Kapitel 4: Merkmalszusammenhänge

Streudiagramme

R bietet die Möglichkeit, verschiedene Arten von Streudiagrammen zu zeichnen. Hierfür werden wir das Paket `ggplot2` verwenden.

```
library(ggplot2)
```

Lesen Sie den Beispieldatensatz ein.

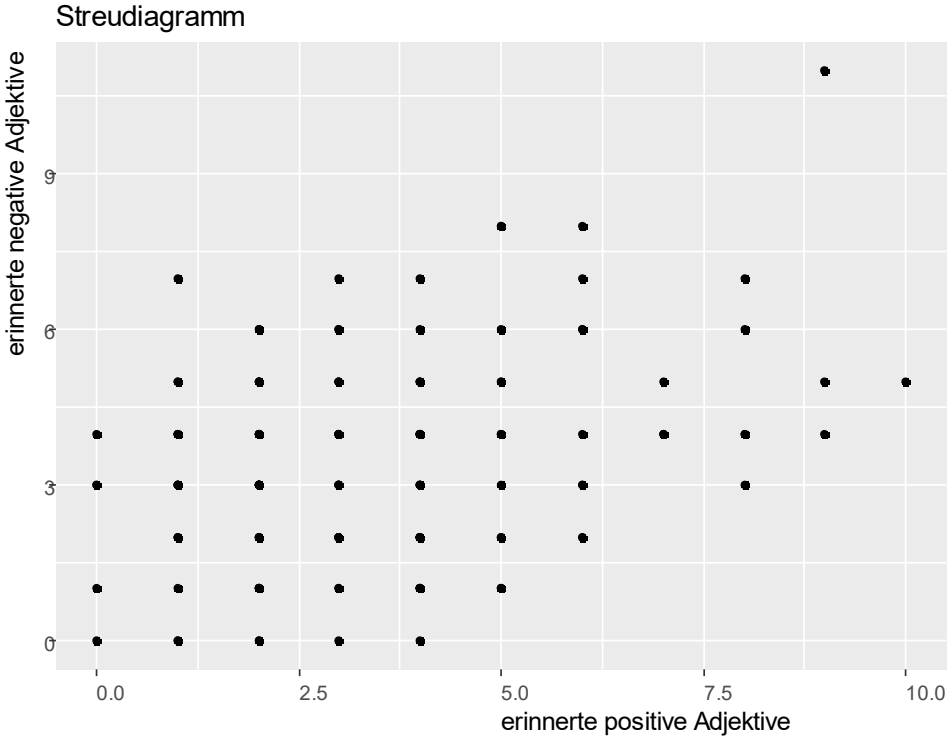
```
library(foreign)
beispiel <- read.spss("Beispieldatensatz.sav",
                      to.data.frame = TRUE)
```

Das Streudiagramm wird mithilfe der Funktion `geom_point()` des Pakets `ggplot2` erstellt.

Eine Frage in Bezug auf das Gedächtnisexperiment könnte lauten, wie die Gedächtnisleistung von positiven und negativen Adjektiven zusammenhängt. Um diese Frage graphisch zu beantworten, tragen Sie die beiden betreffenden Variablen für die x- bzw. y-Achse ein. Welche der beiden Sie für die x- und welche für die y-Achse angeben, spielt in diesem Fall keine Rolle. Es kann aber Fragestellungen geben, in denen Ihnen die Interpretation leichter fällt für eine bestimmte Anordnung der beiden Variablen. Sie erstellen das Diagramm folgendermaßen:

```
streu <- ggplot(beispiel, aes(x = positiv, y = negativ)) +
  geom_point() +
  labs(title = "Streudiagramm",
       x = "erinnerte positive Adjektive",
       y = "erinnerte negative Adjektive")
```

Sie erhalten diesen Output, wenn Sie `streu` in der Konsole eingeben:



Aus der Abbildung ist zu entnehmen, dass Personen, die viele positive Adjektive erinnern tendenziell auch viele negative Adjektive erinnern. Dies spricht für einen positiven Zusammenhang.

Wie bei allen Grafiken in R haben Sie die Möglichkeit, die Abbildung mit Hilfe weiterer Argumente zu verändern und zu gestalten, z.B. das Skalenformat, die Beschriftung etc.

Jeder der Kreise in der Abbildung kann mehrere Fälle, also Versuchspersonen beschreiben.

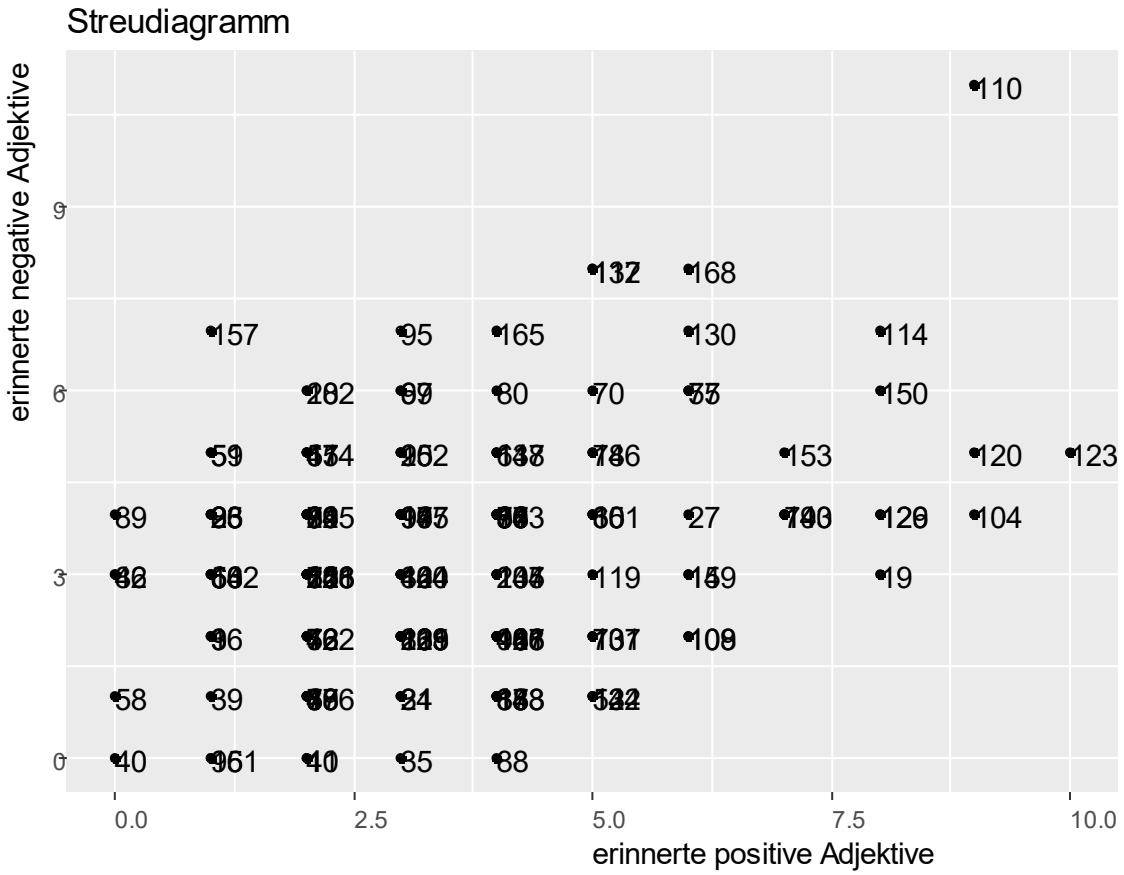
Um herauszufinden, um welche Fälle es sich dabei handelt, benötigen Sie die Funktion `geom_text()`. Außerdem müssen Sie angeben, von welcher Variable die Beschriftungen bezogen werden sollen. In unserem Fall wäre es die Variable „vpnr“, welches als Attribut des Arguments `label` hinzugefügt werden muss, sodass sich folgender Code ergibt:

```
streu.vpnr <- ggplot(beispiel,
                     aes(x = positiv, y = negativ,
                         label = beispiel$vpnr)) +
  geom_point() +
  geom_text() +
  labs(title = "Streudiagramm",
       x = "erinnerte positive Adjektive",
       y = "erinnerte negative Adjektive")
```

Wenn Sie nun `streu.vpnr` eingeben, ist jeder Kreis mit den Nummern der Personen gekennzeichnet, die von diesem Kreis repräsentiert werden.

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.



Wie Sie sehen, überlappen sich die Beschriftungen und die Punkte deutlich, sodass man teilweise nicht zuordnen kann, welcher Punkt welche Versuchsperson repräsentiert. Abhilfe verschafft das Paket `ggrepel`. Installieren und laden Sie es.

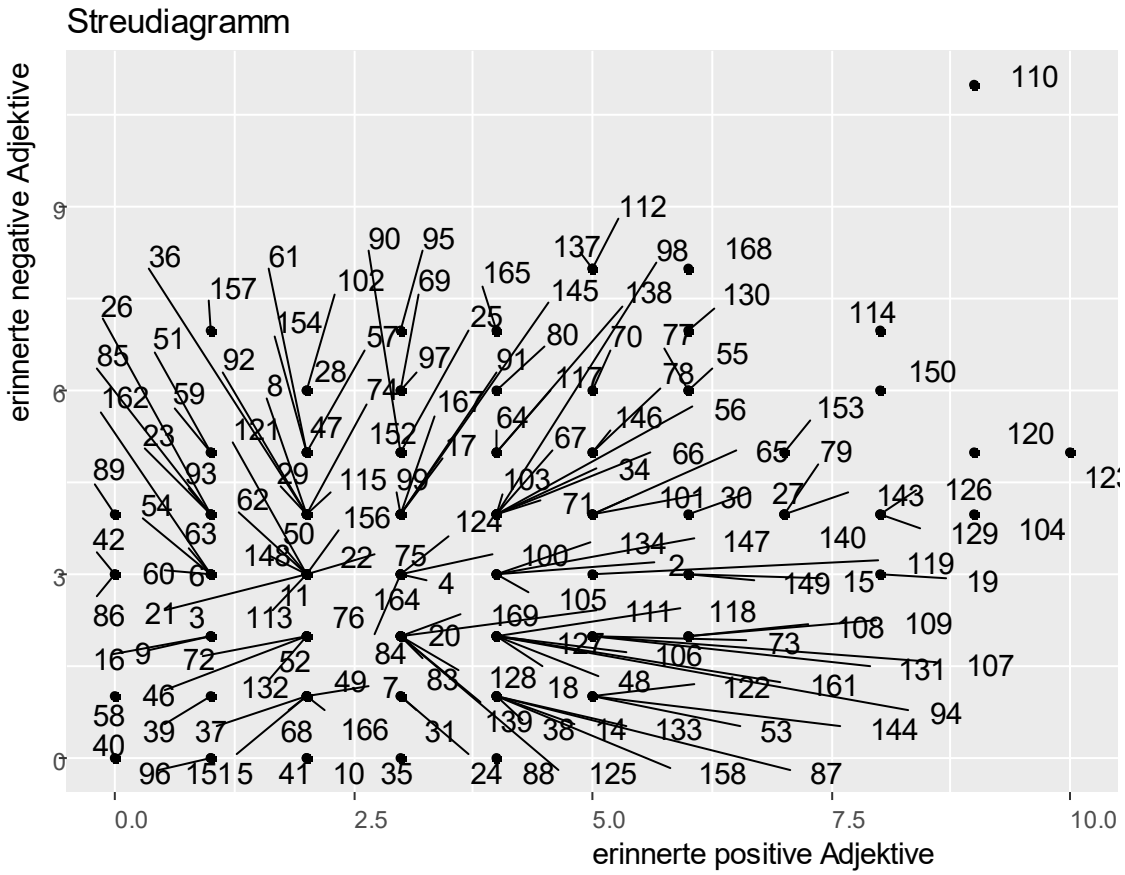
```
install.packages("ggrepel")
library(ggrepel)
```

Als nächstes ersetzen Sie die Funktion `geom_text()` mit der Funktion `geom_text_repel()` und lassen Sie sich erneut das Diagramm anzeigen:

```
streu.vpnr.repel <- ggplot(beispiel,
                           aes(x = positiv, y = negativ,
                               label = beispiel$vpnr)) +
  geom_point() +
  geom_text_repel() +
  labs(title = "Streudiagramm",
       x = "erinnerte positive Adjektive",
       y = "erinnerte negative Adjektive")
```

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.



https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

Korrelationen

Die Korrelation nach Pearson ist ein Maß für den Zusammenhang zweier intervallskalierter Merkmale. Sie lässt sich mithilfe der Funktion `cor()` ermitteln. In die Klammern können Sie die Variablen eintragen, von denen Sie bivariate Korrelationen berechnen möchten. Wenn Sie mehr als zwei Variablen miteinander korrelieren möchten, können Sie diese nicht einfach hintereinander in die Klammern von `cor()` eintragen, sondern Sie müssen mithilfe der Funktion `c()` genau angeben, welche Zeilen und Spalten Sie aus Ihrem Datensatz verwenden möchten. Wir entscheiden uns für die drei Variablen erinnerter positiver, neutraler und negativer Adjektive. Mit dem Attribut `pairwise.complete.obs` geben wir wieder an, dass nur die Fälle berücksichtigt werden sollen, die keine fehlenden Werte bei den interessierenden Variablen haben.

```
cor(beispiel[, c("negativ", "neutral", "positiv")],
    use = "pairwise.complete.obs")
```

Mithilfe der eckigen Klammern können Sie in R angeben, mit welchem Teil des Datensatzes gerechnet werden soll. Dabei werden zunächst die Zeilen angegeben und nach dem Komma die Spalten. In diesem Fall haben wir die Zeilenangabe freigelassen, was bedeutet, dass alle Zeilen berücksichtigt werden sollen. Bei den Spalten wurden die drei Variablen entsprechend eingetragen.

Weiterhin können wir mithilfe des Arguments `method` auswählen, welche Korrelationskoeffizienten wir berechnen möchten. Da es sich bei den interessierenden um intervallskalierte Variablen handelt, entscheiden wir uns für den Korrelationskoeffizienten nach Pearson. Dieser ist bereits die Standardeinstellung, weshalb keine weiteren Angaben notwendig sind. Lägen lediglich rangskalierte Daten vor, wäre der Koeffizient nach Spearman die richtige Wahl. Hierfür würde man den obigen Code um `method = "spearman"` ergänzen. Für eine punktbiseriale Korrelation zwischen einem dichotomen und einem intervallskalierten Merkmal (bspw. Geschlecht und Gesamtzahl erinnerter Adjektive) ist wieder der Koeffizient nach Pearson korrekt (siehe Kapitel 4.1).

Der obige Code liefert diesen Output:

```
          negativ   neutral   positiv
negativ 1.0000000 0.2867774 0.3369060
neutral 0.2867774 1.0000000 0.2948303
positiv 0.3369060 0.2948303 1.0000000
```

Wie lässt sich diese Tabelle interpretieren? Sie sehen, dass die Korrelation einer Variable mit sich selber 1 beträgt. Die drei Werte über und unter dieser Diagonalen entsprechen sich, denn die Korrelation einer Variablen A mit B ist identisch mit der Korrelation von B mit A. Letztlich liefert Ihnen die Tabelle also drei interessante Korrelationen.

Mit der Funktion `cor.test()` können Sie die Signifikanztests für die Korrelationen durchführen. Allerdings können Sie mit dieser Funktion immer nur zwei Variablen einzeln testen. Möchten Sie zu der obigen Tabelle eine Tabelle mit den Signifikanzniveaus haben, können Sie die Funktion `rcorr()` des Pakets `Hmisc` verwenden. Installieren und laden Sie es:

```
install.packages("Hmisc")
```


Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

```
library(Hmisc)
```

Der folgende Code führt die Signifikanztests für die Korrelationen durch:

```
rcorr(as.matrix(beispiel[, c("negativ", "neutral", "positiv")]))
```

Wie Sie sehen, sieht der Code bis auf eine Ausnahme genauso aus wie der obige. Die Funktion `rcorr()` kann nicht bei Objekten im `data.frame` Format angewendet werden, weshalb wir mit der Funktion `as.matrix()` den Datensatz in eine Matrix umwandeln.

Sie erhalten folgenden Output:

```
        negativ neutral positiv
negativ    1.00    0.29    0.34
neutral    0.29    1.00    0.29
positiv    0.34    0.29    1.00

n= 150



P
        negativ neutral positiv
negativ         0.0004  0.0000
neutral 0.0004          0.0002
positiv 0.0000  0.0002
```

Sie sehen, dass alle drei hoch signifikant sind. Es bestehen also zwischen diesen drei Variablen jeweils Zusammenhänge, die nach aller Wahrscheinlichkeit nicht zufällig, sondern systematisch sind.

Ein interessanter Aspekt bezieht sich auf den *p*-Wert, also die von R angezeigte Signifikanz der Korrelationen. Diese sind unter anderem mit „.0000“ angegeben. Dies bedeutet allerdings nicht, dass die Wahrscheinlichkeit für die empirischen Ergebnisse tatsächlich gleich Null ist. Sie ist lediglich kleiner als .0001. Deshalb gibt R diese Werte an. Wenn Sie die Signifikanztests mit der Funktion `cor.test()` durchführen, sehen Sie, die exakte, sehr geringe Wahrscheinlichkeit für das Auftreten der entsprechenden Korrelation unter Annahme der Nullhypothese (also der Annahme, dass kein Zusammenhang zwischen den Variablen besteht). In einem Ergebnisbericht einer empirischen Studie würden Sie als *p*-Wert „< .001“ angeben.

Partialkorrelation

Anmerkung: Für diesen Abschnitt lesen Sie bitte die Datei „Partialkorrelation.sav“ ein.

```
library(foreign)
partial <- read.spss("Partialkorrelation.sav",
                     to.data.frame = TRUE)
```

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

In Kapitel 4.1 haben Sie das Konzept der Partialkorrelation kennen gelernt. An dieser Stelle werden wir das dort besprochene Beispiel nachvollziehen. Dort ging es darum, dass ein hoher Zusammenhang zwischen der Anzahl eingesetzter Feuerwehrleute und der Schadenshöhe besteht. Dieser Zusammenhang ließ vermuten, dass es eine dritte Variable geben könnte, die beide Variablen beeinflusst, z.B. die Schwere des Brandes. Wenn diese Vermutung zutrifft, sollte der Zusammenhang zwischen der Anzahl der Feuerwehrleute und der Schadenshöhe stark abnehmen, wenn man für die Schwere des Brandes kontrolliert.

Berechnen wir zunächst mithilfe der Funktion `rcorr()` des Pakets `Hmisc` die Korrelation zwischen der Anzahl der Feuerwehrleute und der Schadenshöhe, damit zusätzlich das Signifikanzniveau angezeigt wird.

```
library(Hmisc)
rcorr(as.matrix(partial[c("fleute", "schaden")]))
```

Man hätte ebenso `rcorr(partial$fleute, partial$schaden)` eingeben können. Allerdings werden dann in der Korrelationstabelle nicht die Variablennamen angezeigt, sondern „x“ und „y“.

Wir erhalten folgenden Output mit der aus den Ausführungen in Kapitel 4.1 erwarteten Korrelation:

```
        fleute schaden
fleute    1.00    0.63
schaden   0.63    1.00

n= 10



P
        fleute schaden
fleute          0.05
schaden 0.05
```

Hinweis: Das Paket `ggm`, welches Sie im Anschluss installieren werden, verwendet ebenfalls die Funktion `rcorr()`. Wenn Sie dieses Paket nach dem Paket `Hmisc` laden, dann wird diese Funktion überschrieben und Sie werden die Funktion `rcorr()` des Pakets `ggm` verwenden, wodurch Sie einen anderen Output erhalten. Wenn Sie explizit angeben wollen, dass Sie auf die Funktion des Pakets `Hmisc` zugreifen möchten, geben Sie folgendes ein:

```
Hmisc::rcorr(as.matrix(partial[c("fleute", "schaden")]))
```

Im zweiten Schritt berechnen wir die Partialkorrelation und kontrollieren damit für den Einfluss der Schwere des Brands auf die beiden Variablen. Dazu verwenden wir die Funktionen `pcor()` und `pcor.test()` des Pakets `ggm`.

```
install.packages("ggm")
library(ggm)
```

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

Die Partialkorrelation berechnen wir mit der Funktion `pcor()`. Zuerst tragen wir die beiden Variablen ein, die wir miteinander korrelieren möchten, in unserem Fall „Anzahl der Feuerwehrleute“ und „Schadenshöhe“, und als nächstes geben wir Kontrollvariable an. Diese drei Variablen müssen mit der Funktion `c()` zusammengefasst werden. Mit der Funktion `var()` wird der Datensatz angegeben.

```
pcor(c("fleute", "schaden", "brand"), var(partial))
```

Sie erhalten den folgenden Output:

```
[1] 0.1297005
```

Nun berechnen wir die Signifikanz mit der Funktion `pcor.test()`. Hierfür geben wir zunächst die Partialkorrelation an, auf die sich der Test beziehen soll. Man kann die vorige Berechnung in ein Objekt speichern und das angeben oder man kann auch den gesamten Befehl in die Klammern eintragen. Danach gibt man die Anzahl der Kontrollvariablen und die Stichprobengröße an. Die Stichprobengröße können Sie erfahren, indem Sie sich den Datensatz mit `view(partial)` anschauen oder mit `nrow(partial)` sich die Anzahl der Zeilen ausgeben lassen. Die Stichprobengröße beträgt 10.

```
pcor.test(pcor(c("fleute", "schaden", "brand"), var(partial)), 1, 10)
```

```
$tval
[1] 0.3460784

$df
[1] 7

$pvalue
[1] 0.7394524
```

Offenbar hatte die Kontrollvariable Schwere des Brandes einen großen Einfluss auf beide Variablen. Wenn wir für diesen Einfluss kontrollieren, verringert sich die Korrelation beträchtlich. Die Freiheitsgrade verringern sich auf $df = N - 3$ (vgl. Kapitel 4.1).

Lineare Regression

Anmerkung: Bitte lesen Sie für diesen Abschnitt die Datei „Regression.sav“ ein.

```
library(foreign)
regression <- read.spss("Regression.sav",
                        to.data.frame = TRUE)
```

Die in Kapitel 4.2 dargestellten Zusammenhänge zur linearen Regression bilden nur die Spitze des großen „Eisbergs Regression“. Daher bietet R vielfältige Möglichkeiten. Wir werden uns auf den einfachsten Fall, die Vorhersage einer Variablen durch eine andere, beschränken. Sie verwenden hierfür die Funktion `lm()`.

Lassen Sie uns versuchen, das Beispiel aus Kapitel 4.2 an dieser Stelle mit R nachzuvollziehen. Dort ging es um den Einfluss der Alkoholkonzentration auf die Reaktionszeit. Unsere abhängige Variable (Kriterium) ist also die Reaktionszeit. Als Prädiktor dient die Alkoholkonzentration. Sie geben zunächst das Kriterium an, danach eine Tilde und als letztes die Prädiktoren. Das speichern Sie in einem neuen Objekt und lassen sich die Regressionsanalyse mit `summary()` ausgeben.

```
regression.1 <- lm(regression$Reak ~ regression$Alko)
summary(regression.1)
```

Sie erhalten folgenden Output:

```
Call:
lm(formula = regression$Reak ~ regression$Alko)

Residuals:
   Min     1Q Median     3Q    Max
-69.42 -18.35 -11.31  21.63  63.58

Coefficients:
                Estimate Std. Error t value      Pr(>|t|)
(Intercept)       596.50      24.35  24.496 0.00000000824 ***
regression$Alko    53.84      17.21   3.128        0.0141 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43.28 on 8 degrees of freedom
Multiple R-squared:  0.5502,    Adjusted R-squared:  0.4939
F-statistic: 9.784 on 1 and 8 DF,  p-value: 0.01406
```

Alles hat so funktioniert, wie wir es beabsichtigt hatten: Die Reaktionszeit bildet die abhängige Variable und die Alkoholkonzentration den Prädiktor. Im unteren Teil sehen Sie den Determinationskoeffizienten r^2 (in R als „Multiple R-squared“ dargestellt). In unserem Fall einer

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

Regression mit einem Prädiktor entspricht der Koeffizient *r* dem bivariaten Korrelationskoeffizienten. Sie können dies überprüfen, indem Sie die Wurzel aus dem Determinationskoeffizienten mit `sqrt()` ziehen und das Ergebnis mit der Korrelation mit `cor()` vergleichen.

```
sqrt(0.5502)
[1] 0.7417547
```

```
cor(regression$Reak, regression$Alko)
[1] 0.7417239
```

Die Differenzen basieren auf Rundungsungenauigkeiten.

In einer linearen Regression ist es so, dass die Stichprobenwerte den wahren Zusammenhang in der Population überschätzen. Das Modell ist also besser an diese speziellen Daten angepasst als es für die Population richtig wäre. Deshalb bietet R ein korrigiertes r^2 an. In unserem Fall ist der Unterschied zwischen r^2 und dem korrigierten r^2 relativ gering. Der Standardschätzfehler steht über dem Determinationskoeffizienten (in R als „Residual standard error" dargestellt) (siehe Kapitel 4.2).

Die letzte Zeile des Outputs stellt Informationen über eine Varianzanalyse (ANOVA) dar. Dieses Verfahren lernen Sie in den Kapiteln 5 und 6 kennen.

Die Tabelle „Coefficients" liefert Ihnen konkrete Informationen über den von uns gewählten Prädiktor. Sie finden Werte für unstandardisierte Koeffizienten (Kapitel 4.2). Die standardisierten Koeffizienten können Sie manuell berechnen oder Sie können die Funktion `lm.beta()` des Pakets `QuantPsyc` verwenden.

```
# Manuelle Berechnung
53.84*sd(regression$Alko)/sd(regression$Reak)
[1] 0.7416643
```

```
install.packages("QuantPsyc")
library(QuantPsyc)
lm.beta(regression.1)
regression$Alko
      0.7417239
```

Die Differenzen basieren wieder auf Rundungsungenauigkeiten. Standardisierte Koeffizienten sind vor allem dann nützlich, wenn Sie mehr als einen Prädiktor in ein Modell mit aufnehmen. Dann erlauben Ihnen diese standardisierten Werte einen direkten Vergleich der Prädiktoren. Die unstandardisierten Werte unterliegen dagegen der Metrik jedes einzelnen Prädiktors. Diese Metriken können sich durchaus zwischen den Prädiktoren unterscheiden. Dieser Fall einer Regression mit mehr als einem Prädiktor heißt Multiple Regression. Sie ist eng verwandt mit der

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

linearen Regression mit einem Prädiktor, erfährt aber in diesem Buch keine ausführliche Erörterung (für einen Ausblick, siehe Kapitel 4).

Den einzelnen Koeffizienten ist ein *t*-Wert zugeordnet, in diesem Fall für die Konstante und für den einen Prädiktor. Der Wert für die Konstante hat für uns wenig theoretischen und praktischen Belang. Er kennzeichnet die Höhenlage der Regressionsgeraden. Der *t*-Wert für den Prädiktor wird hoch signifikant. Die Interpretation lautet, dass die Alkoholkonzentration ein signifikanter Prädiktor für die Reaktionszeit ist. Wäre dieser *t*-Wert sehr klein und folglich nicht signifikant gewesen, wäre die Vorhersageleistung deutlich schlechter gewesen und eine Vorhersage der Reaktionszeit mit diesem Prädiktor hätte sich als nicht lohnenswert erwiesen. (Kleine Abweichungen zwischen den hier präsentierten Werten und denen im Buch sind auf Rundungsfehler zurückzuführen.)

Hätten wir sowohl die Prädiktorvariable (Alkoholkonzentration) als auch das Kriterium (Reaktionszeit) vor der Analyse mit der Funktion `scale()` z-standardisiert (siehe R-Erläuterungen zu Kapitel 1), stünde der standardisierte Regressionskoeffizient sowohl in der Spalte „Nicht standardisierte Koeffizienten“ als auch in der Spalte „standardisierte Koeffizienten“. Dies liegt daran, dass die Verteilungen beider Variablen in diesem Fall den Mittelwert Null und eine Streuung von Eins haben (siehe Kapitel 1).

```
regression.stand <- lm(scale(regression$Reak) ~ scale(regression$Alko))
summary(regression.stand)
```

```
Call:
lm(formula = scale(regression$Reak) ~ scale(regression$Alko))

Residuals:
    Min      1Q  Median      3Q     Max
-1.1411 -0.3016 -0.1859  0.3556  1.0450

Coefficients:
                                  Estimate            Std. Error t value Pr(>|t|)
(Intercept)            0.0000000000000001042 0.2249614038973900276   0.000   1.0000
scale(regression$Alko) 0.7417239184285653320 0.2371301406482773932   3.128   0.0141 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7114 on 8 degrees of freedom
Multiple R-squared:  0.5502,    Adjusted R-squared:  0.4939
F-statistic: 9.784 on 1 and 8 DF,  p-value: 0.01406
```

Zusammenhang zwischen Korrelation, Regression und *t*-Test

In Kapitel 4.1 haben wir gesehen, dass eine punktbiseriale Korrelation und ein *t*-Test konzeptuell identisch sind. Abbildungen 4.7 und 4.8 veranschaulichen dies. In einem Fall liegt der Fokus auf

dem Mittelwertunterschied zwischen zwei Gruppen (*t*-Test), im anderen Fall darauf, wie der Zusammenhang zwischen einem dichotomen Merkmal (z.B. Zugehörigkeit zu Gruppe A oder Gruppe B) und einem intervallskalierten Merkmal (z.B. Erinnerungsleistung) aussieht.

Kapitel 4.2 hat dann die Zusammenhänge zwischen Korrelation und Regression deutlich gemacht. Dies legt nahe, dass es auch einen Zusammenhang zwischen Regression und *t*-Test geben könnte. Diesen werden wir an Hand von R Outputs nachvollziehen.

Zunächst zur Korrelation: Wir berechnen die punktbiseriale Korrelation zwischen dem Geschlecht und der Erinnerungsleistung in dem Gedächtnisexperiment. Dem Output können wir entnehmen, dass das Geschlecht positiv zu r = 0,15 mit der Erinnerungsleistung korreliert. Da in der Variablenansicht das Geschlecht so kodiert ist, dass Männern eine Eins und Frauen eine Zwei zugeordnet ist, sagt diese punktbiseriale Korrelation also aus, dass Frauen tendenziell mehr Adjektive erinnert haben als Männer. Hohe Werte in der einen Variable gehen tendenziell mit hohen Werten in der anderen Variable einher. Dieser Zusammenhang ist bei zweiseitiger Testung marginal signifikant.

```
library(foreign)
beispiel <- read.spss("Beispieldatensatz.sav",
                      to.data.frame = TRUE)
```

```
library(Hmisc)
rcorr(beispiel$sex, beispiel$ges)
```

```
     x    y
x 1.00 0.15
y 0.15 1.00

n= 150


P
  x      y
x        0.0718
y 0.0718
```

Im zweiten Schritt nutzen wir die Regressionsrechnung, um mit dem Geschlecht die Erinnerungsleistung im Gedächtnisexperiment vorherzusagen. Die damit verbundene Frage lautet: Ist das Geschlecht ein bedeutsamer Prädiktor für die Anzahl erinnerter Adjektive?

```
beispiel.regr <- lm(beispiel$ges ~ beispiel$sex)
summary(beispiel.regr)
```

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

```
Call:
lm(formula = beispiel$ges ~ beispiel$sex)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5408 -3.1923 -0.5408  2.8077 15.4592

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           9.1923     0.6011  15.293   <2e-16 ***
beispiel$sexweiblich  1.3485     0.7436   1.813   0.0718 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.334 on 148 degrees of freedom
Multiple R-squared:  0.02174,   Adjusted R-squared:  0.01513
F-statistic: 3.289 on 1 and 148 DF,  p-value: 0.07179
```

Wir finden die Korrelation von $r = 0{,}15$ wieder, die wir in Schritt 1 berechnet haben, indem wir die Wurzel aus dem Determinationskoeffizienten ziehen.

```
sqrt(0.02174)
[1] 0.1474449
```

Sie entspricht wieder dem standardisierten Regressionsgewicht beta.

```
library(QuantPsyc)
lm.beta(beispiel.regr)
```

```
beispiel$sex
   0.1474334
Warning message:
In var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
na.rm) :
  Calling var(x) on a factor x is deprecated and will become an error.
  Use something like 'all(duplicated(x)[-1L])' to test for a constant
vector.
```

Die Warnung können Sie vernachlässigen. Sie kommt daher, weil die Variable „sex“ als Faktor in die Regression eingegangen ist. Die Warnung erscheint nicht mehr, wenn Sie die Regression mit `as.numeric(beispiel$sex)` durchführen.

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

Wir sehen in der Tabelle „Coefficients“ den *t*-Wert, der darüber Auskunft gibt, ob es sich bei dem Merkmal Geschlecht um einen bedeutsamen Prädiktor für die Erinnerungsleistung handelt. Der *p*-Wert entspricht dem der in Schritt 1 berechneten Korrelation. Dies ist kein Zufall! Wie Sie aus Kapitel 4.1 wissen, lässt sich die punktbiseriale Korrelation in einen *t*-Wert umrechnen. Rechnen Sie dies mit der Formel per Hand nach und Sie werden einen *t*-Wert erhalten, der, abgesehen von Rundungsfehlern, dem im Output vorzufindenden von 1,81 entspricht!

Die Korrelation zeigt uns, dass Frauen im Gedächtnisexperiment tendenziell mehr Wörter erinnern als Männer. Aber wie groß ist dieser Unterschied? Und ist er statistisch bedeutsam? Wir werden dies an Hand eines *t*-Tests für unabhängige Stichproben überprüfen.

Zunächst prüfen wir, ob die Varianzhomogenität angenommen werden darf:

```
library(car)
leveneTest(beispiel$ges, beispiel$sex, center = mean)
```

```
Levene's Test for Homogeneity of Variance (center = mean)
       Df F value Pr(>F)
group   1  0.4901  0.485
      148
```

Der Levene-Test zeigt uns, dass wir die Annahme der Varianzhomogenität beibehalten dürfen, denn er ist bei weitem nicht signifikant.

```
t.unab2.var <- t.test(beispiel$ges ~ beispiel$sex,
                      var.equal = TRUE)
```

```
     Two Sample t-test

data:  beispiel$ges by beispiel$sex
t = -1.8134, df = 148, p-value = 0.07179
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.8180054  0.1209881
sample estimates:
mean in group maennlich  mean in group weiblich
               9.192308                10.540816
```

An den Gruppenstatistiken erkennen wir, dass Frauen im Schnitt etwa 1,3 Worte mehr erinnern als Männer. Dank der großen Stichprobengröße ist der *t*-Test robust gegen die ungleich großen Gruppengrößen. Der *t*-Wert für die Frage, ob der Unterschied in der Erinnerungsleistung zwischen den Geschlechtern signifikant ist, hat ein negatives Vorzeichen. Es hängt von der Reihenfolge ab, welches Geschlecht man als Gruppe 1 und welches als Gruppe 2 laufen lässt. Daher ist das

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.

Vorzeichen inhaltlich unbedeutend, da wir die Richtung des Unterschieds aus den Gruppenstatistiken ersehen können.

Der *t*-Wert hat denselben Betrag wie der *t*-Wert für das Beta-Gewicht in der Regression und auch denselben Signifikanzwert. Zudem entspricht das unstandardisierte Regressionsgewicht in der Regressionsrechnung oben dem Betrag nach genau der Mittelwertdifferenz, die der *t*-Wert bewertet. Die Steigung *b* gibt nämlich wieder, um wie viele Einheiten sich das Kriterium (die Gesamtzahl erinnerter Adjektive) verändert, wenn man von der niedriger kodierten Gruppe (1 = Männer) zur höher kodierten Gruppe (2 = Frauen) „springt". Eine Regression einer intervallskalierten Variable auf eine dichotome Variable ist also konzeptuell identisch mit einem *t*-Test für unabhängige Stichproben! Anders ausgedrückt: Der *t*-Test ist ein Spezialfall der Regression. Die Möglichkeiten des *t*-Tests sind mit dieser Konstellation einer nominalskalierten Gruppenvariable und einer intervallskalierten abhängigen Variable erschöpft. Die Regression kann viel mehr! Neben einem dichotomen Prädiktor wie in diesem Beispiel, sind auch Prädiktoren anderer Skalenniveaus zulässig, wie wir in Kapitel 4.2 und weiter oben bereits für einen intervallskalierten Prädiktor gesehen haben. Wie bereits erwähnt, erlaubt die Regressionsrechnung darüber hinaus auch den Einbezug von mehr als einem Prädiktor. Ein Ausblick in Kapitel 4 führt in die Multiple Regression ein. Eine ausführliche Erörterung ist aus Platzgründen in diesem Buch nicht möglich.

An dieser Stelle schließt sich also der Kreis zwischen Korrelation, Regression und *t*-Test für unabhängige Stichproben. Noch einmal sehr darauf verwiesen, dass die Zusammenhänge mit der Regression nur für den hier behandelten Fall mit einem Prädiktor gelten.

https://lehrbuch-psychologie.springer.com/content/zusatztexte-mit-anleitungen-zu-spss-r-und-gpower-sowie-datensätze

Aus: Rasch, Friese, Hofmann & Naumann (2021). Quantitative Methoden. Band 1, 5. Auflage. Heidelberg: Springer.