

# Übungszettel MANOVA

M.Psy.205, Dozent: Dr. Peter Zezula

Johannes Brachem ([johannes.brachem@stud.uni-goettingen.de](mailto:johannes.brachem@stud.uni-goettingen.de))

## Deutsche Version

### Links

[Übungszettel als PDF-Datei zum Drucken](#)

### Hinweise zur Bearbeitung

1. Bitte beantworten Sie die Fragen in einer .Rmd Datei. Sie können Sie über `Datei > Neue Datei > R Markdown...` eine neue R Markdown Datei erstellen. Den Text unter dem *Setup Chunk* (ab Zeile 11) können Sie löschen. [Unter diesem Link](#) können Sie auch unsere Vorlage-Datei herunterladen (Rechtsklick > Speichern unter...).
2. Informationen, die Sie für die Bearbeitung benötigen, finden Sie auf der [Website der Veranstaltung](#)
3. Zögern Sie nicht, im Internet nach Lösungen zu suchen. Das effektive Suchen nach Lösungen für R-Probleme im Internet ist tatsächlich eine sehr nützliche Fähigkeit, auch Profis arbeiten auf diese Weise. Die beste Anlaufstelle dafür ist der [R-Bereich der Programmiererplattform Stackoverflow](#)
4. Auf der Website von R Studio finden Sie sehr [hilfreiche Übersichtszettel](#) zu vielen verschiedenen R-bezogenen Themen. Ein guter Anfang ist der [Base R Cheat Sheet](#)

### Ressourcen

Da es sich um eine praktische Übung handelt, können wir Ihnen nicht alle neuen Befehle einzeln vorstellen. Stattdessen finden Sie hier Verweise auf sinnvolle Ressourcen, in denen Sie für die Bearbeitung unserer Aufgaben nachschlagen können.

Ressource	Beschreibung
Field, Kapitel 16	Buchkapitel, das Schritt für Schritt erklärt, worum es geht, und wie man Multivariate Varianzanalysen in R durchführt. <b>Große Empfehlung!</b>

### Tipp der Woche

Hinweis: Der Tipp ist diese Woche etwas länger.

#### 1) Befehle nutzen, ohne Pakete zu laden

Normalerweise müssen Sie ein Paket mit `library()` laden, damit Sie Funktionen aus diesem Paket nutzen können. Wenn Sie eine Funktion aber nur sehr selten benötigen, können Sie einen Trick verwenden, nämlich den doppelten Doppelpunkt `::`. Das funktioniert nach dem Schema `package::function()` und wird von uns auch benutzt, um Funktionen mit häufigen Namen richtig einzusetzen.

Wenn Sie diese Schreibweise verwenden, muss das jeweilige Paket nicht geladen sein. Beispiel: `psych::describe()`. Das Paket muss allerdings installiert sein, damit diese Schreibweise funktioniert.

## 2) Wie genau funktioniert eigentlich die Pipe? %>%

Die Pipe kommt aus dem Paket `magrittr` und wird im `tidyverse` häufig verwendet. Sie kennen die Pipe vor allem aus `dplyr`, aber wenn Sie `magrittr` oder `dplyr` geladen haben (d.h. mit `library()` aktiviert), dann können Sie die Pipe für fast alle Befehle in R benutzen, wenn Sie möchten. Das funktioniert so:

Der Code, der auf der *linken* Seite der Pipe steht, wird von R "unter der Haube" auf der *rechten* Seite der Pipe eingesetzt, und zwar standardmäßig immer als erstes *Argument* einer *Funktion*, so dass andere Argumente der Funktion mit einem Komma dahinter gestellt werden. Wenn Sie den Code auf der linken Seite der Pipe an eine andere Stelle auf der rechten Seite der Funktion einsetzen möchten, dann können Sie R mit einem Punkt `.` sagen, wo der Code der linken Seite eingesetzt werden soll (siehe Beispiel 4).

### Beispiel 1

Nehmen wir mal an, wir wollen nur VP über 18 auswählen.

```
# ohne Pipe
filter(example_data, age > 18)

# mit Pipe
example_data %>% filter(age > 18)
```

### Beispiel 2

Das ist besonders nützlich, wenn Befehle verschachtelt sind. Nehmen wir mal an, wir wollen nur VP über 18 auswählen und uns nur die Gruppe und unsere abhängige Variable anzeigen lassen.

```
# ohne Pipe
select(filter(example_data, age > 18), group, dependent_variable)

# mit Pipe
example_data %>%
  filter(age > 18) %>%
  select(group, dependent_variable)
```

### Beispiel 3

Das funktioniert für fast alle Funktionen. Hinweis: `na.rm = TRUE` sorgt dafür, dass Missing Values bei der Berechnung des Mittelwerts weggelassen werden.

```
# ohne Pipe
mean(age, na.rm = TRUE)

# mit Pipe
age %>% mean(na.rm = TRUE)
```

### Beispiel 4

```
# ohne Pipe
lm(dependent_variable ~ group, data = example_data)

# mit Pipe
example_data %>% lm(dependent_variable ~ group, data = .)
```

## Vorgehen

Hinweis: Screenshot aus Field, Miles & Field (2012). Für fast alle Verfahren finden Sie in diesem Lehrbuch ähnliche Übersichten. Diese sind sehr wertvoll für die Prüfungsvorbereitung und darüber hinaus.

### 16.6.2. General procedure for MANOVA <sup>①</sup>

To conduct factorial MANOVA you should follow this general procedure:

- 1 *Enter data.*
- 2 *Explore your data: begin by graphing the data and computing descriptive statistics. You should check multivariate normality and take a look at the variance-covariance matrices for each group.*
- 3 *Set contrasts for all predictor variables: you need to decide what contrasts to do and to specify them appropriately for all of the independent variables in your analysis.*
- 4 *Compute the MANOVA: you can then run the main multivariate analysis of variance. Depending on what you found in the previous step, you might need to run a robust version of the test.*
- 5 *Run univariate ANOVAs: having conducted the MANOVA, you can follow it up with separate ANOVAs for each dependent variable.*
- 6 *Discriminant function analysis: better than the option above, consider running a discriminant function analysis.*

#### 1) Daten einlesen

1. Laden Sie die nötigen Pakete und setzen Sie ein sinnvolles Arbeitsverzeichnis.
2. Laden Sie den Datensatz `ocd_data.dat` über den Link [https://pzezula.pages.gwdg.de/data/ocd\\_data.dat](https://pzezula.pages.gwdg.de/data/ocd_data.dat) in R ein.
3. Kodieren Sie die Variable `group` in `ocd_data` als Faktor mit sinnvoller Baseline. Geben Sie der Gruppe "No Treatment Control" das Label "NT".

#### 2) Überblick über die Daten

##### Bedeutung der Variablen

Unser Beispieldatensatz enthält hypothetische Daten zur Evaluation von Therapieprogrammen bei Zwangsstörungen.

Variable	Beschreibung
<code>group</code>	Faktor, der angibt, welche Art von Therapie die Versuchsperson erhalten hat. NT = Keine Therapie, BT = Verhaltenstherapie, CBT = Kognitive Verhaltenstherapie
<code>actions</code>	Häufigkeit von Zwangshandlungen nach der Behandlung
<code>thoughts</code>	Häufigkeit von Zwangsgedanken nach der Behandlung

1. Erstellen Sie zunächst einen einfachen Scatterplot, der den Zusammenhang zwischen zwanghaftem Verhalten auf der x-Achse und zwanghaften Gedanken auf der y-Achse darstellt.
2. Fügen Sie dem Plot eine Regressionslinie hinzu.
3. Nutzen Sie den Befehl `facet_wrap()`, um den Plot aus Aufgabe 2.2 nach Gruppen getrennt darzustellen.
4. Nutzen Sie den Code `ocd_data %>% select(actions, thoughts) %>% by(ocd_data$group, cov)`, um sich die Varianz-Kovarianz-Matrizen für jede Gruppe anzeigen zu lassen.

- Nutzen Sie den Code `ocd_data %>% by(ocd_data$group, psych::describe)`, um sich für jede Gruppe deskriptive Daten ausgeben zu lassen. Können Sie den Befehl verstehen? *Falls Sie eine Fehlermeldung bekommen, installieren Sie das Paket psych.*
- Führen Sie den unten stehenden Code aus, um die Annahme der multivariaten Normalverteilung an Ihren Daten zu überprüfen. Können Sie den Code verstehen? Sind die Daten in jeder Gruppe multivariat Normalverteilt? *Falls Sie eine Fehlermeldung bekommen, installieren Sie das Paket mvnortest.*
- Der Box's M-Test prüft die Gleichheit der Varianz-Covarianz-Matrizen. Führen Sie den Befehl `boxM()` aus der `library(heplots)` durch, um diese Voraussetzung zu überprüfen. Können wir von Gleichheit der Varianz-Covarianz-Matrizen ausgehen? **Falls Sie eine Fehlermeldung bekommen, installieren Sie das Paket heplots.**

```
# Daten vorbereiten
nt <- ocd_data %>% filter(group == "NT") %>% select(2:3) %>% t()
bt <- ocd_data %>% filter(group == "BT") %>% select(2:3) %>% t()
cbt <- ocd_data %>% filter(group == "CBT") %>% select(2:3) %>% t()

# Tests durchführen
mvnortest::mshapiro.test(nt)
mvnortest::mshapiro.test(bt)
mvnortest::mshapiro.test(cbt)
```

### 3) MANOVA durchführen

#### Vorab: Erklärung

MANOVAS können in R mit dem Befehl `manova()` durchgeführt werden. Dieser Befehl funktioniert genau so wie `lm()` und `aov()` in der Form: `manova(outcome ~ predictor, data = data)`. Der Unterschied ist, dass im Vorfeld alle verwendeten Outcome-Variablen mit dem `cbind()`-Befehl zu einem Objekt "zusammengeschnürt" werden.

- Setzen Sie Kontraste für die Interpretation der Analyse. Das funktioniert genau so wie bei ANOVAs und Regressionen. Schauen Sie zur Not in Ihren Aufzeichnung von früheren Übungszetteln nach.
  - Erster Kontrast: Vergleich von BT und NT
  - Zweiter Kontrast: Vergleich von CBT und NT *Hinweis: Hier handelt es sich um nicht-orthogonale Kontraste. Das ist an dieser Stelle in Ordnung, weil wir nur eine Prädiktorvariable haben. (siehe Field, Kap. 16.6.6: Setting Contrasts) Bemerkung: Wir müssen keine Kontraste setzen. Bei den Default-Einstellungen würde BT zur Referenzgruppe und wir würden den Unterschied, auch der Gruppe CBT als Effekt prüfen.*
- Erstellen Sie das Outcome-Objekt, indem Sie `ocd_data$thoughts` und `ocd_data$actions` mit Hilfe von `cbind()` verbinden.
- Nutzen Sie den Befehl `manova()`, um die Analyse durchzuführen. Speichern Sie das Ergebnis in einem Objekt.
- Wenden Sie auf das Objekt aus 3.3 den Befehl `summary()` mit dem zusätzlichen Argument `intercept = TRUE` an.
- Welchen Schluss können Sie aus dem Output ziehen?

### 4) MANOVA interpretieren

- Wenden Sie den Befehl `summary.aov()` auf ihr MANOVA-Modell an.
- Interpretieren Sie die Ergebnisse kurz.
  - Dürfen wir auf Grundlage dieser Ergebnisse die geplanten Kontraste untersuchen?
- Erstellen Sie für jedes einzelne Outcome ein eigenes ANOVA-Modell und betrachten Sie den Output, um Ihre Kontraste zu interpretieren.
- Was folgern Sie aus den Ergebnissen?

### 5) Rendern (knit)

Lassen Sie die Datei mit `Strg + Shift + K` (Windows) oder `Cmd + Shift + K` (Mac) rendern. Sie sollten nun im "Viewer" unten rechts eine "schön aufpolierte" Version ihrer Datei sehen. Falls das klappt: Herzlichen Glückwunsch! Ihr Code kann vollständig ohne Fehlermeldung gerendert werden. Falls nicht: Nur mut, das wird schon noch! Gehen Sie auf Fehlersuche! Ansonsten schaffen wir es ja in der Übung vielleicht gemeinsam.

## Literatur

*Anmerkung:* Diese Übungszettel basieren zum Teil auf Aufgaben aus dem Lehrbuch *Discovering Statistics Using R* (Field, Miles & Field, 2012). Sie wurden für den Zweck dieser Übung modifiziert, und der verwendete R-Code wurde aktualisiert.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. London: SAGE Publications Ltd.

## English Version

### Links

[Exercise sheet as PDF](#)

### Some hints

1. Please try to solve this sheet in an .Rmd file. You can create one from scratch using File > New file > R Markdown . . . You can delete the text beneath *Setup Chunk* (starting from line 11). Alternatively, you can download our template file unter [this link](#) (right click > save as...).
2. You'll find a lot of the important information on the [website of this course](#)
3. Please don't hesitate to search the web for help with this sheet. In fact, being able to effectively search the web for problem solutions is a very useful skill, even R pros work this way all the time! The best starting point for this is the [R section on the programming site Stackoverflow](#)
4. On the R Studio website, you'll find highly helpful [cheat sheets](#) for many of R topics. The [base R cheat sheet](#) might be a good starting point.

### Ressources

Since this is a hands-on seminar, we won't be able to present each and every new command to you explicitly. Instead, you'll find here references to helpful resources that you can use for completing this sheets.

Ressource	Description
Field, chapter 16	Book chapter explaining step by step the why and how of multivariate analyses of variance in R. <b>Highly recommended!</b>

### Hint of the week

Note: This week's hint will be a bit longer than usual.

#### 1) Using commands without loading packages

Usually, you'll have to load a package using `library()` in order to use functions from that package. However, there's a trick for when you'll only need a function sporadically: Adding the package name in front of the function using a double colon like this: `package::function()`. This also helps in using a function with a name that occurs in multiple packages. E.g., you might have noticed the warning `dplyr::filter() masks stats::filter()` when loading the tidyverse. Here, `filter()` alone will use the dplyr-command. If you want to use the filter-function from the stats-commands, you can do so by typing `stats::filter()`.

The relevant packages don't need to be loaded for this to work. However, they have to be installed on your computer.

#### 2) How exactly does the pipe work? %>%

Originally, the pipe stems from the package `magrittr` and is used frequently in the tidyverse. You mostly know piping from `dplyr`, but as long as you have loaded either `magrittr` or `dplyr` (using the `library()`-function), you'll be able to use the pipe for almost all R commands - if that is something you want to do. It works like this:

R inserts the code on the *left hand side* of the pipe “under the hood” into the *right hand side* of the pipe, the default being using it as the first argument of a function. This way you can add additional arguments with a comma behind it. If you want the left code to be inserted at a different location, you can use the dot `.` to tell R where to put it (see example no. 4).

#### Example no. 1

Say you only want to include subjects older than 18.

```
# without pipe
filter(example_data, age > 18)

# with pipe
example_data %>% filter(age > 18)
```

#### Example no. 2

This is most useful when you have nested commands. Here, we only want to include subjects older than 18, and only look at the group variable and our dependent variable.

```
# without pipe
select(filter(example_data, age > 18), group, dependent_variable)

# with pipe
example_data %>%
  filter(age > 18) %>%
  select(group, dependent_variable)
```

#### Example no. 3

This works for almost all functions. Note: `na.rm = TRUE` excludes missing values when calculating the mean.

```
# without pipe
mean(age, na.rm = TRUE)

# with pipe
age %>% mean(na.rm = TRUE)
```

#### Example no. 4

```
# without pipe
lm(dependent_variable ~ group, data = example_data)

# with pipe
example_data %>% lm(dependent_variable ~ group, data = .)
```

### Procedure

Note: Screenshot from Field, Miles & Field (2012). There are summaries like this for almost all statistical procedures covered in this seminar. They're invaluable for preparing for our exam and for life beyond this seminar.

## 16.6.2. General procedure for MANOVA <sup>①</sup>

To conduct factorial MANOVA you should follow this general procedure:

- 1 *Enter data.*
- 2 *Explore your data:* begin by graphing the data and computing descriptive statistics. You should check multivariate normality and take a look at the variance–covariance matrices for each group.
- 3 *Set contrasts for all predictor variables:* you need to decide what contrasts to do and to specify them appropriately for all of the independent variables in your analysis.
- 4 *Compute the MANOVA:* you can then run the main multivariate analysis of variance. Depending on what you found in the previous step, you might need to run a robust version of the test.
- 5 *Run univariate ANOVAs:* having conducted the MANOVA, you can follow it up with separate ANOVAs for each dependent variable.
- 6 *Discriminant function analysis:* better than the option above, consider running a discriminant function analysis.

### 1) Load data

1. Load the necessary packages and set a reasonable working directory.
2. Load the dataframe `ocd_data.dat` from the link [https://pzezula.pages.gwdg.de/data/ocd\\_data.dat](https://pzezula.pages.gwdg.de/data/ocd_data.dat) into R
3. Code the `group` variable as a factor with a reasonable baseline. Give the “No Treatment Control” group the level “NT”.

### 2) Data overview

#### Meaning of the variables

Our data example contains hypothetical data of an evaluation study on different therapies for obsessive compulsive disorder

Variable	Meaning
<code>group</code>	Factor specifying the therapy method: NT = No Treatment, BT = behavioral therapy, CBT = cognitive behavioral therapy
<code>actions</code>	Frequency of obsessive actions after the therapy
<code>thoughts</code>	Frequency of obsessive thoughts after the therapy

1. First, create a simple scatterplot showing the relationship between obsessive actions on the x-axis and obsessive thoughts on the y-axis.
2. Add a regression line to the plot.
3. Use the command `facet_wrap()` to show the plot from 2.2 separately for each group.
4. Use the command `ocd_data %>% select(actions, thoughts) %>% by(ocd_data$group, cov)` to get separate variance-covariance-matrices for each group.
5. Use the command `ocd_data %>% by(ocd_data$group, psych::describe)` to get descriptive statistics for each group. Are you able to understand this command? **If you get an error message, you might have to install the package `psych`.**
6. Use the code below to investigate whether the data satisfy the assumption of being normally distributed. Can you understand the code? Is the data multivariately normally distributed for each group? **If you get an error message, you might have to install the package `mvnortest`.**

7. The Box's M-test checks for equal variance-covariance-matrices. Use the command `boxM()` of library(`heplots`) to check that. Do we violate this assumption? **If you get an error message, you might have to install the package `heplots`.**

```
# prepare data
nt <- ocd_data %>% filter(group == "NT") %>% select(2:3) %>% t()
bt <- ocd_data %>% filter(group == "BT") %>% select(2:3) %>% t()
cbt <- ocd_data %>% filter(group == "CBT") %>% select(2:3) %>% t()

# run the tests
mvnormtest::mshapiro.test(nt)
mvnormtest::mshapiro.test(bt)
mvnormtest::mshapiro.test(cbt)
```

### Subtask 7

```
res.boxm <- heplots::boxM(ocd_data[,c('actions', 'thoughts')], group=ocd_data$group)
res.boxm
```

```
##
## Box's M-test for Homogeneity of Covariance Matrices
##
## data: ocd_data[, c("actions", "thoughts")]
## Chi-Sq (approx.) = 8.8932, df = 6, p-value = 0.1797
# summary(res.boxm) # for details
```

The p-value of our BoxM-test is not below 0.05, therefore we stay with  $H_0$  and state, that the variance-covariance-matrices don't differ significantly.

## 3) Run the MANOVA

### First, some explanation

You can run a MANOVA in R using the command `manova()`. This works exactly like the `lm()` and `aov()` commands you already know, the pattern being `manova(outcome ~ predictor, data = data)`. The difference, however, is that you have to bind together all relevant outcome variables beforehand, using the `cbind()` command.

1. Set the contrasts for this analysis. This works the same way as for ANOVAs and regressions. If you're having trouble, consult your notes on earlier sheets covering these topics.
  - a) First contrast: comparison between BT and NT
  - b) Second contrast: comparison between CBT and NT *Note: These contrasts are non-orthogonal. Here, that's ok because we only have one predictor variable (cf. Field, ch. 16.6.6: Setting Contrasts) Note: We don't have to specify contrasts manually. Default contrasts would take BT as reference group and the difference to group CBT would be one of our effects.*
2. Create the necessary outcome object by using `cbind()` to join `ocd_data$thoughts` and `ocd_data$actions` together.
3. Use the `manova()` command to run the analysis. Save the result as an object.
4. Use the `summary()` command on the MANOVA object, including the additional argument `intercept = TRUE`.
5. What conclusions can you draw from the output?

## 4) Interpreting the MANOVA

1. Use the command `summary.aov()` on the MANOVA model.
2. Briefly interpret the results.
  - a) Do these results justify analysing the planned contrasts?
3. Create a separate ANOVA model for each of the outcome variables and investigate the output in relation to your contrasts.



4. What do you conclude from these results?

## 5) Rendering (knit)

Render this file using `Ctrl + Shift + K` (Windows) or `Cmd + Shift + K` (Mac). In the viewer you should see a pretty versionn of your file. If this works: Congratulations! Your code can be rendered completely and without error codes! If it doesn't: No worries, you'll get there! Go hunting for errors in your code! Otherwise, we'll get it to render in the next seminar session!

## Literature

*Note:* These sheets are based partially on exercises from the book *Discovering Statistics Using R* (Field, Miles & Field, 2012). They've been modified for the porpuses of this seminar, and the R code was updated.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. London: SAGE Publications Ltd.