

QM6:

Schätzgenauigkeit:

Punktschätzer, Standardfehler (SE) und Konfidenzintervall

Die Differenz als Schätzfehler

je größer die Stichprobe, desto näher sind die Stichprobenmittelwerte am wahren Populationsmittelwert

```
> favstats(flights_clean$arr_delay) # Population <- normalerweise unbekannt
min  Q1 median Q3  max      mean      sd      n missing
-86 -17      -5 14 1272  6.895377 44.63329 327346      0
```

	min	Q1	median	Q3	max	mean	sd	n	missing	.row		min	Q1	median	Q3	max	mean	sd	n	missing
...1	-41	-13.00	-1.5	28.25	146	22.3	60.26617	10	0	1	...1	-75	-17	-5	14	915	7.04148	44.61456	100000	0
...2	-28	-6.75	16.5	43.75	87	21.5	36.05936	10	0	1	...2	-86	-17	-5	14	1109	6.97302	44.68221	100000	0
...3	-26	-7.75	10.5	33.25	123	19.3	42.52594	10	0	1	...3	-86	-17	-5	14	1272	6.96174	44.85185	100000	0
...4	-36	-20.00	-13.5	6.50	14	-10.5	18.00154	10	0	1	...4	-74	-17	-5	14	1109	6.91156	44.84390	100000	0
...5	-29	-14.25	-5.0	4.75	205	13.3	68.33911	10	0	1	...5	-86	-17	-5	14	1272	6.79285	44.78490	100000	0
...6	-35	-20.50	-8.0	7.00	83	3.3	37.13055	10	0	1	...6	-86	-17	-5	14	1127	7.06829	44.74456	100000	0

Beobachtung:

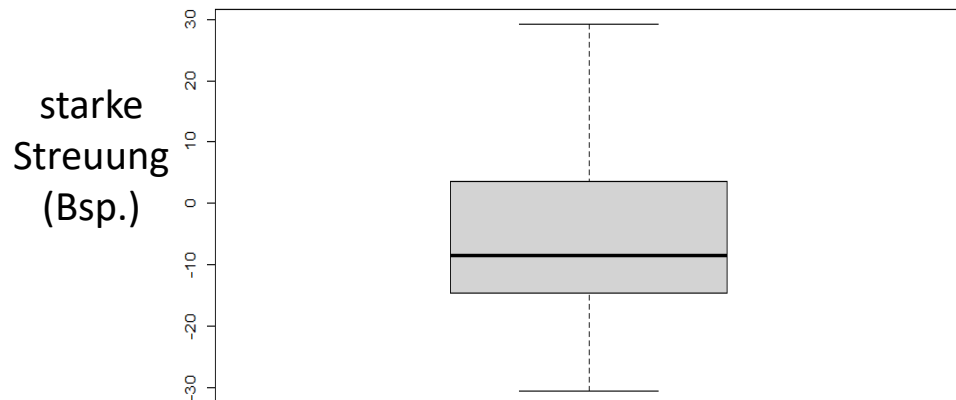
- Kennzahlen ‚in der Mitte‘ unserer Stichprobe sind bei kleineren Stichproben bessere Schätzer (Median, iqr)
- Aggregierende Kennzahlen haben die Tendenz zur Mitte, der MW (mean) ist deshalb der häufigste Schätzer

Die Differenz als Schätzfehler

je größer die Stichprobe, desto näher sind die Stichprobenmittelwerte am wahren Populationsmittelwert

```
> favstats(flights_clean$arr_delay) # Population <- normalerweise unbekannt
min  Q1 median  Q3  max      mean      sd      n missing
-86 -17      -5  14 1272  6.895377 44.63329 327346      0
```

	min	Q1	median	Q3	max	mean	sd	n	missing	.row		min	Q1	median	Q3	max	mean	sd	n	missing
...1	-41	-13.00	-1.5	28.25	146	22.3	60.26617	10	0	1	...1	-75	-17	-5	14	915	7.04148	44.61456	100000	0
...2	-28	-6.75	16.5	43.75	87	21.5	36.05936	10	0	1	...2	-86	-17	-5	14	1109	6.97302	44.68221	100000	0
...3	-26	-7.75	10.5	33.25	123	19.3	42.52594	10	0	1	...3	-86	-17	-5	14	1272	6.96174	44.85185	100000	0
...4	-36	-20.00	-13.5	6.50	14	-10.5	18.00154	10	0	1	...4	-74	-17	-5	14	1109	6.91156	44.84390	100000	0
...5	-29	-14.25	-5.0	4.75	205	13.3	68.33911	10	0	1	...5	-86	-17	-5	14	1272	6.79285	44.78490	100000	0
...6	-35	-20.50	-8.0	7.00	83	3.3	37.13055	10	0	1	...6	-86	-17	-5	14	1127	7.06829	44.74456	100000	0



uns interessiert das
95%- Intervall der
Schwankung des means

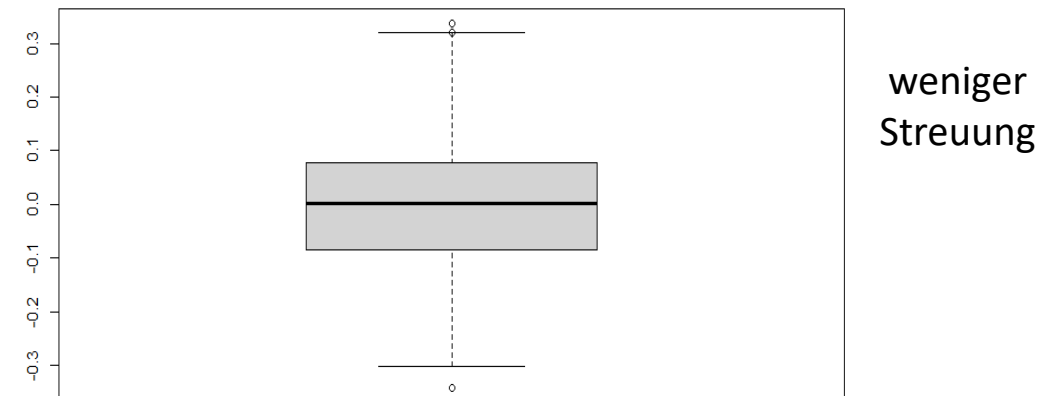
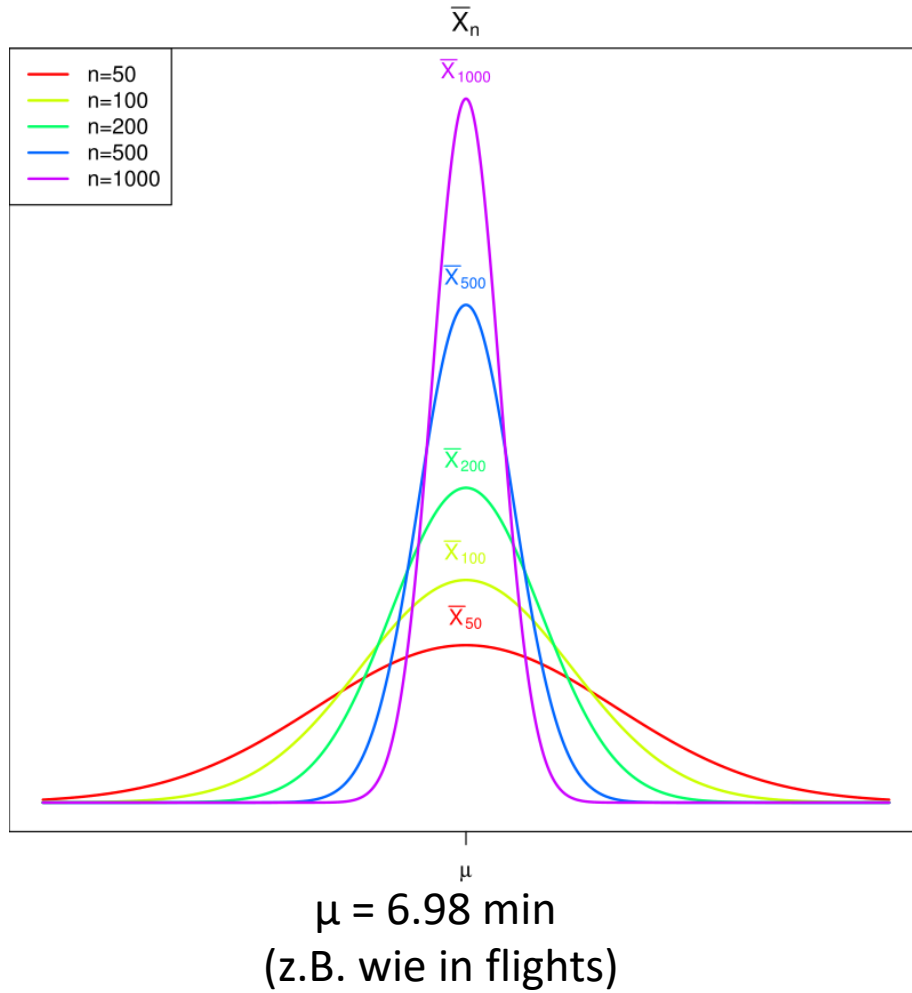


Fig.: Verteilung der Differenz zwischen population_mean und Stichprobenmittelwerte

Verteilung von Stichprobenmittelwerten

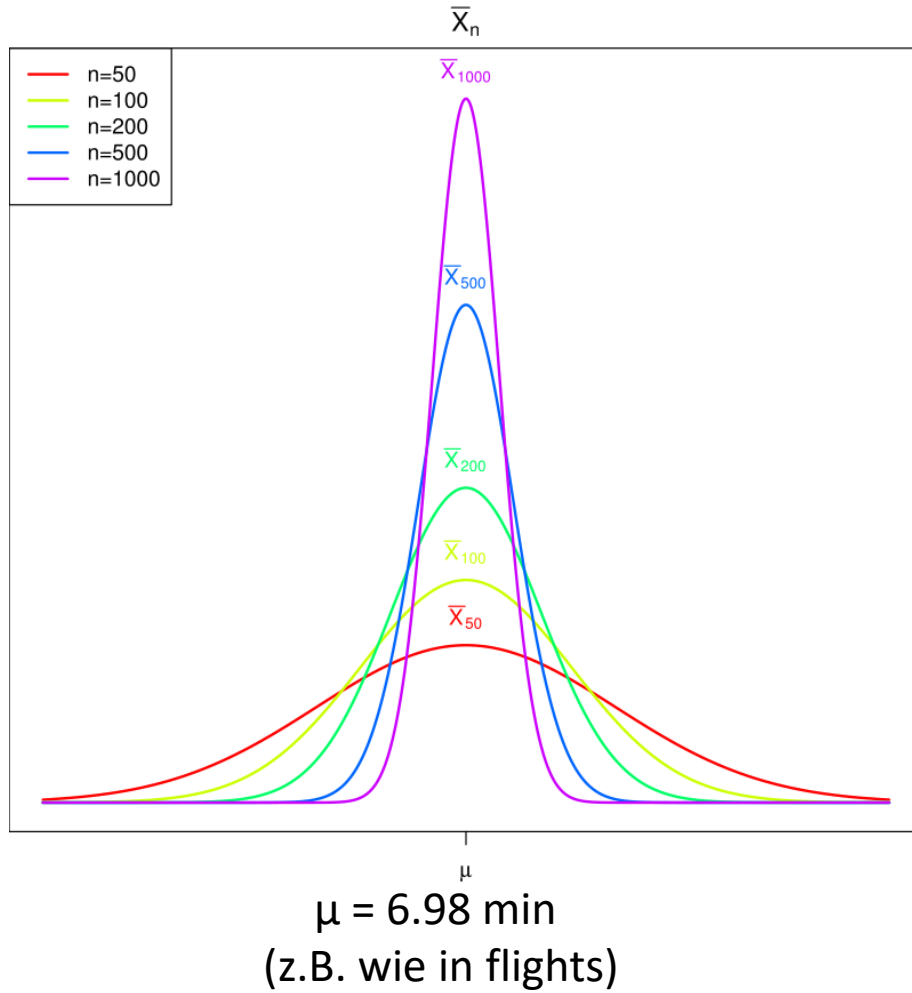
z.B. ‚flights‘



- Wir können nie wissen wie nah unser MW aus dem sample am wahren μ liegt, da μ in der Regel gesucht wird.
- Je größer die Stichprobe, desto schmaler ist die Verteilung der Stichprobenmittelwerte. Dadurch wird der Populationsmittelwert präziser geschätzt. Die Breite der Stichprobenverteilung wird durch die Streuung/ sd bestimmt.

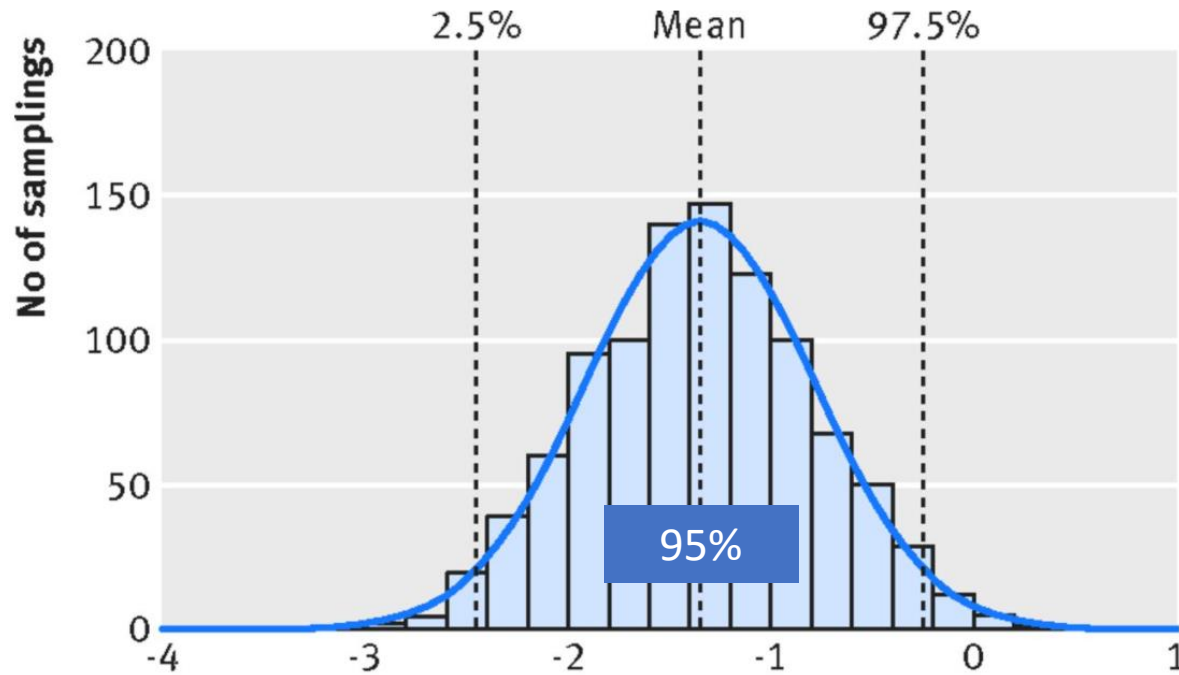
Verteilung von Stichprobenmittelwerten

z.B. ‚flights‘



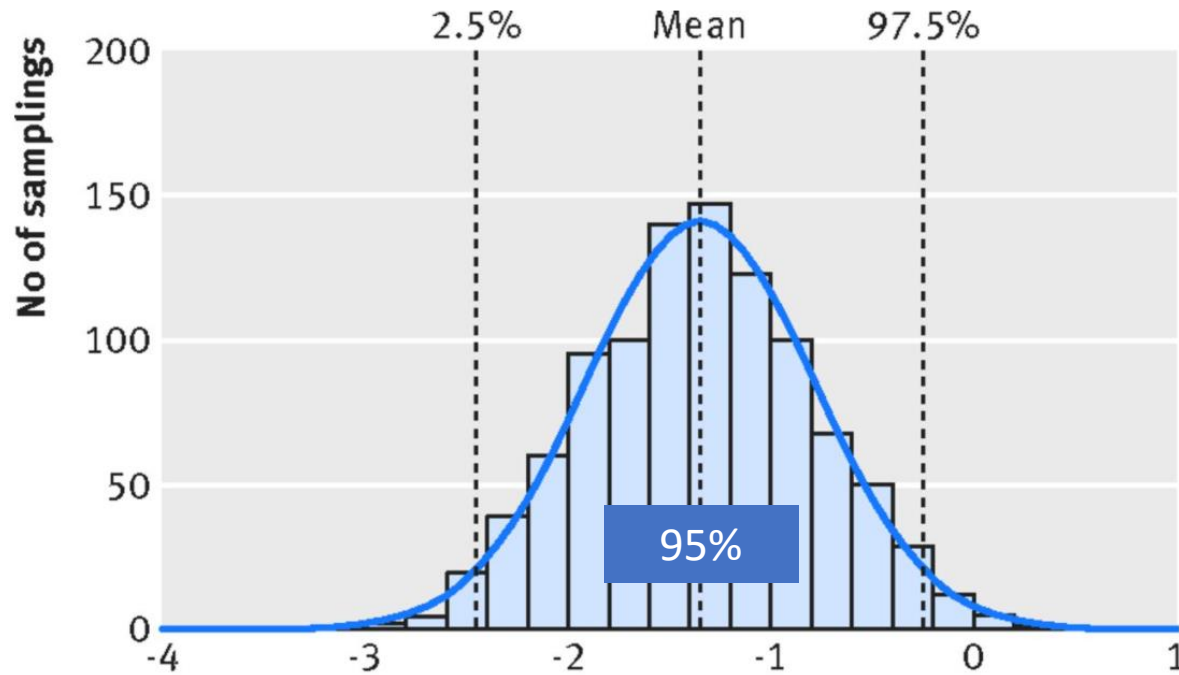
- **Wir benötigen demnach nur die sd der Stichprobenmittelwerte um die Genauigkeit unserer Punktschätzung anzugeben**
- Vorgehen:
 - 1) Standardfehler berechnen (theoretisch oder praktisch)
 - 2) 95%- Intervall berechnen

Punktschätzer und Konfidenzintervall



- Die Schätzgenauigkeit kann man oft besser durch eine untere und eine obere Grenze einordnen.
- Der Stichprobenmittelwert gilt dabei als **Punktschätzer**.
- Das Intervall nennt sich Vertrauensintervall, oder häufiger: **Konfidenzintervall**.

Punktschätzer und Konfidenzintervall

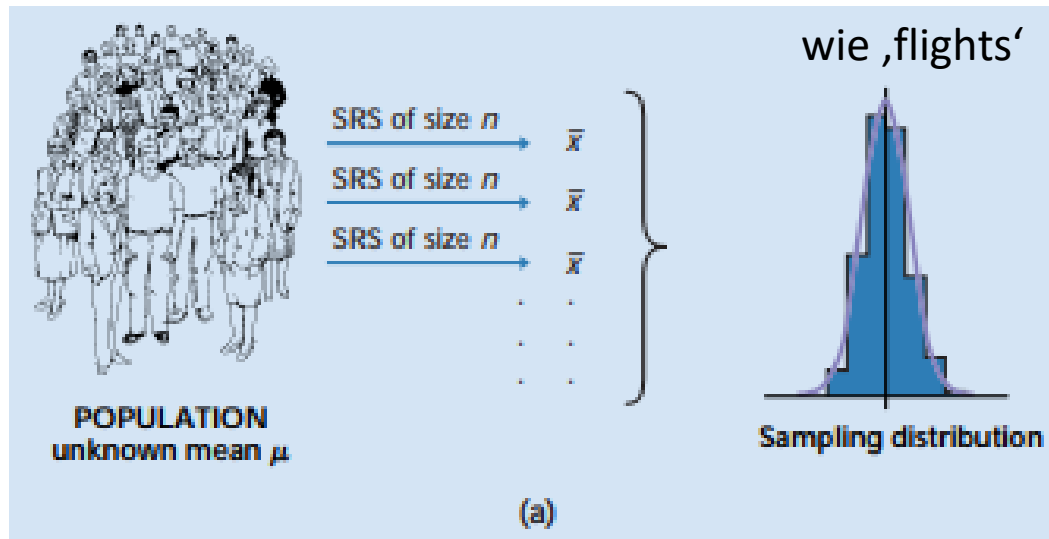


- Die Schätzgenauigkeit kann man oft besser durch eine untere und eine obere Grenze einordnen.
- Der Stichprobenmittelwert gilt dabei als **Punktschätzer**.
- Das Intervall nennt sich Vertrauensintervall, oder häufiger: **Konfidenzintervall**.

$$x_u = \bar{x} - z \cdot \frac{\sigma}{\sqrt{n}} \quad x_o = \bar{x} + z \cdot \frac{\sigma}{\sqrt{n}}$$

Mittels des Standardfehlers SE wird häufig die 95%-
Umgebung um den Punktschätzer angegeben.

Herangehensweise: empirische oder theoretische **Berechnung des Standardfehlers** (standard error: SE)



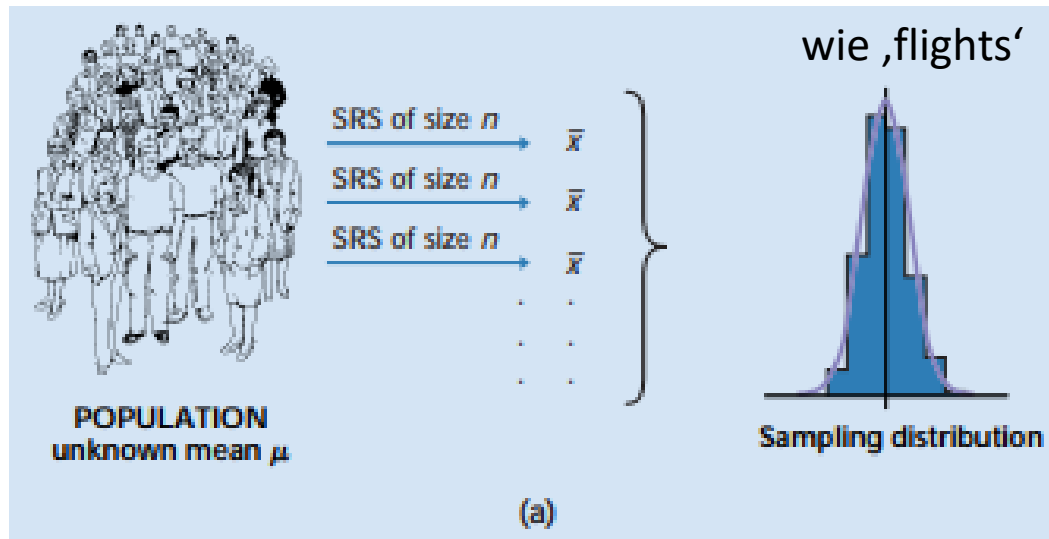
Praktische Berechnung:

ziehe samples

berechne means

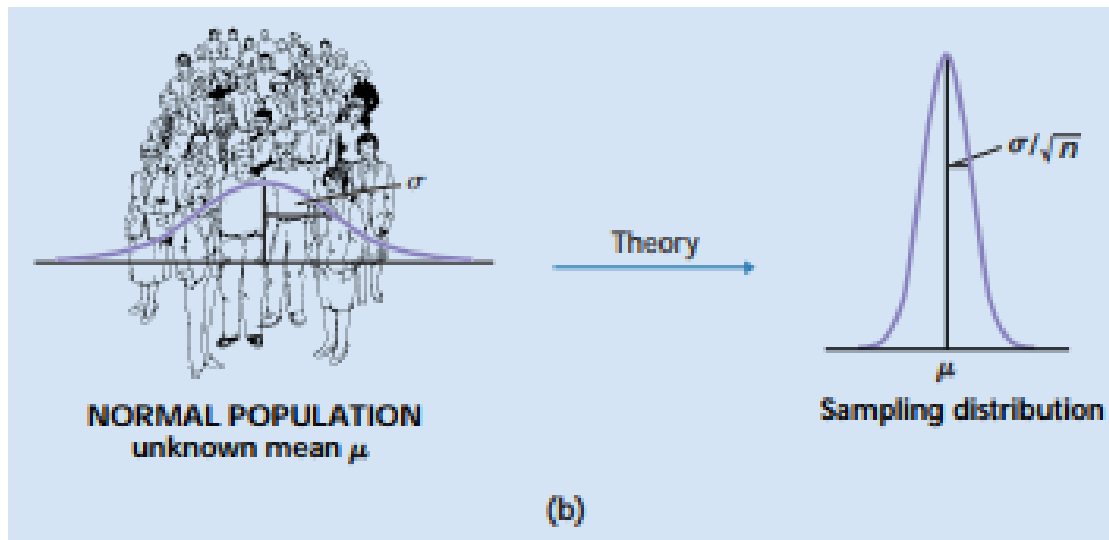
berechne sd dieser means (=Standardfehler)

Herangehensweise: empirische oder theoretische **Berechnung des Standardfehlers** (standard error: SE)



Praktische Berechnung:

ziehe samples
berechne means
berechne sd dieser means (=Standardfehler)



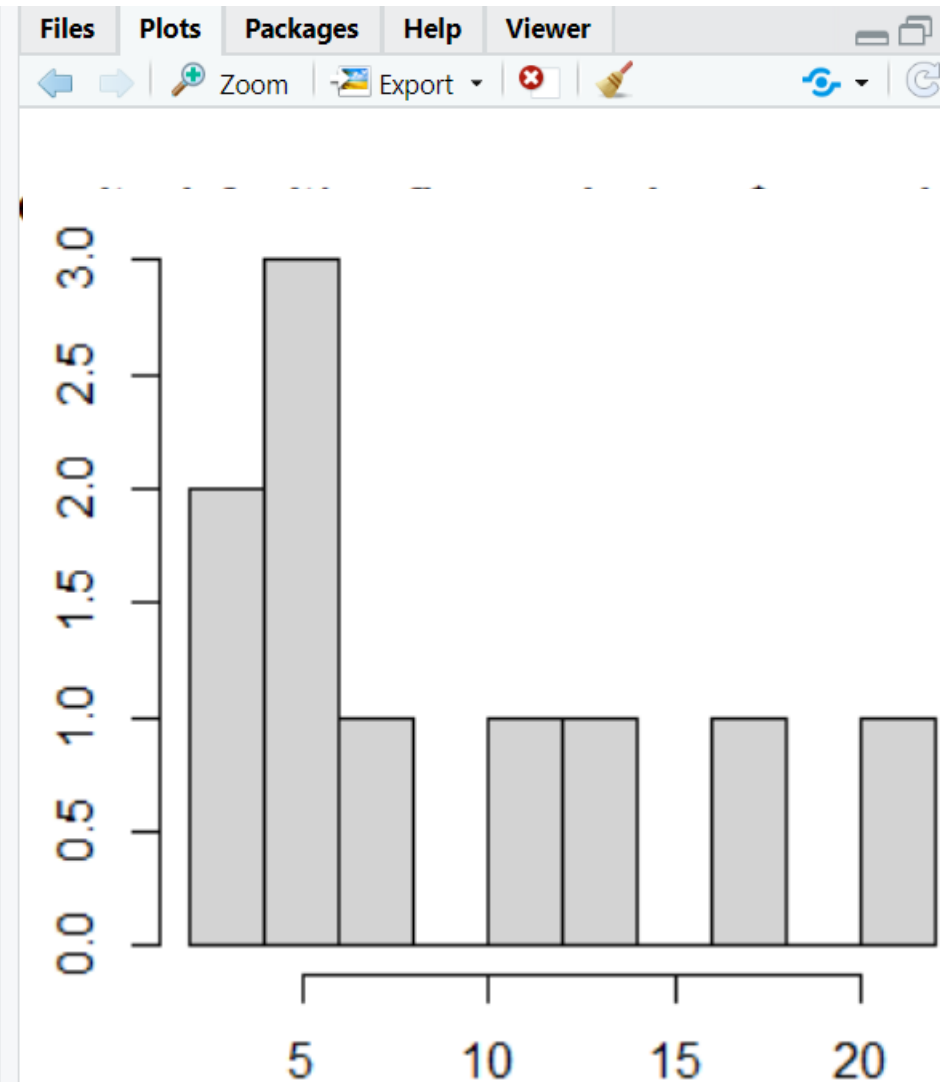
Theoretische Berechnung:

ziehe ein sample
berechne sd
teile sd durch Wurzel aus n (=Standardfehler)

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$

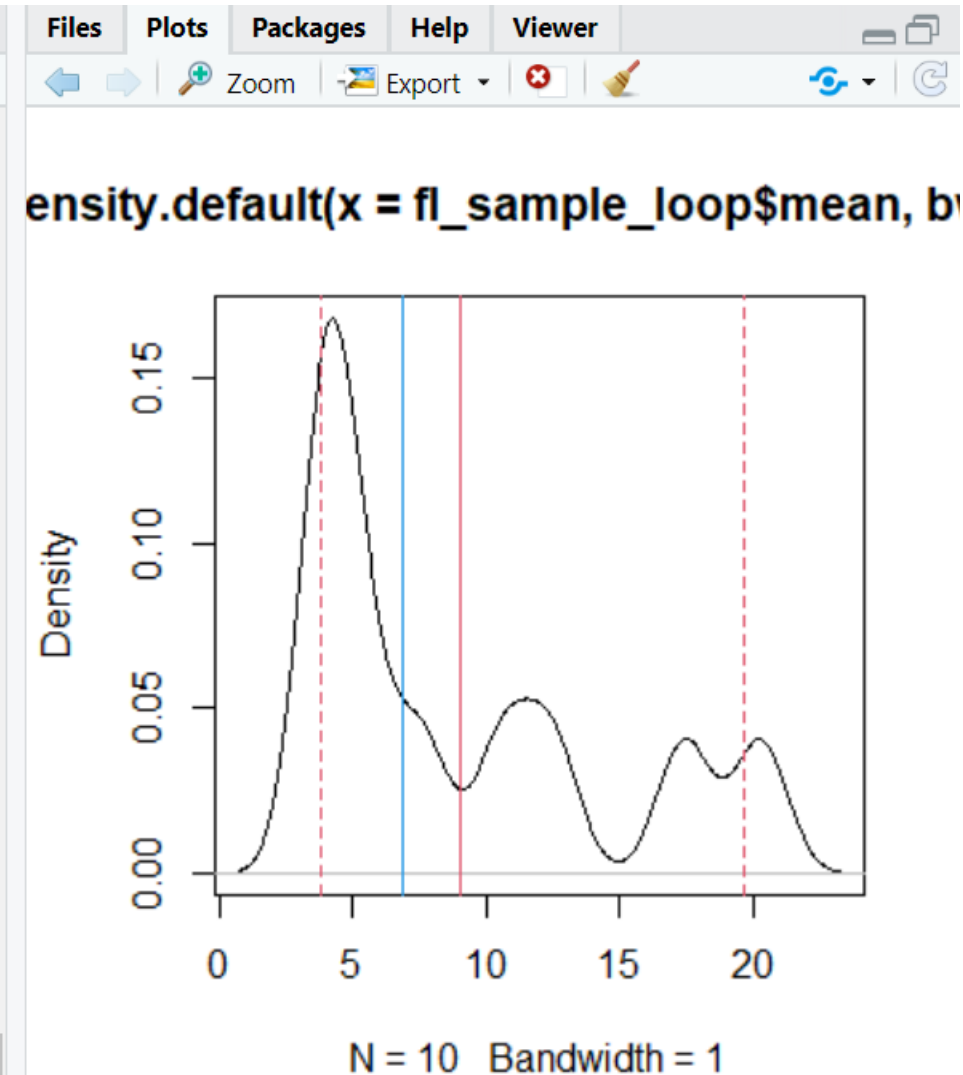
Kleine(re) Stichprobe (n=100)

```
Console Terminal x Jobs x
~/P2/
> # erzeuge x samples der GroÙe size und zeige favstats
> fl_sample_loop <- do(10) *
+   favstats(sample(flights_clean$arr_delay, size = 100))
>
> hist(fl_sample_loop$mean, breaks=10) #Histogramm
> plot(density(fl_sample_loop$mean,bw=1)) #Glättung Dichteplot
>
> mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
[1] 6.895377
> abline(v = mean(flights_clean$arr_delay), col = 4) #mean population
>
> m_<-mean(fl_sample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 9.044
> abline(v = mean(fl_sample_loop$mean), col = 2) #mean StichprobenMW
>
> ## 2.5% und 97.5% quantile (mittleren 95%)
> q<-quantile(fl_sample_loop$mean, prob=c(0.025,0.975));q
      2.5%      97.5%
3.79700 19.64875
> abline(v = q, col = 2, lty = 2)
>
> sd(fl_sample_loop$mean) #Standardfehler, empirisch
[1] 5.991919
>
> # Berechnung des theoretischen SE = sd/wurzel n
> se_t<-sd(sample(flights_clean$arr_delay, size = 100))/sqrt(100);se_t
[1] 3.106828
```



Kleine(re) Stichprobe (n=100)

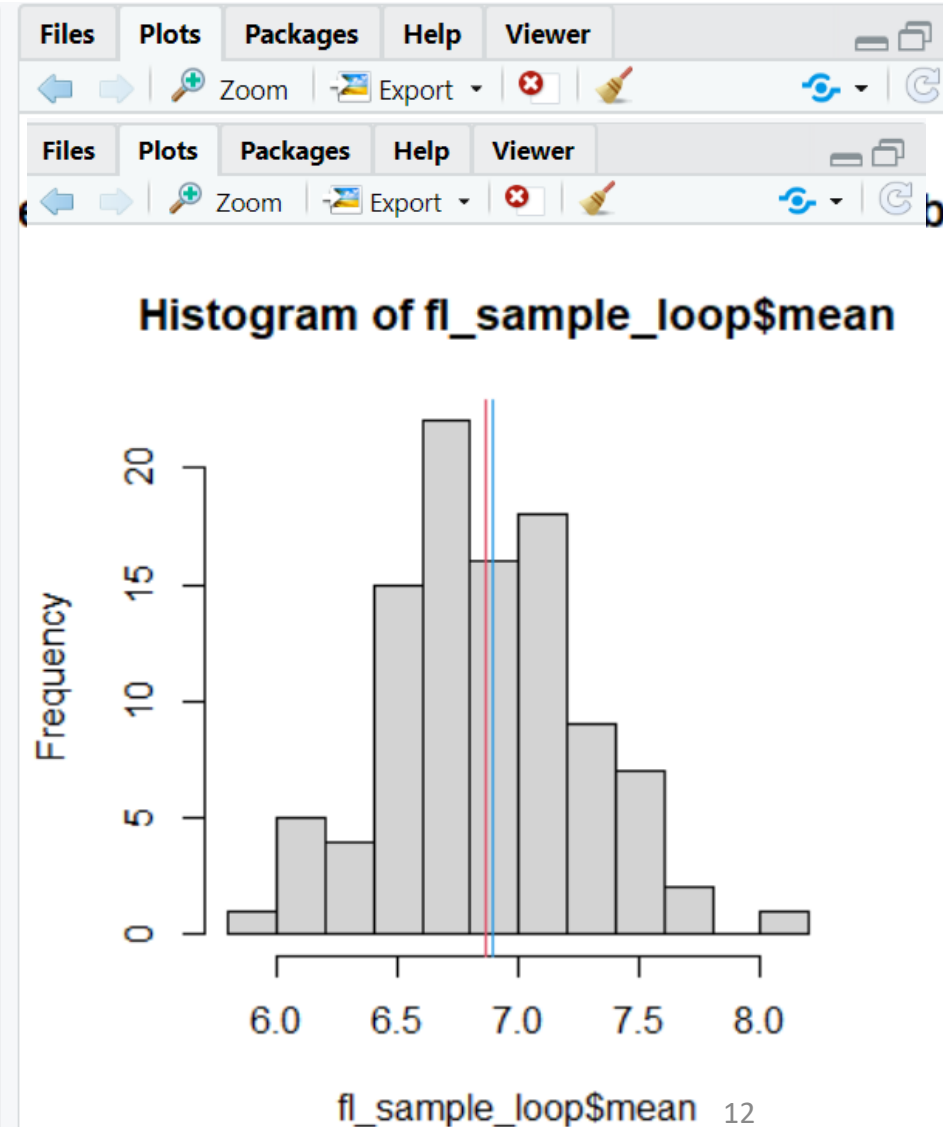
```
Console Terminal x Jobs x
~/P2/
> # erzeuge x samples der GroÙe size und zeige favstats
> fl_sample_loop <- do(10) *
+   favstats(sample(flights_clean$arr_delay, size = 100))
>
> hist(fl_sample_loop$mean, breaks=10) #Histogramm
> plot(density(fl_sample_loop$mean,bw=1)) #Glättung Dichteplot
>
> mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
[1] 6.895377
> abline(v = mean(flights_clean$arr_delay), col = 4) #mean population
>
> m_<-mean(fl_sample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 9.044
> abline(v = mean(fl_sample_loop$mean), col = 2) #mean StichprobenMW
>
> ## 2.5% und 97.5% quantile (mittleren 95%)
> q<-quantile(fl_sample_loop$mean, prob=c(0.025,0.975));q
      2.5%      97.5%
3.79700 19.64875
> abline(v = q, col = 2, lty = 2)
>
> sd(fl_sample_loop$mean) #Standardfehler, empirisch
[1] 5.991919
>
> # Berechnung des theoretischen SE = sd/wurzel n
> se_t<-sd(sample(flights_clean$arr_delay, size = 100))/sqrt(100);se_t
[1] 3.106828
```



Große Stichprobe (n=10.000)

```
Console Terminal x Jobs x
~/P2/ ↵
> # erzeuge x samples der Größe size und zeige favstats
> fl_sample_loop <- do(100) *
+   favstats(sample(flights_clean$arr_delay, size = 10000))
>
> hist(fl_sample_loop$mean, breaks=10) #Histogramm
> plot(density(fl_sample_loop$mean,bw=1)) #Glättung Dichteplot
>
> mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
[1] 6.895377
> abline(v = mean(flights_clean$arr_delay), col = 4) #mean population
>
> m_<-mean(fl_sample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 6.967949
> abline(v = mean(fl_sample_loop$mean), col = 2) #mean StichprobenMW
>
> ## 2.5% und 97.5% quantile (mittleren 95%)
> q<-quantile(fl_sample_loop$mean, prob=c(0.025,0.975));q
      2.5%      97.5%
5.965695 7.818585
> abline(v = q, col = 2, lty = 2)
>
> sd(fl_sample_loop$mean) #Standardfehler, empirisch
[1] 0.4835039
>
> # Berechnung des theoretischen SE = sd/wurzel n
> se_t<-sd(sample(flights_clean$arr_delay, size = 10000))/sqrt(10000);se_t
[1] 0.4513942
```

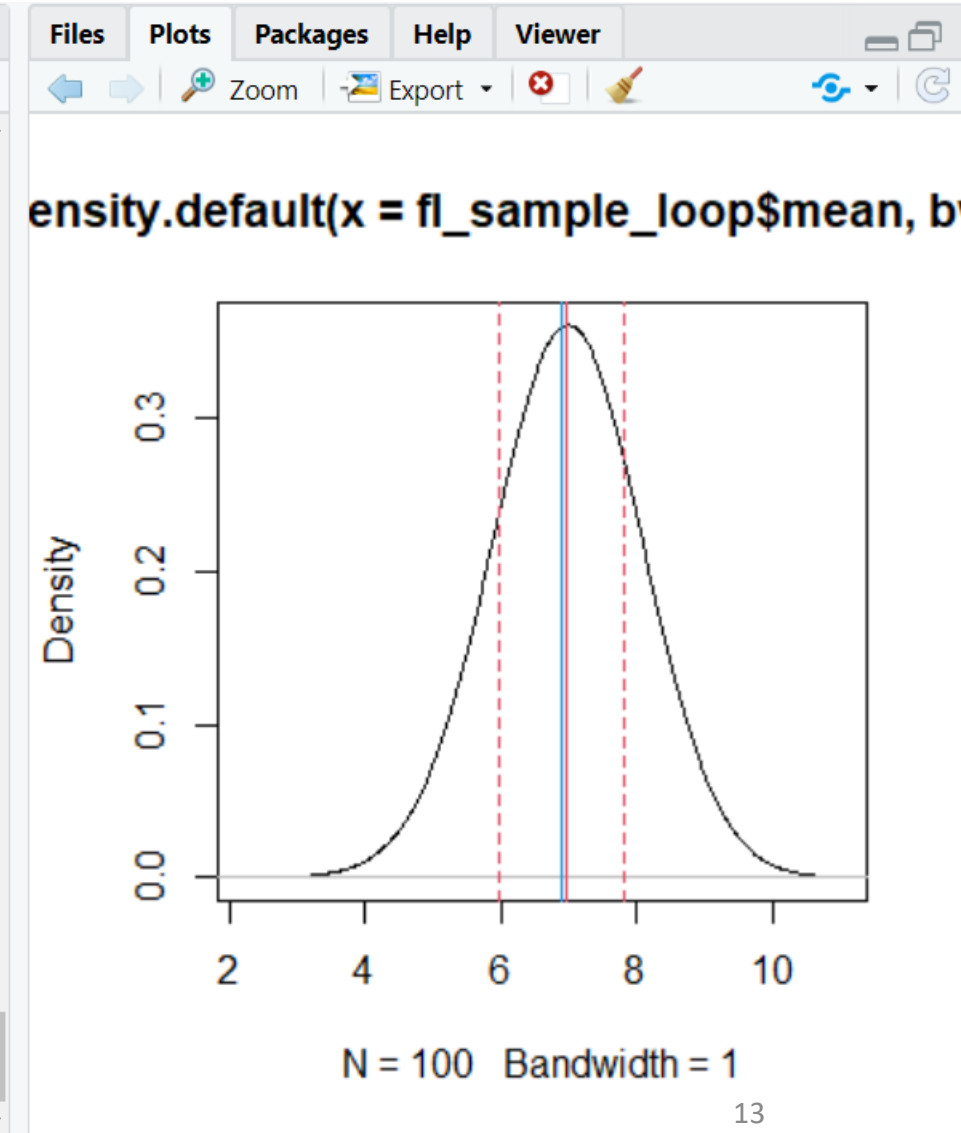
Prof. Dr. rer. nat. T. Wiebringhaus



Große Stichprobe (n=10.000)

```
Console Terminal x Jobs x
~/P2/ ↗
> # erzeuge x samples der Größe size und zeige favstats
> fl_sample_loop <- do(100) *
+   favstats(sample(flights_clean$arr_delay, size = 10000))
>
> hist(fl_sample_loop$mean, breaks=10) #Histogramm
> plot(density(fl_sample_loop$mean,bw=1)) #Glättung Dichteplot
>
> mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
[1] 6.895377
> abline(v = mean(flights_clean$arr_delay), col = 4) #mean population
>
> m_<-mean(fl_sample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 6.967949
> abline(v = mean(fl_sample_loop$mean), col = 2) #mean StichprobenMW
>
> ## 2.5% und 97.5% quantile (mittleren 95%)
> q<-quantile(fl_sample_loop$mean, prob=c(0.025,0.975));q
      2.5%      97.5%
5.965695 7.818585
> abline(v = q, col = 2, lty = 2)
>
> sd(fl_sample_loop$mean) #Standardfehler, empirisch
[1] 0.4835039
>
> # Berechnung des theoretischen SE = sd/wurzel n
> se_t<-sd(sample(flights_clean$arr_delay, size = 10000))/sqrt(10000);se_t
[1] 0.4513942
```

Prof. Dr. rer. nat. T. Wiebringhaus



QM7

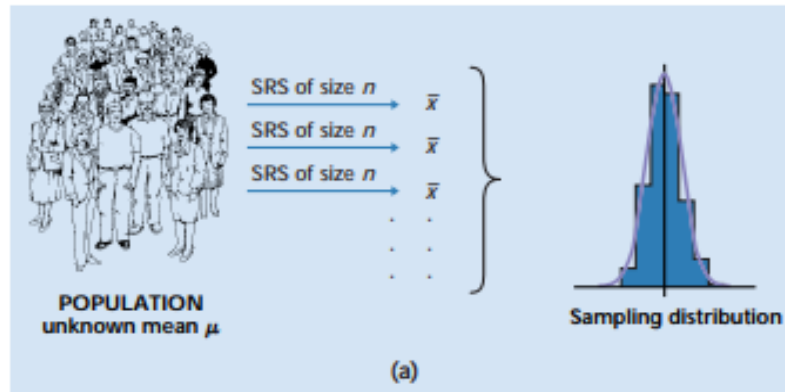
Einführung simulationsbasierte Inferenz (SBI):

Bootstrap (resample) Konfidenzintervall

Schätzen vs. Hypothesen testen

Ein Ausreißertest

Herangehensweise: empirische oder theoretische **Berechnung des Standardfehlers** (standard error: SE)

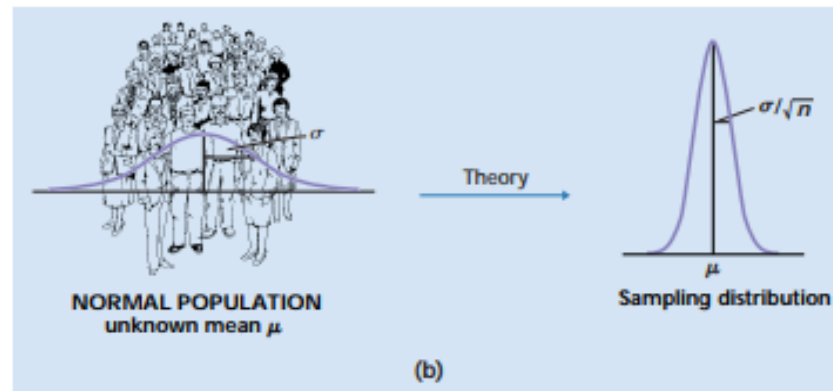


Praktische Berechnung:

ziehe samples

berechne means

berechne sd dieser means (=Standardfehler)



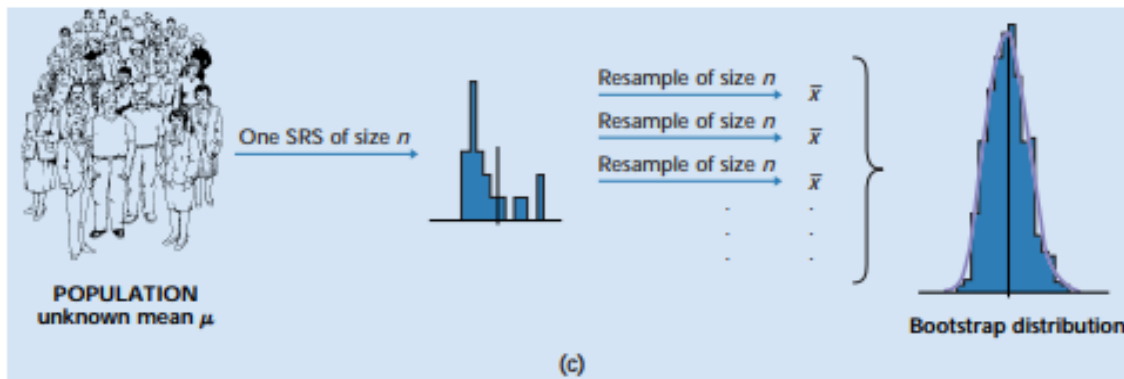
Theoretische Berechnung:

ziehe ein sample

berechne sd

teile sd durch Wurzel aus n (=Standardfehler)

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$



<- QM7: Resampling

samplen vs. resamplen

```
QM5c.R x Funktionen.R x QM6c QM7.R* x
Source on Save Run Source
11
12 ### Resampeln: mehrfaches Ziehen aus EINER Stichprobe #####
13
14 # Schritt 1: ziehe EIN sample
15
16 fl_sample<- sample(flights_clean$arr_delay, size = 3) #2500
17 fl_sample
18 mean(fl_sample)
19 |
20 # Schritt 2: resample: Ziehen mit Zurücklegen aus unserem (einzigem) sample
21 fl_resample <- resample(fl_sample, size = 3)
22 fl_resample
23
19:1 (Top Level) R Script
Console Terminal x Jobs x
~/P2/
> fl_sample<- sample(flights_clean$arr_delay, size = 3) #2500
> fl_sample
[1] -12 -29 128
> fl_resample <- resample(fl_sample, size = 3)
> fl_resample
[1] 128 -29 128
>
Mehrfache können gezogen werden
```

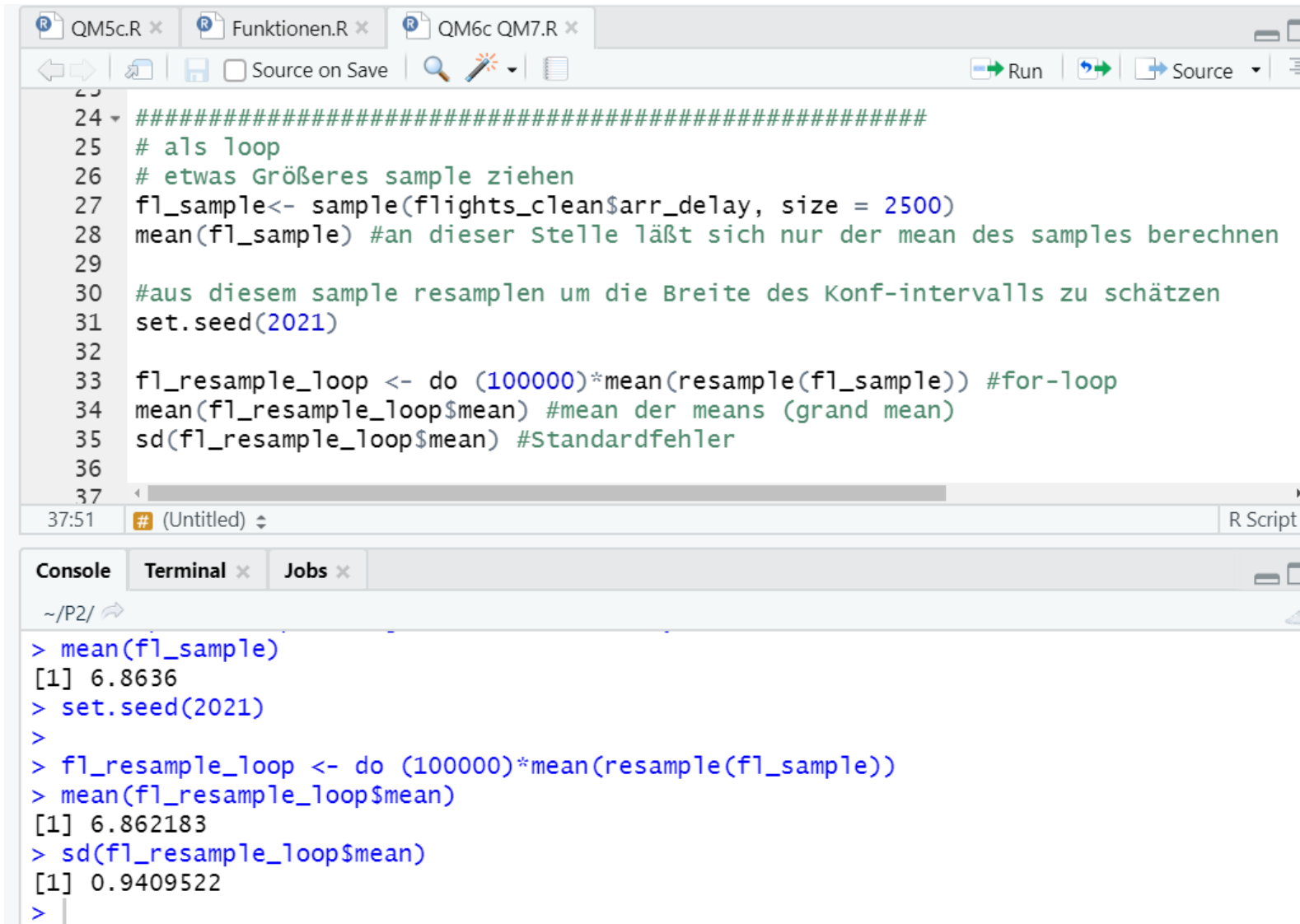
Ziehen ohne Zurücklegen
(samplen)

Ziehen mit Zurücklegen
(resampeln)

Ziehe ein sample der Größe = 3

Ziehe daraus ein sample der Größe 3.
Zwischendurch zurücklegen

Wiederholtes resamplen (bootstrap)



The screenshot displays the RStudio environment. The top pane shows an R script with the following code:

```
23  
24 #####  
25 # als loop  
26 # etwas Größeres sample ziehen  
27 fl_sample<- sample(flights_clean$arr_delay, size = 2500)  
28 mean(fl_sample) #an dieser Stelle lässt sich nur der mean des samples berechnen  
29  
30 #aus diesem sample resamplen um die Breite des Konf-intervalls zu schätzen  
31 set.seed(2021)  
32  
33 fl_resample_loop <- do (100000)*mean(resample(fl_sample)) #for-loop  
34 mean(fl_resample_loop$mean) #mean der means (grand mean)  
35 sd(fl_resample_loop$mean) #Standardfehler  
36  
37
```

The bottom pane shows the console output for the executed code:

```
> mean(fl_sample)  
[1] 6.8636  
> set.seed(2021)  
>  
> fl_resample_loop <- do (100000)*mean(resample(fl_sample))  
> mean(fl_resample_loop$mean)  
[1] 6.862183  
> sd(fl_resample_loop$mean)  
[1] 0.9409522  
>
```

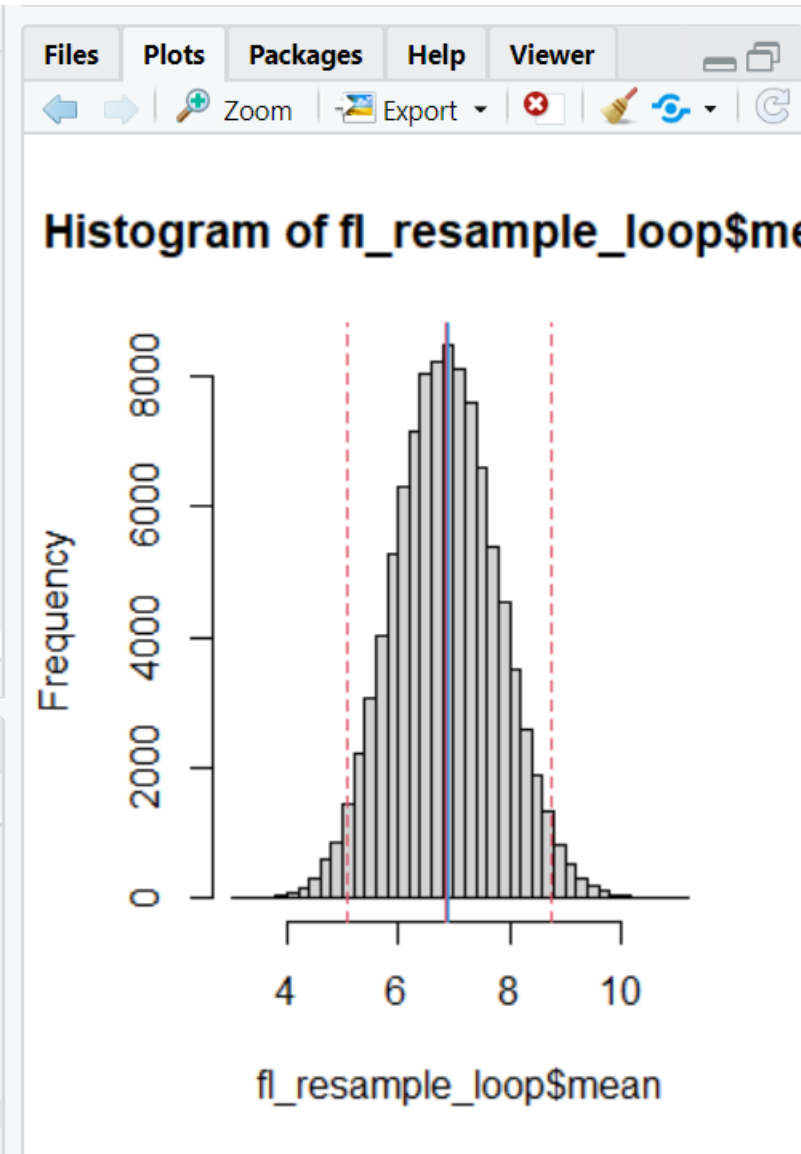
Wiederholtes resamplen (bootstrap)

```
37 hist(fl_resample_loop$mean, breaks=50) #Histogramm
38 plot(density(fl_resample_loop$mean,bw=1)) #Glättung Dichteplot
39
40 mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
41 abline(v = mean(flights_clean$arr_delay), col = 4) #mean population
42
43 m_<-mean(fl_resample_loop$mean);m_ # MW der Mittelwerte der samples
44 abline(v = mean(fl_resample_loop$mean), col = 2) #mean StichprobenMW
45
46 ## 2.5% und 97.5% quantile (mittleren 95%)
47 q<-quantile(fl_resample_loop$mean, prob=c(0.025,0.975));q
48 abline(v = q, col = 2, lty = 2)
49
50 paste("Der Punktschätzer beträgt",mean(fl_resample_loop$mean),". Der wahre unbek
51      in 95% der Stichproben zwischen ",q[1]," und",q[2])
52
```

37:1 # (Untitled) R Script

Console Terminal Jobs

```
> mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
[1] 6.895377
> m_<-mean(fl_resample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 6.862183
> paste("Der Punktschätzer beträgt",mean(fl_resample_loop$mean),". Der wahre unbekannt
e Populationsparameter  $\mu$  liegt
+      in 95% der Stichproben zwischen ",q[1]," und",q[2])
[1] "Der Punktschätzer beträgt 6.862183416 . Der wahre unbekannte Populationsparameter
 $\mu$  liegt\n      in 95% der Stichproben zwischen 5.06519 und 8.74161"
```



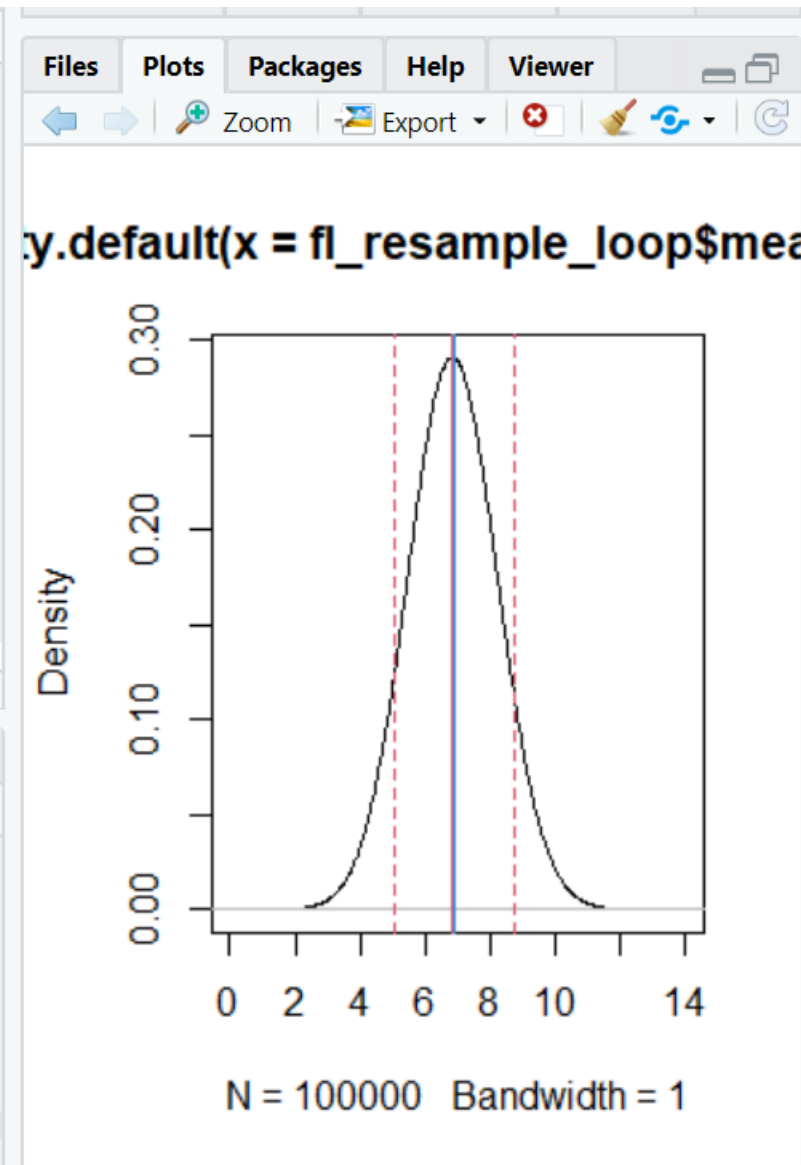
Wiederholtes resamplen (bootstrap)

```
37 hist(fl_resample_loop$mean, breaks=50) #Histogramm
38 plot(density(fl_resample_loop$mean,bw=1)) #Glättung Dichteplot
39
40 mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
41 abline(v = mean(flights_clean$arr_delay), col = 4) #mean population
42
43 m_<-mean(fl_resample_loop$mean);m_ # MW der Mittelwerte der samples
44 abline(v = mean(fl_resample_loop$mean), col = 2) #mean StichprobenMW
45
46 ## 2.5% und 97.5% quantile (mittleren 95%)
47 q<-quantile(fl_resample_loop$mean, prob=c(0.025,0.975));q
48 abline(v = q, col = 2, lty = 2)
49
50 paste("Der Punktschätzer beträgt",mean(fl_resample_loop$mean),". Der wahre unbek
51      in 95% der Stichproben zwischen ",q[1]," und",q[2])
52
```

37:1 # (Untitled) R Script

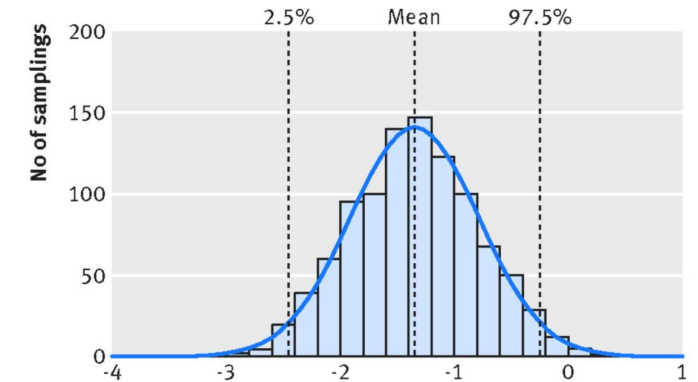
Console Terminal Jobs

```
> mu<-mean(flights_clean$arr_delay);mu # Mittelwert der population
[1] 6.895377
> m_<-mean(fl_resample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 6.862183
> paste("Der Punktschätzer beträgt",mean(fl_resample_loop$mean),". Der wahre unbekannt
e Populationsparameter  $\mu$  liegt
+      in 95% der Stichproben zwischen ",q[1]," und",q[2])
[1] "Der Punktschätzer beträgt 6.862183416 . Der wahre unbekannte Populationsparameter
 $\mu$  liegt\n      in 95% der Stichproben zwischen 5.06519 und 8.74161"
```



Schätzen vs. Hypothesen prüfen

- Estimation (Schätzung) -> Confidence interval
 - In welchem Intervall liegt der unbekannte Wert?
- Decision (Entscheidung) -> Hypothesis test



Forschungsfragen beantworten (z.B.):

Haben die Flüge am JFK eine größere Verspätung?

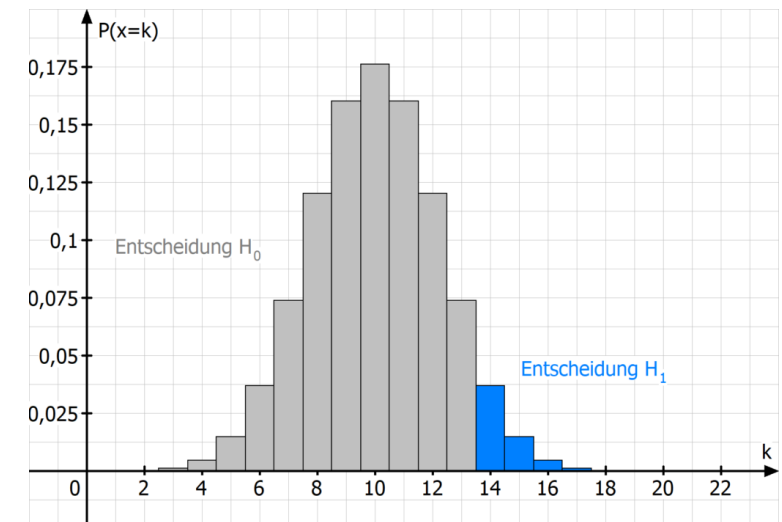
(angenommen die Gesamtpopulation würde nicht vorliegen)

Hypothesenpaar formulieren

H_1 : Die Flüge am JFK haben eine größere Verspätung

H_0 : Die Flüge am JFK haben **keine** größere Verspätung

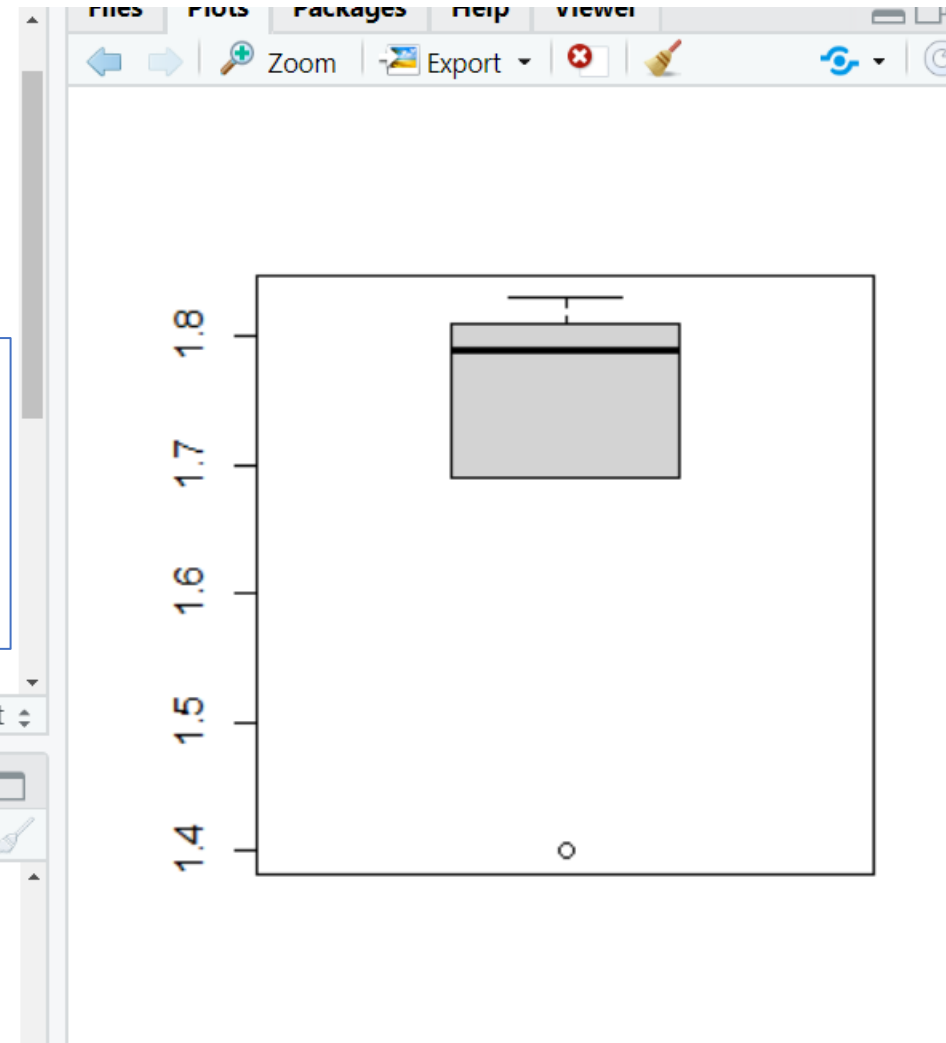
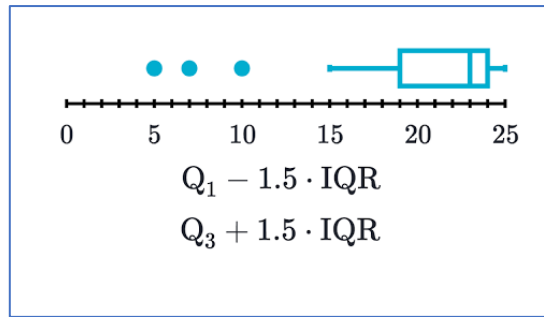
(in der Nullhypothese wird von Gleichheit ausgegangen)



Ein Ausreißertest

Ausreißertest nach IQR

```
3 # Ausreißertest: Es liegt eine Messung von Körpergrößen vor
4
5 height <- c(1.78,1.69,1.81,1.83,1.80,1.40) #c für Vektor
6 favstats(height) #Kennzahlen anzeigen
7 boxplot(height) #visualisieren
8 # der Wert 1.40 m wird hier als Ausreißer (outlier) gesehen, siehe
9 # Punkt unten unterhalb des boxplots
10 #iqr_test (nach boxplot), untere und obere Grenze
11 quantile(height) # wir brauchen Q1 und Q3
12
13 iqr_test_u <- function(x){
14   quantile(x)[2]-1.5*(iqr(x))    #Q1-1.5*IQR
15 }
16 iqr_test_o <- function(x){
17   quantile(x)[4]+1.5*(iqr(x))    #Q3+1.5*IQR
18 }
19 u<-iqr_test_u(height);u
20 o<-iqr_test_o(height);o
```



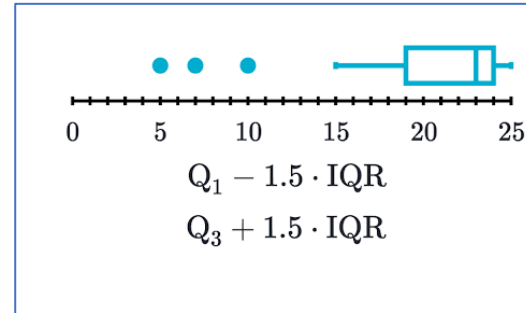
```
20:24 (Top Level) R Script
```

min	Q1	median	Q3	max	mean	sd	n	missing
1.4	1.7125	1.79	1.8075	1.83	1.718333	0.1633911	6	0

```
> quantile(height)
 0%    25%   50%   75%  100%
1.4000 1.7125 1.7900 1.8075 1.8300
```

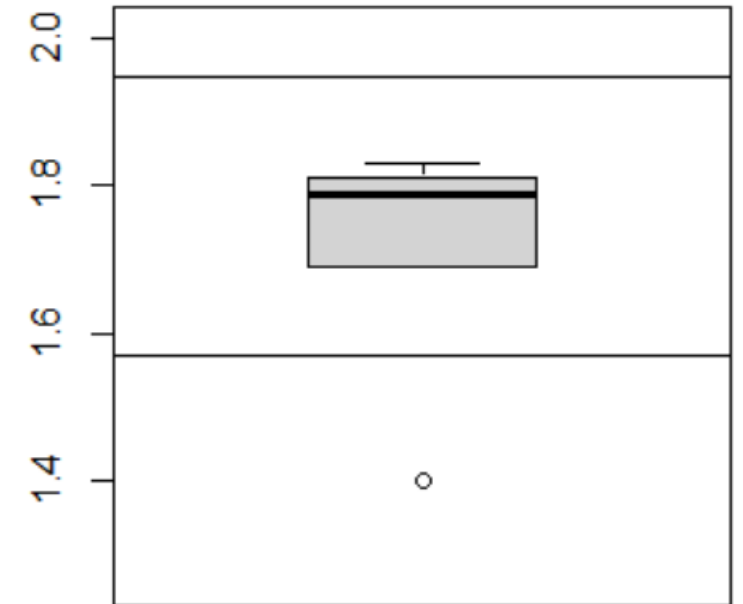
Ausreißertest nach IQR

```
10 #iqr_test (nach boxplot), untere und obere Grenze
11 quantile(height) # wir brauchen Q1 und Q3
12
13 iqr_test_u <- function(x){
14   quantile(x)[2]-1.5*(iqr(x)) #Q1-1.5*IQR
15 }
16 iqr_test_o <- function(x){
17   quantile(x)[4]+1.5*(iqr(x)) #Q3+1.5*IQR
18 }
19 u<-iqr_test_u(height);u
20 o<-iqr_test_o(height);o
21
22 boxplot(height,ylim=c(0.9*min(height),1.1*max(height)))
23 abline(h=u); abline(h=o)
```



Files Plots Packages Help Viewer

Zoom Export



Console Terminal x Jobs x

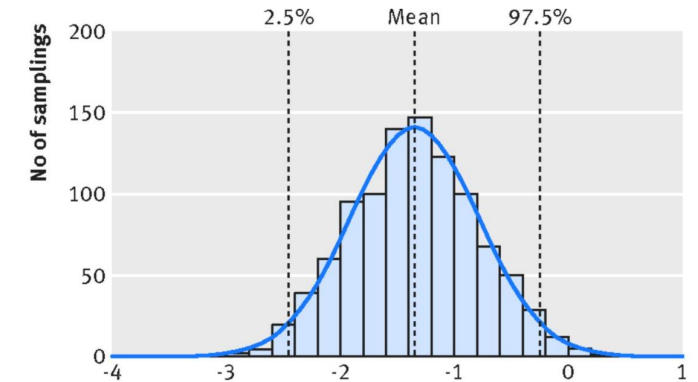
~/P2/

```
>
> u<-iqr_test_u(height);u
25%
1.57
> o<-iqr_test_o(height);o
75%
1.95
> boxplot(height,ylim=c(0.9*min(height),1.1*max(height)))
> abline(h=u); abline(h=o)
```


Simulationsbasierte Inferenz (SBI): ein Hypothesentest

Schätzen vs. Hypothesen prüfen

- Estimation (Schätzung) -> Confidence interval
 - In welchem Intervall liegt der unbekannte Wert?



- Decision (Entscheidung) -> Hypothesis test

Forschungsfragen beantworten (z.B.):

Haben die Flüge am JFK eine *kleinere* Verspätung?

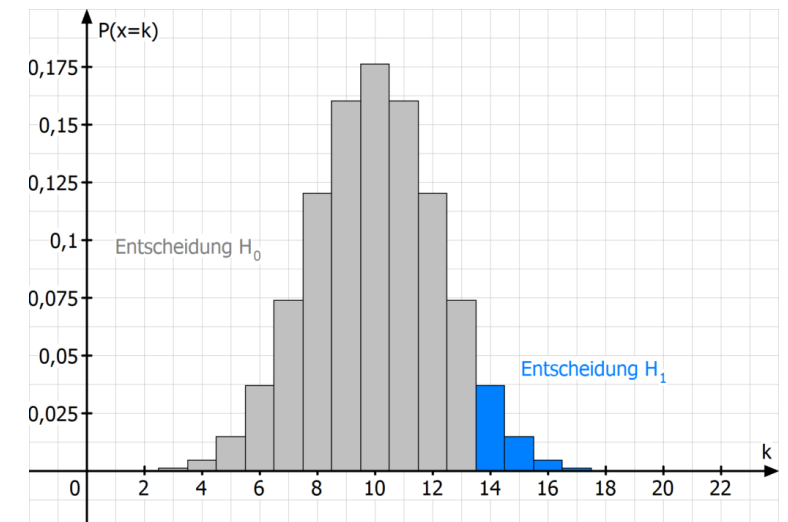
(angenommen die Gesamtpopulation würde nicht vorliegen)

Hypothesenpaar formulieren

H_1 : Die Flüge am JFK haben eine kleinere Verspätung

H_0 : Die Flüge am JFK haben **keine** kleinere Verspätung

(in der Nullhypothese wird von Gleichheit/ keine Auffälligkeit ausgegangen)



Hypothesen testen: formaler Aufbau

H_1 : Es gibt einen Unterschied (etwas ist größer/kleiner etc.)

H_0 : Es gibt **keinen** Unterschied

Beispiele:

Frage: Sind die Ankünfte am JFK pünktlicher?

H_1 : Die Flüge am JFK haben weniger Verspätung

H_0 : Die Flüge am JFK haben **nicht weniger/ gleiche** Verspätung

Frage: Hilft der Impfstoff?

H_1 : Der Impfstoff erhöht die Immunität

H_0 : Der Impfstoff erhöht **nicht** die Immunität

Hilft: Ist die Münze fair?

H_1 : Die Münze ist gezinkt

H_0 : Die Münze ist **nicht** gezinkt

μ : unbekannter **Mittelwert** in der Gesamtheit(Population)

(π : unbekannter **Anteil** in der Gesamtheit(Population))

$$H_1: \mu_{\text{JFK}} < \mu_{\text{non_JFK}}$$

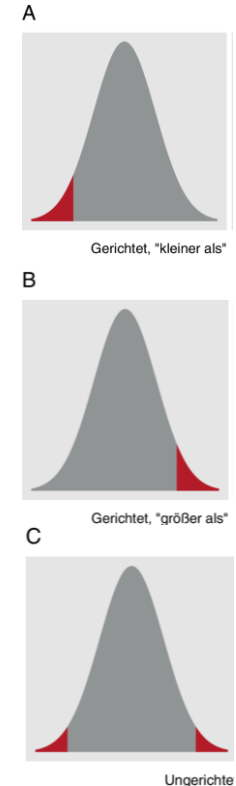
$$H_0: \mu_{\text{JFK}} \geq \mu_{\text{non_JFK}}$$

$$H_1: \mu_{\text{impf}} > \mu_{\text{non_impf}}$$

$$H_0: \mu_{\text{impf}} \leq \mu_{\text{non_impf}}$$

$$H_1: \mu_{\text{Kopf}} \neq \mu_{\text{Zahl}}$$

$$H_0: \mu_{\text{Kopf}} = \mu_{\text{Zahl}}$$

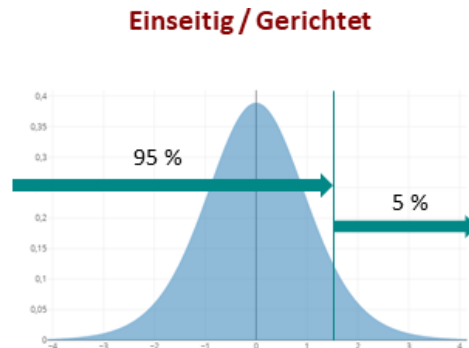
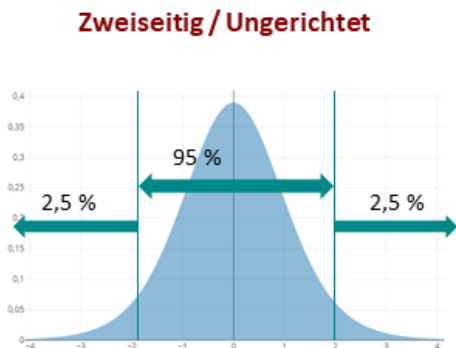


gerichtete (einseitige) vs. ungerichtete (zweiseitige) Hypothesen

Ein Hypothesentest hat das Ziel eine wissenschaftliche Forschungsfrage mit empirischen Methoden zu beantworten. Dazu benötigen wir Stichproben, die mit Unsicherheit behaftet sind (z.B. durch einen zu kleinen Stichprobenumfang oder fehlender Repräsentativität).

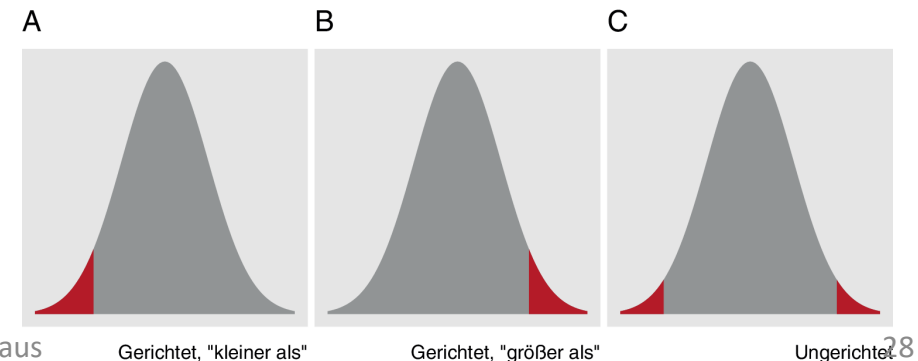
Wir prüfen z.B. ob ein Mittelwert noch mit der Schwankung des Zufallsprozesses (Streuung) zu erklären ist oder nicht. Die Verteilung und Streuung der Stichprobenmittelwerte (Standardfehler) zeigt welche Werte unter der Nullhypothese (kein Unterschied) am Wahrscheinlichsten sind (basierend auf den vorliegenden Daten).

Die Übergänge zu den weniger wahrscheinlichen Beobachtungen sind fließend sodass eine Regel für die Entscheidung festgelegt werden muss: typischerweise sagen wir dass wenn eine Beobachtung eine Wahrscheinlichkeit von $< 5\%$ hat, dieses Ergebnis *selten* ist. In diesem Fall wird die Grundannahme (dass es keine Auffälligkeit gibt: die Nullhypothese) verworfen (falsifiziert). Andernfalls, wenn das Ergebnis nicht selten ist unter der Grundannahme ($> 5\%$), können wir die Nullhypothese nicht verwerfen. In diesem Fall wird sie bestätigt (verifiziert).



Prof. Dr. rer. nat. T. Wiebringhaus

Beispiel: Intelligenzquotient

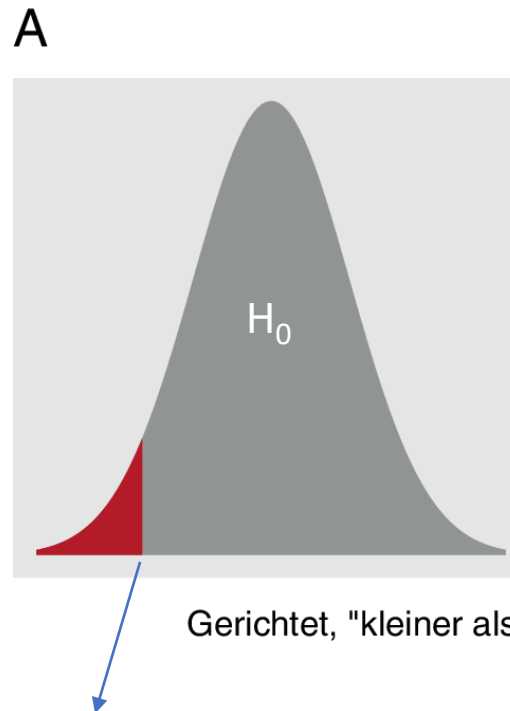


Beispiel Hypothesentest

Frage: Sind die Ankünfte am JFK weniger verspätet als an non_JFK?

H_1 : Die Flüge am JFK haben weniger Verspätung (Ankunft) \leftarrow gilt wenn $< 5\%$

H_0 : Die Flüge am JFK haben **nicht weniger** Verspätung (Ankunft) \leftarrow bleibt bei $\geq 5\%$



$H_1: \mu_{\text{JFK}} < \mu_{\text{non_JFK}}$ \leftarrow gilt wenn $< 5\%$

$H_0: \mu_{\text{JFK}} \geq \mu_{\text{non_JFK}}$ \leftarrow bleibt bei $\geq 5\%$

$\alpha = 5\%$ (auch 1% oder 0,1%) Signifikanzniveau/ Irrtumswahrscheinlichkeit

Beispiel Hypothesentest

Frage: Sind die Ankünfte am JFK weniger verspätet als an non_JFK?

```
QM6c QM7 QM8.R x flights x Funktionen.R x jfk x
Source on Save Run Source
56
57 ##### QM8 #####
58
59 # arr_delay filtern nach 'JFK' und 'non_JFK'
60 #flights_clean <- drop_na(flights, arr_delay) #ohne NA
61
62 # nur jfk
63 jfk <- flights_clean %>% filter(origin == "JFK") %>% select(arr_delay)
64 str(jfk) #109.079
65
66 # nicht jfk
67 non_jfk <- flights_clean %>% filter(origin != "JFK") %>% select(arr_de
68 str(non_jfk) #218.267
69 # 109.079 + 218.267 = 327.346
70
71 # Anzahl prüfen mit Pupulation (ohne NA)
72 str(flights_clean) #327,346 x 19 --> stimmt
73
```

JFK- Daten von den beiden anderen
Flughäfen trennen (non_jfk)

kontrollieren

```
Console Terminal x Jobs x
~/
> str(jfk) #109.079
tibble [109,079 x 1] (S3: tbl_df/tbl/data.frame)
 $ arr_delay: num [1:109079] 33 -18 -8 -2 -3 7 -4 -8 14 4 ...
> # nicht jfk
> non_jfk <- flights_clean %>% filter(origin != "JFK") %>% select(arr_delay)
> str(non_jfk) #218.267
tibble [218,267 x 1] (S3: tbl_df/tbl/data.frame)
```

```
QM6c QM7 QM8.R x flights x Funktionen.R x jfk x
Source on Save Run Source
95 |
96 #####
97 ### H1: Die Flüge am JFK haben weniger Verspätung (Ankunft)
98 ### H0: Die Flüge am JFK haben nicht weniger Verspätung (Ankunft)
99
100
101 # ein JFK sample ziehen
102 jfk_sample<- sample(jfk$arr_delay, size = 10000)
103 x_<-mean(jfk_sample) #an dieser Stelle laesst sich nur der
104 # mean des samples berechnen
105 x_
106
107 non_jfk_sample<- sample(non_jfk$arr_delay, size = 10000)
108 mu<-mean(non_jfk_sample) #an dieser Stelle laesstt sich nur der
109 # mean des samples berechnen
110 mu
111
112
113
114
95:1 # QM8 R Script
Console Terminal x Jobs x
~/
> x_
[1] 6.0769
> mu
[1] 7.2063
```

Ein sample aus jfk ziehen

Ein sample aus non_jfk ziehen

```
QM6c QM7 QM8.R x flights x Funktionen.R x jfk x
Source on Save Run Source
115 #aus diesem sample resampeln um die Breite des Konf-intervalls zu schÄ
116 set.seed(2021)
117
118 # resampeln mit der H_0: non-jfk
119 non_jfk_resample_loop <- do (10000)*mean(resample(non_jfk_sample)) #fo
120 mean(non_jfk_resample_loop$mean) #mean der means (grand mean)
121 sd(non_jfk_resample_loop$mean) #Standardfehler
122
123 hist(non_jfk_resample_loop$mean, breaks=50) #Histogramm
124 plot(density(non_jfk_resample_loop$mean,bw=1)) #GlÄttung Dichteplot
125
126 mu<-mean(non_jfk$arr_delay);mu # gesamte JFK
127 abline(v = mean(non_jfk$arr_delay), col = 4) #mean alle JFK
128
129 m_<-mean(non_jfk_resample_loop$mean);m_ # MW der Mittelwerte der samp1
130 abline(v = mean(non_jfk_resample_loop$mean), col = 2) #mean Stichprobe
131
```

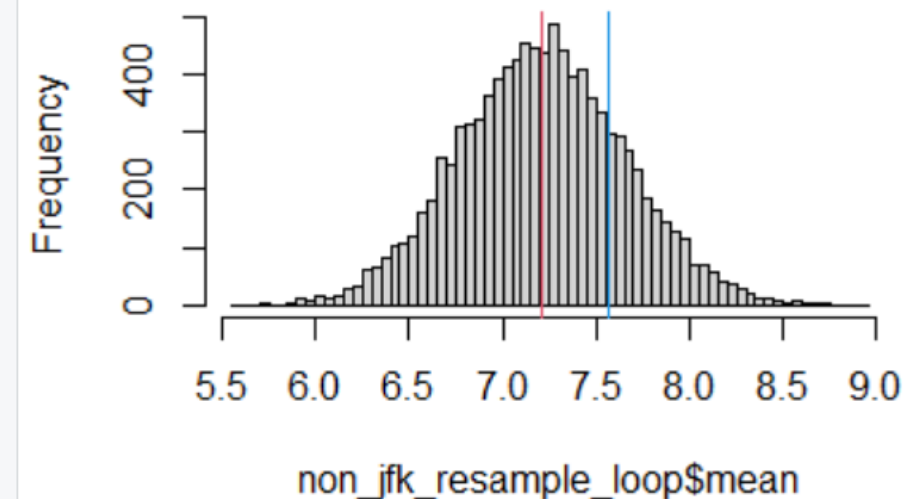
```
Console Terminal x Jobs x
~/
> abline(v = mean(non_jfk$arr_delay), col = 4) #mean origins
> mu<-mean(non_jfk$arr_delay);mu # Mittelwert aller origins
[1] 7.566989
> hist(non_jfk_resample_loop$mean, breaks=50) #Histogramm
> m_<-mean(non_jfk_resample_loop$mean);m_ # MW der Mittelwerte der samples
[1] 7.207149
> abline(v = mean(non_jfk_resample_loop$mean), col = 2) #mean StichprobenMW
```

Environment	History	Connections	Tutorial
R Global Environment			
m_	7.20714884		
mu	7.5669890546899		
mu2	7.20714884		
non_jfk_sam...	num [1:10000] 4 15 -32 -25 -12 ...		

Files Plots Packages Help Viewer

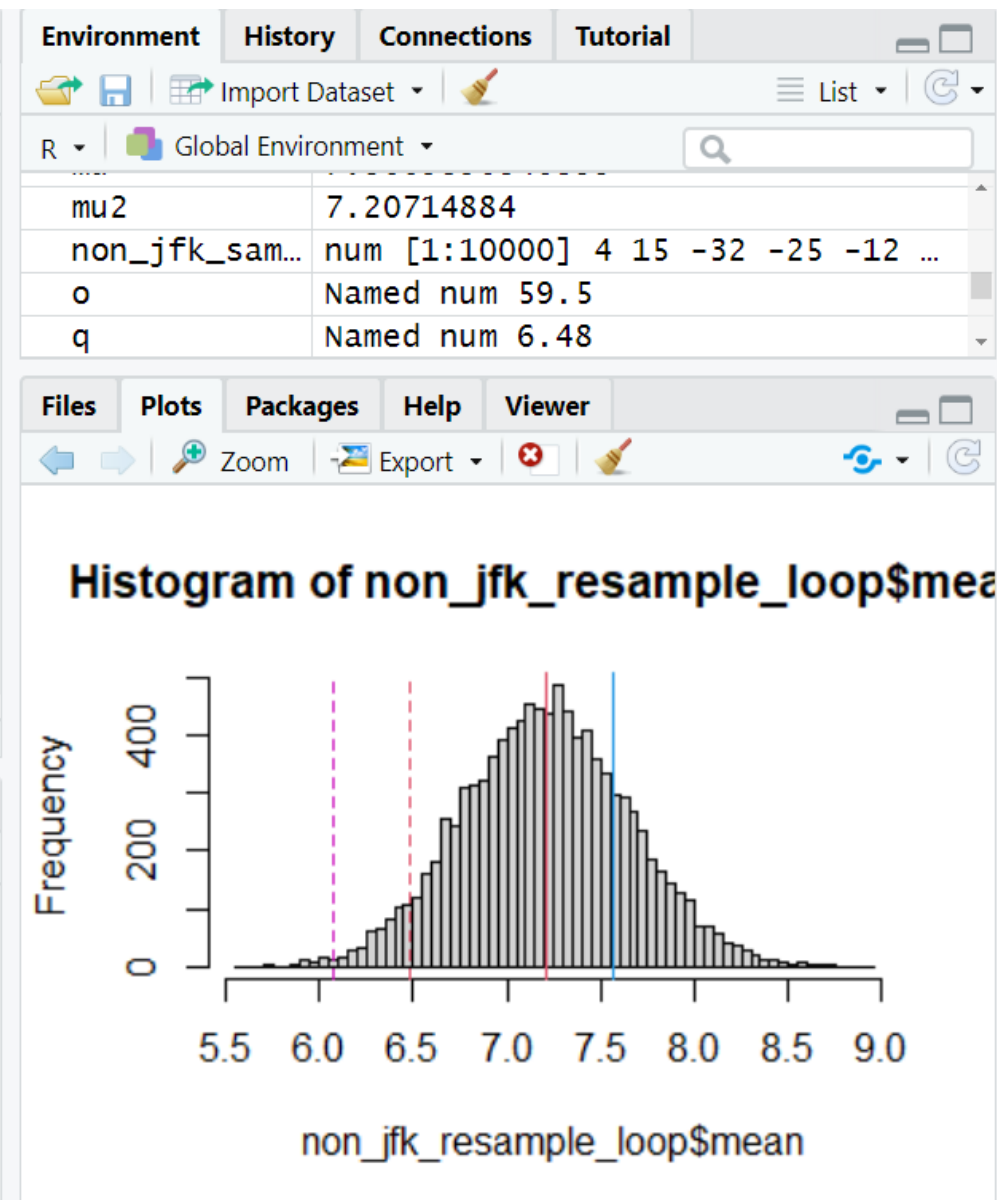
Zoom Export

Histogram of non_jfk_resample_loop\$mean



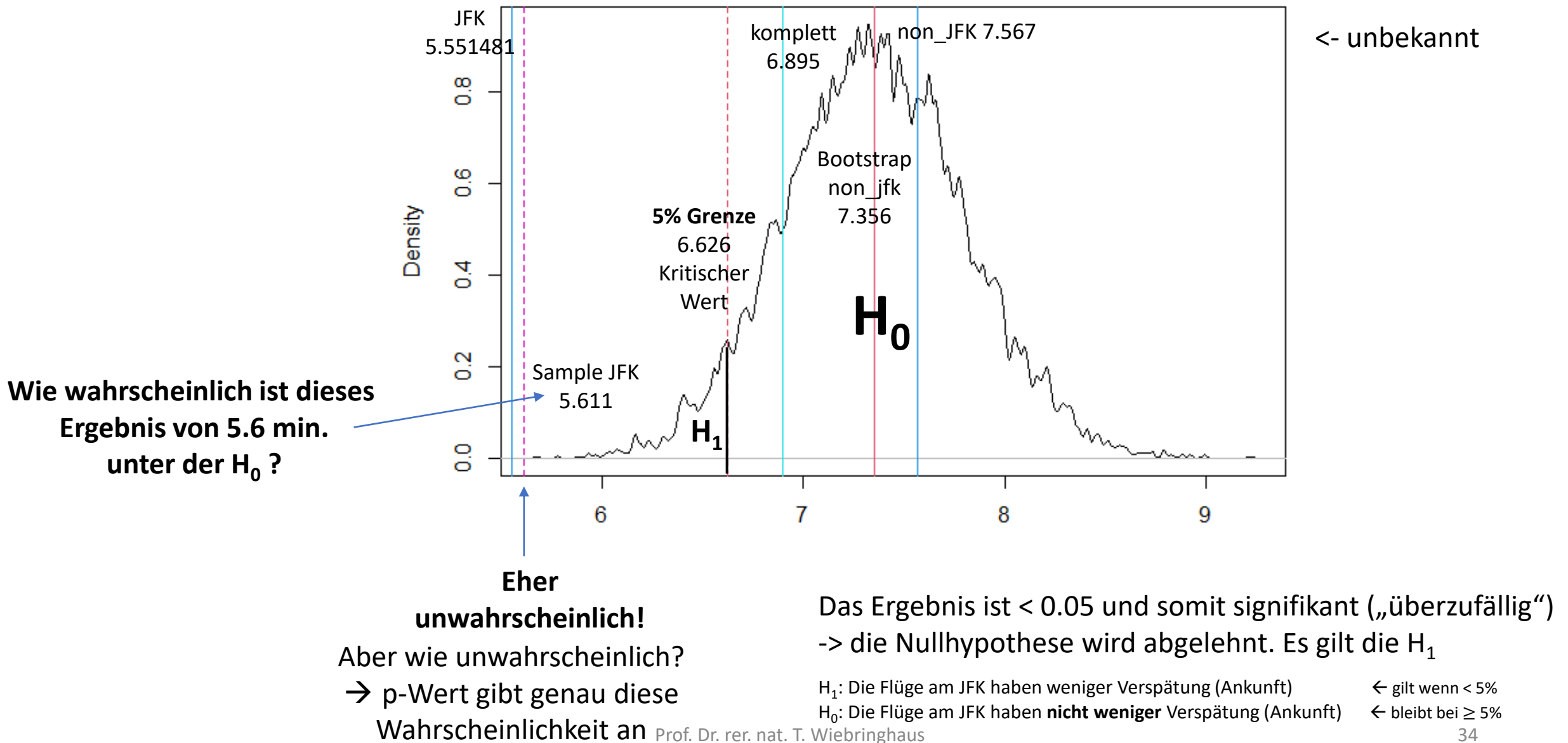

```
QM6c QM7 QM8.R* x flights x Funktionen.R x jfk x
Source on Save Run Source
131
132 ## 5% quantil (von links)
133
134 #H1: Die Flüge am JFK haben weniger Verspätung (Ankunft)
135 #H0: Die Flüge am JFK haben nicht weniger Verspätung (Ankunft)
136
137 q<-quantile(non_jfk_resample_loop$mean, prob=c(0.05));q
138 abline(v = q, col = 2, lty = 2)
139
140 abline(v = x_, col = 6, lty = 2)
141
142 x_ < q #Test: ist der mean jfk kleiner 5% der non_jfk?
143 paste("Der Stichprobenmittelwert von JFK beträgt",x_)
144 paste("Die 5% Grenze bei non_jfk liegt bei",q)
145
146 #####
146:1 # (Untitled) R Script
```

```
~/
> x_ < q #Test: ist der mean jfk kleiner 5% der non_jfk?
5%
TRUE
> paste("Der Stichprobenmittelwert von JFK beträgt",x_)
[1] "Der Stichprobenmittelwert von JFK beträgt 6.0769"
> paste("Die 5% Grenze bei non_jfk liegt bei",q)
[1] "Die 5% Grenze bei non_jfk liegt bei 6.477565"
>
```



Beispiel Hypothesentest

Frage: Sind die Ankünfte am JFK weniger verspätet als an non_JFK?



Korrigierte Stichprobenvarianz

Varianz = Streuung **um** den arithmetischen Mittelwert

Erwartungstreue
der Varianz

Jungen	x_i	$(x_i - \bar{x})$	$(x_i - \bar{x})^2$
	3,2	0,2	0,04
	3,5	0,5	0,25
	2,9	-0,1	0,01
	3,3	0,3	0,09
	3,4	0,4	0,16
	2,5	-0,5	0,25
	2,7	-0,3	0,09
	2,8	-0,2	0,04
	3,1	0,1	0,01
	2,6	-0,4	0,16
Summe	30	0	1,1

$$1,1 / 10 = 0,11$$

1,1/9 = `> var(Schule$boys)`
[1] 0.1222222

28,58/9 = `> var(Schule$girls)`
[1] 3.175556

default ist die sample variance,
nicht die population variance

`> # function for population variance`
`> var_pop <- function(x){mean((x - mean(x))^2)}`
`> var_pop(Schule$boys)`

1,1/10 = [1] 0.11

`> var_pop(Schule$girls)`
28,58/10 = [1] 2.858

Mädchen	x_i	$(x_i - \bar{x})$	$(x_i - \bar{x})^2$
	1	-2	4
	1	-2	4
	2	-1	1
	2,5	-0,5	0,25
	3,2	0,2	0,04
	2,8	-0,2	0,04
	3,5	0,5	0,25
	2	-1	1
	6	3	9
	6	3	9
Summe	30	0	28,58

$$28,58 / 10 = 2,858$$

$$\text{Sample Variance} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

← Sample Correction

$$\text{Population Variance} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

oder nur über die Nennen korrigieren mit $*(n-1)/n$

#von der built-in sample variance zur population variance:
`var_pop <- function(x){var(x)*(length(x)-1) / length(x)}`

Import „Schule“

Delimiter: tab

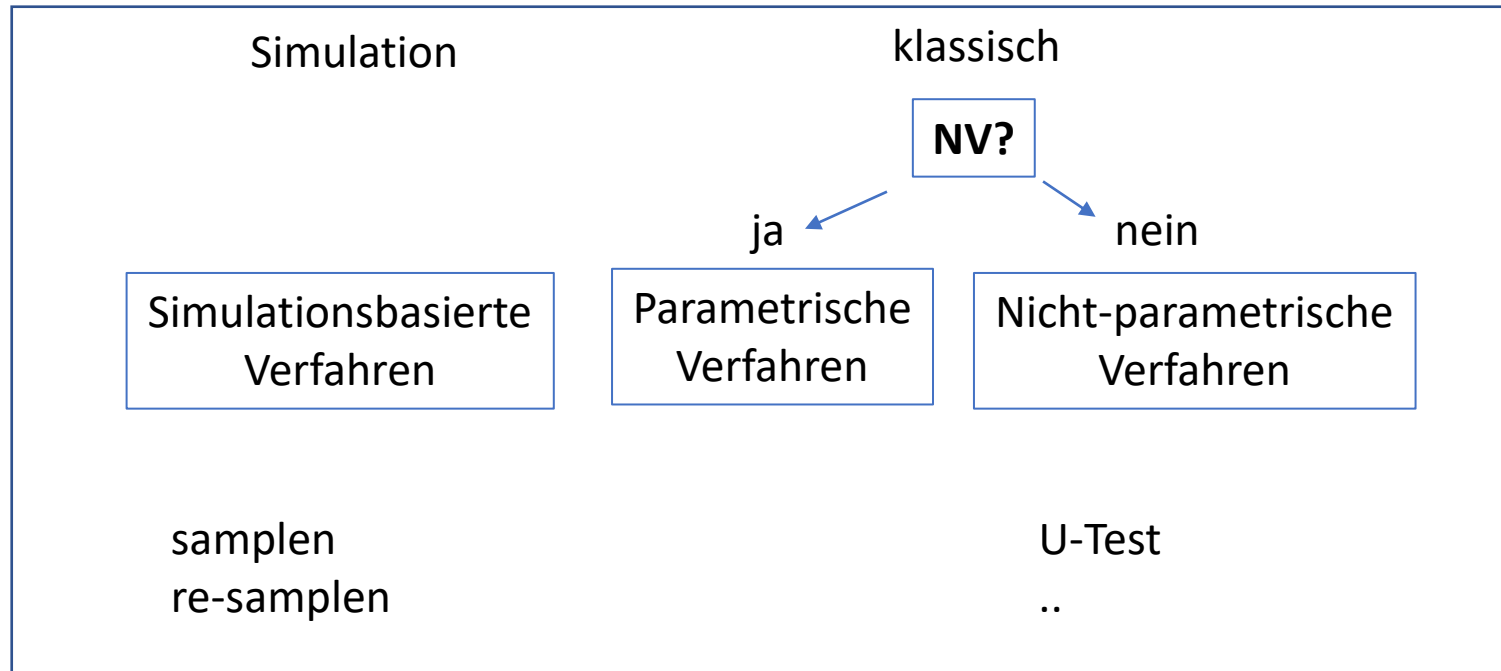
Decimal Mark: Comma

Data type: double

QM9
Testauswahl
Prüfen auf Normalverteilung
Ein Beispiel

Hypothesentest: Beispiel jfk vs non_jfk

- 1) H_1 : Die Flüge am JFK haben weniger Verspätung (Ankunft)
 H_0 : Die Flüge am JFK haben **nicht weniger** Verspätung (Ankunft)
- 2) Stichproben an den Flughäfen vornehmen



Übersicht Hypothesentest

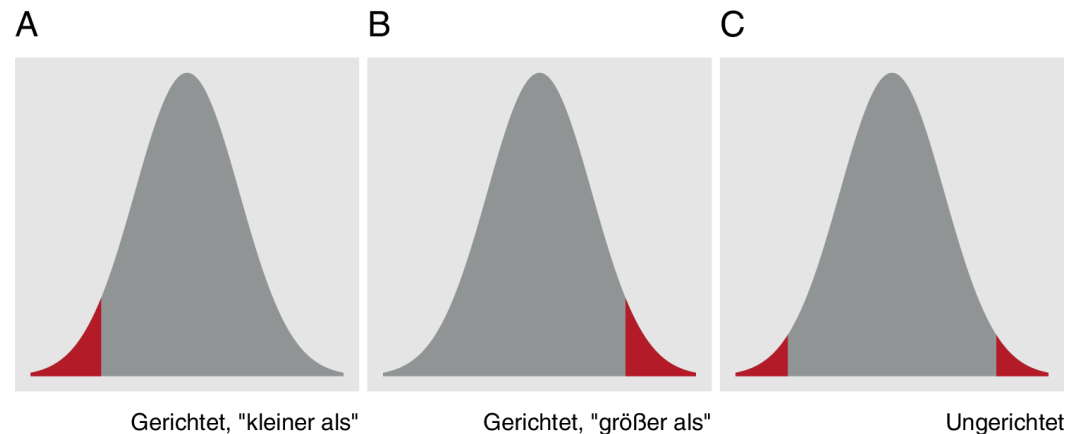
- 1) Hypothesenpaar aufstellen
- 2) Daten generieren
- 3) Explorative Analyse: z.B. Daten normalverteilt?
- 4) Testauswahl und -durchführung

3) Prüfen ob die Daten normalverteilt sind: **QQ-plot und Shapiro-Wilk Test**

4) ..

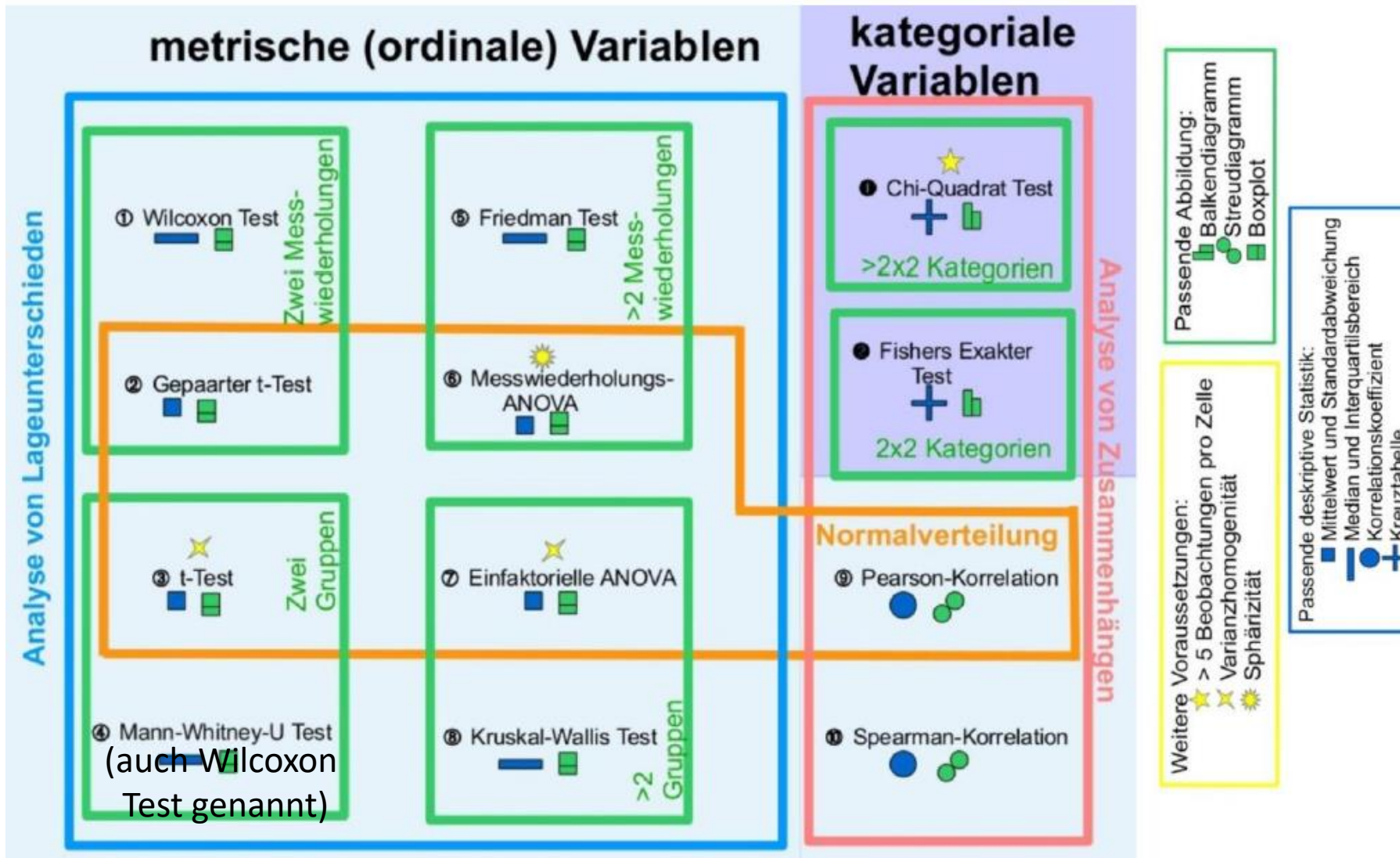
Testauswahl

- Wenn Daten einer Normalverteilung (NV) folgen, wendet man in der Praxis sog. **parametrische** Hypothesentests an
- Parametrisch weil die NV durch 2 Parameter (Mittelwert und Standardabweichung) bestimmt wird
- Eine weit verbreitete Klasse von parametrischen Tests sind t-Tests (bei z.B. höchstens 2 Gruppen)
- Wir unterscheiden allgemein **einseitige** (gerichtete) von **zweiseitigen** (ungerichteten) Tests,



- sowie **abhängige** (vorher-nachher, dieselbe Gruppe) von **unabhängigen** (2 unterschiedliche Gruppen, z.B. geimpft vs. ungeimpft) Tests
- Die Kenntnis ob Daten normalverteilt sind, ist wichtig für die richtige Testauswahl (siehe Schaubild nächste Folie)

Beispiele: Parametrische (NV) und nicht-parametrische Testverfahren



QQ-plot prüft visuell auf Normalverteilung

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for file operations and a search bar. The source editor on the left contains the following R code:

```
1 library(mosaic)
2 library(tidyverse)
3
4 # Demonstration: Testen auf Normalverteilung
5 #1) Visuelles Testen auf NV mit einem QQ-plot
6 # ?qqplot
7 # Normalverteilte Daten liegen im QQ-plot auf der Diagonalen
8
9 #?rnorm
10 nv<-rnorm(5000,0,1) #erzeuge normalverteilten Daten
11 hist(nv) #Gaussische Glockenkurve bei normalverteilten Daten
12 qqnorm(nv) # Normalverteilte Daten liegen auf der Diagonalen
13 qqline(nv) #Diagonale Linie einzeichnen
14
15 # Gegenbeispiel: gleichverteilte Daten (z.B. beim würfeln)
16 #?runif
17 uni<-runif(5000,0,1) #erzeuge gleichverteilten Daten
18 hist(uni) #gleichverteilte Daten haben dieselben Wahrscheinlichkeiten
19 qqnorm(uni) #nicht-gleichverteilte Daten liegen nicht auf der Diagonale
20 qqline(uni) #Diagonale Linie einzeichnen
21
22
```

The bottom status bar shows the time as 11:50, the current location as (Top Level), and the file type as R Script.

The right pane displays the R documentation for `qqnorm` and `qqplot`. The tabs at the top of the right pane are Environment, History, Connections, Tutorial, Files, Plots, Packages, Help, and Viewer. The Help tab is selected, showing the documentation for `R: Quantile-Quantile Plots`. The documentation includes the following sections:

- Files**: `qqnorm {stats}`
- Plots**: `qqnorm {stats}`
- Packages**: `stats`
- Help**: `qqnorm {stats}`
- Viewer**: `qqnorm {stats}`

The documentation text is as follows:

Quantile-Quantile Plots

Description

`qqnorm` is a generic function the default method of which produces a normal QQ plot of the values in `y`. `qqline` adds a line to a "theoretical", by default normal, quantile-quantile plot which passes through the `probs` quantiles, by default the first and third quartiles.

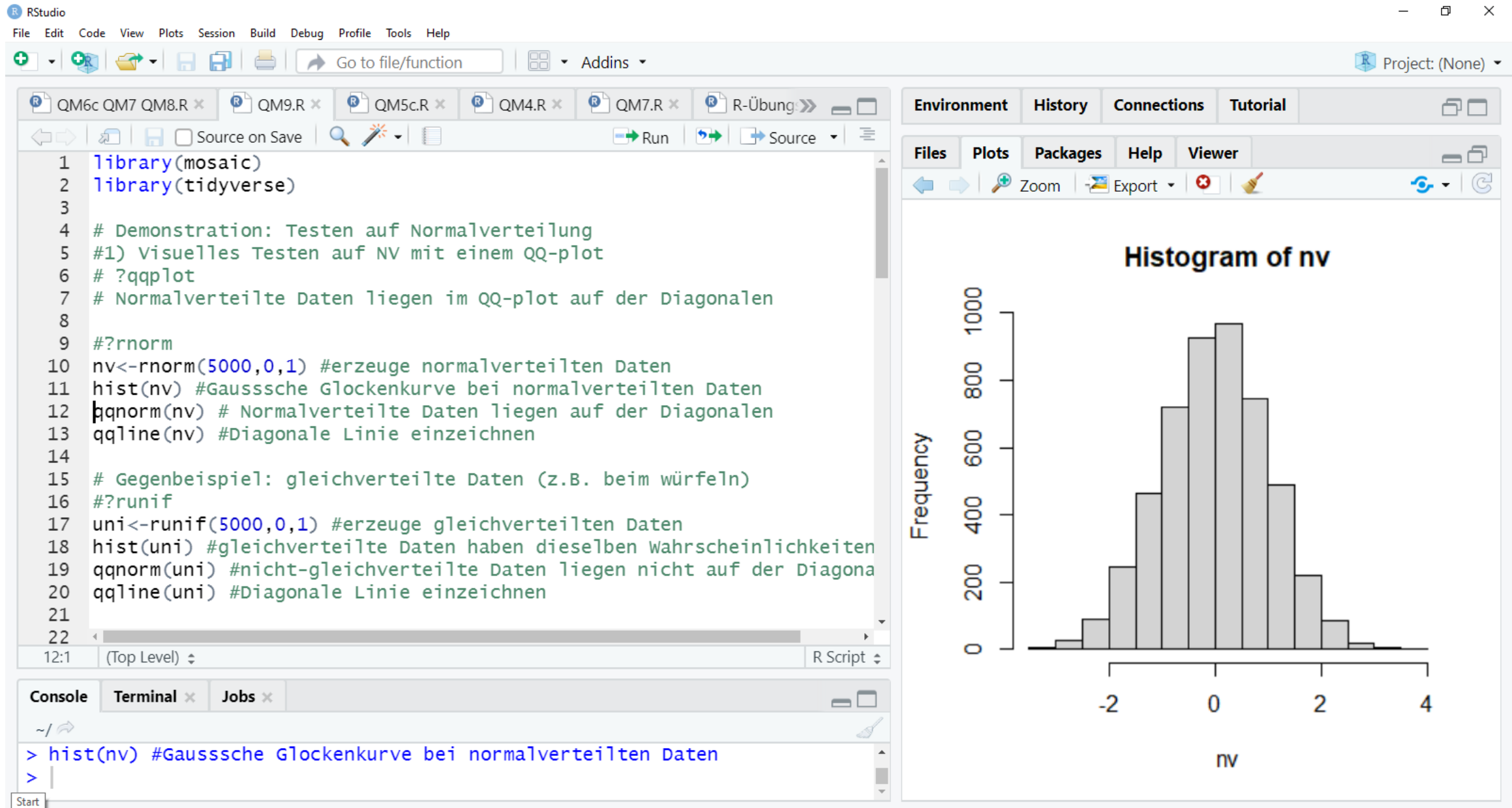
`qqplot` produces a QQ plot of two datasets.

Graphical parameters may be given as arguments to `qqnorm`, `qqplot` and `qqline`.

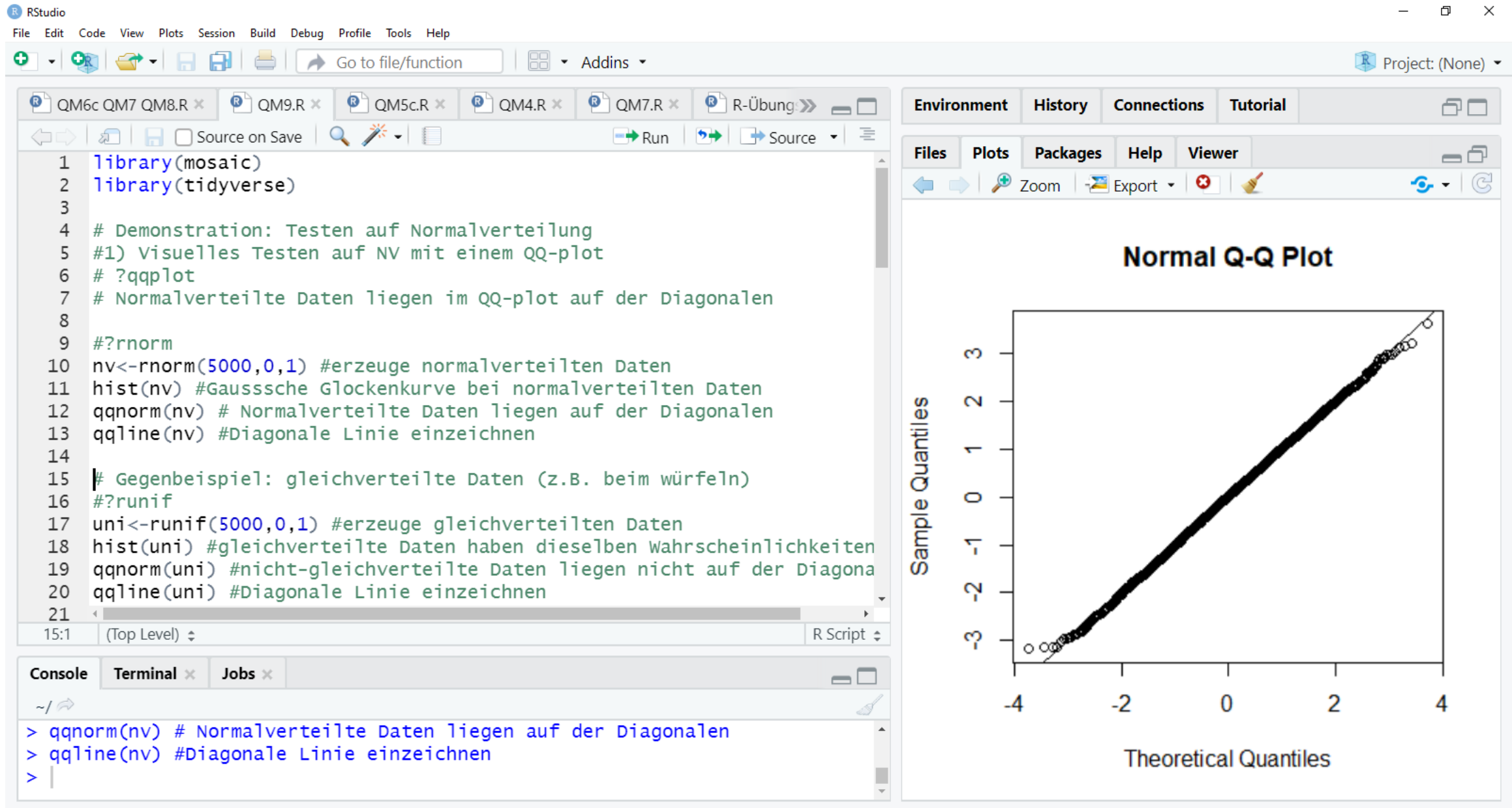
Usage

```
qqnorm(y, ...)
## Default S3 method:
qqnorm(y, ylim, main = "Normal Q-Q Plot",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
       plot.it = TRUE, datax = FALSE, ...)
```

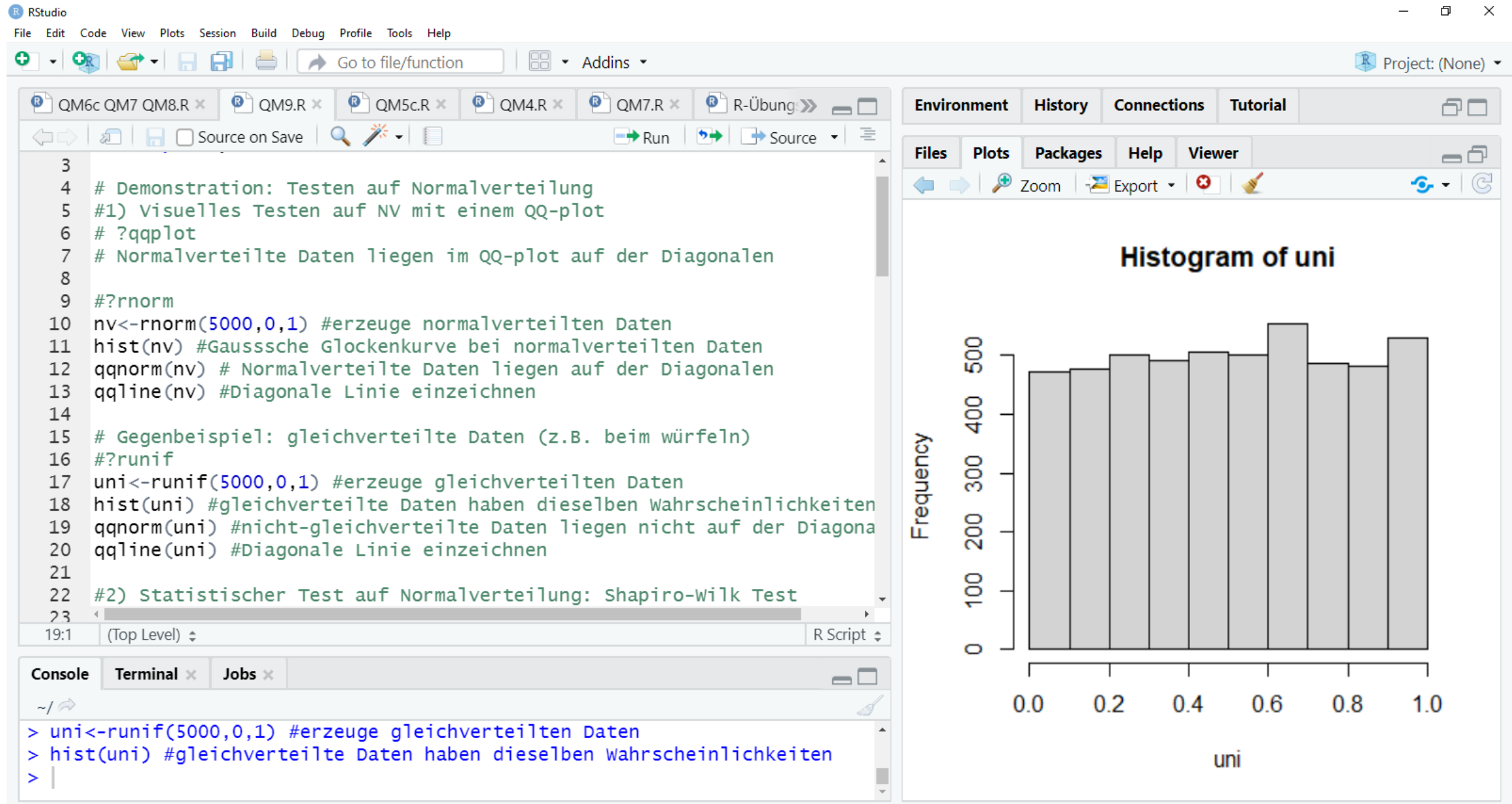
Beispiel: Gausssche Glockenkurve bei normalverteilten Daten



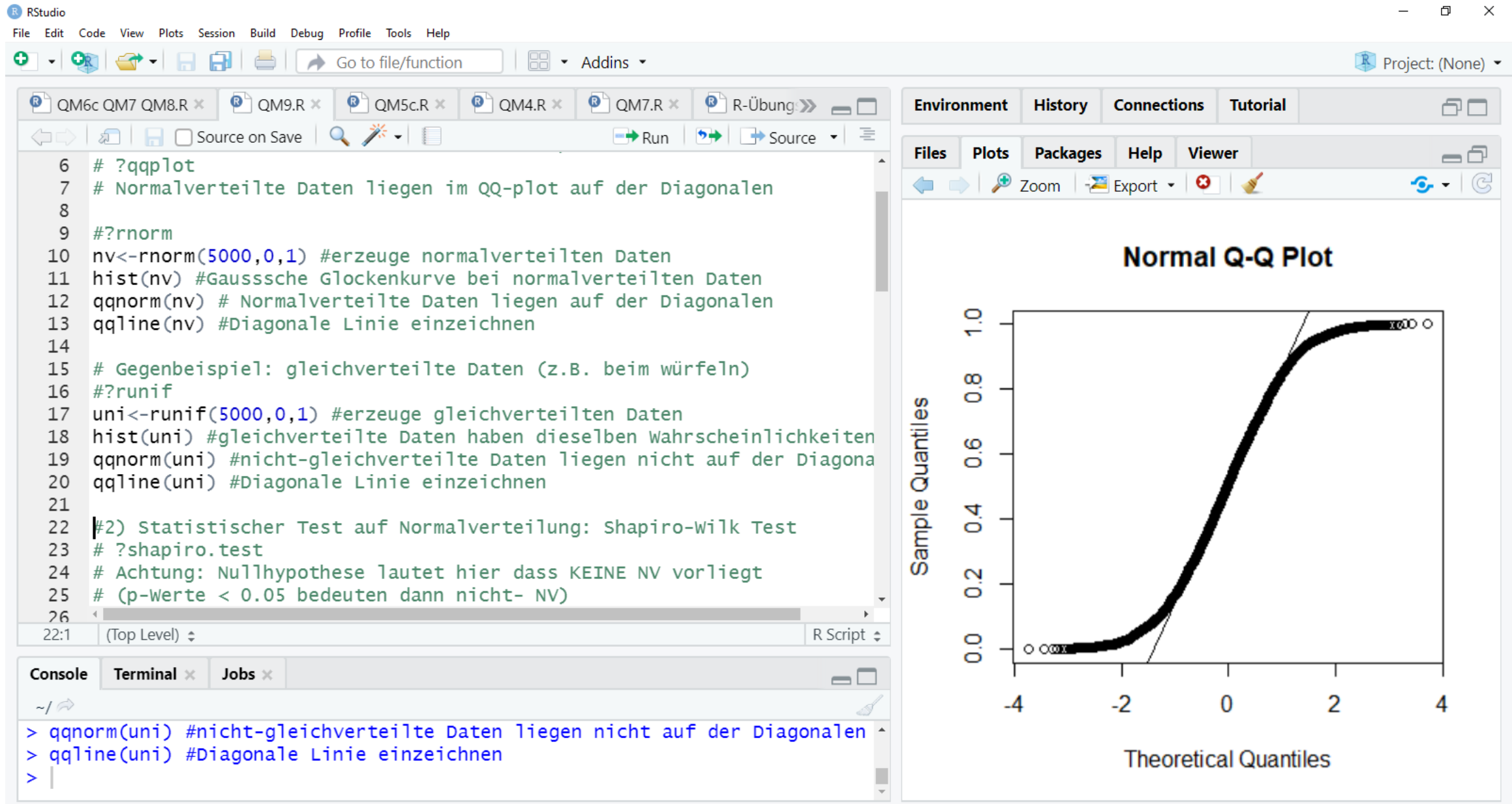
Beispiel: Diagonale im QQ-plot bei normalverteilten Daten



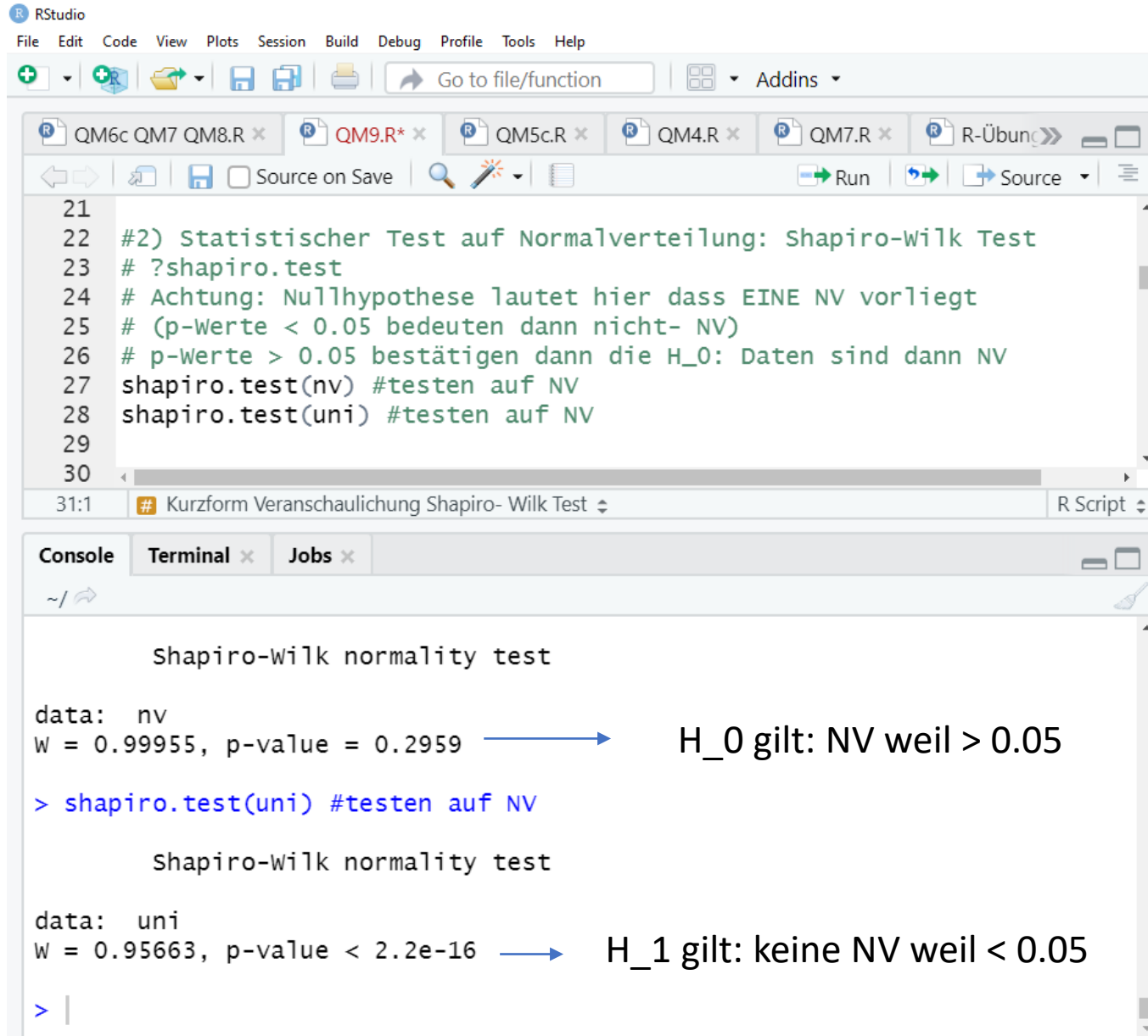
Beispiel: keine Gaussische Glockenkurve bei nicht-normalverteilten Daten



Beispiel: keine Diagonale im QQ-plot bei nicht-normalverteilten Daten



Beispiel: Shapiro- Wilk Test testet auf Normalverteilung



```
21
22 #2) Statistischer Test auf Normalverteilung: Shapiro-Wilk Test
23 # ?shapiro.test
24 # Achtung: Nullhypothese lautet hier dass EINE NV vorliegt
25 # (p-Werte < 0.05 bedeuten dann nicht- NV)
26 # p-Werte > 0.05 bestätigen dann die H_0: Daten sind dann NV
27 shapiro.test(nv) #testen auf NV
28 shapiro.test(uni) #testen auf NV
29
30
```

```
31:1 # Kurzform Veranschaulichung Shapiro- Wilk Test
```

Console

```
Shapiro-Wilk normality test
data:  nv
W = 0.99955, p-value = 0.2959 → H_0 gilt: NV weil > 0.05
> shapiro.test(uni) #testen auf NV
Shapiro-Wilk normality test
data:  uni
W = 0.95663, p-value < 2.2e-16 → H_1 gilt: keine NV weil < 0.05
> |
```

Shapiro- Wilk Test

Nullhypothese: Daten **sind** NV ($p > 0.05$)

Alternativhypothese: Daten **sind nicht** NV ($p > 0.05$)

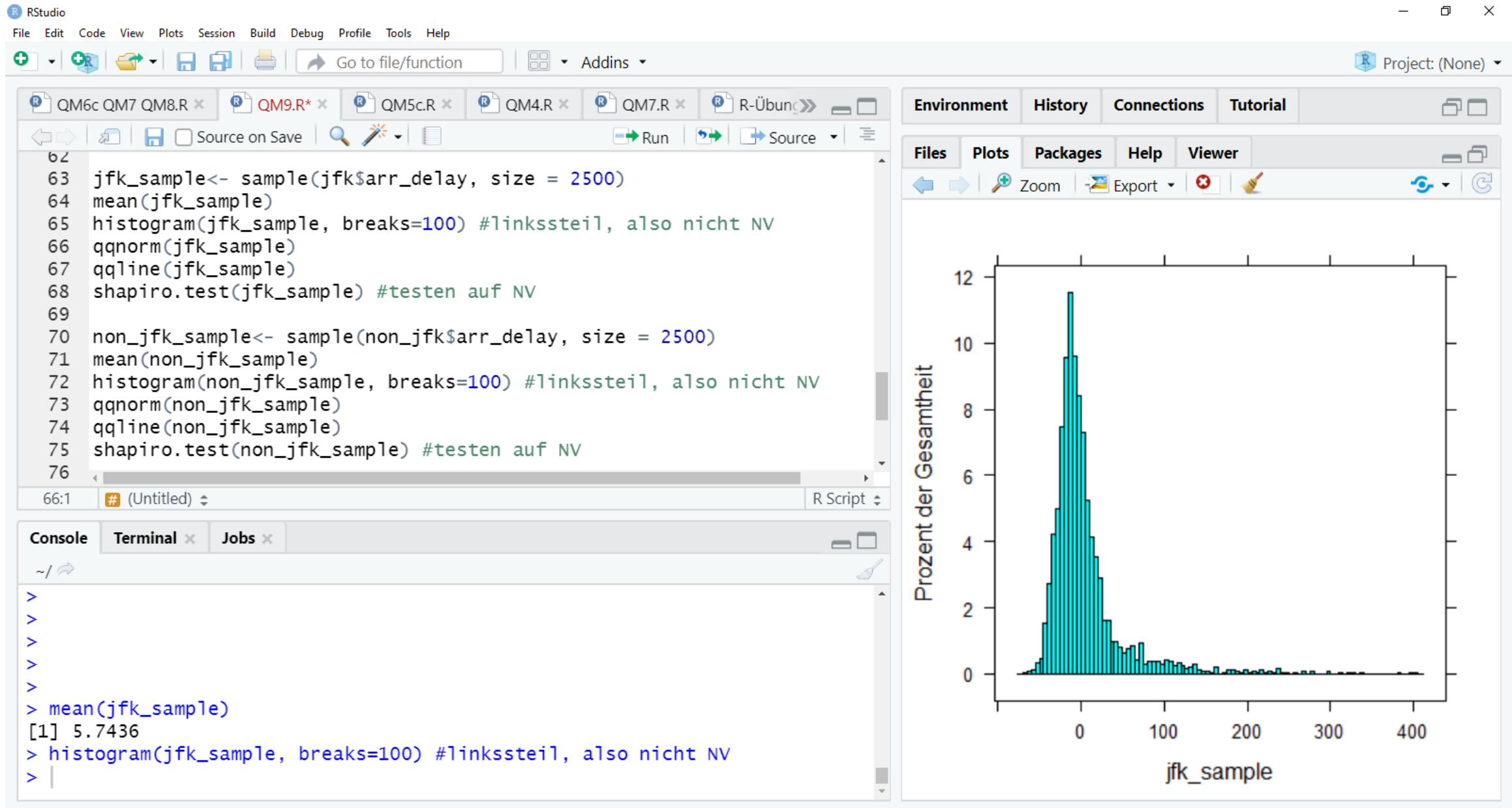
Beispiel: jfk und non_jfk auf Normalverteilung prüfen

The screenshot shows the RStudio interface with the following components:

- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations, a search bar with "Go to file/function", and a "Run" button.
- Script Editor:** Contains R code for installing and analyzing flight data. The code is as follows:

```
38  
39 # installieren und importieren von nycflights13  
40 install.packages("nycflights13") #installieren  
41 library(nycflights13) #importieren  
42 View(flights)  
43 #?flights  
44 # zunächst bereinigen von NA und selektieren nur arr_delay  
45 flights_clean <- drop_na(flights, arr_delay) #ohne NA  
46  
47 # arr_delay filtern nach 'JFK' und 'non_JFK'  
48 # nur jfk  
49 jfk <- flights_clean %>% filter(origin == "JFK") %>% select(arr_delay  
50 str(jfk) #109.079  
51  
52 # nicht jfk  
53 non_jfk <- flights_clean %>% filter(origin != "JFK") %>% select(arr_d  
54 str(non_jfk) #218.267  
55 # 109.079 + 218.267 = 327.346  
56  
57 # Anzahl prüfen mit Population (ohne NA)  
58 str(flights_clean) #327,346 x 19 --> stimmt  
59  
60 favstats(jfk$arr_delay)  
61 favstats(non_jfk$arr_delay)  
62
```
- Environment Panel:** Shows "Project: (None)".
- Plots Panel:** Includes tabs for Files, Plots, Packages, Help, and Viewer. The Plots tab is active, showing a zoom and export toolbar.
- Console:** Displays the command "# Kurzform Veranschaulichung Shapiro- Wilk Test" at line 31:1.

Beispiel: jfk ist linkssteil



Beispiel: jfk ist eher nicht NV

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

QM6c QM7 QM8.R QM9.R* QM5c.R QM4.R QM7.R R-Übung

```
62
63 jfk_sample<- sample(jfk$arr_delay, size = 2500)
64 mean(jfk_sample)
65 histogram(jfk_sample, breaks=100) #linkssteil, also nicht NV
66 qqnorm(jfk_sample)
67 qqline(jfk_sample)
68 shapiro.test(jfk_sample) #testen auf NV
69
70 non_jfk_sample<- sample(non_jfk$arr_delay, size = 2500)
71 mean(non_jfk_sample)
72 histogram(non_jfk_sample, breaks=100) #linkssteil, also nicht NV
73 qqnorm(non_jfk_sample)
74 qqline(non_jfk_sample)
75 shapiro.test(non_jfk_sample) #testen auf NV
76
```

68:1 # (Untitled) R Script

Console Terminal Jobs

```
>
>
>
> mean(jfk_sample)
[1] 5.7436
> histogram(jfk_sample, breaks=100) #linkssteil, also nicht NV
> qqnorm(jfk_sample)
> qqline(jfk_sample)
>
```

Environment History Connections Tutorial

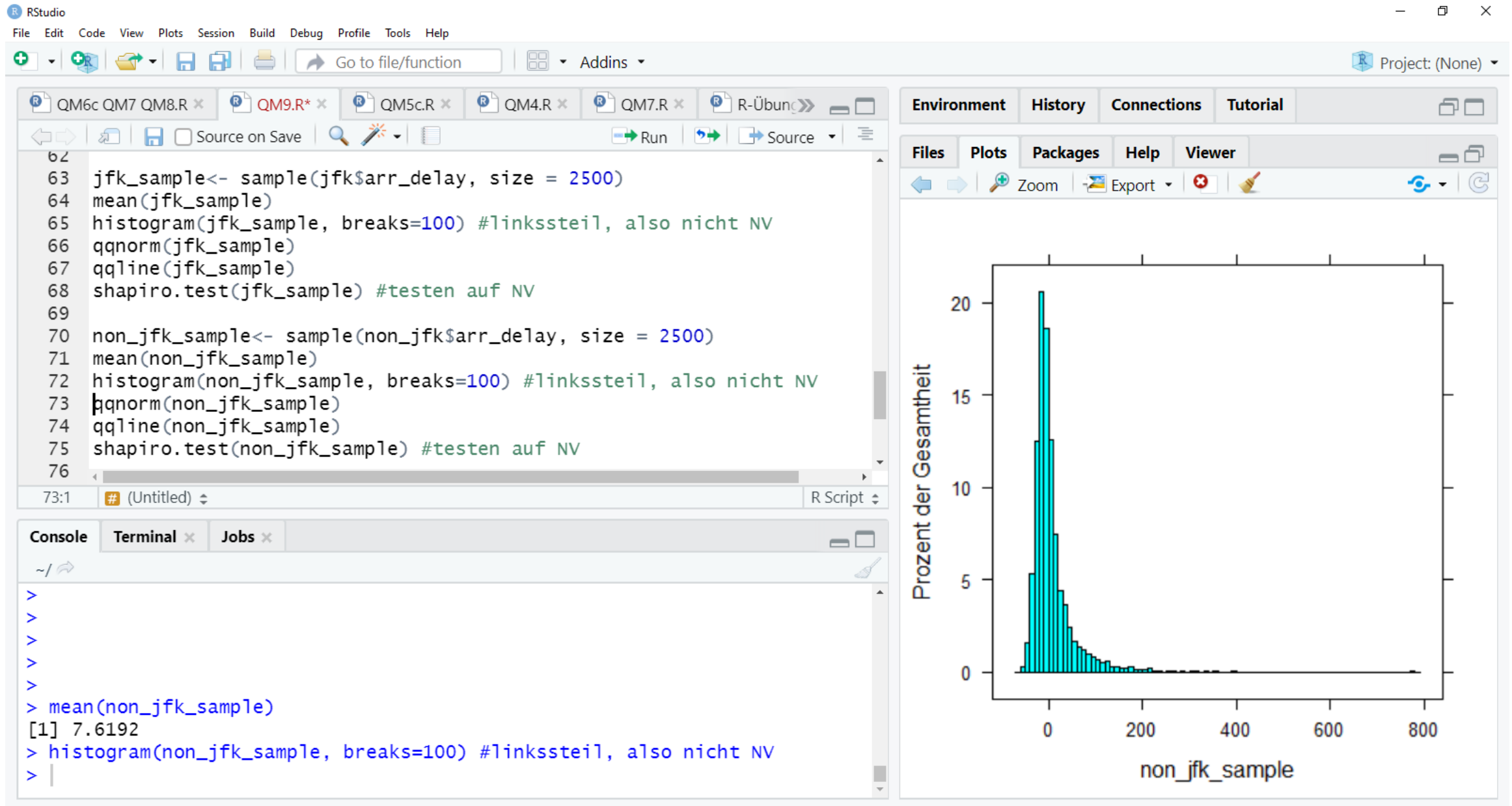
Files Plots Packages Help Viewer

Zoom Export

Normal Q-Q Plot

The plot displays Sample Quantiles on the y-axis (ranging from 0 to 400) against Theoretical Quantiles on the x-axis (ranging from -3 to 3). A solid diagonal line represents the expected normal distribution. The data points, shown as open circles, follow the line initially but then curve sharply upwards for theoretical quantiles greater than 1, indicating a right-skewed distribution.

Beispiel: non-jfk ist linkssteil



Beispiel: non_jfk ist eher nicht NV

RStudio interface showing R code and a Normal Q-Q Plot.

Code Editor (R Script):

```
62  
63 jfk_sample<- sample(jfk$arr_delay, size = 2500)  
64 mean(jfk_sample)  
65 histogram(jfk_sample, breaks=100) #linkssteil, also nicht NV  
66 qqnorm(jfk_sample)  
67 qqline(jfk_sample)  
68 shapiro.test(jfk_sample) #testen auf NV  
69  
70 non_jfk_sample<- sample(non_jfk$arr_delay, size = 2500)  
71 mean(non_jfk_sample)  
72 histogram(non_jfk_sample, breaks=100) #linkssteil, also nicht NV  
73 qqnorm(non_jfk_sample)  
74 qqline(non_jfk_sample)  
75 shapiro.test(non_jfk_sample) #testen auf NV  
76
```

Console:

```
>  
>  
>  
> mean(non_jfk_sample)  
[1] 7.6192  
> histogram(non_jfk_sample, breaks=100) #linkssteil, also nicht NV  
> qqnorm(non_jfk_sample)  
> qqline(non_jfk_sample)  
>
```

Normal Q-Q Plot:

The plot displays Sample Quantiles on the y-axis (ranging from 0 to 800) and Theoretical Quantiles on the x-axis (ranging from -3 to 3). The data points follow a straight line for negative theoretical quantiles but curve sharply upwards for positive theoretical quantiles, indicating a right-skewed distribution.

Beispiel: Shapiro- Wilk Test lehnt NV ab

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

Environment History Connections Tutorial

Files Plots Packages Help Viewer

Zoom Export

```
77 shapiro.test(jfk_sample) #testen auf NV
78 shapiro.test(non_jfk_sample) #testen auf NV
79
80 #FAZIT: beide Datensätze sind nicht normalverteilt.
81 # Es können deshalb keine parametrischen Verfahren angewandt werden.
82 # Es liegen 2 unabhängige Gruppen vor (kein vorher-nachher)
83
84
```

92:1 # (Untitled) R Script

Console Terminal Jobs

```
>
> shapiro.test(jfk_sample) #testen auf NV

Shapiro-Wilk normality test

data:  jfk_sample
W = 0.68492, p-value < 2.2e-16 → H_1 gilt: keine NV weil < 0.05

> shapiro.test(non_jfk_sample) #testen auf NV

Shapiro-Wilk normality test

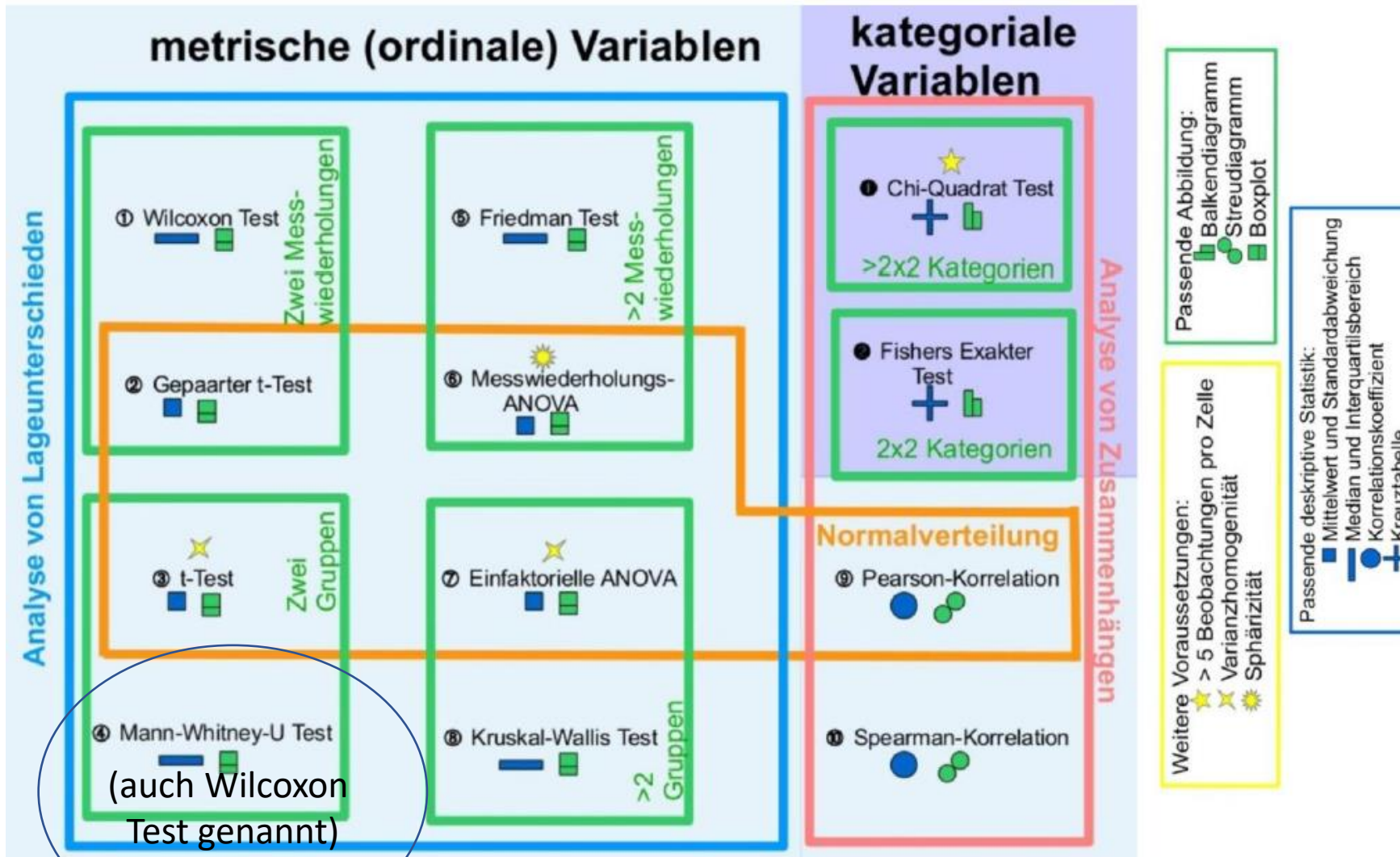
data:  non_jfk_sample
W = 0.67879, p-value < 2.2e-16 → H_1 gilt: keine NV weil < 0.05

> |
```

Normal Q-Q Plot

The plot displays Sample Quantiles on the y-axis (ranging from 0 to 800) and Theoretical Quantiles on the x-axis (ranging from -3 to 3). A solid diagonal line represents the expected normal distribution. Data points are plotted as open circles. Most points follow the line, but there is a distinct upward deviation for theoretical quantiles greater than 1, with one point at approximately (3.5, 800) being a significant outlier above the line.

Hilfestellung: Parametrische (NV) und nicht-parametrische Testverfahren



Ablauf Hypothesentest

- 1) Hypothesenpaar aufstellen
- 2) Daten generieren
- 3) Explorative Analyse: z.B. Daten normalverteilt?
- 4) Testauswahl und -durchführung

Hypothesentest: Beispiel jfk vs non_jfk

- 1) H_1 : Die Flüge am JFK haben weniger Verspätung (Ankunft)
 H_0 : Die Flüge am JFK haben **nicht weniger** Verspätung (Ankunft)

$$H_1: \mu_{\text{JFK}} < \mu_{\text{non_JFK}}$$

$$H_0: \mu_{\text{JFK}} \geq \mu_{\text{non_JFK}}$$

- 2) Stichproben an den Flughäfen vornehmen
- 3) Prüfen ob die Daten normalverteilt sind: QQ-plot und Shapiro-Wilk Test
- 4) **2 unabhängige nicht-normalverteilte Gruppen:**
Mann-Whitney U – Test (auch Wilcoxon Test genannt)

Ablauf Hypothesentest

- 1) Hypothesenpaar aufstellen
- 2) Daten generieren
- 3) Explorative Analyse: z.B. Daten normalverteilt?
- 4) Testentscheidung und -durchführung

Beispiel: Wilcoxon Test ist signifikant. Die Nullhypothese (nicht weniger Verspätung) wird abgelehnt

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

QM6c QM7 QM8.R QM9.R* QM5c.R QM4.R QM7.R R-Übung

Source on Save Run Source

```
81 # Es können deshalb keine parametrischen Verfahren angewandt werden.
82 # Es liegen 2 unabhängige Gruppen vor (kein vorher-nachher)
83
84 # Laut Diagramm 'Testauswahl' kommt der Mann-Whitney U-Test
85 # (auch Wilcoxon Test) zum Einsatz
86 #?wilcox.test
87 wilcox.test(jfk_sample,non_jfk_sample)
88
89
90
91
92
93
```

88:1 # (Untitled) R Script

Console Terminal Jobs

```
> #?wilcox.test
> wilcox.test(jfk_sample,non_jfk_sample)

wilcoxon rank sum test with continuity correction

data: jfk_sample and non_jfk_sample
W = 3006925, p-value = 0.02068
alternative hypothesis: true location shift is not equal to 0

> |
```

< 0.05 signifikant

Nullhypothese wird abgelehnt

Normal Q-Q Plot

The figure is a Normal Q-Q Plot. The y-axis is labeled 'Sample Quantiles' and ranges from 0 to 800. The x-axis is labeled 'Theoretical Quantiles' and ranges from -3 to 3. The plot shows a series of open circles representing the data points. A solid diagonal line represents the expected normal distribution. The points closely follow the line, suggesting the data is approximately normally distributed.

Ende