Testat 2

Gruppe 1: Jannis Seefeld, Jan Hofmann, Rabea Götz

Zentrale Grenzwertsatz

Für jedes $n = 1, 2, \ldots$ seien die Zufallsvariablen $X_1, X_2, \ldots, X_n$ unabhängig und besitzen die gleiche Verteilung mit dem Erwartungswert $\mu = E(X_i)$ und der Varianz $\sigma^2 = Var(X_i) =$. Dann gilt für die Verteilungsfunktion der standardisierten Summen $G_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma}$:

$$\lim_{n\to\infty} P\left(G_n \leq x\right) = \Phi(x)$$

Dabei ist Φ die Verteilungsfunktion der Standard-Normalverteilung $N(0, 1)$.

Überprüfen Sie mit R den zentralen Grenzwertsatz.

Verteilung

Erzeugen Sie einen Data Frame, der die gezogenen Zufallszahlen G_n enthält. Die Stichprobe soll einen Umfang von 50000 haben (Anzahl Zeilen der Tabelle). Der Data Frame hat sechs Spalten, die die Werte für die Zahlen $n = 1, 2, 3, 10, 100, 1000$ enthalten. Geben Sie die ersten 10 Zeilen des Data Frames aus.

Welche Verteilung Sie für X_i nehmen, bleibt Ihnen überlassen. Es soll nur keine Normalverteilung sein.

```
# Ihre Lösung:
set.seed(42)
N = c(1, 2, 3, 10, 100, 1000)
columns = list()

for (i in 1:length(N)) {
  n = N[i]
  l = sapply(1:n, function(x) {
    rchisq(n=50000, df = n)
  })
  mu = n
  sigma = sqrt(2 * n)
  columns[[i]] = ((rowSums(l) - n * mu) / (sqrt(n) * sigma))
}
df = as.data.frame(do.call(cbind, columns))
head(df, 10)
```

```
##            V1          V2         V3          V4         V5         V6
## 1   0.3684264 -0.24885795 -0.1562018  0.73054360 -0.2804200 -0.7318885
## 2  -0.2917558 -0.77937998 -1.2150235  1.21856642 -0.7862830 -1.7313907
## 3   1.9323015 -0.52223565 -0.3454037 -0.49618362 -0.8015317 -0.1290398
## 4   1.3883890  0.50419539 -0.7234210  0.29585330  0.1013728  1.4486261
## 5  -0.4052809  1.07737195  0.5084420 -0.09659746 -0.4117895 -0.6619536
## 6  -0.4085135 -0.01862378 -0.9812555 -0.21871012  0.6052062 -0.2372406
## 7  -0.7070759 -0.47873736 -0.1903704  0.36184267 -0.2950790  0.7678741
```

```
## 8  -0.7070002 -0.12324932 -1.3122720 -1.00678020 -1.2741284 -1.3548954
## 9   1.4267777 -1.14318985 -1.0135923 -0.30245825  0.2793168 -0.3120476
## 10 -0.4215231  2.48655956  0.7710699  0.34231531 -0.5581551  0.1032683
```

Plot

Plotten Sie sechs Histogramme, die je für $n = 1, 2, 3, 10, 100, 1000$ die Verteilung im Vergleich zu einer $(0, 1)$-Normalverteilung zeigen. Die Intervallbreite soll 0,25 sein.

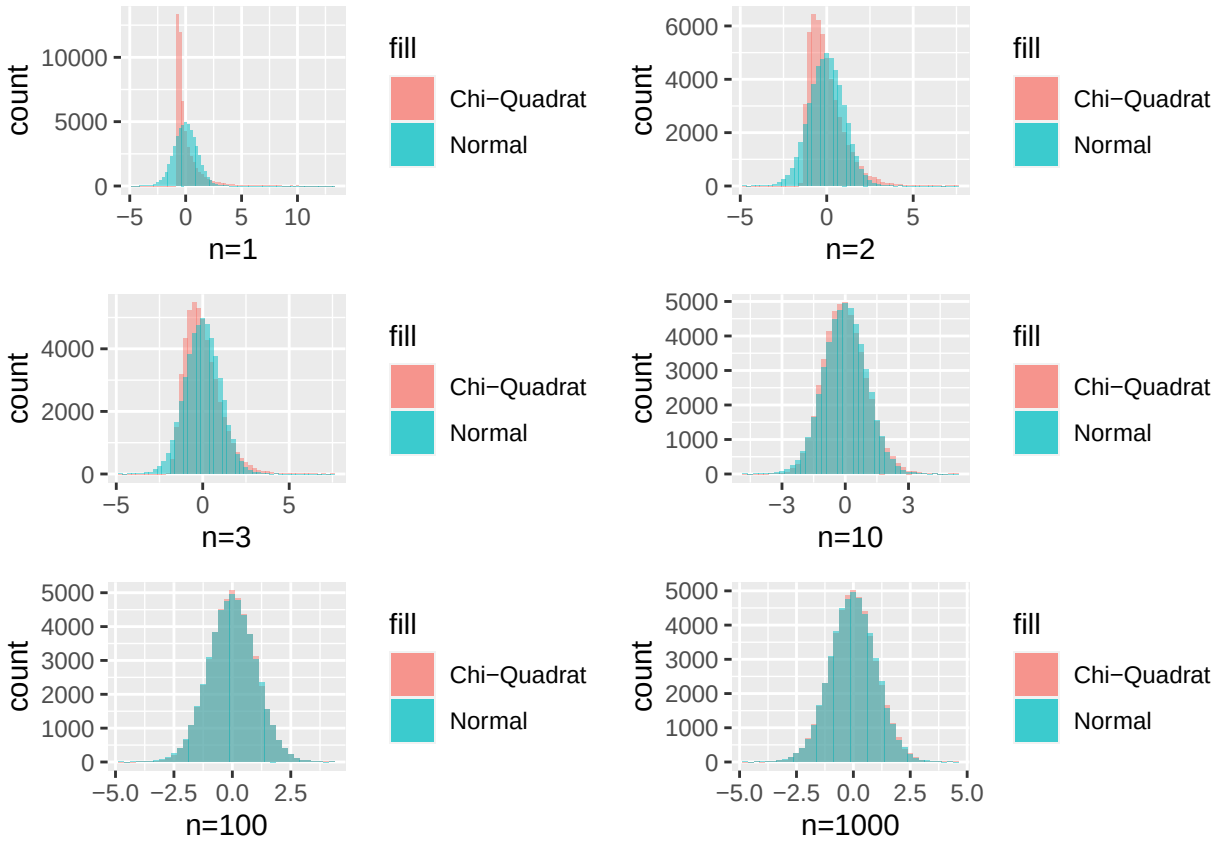
Tipp: Platzieren Sie die sechs Plots auf einem 3×2-Gitter.

```
# Ihre Lösung:
library(ggplot2)
library(gridExtra)

norm = data.frame(rnorm(n=50000))

hist1 = ggplot(df) +
  geom_histogram(aes(x = df[,1], fill='Chi-Quadrat'),  alpha = 0.5, binwidth = 0.25) +
  geom_histogram(aes(x = norm[,1], fill='Normal'),alpha = 0.5, binwidth = 0.25) +
  labs(x='n=1')
hist2 = ggplot(df) +
  geom_histogram(aes(x = df[,2], fill='Chi-Quadrat'), alpha = 0.5, binwidth = 0.25) +
  geom_histogram(aes(x = norm[,1], fill='Normal'), alpha = 0.5, binwidth = 0.25) +
   labs(x='n=2')
hist3 = ggplot(df) +
  geom_histogram(aes(x = df[,3], fill='Chi-Quadrat'), alpha = 0.5, binwidth = 0.25) +
  geom_histogram(aes(x = norm[,1], fill='Normal'), alpha = 0.5, binwidth = 0.25) +
   labs(x='n=3')
hist4 = ggplot(df) +
  geom_histogram(aes(x = df[,4], fill='Chi-Quadrat'), alpha = 0.5, binwidth = 0.25) +
  geom_histogram(aes(x = norm[,1], fill='Normal'), alpha = 0.5, binwidth = 0.25) +
   labs(x='n=10')
hist5 = ggplot(df) +
  geom_histogram(aes(x = df[,5], fill='Chi-Quadrat'), alpha = 0.5, binwidth = 0.25) +
  geom_histogram(aes(x = norm[,1], fill='Normal'), alpha = 0.5, binwidth = 0.25) +
   labs(x='n=100')
hist6 = ggplot(df) +
  geom_histogram(aes(x = df[,6], fill='Chi-Quadrat'), alpha = 0.5, binwidth = 0.25) +
  geom_histogram(aes(x = norm[,1], fill='Normal'), alpha = 0.5, binwidth = 0.25) +
   labs(x='n=1000')

grid.arrange(hist1, hist2, hist3, hist4, hist5, hist6, nrow = 3, ncol = 2)
```



Abweichung

Nun soll die Abweichung der Standardisierten G von der Normalverteilung für alle n quantifiziert werden. Hierzu soll für alle Balken (bins) der Histogramme aus der vorigen Aufgabe die Differenz von G zur Normalverteilung gebildet und quadriert werden. Diese Werte werden aufaddiert und durch die Anzahl der Intervalle geteilt. Daraus wird die Wurzel gezogen.

Sie können sich auch ein anderes Maß zur Bestimmung der Abweichung überlegen.

Geben Sie die Abweichungen aus. Stimmt es, dass die Abweichungen mit größerem n kleiner werden?

Tipp: `hist(plot = FALSE)` erzeugt ein Histogramm, ohne es zu plotten. Gerne können Sie auch Ihr eigenes Histogramm nutzen.

```
norm_bin_height = ggplot_build(hist1)$data[[2]]$count

bin_heights = list(ggplot_build(hist1)$data[[1]]$count,
  ggplot_build(hist2)$data[[1]]$count,
  ggplot_build(hist3)$data[[1]]$count,
  ggplot_build(hist4)$data[[1]]$count,
  ggplot_build(hist5)$data[[1]]$count,
  ggplot_build(hist6)$data[[1]]$count)

vars = list()

for (h in bin_heights) {
  diffs = sapply(1:length(h), function(i) {
    (norm_bin_height[i] - h[i]) ^ 2
  })
```

```
  vars = append(vars, sqrt(sum(diffs) / length(diffs)))
}

for (i in 1:6) {
  print(paste0('n=', N[i], ': ', vars[[i]]))
}
```

```
## [1] "n=1: 1590.80761775657"
## [1] "n=2: 748.006791413019"
## [1] "n=3: 426.018497251"
## [1] "n=10: 124.039922762893"
## [1] "n=100: 38.9059850881028"
## [1] "n=1000: 50.7304538454804"
```

Untersuchungen zur koronaren Herzkrankheit

In diesem Abschnitt sollen Daten von Probanden bzw. Patienten auf das Risiko für koronare Herzkrankheit untersucht werden. Dies ist eine Erkrankung der Herzkranzgefäße (Koronararterien), die sich durch Ablagerungen in den Gefäßwänden verengen. Der Original-Herz-Datensatz ist unter

- https://archive.ics.uci.edu/ml/datasets/Heart+Disease

beschrieben. Wir nutzen eine konsolidierte CSV-Datei, die bereits Header enthält. Download unter:

- https://oc.informatik.hs-mannheim.de/s/wyzFq34K9HiNjXR/download

Die Datei enthält 13 Merkmale, die einen Einfluss auf eine koronare Herzkrankheit haben können. Das 14. Merkmal `goal` (im Original auch `num`) ist die Diagnose (Klassifizierung). Der Wert ist 0, falls keine krankhafte Verengung der Gefäße vorliegt, oder 1, 2, 3 oder 4, falls – je nach Stärke – eine krankhafte Verengung der Gefäße vorliegt. Unter

- https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/heart-disease.names

finden Sie eine Beschreibung aller Attribute. Hier ist eine Zusammenfassung. Wir benötigen insbesondere die Merkmale `sex`, `trestbps`, `chol` und `goal`.

Feld	Bedeutung
age	age in years
sex	sex (1 = male; 0 = female)
cp	chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic)
trestbps	resting systolic blood pressure (in mmHg on admission to the hospital)
chol	serum cholestoral in mg/dl
fbs	fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
restecg	resting electrocardiographic results (0 = normal; 1 = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV); 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
thalach	maximum heart rate achieved
exang	exercise induced angina (1 = yes; 0 = no)
oldpeak	ST depression induced by exercise relative to rest.[1]
slope	slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = downsloping)
ca	number of major vessels (0-3) colored by flourosopy
thal	3 = normal; 6 = fixed defect; 7 = reversable defect

[1] ST depression refers to a finding on an electrocardiogram, wherein the trace in the ST segment is abnormally low below the baseline.

Feld	Bedeutung
goal	diagnosis of heart disease (0: $< 50\%$ diameter narrowing ; 1, 2, 3, 4: $> 50\%$ diameter narrowing)

Einlesen der Herz-Daten

Lesen Sie die Datei aus der URL als Data Frame zur weiteren Bearbeitung ein. Überlegen Sie, ob sie Faktoren sinnvoll einsetzen können. Geben Sie die ersten drei Zeilen und fünf Spalten aus[2]:

```
# Ihre Lösung:
file = url('https://oc.informatik.hs-mannheim.de/s/wyzFq34K9HiNjXR/download')
df = data.frame(read.csv(file, sep = ',', header = TRUE, stringsAsFactors = TRUE))
df$sex =  factor(ifelse(df$sex==0, 'female', 'male'))
df$goal = factor(ifelse(df$goal == 0, 0, 1))

df[1:3,1:5]
```

```
##   age  sex cp trestbps chol
## 1  63 male  1      145  233
## 2  67 male  4      160  286
## 3  67 male  4      120  229
```

Cholesterin im Vergleich Männer/Frauen

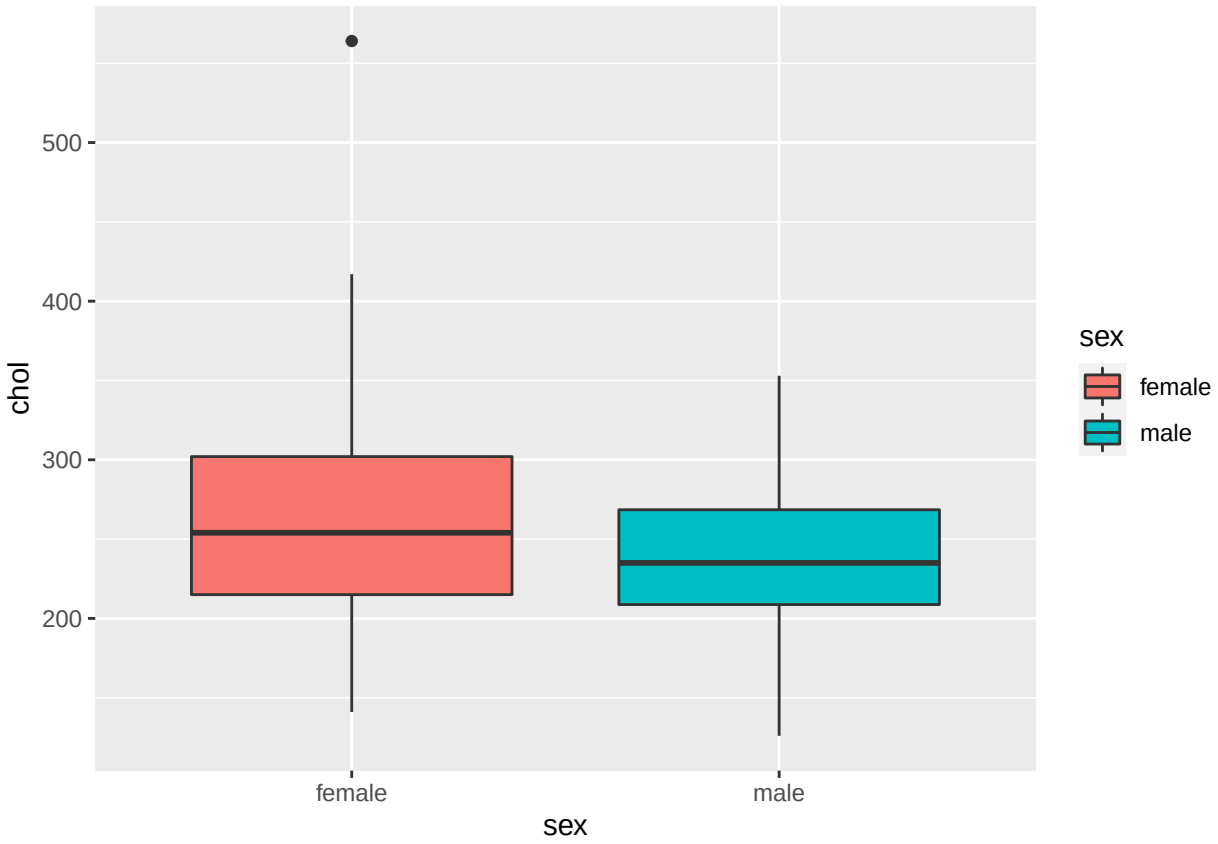
Nun sollen die Cholerstin-Werte untersucht werden – zunächst im Vergleich Männer zu Frauen.

Überblick über Cholesterin-Daten

Verschaffen Sie sich einen Überblick, indem Sie ein Boxplot für das Cholesterin gruppiert nach dem Geschlecht plotten.

```
# Ihre Lösung:
ggplot(df) +
  geom_boxplot(aes(sex, chol, fill=sex))
```

[2] Möglicherweise kommt es zu einem Fehler beim Einlesen des ersten Attributs (`age`). Manuelles Umbenennen hilft.



Konfidenz-Intervall

Berechnen Sie das Konfidenz-Intervall (Niveau 95%) für den Cholesterin-Level jeweils für Männer und Frauen.

Tabelle Geben Sie das Ergebnis als `kable`-Tabelle aus:

```
# Ihre Lösung:
t_m = t.test(df[df$sex == 'male',]$chol, conf.level = 0.95, alternative = 'two.sided')
t_f = t.test(df[df$sex == 'female',]$chol, conf.level = 0.95, alternative = 'two.sided')
result = data.frame(t_m$conf.int, t_f$conf.int)
rownames(result) = c('lower', 'upper')
colnames(result) = c('male', 'female')
knitr::kable(result)
```

	male	female
lower	233.7432	248.6722
upper	245.4607	274.8330

Überlappung? Überlappen sich die Bereiche?

Antwort: Nein

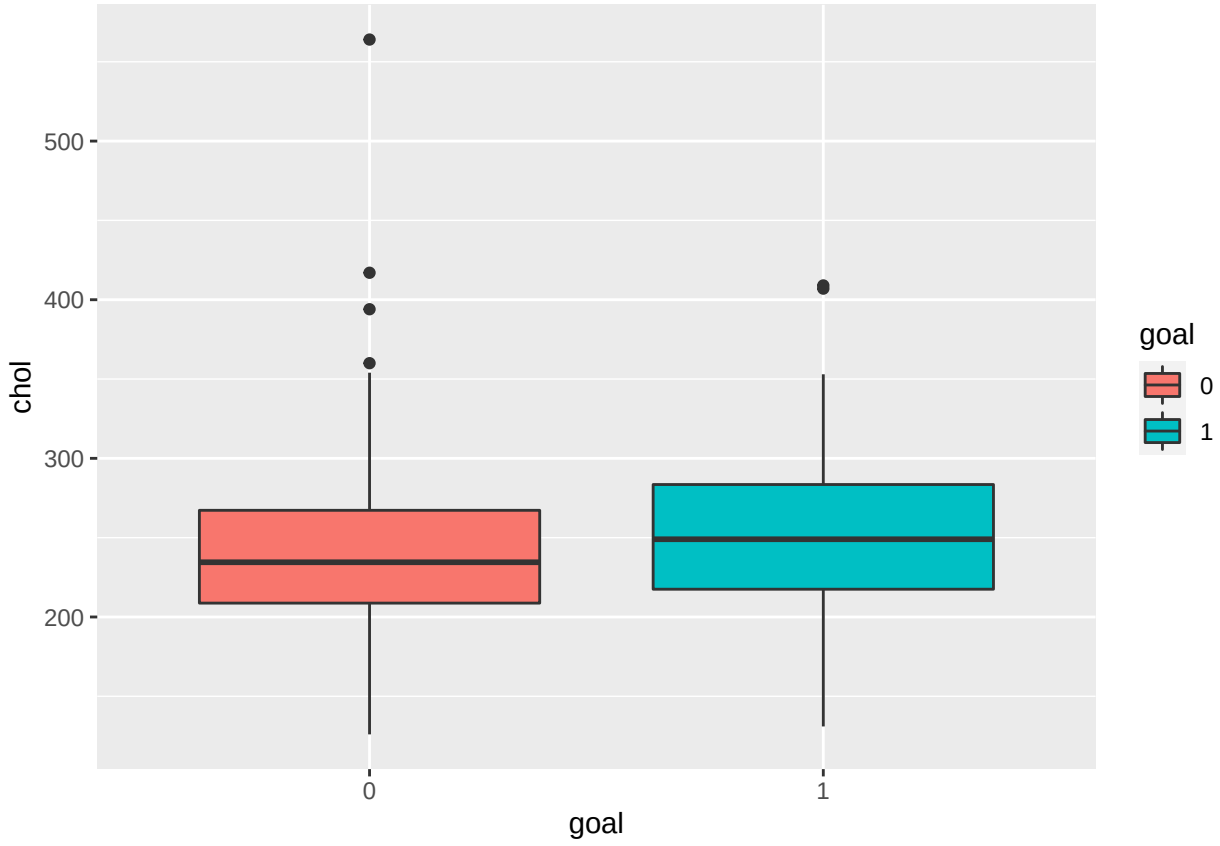
Cholesterin im Vergleich zur Erkrankung

Nun sollen die Cholesterin-Werte in Abhängigkeit der Diagnose untersucht werden.

Überblick über Cholesterin-Daten

Verschaffen Sie sich einen Überblick, indem Sie ein Boxplot für das Cholesterin gruppiert nach der Diagnose plotten.

```
# Ihre Lösung:
ggplot(df) +
  geom_boxplot(aes(goal, chol, fill = goal))
```



Konfidenz-Intervall

Berechnen Sie die Konfidenzintervalle für beide Gruppen und geben Sie das Ergebnis als `kable`-Tabelle aus:

```
# Ihre Lösung:
t_negative = t.test(df[df$goal == 0,]$chol, conf.level = 0.95, alternative = 'two.sided')
t_positive = t.test(df[df$goal == 1,]$chol, conf.level = 0.95, alternative = 'two.sided')
result = data.frame(t_negative$conf.int, t_positive$conf.int)
rownames(result) = c('lower', 'upper')
colnames(result) = c('negative', 'positive')
knitr::kable(result)
```

	negative	positive
lower	234.3977	243.1752
upper	250.8828	259.7744

Test

Es sieht so aus, als ob der Cholesterin-Wert bei den erkrankten Patienten höher ist als bei den nicht bzw. weniger erkrankten. Überprüfen Sie das mit einem Hypothesen-Test.

Wie lauten die Hypothesen? Formulieren Sie die Hypothesen (H_0 und H_1).

H_0: Der Cholesterin-Wert bei erkrankten Patienten ist nicht höher als bei nicht erkrankten $E(X) = \mu \leq 0$
H_1: Der Cholesterin-Wert bei erkrankten Patienten ist höher als bei nicht erkrankten $E(X) = \mu > 0$

Testanwendung Wenden Sie den Test mit R an. Was ist das Ergebnis?

```
# Ihre Lösung:
t.test(df[df$goal == 1,]$chol, conf.level = 0.95, mu = 0, alternative = 'greater')
```

```
##
##  One Sample t-test
##
## data:  df[df$goal == 1, ]$chol
## t = 59.912, df = 138, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  244.524     Inf
## sample estimates:
## mean of x
##  251.4748
```

Da p kleiner als 0,05 ist, wird H_0 verworfen. Der Cholesterin-Wert bei Erkrankten ist somit höher als bei nicht Erkrankten.

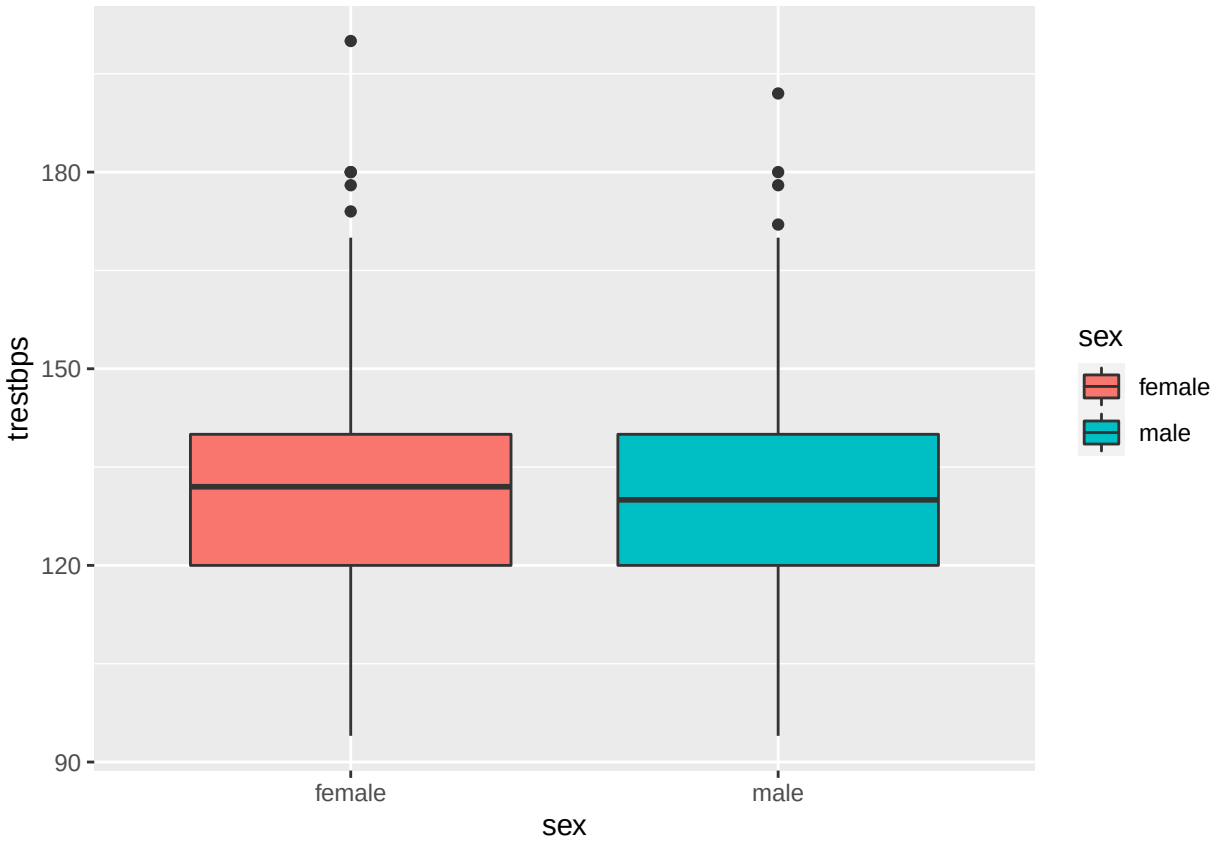
Systolischer Ruheblutdruck

Der systolische Blutdruck liegt beim gesunden Menschen bei ca. 120 mmHg.

Überblick über Blutdruck

Plot Verschaffen Sie sich einen Überblick, indem Sie ein Boxplot für den Blutdruck in Ruhe gruppiert nach dem Geschlecht plotten.

```
# Ihre Lösung:
ggplot(df) +
  geom_boxplot(aes(sex, trestbps, fill = sex))
```



Normalverteilt? Kann überhaupt davon ausgegangen werden, dass die Daten normalverteilt sind? Antwort: Ja, siehe zentraler Grenzwertsatz.

Konfidenzintervalle nach Erkrankung

Berechnen Sie die Konfidenzintervalle für den Ruheblutdruck aufgeschlüsselt nach der Diagnose (erkrankt/nicht erkrankt) und geben Sie das Ergebnis als `kable`-Tabelle aus:

```
# Ihre Lösung:
t_negative = t.test(df[df$goal == 0,]$trestbps, conf.level = 0.95)
t_positive = t.test(df[df$goal == 1,]$trestbps, conf.level = 0.95)
result = data.frame(t_negative$conf.int, t_positive$conf.int)
rownames(result) = c('lower', 'upper')
colnames(result) = c('negative', 'positive')
knitr::kable(result)
```

	negative	positive
lower	126.7514	131.4205
upper	131.7486	137.7161

Test, ob Kranke höheren Ruhe-Blutdruck haben

Überprüfen Sie mit einem Hypothesen-Test, ob Erkrankte einen höheren Ruhe-Blutdruck haben als gesunde Probanden.

Wie lauten die Hypothesen? Formulieren Sie die Hypothesen (H_0 und H_1). H_0: Der Ruhe-Blutdruck bei erkrankten Patienten ist nicht höher als bei gesunden $E(X) = \mu \leq 0$ H_1: Der Ruhe-Blutdruck bei erkrankten Patienten ist höher als bei gesunden $E(X) = \mu > 0$

Testanwendung Wenden Sie den Test mit R an. Was ist das Ergebnis?

```
# Ihre Lösung:
t.test(df[df$goal == 1,]$trestbps, df[df$goal == 0,]$trestbps, conf.level = 0.95, alternative = 'greate
```

```
##
##  Welch Two Sample t-test
##
## data:  df[df$goal == 1, ]$trestbps and df[df$goal == 0, ]$trestbps
## t = 2.6152, df = 274.64, p-value = 0.004705
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1.962045      Inf
## sample estimates:
## mean of x mean of y
##  134.5683  129.2500
```

Da p kleiner als 0,05 ist, wird H_0 verworfen. Der Ruhe-Blutdruck bei Erkrankten ist somit höher als bei Gesunden.

Clustering und PCA auf die Herzdaten

Einlesen der Herz-Daten

Es werden wieder die Herzdaten aus der letzten Aufgabe genutzt. Lesen Sie diese als Data Frame ein.

```
# Ihre Lösung:
file = url('https://oc.informatik.hs-mannheim.de/s/wyzFq34K9HiNjXR/download')
df = data.frame(read.csv(file, sep = ',', header = TRUE, stringsAsFactors = TRUE))
df$sex = factor(df$sex)
df$goal = factor(df$goal)
```

Bedeutet "ähnliche Merkmale" auch "gleiche Diagnose"?

Für jeden Datensatz ist bekannt, zu welcher Klasse er gehört: 0 (gesund) und 1 (erkrankt). Wir wollen untersuchen, wie gut *ähnliche* Datensätze zur gleichen Klasse gehören. Dafür soll mit dem k-means-Clusterverfahren der Datensatz in zwei Cluster eingeteilt werden.

Nur reelle Merkmale

Zunächst sollen **nur die nummerischen Merkmale** benutzt werden und nicht jene, die Faktoren sind.

Clustering Clustern Sie diese Daten. Überlegen Sie, ob Sie die Daten standardisieren wollen.

```
# Ihre Lösung:
df_scaled = scale(df |> dplyr::select_if(is.numeric))
clustered_km = stats::kmeans(df_scaled, centers = 2)
```

Richtig? Berechnen Sie, wie viel Prozent der Datensätze richtig einem Cluster eingeordnet wurden und geben Sie die Zahl auf zwei Nachkommastellen gerundet aus.

Hinweis: Berücksichtigen Sie, dass die Vergabe der Clusternummern zufällig ist. D.h. sowohl die Cluster (1, 2) wie auch (2, 1) sind möglich.

```
# Ihre Lösung:
calculate_percentage_correct = function (goal, cluster) {
  goal = sapply(goal, function(x) if (x == 0) 0 else 1)
  goal_class_1 = goal[1]
  goal_class_2 = 1 - goal_class_1
  cluster_class_1 = cluster[1]
  mapped_cluster = sapply(cluster, function(x) if (x == cluster_class_1) goal_class_1 else goal_class_2)
  difference = abs(goal - mapped_cluster)
  proportion_correct = 1 - sum(difference) / length(difference)
  percentage_correct = proportion_correct * 100
  return(list(percentage = round(percentage_correct, 2), difference = difference))
}

result = calculate_percentage_correct(df$goal, clustered_km$cluster)
result$percentage
```

```
## [1] 24.09
```

Scatterplot age vs. thalach Plotten Sie die Merkmale `age` und `thalach` als Scatterplot. Färben Sie die Punkte gemäß ihrer Clusterzuordnung ein. Die Form (`shape`) eines Punkts soll zeigen, ob die Klassifikation (d.h. der Cluster) richtig oder falsch ist.

```
# Ihre Lösung:
Zuordnung = factor(clustered_km$cluster)
Korrektheit = factor(result$difference * 4)
ggplot(df) +
  geom_point(aes(age, thalach, color = Zuordnung, shape = Korrektheit))
```

Mit Dummy-Variablen

Nun sollen **alle Merkmale** benutzt werden.

Clustering Clustern Sie diese Daten. Überlegen Sie, wie die Faktoren zu Zahlen werden.

```
# Ihre Lösung:
df_transformed = fastDummies::dummy_columns(df) |> dplyr::select_if(is.numeric)
df_all_scaled = scale(df_transformed)
clustered_all_km = stats::kmeans(df_all_scaled, centers = 2)
```

Richtig? Berechnen Sie für diesen Fall, wie viel Prozent der Datensätze richtig einem Cluster eingeordnet wurden und geben Sie die Zahl auf zwei Nachkommastellen gerundet aus. Wie hat sich der Wert verändert? Warum ist dies so?

```
# Ihre Lösung:
result_all = calculate_percentage_correct(df$goal, clustered_all_km$cluster)
result_all$percentage
```

```
## [1] 90.76
```

Der Wert ist deutlich höher als bei der vorherigen Zuweisung. Das leigt daran, dass jetzt mehr Merkmale verwendet werden konnten.

Scatterplot age vs. thalach Plotten Sie erneut und schauen Sie, wie die richtigen Punkte nun verteilt sind. Die nun hinzugenommenen Merkmale scheinen also von Bedeutung für die Zuordnung zu sein und folglich vermutlich mit der Erkrankung zusammenzuhängen.

```r
# Ihre Lösung:
Zuordnung = factor(clustered_all_km$cluster)
Korrektheit = factor(result_all$difference * 4)
ggplot(df) +
  geom_point(aes(age, thalach, color = Zuordnung, shape = Korrektheit))
```

PCA

Wenden Sie eine PCA auf diesen Datensatz an. Es sollen alle Merkmale berücksichtigt werden.

Wichtige Merkmale

Welche Merkmale der ersten Hauptkomponente tragen am meisten zu der Varianz bei? Geben Sie die TOP 10 Merkmale an.

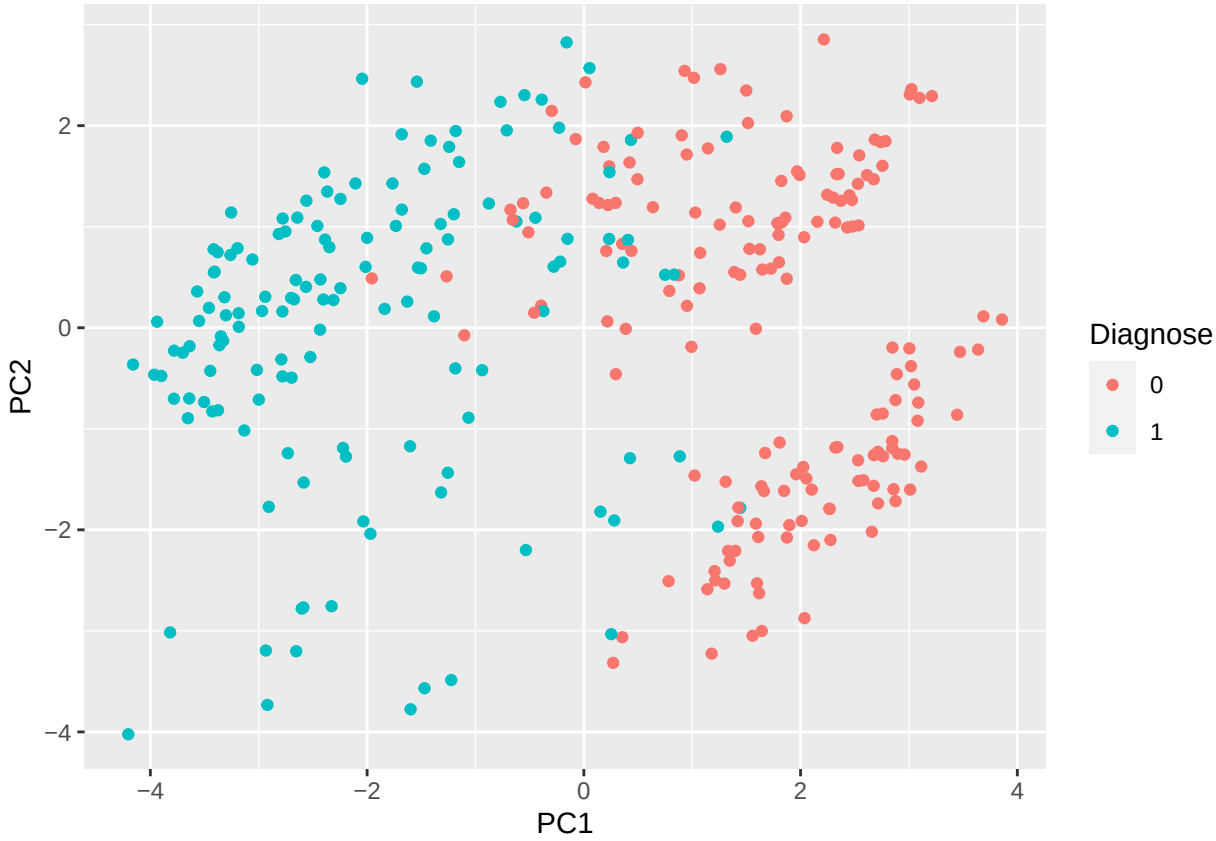
```r
# Ihre Lösung:
df_transformed_pca = fastDummies::dummy_columns(df) |> dplyr::select_if(is.numeric)
pca = stats::prcomp(df_transformed_pca, scale = TRUE)
sort(abs(pca$rotation[,'PC1']), decreasing = TRUE)[1:10]
```

```
##    goal_0    thal_3    thal_7   oldpeak   thalach      ca_0     exang     slope
## 0.3771421 0.3360315 0.2997635 0.2734078 0.2659951 0.2634484 0.2526032 0.2263581
##        cp    goal_3
## 0.2171080 0.2037185
```

Erste und zweite Hauptkomponente

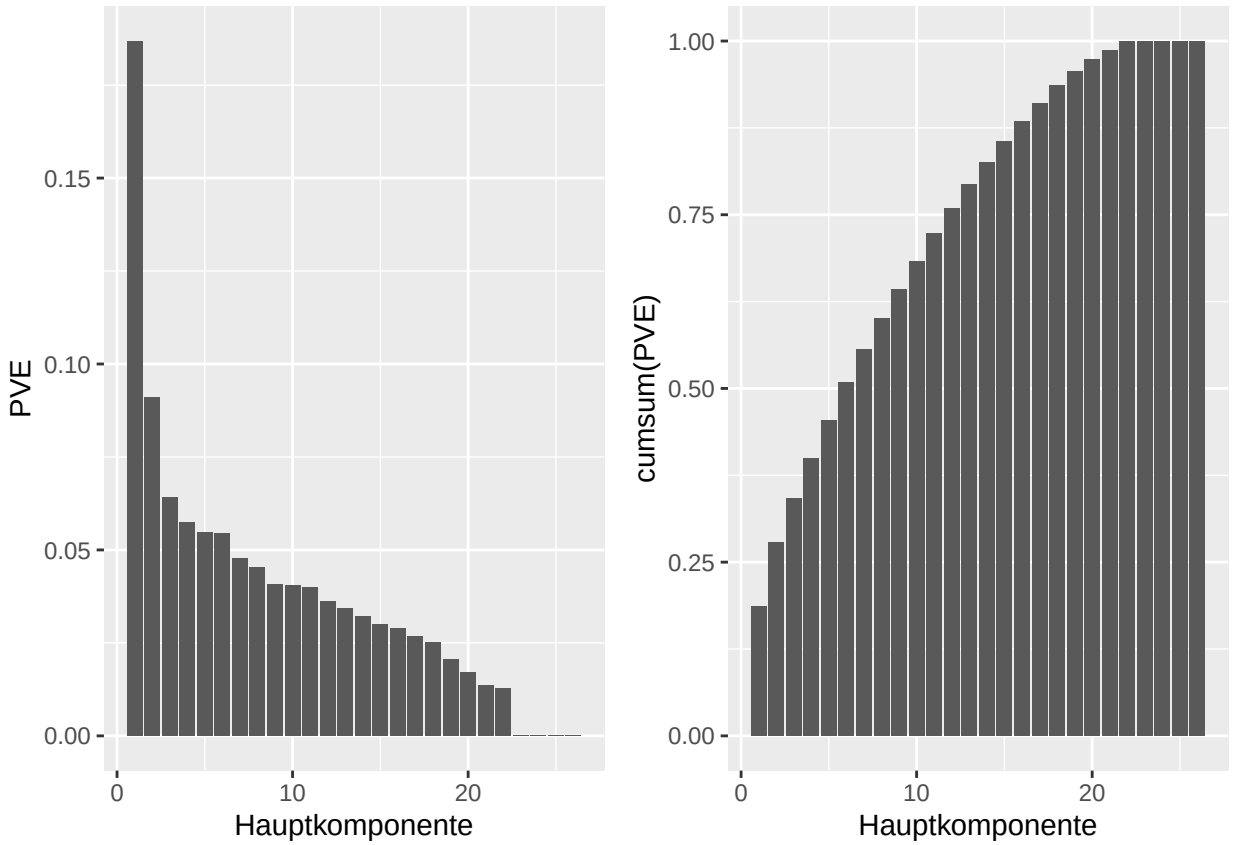
Plotten Sie die erste und zweite Hauptkomponente als Scatterplot. Färben Sie die Punkte gemäß ihrer Klasse (Disease) ein.

```
# Ihre Lösung:
Diagnose = factor(sapply(df$goal, function(x) if (x == 0) 0 else 1))
ggplot(data.frame(pca$x)) +
  geom_point(aes(PC1, PC2, color = Diagnose))
```



PVE

Plot Plotten Sie die Proportion of Variance explained (PVE) für jede Hauptkomponente sowie die akkumulierte PVE.



Wichtige Hauptkomponenten Wie viele Hauptkomponenten erklären mehr als 50% der Varianz?

Keine.

Möglicherweise tragen bei Ihrem Ergebnis die letzten Hauptkomponenten keine Varianz mehr bei. Überlegen Sie, woran das liegen könnte.

Das kann daran liegen, dass nicht alle Dimensionen zwangsweise mit der Klassifizierung zu tun haben. Dimensionen können auch unkorreliert sein.