Statistische Modellierung und Regressionsanalyse

Einleitung EDA

Mit der Checkliste wird erst die EDA verfolgt, um anschliessend besser die gestellten Fragen beantworten zu können. Bei der Auswahl der Fragen kann man viele Fragen stellen, hier kommen zwei zur Auswahl. Bei manchen Zeilen ist head verwendet worden, um das Maximum von 100 Seiten eher erreichen zu können.
Checkliste der explorativen Datenanalyse:

- Frage/n formulieren
- Daten einlesen
- str() zur ersten Überprüfung
- Typen überprüfen
- Dimensionen überprüfen
- Überprüfe den Anfang und das Ende des Datensatzes
- Unwichtige Spalten fallen lassen
- Duplikate erkennen
- Namen der Spalten ändern
- Fehlende Werte fallen lassen, imputieren
- Darstellung und Umgang mit Outliers
- Daten erkunden
- Überprüfung von Korrelation
- Mit mindestens einer weiteren Quelle überprüfen
- Weitere Fragen und Anmerkungen
- Quellenangaben

Fragen formulieren:

- Welches Land erhält die höchste/niedrigste Punktzahl und aus welcher Provinz kommen diese Weine her?
- Welche Degustierende kommen häufig vor und wo bewegen sich diese hauptsächlich

Statistisches Vorgehen EDA

Als erstes sollen die Daten eingelesen werden. Es handelt sich um einen Datensatz über Weine. Mit str können die Typen angesehen werden, mit head eine erste Einsicht gewonnen werden. Des Weiteren soll kurz die Währung angesehen werden, die Dimensionen den Anfang und das Ende des Datensatzes.

Überprüfung der Daten in mehreren Schritten

In [2]:
```
data_eda <- read.csv('winemag-data-130k-v2.csv') # Hier den Pfad ändern, falls notw

## Erste Überprüfung, Typen überprüfen
str(data_eda)

# Head
head(data_eda, n = 2)

# Es werden nun die Preise auf Ihre Währung überprüft
# Beispiel Zeile 217 und 221
print(data_eda[217,])
print(data_eda[221,])

# Dimensionen
dim(data_eda)

# Überprüfe den Anfang und das Ende des Datensatzes
head(data_eda,1)
tail(data_eda,1)
```

```
'data.frame':   129971 obs. of  14 variables:
 $ id                    : int  0 1 2 3 4 5 6 7 8 9 ...
 $ country               : chr  "Italy" "Portugal" "US" "US" ...
 $ description           : chr  "Aromas include tropical fruit, broom, brimstone and
dried herb. The palate isn't overly expressive, offering un"| __truncated__ "This is
ripe and fruity, a wine that is smooth while still structured. Firm tannins are fill
ed out with juicy r"| __truncated__ "Tart and snappy, the flavors of lime flesh and
rind dominate. Some green pineapple pokes through, with crisp ac"| __truncated__ "Pi
neapple rind, lemon pith and orange blossom start off the aromas. The palate is a bi
t more opulent, with note"| __truncated__ ...
 $ designation           : chr  "Vulkà Bianco" "Avidagos" "" "Reserve Late Harvest"
...
 $ points                : int  87 87 87 87 87 87 87 87 87 87 ...
 $ price                 : int  NA 15 14 13 65 15 16 24 12 27 ...
 $ province              : chr  "Sicily & Sardinia" "Douro" "Oregon" "Michigan" ...
 $ region_1              : chr  "Etna" "" "Willamette Valley" "Lake Michigan Shore"
...
 $ region_2              : chr  "" "" "Willamette Valley" "" ...
 $ taster_name           : chr  "Kerin O’Keefe" "Roger Voss" "Paul Gregutt" "Alexande
r Peartree" ...
 $ taster_twitter_handle: chr  "@kerinokeefe" "@vossroger" "@paulgwine " "" ...
 $ title                 : chr  "Nicosia 2013 Vulkà Bianco  (Etna)" "Quinta dos Avida
gos 2011 Avidagos Red (Douro)" "Rainstorm 2013 Pinot Gris (Willamette Valley)" "St.
Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)" ...
 $ variety               : chr  "White Blend" "Portuguese Red" "Pinot Gris" "Rieslin
g" ...
 $ winery                : chr  "Nicosia" "Quinta dos Avidagos" "Rainstorm" "St. Juli
an" ...
```

A data.frame: 2 × 14

	id	country	description	designation	points	price	province	region_1	region_2	ta
	<int>	<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>	
1	0	Italy	Aromas include tropical fruit, broom, brimstone and dried herb. The palate isn't overly expressive, offering unripened apple, citrus and dried sage alongside brisk acidity.	Vulkà Bianco	87	NA	Sicily & Sardinia	Etna		
2	1	Portugal	This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with juicy red berry fruits and freshened with acidity. It's already drinkable, although it will certainly be better from 2016.	Avidagos	87	15	Douro			

```
     id country
217 216   Italy

description
217 Black-skinned berry, pipe tobacco and Mediterranean herb aromas waft out of the
glass and carry over to the firm linear palate along with morello cherry, licorice a
nd a hint of clove. Taut fine-grained tannins and fresh acidity provide the backbone
while orange peel marks the finish. Drink 2019–2025.
    designation points price province              region_1 region_2
217                 90    57  Tuscany Brunello di Montalcino
      taster_name taster_twitter_handle
217 Kerin O’Keefe          @kerinokeefe
                                          title    variety           winery
217 Podere Scopetone 2012  Brunello di Montalcino Sangiovese Podere Scopetone
     id country
221 220      US

description
221 Briny green olive notes underlie a tropically ripe, full-bodied expression of th
e variety. The wine's apple, pineapple and mango fruit is wrapped in toasted oak and
moderate acidity.
    designation points price   province     region_1 region_2    taster_name
221                 90    40 California Sonoma Coast   Sonoma Virginie Boone
    taster_twitter_handle                                          title
221               @vboone Sixteen by Twenty 2014 Chardonnay (Sonoma Coast)
       variety            winery
221 Chardonnay Sixteen by Twenty
129971 · 14
```

A data.frame: 1 × 14

	id	country	description	designation	points	price	province	region_1	region_2	ta
	<int>	<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>	
1	0	Italy	Aromas include tropical fruit, broom, brimstone and dried herb. The palate isn't overly expressive, offering unripened apple, citrus and dried sage alongside brisk acidity.	Vulkà Bianco	87	NA	Sicily & Sardinia	Etna		

A data.frame: 1 ×

	id	country	description	designation	points	price	province	region_1	regio
	<int>	<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<cl
129971	129970	France	Big, rich and off-dry, this is powered by intense spiciness and rounded texture. Lychees dominate the fruit profile, giving an opulent feel to the aftertaste. Drink now.	Lieu-dit Harth Cuvée Caroline	90	21	Alsace	Alsace	

Resultate der Überprüfungen

Eine Auflistung der Resultate, mit der Reihenfolge Typen, Währung, Dimensionen und Formatierung.

- id: Ist vom Typ int, was Sinn macht, denn es handelt sich hier um eine Identifikation jedes einzelnen Eintrags. Für die exploratie Datenanalyse eher untwichtig und nicht aussagekräftig dann auch für die Verwendung in einer linearen oder logistischen Regressions ungünstig. Ist wohl noch ein Auszug aus einer DB Tabelle mit einem primary key. Da man Zeilen auch anders zählen kann, wird diese Spalte dann auch gelöscht (dropped), was man an Daten einsparen kann und dem Modell nicht dienlich ist, sollte man entfernen.

- points: Ist vom Typ int, macht Sinn, denn üblicherweise ist die Punktevergabe mit ganzen Zahlen vorhanden.

- price: Ist vom Typ int, und is der Preis des Weines.Hier wird anscheined gerundet, Rappen wie beispielsweise bei Coop oder Migros gibt es auch. Somit üblicherweise verkaufen Läden Ihre Produkte nicht gerundet.

- country, description, designation, province, region_1, region_2, taster_name, taster_twitter_handle, title, variety, winery: Sind vom Typ chr, macht Sinn, denn die Einträge sind alle Strings. Eventuell für ein Modell müsste man hier dann numerische Werte verwenden (One hot encoding?).

Resultat zu den Features selbst:

- country: Gibt das Land an
- description: Beschreibt den Wein, es handelt sich ja um eine Degustation
- designation: Name oder Titel des Weines
- ponts: Punktevergabe
- price: Preis des Weines, es wird jedoch keine Währung angegeben
- province: Die Hauptregion oder Bundesland
- region_1: Weinregion innerhalb der Provinz (wird noch unbenannt)
- region_2: Unterregion von region_1 (wird noch unbenannt)
- variety: Die Rebsorte, die verwendert wird
- winery: Weingut

Ein Wein stammt aus der Toskana (Preis 57) und eine aus Kalifornien (Preis 40) Die Überprüfung der Webpage WineEnthusiast (Ist auch die Quelle,) gibt stets US Dollars an. Somit nehmen wir an, dass die Preise in US Dollar vorkommen.
14 Spalten, 129971 Zeilen
Zeilen und Spalten können sich im Verlaufe der EDA verändern. Für ein Modell werden kleine Datensätze bevorzugt. Dies sind die nächsten Schritte nach der EDA, wobei es dann um Feature Eingineering / Selection geht.

Die Formatierung schein zu stimmen, Umlaute und/oder spezielle Zeichen könnte man ebenfalls ausbessern.

Spalten, Duplikate, Imputation

Es sollen nun Spalten, Duplikate und die Imputation angewendet werden.

In [3]:
```
library(ggplot2)
# Unwichtige Spalten fallen lassen
data_eda2 <- data_eda[, c("country", "description", "designation", "points", "price

# Duplikate erkennen
table(duplicated(data_eda2))
duplicated_rows <- data_eda2[duplicated(data_eda2), ]
dim(duplicated_rows)
head(duplicated_rows,2)
```

```
 FALSE   TRUE
119987   9984
```

9984 · 11

A data.frame: 2 × 11

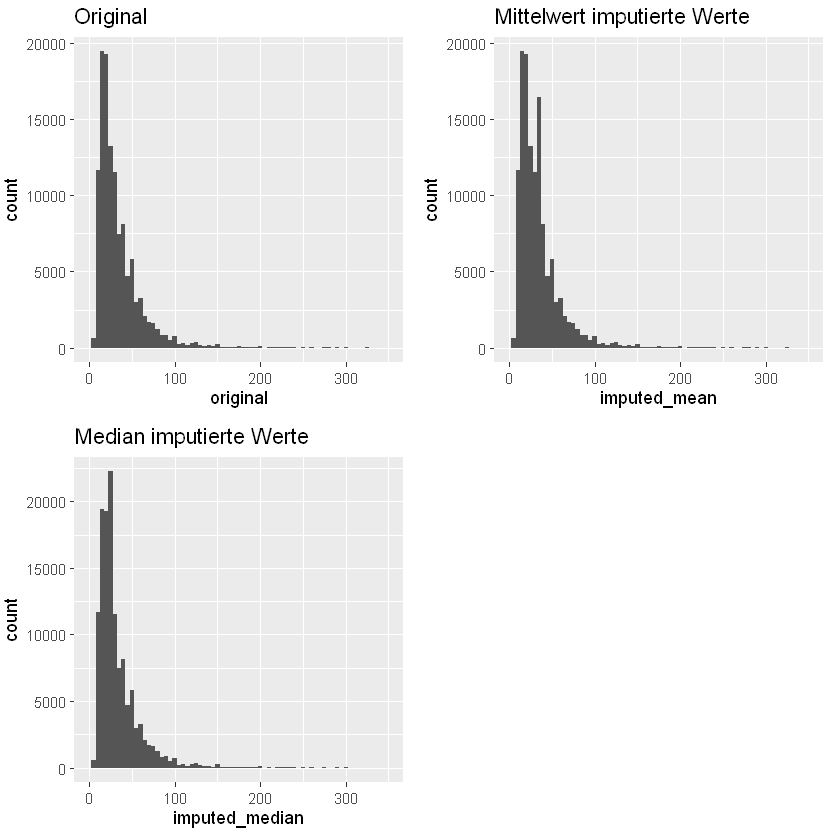
	country	description	designation	points	price	province	region_1	region_2	taster
	<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>	
2409	US	This is weighty, creamy and medium to full in body. It has plenty of lime and pear flavors, plus slight brown sugar and vanilla notes.		85	14	California	North Coast	North Coast	V
2410	Italy	There's a touch of toasted almond at the start, but then this Grillo revs up in the glass to deliver notes of citrus, stone fruit, crushed stone and lemon tart. The mouthfeel is crisp and simple.	Sallier de la Tour	85	13	Sicily & Sardinia	Sicilia		

In [4]:
```
# Namen der Spalten ändern
names(data_eda2)[names(data_eda2) == "region_1"] <- "region"
names(data_eda2)[names(data_eda2) == "region_2"] <- "subregion"

# Fehlende Werte fallen lassen, imputieren
sapply(data_eda2, function(x) sum(is.na(x)))
```

country: 0 **description:** 0 **designation:** 0 **points:** 0 **price:** 8996 **province:** 0 **region:** 0 **subregion:** 0 **taster_name:** 0 **variety:** 0 **winery:** 0

In [5]:
```
library(gridExtra)
value_imputed <- data.frame(
  original = data_eda2$price,
```

```
  imputed_mean = replace(data_eda2$price, is.na(data_eda2$price), mean(data_eda2$pr
  imputed_median = replace(data_eda2$price, is.na(data_eda2$price),
  median(data_eda2$price, na.rm = TRUE)))

original <- suppressWarnings(ggplot(value_imputed, aes(x = original)) +
  geom_histogram(binwidth = 5) + xlim(0,350) +
  ggtitle("Original"))

mean_imputed <- suppressWarnings(ggplot(value_imputed, aes(x = imputed_mean)) +
  geom_histogram(binwidth = 5) + xlim(0,350) +
  ggtitle("Mittelwert imputierte Werte"))

median_imputed <- suppressWarnings(ggplot(value_imputed, aes(x = imputed_median)) +
  geom_histogram(binwidth = 5) + xlim(0,350) +
  ggtitle("Median imputierte Werte"))

suppressWarnings(grid.arrange(original, mean_imputed, median_imputed, ncol = 2))
```



Resultate der Anpassungsschritte

- Die Spalte taster_twitter wird nicht weiter verwendet. Je nach Fragestellung ist eine Spalte mehr oder weniger wichtig. Insgesamt gesehen, könnte man so auch nachträglich weitere Fragen stellen, somit spricht dies für den Erhalt der Spalten.

- Es gibt 9984 Duplikate im Datensatz. Hierbei handelt es sich aber um Zeilen, die teilweise übereinstimmende Werte haben. Beipsiel die Punkteanzahl von 85. Da das Feld Beschreibung üblicherweise immer ein Unikat ist, kann man auch nie reine Duplikate erkennen. Somit gibt es hier kein reinen Duplikate.

- Bei der Spalte Preise gibt es 8996 NA Werte. Die Idee ist nun das Histogramm anzusehen, dies ohne Imputation und mit Imputation. Die Berechnungen sind der Mittelwert und der Median.
- Es gibt sehr hohe Preise für Weine (Beispiel 3300). Diese werden eher bei den Outliers behandelt. Hier geht es in erster Linie um die Imputation und die Auswirkungen davon.
- Die Mittelwert Imputation würde bei dem Bin 7 eine Erhöung von 7457 auf 16453 bedeuten.Die Bin Weite ist 5, 7*5 ist 35. Somit gibt es einen Anstieg der Weine mit den Preisen zwischen 30 USD und 35 USD um 8996.
- Die Median Imputation würde bei dem Bin 5 eine Erhöhung von 13245 auf 22241 bedeuten. Die Bin Weite ist 5, 5*5 ist 25. Somit gibt es einen Anstieg der Weine mit den Preisen zwischen 20 USD und 25 USD.
- Wie nun diese Imputationen dann richtige oder eher falsche Ergebnisse erzielen, könnte man weiterhin untersuchen, würde aber etwas den Rahme der Arbeit sprengen, da es sich nur um eine Teilarbeit handelt, die Auswirkungen von Imputation liegt nicht im Fokus der Semesterarbeit.
- Da nur die Preise teilweise fehlen, aber alle anderen 14 Features keine NA Werter aufweisen, werden wir nicht imputieren und die Zeilen auch nicht fallen gelassen. Ebenfalls für die Fragestellung ist dies Spalte weniger relevant.

Outliers bei Punkte und Preise und Korrelation

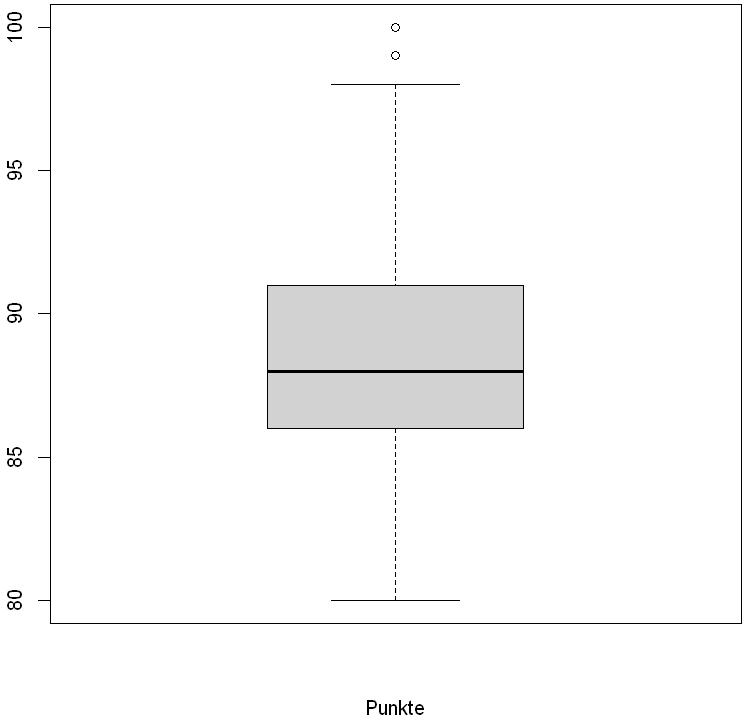
Outliers bei den Punkte und Preisen werden anhand von Boxplots erkannst. Mit dem upper und lower boudn können dann die Outliers aus dem Datensatz gelesen werden. Die Prüfung der Korrelation soll anschliessend mehr über den Zusammenhang von Punkte und Preise zeigen.

In [6]:
```
# Darstellung und Umgang mit Outliers (Boxplot) Punkte
# Boxplot Punkte
boxplot(data_eda2$points, xlab="Punkte")
summary(data_eda2$points)
q3_points <- quantile(data_eda2$points, 0.75)
iqr_points <- IQR(data_eda2$points)
upper_bound_points <- q3_points + 1.5*iqr_points
upper_bound_points
q1_points <- quantile(data_eda2$points, 0.25)
lower_bound_points <- q1_points - 1.5*iqr_points
lower_bound_points
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  80.00   86.00   88.00   88.45   91.00  100.00 
```

75%: 98.5

25%: 78.5



In [7]:
```r
# Empty values
sum(is.na(data_eda2$points))

# Select Outliers Punkte
outliers_points <- data_eda2[data_eda2$points >= 98.5, ]
head(outliers_points,2)
dim(outliers_points)
```

0

A data.frame: 2 × 11

	country	description	designation	points	price	province	region	subregion
	<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>
346	Australia	This wine contains some material over 100 years old, but shows no signs of fragility. Instead, it's concentrated through age and should hold in the bottle indefinitely. It's dark coffee-brown in color, with delectable aromas of rancio, dried fig, molasses and black tea, yet despite enormous concentration avoids excessive weight. And it's amazingly complex and fresh on the nearly endless finish.	Rare	100	350	Victoria	Rutherglen	
1557	US	The flagship wine from Quilceda Creek offers exotic scents of plum, cassis, loam, coffee and pine sap, a rich and evocative blend. The wine delivers all that is		99	125	Washington	Columbia Valley (WA)	Columbia Valley

country	description	designation	points	price	province	region	subregion
<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>
	promised and more; it is deep and dense with flavor, polished, focused and persistent. Vanilla, espresso, fine tannins, luscious acids and cascading fruits.						

52 · 11

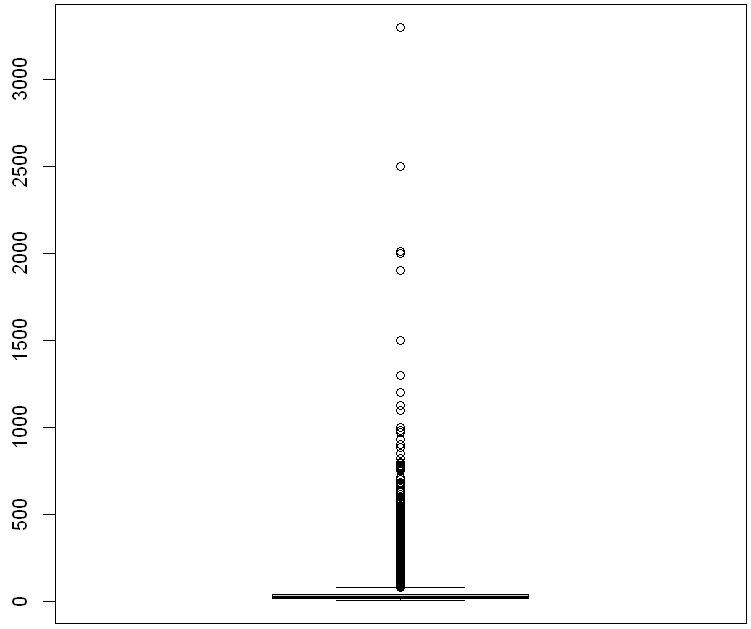
- Die Spalte Punkte hat keine leeren Werte, und es gibt 52 Outliers, die alle über dem upper bound von 98.5 liegen.

In [8]:
```
# Darstellung und Umgang mit Outliers (Boxplot) Preis
# Boxplot Preis
boxplot(data_eda2$price, xlab="Preis")
summary(data_eda2$price)
q3_price <- quantile(data_eda2$price, 0.75, na.rm=TRUE)
iqr_price <- IQR(data_eda2$price, na.rm=TRUE)
upper_bound_price <- q3_price + 1.5*iqr_price
upper_bound_price
q1_price <- quantile(data_eda2$price, 0.25, , na.rm=TRUE)
lower_bound_price <- q1_price - 1.5*iqr_price
lower_bound_price
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   4.00   17.00   25.00   35.36   42.00 3300.00    8996
```

75%: 79.5

25%: -20.5



In [9]:
```r
# Empty values
sum(is.na(data_eda2$price))

# Select Outliers Preis
outliers_price <- data_eda2[data_eda2$price >= 79.5, ]
tail(outliers_price,2)
dim(outliers_price)
```

8996

A data.frame: 2 × 11

	country	description	designation	points	price	province	region	subregion
	<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>
129932	France	A powerful, chunky wine, packed with solid tannins that promise good aging. It has a sense of drama in its spice, dark tannins and spacious fruit. This is going to develop well over 5–10 years.		91	107	Burgundy	Grands-Echezeaux	
NA.8995	NA	NA	NA	NA	NA	NA	NA	NA

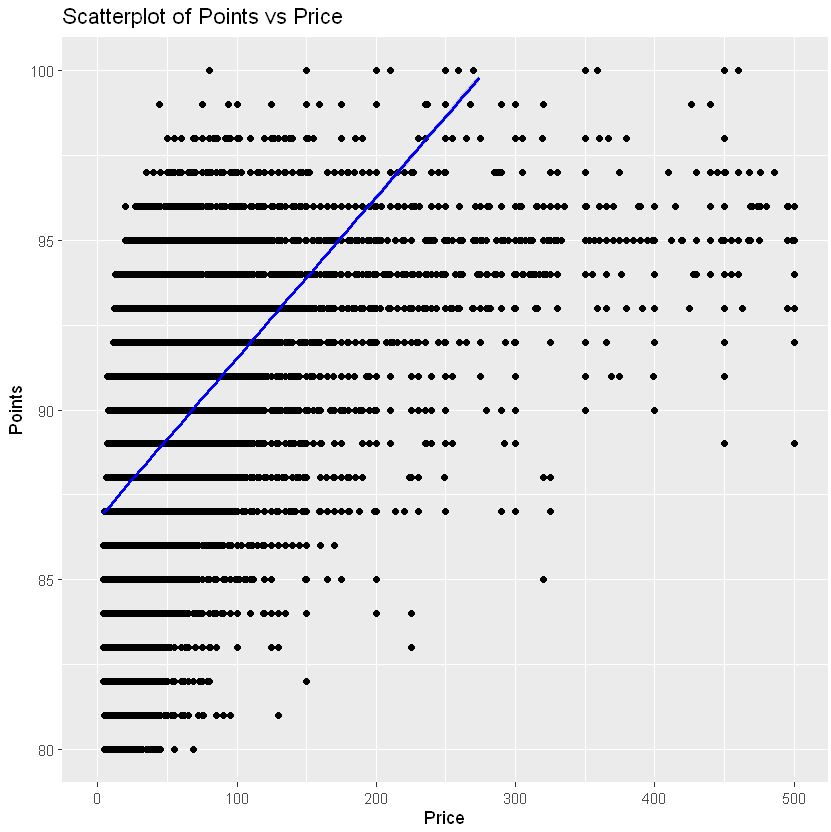
16237 · 11

- Die Spalte Punkte hat 8996 leere Werte, und es gibt 16237 Outliers, die alle über dem upper bound von 79.5 liegen.

In [10]:
```
# Überprüfung der Korrelation mit einem Scatterplot
numeric_data <- data_eda2[, c("points", "price")]
numeric_data <- na.omit(numeric_data)
correlation <- cor(numeric_data$points, numeric_data$price, use = "complete.obs")
print(correlation)

suppressWarnings(print(ggplot(numeric_data, aes(x = price, y = points)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Scatterplot of Points vs Price", x = "Price", y = "Points") +
  xlim(0,500) +
  ylim(80,100)))
```

```
[1] 0.4161667
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Resultate Outliers bei Punkte und Preise und Korrelation

Resultat für die Spalte Punkte:

- Mit dem IQR kann bestimmt werden, was sich noch in dem Boxplot und in den Whiskers befindet. Hier ist 98.5 als upper bound berechnet. Alles was über 98.5 liegt sind Outliers. Die Punktzahl 99 und 100 ist nun nicht besonders weit entfernt von 98.5, daher ist es fraglich, ob man diese trotzdem Outliers nennen kann, obwohl die Rechnung es beweist.
- Der lower bound liegt bei 78.5, hier gibt es keine Outliers, denn das Minimum an Punkten die erreicht werden können ist 80.
- Aus dem Summary kann man erkennen, dass das Minimum 80 und das Maximum 100 beträgt. 52 Einträge (Zeilen) sind somit Outliers, siehe dim.
- Will man nun die Outliers, die über den upper bound gehen löschen? Auf keinen Fall, es handelt sich hierbei um Weine, die die höchste Punktzahl erreicht haben. Es wäre noch interessant zu erfahren, warum diese Weine derart hoch ausgezeichnet werden. Dies in einem nächsten Schritt.

Resultat für die Spalte Preise:

- Mit dem IQR kann bestimmt werden, was sich noch in dem Boxplot und in den Whiskers befindet. Hier ist 79.5 als upper bound berechnet. Alles was über 79.5 liegt sind Outliers. 7241 von 129971 sind Outliers. Der lower bound beträgt -20.5. Die kleinste Preis ist 4

und logischerweise wird kein Wein gratis verkauft. Somit gibt es keine Outliers in diesem Bereich. Aus dem Summary kann man erkennen, dass das Minimum 4 und das Maximum 3300 beträgt.
- Will man nun die Outliers, die über den upper bound gehen löschen? Auf keinen Fall, es handelt sich hierbei um Weine, die nebst der hohen Punktzahl auch einen hohen Preis erzielt haben. Es wäre noch interessant zu erfahren, warum diese Weine derart hoch ausgezeichnet werden. Dies in einem nächsten Schritt.

- Es ist anzumerken, dass 8996 Werte in dem Boxplot ignoriert sind, weil diese NA Werte aufweisen. 7241 Werte werden als Outliers gekennzeichnet, ev. könnte man diese ignorieren. Jedoch können Outliers auch so wichtige Informationen liefern. Wie nun die vorangegangene Imputation die Resultate hätte verändern können bedarf einer weiteren Untersuchung. Ist hier aber out-of-scope. Ebenfalls out-of-scope ist die Untersuchung des Modell, sollte man hier die Outliers ignorieren wollen.

Resultat der Korrelation:

- Mit eine Wert von 0.4161667 kann gesagt werden, dass eher keine Korrelation vorliegt. Es ist näher zu 0 als zu 1 oder -1. Man kann erkennen, dass eine höhere Punktzahl auch einen höheren Preis bedeutet. Es gitb aber viele kostengünstigere Weine, die eine hohe Punktzahl erhalten und umgekehrt.

Resultat des Scatterplots, das die Ergebnisse der Korrelation weiter unterstütz:

- Es ist interessant zu sehen, dass eine hohe Anzahl an Punkten nicht immer auch einen hohen Preis bedeutet. Der eine 3300 US Dollar hat eine Punktzahl von 88 und kommt aus Frankreich, Bordeaux (visuell aus der Tabelle). Der eine 350 US Dollar hat eine Punktzahl von 100 und kommt aus Australien Victoria (visuell aus der Tabelle). Zu erkennen ist eine Zunahme des Preises mit der Zunahme der Punkte. 94, 95, 96 und 97 sind das zu erreichende Maximum, danach folgt eine Abnahme

Fragen beantworten

- Welches Land erhält die höchste/niedrigste Punktzahl und aus welcher Provinz kommen diese Weine her? ACHTUNG: Mit head gekürzt um die erlaubten 100 Seiten erreichen zu können.
- Welche Degustierende kommen häufig vor und wo bewegen sich diese hauptsächlich.

In [11]:
```
# Welches Land erhält die höchste/niedrigste Punktzahl und aus welcher Provinz komm

# Maximum points
max_points <- max(data_eda2$points, na.rm = TRUE)
highest_points_data <- subset(data_eda2, points == max_points)
highest_points_info <- highest_points_data[, c("country", "province", "points")]
head(highest_points_info)
```

```r
# Minimum points
min_points <- min(data_eda2$points, na.rm = TRUE)
lowest_points_data <- subset(data_eda2, points == min_points)
lowest_points_info <- lowest_points_data[, c("country", "province", "points")]
head(lowest_points_info)
```

A data.frame: 6 × 3

	country	province	points
	<chr>	<chr>	<int>
346	Australia	Victoria	100
7336	Italy	Tuscany	100
36529	France	Champagne	100
39287	Italy	Tuscany	100
42198	Portugal	Douro	100
45782	Italy	Tuscany	100

A data.frame: 6 × 3

	country	province	points
	<chr>	<chr>	<int>
345	Chile	Leyda Valley	80
3641	Portugal	Vinho Verde	80
3642	Chile	Maule Valley	80
4557	Italy	Central Italy	80
4558	Spain	Catalonia	80
5906	Argentina	Mendoza Province	80

In [12]:
```r
# Welche Degustierende kommen vor und wo bewegen sich diese hauptsächlich?
taster_country_data <- data_eda2[, c("taster_name", "country")]
head(taster_country_data)
taster_summary <- aggregate(country ~ taster_name, data = taster_country_data,
                            function(x) paste(unique(x), collapse = ", "))
head(taster_summary,5)
```

A data.frame: 6 × 2

	taster_name	country
	<chr>	<chr>
1	Kerin O'Keefe	Italy
2	Roger Voss	Portugal
3	Paul Gregutt	US
4	Alexander Peartree	US
5	Paul Gregutt	US
6	Michael Schachner	Spain

A data.frame: 5 × 2

	taster_name	country
	<chr>	<chr>
1		Italy, US, France, South Africa, Australia, Mexico, Chile, New Zealand, Germany, Spain, Israel, Argentina, Portugal, Austria, Hungary, Canada
2	Alexander Peartree	US
3	Anna Lee C. Iijima	Germany, US, Romania, Czech Republic, Slovenia, Canada, Croatia, Bulgaria, Hungary, Morocco, Bosnia and Herzegovina, Slovakia, Georgia, Turkey, Lebanon, Moldova, Ukraine, Macedonia
4	Anne Krebiehl MW	Austria, France, England,
5	Carrie Dykes	US

Resultate der Fragen

- Höchste Punktzahl, siehe Output: Australien, Frankreich, Italien, Portugal und US. Frankeich ist mit Bordeaux und Champagne gut vertreten. Italien mit der Toskana. Portugal mit Duoro und Port. US mit California und Washington. Australien einmal mit Victoria.
- Bemerkung: Es ist nicht überraschend, das Frankreich und Italien gut vertreten sind. Es gehört vorallem bei Frankreich zur Kultur, haben das passende Klima und Weine stellen diese schon seit Jahrzenten her. US mit Kalifornien hat ebenfalls ein gutes Klima. Australien kommt nur einmal vor. Insgesamt kann bestimmt auch das Klima berücksichtig werden, diese Untersuchung wäre aber out-of-scope.
- Niedrigste Punktzahl, siehe Output: Hier auch erreichen die Länder mit der Maximalanzahl der Punkte auch das Minimun, das zu erreichen ist. Beispielweise liefert US Kalifornien ebenfalls Weine mit der niedrigsten Anzahl Punkte.

- Siehe Output: Anna Lee C. Iijima, Jeff Jenssen, Joe Czerwinski, Susan Kostrzewa sind sehr weit gekommen. Deren Aufenthalt ist in der USA, in Europa und auch in Nahosten.

Übeprüfung mit einer zweite Quelle

Bei der EDA wird ebenfalls noch mit einer weiteren Quelle verglichen. Zum einen kann man so auch die Aussagen von Fragen unterstützen oder auch Unterschiede erkennen und daraus erneut Wissen generieren. Dieses mal wird nur das Land Frankreich mit Bordeaux und Burgundy berücksichtigt. DIe Boxplots in Vergleich zeigen einen starken Unterschied. Ebenfalls die teursten Weine unterscheiden sich im Preis.

Resultat des Quellenvergleichs

Auch bei der zweiten Quelle wird Frankreich, Bordeaux und Burgundy angesehen, siehe Quelle2_FR. Die Preisliste ist in EUR, das Maximum ist 60.45 Euro, im Vergleich zur ersten Quelle mit 3300 USD.

- Hier ein Bespiel des teuersten Weins aus der ersten Quelle in USD: https://www.wineenthusiast.com/buying-guide/chateau-les-ormes-sorbet-2013-medoc/?queryID=ac1352b7cc25cfd3c23491b18663e1af&objectID=wine#230236&indexName=PROI

- Wir haben Anzahl Zeilen aus Quelle 1 mit 9921 und aus der Quelle 2 131 Einträge. Der Mittelwert der Quelle 1 liegt bei 52.96 bei der Quelle 2 14.46. Aufgrund der Outliers macht es auch Sinn den Median zu betrachten: Quelle 1 ist 28, Quelle 2 ist 10.85

- Aufgrund der wenigen Zeilen, die die zweite Quelle hergibt, kann nicht besonders viel daraus hergeleitet werden. Die Zunahme der zweiten Quelle besagt aber, dass es viele Degustierende geben kann, die Weine von unterschiedlicher Qualität testen. Wine Enthusiast, wie der Name bereits sagt, berücksichtigt auch sehr teure Weine, es gibt deutlich mehr Daten, möglicherweise werden auch andere Weinorte berücksichtigt, die eher einem höheren Standard entsprechen. Jeder kann Wein degustieren, eine Liste erstellen, Votings hinzufügen etc. Nicht jeder kann aber Weine auf hohem Niveau bewerten.

Weitere Fragen und Anmerkungen:

- Wie die teuersten Weine zustandekommen ist wohl ein Thema von Jahrgang und das Klima zu jener Zeit, wie die Trauben wachsen (Boden, Licht. Wetter) usw.. Interessant ist, dass es auch hohe Punktzahlen gibt aber der Preis nicht übetrieben ist, siehe auch Resultat bei den Outliers in der ersten Quelle.

- Die zweite Quelle liefert deutliche Unterschiede in den Preisen der Weine. Die erwähnte Outliers aus der ersten Quelle kommen da nicht vor. Der einzige sehr teuer Wein mit 3410.79 EUR stammt aus dem Pomerol.

- Die erste Quelle liefert nebst "normalen" Weinen auch sehr qualitativ hochwertige Weine an.

Beschreibung Ergebnisse EDA

Die Ergebnisse kommen alle bei den bereits erwähnten Resultaten vor.

Analyse/Hinterfragen Ergebnisse EDA

Die Analyse der Ergebnisse kommt bei den bereits erwähnten Resultaten vor.

Einleitung lineare Regression

Statistisches Vorgehen lineare Regression

Die Schritte der linearen Regression sind Modellannahmen prüfen, Residuen überprüfen und das Treffen von Vorhersagen. Der Datensatz wird dazu dann aufgetrennt in 98% und 2%. Die 98% dienen der Modellbildung, 2% dienen dann den Vorhersagen.

Multiple lineare Regression, Modellannahmen prüfen

In [13]:
```r
## Multiple lineare Regressionsanalyse
# Modell erstellen vor der Kürzung des Datensatz
# (Die Kürzung wird dann für die Vorhersage verwendet. Der Datensatz wird dann um 2
data_eda2$country <- factor(data_eda2$country)
data_eda2$variety <- factor(data_eda2$variety)
data_eda2 <- na.omit(data_eda2)
```

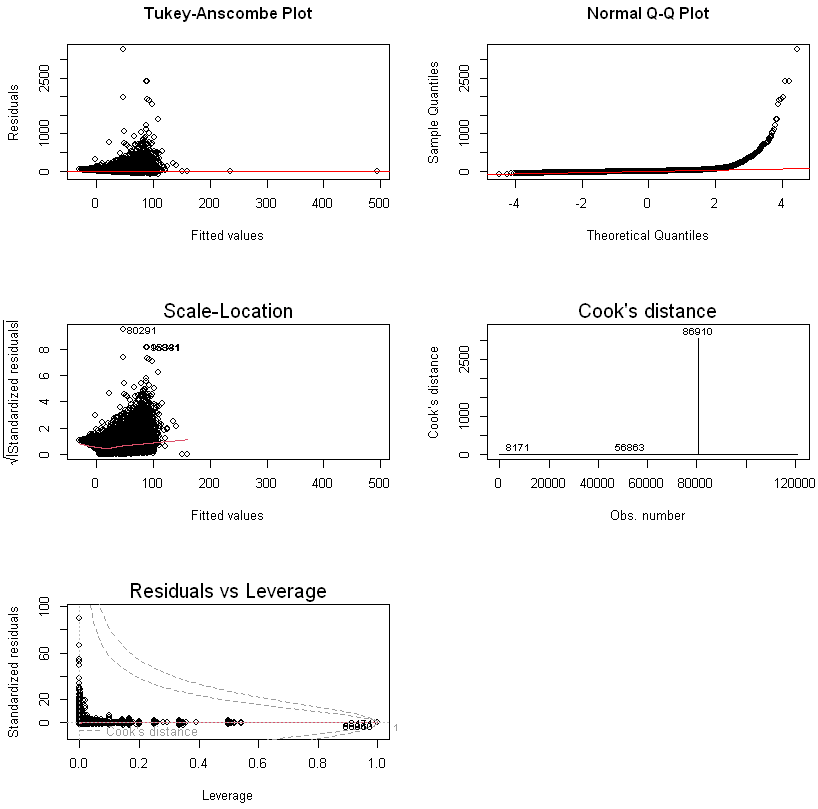
In [14]:
```r
mlr <- lm(price ~ points + country + variety, data=data_eda2)
coefficients <- summary(mlr)$coefficients
significant_coefficients <- coefficients[coefficients[, "Pr(>|t|)"] < 0.05, ]
print(significant_coefficients)
```

```
                          Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)             -436.642229 36.72037107 -11.891008 1.374768e-32
points                     5.130424  0.03669983 139.794190 0.000000e+00
countryFrance             15.265413  5.18057767   2.946662 3.212854e-03
countryGermany            19.054967  5.25809419   3.623930 2.902793e-04
countryItaly              12.386156  5.19013041   2.386483 1.701195e-02
countryMexico             15.846382  6.77765172   2.338034 1.938710e-02
countrySwitzerland        61.253042 14.81510780   4.134499 3.559715e-05
countryUruguay            14.170548  6.43754410   2.201235 2.772128e-02
varietyFrancisa          117.387529 51.19676338   2.292870 2.185720e-02
varietyRamisco           445.322041 51.22041539   8.694229 3.534735e-18
varietyRosenmuskateller  110.180384 51.20577802   2.151718 3.142157e-02
varietyTerrantez         176.061194 51.22065288   3.437309 5.877244e-04
```

- Dies sind die Variablen, die dem Modell am meisten beisteuern.

In [15]:

```r
# Modellprüfung
 # Residuals vs. Fitted, Tukey-Anscombe
par(mfrow = c(3, 2))
suppressWarnings(plot(mlr$fitted.values, mlr$residuals,
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Tukey-Anscombe Plot"))
abline(h = 0, col = "red")

# Q-Q Plot
suppressWarnings(qqnorm(mlr$residuals))
qqline(mlr$residuals, col = "red")

suppressWarnings(plot(mlr, which=3)) # Scale-Location
suppressWarnings(plot(mlr, which=4)) # Cook's distance
suppressWarnings(plot(mlr, which=5)) # Residuals vs. Leverage
```



- Man erkennt bei den Plots, dass die Resultate schlecht ausfallen. Das Modell ist ungeeignet. Mehr dazu bei den Resultaten.

In [17]:
```
# Kürzung des Datensatz
# Antstelle der vorhandenen Libraries wie caret oder caTools
# werde ich von Hand 98% der Daten für das Modell wählen und 2% dann für die Vorher
# 98% sind gerundet 127373 Zeilen, 2% sind 2598 Zeilen
library(dplyr)
# Mixen
set.seed(123)
shuffled_dataset <- data_eda2[sample(nrow(data_eda2)), ]
data_frame <- as.data.frame(shuffled_dataset)
dim(data_eda2)
# 98% wählen, mit diesen wird das Modell neu gebildet
data_98 <- data_frame %>% slice(1:118556)

# 2% wählen, mit diesen wird die Vorhersage getan, aus diesen kann stichprobenweise
#data_2 <- data_frame %>% filter(row_number() >= 127374 & row_number() <= 129970)
data_2 <- data_frame %>% slice(118557:120975)

# Modell erneut bilden und auswerten
mlr2 <- lm(price ~ points + country + variety, data=data_98)
coefficients2 <- summary(mlr2)$coefficients
significant_coefficients2 <- coefficients2[coefficients2[, "Pr(>|t|)"] < 0.05, ]
print(suppressWarnings(significant_coefficients2))
```

```
                          Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)             -436.336579 36.81804993 -11.851159 2.214299e-32
points                     5.127084  0.03716126 137.968514 0.000000e+00
countryBrazil             14.928660  7.46652778   1.999411 4.556616e-02
countryFrance             15.358101  5.19479944   2.956438 3.112767e-03
countryGermany            18.815741  5.27399214   3.567647 3.603446e-04
countryItaly              12.343490  5.20458973   2.371655 1.771023e-02
countryMexico             15.869392  6.79576957   2.335187 1.953533e-02
countrySwitzerland        72.884771 15.94394951   4.571312 4.851711e-06
countryUruguay            14.305116  6.47054738   2.210805 2.705129e-02
varietyFrancisa          117.428477 51.32853464   2.287782 2.215201e-02
varietyRamisco           445.389115 51.35257816   8.673160 4.255012e-18
varietyRosenmuskateller  110.233860 51.33772797   2.147229 3.177709e-02
varietyTerrantez         176.134947 51.35282033   3.429898 6.040144e-04
```

- Auch bei dem gekürzten Datensatz (noch 98%) sind es diese Variablen, die für das Modell wichtig sind. Dieser Datensatz wird bei den Vorhersagen verwendet.

Resultat Multiple lineare Regression, Modellannahmen prüfen

- Die wichtisten Variablen, die anscheinend Aussagekraft bieten sind: Punkte, Frankreich, Deutschland, Italien, Mexiko, Schweiz und Uruguay. Die wichtigen variety Spalten sind hier: varietyFrancisa, varietyRamisco, varietyRosenmuskateller, varietyTerrantez Aufgrund der Ladezeit habe ich mit Absicht nur wenige Features gewählt. Idealerweise würde mann alle Features verwenden.

- Als nächstes würde man untersuchen, warum genau dies Variablen zum Modell Wert beisteuern. Dies jedoch gerade out-o-scope für eine Teilaufgabe.

- Residuals vs. Fitted, Tukey-Anscombe: Die Punkte streuen nicht um den Wert 0, es sieht mehr nach einem Funnel Form aus, was auf Heteroskedastizität hindeutet, also die Varianz scheint nich konstant zu sein. Ebenfalls kann dies auf Nicht-Linearität hinweisen. Ebenfalls die Kurvenform deutet wie die Funnel Form auf dasselbe Problem.

- Q-Q Plot: Die Punkte weichen nach 2 stark von der Linie ab. Somit folgen die Residuen keiner Normalverteilung.

- Scale-Location: Die rote Linie ist nicht horizontal und die Punkte sind nicht zufällig verteilt. Homoskedastizität kommt somit hier wohl nicht vor. Mit Breusch-Pagan kann auch Homoskedastizität geprüft werden.

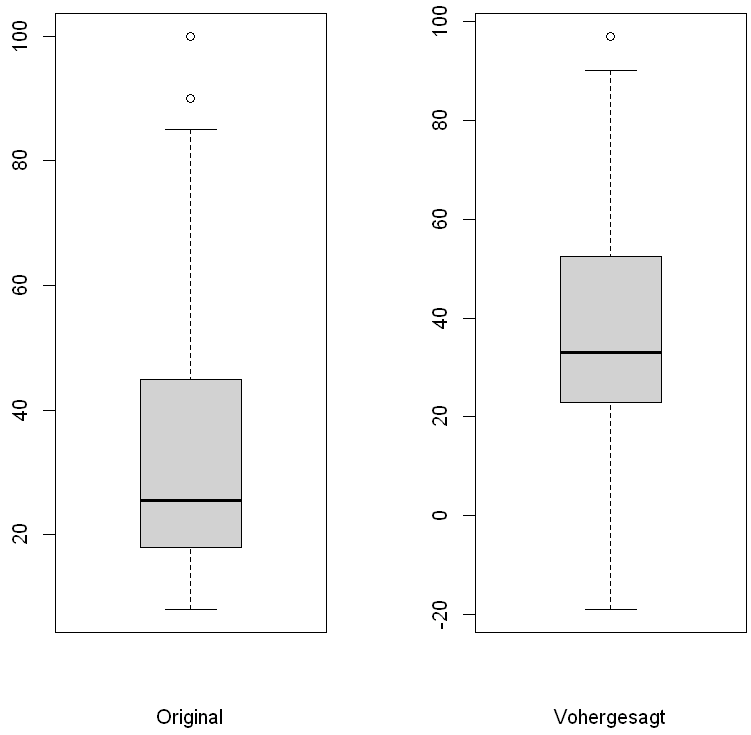
- Cook's distance: Es gibt Outliers und diese könnten das Modell beeinflussen. Hier sind diese aber sehr interessant, denn es handelt sich um Weine, die nicht der Norm entsprechen. Interessant wäre zu wissen, warum diese zu Outliern werden. Ev. gibt es dazu mehr über den Wachstum der Trauben, Boden, Lichteinfluss, Klima usw.. zu erfahren. Vorausgesetzt, dass die Daten stimmen.

- Residuals vs. Leverage: Hohe Hebelwirkung aber kleine Resiuden deutet eher darauf hin, dass diese Outliers keinen grossen Einfluss auf das Modlel haben.
- Die beiden ersten Plots könnte man ebenso mit "which" zeichnen. Ich habe dies nun ohne "which", weil diese schneller zeichnen.
- Mit eine Transformation könnte man nun das Modell verbessern, damit es den Modellannahmen besser entspräche. Wie eine lineare, logistische Transformation oder Wurzeltransformation. Ev. eignen sich auch nicht lineare Modell ohne Transformation.
- Auch nach der Kürzung des Datensatzes gibt es natürlich keine Verbesserung am Modell. Die Resultate sind dieselben Werte. Das gebildete Modell mlr2 auf den gekürzten Datensatz soll für die Vorhersagen verwendet werden. Der Rest des Datensatzes würd für die Vorhersage verwendet.
- 1209756 wird entsprechend rechnerisch von Hand in 98% und 2% mit slice getrennt. Dies sind dann die 1:118556 und die 118557:120975.

Multiple lineare Regression, Vorhersagen treffen

In [18]:
```
# Einige Vorhersagen des Modells mlr2

# Herstellung des leeren dataframes für die Originalwerte price und vorhergesagten
empty_df <- data.frame(
  original = numeric(),
  predicted = numeric()
)

# 100 Vorhersagen von dem davor generierten data_2, das somit als Testdatensatz die
for (x in 1:100) {
  prediction <- suppressWarnings(predict(mlr2, newdata = data_2[x,]))
  empty_df[x, ] <- c(data_2[x,]$price,round(prediction))
}
print(head(empty_df,10))

par(mfrow = c(1, 2))
boxplot(empty_df$original, xlab="Original")
boxplot(empty_df$predicted, xlab="Vohergesagt")

# Beispiel Zeile 79
print(empty_df[79,])
```

```
   original predicted
1        17        15
2        48        37
3        20        33
4        34        42
5        24        31
6        13        22
7        14        17
8        25        24
9        22        30
10       18        22
   original predicted
79       25        38
```



Multiple lineare Regression, Vorhersagen treffen

- Wie erwartet sind die Vorhersagen nicht besonders genau. Das hat auch die visuelle Modellbeurteilung ergeben.
- Bei der Liste mit den zehn Einträgen kann man erkennen, dass es wenige gibt, die eine gute Vorhersage liefern, siehe dazu die 10 Zeilen.
- Die Boxplots sind nicht übereinstimmend aber auch nicht krass unterschiedlich, es unterstützt die Zahlenwerte, die manchmal besser stimmen und manchmal nicht. Eine weitere Untersuchung dieser Werte wird nicht unternommen.

Beschreibung Ergebnisse multiple lineare Regression

Die Ergebnisse kommen alle bei den bereits erwähnten Resultaten vor.

Analyse/Hinterfragen Ergebnisse multiple lineare Regression

Die Analyse der Ergebnisse kommt bei den bereits erwähnten Resultaten vor.

- Es könnte weiterhin gesehen werden, wie die Resutlate sich ändern, wenn man mehrere shuffle vornehmen und wiederum diese vergleichen würde.
- Mit einer Transformation würde ertwartet werden, dass sich die Werte verbessern.

Einleitung logistische Regression

Anbei nun die logistische Regression.

Statistisches Vorgehen logistische Regression

Auch hier folgt diese typische Vorgehensweise: Modellprüfung, Residuenanalyse und die Vorhersagen.

Modellprüfung logistische Regression

Auch hier gilt es das Modell auf Tauglichkeit zu prüfen, was im Endeffekt beduten wird, wie gut die Vorhersagen sind.

In [19]:
```
# Logistische Regression
# Auswahl eines Datensatzes, der mehr numerische Werte hat, in diesem Fall der movi
# Für die logistische Regression kommt nun auch noch eine Spalte mit 0 und 1 dazu.
# Bei einem Wert von 6 oder darüber ergibt dies eine 1, sonst eine 0
# Die 6 ist hier willkürlich gewählt, ev. müsste man genauer angeben, wie man den T
# Hier entspricht die 6 einem Minimum.

# Neuer Datensatz
dataset2 <- read.csv('movies.csv') # Hier den Pfad ändern, falls notwendig
votes <- ifelse(dataset2$vote_average >= 6, 1, 0)
```

```r
# Modell erstellen
logit_model <- glm(votes ~ vote_count + popularity + runtime, data = dataset2, fami
summary(logit_model)
```

```
Call:
glm(formula = votes ~ vote_count + popularity + runtime, family = binomial(),
    data = dataset2)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.288e-01  8.340e-02  -8.738   <2e-16 ***
vote_count   4.191e-04  2.694e-05  15.554   <2e-16 ***
popularity   1.028e-03  4.152e-04   2.475   0.0133 *
runtime      1.334e-02  8.673e-04  15.384   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 11873  on 9988  degrees of freedom
Residual deviance: 10857  on 9985  degrees of freedom
  (25 observations deleted due to missingness)
AIC: 10865

Number of Fisher Scoring iterations: 6
```

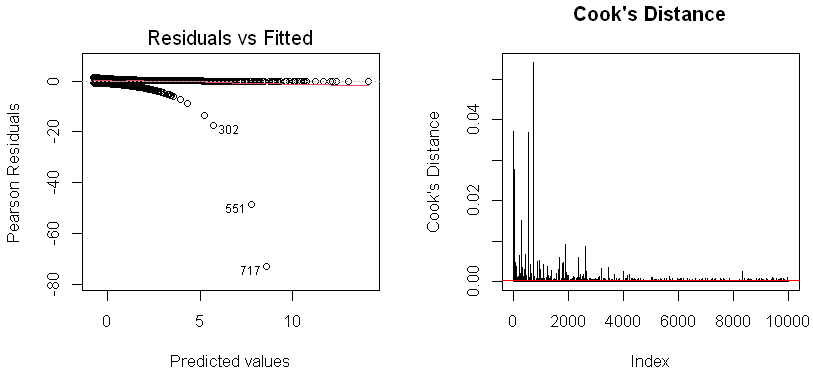
Resultat Modellprüfung logistische Regression

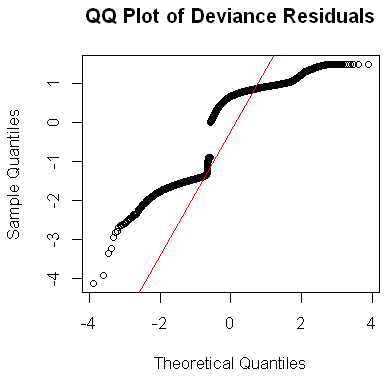
- vote_count und runtime scheinen für dieses Modell wichtige Features zu sein.

Residuenanalyse logistische Regression

Es folgt die Residuenanalyse.

In [20]:
```r
par(mfrow = c(2, 2))
# Residuenanalyse
plot(logit_model, which = 1)
# Cook's distance
cooks_d <- cooks.distance(logit_model)
plot(cooks_d, type = "h", main = "Cook's Distance", ylab = "Cook's Distance")
abline(h = 4 / nrow(dataset2), col = "red")  # Rule of thumb threshold
# Q-Q Plot
deviance_residuals <- residuals(logit_model, type = "deviance")
qqnorm(deviance_residuals, main = "QQ Plot of Deviance Residuals")
qqline(deviance_residuals, col = "red")
```





Resultat Residuenanalyse logistische Regression

- Der Residuenplot zeigt Übereinstimmungen und Abweichungen. Der Erwartungswert ist nahe bei 0 mit Abweichungen. Die Werte sind am Anfang gut verstreut, weichen dann aber in eine Kurve aus mit Outliers.
- Cook's Distanz zeigt hier Werte an, die das Modell beinflussen könnten.
- Der Q-Q Plot zeigt hier weniger eine Normalverteilung der Residuen an (gibt es bei der logistischen Regression nicht), sondern zeigt eher Outliers an.

Vorhersage logistische Regression

In [21]:
```
# Vorhersage
# Dieses Mal wird der Datensatz nicht aufgetrennt, sondern es wird willkürlich ein
data_log <- data.frame(original_title="Some title", popularity=5402.308, release_da

pred <- predict(logit_model, newdata = data_log,
type = "response")
print(pred)
```

```
        1
0.9989793
```

Resultat Vorhersage logistische Regression

Das Resultat entspricht einer 1. Der vote_average ist bei der ersten Zeile auf grösser als 6, was dann auch eine 1 ergibt. Da nun data_log nur geringfügig sich von der Originalzeile unterscheidet, wird hier als pred auch eine 1 erwartet.

- Das Resultat könnte aber auch nur Zufall sein, den auch bei dem logistischen Modell sollte man eine Transformation versuchen, um das Modell verbessern zu können.
- Hier habe ich auf 100 Werte verzichtet, da wie bei der linearen Regression auch die Werte nur aus Zufall besser oder schlechter sind, da eine Transformation nicht angewendet worden ist.
- ROC ist nicht thematisiert worden

Anmerkungen:

- Aufgrund der Ladezeit habe ich nur wenige Features bei dem dataset2 verwendet.
- Für die logistische Regression habe ich einen neuen Datensatz dataset2 verwendet, weil dataset nur zwei numerische Spalten vorweisen kann und die Umrechnung nicht numerischer Spalten in numerische Spalten nicht Thema dieser Teilarbeit ist. Eine Schwäche sind die 0 Werte bei den Einnahmen. Diese könnten für das Modell ein Problem darstellen. Macht auch nicht Sinn, denn die Einnahmen sind immer gösser als 0.

Beschreibung Ergebnisse logistische Regression

Die Ergebnisse kommen alle bei den bereits erwähnten Resultaten vor.

Analyse/Hinterfragen Ergebnisse logistische Regression

- Die Wahl der Zahl 6, um die 1 und 0 zu bestimmen ist willkürlich gewählt. Es könnte weiter untersucht werden, welcher Wert am besten ist und wie sich die Resultate dann unterscheiden.

Quellenangaben

- Datensatz für EDA, erste Quelle: https://mavenanalytics.io/data-playground?order=date_added%2Cdesc&search=wine (1.11.2024)

- Datensatz für EDA, zweite Quelle: https://www.kaggle.com/datasets/budnyak/wine-rating-and-price (1.11.2024)
- Datensatz für lineare Regression: https://mavenanalytics.io/data-playground?order=date_added%2Cdesc&search=wine (1.11.2024)
- Datensatz für logistische Regression: https://www.kaggle.com/datasets/omkarborikar/top-10000-popular-movies (1.11.2024)

In [ ]:

Hypothesentest und Varianzanalyse

Einleitung Hypothesentest und Varianzanalyse

Mit der Checkliste wird Schritt für Schritt der Hypothesentest und die Varianzanalyse berechnet.
Vorgehensweise:

- Daten einlesen
- Annahmen für einen t-Test prüfen
- Tranformation testen
- Ein-Stichproben t-Test
- Zwei-Stichproben t-Test, gepaart, ungepaart
- Effektstärken
- ANOVA

Begründung für die Auswahl des Datensatzes CO2 von R:

- Da bereits bei der Semesterarbeit 1 und Semesterarbeit 2 das Thema EDA vorgekommen ist, wird hier nicht erneut der Datensatz bereinigt oder auf Schwachstellen überprüft. Somit entfällt die ganze Standardisierung, Normalisierung, Imuputation, usw. Der Datensatz kann somit gleich verwendet werden (Transformation wird kurz angesehen). Eher sollen die Vorbedingungen für den t-test und ANOVA überprüft werden.

- Obwohl die Stichprobengrösse > 30 ist, ist jedoch die Varianz der Grundgesamtheit unbekannt, daher die Auswahl des t-test. Der Z-test käme zur Auswahl bei >= 30 und Varianz der Grundgesamtheit ist bekannt.

Resultate der Anpassungsschritte

- Es handelt sich um die CO2 Aufnahme von Grass Pflanzen Echinochloa crus-galli, auch genannt Hühnerhirse.

Plant: Ordinale Factors

Type: Factors

Treatment: Factors

conc: Numerisch

Uptak: Numersich

Statistisches Vorgehen Hypothesentest und Varianzanalyse

Annahmen für einen unabhängigen t-Test überprüfen

Die zu püfenden Annahmen sind: Numerisch mit str, Beobachtungen sind unabhänig voneinander mit Scatterplot und Überprüfung der Korrelation, Varianzüberprüfung mit einem Sample und den Boxplots, Überprüfung der Normalverteilung mit QQ-Plot (separat chilled und nonchilled un darin für die Spalten conc und uptake) und Shapiro-Wilk (ebenfalls auch chilled und nochchilled mit den Spalten con und uptake).

In [46]:
```
# Daten einlesen
# Verwendung des Datensatzes CO2
data_co2 <- CO2
head(CO2)
# Überprüfen
str(data_co2)
```

A nfnGroupedData: 6 × 5

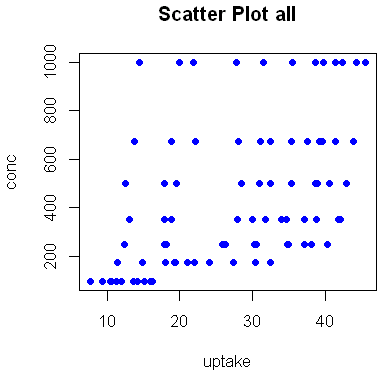
	Plant	Type	Treatment	conc	uptake
	<ord>	<fct>	<fct>	<dbl>	<dbl>
1	Qn1	Quebec	nonchilled	95	16.0
2	Qn1	Quebec	nonchilled	175	30.4
3	Qn1	Quebec	nonchilled	250	34.8
4	Qn1	Quebec	nonchilled	350	37.2
5	Qn1	Quebec	nonchilled	500	35.3
6	Qn1	Quebec	nonchilled	675	39.2

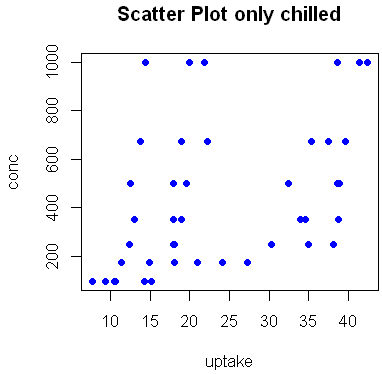
```
Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':      84 o
bs. of  5 variables:
 $ Plant    : Ord.factor w/ 12 levels "Qn1"<"Qn2"<"Qn3"<..: 1 1 1 1 1 1 1 2 2 2 ...
 $ Type     : Factor w/ 2 levels "Quebec","Mississippi": 1 1 1 1 1 1 1 1 1 1 ...
 $ Treatment: Factor w/ 2 levels "nonchilled","chilled": 1 1 1 1 1 1 1 1 1 1 ...
 $ conc     : num  95 175 250 350 500 675 1000 95 175 250 ...
 $ uptake   : num  16 30.4 34.8 37.2 35.3 39.2 39.7 13.6 27.3 37.1 ...
 - attr(*, "formula")=Class 'formula'  language uptake ~ conc | Plant
  .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
 - attr(*, "outer")=Class 'formula'  language ~Treatment * Type
  .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
 - attr(*, "labels")=List of 2
  ..$ x: chr "Ambient carbon dioxide concentration"
  ..$ y: chr "CO2 uptake rate"
 - attr(*, "units")=List of 2
  ..$ x: chr "(uL/L)"
  ..$ y: chr "(umol/m^2 s)"
```

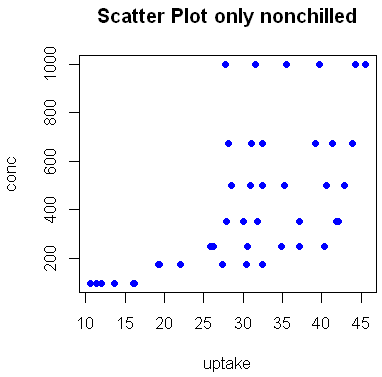
In [47]:

```r
library(dplyr)
# Annahmen prüfen
# Die Daten sind numerisch, siehe Datensatz

# Die Beobachtungen sind unabhängig voneinander
chilled <- data_co2 %>% filter(data_co2$Treatment == "chilled")
nonchilled <- data_co2 %>% filter(data_co2$Treatment == "nonchilled")

par(mfrow = c(2, 2))
plot(data_co2$uptake, data_co2$conc, main = "Scatter Plot all",
     xlab = "uptake", ylab = "conc",
     pch = 19, col = "blue")

plot(chilled$uptake, chilled$conc, main = "Scatter Plot only chilled",
     xlab = "uptake", ylab = "conc",
     pch = 19, col = "blue")

plot(nonchilled$uptake, nonchilled$conc, main = "Scatter Plot only nonchilled",
     xlab = "uptake", ylab = "conc",
     pch = 19, col = "blue")

correlation <- cor(data_co2$uptake, data_co2$conc)
print(correlation)

correlation <- cor(chilled$uptake, chilled$conc)
print(correlation)

correlation <- cor(nonchilled$uptake, nonchilled$conc)
print(correlation)
```

```
[1] 0.4851774
[1] 0.4277141
[1] 0.6081648
```



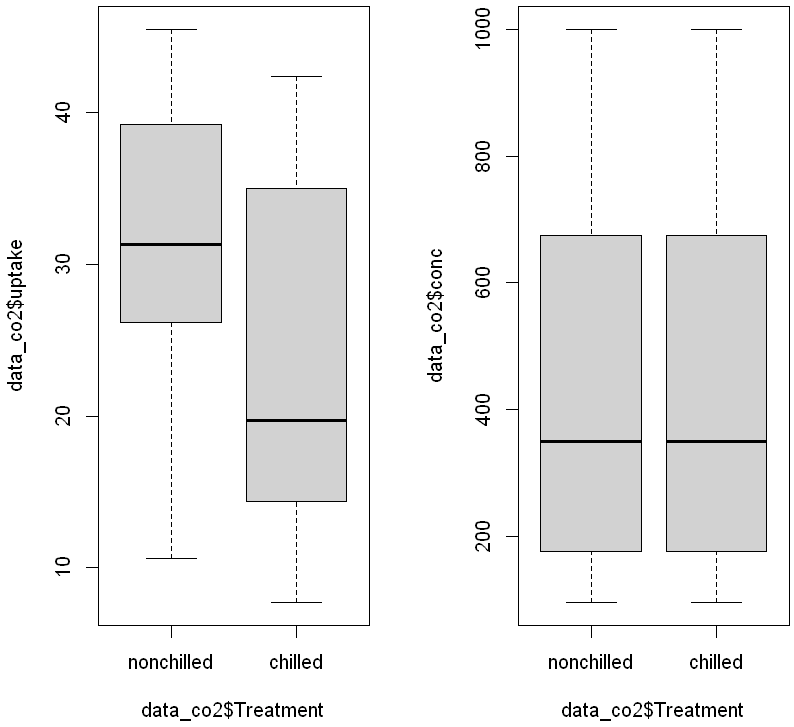




Sowhol Scatterplot wie auch die Zahlenwerte deuten auf Unabhängigkeit, dazu mehr bei den Resultaten.

In [48]:
```r
par(mfrow = c(1, 2))
data_frame1 <- boxplot(data_co2$uptake~data_co2$Treatment, data=data_co2)
data_frame2 <- boxplot(data_co2$conc~data_co2$Treatment, data=data_co2)

# Sample size, Mean, SD für chilled, nonchilled und uptake
df_summary1 <- data_co2 %>%
  group_by(Treatment) %>%
  summarise(
    sample_size = n(),
    mean_value = mean(uptake),
    sd_value = sd(uptake)
  )

print(df_summary1)

# Sample size, Mean, SD für chilled, nonchilled und conc
df_summary2 <- data_co2 %>%
  group_by(Treatment) %>%
  summarise(
    sample_size = n(),
    mean_value = mean(conc),
    sd_value = sd(conc)
  )

print(df_summary2)
```

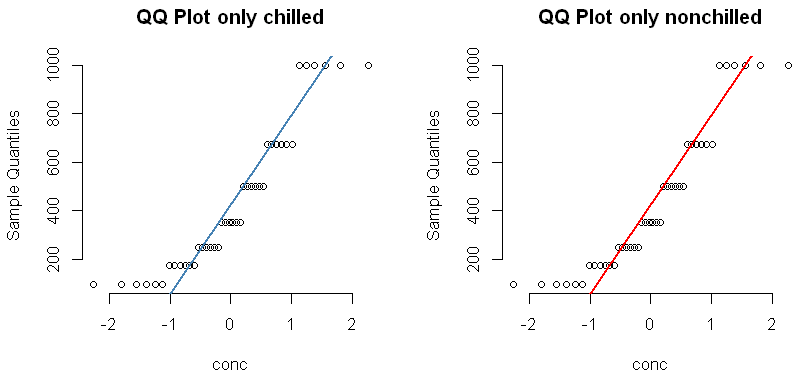
```
# A tibble: 2 × 4
  Treatment  sample_size mean_value sd_value
  <fct>            <int>      <dbl>    <dbl>
1 nonchilled          42       30.6     9.70
2 chilled             42       23.8    10.9
# A tibble: 2 × 4
  Treatment  sample_size mean_value sd_value
  <fct>            <int>      <dbl>    <dbl>
1 nonchilled          42        435     298.
2 chilled             42        435     298.
```

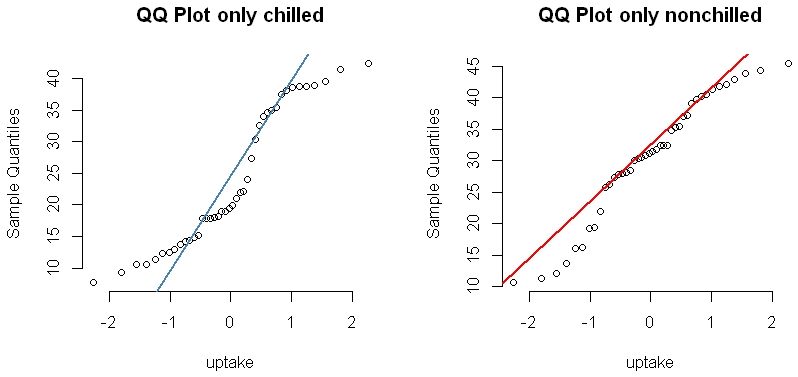


Die Überprüfung der Varianz ist bei uptake mit 9.7 und 10.9 ähnlich, bei conc gleich.

In [49]:
```r
# Stichproben sind annährend normalverteilt, mit QQ Plot und Shapiro-Wilk
# Erstellung der Data frames
chilled <- data_co2 %>% filter(data_co2$Treatment == "chilled")
nonchilled <- data_co2 %>% filter(data_co2$Treatment == "nonchilled")

par(mfrow = c(2, 2))
# QQ  Plots
qqnorm(chilled$conc, pch = 1, frame = FALSE, main = "QQ Plot only chilled",
     xlab = "conc")
qqline(chilled$conc, col = "steelblue", lwd = 2)
qqnorm(nonchilled$conc, pch = 1, frame = FALSE,  main = "QQ Plot only nonchilled",
     xlab = "conc")
qqline(nonchilled$conc, col = "red", lwd = 2)
qqnorm(chilled$uptake, pch = 1, frame = FALSE,  main = "QQ Plot only chilled",
     xlab = "uptake")
qqline(chilled$uptake, col = "steelblue", lwd = 2)
qqnorm(nonchilled$uptake, pch = 1, frame = FALSE,   main = "QQ Plot only nonchilled
```

```
     xlab = "uptake")
qqline(nonchilled$uptake, col = "red", lwd = 2)
```

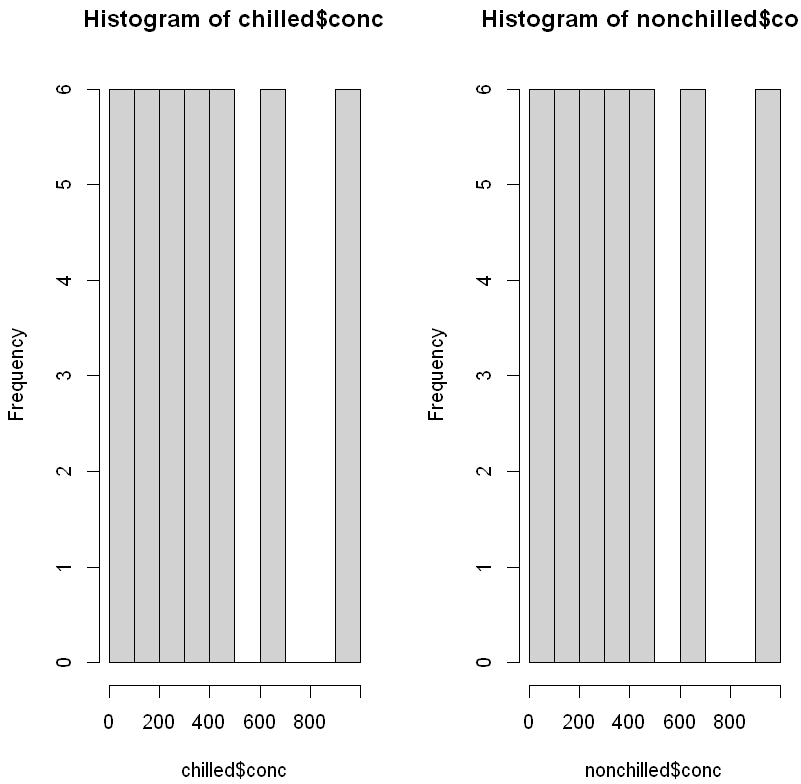




Die Normalverteilung ist eher nicht gegeben.

In [50]:
```
# Shapiro-Wilk mit Histogramm für Spalte conc
par(mfrow = c(1, 2))
hist(chilled$conc)
shapiro.test(chilled$conc)
hist(nonchilled$conc)
shapiro.test(nonchilled$conc)
```

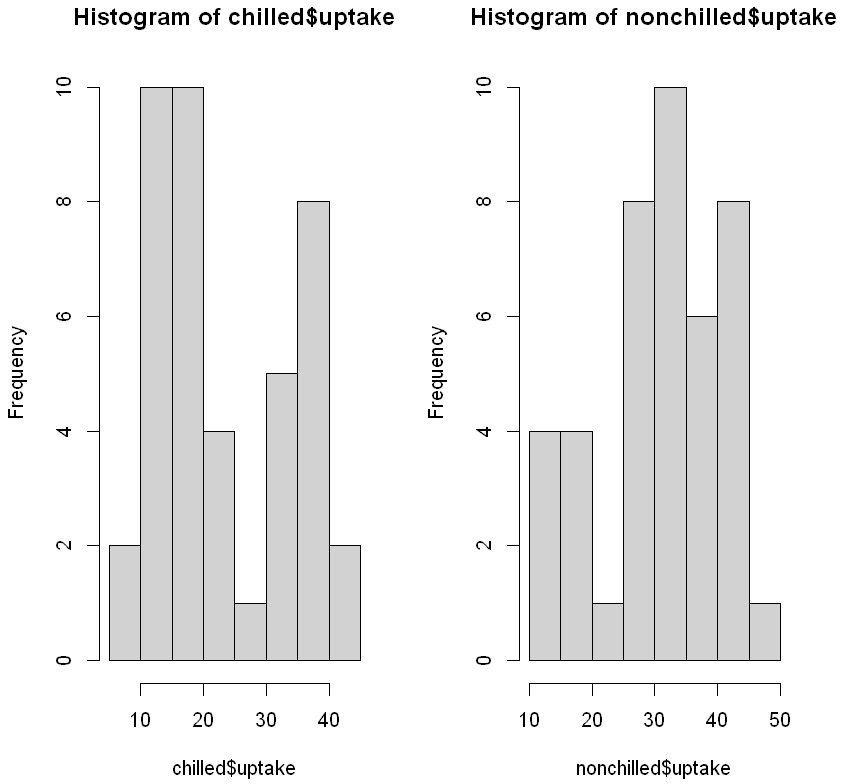
```
        Shapiro-Wilk normality test

data:  chilled$conc
W = 0.87236, p-value = 0.0002367
        Shapiro-Wilk normality test

data:  nonchilled$conc
W = 0.87236, p-value = 0.0002367
```



Für die Spalte conc zeigt weder das Histogramm noch der Shapiro mit p-value = 0.0002367 eine Normalverteilung.

In [52]:
```
# Shapiro-Wilk mit Histogramm für Spalte uptake
par(mfrow = c(1, 2))
hist(chilled$uptake)
shapiro.test(chilled$uptake)
hist(nonchilled$uptake)
shapiro.test(nonchilled$uptake)
```

```
        Shapiro-Wilk normality test

data:  chilled$uptake
W = 0.89789, p-value = 0.001245
        Shapiro-Wilk normality test

data:  nonchilled$uptake
W = 0.94505, p-value = 0.04302
```



Für die Spalte uptake zeigt weder das Histogramm noch der Shapiro mit p-value = 0.001245 und p-value = 0.04302 eine Normalverteilung. Hier gibt es zwei unterschiedliche p-values weil sich die Werte hier nicht wiederholen. Bei conc wiederholen sich diese Werte ja.

Resultat Annahmen für einen unabhängigen t-Test überprüfen

- Ja, die zu untersuchenden Spalten sin numerisch.
- Die Daten sind unabhänig voneinander. Mit dem Scatterplot kann man kein Muster erkennen. Die Werte 0.4851774 (conc und uptake), 0.4277141 (Subset chilled conc und uptake), 0.6081648 (Subset nonchilled conc und uptake) deuten auf eine geringe Korrelation hin. Trotzdem kann man annehmen, dass bei der Aufnahme von Co2 auch die Konzentration steigt, dies für bei der Kühlung oder bei keiner Kühlung. Es gibt einen Zusammenhang.
- df_summary1 der Mittelwert ist unterschiedlich, anhand des Boxplots erkennt man eine ähnliche Streuung, es scheint, dass die "gleiche Varianz" Annahme nicht verletzt ist. Dies mit den Werten 9.7 und 10.9.
- df_summary2 zeigt gleiche Boxplots, weil die Werte sich wiederholen in der conc Spalte. Die ähnliche Streuung ist hier klar, die "gleiche Varianz" Annahme ist nicht verletzt. Dies mit den Werten 298 und 298.
- Die QQ Plot deuten eher auf keine Normalverteilung. Auch hier wird für chilled und nonchilled mit den Spalten conc und uptake überprüft.

- Für die Spalte conc gibt das Histogramm an, dass es sich nicht um eine Normalverteilung handelt, der Shapiro mit p-value = 0.0002367 verwirft die Nullhypothese ebenfalls, es ist keine Normalverteilung. Es sei angemerkt, dass beide Histogramme und Shapiros gleich sind, das verwundert nicht, die Werte wiederholen sich ja.
- Für die Spalte uptake gibt das Histogramm an, dass es sich nicht um eine typische Normalverteilung handelt (bimodal?), der Shapiro mit p-value = 0.001245 und p-value = 0.04302 verwirft ebenfalls die Normalverteilung.
- Des Weiteren: Hier habe ich nur Treatment berücksichtigt, ohne Type.
- Die folgende Log und Square Transformation liefern keine besseren Resultate. Wir haben aber annährend Normalverteilung und könnten p = 0.01 wählen, sollte man den Fehler tolerieren wollen. Es wird nun mit dem Datensatz gearbeitet, ohne Transformation.

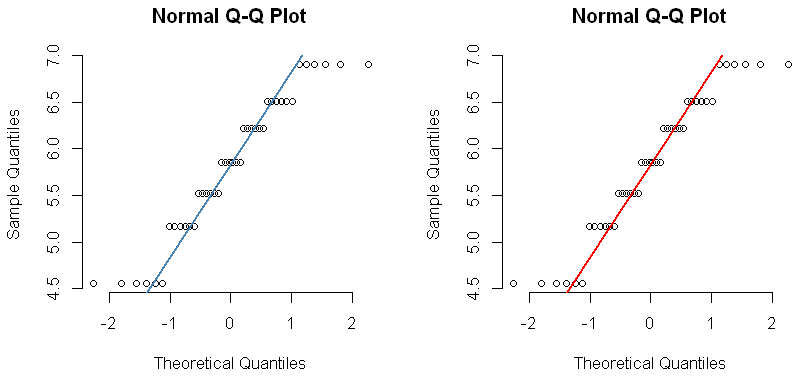
Transformation Log

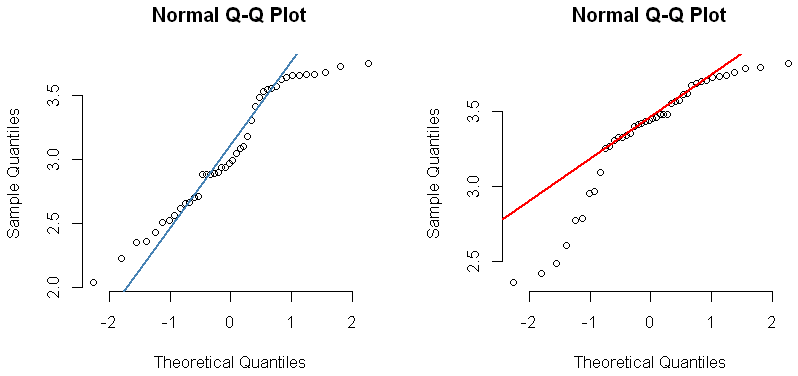
Mit der Log Tansformation soll gesehen werden, ob sich die Resultate verbessern.

In [53]:
```
# Log
data_co2$log_conc <- log(data_co2$conc)
data_co2$log_uptake <- log(data_co2$uptake)
chilled$log_conc <- log(chilled$conc)
chilled$log_uptake <- log(chilled$uptake)
nonchilled$log_conc <- log(nonchilled$conc)
nonchilled$log_uptake <- log(nonchilled$uptake)

par(mfrow = c(2, 2))
qqnorm(chilled$log_conc, pch = 1, frame = FALSE)
qqline(chilled$log_conc, col = "steelblue", lwd = 2)
qqnorm(nonchilled$log_conc, pch = 1, frame = FALSE)
qqline(nonchilled$log_conc, col = "red", lwd = 2)
qqnorm(chilled$log_uptake, pch = 1, frame = FALSE)
qqline(chilled$log_uptake, col = "steelblue", lwd = 2)
qqnorm(nonchilled$log_uptake, pch = 1, frame = FALSE)
qqline(nonchilled$log_uptake, col = "red", lwd = 2)

# Shapiro-Wilk mit Histogramm für Spalte conc
hist(chilled$log_conc)
shapiro.test(chilled$log_conc)
hist(nonchilled$log_conc)
shapiro.test(nonchilled$log_conc)

# Shapiro-Wilk mit Histogramm für Spalte uptake
hist(chilled$log_uptake)
shapiro.test(chilled$log_uptake)
```

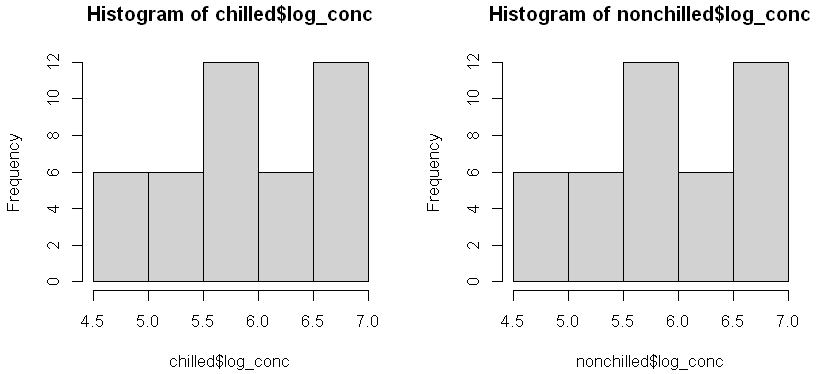
```
hist(nonchilled$log_uptake)
shapiro.test(nonchilled$log_uptake)
```

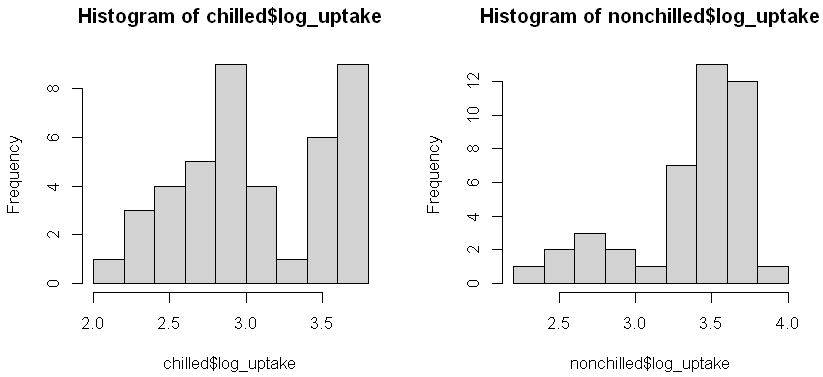
```
        Shapiro-Wilk normality test

data:  chilled$log_conc
W = 0.92636, p-value = 0.009783
        Shapiro-Wilk normality test

data:  nonchilled$log_conc
W = 0.92636, p-value = 0.009783
        Shapiro-Wilk normality test

data:  chilled$log_uptake
W = 0.93472, p-value = 0.01876
```






```
        Shapiro-Wilk normality test

data:  nonchilled$log_uptake
W = 0.86729, p-value = 0.0001733
```





Resultat Transforamtion Log

- Log verbessert den Datensatz nicht wirklich. Die Shapiros p-value = 0.009783, p-value = 0.01876, p-value = 0.0001733 verwerfen die Normalverteilung.
- ODER: Für chilled uptake und nonchilled uptake sieht es gemäss den Histogrammen nach einer bimodal normal distribution aus, somit wäre die Normalverteilung ebenfalls erreicht. Ob dies so zu verwenden ist, kann untersucht werden.

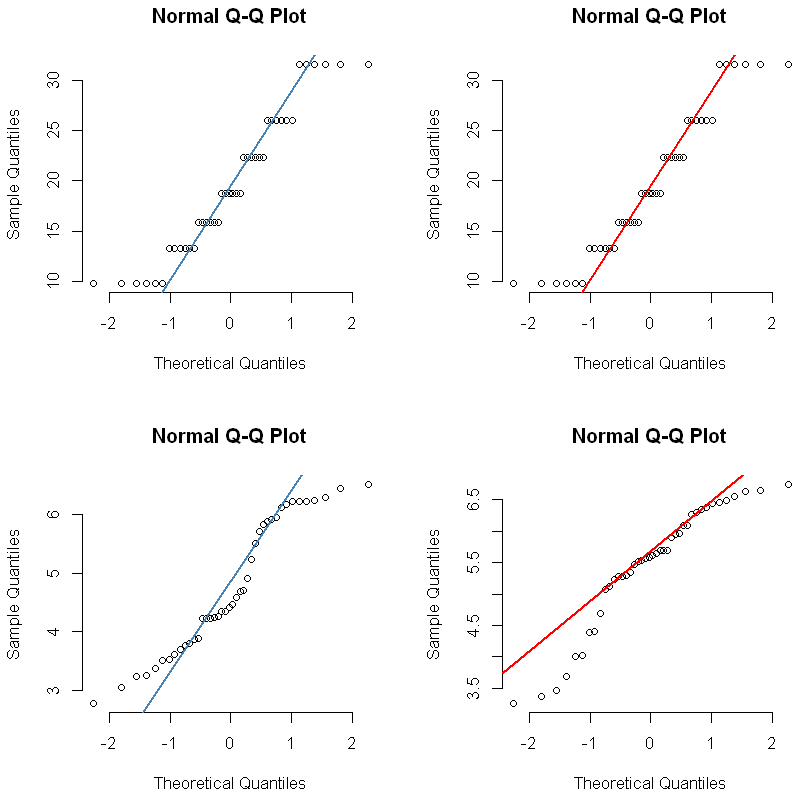
Transformation Square

Mit der Square Tansformation soll gesehen werden, ob sich die Resultate verbessern.

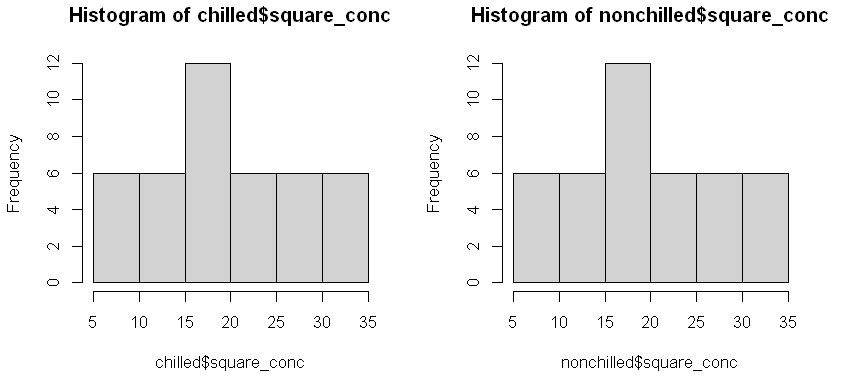
In [54]:
```
# Square
data_co2$square_conc <- sqrt(data_co2$conc)
data_co2$square_uptake <- sqrt(data_co2$uptake)
chilled$square_conc <- sqrt(chilled$conc)
chilled$square_uptake <- sqrt(chilled$uptake)
nonchilled$square_conc <- sqrt(nonchilled$conc)
nonchilled$square_uptake <- sqrt(nonchilled$uptake)

par(mfrow = c(2, 2))
qqnorm(chilled$square_conc, pch = 1, frame = FALSE)
qqline(chilled$square_conc, col = "steelblue", lwd = 2)
qqnorm(nonchilled$square_conc, pch = 1, frame = FALSE)
qqline(nonchilled$square_conc, col = "red", lwd = 2)
```

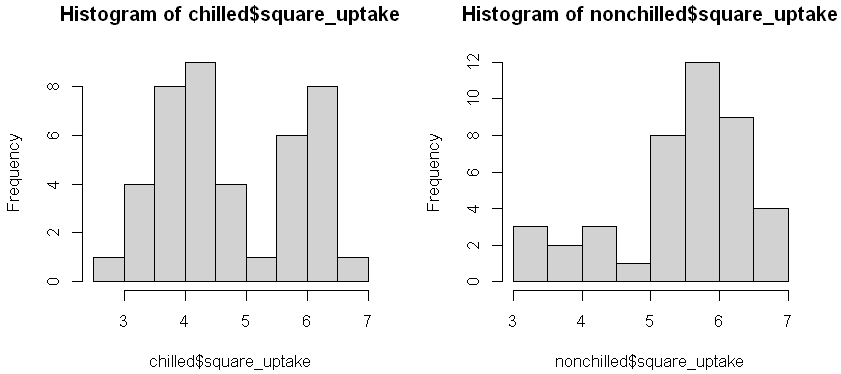
```r
qqnorm(chilled$square_uptake, pch = 1, frame = FALSE)
qqline(chilled$square_uptake, col = "steelblue", lwd = 2)
qqnorm(nonchilled$square_uptake, pch = 1, frame = FALSE)
qqline(nonchilled$square_uptake, col = "red", lwd = 2)

# Shapiro-Wilk mit Histogramm für Spalte conc
hist(chilled$square_conc)
shapiro.test(chilled$square_conc)
hist(nonchilled$square_conc)
shapiro.test(nonchilled$square_conc)

# Shapiro-Wilk mit Histogramm für Spalte uptake
hist(chilled$square_uptake)
shapiro.test(chilled$square_uptake)
hist(nonchilled$square_uptake)
shapiro.test(nonchilled$square_uptake)
```

```
        Shapiro-Wilk normality test

data:  chilled$square_conc
W = 0.92088, p-value = 0.00646
        Shapiro-Wilk normality test

data:  nonchilled$square_conc
W = 0.92088, p-value = 0.00646
        Shapiro-Wilk normality test

data:  chilled$square_uptake
W = 0.92156, p-value = 0.006797
```



```
        Shapiro-Wilk normality test

data:  nonchilled$square_uptake
W = 0.91392, p-value = 0.003862
```





Resultat Transformation Square

- Square verbessert den Datensatz nicht wirklich. Die Shapiros p-value = 0.00646, p-value = 0.006797, p-value = 0.003862 verwerfen die Normalverteilung.
- ODER: Für chilled uptake und nonchilled uptake sieht es gemäss den Histogrammen nach einer bimodal normal distribution aus, somit wäre die Normalverteilung ebenfalls erreicht. Ob dies so zu verwenden ist, kann untersucht werden.

Ein-Stichproben t-Test

Hier wird überprüft ob sich der Mittelwert stark von einem theoretischen Wert unterscheidet.

In [55]:
```r
# Einstichprobentest mit theoretischem Mittelwert Abgleich
t.test(data_co2$uptake, mu=15)
t.test(data_co2$uptake, mu=20)
t.test(data_co2$uptake, mu=25)
t.test(data_co2$uptake, mu=27)
t.test(data_co2$uptake, mu=30)
t.test(data_co2$uptake, mu=35)
t.test(data_co2$uptake, mu=100)
```

```
        One Sample t-test

data:  data_co2$uptake
t = 10.351, df = 83, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 15
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
        One Sample t-test

data:  data_co2$uptake
t = 6.1131, df = 83, p-value = 3.053e-08
alternative hypothesis: true mean is not equal to 20
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
        One Sample t-test

data:  data_co2$uptake
t = 1.8756, df = 83, p-value = 0.06423
alternative hypothesis: true mean is not equal to 25
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
        One Sample t-test

data:  data_co2$uptake
t = 0.1806, df = 83, p-value = 0.8571
alternative hypothesis: true mean is not equal to 27
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
        One Sample t-test

data:  data_co2$uptake
t = -2.3619, df = 83, p-value = 0.02052
alternative hypothesis: true mean is not equal to 30
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
```

```
        One Sample t-test

data:  data_co2$uptake
t = -6.5994, df = 83, p-value = 3.64e-09
alternative hypothesis: true mean is not equal to 35
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
        One Sample t-test

data:  data_co2$uptake
t = -61.686, df = 83, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 100
95 percent confidence interval:
 24.86622 29.55997
sample estimates:
mean of x
  27.2131
```

Resultat Ein-Stichproben t-Test

- Bei mu=25 und mu=27 verwerfen wir die Nullhypothese nicht, der angenommene theoretische Mittelwert der Population scheint zu stimmen, es gibt keinen signifikanten Unterschied. Mit p-values 0.06423 und 0.8571 nehmen wir die Nullhypothese an. Das Konfidenzintervall liegt bei 24.86622 29.55997.
- Man erkennt die Zunahme und Abbnahme des p-values bei den ausgewählten theoretischen Populationsmittelwerten.

Zwei-Stichproben t-Test, unabhängig

Hier wird untersucht, ob sich die Mittelwerte von zwei Stichproben sich signifikant unterscheiden.

In [56]:
```r
# gleiche Varianz
t.test(data_co2$uptake ~ data_co2$Treatment, var.equal = TRUE, alternative= "two.si
t.test(data_co2$uptake ~ data_co2$Treatment, var.equal = TRUE, alternative= "less")
t.test(data_co2$uptake ~ data_co2$Treatment, var.equal = TRUE, alternative= "greate
```

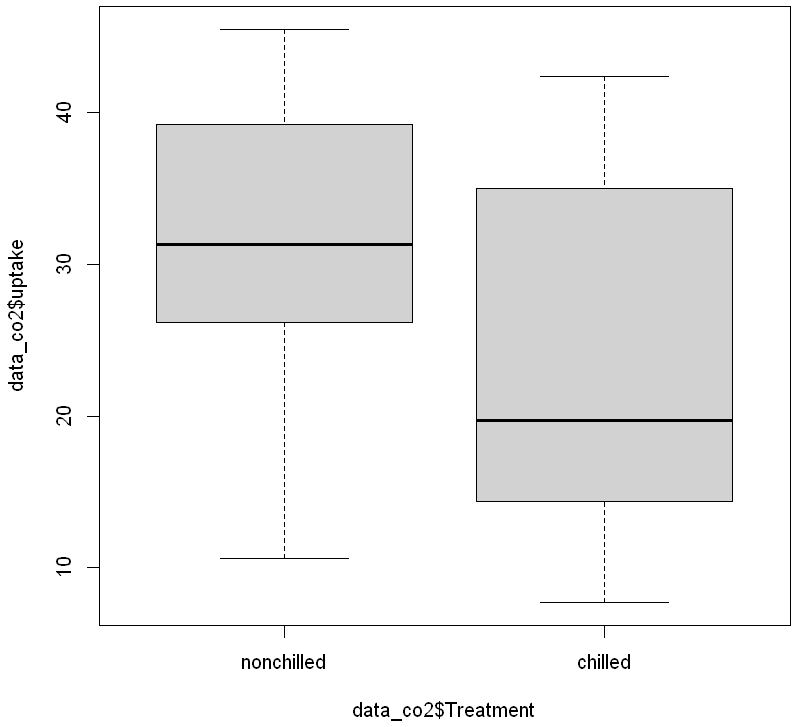
```
        Two Sample t-test

data:  data_co2$uptake by data_co2$Treatment
t = 3.0485, df = 82, p-value = 0.003096
alternative hypothesis: true difference in means between group nonchilled and group
chilled is not equal to 0
95 percent confidence interval:
  2.38324 11.33581
sample estimates:
mean in group nonchilled    mean in group chilled
                30.64286                 23.78333
        Two Sample t-test

data:  data_co2$uptake by data_co2$Treatment
t = 3.0485, df = 82, p-value = 0.9985
alternative hypothesis: true difference in means between group nonchilled and group
chilled is less than 0
95 percent confidence interval:
   -Inf 10.603
sample estimates:
mean in group nonchilled    mean in group chilled
                30.64286                 23.78333
        Two Sample t-test

data:  data_co2$uptake by data_co2$Treatment
t = 3.0485, df = 82, p-value = 0.001548
alternative hypothesis: true difference in means between group nonchilled and group
chilled is greater than 0
95 percent confidence interval:
 3.116048      Inf
sample estimates:
mean in group nonchilled    mean in group chilled
                30.64286                 23.78333
```

In [57]:
```r
# ungleiche Varianz
t.test(data_co2$uptake ~ data_co2$Treatment, var.equal = FALSE, alternative= "two.s
t.test(data_co2$uptake ~ data_co2$Treatment, var.equal = FALSE, alternative= "less"
t.test(data_co2$uptake ~ data_co2$Treatment, var.equal = FALSE, alternative= "great
```

```
        Welch Two Sample t-test

data:  data_co2$uptake by data_co2$Treatment
t = 3.0485, df = 80.945, p-value = 0.003107
alternative hypothesis: true difference in means between group nonchilled and group
chilled is not equal to 0
95 percent confidence interval:
  2.382366 11.336682
sample estimates:
mean in group nonchilled    mean in group chilled
                30.64286                 23.78333
```

```
        Welch Two Sample t-test

data:  data_co2$uptake by data_co2$Treatment
t = 3.0485, df = 80.945, p-value = 0.9984
alternative hypothesis: true difference in means between group nonchilled and group
chilled is less than 0
95 percent confidence interval:
     -Inf 10.60356
sample estimates:
mean in group nonchilled    mean in group chilled
                30.64286                 23.78333
        Welch Two Sample t-test

data:  data_co2$uptake by data_co2$Treatment
t = 3.0485, df = 80.945, p-value = 0.001553
alternative hypothesis: true difference in means between group nonchilled and group
chilled is greater than 0
95 percent confidence interval:
 3.11549     Inf
sample estimates:
mean in group nonchilled    mean in group chilled
                30.64286                 23.78333
```

In [58]:
```
boxplot(data_co2$uptake~data_co2$Treatment, data=data_co2)
```



Resultat Zwei-Stichproben t-Test, unabhängig

s - Ungleiche Varianz wird trotzdem getestet, auch wenn wir aus den Annahmen erkennen, dass die gleiche Varianz Annahme nicht verletzt ist.

- Mit gleicher Varianz und ungleicher Varianz:

Mit gleicher Varianz:

- p-value = 0.003096 (two sided)
- p-value = 0.9985 (less)
- p-value = 0.001548 (greater)

Mit ungleicher Varianz:

- p-value = 0.003107 (two sided)
- p-value = 0.9984 (less)
- p-value = 0.001553 (greater)
- Mit p-value = 0.9985 kann klar gesagt werden, dass bei den Gruppen chilled und nonchilled die Mittelwerte sich unterscheiden, nonchilled 30.64286 und chilled 23.78333, siehe auch Boxplot. Der umgekehre Fall ist ebenfalls korrekt, mit p-value = 0.001548 müssen wir die Nullhypothese verwerfen, denn ist der eine Wert grösser, kann er nicht auch gleichzeitig kleiner sein. Mit p-value = 0.003096 kann die Nullhypothese verworfen werden, es muss ein kleiner gleich oder grösser gleich exisitieren, die Mittelwerte können nicht gleich sein.

Gepaarter t-test, Paardifferenzentest, abhängig

In [59]:
```
t.test(data_co2$uptake[data_co2$Treatment == "chilled"], data_co2$uptake[data_co2$T
```

```
        Paired t-test

data:  data_co2$uptake[data_co2$Treatment == "chilled"] and data_co2$uptake[data_co2
$Treatment == "nonchilled"]
t = -7.939, df = 41, p-value = 8.051e-10
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -8.604458 -5.114589
sample estimates:
mean difference
      -6.859524
```

Resultat Gepaarter t-test, Paardifferenzentest, abhängig

- Es gibt mit p-value = 8.051e-10 einen signifikanten Unterschied zwischen den Mittelwerten der Gruppen.

Effect sizes

Mit der Effektstärke kann nun die Grösse des Unterschieds berechnet werden.

In [60]:
```r
# Cohen's d One-sample
# d = x - mu / s
# x: Mittelwert der Stichprobe
# mu: Mittwert der Nullhypothese
# s: Standardabweichung der Stichprobe
install.packages("rstatix")
library(rstatix)

# One-sample t-test effect size
cohens_d(formula = uptake ~ 1, data=data_co2, mu=15)
cohens_d(formula = uptake ~ 1, data=data_co2, mu=20)
cohens_d(formula = uptake ~ 1, data=data_co2, mu=25)
cohens_d(formula = uptake ~ 1, data=data_co2, mu=27)
cohens_d(formula = uptake ~ 1, data=data_co2, mu=30)
cohens_d(formula = uptake ~ 1, data=data_co2, mu=35)
cohens_d(formula = uptake ~ 1, data=data_co2, mu=100)

# Cohen's d Two indepedent samples t-test effect size
# d = x1 -x2 / sqrt((s1^2+s2^2) / 2)
# x1,x2: Mittelwert der Stichproben
# s1^2,s2^2: Varianzen der Stichproben

#  Two-sample t-test effect size
cohens_d(formula = uptake ~ Treatment, paired = FALSE, data=data_co2)

# Cohen's d Two indepedent samples t-test effect size mit den Gruppen
# d = (x1 -x2) / s
# x1,x2: Mittelwerte der Gruppe chilled und nonchilled
# s: Standardabweichng der Population
# Geht nicht!
```

```
Warning message:
"package 'rstatix' is in use and will not be installed"
```

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	1.129335	84	large

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	0.6669891	84	moderate

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	0.2046431	84	small

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	0.01970475	84	negligible

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	-0.2577028	84	small

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	-0.7200488	84	moderate

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
Cohen's d	uptake	1	null model	-6.730546	84	large

A rstatix_test: 1 × 7

	.y.	group1	group2	effsize	n1	n2	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<int>	<ord>
Cohen's d	uptake	nonchilled	chilled	0.6652288	42	42	moderate

Resultat Effect sizes

Nebst der statistischen Signifikanz soll der Unterschied mit der Effektstärke bestimt werden. Mit der Effektstärke kann die Grösse des Unterschieds bestimmt werden. Diese wird dann anschliessend füt den power-test als Input verwendet.

Resultat:

- One-sample t-test effect size Resultate:

mu=15 1.129335

mu=20 0.6669891

mu=25 0.2046431

mu=27 0.01970475

mu=30 -0.2577028

mu=35 -0.7200488

mu=100 -6.730546

abs(d) < 0.2 vernachlässigbar

0.2 <= abs(d) < 0.5 klein

0.5 <= abs(d) < 0.8 mittel

abs(d) >= 0.8 gross

- Bei mu=27 erkennt man den Unterschied "vernachlässigbar". Dies deckt sich auch mit dem Test der Signifikanz, der bei 27 ein besseres Resultat liefert.
- Die neagtiven Zahlen können ignoriert werden, der Test würde sagen, diese würde wieder besser abschneiden, aber nur weil d als absoluter Wert verwendet wird.
- Two-sample t-test effect size:

Die Effektstärke ist 0.6652288 und das kommt hin, wenn man den Test von früher berücksichtigt, es muss ein Unterschied vorhanden sein, was die Mittelwerte aus dem Test so zeigen: Nonchilled: 30.64286 Chilled: 23.78333

- Die zweite Variante des Cohen's d Two indepedent samples t-test geht nicht, da die Standardabweichung der Population nicht bekannt ist.

Trennschärfe, power of a test für den t-test

In [61]:
```r
# Power Analyse für den t-test one-sample
# d= sind die Effektstärken
power.t.test(n=84, d=1.129335, type=c("one.sample"))
power.t.test(n=84, d=0.6669891, type=c("one.sample"))
power.t.test(n=84, d=0.2046431, type=c("one.sample"))
power.t.test(n=84, d=0.01970475, type=c("one.sample"))

# Power Analyse für den t-test two-sample
power.t.test(n=84, d=0.6652288, type=c("two.sample"))
```

```
     One-sample t test power calculation

              n = 84
          delta = 1.129335
             sd = 1
      sig.level = 0.05
          power = 1
    alternative = two.sided
     One-sample t test power calculation

              n = 84
          delta = 0.6669891
             sd = 1
      sig.level = 0.05
          power = 0.9999776
    alternative = two.sided
     One-sample t test power calculation

              n = 84
          delta = 0.2046431
             sd = 1
      sig.level = 0.05
          power = 0.4577539
    alternative = two.sided
     One-sample t test power calculation

              n = 84
          delta = 0.01970475
             sd = 1
      sig.level = 0.05
          power = 0.03741997
    alternative = two.sided
     Two-sample t test power calculation

              n = 84
          delta = 0.6652288
             sd = 1
      sig.level = 0.05
          power = 0.9899945
    alternative = two.sided

NOTE: n is number in *each* group
```

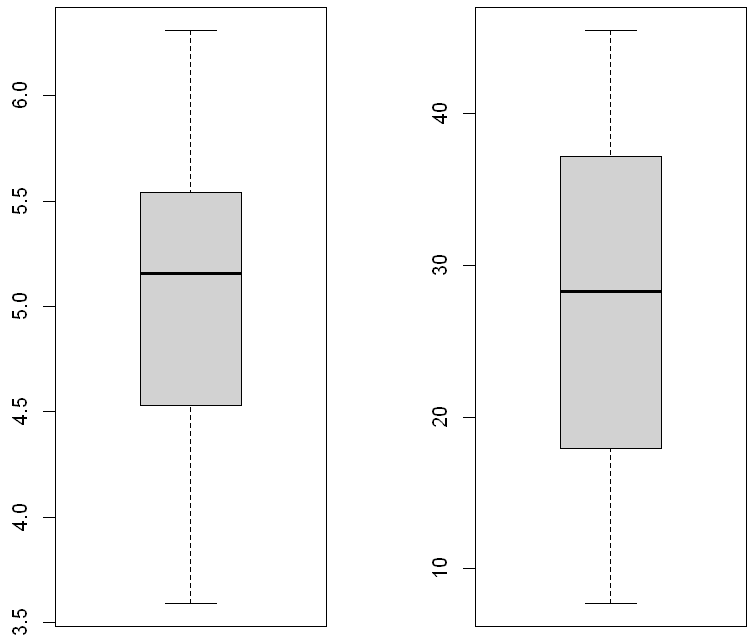
Resultat Trennschärfe, power of a test für den t-test

- Die Resultate sind ähnlich wie bei der Effektstärke und dem vorangegangenen T-test für eine Stichprobe: Mit einer Teststärke von 1 kann gesagt werden, dass ein Unterschied vorhanden ist, mit 3.741% wird kein Unterschied mehr erkannt. Die 3.741% decken sich mit dem Resultat auch vom t-Test mit mu=27.
- Bei dem Zwistichprobentest ist die Effektstärke 66.5%, somit im Bereich "Mittel", die Teststärke 98.99%, ein Unterschied kann hier tatsächlich gefunden werden, sollte einer existieren.

ANOVA (einfaktorielle Varianzanalyse)

In [62]:
```
# Neuer Datensatz für ANOVA
plant <- PlantGrowth

# Annahmen prüfen, Outliers
par(mfrow = c(1, 2))
boxplot(plant$weight, data=plant)
boxplot(data_co2$uptake, data=plant)

# AOV
aov <- aov(plant$weight~plant$group)
summary(aov)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
plant$group  2  3.766  1.8832   4.846 0.0159 *
Residuals   27 10.492  0.3886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



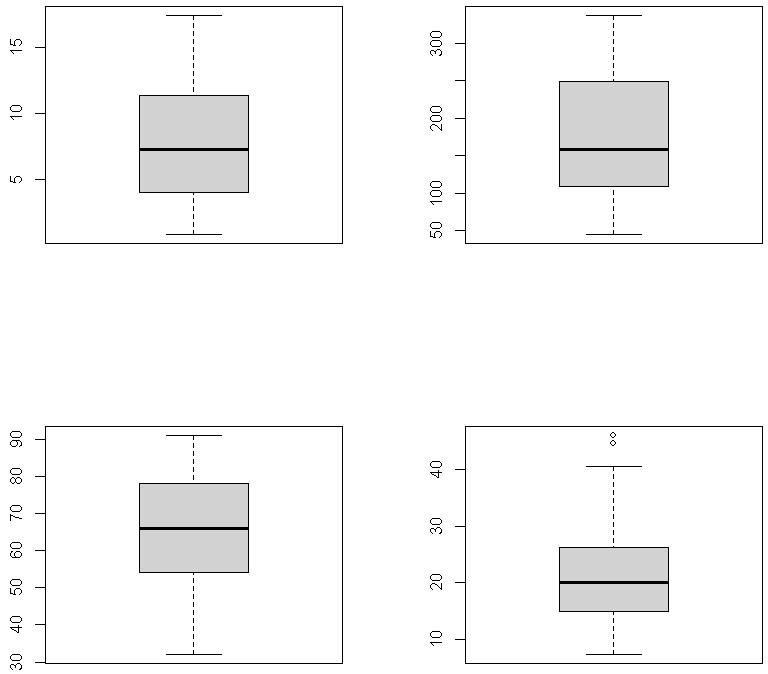
In [63]:
```
# Bonferroni Anpassung
pairwise.t.test(plant$weight,plant$group, p.adjust="bonferroni")
```

```
        Pairwise comparisons using t tests with pooled SD

data:  plant$weight and plant$group

     ctrl  trt1
trt1 0.583 -
trt2 0.263 0.013

P value adjustment method: bonferroni
```

Resultat ANOVA (einfaktorielle Varianzanalyse) auf Datensatz plant

- Es gibt keine Outliers, somit sind alle Annahmen (siehe zuoberst + Outlier) unverletzt. Im Boxplot sind keine Outliers zu erkennen.
- (AOV) Mit einem Wert von 0.0159 kann die Nullhypothese verworfen werden, die Gleichheit der Mittelwerte über die Gruppen hinweg kann nicht vorkommen.
- Mit Bonferroni vorallem kann nun bestimmt werden, welche Gruppen sich näher sind und welche nicht. Mit 0.583 und 0.263 kann die Annahme von Nähe nicht verworfen werden, mit 0.013 schon. Ähnlich sind somit nur ctrl und trt1 und ctrl trt2.

In [64]:
```r
# Lineare Regression und ANOVA, neuer Datensatz
USArrests <- USArrests

# Annahmen prüfen, Outliers
par(mfrow = c(2, 2))
boxplot(USArrests$Murder, data=USArrests)
boxplot(USArrests$Assault, data=USArrests)
boxplot(USArrests$UrbanPop, data=USArrests)
boxplot(USArrests$Rape, data=USArrests)
```



In [65]:
```r
# LM Model
lm_model2 <- lm(Murder ~ Assault + UrbanPop + Rape, data=USArrests)
summary(lm_model2)

# AOV
USArrests.aov <- aov(lm_model2)
summary(USArrests.aov)
```

```
Call:
lm(formula = Murder ~ Assault + UrbanPop + Rape, data = USArrests)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3990 -1.9127 -0.3444  1.2557  7.4279

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.276639   1.737997   1.885   0.0657 .
Assault      0.039777   0.005912   6.729 2.33e-08 ***
UrbanPop    -0.054694   0.027880  -1.962   0.0559 .
Rape         0.061399   0.055740   1.102   0.2764
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.574 on 46 degrees of freedom
Multiple R-squared:  0.6721,    Adjusted R-squared:  0.6507
F-statistic: 31.42 on 3 and 46 DF,  p-value: 3.322e-11
            Df Sum Sq Mean Sq F value   Pr(>F)
Assault      1  597.7   597.7  90.195 2.05e-12 ***
UrbanPop     1   19.0    19.0   2.864   0.0974 .
Rape         1    8.0     8.0   1.213   0.2764
Residuals   46  304.8     6.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Resultat ANOVA (einfaktorielle Varianzanalyse) lineare Regression

- Es gibt keine Outliers (nur zwei bei rape, sind vernachlässigbar), somit sind alle Annahmen unverletzt. Im Boxplot sind keine Outliers zu erkennen.
- Die Variable Assault ist dem Modell am wichtigsten. Der p-value 3.322e-11 bei dem AOV sagt, dass die Mittelwerte über den Gruppen nicht gleich sein kann.

Beschreibung Ergebnisse Hypothesentest und Varianzanalyse

Die Ergebnisse kommen alle bei den bereits erwähnten Resultaten vor.

Analyse/Hinterfragen Ergebnisse Hypothesentest und Varianzanalyse

Die Analyse der Ergebnisse kommt bei den bereits erwähnten Resultaten vor.

Quellenangaben

- Daten von R (24.12.2024)
- Einfaktorielle Varianzanalyse (ANOVA) in R rechnen, https://bjoernwalther.com/einfaktorielle-varianzanalyse-anova-in-r-rechnen/ (13.10.2024)
- Chapter 3 Effect sizes, https://bookdown.org/content/f9d035ed-86ea-4779-ad01-31acc973f0dd/3-effect-sizes.html (13.10.2024)
- How to Interpret Cohen's d (With Examples), https://www.statology.org/interpret-cohens-d/ (13.10.2024)
- Quellen FFHS Videos, (13.10.2024)

In [ ]:

Hauptkomponenten und Faktorenanalyse

Einleitung Hauptkomponenten

Vorgehensweise Hauptkomponentenanalyse (stark an dem Buch orientiert, bei manchen Zeilen ist head verwendet worden, um das Maximum von 100 Seiten eher erreichen zu können.):

- Daten einlesen
- Hauptkomponentenanalyse vornehmen (mit und ohne princomp):
    - Liegt ein hochdimensionaler Datensatz vor, der in einem niedrigdimensionalen Raum dargestellt werden soll.
    - Erfüllen die Daten die Voraussetzungen (mindestens eines davon): Alle Merkmale sind quantitativ, die Daten liegen als Varianz-Kovarianz-Matrix oder Korrelationsmatrix vor.
    - Die Hauptkomponentenanalyse soll auf Basis der Varianz-Kovarianz-Matrix oder Korrelatiosnmatrix durchgeführt werden.
    - Die Eigenwerte und Eigenvektoren werden bestimmt.
    - Auswahl der Hauptkomponenten: Anteil der Gesamtstreuung, die durch die Hauptkomponenten erklärt wird, Kaiser-Kriterium, Jolliffe-Kriterium, Scree-Plot
    - Darstellung der minimal spannenden Bäume, grafische Beuteilung in R^2.
    - Hauptkomponenten interpretieren.

Statistisches Vorgehen Hauptkomponenten

Vorgehensweise ohne princomp Varianz

Die Kovarianzmatrix und Korrelationsmatrix in Vergleich bringen.

In [10]:
```
# Daten einlesen
# Verwendung des Datensatzes rock
rock_data <- data.frame(rock)
head(rock_data)
# Berechnung der Varianz-Kovarianzmatrix
```

```
cov(rock_data)

# Berechnung der Korrelationsmatrix
cor(rock_data)
```

A data.frame: 6 × 4

	area	peri	shape	perm
	<int>	<dbl>	<dbl>	<dbl>
1	4990	2791.90	0.0903296	6.3
2	7002	3892.60	0.1486220	6.3
3	7558	3930.66	0.1833120	6.3
4	7352	3869.32	0.1170630	6.3
5	7943	3948.54	0.1224170	17.1
6	7979	4010.15	0.1670450	17.1

A matrix: 4 × 4 of type dbl

	area	peri	shape	perm
area	7203044.71232	3160367.49330	-40.820823047	-466063.55213
peri	3160367.49330	2049653.68934	-51.775231267	-463032.47715
shape	-40.82082	-51.77523	0.006971657	20.35164
perm	-466063.55213	-463032.47715	20.351635275	191684.79915

A matrix: 4 × 4 of type dbl

	area	peri	shape	perm
area	1.0000000	0.8225064	-0.1821611	-0.3966370
peri	0.8225064	1.0000000	-0.4331255	-0.7387158
shape	-0.1821611	-0.4331255	1.0000000	0.5567208
perm	-0.3966370	-0.7387158	0.5567208	1.0000000

Resultat Vorgehensweise ohne princomp Varianz

- Die Varianzen sind: 7203044.71232, 2049653.68934, 0.006971657, 191684.79915. Da sich diese stark unterscheiden, wird die Korrelationsmatrix für die weiteren Berechnungen verwendet.

Vorgehensweise ohne princomp Kriterien

Ohne dem princomp Befehl sollen nun die Hauptkomponenten bestimmt werden, die den Datzensatz am besten repräsentieren.

In [11]:
```
library(ggplot2)
# Eigenwerte und Eigenvektoren berechnen

eig <- eigen(cor(rock_data))$values
eig
eigenv <- eigen(cor(rock_data))$vectors
eigenv
```

2.60896149658739 · 0.922919591532206 · 0.39605650068641 · 0.0720624111939854

A matrix: 4 × 4 of type dbl

-0.4744238	-0.6046129	0.37639417	0.51739034
-0.5886286	-0.2366588	-0.06308377	-0.77040862
0.3932268	-0.7054215	-0.58863352	-0.03554881
0.5232697	-0.2842821	0.71264188	-0.37082890

Die Bestimmung der Eignwerte und Eigenvektoren sind der erste Schritt.

In [12]:
```
library(ggplot2)
# Anteil der Gesamtstreuung, die durch die Hauptkomponenten erklärt wird
first <- 100*(eig[1]/(sum(eig[1]+eig[2]+eig[3]+eig[4])))
second <- 100*((eig[1]+eig[2])/(sum(eig[1]+eig[2]+eig[3]+eig[4])))
third <- 100*((eig[1]+eig[2]+eig[3])/(sum(eig[1]+eig[2]+eig[3]+eig[4])))
fourth <- 100*((eig[1]+eig[2]+eig[3]+eig[4])/(sum(eig[1]+eig[2]+eig[3]+eig[4])))
first
second
third
fourth
```

65.2240374146849

88.2970272029901

98.1984397201504

100

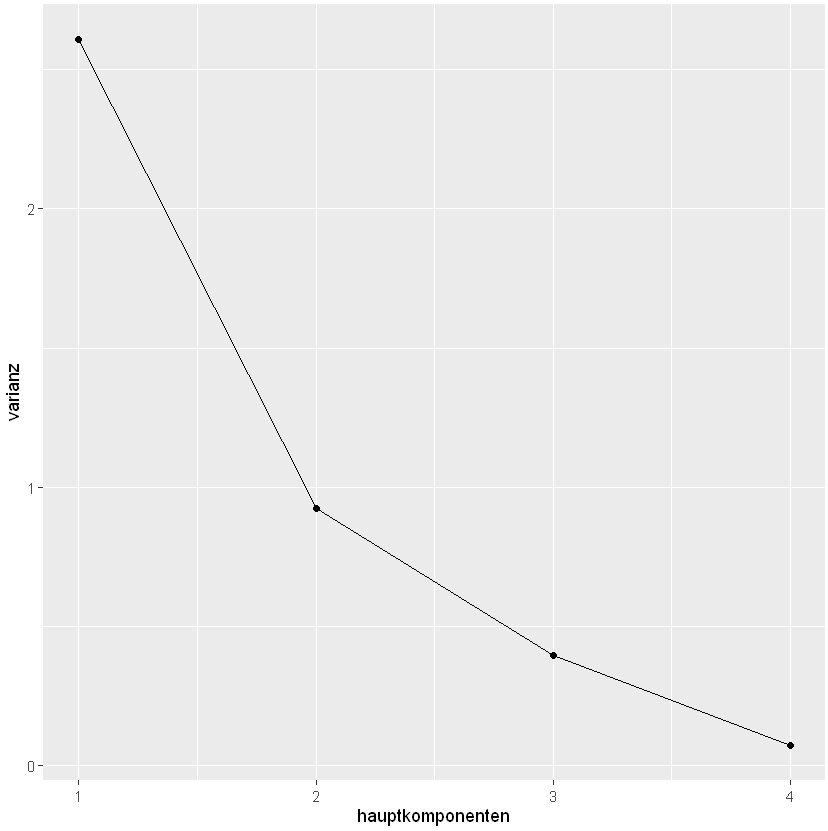
Die Werte erhöhen sich, was logisch ist. Nimmt man alle Hauptkomponenten kann man die ganze Varianz des Datensatzes repräsentieren, hat aber nichts eingespart. Mit der Hinzunahme von zwei der Hauptkomponenten kann man 88.3% der Totalvarianz beibehalten.

Dies sind die beiden Kriterien in Vergleich.

In [13]:
```
# Kaiser Kriterium, Mittelwert der Eigenwerte
kaiser <- (eig[1]+eig[2]+eig[3]+eig[4])/4
kaiser

# Jolliffe-Kriterum, Mittelwert * 0.7
jolliffe <- 0.7 * kaiser
jolliffe
```

0.999999999999999
0.699999999999999

In [14]:
```
# Scree-Plot
frame <- data.frame(hauptkomponenten = c(1,2,3,4), varianz = c(eig[1], eig[2], eig[

ggplot(data=frame, aes(x=hauptkomponenten, y=varianz, group=1)) +
  geom_line()+
  geom_point()
```



Resultat Vorgehensweise ohne princomp Kriterien

- Nimmt man eine Grenze von 80% an, erfüllen zwei Hauptkomponenten, eig[1] = 2.608961 und eig[2] = 0.9229196 die Voraussetzung. Mit der Variable second kann man dies erfüllen: second = 88.29703.
- Nimmt man das Kaiser-Kriterium an, so muss in diesem Fall der Eigenwert grösser als 1 sein, denn der Mittelwert ist 0.9999 und dies ist der einzige Eigenwert, der diesen Mittelwert überschreitet und dies mit 2.60896149658739. eig[1] = 2.608961 erfüllt diese Voraussetzung. Dies spricht für eine Hauptkomponente, man benötigt nur eine Hauptkomponente.
- Nimmt man das Jolliffe-Kriterium an, so muss in diesem Fall der Eigenwert grösser als 0.7 sein, eig[1] = 2.60896150 und eig[2] = 0.92291959 erfüllen die Voraussetzung. Diese beiden erreichen ebenfalls erneut die 88.29703%.

- Ein wirklicker Knick ist im Scree-Plot nicht zu erkennen. Meistens kann dieser Plot verwendet werden, um die zu verwendenden Hauptkomponenten bestimmen zu können. Meisten erkennt man dabei einen Knick.

Vorgehensweise mit princomp

Mit princomp Befehl sollen nun die Hauptkomponenten bestimmt werden, die den Datzensatz am besten repräsentieren.

In [15]:
```
install.packages("vegan")
library(vegan)
# Berechnung der Eigenwerte
e <- princomp(rock_data, cor=TRUE)
eig2 <-e$sdev^2
eig2
```

```
Warning message:
"package 'vegan' is in use and will not be installed"
```

Comp.1: 2.60896149658739 **Comp.2:** 0.922919591532206 **Comp.3:** 0.39605650068641
Comp.4: 0.0720624111939852

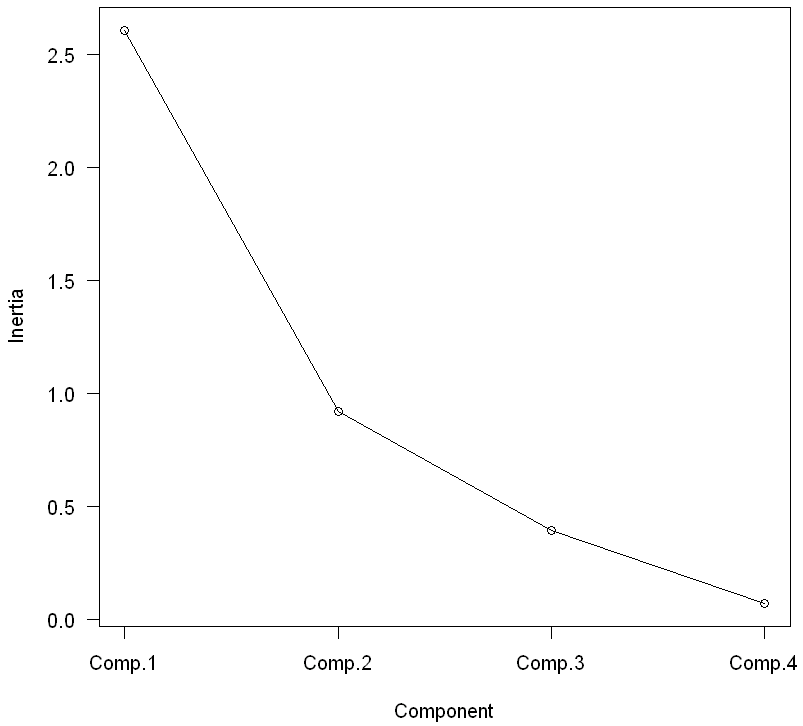
Anbei erkennt man dieselben Eigenwerte, sprich die Hauptkomponenten, wie man sie ohne princomp errechnen kann.

In [16]:
```
# Kaiser Kriterium, Mittelwert der Eigenwerte
kaiser2 <- mean(eig2)
kaiser2

# Jolliffe-Kriterum, Mittelwert * 0.7
jolliffe2 <- 0.7 * mean(eig2)
jolliffe2
```

0.999999999999999
0.699999999999999

Anbei erkennt man dieselben Werte, wie bei den Resultaten ohne princomp.

In [17]:
```
# Scree-Plot
par(las=1)
screeplot(e, type="l", main="")
```



Dieser Scree-Plot sieht genau so aus wie der Scree-Plot ohne princomp.

In [18]:
```
# Summary
summary(e)

# Die Hauptkomponenten
loadings(e)
```

```
Importance of components:
                          Comp.1    Comp.2     Comp.3    Comp.4
Standard deviation     1.6152280 0.9606870 0.62933020 0.2684444
Proportion of Variance 0.6522404 0.2307299 0.09901413 0.0180156
Cumulative Proportion  0.6522404 0.8829703 0.98198440 1.0000000
Loadings:
      Comp.1 Comp.2 Comp.3 Comp.4
area   0.474  0.605  0.376  0.517
peri   0.589  0.237        -0.770
shape -0.393  0.705 -0.589
perm  -0.523  0.284  0.713 -0.371

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

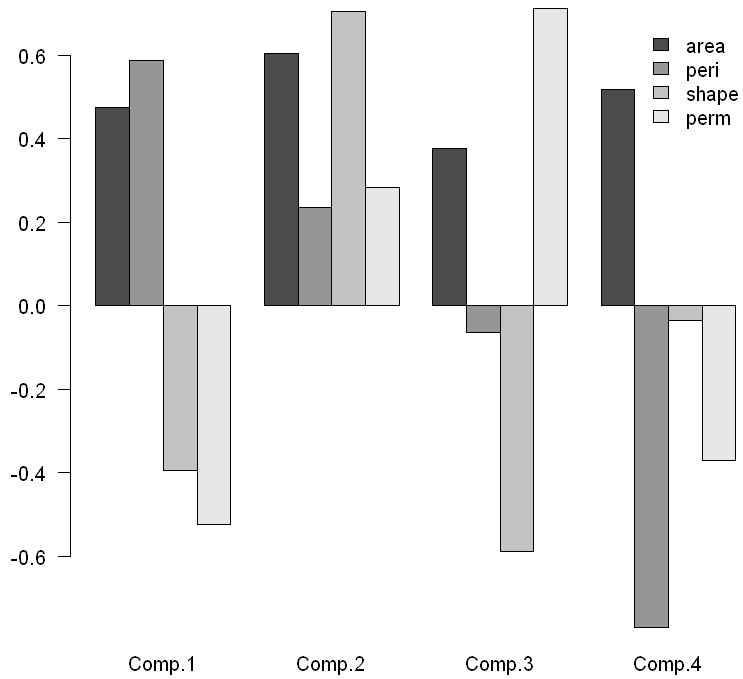
Auch hier erkennt man dieselben Werte wie bei den Berechnungen ohne princomp. Die Werte bei "Cumulative Proportion" entsprechen den Variablen first, second, third und fourth. Dies sind ebenfalls 65.22404 88.29703 98.198440 100.00000 Prozent. Die Loadings erklären

wie stark diese den Hauptkomponenten beisteuern, sprich welche am winchtigsten sind, den Datensatz in der gekürzten Version darstellen.

In [19]:
```
# Barplot mit Hauptkomponenten
par(las=1)
barplot(loadings(e), beside=TRUE, legend=colnames(rock_data), args.legend=list(bty=

# Scores
head(e$scores)
```

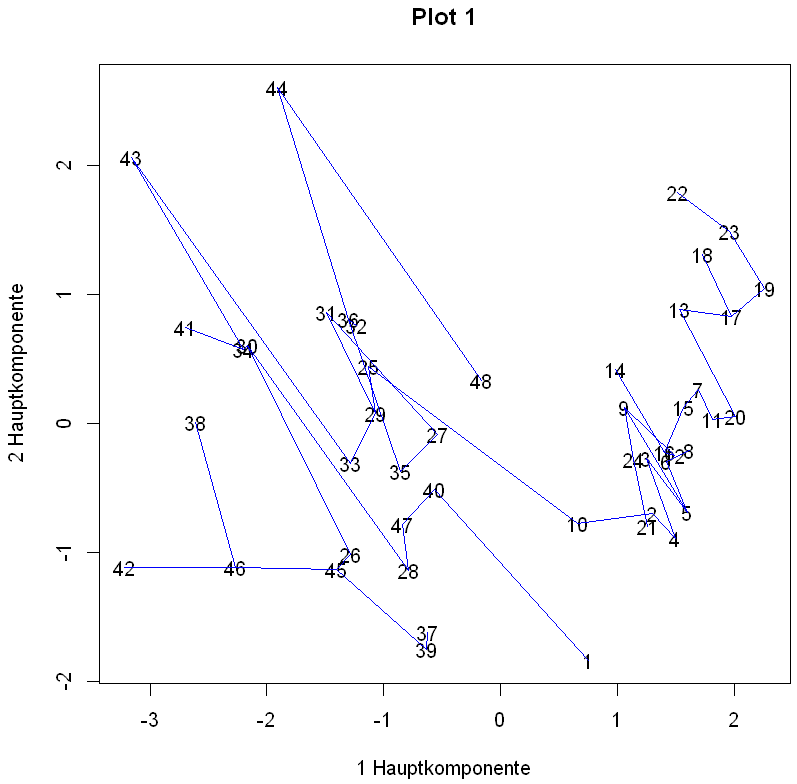
A matrix: 6 × 4 of type dbl

Comp.1	Comp.2	Comp.3	Comp.4
0.7553055	-1.8414783	-0.07902806	-0.08261715
1.2946396	-0.7018505	-0.25818326	-0.31430060
1.2446762	-0.2727316	-0.42822279	-0.24160439
1.4976909	-0.8955061	0.01729743	-0.21987529
1.5976575	-0.6949249	0.07715245	-0.15936618
1.4172878	-0.2954063	-0.23843681	-0.20505878



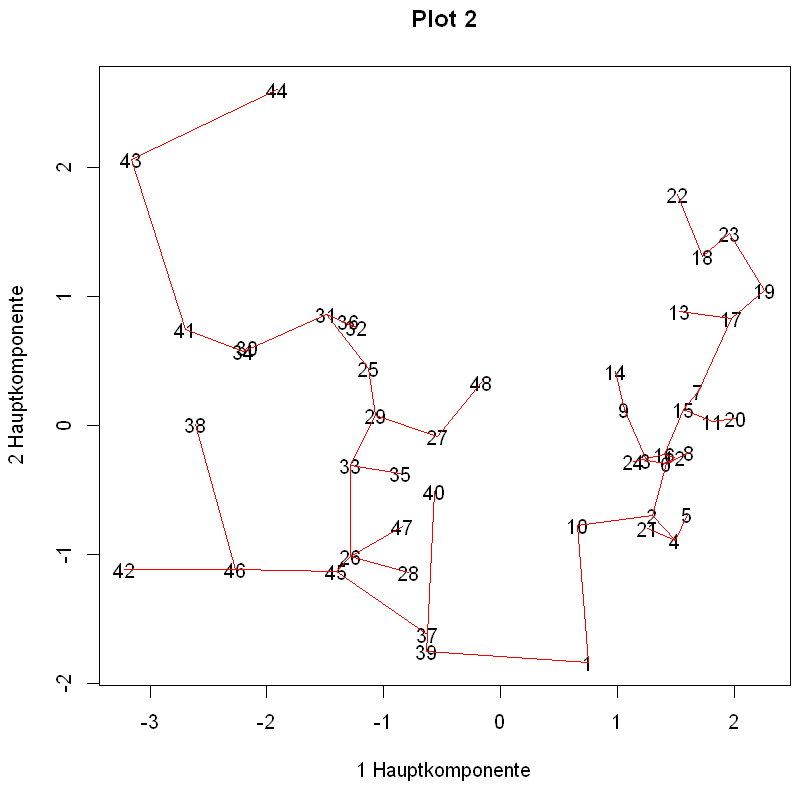
In [20]:
```
# Plot 1 der Scores in den zwei Hauptkomponenten
plot(e$scores, xlab="1 Hauptkomponente", ylab="2 Hauptkomponente", type="n", main="
text(e$scores,rownames(e$scores))

# Minimal spantree of rock_data
```

```
sp_tree <- spantree(dist(rock_data, method = "euclidean"))
distance1 <- dist(rock_data, method = "euclidean")

for (x in 2:48){
line1x <- c(e$scores[,1][x],e$scores[,1][sp_tree$kid[x-1]])
line1y <- c(e$scores[,2][x],e$scores[,2][sp_tree$kid[x-1]])
lines(line1x, line1y, pch = 18, col = "blue", type ="l", lty = 1)
}
```



Dies ist der Plot, wie es generiert wird anhand der Anleitung S. 154 des Kuhlenkasper "Multivariate Analysemethode".

In [21]:
```
# Plot 2 der Scores in den zwei Hauptkomponenten
plot(e$scores, xlab="1 Hauptkomponente", ylab="2 Hauptkomponente", type="n", main="
text(e$scores,rownames(e$scores))

# Minimal spantree of scores
sp_tree2 <- spantree(dist(e$scores, method = "euclidean"))
distance2 <- dist(e$scores, method = "euclidean")

for (x in 2:48){
line1x <- c(e$scores[,1][x],e$scores[,1][sp_tree2$kid[x-1]])
line1y <- c(e$scores[,2][x],e$scores[,2][sp_tree2$kid[x-1]])
lines(line1x, line1y, pch = 18, col = "red", type ="l", lty = 1)
}
```



Die ist der Plot unter Berücksichtigung der Scores.

Resultat Vorgehensweise mit princomp

- Mit princomp sind es dieselben Resultate, wie bei den obigen Berechnungen ohne princomp. Das heisst die Eigenwerte sind dieselben und folglich die darauf aufbauenden Berechnungen.
- Mit dem Summary kann man erkennen, dass mit den drei Hauptkomponenten 98.2% der Gesamtstreuung erklärt werden können. Mit den ersten beiden Hauptkomponenten kann man 88.3% der Gesamstreuung erklären. Mit der eigens gesetzten Grenze von 80% oder dem Jolliffe-Kriterum würden diese beiden ausreichen. Kaiser wiederum würde nur eine Hauptkomponente angeben, wie bereits erwähnt.
- Die Variable area ist stark in der zweiten Hauptkomponente vokrommende, peri bei der ersten Hauptkomponente, shape bei der zweiten Hauptkomponente und perm bei der dritten Hauptkomponente. Inwieweit die Variable perm bei der Auswahl von nur zwei Hauptkomponenten dann noch gut genug erklärt werden kann wird nicht weiter untersucht.
- Mit dem Barplot kann man erkennen, welche Variable am meisten zur Hauptkomponente beisteuert und in welcher Richtung. Bei der Hauptkomponenten 1 area und peri, die in positiver Richung gehen und shape und perm in die negative Richtung. Peri und perm haben den grössten Einfluss. Bei der Hauptkomponente 2 ist es

shape in positiver Richtung. Bei der Hauptkomponente 3 ist es perm in positiver Richtung und shape in negativer Richgung. Bei der Hauptkomponente 4 ist es peri in negativer Richtung.

- Plot 1 stellt den minimal spannenden Baum dar, den man erhält, wenn man die Distanzen anhand des Datensatzes berechnet. Plot 2 stellt den minimal spannenden Baum dar, wenn ma die Distanzen anhand der Scores berechnet. Also einmal dist(rock_data) und einmal dist(e$scores).

- Beide Plots unterscheiden sich doch sehr und somit kann anhand der grafischen Beurteilung gesagt werden, dass diese beiden Hauptkomponenten Probleme liefern und somit den Datensatz weniger gut repräsentieren. Es ist etwas schwierig zu beurteilen, weil im Buch auf Seite 148 (Handl, Kuhlenkasper) zwar auch auf Fehler eingegangen wird, aber nicht erklärt, inwieweit Fehler vorhanden sein dürfen. Es gibt keine angegebene Grenze, die klar aufzeigt, dass die Hauptkomponenten den Datensatz besser oder weniger gut repräsentieren. Somit könnte auch gesagt werden, dass der Plotvergleich positiv ausfällt.

- Mit distance1 und distance2 könnte man die Distanzmatrizen nebst der grafischen Beurteilung vergleichen.

- Die Interpretation der beiden Hauptkomponenten: Ist dieselbe wie bei dem bereits erwähnten Barplot.

Check der Spantree

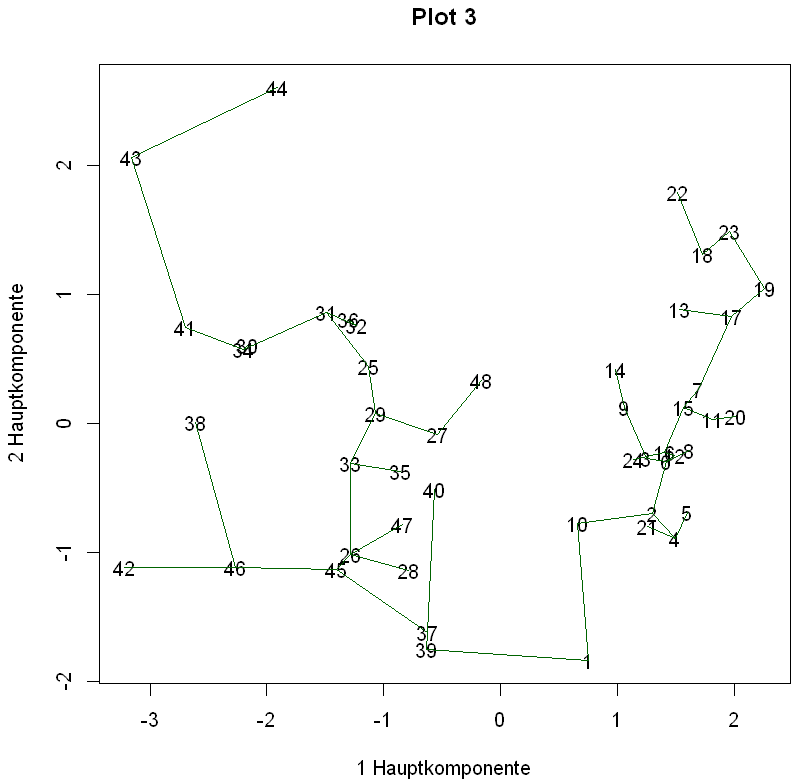
Nun soll der Datensatz normalisiert und standardisiert werden, um darin eine grafische Übereinstimmung des Plot 1 und Plot 2 zu erreichen. Die Übereinstimmung ist wichtig, damit die Qualität des PCA bewertet werden kann.

In [22]:
```
# Normalize installieren und aufrufen
install.packages("normalize")
library("normalize")

# Plot 3 der Scores in den zwei Hauptkomponenten
plot(e$scores, xlab="1 Hauptkomponente", ylab="2 Hauptkomponente", type="n", main="
text(e$scores,rownames(e$scores))

# Normalisieren (Methode standardize)
distance_c1 <- normalize(rock_data, method="standardize")

# Minimal spantree of rock_data
sp_tree3 <- spantree(dist(distance_c1, method = "euclidean"))
distance3 <- dist(distance_c1, method = "euclidean")

for (x in 2:48){
line2x <- c(e$scores[,1][x],e$scores[,1][sp_tree3$kid[x-1]])
line2y <- c(e$scores[,2][x],e$scores[,2][sp_tree3$kid[x-1]])
```

```
lines(line2x, line2y, pch = 18, col = "darkgreen", type ="l", lty = 1)
}
```

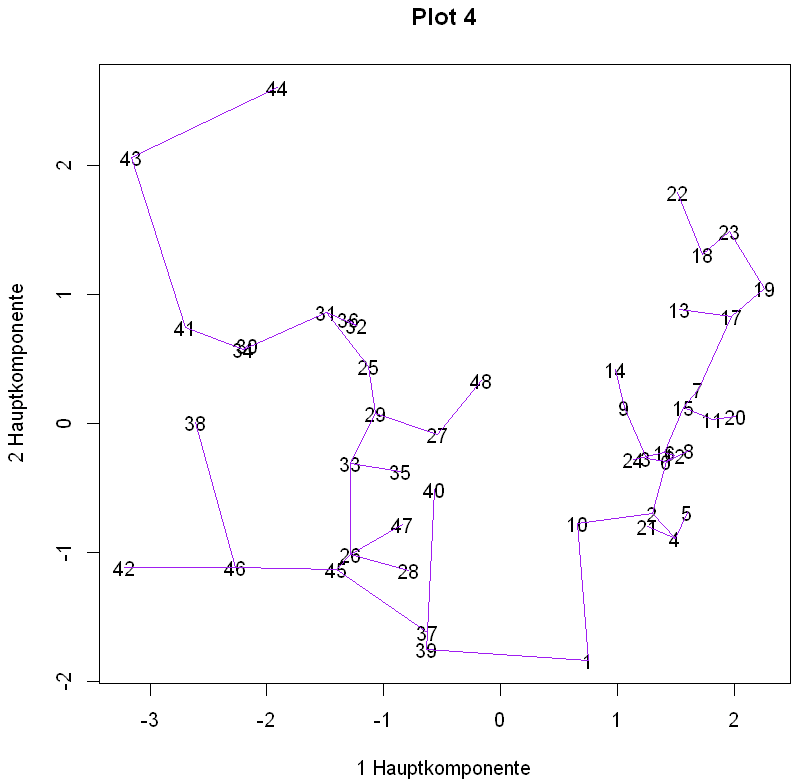
```
Installing package into 'C:/Users/olivi/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
```

```
package 'normalize' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\olivi\AppData\Local\Temp\Rtmp6hlsNn\downloaded_packages
```



Plot nach dem Normalisieren

In [23]:
```
# Plot 4 der Scores in den zwei Hauptkomponenten
plot(e$scores, xlab="1 Hauptkomponente", ylab="2 Hauptkomponente", type="n", main="
text(e$scores,rownames(e$scores))

# Standardisieren
distance_c2 <- scale(rock_data)

# Minimal spantree of rock_data
sp_tree4 <- spantree(dist(distance_c2, method = "euclidean"))
distance4 <- dist(distance_c2, method = "euclidean")

for (x in 2:48){
line3x <- c(e$scores[,1][x],e$scores[,1][sp_tree4$kid[x-1]])
line3y <- c(e$scores[,2][x],e$scores[,2][sp_tree4$kid[x-1]])
lines(line3x, line3y, pch = 18, col = "purple", type ="l", lty = 1)
    }
```



Plot nach dem Skalieren

Resultat Check der Spantree

- Nach dem Normalisieren des Datensatzes (standardize), stimmen die Plots überein. Also der Plot 2 aus dem vorangegangenen Beispiel und Plot 3 aus diesem Beispiel. Somit gibt es keine Problem oder Fehler mehr, nach dem Normalisieren.
- Nach dem Skalieren des Datensatzes (standardize), stimmen die Plots überein. Also der Plot 2 aus dem vorangegangenen Beispiel und Plot 4 aus diesem Beispiel. Somit gibt es keine Problem oder Fehler mehr, nach dem Normalisieren.
- Normalisieren und skalieren hat hier dasselbe Resultat.
- Die dazugehörige Distazmatrix kann mit distance3, distance4 abgerufen werden.

Einleitung Faktorenanalyse

Vorgehensweise Explorative Faktorenanalyse (stark an dem Buch orientiert):

- Bestimmung der Bestimmtheitsmasse
- Ersetzung der Hauptdiagonalen
- Berechung der Eigenwerte

- Index Bestimmung der Eigenwerte
- Faktoanalyse vornehmen ohne Rotation
- Faktoanalyse vornehmen mit Rotation
- Interpretation

Statistisches Vorgehen Faktorenanalyse

Explorative Faktorenanalyse

In [24]:
```
# Bestimmung der Bestimmtheitsmasse
rquadrat <- 1/diag(solve(cor(rock_data)))
rquadrat

rh <- cor(rock_data)

# Ersetzung der Hauptdiagonalen
diag(rh) <- rquadrat

# Berechung der Eigenwerte
ew <- eigen(rh)
ew

# Index Bestimmung der Eigenwerte
index <- min((1:length(ew$values))[cumsum(ew$values)>sum(ew$values)])
index
```

area: 0.219548540763207 **peri:** 0.118485331814595 **shape:** 0.670766210707031 **perm:** 0.295589717400253

```
eigen() decomposition
$values
[1]  1.9049653  0.3969537 -0.2379407 -0.7595885

$vectors
           [,1]       [,2]       [,3]       [,4]
[1,] -0.4394835 -0.53748163  0.49993926  0.51771464
[2,] -0.5385453 -0.30148074  0.01705795 -0.78663041
[3,]  0.4908081 -0.78719910 -0.37098543 -0.04236459
[4,]  0.5252910 -0.02324736  0.78239349 -0.33375050
```

1

In [25]:
```
# Faktoanalyse vornehmen ohne Rotation
f1 <- factanal(covmat=cor(rock_data), factors=1, rotation="none")
f1
f1$loadings

# Faktoanalyse vornehmen mit Rotation
f2 <- factanal(covmat=cor(rock_data), factors=1, rotation="varimax")
```

```
f2
f2$loadings
```

```
Call:
factanal(factors = 1, covmat = cor(rock_data), rotation = "none")

Uniquenesses:
 area  peri shape  perm
0.324 0.005 0.812 0.455

Loadings:
      Factor1
area   0.822
peri   0.998
shape -0.434
perm  -0.738

               Factor1
SS loadings      2.405
Proportion Var   0.601

The degrees of freedom for the model is 2 and the fit was 0.5614
Loadings:
      Factor1
area   0.822
peri   0.998
shape -0.434
perm  -0.738

               Factor1
SS loadings      2.405
Proportion Var   0.601
Call:
factanal(factors = 1, covmat = cor(rock_data), rotation = "varimax")

Uniquenesses:
 area  peri shape  perm
0.324 0.005 0.812 0.455

Loadings:
      Factor1
area   0.822
peri   0.998
shape -0.434
perm  -0.738

               Factor1
SS loadings      2.405
Proportion Var   0.601

The degrees of freedom for the model is 2 and the fit was 0.5614
```

```
Loadings:
      Factor1
area   0.822
peri   0.998
shape -0.434
perm  -0.738

               Factor1
SS loadings      2.405
Proportion Var   0.601
```

Resultat Explorative Faktorenanalyse

- Gemäss der Bestimmung des Indexes kann gesehen werden, dass nur ein Faktor zu verwenden ist. Die Faktoranalyse mit und ohne Rotation bestätigt dies, beide Resultate sind dieselben.
- Factor1 ist stark postiv assoziiert mit peri und area, stark negativ assoziiet mit perm. 60.1% der Varianz kann mit diesem einen Faktor erhalten/erklärt werden.
- PCA eröffnet mehr Möglichkeiten, sei dies bei einer Auswahl einer eigenen Grenze, also beispielweise die erwähnten 80%, das Kaiser-Kriterium, oder das Jolliffe-Kriterium. Mit PCA bewegen wir und zwischen der Auswahl einer Hauptkomponente oder zwischen der Auswahl von zwei Komponenten oder mehr, sollte man mehr Varianz erhalten wollen.
- EFA zeigt direkt über die Indexbestimmung an, wieviele Faktoren es sein sollen.In diesem Fall einen.

Beschreibung Ergebnisse

Die Ergebnisse kommen alle bei den bereits erwähnten Resultaten vor.

Analyse/Hinterfragen Ergebnisse

Die Analyse der Ergebnisse kommt bei den bereits erwähnten Resultaten vor.

Quellenangaben

- Daten von R (24.12.2024)
- Andreas Handl, Torben Kuhlenkasper. (2017) Multivariate Analysemethoden (3. Auflage). Springer Spektrum.

Clusteranalyse

Einleitung hierarchische Clusteranalyse

Vorgehensweise hierarchische Clusteranalyse (bei manchen Zeilen ist head verwendet worden, um das Maximum von 100 Seiten eher erreichen zu können):

- Daten einlesen
- Daten normalisieren
- Distanzen berechnen, euclidean
- Boxplots für Anzeige von Outliers
- Berechnungen vornehmen mit single-linkage, complete-linkage, average-linkage, centroid-linkage, ward-linkage
- Anzeige der Dendrogramme
- Interpretation, Versuch einer Erklärung

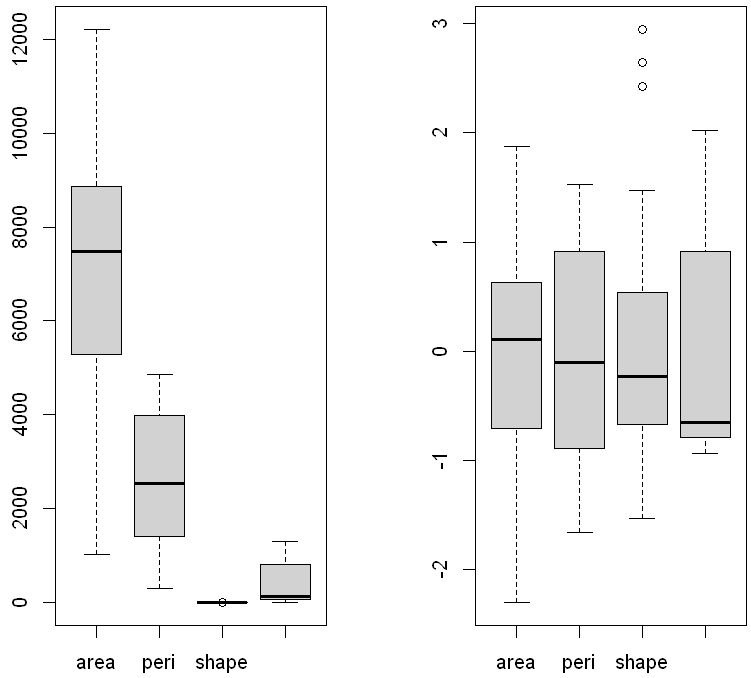
Statistisches Vorgehen hierarchische Clutseranalyse

In [10]:
```r
# Normalize installieren und aufrufen
install.packages("normalize")
library("normalize")

# Daten einlesen
# Verwendung des Datensatzes rock
rock_data <- data.frame(rock)
head(rock_data)
# Normalisieren (Methode standardize)
data_n <- normalize(rock_data, method="standardize")

par(mfrow = c(1, 2))
# Boxplots für die Erkennung von Outliers
boxplot(rock_data)
boxplot(data_n)
```

```
Warning message:
"package 'normalize' is in use and will not be installed"
```

A data.frame: 6 × 4

	area	peri	shape	perm
	<int>	<dbl>	<dbl>	<dbl>
1	4990	2791.90	0.0903296	6.3
2	7002	3892.60	0.1486220	6.3
3	7558	3930.66	0.1833120	6.3
4	7352	3869.32	0.1170630	6.3
5	7943	3948.54	0.1224170	17.1
6	7979	4010.15	0.1670450	17.1

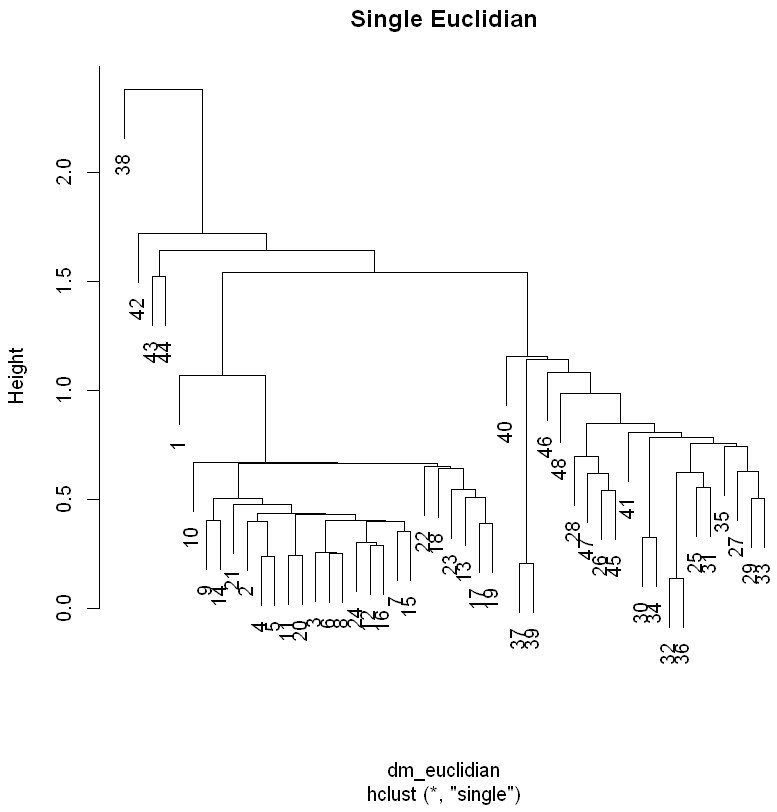


Hierbei handelt es sich um die Distanzmessungen, die für das hclust verwendet werden.

In [11]:
```
# Distanzen berechnen
dm_euclidian <- dist(data_n, method = "euclidean")
```

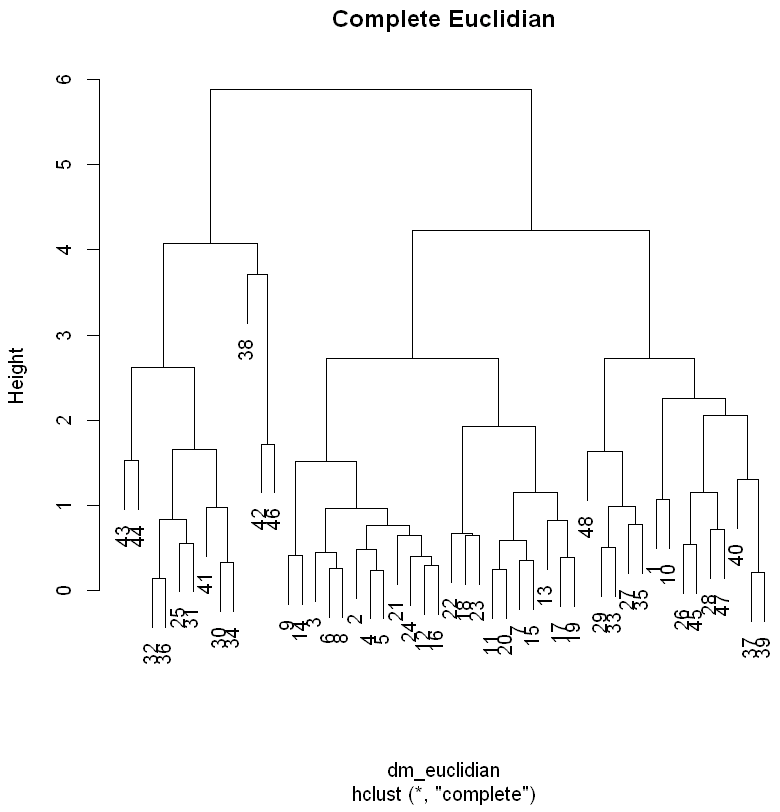
Hierbei handelt es sich um hclust single-linkage

In [12]:
```
# hclust Berechnung, single-linkage
hclust_single1 <- hclust(dm_euclidian, method="single")
plot(hclust_single1, main = "Single Euclidian")
```



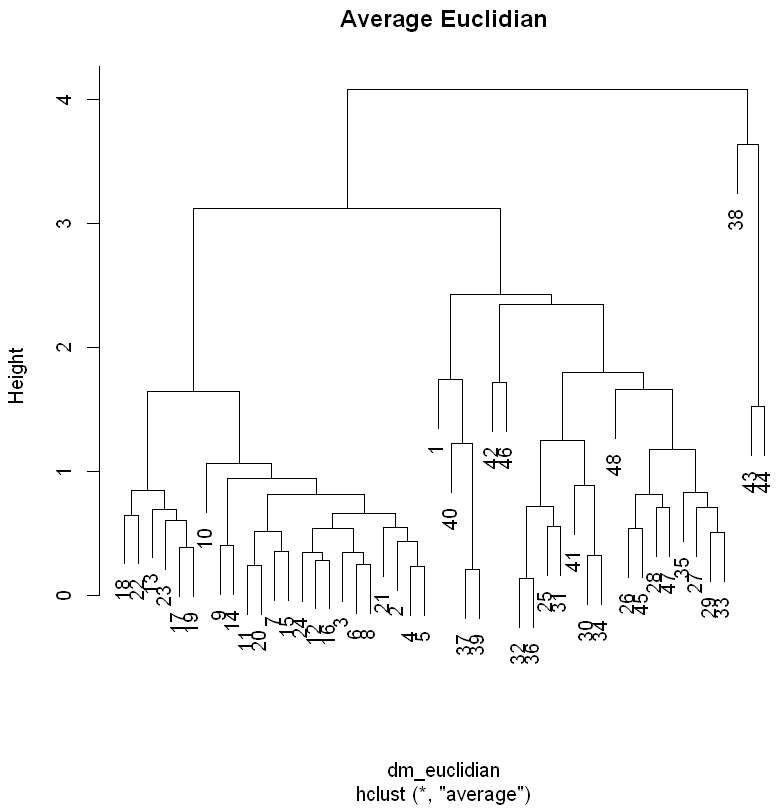
Hierbei handelt es sich um hclust complete-linkage

In [13]:
```
# hclust Berechnung, complete-linkage
hclust_single1 <- hclust(dm_euclidian, method="complete")
plot(hclust_single1, main = "Complete Euclidian")
```



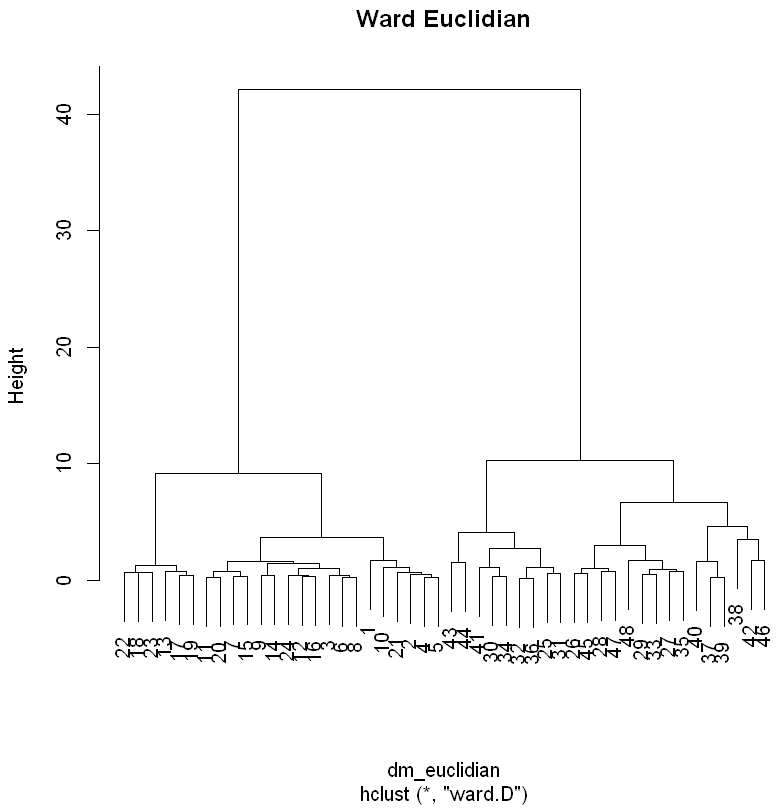
Hierbei handelt es sich um hclust average-linkage

In [14]:
```
# hclust Berechnung, average-linkage
hclust_single1 <- hclust(dm_euclidian, method="average")
plot(hclust_single1, main = "Average Euclidian")
```



Hierbei handelt es sich um hclust ward-linkage

In [15]:
```
# hclust Berechnung, ward-linkage
hclust_single1 <- hclust(dm_euclidian, method="ward.D")
plot(hclust_single1, main = "Ward Euclidian")
```



Resultat Vorgehensweise hierarchische Clutseranalyse

- Die Outliers sind sehr gering in der Anzahl, sowohl im nicht normalisierten Datensatz, wie auch nach dem Normalisieren. Dies kann man gut bei shape erkennen. Nach dem Normalisieren gibt es drei Outliers, die weiter vom Median entfernt sind. Insgesamt gesehen, sollten diese wenigen Werte keinen grossen Einfluss auf das Clustering haben. Ansonsten müsste man sehen, ob man jene entfernen kann und/oder eine Methode auswählen, die mit Outliers umgehen kann.

- Je nach gewählter Distanz und Methode fallen die Resultate anders aus. Single kann Outliers gut entdecken, complete formt kompaktere Cluster, average ist ähnlich zu complete und Ouliers werden besser berücksichtigt, centroid funktioniert bei Daten mit wenig Gleichheit, ward formt ebenfalls kompaktere Cluster. (Siehe Tabelle in https://rpubs.com/pjmurphy/599072)

- Auffallend ist bei der Verwendung von ward, dass die Dendrogramme sich sehr ähnlich sind, im Vergleich zu den anderen Methoden.

- Welches Clustering nun besser ist oder sich weniger eignet, in Zusammenhang mit der gewählten Mthode kann untersucht werden.

Einleitung k-means Clusteranalyse

Vorgehensweise k-means Clusteranalyse:

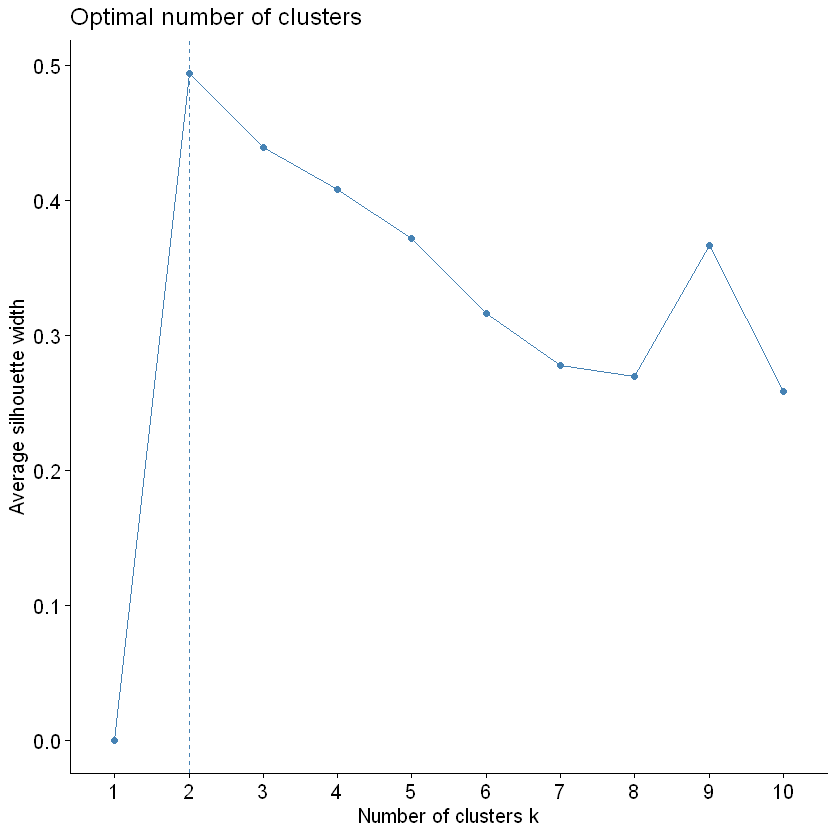
- Mit Screeplot und Silhouette die geeignete Clusteranzahl bestimmen
- R kmeans verwenden
- Cluster Plot für k-Means und k-Medoid
- Silhouette Plot für k-Means und k-Medoid

Statistisches Vorgehen k-means Clusteranalyse

In [17]:
```
suppressMessages(suppressWarnings(install.packages("tidyverse")))
suppressPackageStartupMessages(library(tidyverse))
suppressMessages(suppressWarnings(install.packages("factoextra")))
suppressPackageStartupMessages(library(factoextra))
suppressMessages(suppressWarnings(install.packages("fpc")))
suppressPackageStartupMessages(library(fpc))
suppressPackageStartupMessages(library(ggplot2))
suppressMessages(suppressWarnings(install.packages("cluster")))
suppressPackageStartupMessages(library(cluster))
```

Der Silhouette Plot zeigt an, dass zwei Cluster ideal sind.

In [18]:
```
# Clusteranzahl bestimmen, Silhouette
fviz_nbclust(data_n, kmeans, method='silhouette')
```



Anbei die Elemente pro Cluster und die Bestimmung der Centroids.

In [19]:
```
# kmeans Berechnung
means <- kmeans(data_n, 2)
means$size
means$centers
```

24 · 24

A matrix: 2 × 4 of type dbl

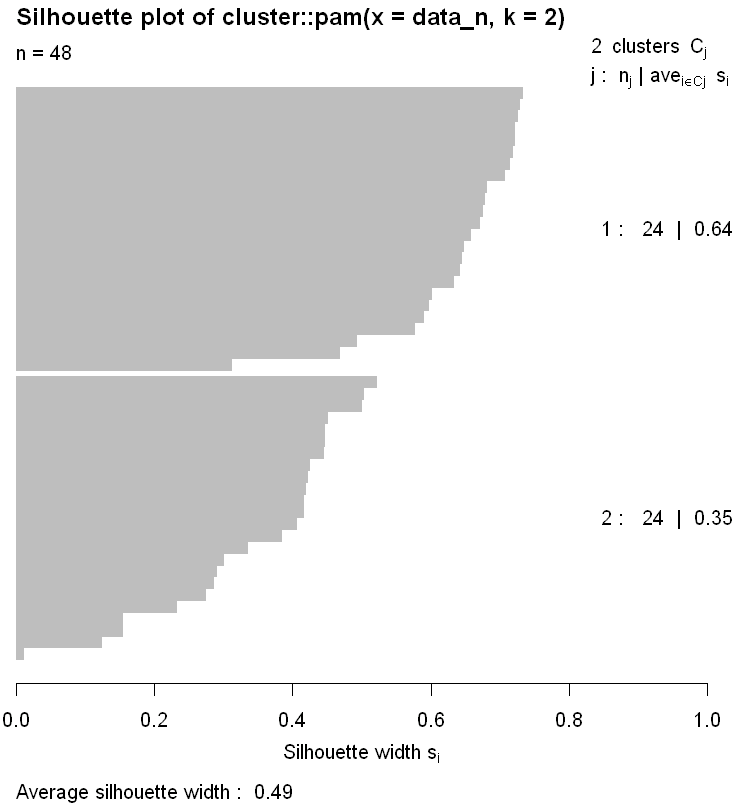
	area	peri	shape	perm
1	0.653376	0.9292371	-0.4900107	-0.7869704
2	-0.653376	-0.9292371	0.4900107	0.7869704

Ein Silhouette Plot mit pam.

In [20]:
```
# Silhouette Scores und Plot (M)
install.packages("cluster")
score <- cluster::pam(data_n, k = 2)
score
plot(cluster::silhouette(score))
```

```
Warning message:
"package 'cluster' is in use and will not be installed"
```

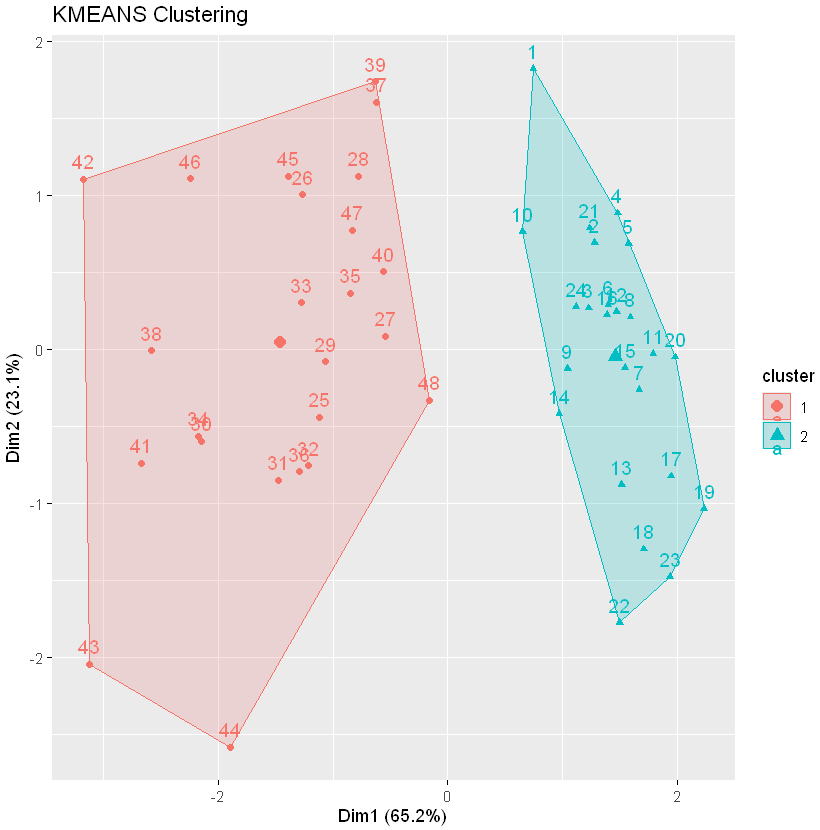
```
Medoids:
     ID       area       peri       shape       perm
[1,] 15  0.8306246  0.9200208 -0.54545349 -0.7607038
[2,] 29 -0.2528940 -0.5804460  0.08507632  1.0838973
Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2
Objective function:
   build     swap
1.236789 1.200561

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```



Der k-Means Cluster Plot.

In [21]:
```
# Cluster Plot für k-Means
km <- eclust(data_n, "kmeans", hc_metric="euclidean",k=2)
km
```

```
K-means clustering with 2 clusters of sizes 24, 24

Cluster means:
       area       peri      shape       perm
1 -0.653376 -0.9292371  0.4900107  0.7869704
2  0.653376  0.9292371 -0.4900107 -0.7869704

Clustering vector:
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 63.19736 21.61153
 (between_SS / total_SS =  54.9 %)

Available components:

 [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
 [6] "betweenss"    "size"         "iter"         "ifault"       "clust_plot"
[11] "silinfo"      "nbclust"      "data"
```



Der k-Medoid Cluster Plot.

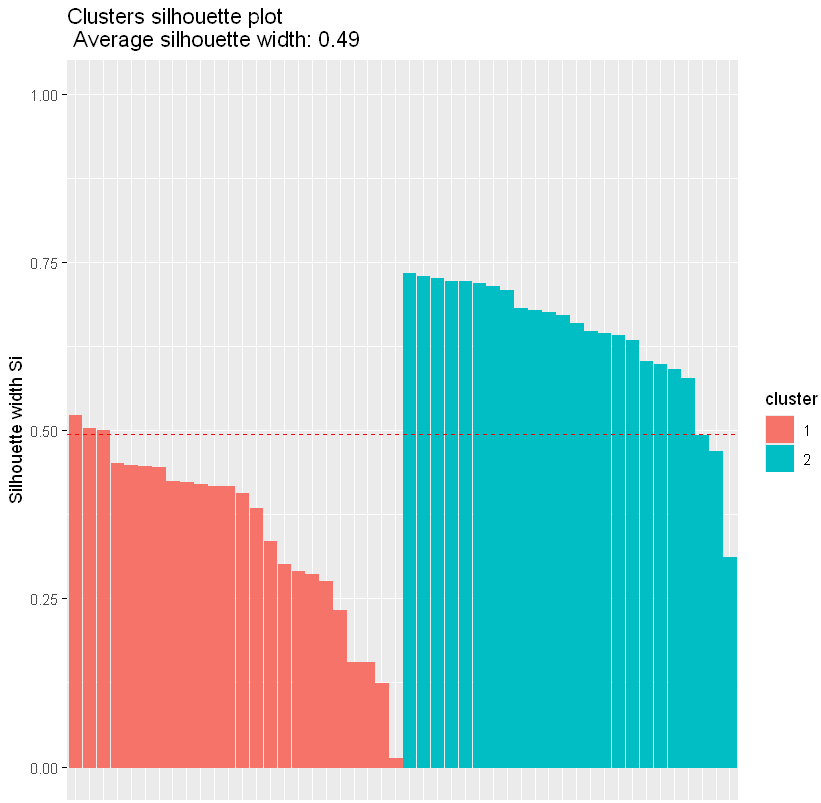
In [22]:
```
# Cluster Plot für k-Medoid
km2 <- eclust(data_n, "pam", hc_metric="euclidean",k=2)
km2
```

```
Medoids:
     ID       area       peri       shape       perm
[1,] 15  0.8306246  0.9200208 -0.54545349 -0.7607038
[2,] 29 -0.2528940 -0.5804460  0.08507632  1.0838973
Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2
Objective function:
   build     swap
1.236789 1.200561

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
[11] "clust_plot" "nbclust"
```

Der Cluster Silhouette Plot

In [23]:
```
# Cluster Silhouette Plot
a1 = fviz_silhouette(km)
a1
```

```
  cluster size ave.sil.width
1       1   24          0.35
2       2   24          0.64
```



Resultat k-means Clusteranalyse

- Die Clusterbestimmung mit Silhouette zeigt an, dass 2 Cluster ideal sind (Optimal number of clusters).
- Means zeigt die Centroid Positionen an, hier zwei Cluster und ebenfalls die Clustergrösse von 24 pro Cluster.
- Der Silhouette Plot zeigt pro Cluster an, wie gut die Datenpunkte dem Cluster zugehörig sind. Bei dem zweiten Cluster sieht man wenige Werte, die nahe der Null sich befinden. Im ersten Cluster sind die meisten Punkte weit von der Null entfernt. Der ave.sil.width von 0.35 zeigt aber eher eine schwaches Resultat und 0.64 eher eine stärkeres Resultat.Der Mittelwert von 0.49 ist eher gering.
- Der Cluster Plot zeigt die beiden Cluster an, man erkennt eine klare Auftrennung. Im Gegenteil zu der Bewertung mit dem ave.sil.width kann man hier sagen, dass gut geclustert wird.
- Der Cluster Silhouette Plot zeigt beim ersten Cluster Schwächen an, dass fast alle Werte unter der 0.5 Marke sind. Bei dem zweiten Cluster sind fast alle Werte über der 0.5 Marke. Es sind keine grossen Fluktuationen vorhanden, die Breiten sind bei allen fast gleich gross.
- Es soll noch angemerkt werden, dass mit pam eher eine k-Medoid Cluseranalyse vorgenommen wird. Daher macht es auch Sinn, dass die beiden Silhouette Plots sich

unterscheiden, denn der eine wird mit k-Medoid berücksichtigt, der andere mit k-Means. Die visuelle Analyse gibt dieselben Resultate. Somit ist klar, dass mit einer visuellen Bestimmung die Zahlen nicht minder wichtig sind, um eine gute Beurteilung vornehmen zu können.

Beschreibung Ergebnisse

Die Ergebnisse kommen alle bei den bereits erwähnten Resultaten vor.

Analyse/Hinterfragen Ergebnisse

Die Analyse der Ergebnisse kommt bei den bereits erwähnten Resultaten vor.

Quellenangaben

- Daten von R (24.12.2024)
- Andreas Handl, Torben Kuhlenkasper. (2017) Multivariate Analysemethoden (3. Auflage). Springer Spektrum.
- Cluster analysis, https://rpubs.com/Tetiana/clustering-in-r (24.12.2024)