

Inferenzstatistik mit Simulationstechniken

Kurzfassung

Sebastian Sauer

Agenda

1. Wozu Simulationstechniken?
2. Der Biertest -- Konfidenzintervall
3. Der Pringels-Test -- Hypothesen testen, Teil 1
4. Die Lächelstudie -- Hypothesen testen, Teil 2
5. Fazit

1 Wozu Simulationstechniken?

Simu ... was?

Simulationstechniken in der Datenanalyse nutzen die Stichprobendaten, um inferenzstatistische Schlüsse zu ziehen.

Didaktiver Nutzen:

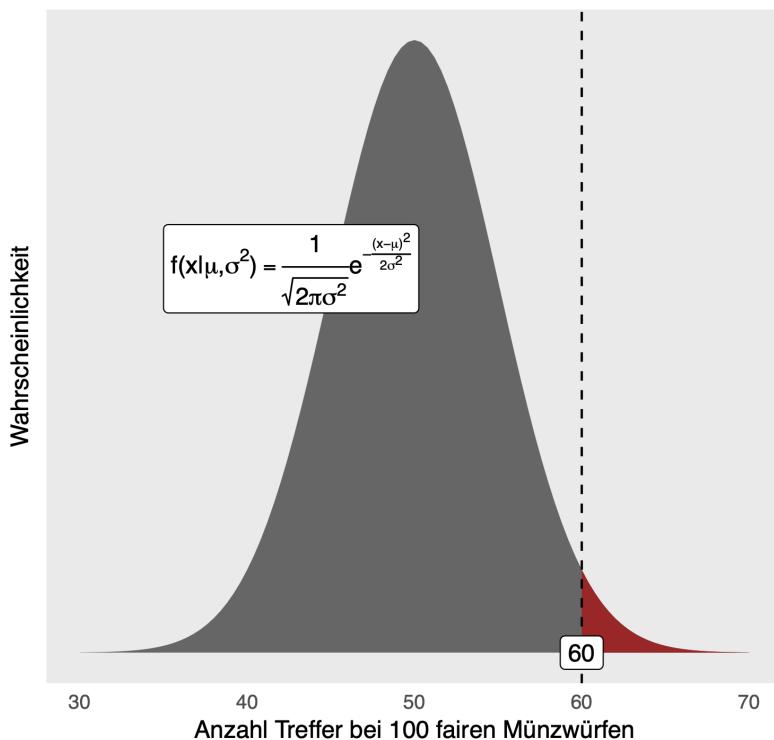
- A) weniger abstrakt
- B) EIN Verfahren für (fast) alle Situationen

Inhaltlicher Nutzen

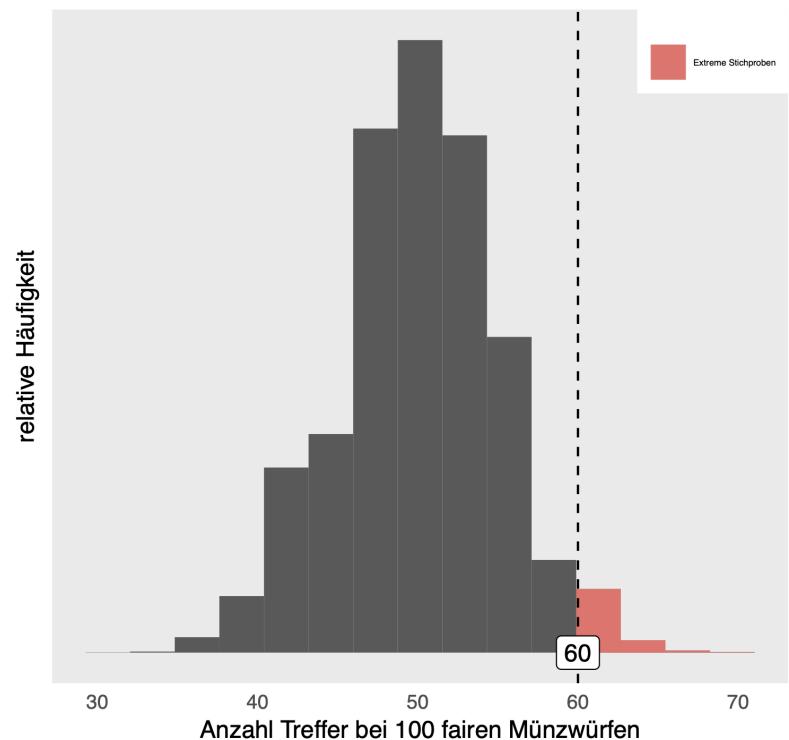
- A) Manchmal präziser
- B) Manchmal gibt es keine andere Möglichkeit

Verteilungsbasiert vs. simulationsbasiert

Berechne das Integral der Fläche unter der Kurve

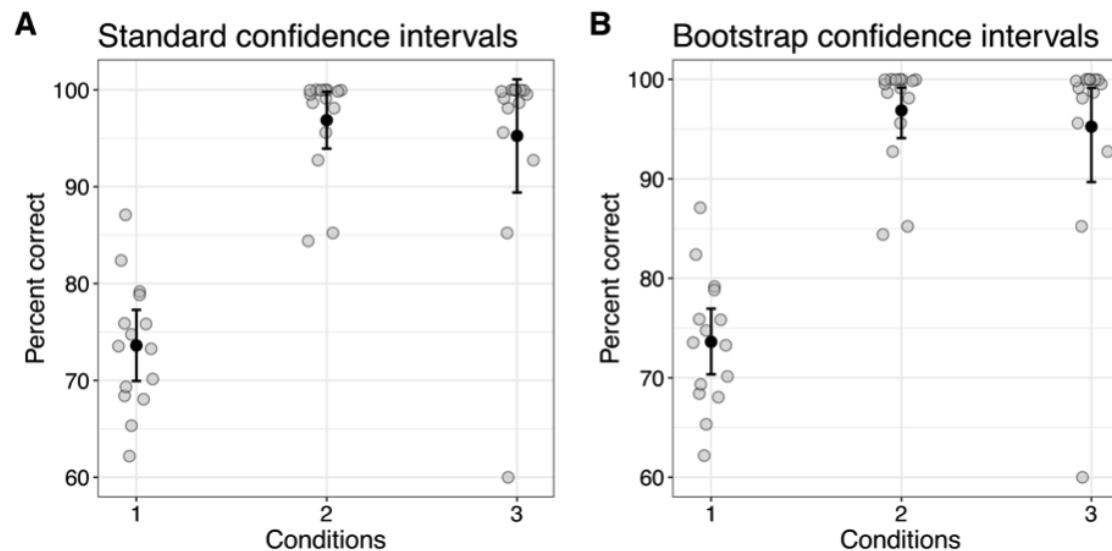


Führe das Experiment häufig aus; zähle die Treffer



Verteilungsannahmen

... können falsch sein



Rousselet, G. A., Pernet, C. R., & Wilcox, R. R. (2019). A practical introduction to the bootstrap: A versatile method to make inferences by using data-driven simulations [Preprint]. PsyArXiv. <https://doi.org/10.31234/osf.io/h8ft7>

2 Der Biertest -- Konfidenzintervall

Live-Experiment: Biersorte erschmecken

Erschmecke das "Biligbier"



Du hast die Wahl



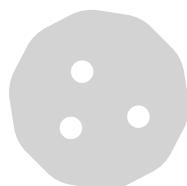
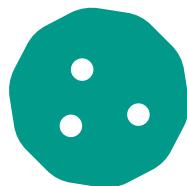
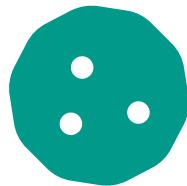
Im Rahmen des ~~normalen Unterrichts~~ einer Sonderveranstaltung der FOM wurde der Versuch durchgeführt:

- $n = 34$ Versuchspersonen*
- $x = 12$ Treffer

* Selbstloser Einsatz für die Wissenschaft

OK, kein Bier, nur Pringels

Ablauf: Schmeck den Pringel heraus!



1. Je zwei Personen, *A* und *B* finden sich in Pärchen
2. *A* wählt 1 Pringel-Chip und 2 Noname-Chips
3. *A* reicht *B* nacheinander die 3 Chips in zufälliger Reihenfolge, *B* hat dabei die Augen geschlossen
4. *B* entscheidet sich, welcher Chips vermutlich der Pringle-Chip ist
5. Das Ergebnis (Treffer ja/nein) kann hier eingetragen werden:
<https://forms.gle/w1bUMGvdDofadih68>
6. Bitte notieren Sie das Ergebnis (Treffer ja/nein) auch auf einen Zettel (den zusammenfalten)

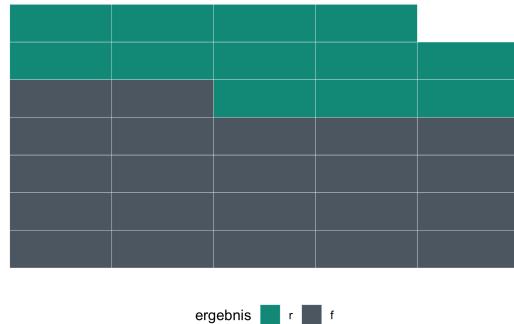
Barcode zum Link:



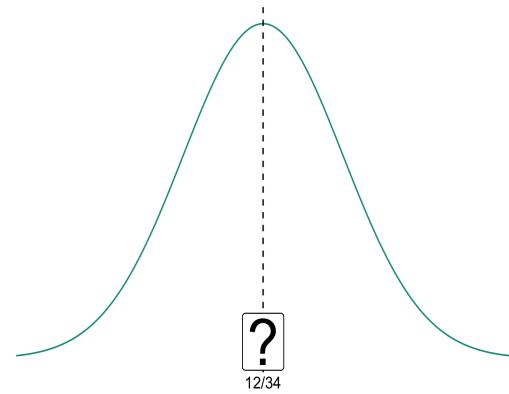
<https://forms.gle/w1bUMGvdDofadih68>

Was ist der wahre Erschmecker-Anteil?

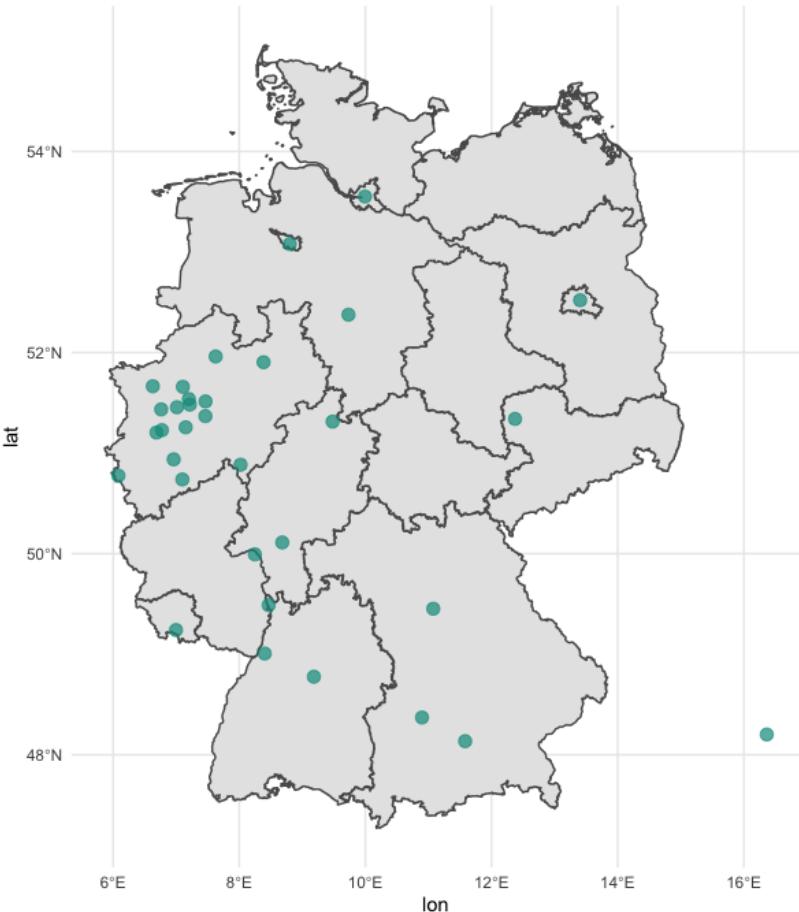
Anteil Stichprobe: 12/34



Anteil in Population der FOM-Professoren???



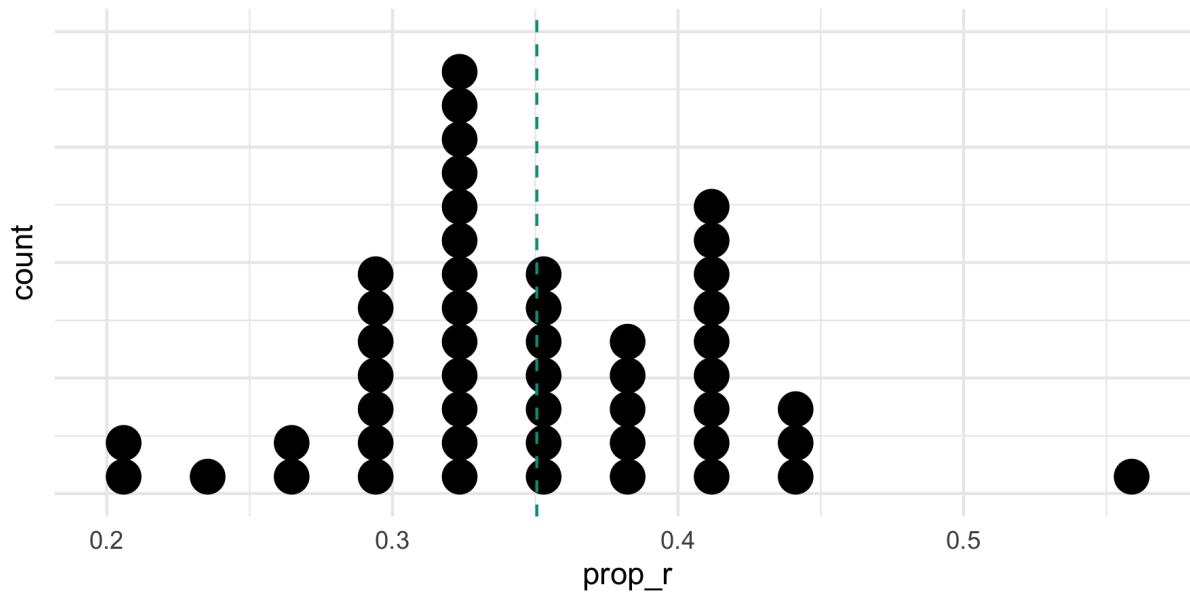
Idee 1: Wir testen alle FOM-Pros!



super. Dauert aber.

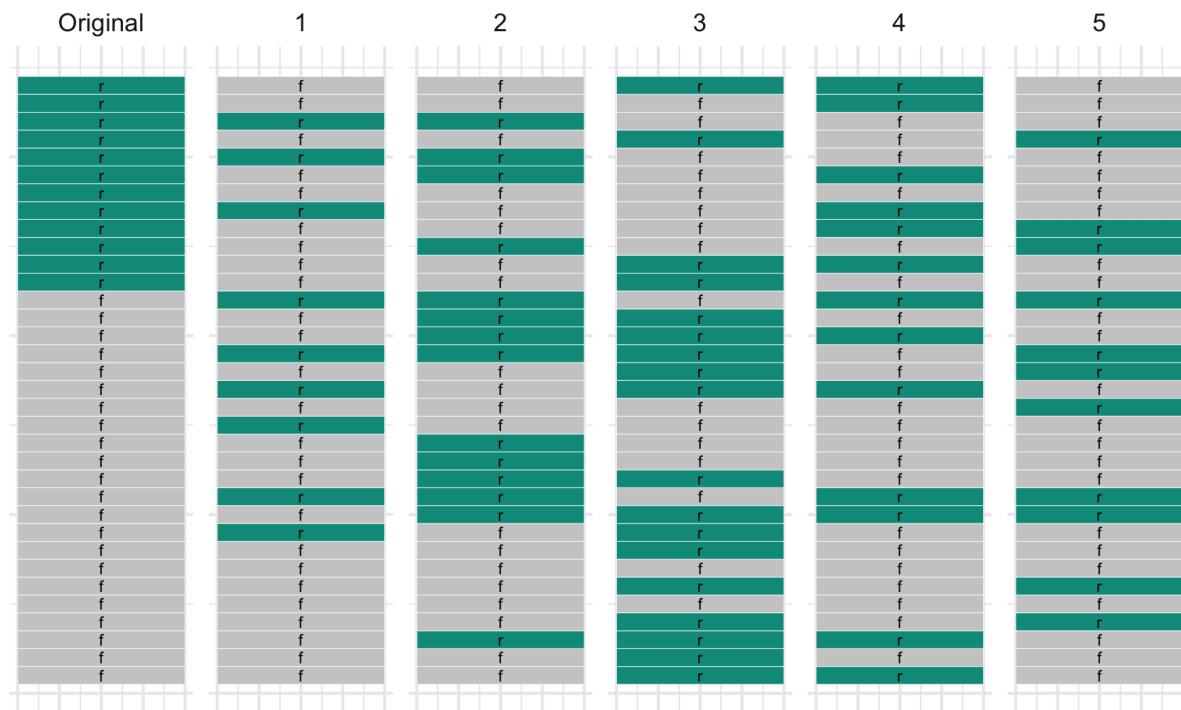
Idee 2: Wir wiederholen der Versuch oft!

z.B. 100 Mal:



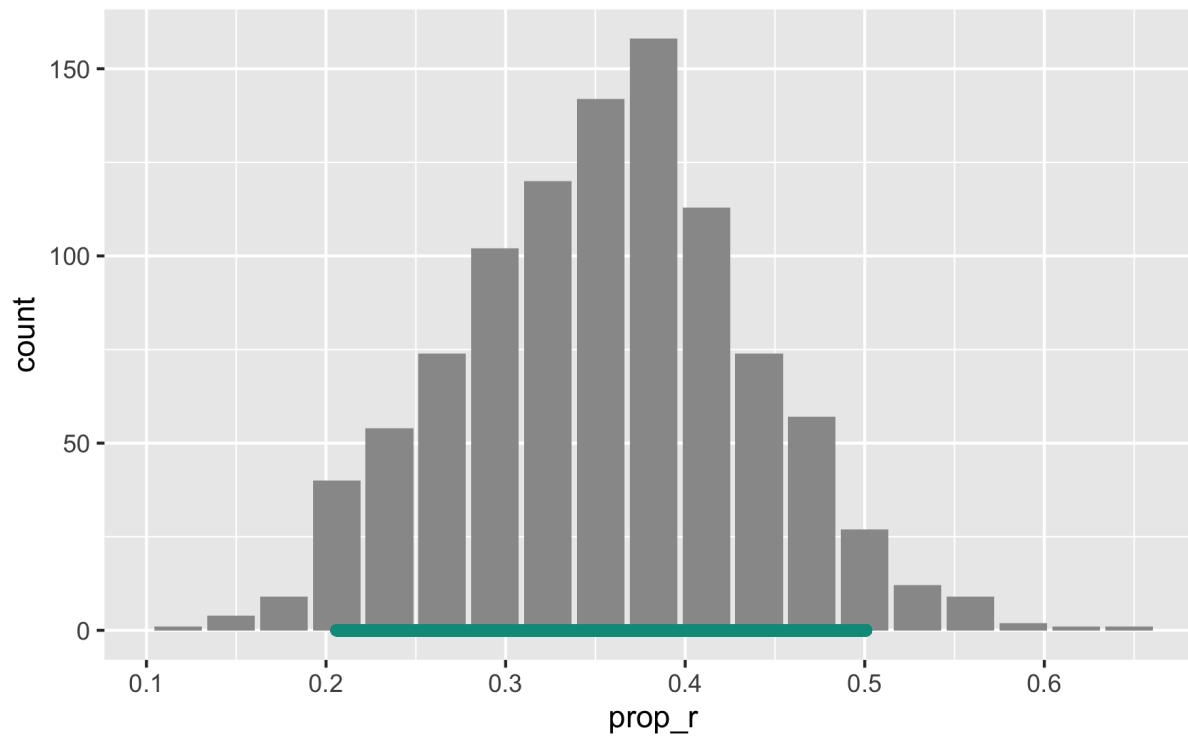
super. Dauert aber.

Idee 3: Der Münchhausen-Trick



Ziele viele Stichproben mit Zurücklegen

Voila -- Die Bootstrap-Verteilung:



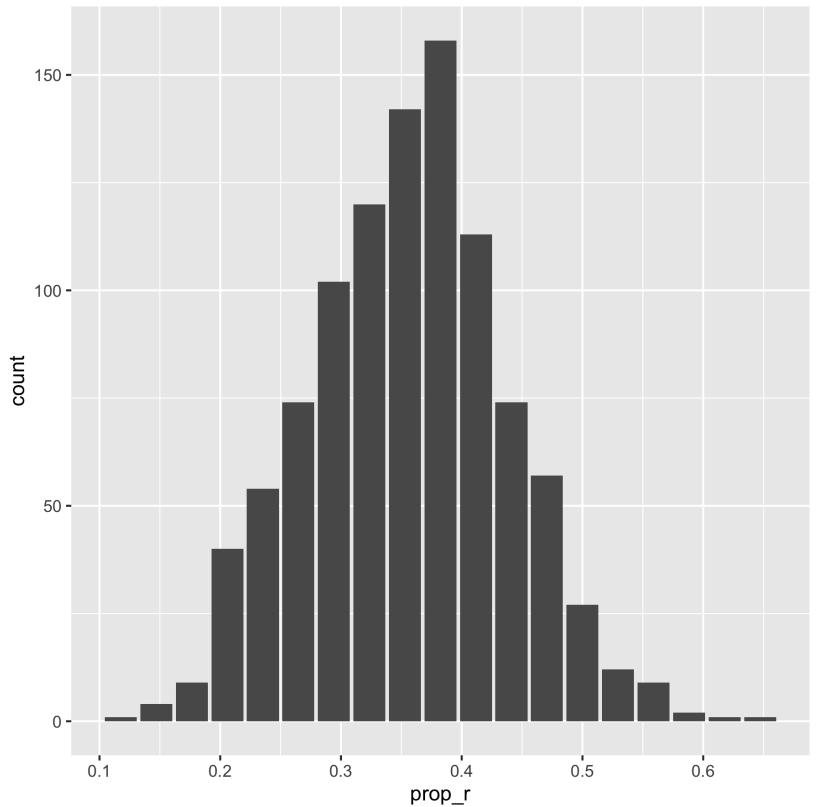
Auf Errisch geht das so...

```
library(mosaic)

# unsere Stichprobe:
stipro <- rep(factor(c("f","r")),
               c(22, 12))

# 3 Bootstrap-Stichproben:
boot1 <- mosaic:::do(3) *
  prop(~ resample(stipro),
       success = "r")
boot1
#>      prop_r
#> 1 0.3529412
#> 2 0.3823529
#> 3 0.3529412
```

```
# Histogramm zeichnen:
gf_bar(~ prop_r, data = Bootvtlg)
```



Bootstrap-Kochrezept

Voraussetzungen

- Zufallsstichprobe
- Nicht zu kleine Stichprobe, $n \geq 35$

Ablauf

1. Ziehe 1000 Bootstrap-Stichproben
2. Berechne jeweils Statistik (z.B. Anteil)
3. Sortiere die Stichproben nach ihrem Wert
4. Zeichne Histogramm
5. Schneide links/rechts jeweils 25 Stichproben ab

Übung: Ist 1/3 ein plausibler Wert in der FOM-Prof-Population?



```
boot2 <- mosaic:::do(1000) * prop(~ resample(stipro), success = "r")
confint(boot2)
#>      name    lower     upper level   method estimate
#> 1 prop_r 0.2058824 0.5294118  0.95 percentile 0.3529412
```

3 Der Pringels-Test -- Hypothesen testen, Teil 1

Echte Pringels kann man nicht rausschmecken (?)



Wie groß ist die Wahrscheinlichkeit π , einen Pringel unter drei Proben rein zufällig, also durch Raten, herauszuschmecken?

- A. $\pi = 0$
- B. $\pi = 1/3$
- C. $\pi = 1/2$
- D. $\pi = 2/3$
- B. $\pi = 1$

Angenommen, Pringels schmecken genau wie NoName-Chips



dann: Trefferchance = 1/3

Stellen wir den Versuch anhand von Münzwürfen nach

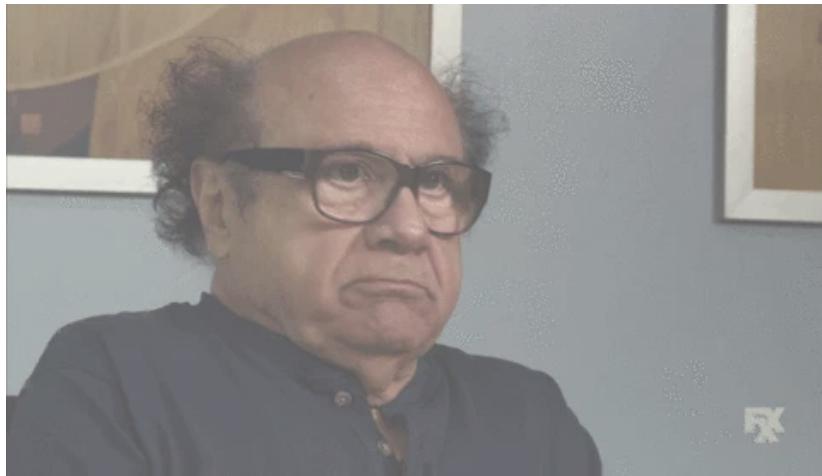
Werfen wir *eine* gezinkte Münze (Trefferquote 1/3):

```
rflip(prob = 1/3)
#>
#> Flipping 1 coin [ Prob(Heads) = 0.333333333333333 ] ...
#>
#> T
#>
#> Number of Heads: 0 [Proportion Heads: 0]
```

Jetzt $n = 34$ gezinkte Münzen:

```
rflip(n = 34, prob = 1/3)
#>
#> Flipping 34 coins [ Prob(Heads) = 0.333333333333333 ] ...
#>
#> T T T T T T T H T H T H H T T H H T H H T T H T T T T T T T T
#>
#> Number of Heads: 10 [Proportion Heads: 0.294117647058824]
```

"Pringels kann man nicht raußschmecken!"



$H_0 : \pi \leq 1/3.$

$H_A : \pi > 1/3.$

via GIPHY

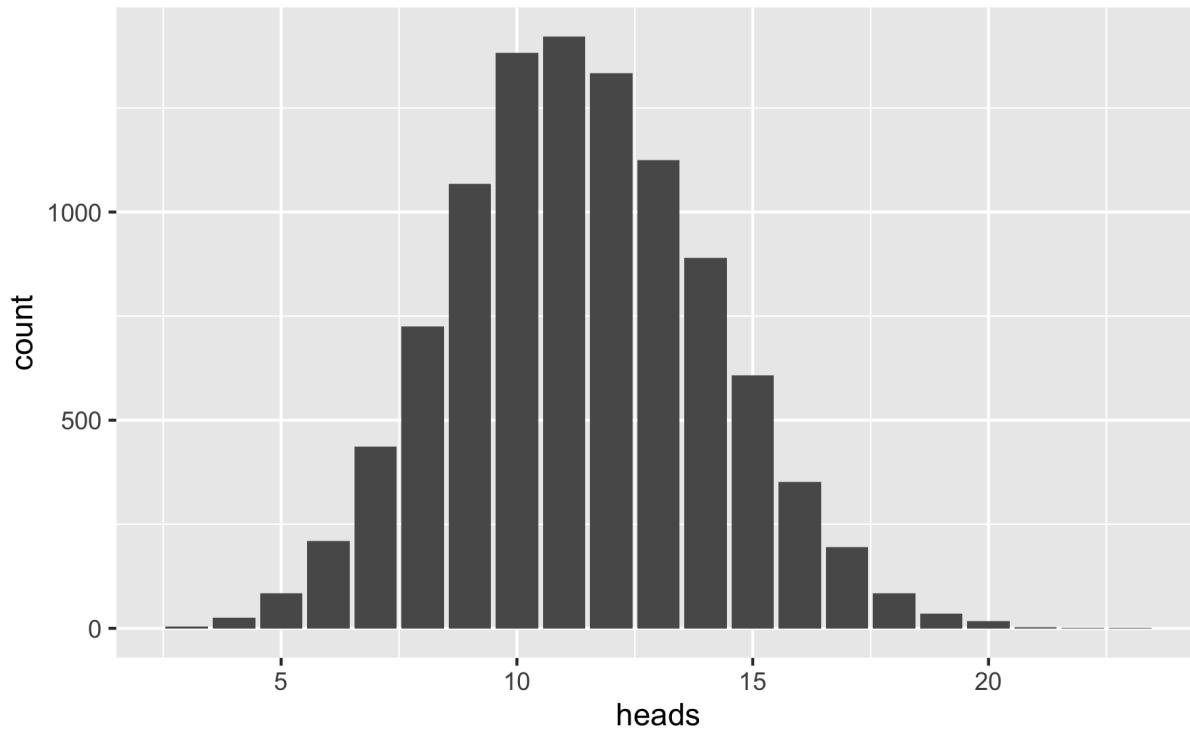
Wir simulieren den Versuch oft ein paar Mal

```
rflip(n = 34, prob = 1/3)
#>
#> Flipping 34 coins [ Prob(Heads) = 0.333333333333333 ] ...
#>
#> T H T H T T H T T T T H T T T T T H H H T H H T T H T H T T H T
#>
#> Number of Heads: 13 [Proportion Heads: 0.382352941176471]
rflip(n = 34, prob = 1/3)
#>
#> Flipping 34 coins [ Prob(Heads) = 0.333333333333333 ] ...
#>
#> T H H T T T T T H H T H H T H T T T T T H T T T T H T H T H H
#>
#> Number of Heads: 13 [Proportion Heads: 0.382352941176471]
```

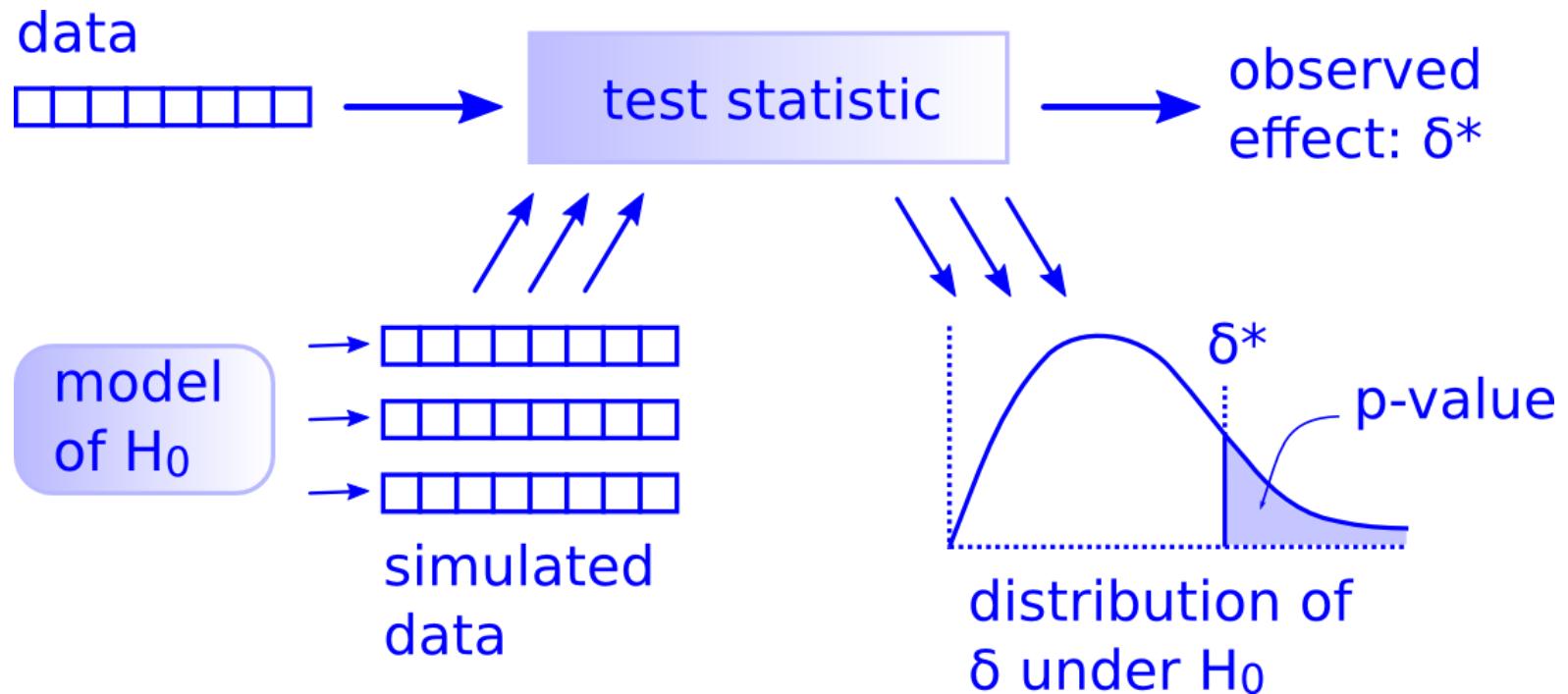
...

Wir erzeugen 1000 Rate-Stichproben

```
Nullvtlg <- mosaic::do(10000) * rflip(n = 34, prob = 1/3)
gf_bar( ~ heads, data = Nullvtlg )
```



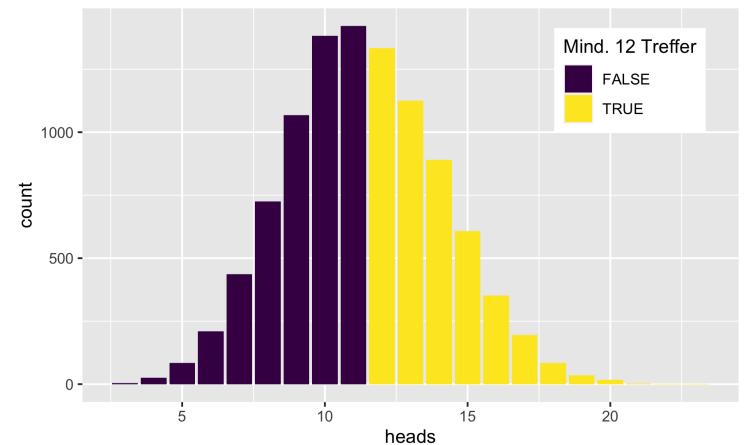
Die Blaupause eines (jeden) statistischen Tests



p-Wert

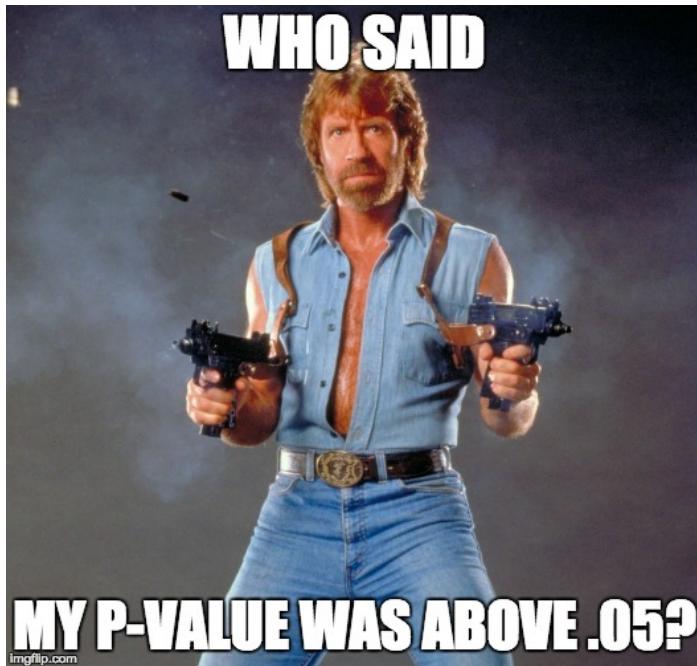
p-Wert: Anteil der Stichproben mit mindestens 12 Treffer:

```
gf_bar( ~ heads,
       data = Nullvtlg)
```



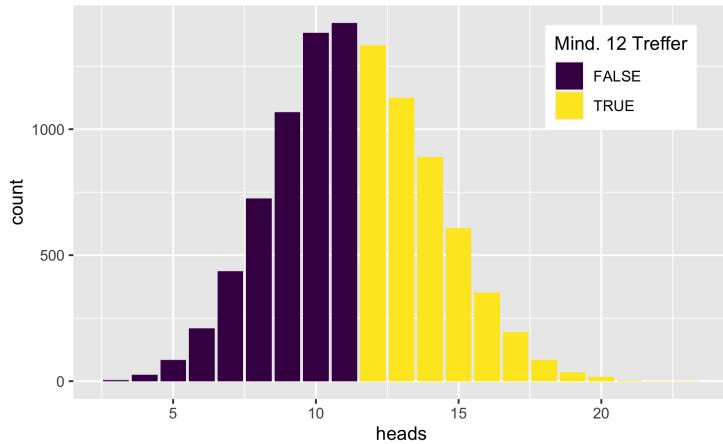
```
prop( ~ heads >= 12,
      data = Nullvtlg)
#> prop_TRUE
#> 0.4644
```

Als sehr, sehr wichtig



... wird der p-Wert von vielen erachtet.

Die Daten sind plausibel unter der Nullhypothese.



Die Daten sind mit der H_0 kompatibel.

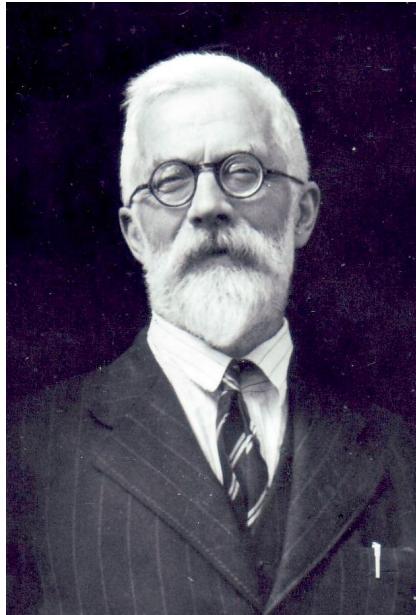
Wir können die Nullhypothese also nicht ablehnen.

p-Wert: Eine gute Geschichte

... verdient, aufgebauscht zu werden (?)

<u>P-VALUE</u>	<u>INTERPRETATION</u>
0.001	
0.01	
0.02	HIGHLY SIGNIFICANT
0.03	
0.04	
0.049	SIGNIFICANT
0.050	OH CRAP. REDO CALCULATIONS.
0.051	ON THE EDGE OF SIGNIFICANCE
0.06	
0.07	HIGHLY SUGGESTIVE,
0.08	SIGNIFICANT AT THE $P < 0.10$ LEVEL
0.09	
0.099	HEY, LOOK AT THIS INTERESTING SUBGROUP ANALYSIS
≥ 0.1	

Forschung über dem 5%-Niveau



"Mein p-Wert ist der ~~größte~~ kleinste."

Bildquelle

"... some statisticians prefer to supplement or even replace p-values with other approaches. These include methods that emphasize estimation over testing, such as confidence ... intervals ..."

"Good statistical practice ... emphasizes principles of good study design ... , a variety of numerical and graphical summaries of data, understanding of the phenomenon under study, interpretation of results in context, complete reporting and proper logical and quantitative understanding of what data summaries mean."

... No single index should substitute for scientific reasoning.

Ronald L. Wasserstein & Nicole A. Lazar (2016) The ASA Statement on p-Values: Context, Process, and Purpose, *The American Statistician*, 70:2, 129-133, DOI: 10.1080/00031305.2016.1154108

... 0.051 fast genauso sehr wie 0.049

Im Übrigen ...



God

@TheTweetOfGod



I have no interest in you or your life.

33K 5:45 AM - Jan 4, 2019



7,182 people are talking about this



Gelman, A., & Stern, H. (2006). The Difference Between “Significant” and “Not Significant” is not Itself Statistically Significant. *The American Statistician*, 60(4), 328–331. <https://doi.org/10.1198/000313006X152649>

4 Die Lächelstudie -- Hypothesen testen, Teil 2

Stimmt Lächeln nachsichtiger?

Stimmt Lächeln nachsichtiger?



Bildquelle

Skeptiker:

"Lächeln bringt doch nichts!"

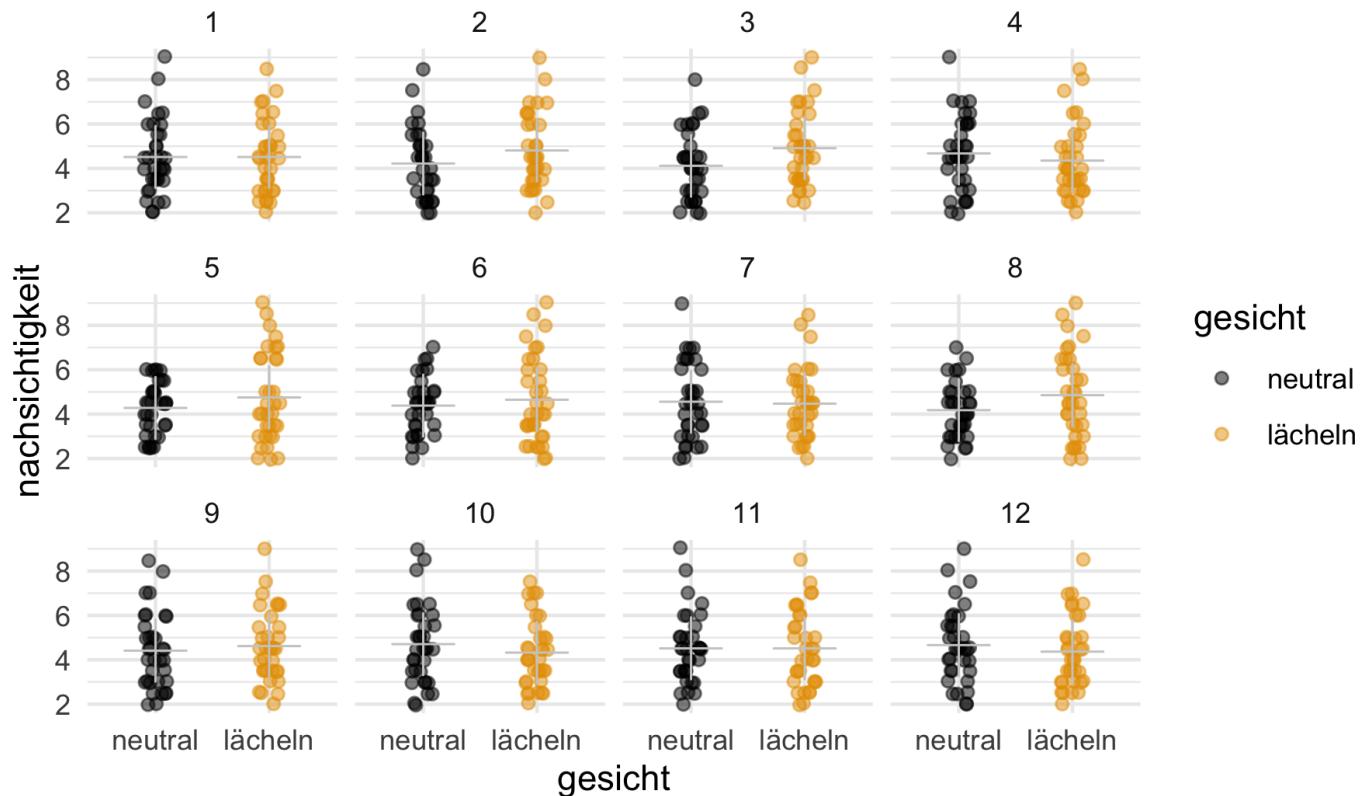
$$H_0 : \mu_{\text{Lächeln}} \leq \mu_{\text{Neutral}}$$

$$H_A : \mu_{\text{Lächeln}} > \mu_{\text{Neutral}}$$

LaFrance, M., & Hecht, M. A. (1995). Why smiles generate leniency. *Personality and Social Psychology Bulletin*, 21(3), 207-214, <https://doi.org/10.1177%2F0146167295213002>

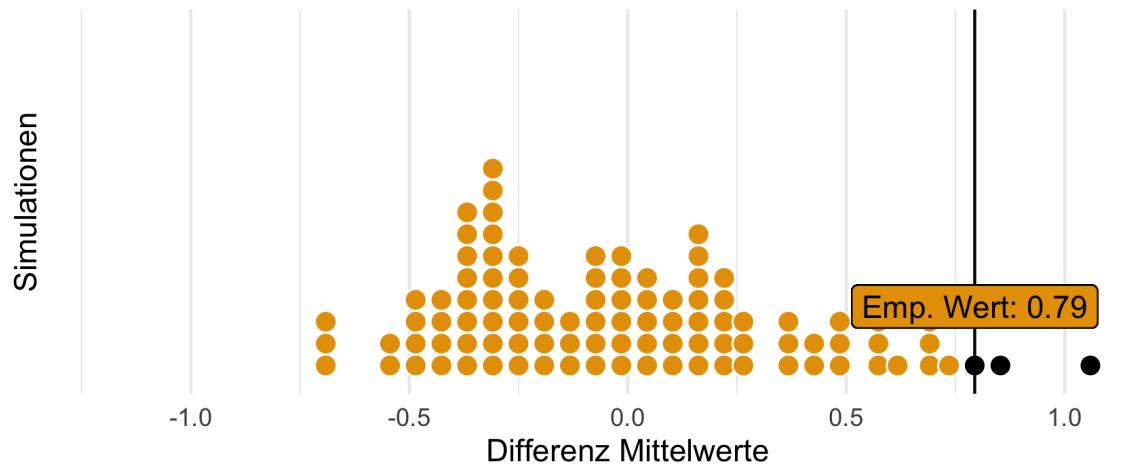
Finde den echten Datensatz

11 Datensätze wurden so simuliert, dass es keinen Unterschied in den Mittelwerten der Populationen gibt, 1 Datensatz ist echt.

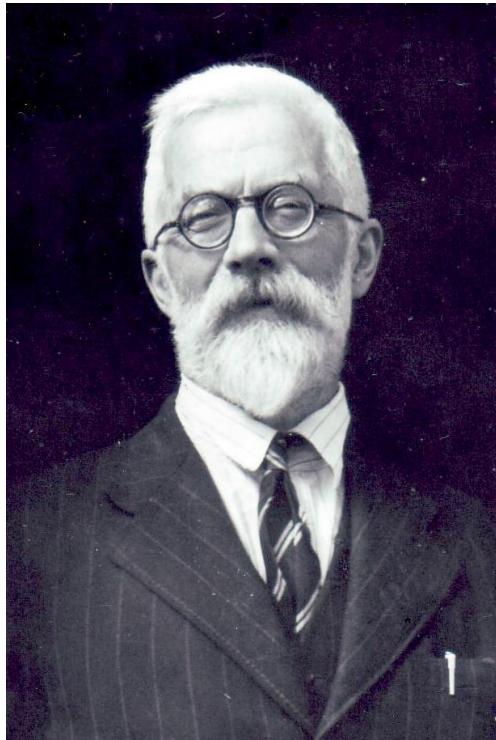


Angenommen, es gäbe keinen Zusammenhang

... von Lächeln und Nachsichtigkeit. Wie häufig wäre unser empirischer Wert in diesen Stichproben?



Der p-Wert schlägt zurück



"Wer die Definition vergisst, dem tätteviere ich *p-value* auf den Unterarm."

Bildquelle

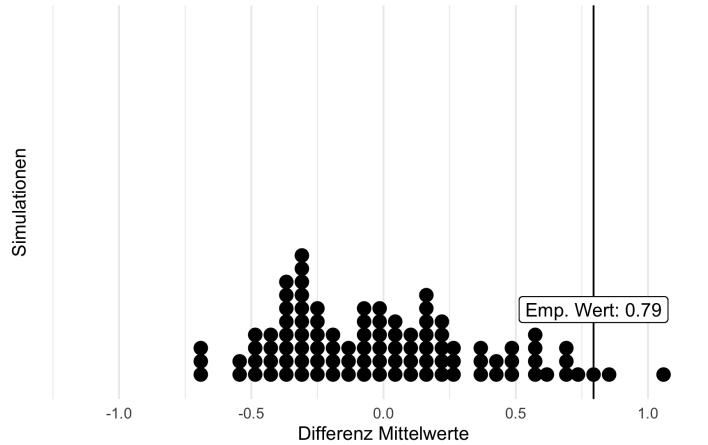
*Wenn die Daten aus einer Population stammen, in der die Nullhypothese stimmt, dann ist unser empirisches Ergebnis *selten* (unplausibel).*

Auf dieser Basis entscheiden wir uns in diesem Fall, die H_0 zu verwerfen (bzw. nicht mehr so stark wie vorher an sie zu glauben).

Wie stellt man die Nullverteilung her?

1.

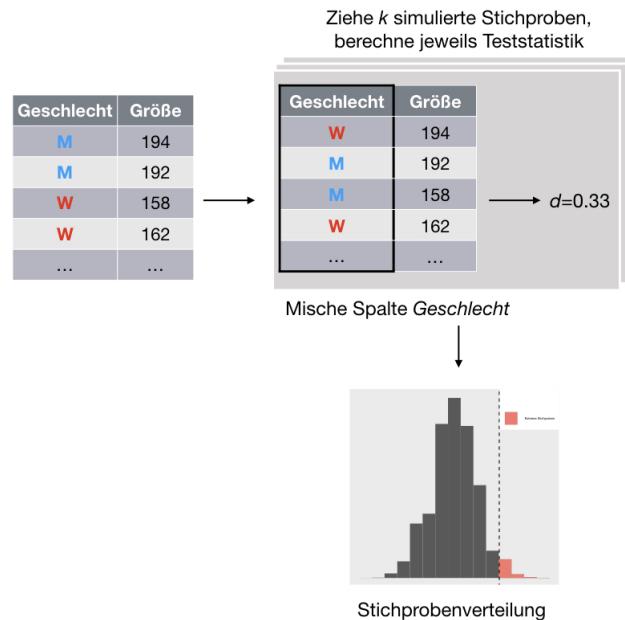
Man reist in ein Land, in dem es keinen Zusammenhang gibt (zwischen Lächeln und Nachsichtigkeit) und zieht dort viele Stichproben.



... Oder man mischt eine Spalte

2.

Man mischt die Spalte (z.B. UV) durch, so dass der Zusammenhang zwischen der Spalte UV und einer anderen Spalte (AV) aufgelöst wird.



So mischt man eine Spalte mit R

```
Nullvtlg_gesicht <- mosaic::do(100)*diffmean(nachsichtigkeit ~ shuffle(gesicht),  
data = Laecheln)
```

gesicht	gesicht_gemischt	nachsichtigkeit
lächeln	neutral	7.0
lächeln	neutral	3.0
lächeln	lächeln	6.0
lächeln	lächeln	4.5
lächeln	neutral	3.5
lächeln	lächeln	4.0

Übung: Konsumieren Raucher im Schnitt mehr? (1/3)

$$H_0 : \mu_R = \mu_{NR}$$

$$H_A : \mu_r \neq \mu_{NR}$$

Daten laden:

```
download.file("https://goo.gl/whKjnl",
              destfile = "tips.csv")
tips <- read.csv2("tips.csv")
```

Nullverteilung berechnen:

```
library(mosaic)
Nullvtlg_Raucher <- mosaic::do(1000) *
  diffmean(total_bill ~ shuffle(smoker),
           data = tips)
```

Die ersten paar Werte aus den Stichproben der Nullverteilung:

diffmean

-1.2006729

1.5180254

2.1377989

0.2732657

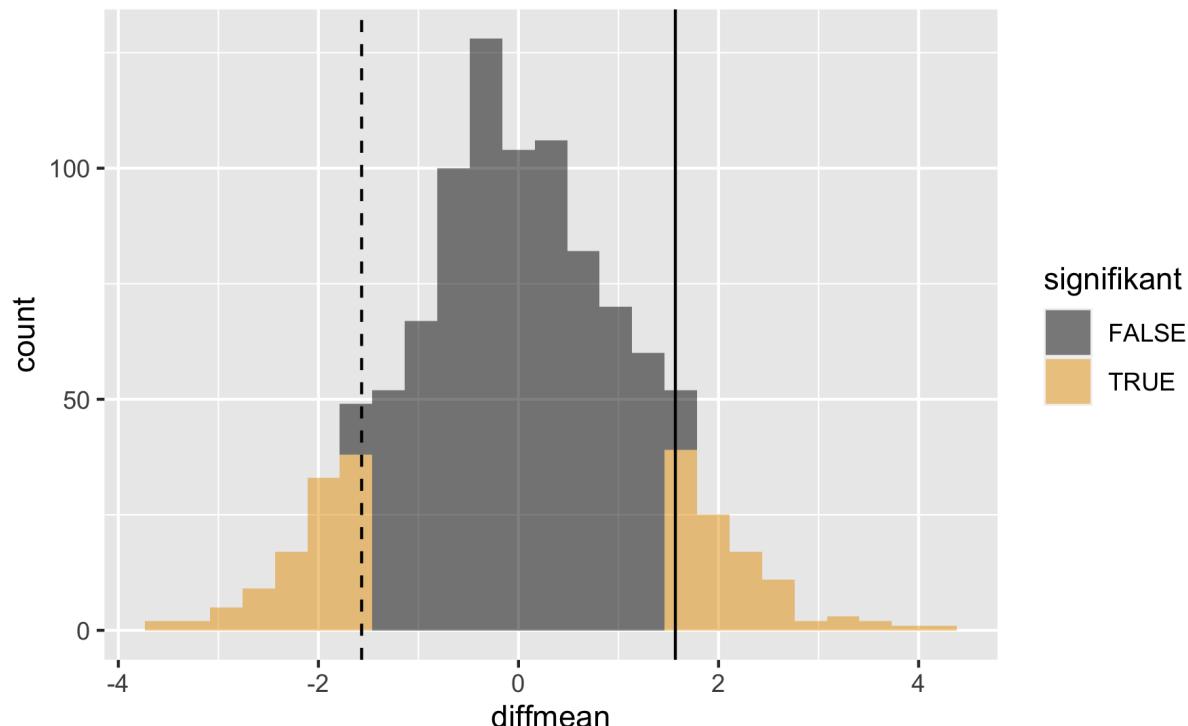
1.6922987

0.0890885

Übung: Konsumieren Raucher im Schnitt mehr? (2/3)

H_0 -Verteilung visualisieren:

```
gf_histogram( ~ diffmean, data = Nullvtlg_Raucher) %>%
  gf_vline(xintercept = ~diffmean(total_bill ~ smoker, data = tips))
```



Übung: Konsumieren Raucher im Schnitt mehr? (2/3)

Empirische Differenz/Wert in der Stichprobe:

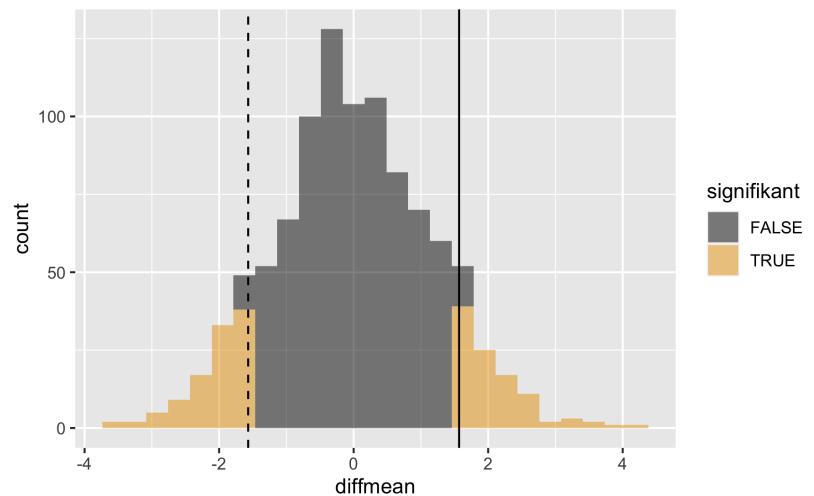
```
#> diffmean
#> 1.568066
```

Anteil der Stichproben, die mind. so groß sind wie der emp. Wert:

```
#> prop_TRUE
#> 0.099
```

Mal zwei nehmen, da ungerichtete Hypothese:

```
#> prop_TRUE
#> 0.198
```

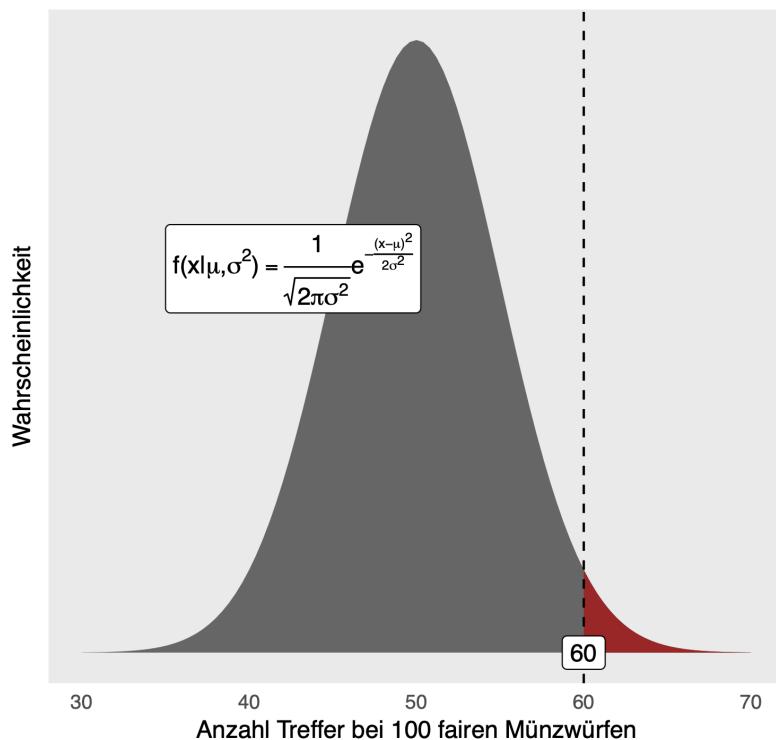


5 Fazit

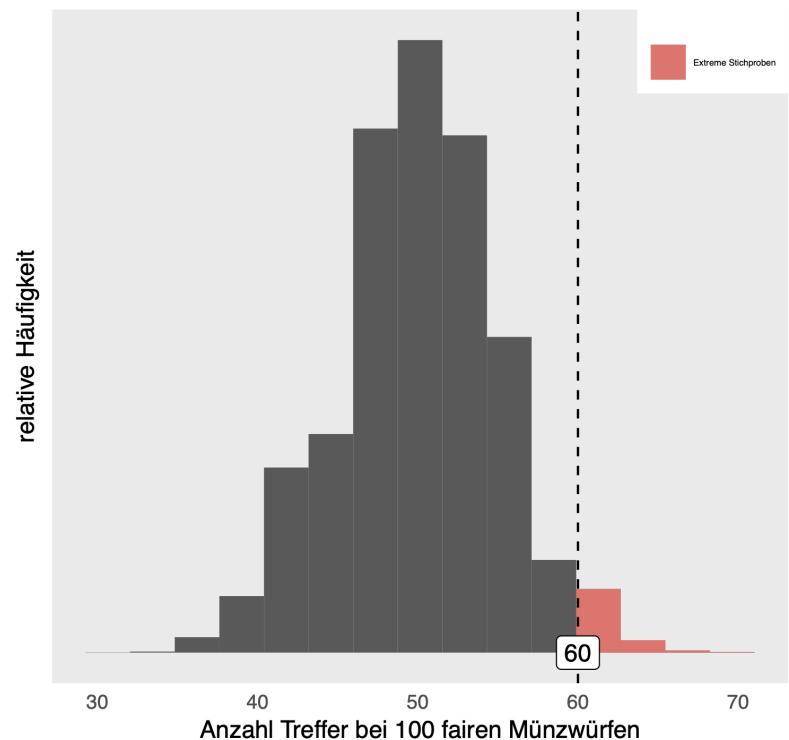
Wie war das nochmal im Mittelteil?

Verteilungsbasiert vs. simulationsbasiert

Berechne das Integral der Fläche unter der Kurve



Führe das Experiment häufig aus; zähle die Treffer



Drei Varianten von Simulationstechniken

1. **Bootstrapping:** Ziehe viele Stichproben mit Zurücklegen aus Originalstichprobe, um Konfidenzintervall zu erhalten
2. **Permutationtest:** Testen von Zusammenhangs-/Unterschiedshypothesen
3. **Einfache Simulation:** Führe den Versuch oft durch, unter Annahme von H_0

Drei Varianten von Simulationstechniken -- mit R

1. Bootstrapping: Ziehe viele Stichproben mit Zurücklegen aus Originalstichprobe, um Konfidenzintervall zu erhalten

```
do(oft) * statistik(y ~ x, data = resample(Daten))
```

2. Permutationtest: Testen von Zusammenhangs-/Unterschiedshypothesen

```
do(oft) * statistik(y ~ shuffle(x), data = Daten)
```

3. Einfache Simulation: Führe den Versuch oft durch, unter Annahme von H_0

```
do(oft) * ziehe_aus_verteilung(n, parameterwerte)
```

Übersicht Teststatistiken (Auswahl)

Y	X	Statistik
binär	NA	Anteil ` p `
kategorial	NA	Verhältnis beobachtet/erwartet: ` χ^2 `
kategorial	NA	Verhältnis beobachtet/erwartet": ` χ^2 `
numerisch	NA	Mittelwert ` \bar{x} `
binär	binär	Differenz der Anteile ` $p_B - p_A$ `
numerisch	binär	Differenz der Mittelwerte ` $\bar{x}_B - \bar{x}_A$ `
kategorial	kategorial	Verhältnis beobachtet/erwartet: ` χ^2 `
numerisch	kategorial	Verhältnis Varianz zwischen/innerhalb der Gruppen: ` F `
numerisch	numerisch	Korrelation oder Regression ` $r\hat{\beta}$ `
kategorial	numerisch	Regression ` $\hat{\beta}$ ` (logistische oder multinomiale Regression)

Übersicht Inferenzverfahren R mosaic (Auswahl)

Y	X	simulationsbasiert	konventionell
binär	.	prop()	binom.test()
kategorial	.	xchisq.test()	xchisq.test()
kategorial	.	chisq.test()	chisq.test()
numerisch	.	mean()	t.test()
binär	binär	diffprop()	prop.test()
numerisch	binär	diffmean()	t.test()
kategorial	kategorial	xchisq.test()	xchisq.test()
numerisch	kategorial	aov()	aov()
numerisch	numeric	cor(), lm()	cor.test(), lm()
kategorial	numerisch	glm(family= 'binomial')	glm(family= 'binomial')

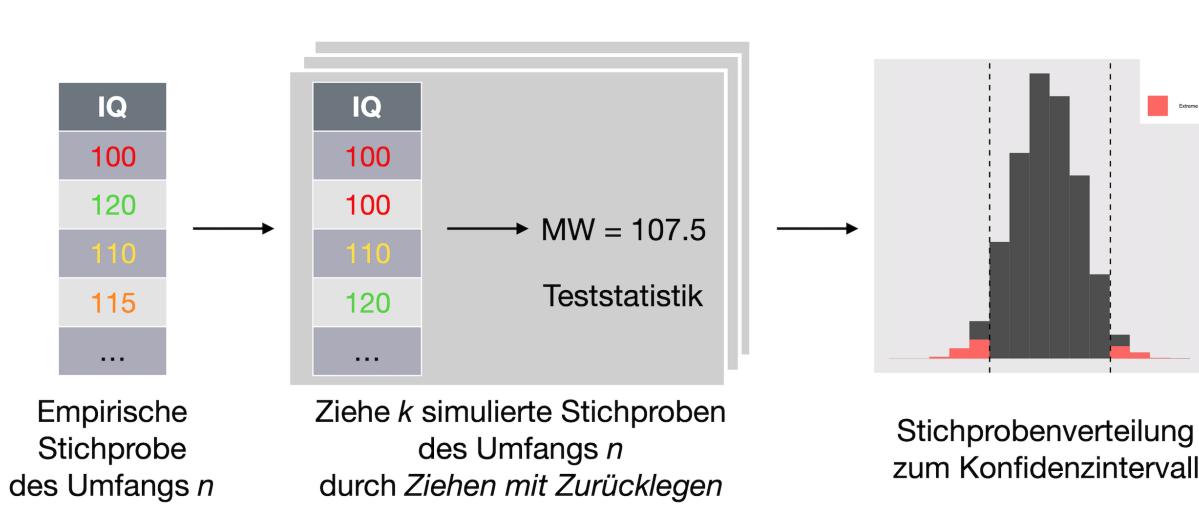


- Ein Prinzip für alle gängigen Verfahren
- Wenige Voraussetzungen
- Einfach, wenig abstrakt

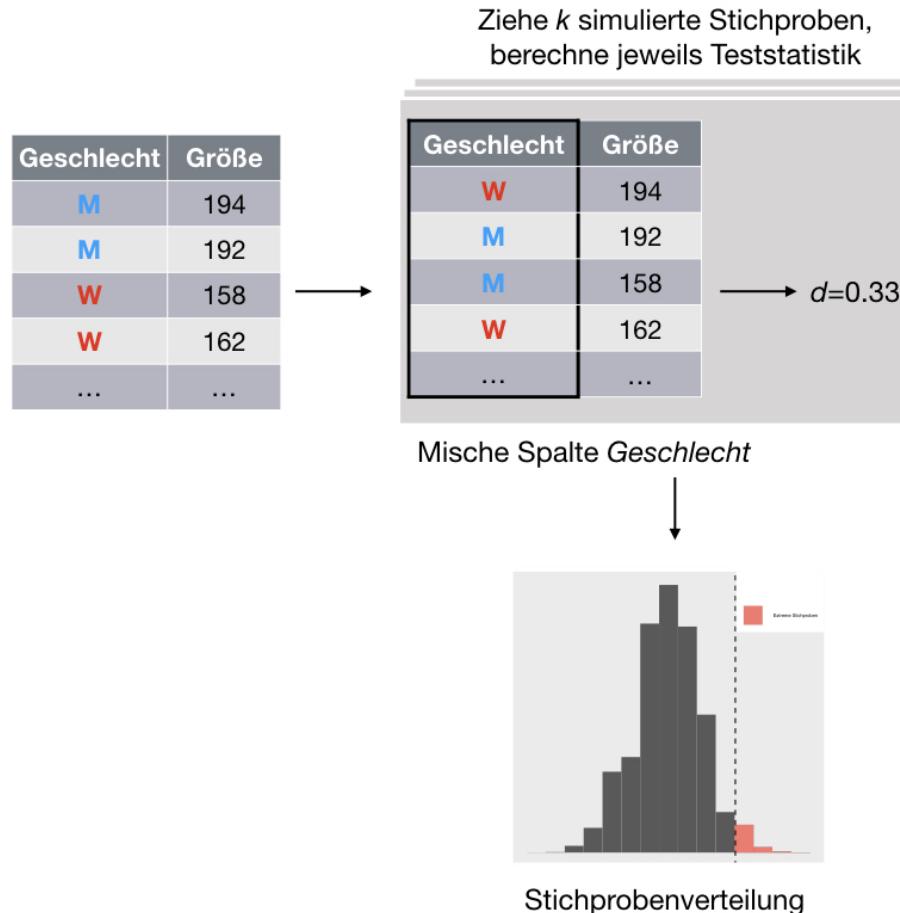


- Zur Anschlussfähigkeit sollten Namen konventioneller Verfahren (wie *t*-Test) weiterhin gelehrt werden

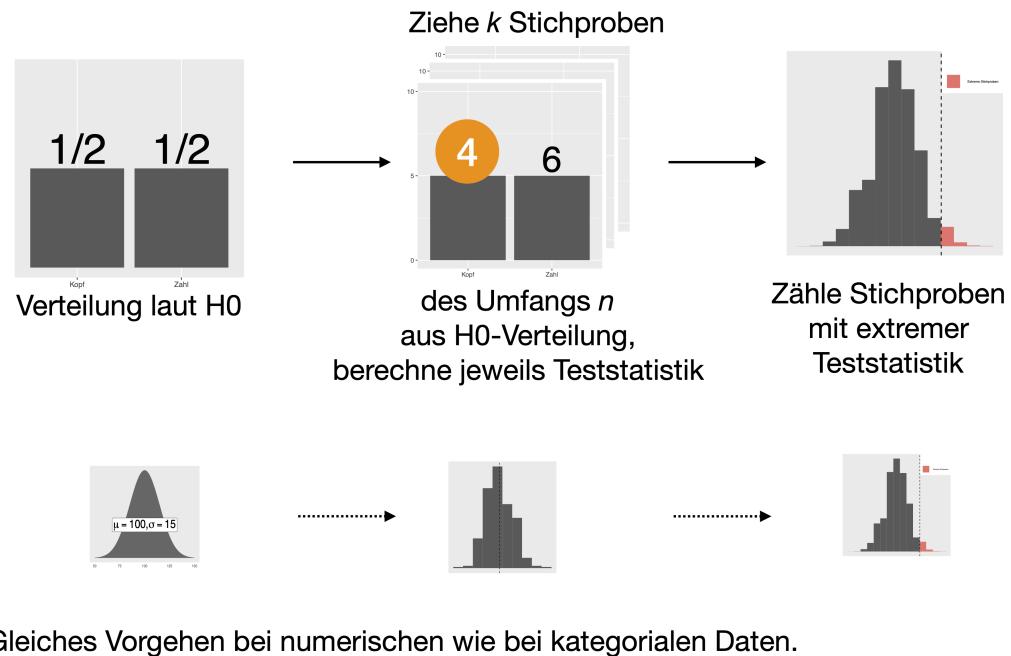
Sinnbild: Bootstrap



Sinnbild: Zusammenhang zweier Variablen (Permutationstest)



Sinnbild: Test auf bestimmten Wert einer Variablen



Fragen, Feedback, Feierlichkeiten?

Sprechen Sie uns an!

✉ sebastian.sauer@fom.de

Diese Folien wurden von Autor*innen der FOM <https://www.fom.de/> entwickelt und stehen unter der Lizenz CC-BY-SA 3.0 de: <https://creativecommons.org/licenses/by-nc-sa/3.0/de/>

Technical details

- Last update: 2020-06-26
- Slides built with `xaringan`, based on `rmarkdown`
- Packages: assertthat 0.2.1, backports 1.1.7, blob 1.2.1, broom 0.5.6, cellranger 1.1.0, cli 2.0.2, codetools 0.2-16, colorspace 1.4-1, cowplot 1.0.0, crayon 1.3.4, crosstalk 1.1.0.1, curl 4.3, DBI 1.1.0, dbplyr 1.4.4, digest 0.6.25, dplyr 1.0.0, DT 0.13, ellipsis 0.3.1, emoji 0.0.0.9000, evaluate 0.14, extrafont 0.17, extrafontdb 1.0, fansi 0.4.1, farver 2.0.3,forcats 0.5.0, fs 1.4.1, generics 0.0.2, ggdendro 0.1-20, ggforce 0.3.1, ggformula 0.9.4, ggplot2 3.3.1, ggrepel 0.8.2, ggstance 0.3.4, ggthemes 4.2.0, glue 1.4.1, gridExtra 2.3, gtable 0.3.0, haven 2.3.0, highr 0.8, hms 0.5.3, htmltools 0.4.0, htmlwidgets 1.5.1, httr 1.4.1, icon 0.1.0, jsonlite 1.6.1, kableExtra 1.1.0, knitr 1.28, labeling 0.3, lattice 0.20-41, lazyeval 0.2.2, leaflet 2.0.3, lifecycle 0.2.0, lubridate 1.7.9, magrittr 1.5, MASS 7.3-51.6, Matrix 1.2-18, modelr 0.1.8, mosaic 1.7.0, mosaicCore 0.6.0, mosaicData 0.18.0, munsell 0.5.0, nlme 3.1-148, pillar 1.4.4, pkgconfig 2.0.3, plyr 1.8.6, polyclip 1.10-0, purrr 0.3.4, R6 2.4.1, RColorBrewer 1.1-2, Rcpp 1.0.4.6, readr 1.3.1, readxl 1.3.1, reprex 0.3.0, rlang 0.4.6, rmarkdown 2.2, rstudioapi 0.11, Rttf2pt1 1.3.8, rvest 0.3.5, scales 1.1.1, sessioninfo 1.1.1, stringi 1.4.6, stringr 1.4.0, tibble 3.0.1, tidyverse 1.3.0, tidyverse 1.3.0, tweenr 1.0.1, vctrs 0.3.0, viridisLite 0.3.0, waffle 1.0.1, webshot 0.5.2, withr 2.2.0, xaringan 0.16, xfun 0.14, xml2 1.3.2, yaml 2.2.1
- For detailed `session_info()` check out this [text file](#).
- Thanks to all the open source developers.