehr ihre eigene Spalte, sondern sind über vier Spalten verteilt. Die rechte Tabelle ist ein Beispiel für eine Tabelle in Normalform, die linke nicht.

Eine der ersten Aktionen einer Datenanalyse sollte also die "Normalisierung" Ihrer Tabelle sein. In R bietet sich dazu das Paket `tidyr` an, mit dem die Tabelle von Breit- auf Langformat (und wieder zurück) geschoben werden kann.

Ein Beispiel dazu:

```
meindf <- read.csv("http://stanford.edu/~ejdemyr/r-tutorials/data/unicef-u5mr.csv")

df_lang <- gather(meindf, year, u5mr, U5MR.1950:U5MR.2015)

df_lang <- separate(df_lang, year, into = c("U5MR", "year"), sep = ".")
```

- Die erste Zeile liest die Daten aus einer CSV-Datei ein; praktischerweise direkt von einer Webseite.
- Die zweite Zeile formt die Daten von breit nach lang um. Die neuen Spalten, nach der Umformung heißen dann `year` und `u5mr` (Sterblichkeit bei Kindern unter fünf Jahren). In die Umformung werden die Spalten `U5MR 1950` bis `U5MR 2015` einbezogen.
- Die dritte Zeile "entzerrt" die Werte der Spalte `year`; hier stehen die ehemaligen Spaltenköpfe. Man nennt sie auch `key` Spalte daher. Steht in einer Zelle von `year` bspw. `U5MR 1950`, so wird `U5MR` in eine Spalte mit Namen `U5MR` und `1950` in eine Spalte mit Namen `year` geschrieben.

0.15.3 Daten aufbereiten mit `dplyr`

Es gibt viele Möglichkeiten, Daten mit R aufzubereiten; `dplyr` ist ein populäres Paket dafür. Die Philosophie dabei ist, dass es ein paar wenige Grundbausteine geben sollte, die sich gut kombinieren lassen. Sprich: Wenige grundlegende Funktionen mit eng umgrenzter Funktionalität. Der Autor, Hadley Wickham, sprach einmal in einem Forum (citation needed), dass diese Befehle wenig können, das Wenige aber gut. Ein Nachteil dieser Konzeption kann sein, dass man recht viele dieser Bausteine kombinieren muss, um zum gewünschten Ergebnis zu kommen. Außerdem muss man die Logik des Baukastens gut verstanden habe - die Lernkurve ist also erstmal steiler. Dafür ist man dann nicht darauf angewiesen, dass es irgendwo "Mrs Right" gibt, die genau das kann, so wie ich das will. Außerdem braucht man sich auch nicht viele Funktionen merken. Es reicht einen kleinen Satz an Funktionen zu kennen (die praktischerweise konsistent in Syntax und Methodik sind).

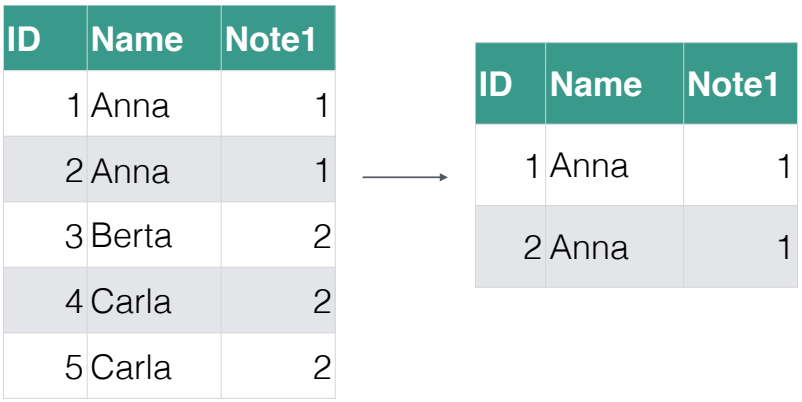
`dplyr` hat seinen Namen, weil es sich ausschließlich um *D*ataframes bemüht; es erwartet einen Dataframe als Eingabe und gibt einen Dataframe zurück[4].

Diese Bausteine sind typische Tätigkeiten im Umgang mit Daten; nichts Überraschendes. Schauen wir uns diese Bausteine näher an.

0.15.3.1 Zeilen filtern mit `filter`

Häufig will man bestimmte Zeilen aus einer Tabelle filtern. Zum Beispiel man arbeitet für die Zigarettenindustrie und ist nur an den Rauchern interessiert (die im Übrigen unser Gesundheitssystem retten (Krämer, 2011)), nicht an Nicht-Rauchern; es sollen die nur Umsatzzahlen des letzten Quartals untersucht werden, nicht die vorherigen Quartale; es sollen nur die Daten aus Labor X (nicht Labor Y) ausgewertet werden etc.

Ein Sinnbild:



Merke: > Die Funktion filter filtert Zeilen aus einem Dataframe.

Schauen wir uns einige Beispiel an.

[4] zumindest bei den meisten Befehlen.

```
data(profiles, package = "okcupiddata")  # Das Paket muss installiert sein
df_frauen <- filter(profiles, sex == "f")  # nur die Frauen
df_alt <- filter(profiles, age > 70)  # nur die alten
df_alte_frauen <- filter(profiles, age > 70, sex == "f")  # nur die alten Frauen
df_nosmoke_nodrinks <- filter(profiles, smokes == "no" | drinks == "not at all")
# liefert alle Personen, die Nicht-Raucher *oder* Nicht-Trinker sind
```

Box:

caution

tip

Visualisierung

- Nutzen (Anscombe)
- Prinzipien nach Tufte
- Cleveland
- ggplot2

Statistisches Modellieren

- Was sind Modelle?
- Überanpassung
- Prädiktion vs. Explanation
- Numerische vs. klassifizierende Modelle
- Geleitete vs. ungeleitete Modelle
- Parametrische vs. nichtparametrische Modelle
- Fehler- vs. Varianzreduktion
- Modellgüte

Numerische Modelle

- Lineare Regression
  - Grundlagen
  - Multiple Regression
  - Interaktion
  - Eisberge
- Logistische Regression
- Penalisierende Regression
- Baumbasierte Verfahre
- Ausblick

Klassifizierende Modelle

- Clusteranalyse
- Nächste-Nachbarn-Analyse

Literaturverzeichnis

Bibliography

(2006). *The Oxford Dictionary of Statistical Terms*. Oxford University Press.

Briggs, W. M. (2008). *Breaking the Law of Averages: Real-Life Probability and Statistics in Plain English*. Lulu.com.

Briggs, W. M. (2016). *Uncertainty: The Soul of Modeling, Probability & Statistics*. Springer.

Gigerenzer, G. (2004). Mindless statistics. *The Journal of Socio-Economics*, 33(5):587–606.

Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., and Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLOS Biology*, 13(3):e1002106.

Krämer, W. (2011). *Wie wir uns von falschen Theorien täuschen lassen*. Berlin University Press.

Romeijn, J.-W. (2016). Philosophy of statistics. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Winter 2016 edition.

Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems ofp values. *Psychonomic Bulletin & Review*, 14(5):779–804.

Wicherts, J. M., Veldkamp, C. L. S., Augusteijn, H. E. M., Bakker, M., van Aert, R. C. M., and van Assen, M. A. L. M. (2016). Degrees of freedom in planning, running, analyzing, and reporting psychological studies: A checklist to avoid p-hacking. *Frontiers in Psychology*, 7.

Wickham, H. and Grolemund, G. (2016). *R for Data Science: Visualize, Model, Transform, Tidy, and Import Data*. O'Reilly Media.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.