LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Datenanalyse

Sitzung 11: Datenvisualisierung und Fragestunde vor der Klausur

Institut für Kommunikationswissenschaft und Medienforschung
Ludwig-Maximilians-Universität München

Ablauf der Sitzung

1. Infos zur Klausur & Probeklausur
2. Antworten auf die FAQs zu R
3. Das WARUM der Datenvisualisierung
4. Das WIE der Datenvisualisierung
5. Datenvisualisierung mit ggplot: Grammar of Graphics
   - Boxplot: geom_boxplot()
   - Histogram.: geom_histogram()
   - Säulen / Balkendiagramm: geom_bar()
   - Streudiagramm: geom_point()
6. Datenvisualisierung mit tidycomm
7. Übung 1: Datenvisualisierung

1. Infos zur Klausur & Probeklausur

Klausur am 19. Juli

- 90 Minuten
- Open-Book-Klausur (dazu gleich mehr)
- Beispiele für typische Aufgabentypen:
  - **Rechnen**: Sie müssen z. B. richtigen Test wählen, statistische Kennzahlen berechnen, ggf. Entscheidung mit Blick auf Hypothese treffen
  - **Multiple-Choice**: Sie müssen sich z. B. entscheiden, welche Aussagen zu statistischen Tests (in-)korrekt sind
  - **R-Output/R-Code interpretieren**: Sie kriegen z. B. den Output eines statistischen Tests und müssen dokumentieren, wie die Ergebnisse zu interpretieren sind.

Probeklausur

- Sie finden unter Moodle eine Probeklausur mit und ohne Lösungen

Probeklausur

- Sie finden unter Moodle eine Probeklausur mit und ohne Lösungen
- Bereiten Sie sich **zuerst** auf die Klausur vor, als ob Sie „morgen“ wäre (wie – dazu gleich mehr).
- Rechnen Sie dann die Probeklausur (ohne Lösungen!) unter **„realen“ Bedingungen**: mit Notizen, ohne Handy, ggf. mit „Zeit stoppen“
- Schauen Sie **erst anschliessend** in die Lösungen & überprüfen Sie Ihre Ergebnisse: Was hat noch nicht funktioniert? Bei welchen Aufgaben hatten Sie inhaltlich/zeitlich Schwierigkeiten?

Probeklausur – Beispielaufgabe Rechnen

Frage 6

Punkte: **6 Punkte**

Wissenschaftler:innen der LMU nehmen an, dass LMU-Studierende pro Woche durchschnittlich mehr Zeit an der Uni verbringen als die Studierenden der anderen bayerischen Unis. Hierzu fanden die Wissenschaftler:innen heraus, dass die 64 zufällig ausgewählten LMU-Studierenden im Durchschnitt 24,50 Stunden pro Woche an der Uni verbringen (s = 8,00). Als Referenzwert dient der Durchschnittswert von 22,50 Stunden pro Woche unter Studierenden an allen bayerischen Unis. Das Signifikanzniveau wird auf $\alpha = 1\ \%$ festgelegt.

a) Mit welchem Verfahren kann die Hypothese geprüft werden?

b) Berechnen Sie die entsprechende Prüfgröße und die Freiheitsgrade.

c) Ermitteln Sie den kritischen Wert.

d) Erläutern Sie, ob sich die Hypothese anhand der Ergebnisse bestätigen lässt und begründen Sie Ihre Hypothesenentscheidung.

Frage 1: Um welches Verfahren geht es hier?

Frage 2: Welche Kennwerte sind gegeben – und welche Kennwerte muss ich berechnen?

Frage 3: Was sind die relevanten Formeln?

Frage 4: Wie interpretiere ich die Ergebnisse korrekt?

Probeklausur – Beispielaufgabe Multiple Choice

Frage 5

Punkte: **3 Punkte**

Schauen Sie sich den folgenden R-Code genau an. Beantworten Sie anschließend die Fragen. Kreuzen Sie für jede Aussage jeweils "richtig" oder "falsch" an.

```
WoJ_filtered <- WoJ %>%
  filter(country == "UK" | country = "Denmark")


WoJ_filtered %>%
  dplyr::group_by(country) %>%
  describe(ethics_3)


WoJ_filtered %>%
  unianova(country, ethics_3)
```

	richtig	falsch
a) Mit der Funktion `filter()` können bestimmte Fälle aus dem Datensatz ausgewählt werden.	o	o
b) Mit der Funktion `filter()` können bestimmte Variablen aus dem Datensatz ausgewählt werden.	o	o
c) Durch den Code `filter(country == "UK" | country = "Denmark")` werden alle Fälle im Datensatz ausgewählt, die bei der Variable `country` die Ausprägungen `"UK"` und `"Denmark"` enthalten.	o	o
d) Der Code ist fehlerfrei.	o	o
e) Da es sich bei `country` um die Ausprägung einer Variable handelt, müsste diese im R-Code in Anführungszeichen gesetzt werden (wie z.B. `"UK"`).	o	o
f) Der Befehl unianova() ist als statistischer Signifikanztest am besten dafür geeignet, zwei Gruppen hinsichtlich systematischer Differenzen der Mittelwerte zu analysieren.	o	o

Probeklausur – Beispielaufgabe R-Code/Output interpretieren

Frage 7

Punkte: **2 Punkte**

Für eine Studie wollen wir auf Basis des Worlds-of-Journalism-Datensatzes für Journalist:innen in der Schweiz sowie in Deutschland testen, ob das Land, in dem Journalist:innen arbeiten ("country", Ausprägungen: "Germany" sowie "Switzerland") mit der Art der Anstellung korrelieren ("employment", Ausprägungen: "Full-Time" sowie "Part-Time"). Schauen Sie sich den folgenden R-Code und die zugehörige Ausgabe genau an:

```
1  library(tidyverse)
2  library(tidycomm)

3  WoJ %>%
4     filter(country == "Switzerland" | country == "Germany",
5     employment == "Full-time" | employment == "Part-time") %>%
      crosstab(country, employment, chi_square = TRUE)
```

Ausgabe:

```
A tibble: 2 × 3
employment Germany Switzerland
<chr>        <dbl>       <dbl>
Full-time      139         154
Part-time        5          69
Chi-square = 39.326, df = 1, p < 0.001, V = 0.327
```

Zu welcher Interpretation auf Basis dieser Ergebnisse kommen Sie im Hinblick auf einen Zusammenhang zwischen dem Land, in dem Journalist:innen arbeiten, und der Art der Anstellung? Interpretieren Sie den Effekt und beschreiben Sie, inwiefern sich dieser auf die Grundgesamtheit übertragen lässt.

Frage 1: Um welches Verfahren geht es hier bzw. was wurde gerechnet?

Frage 2: Wie interpretiere ich die Ergebnisse korrekt?

„Open Book“-Klausur: Was heißt das?

Sie können mitbringen…:

- Vorlesungsfolien; Übungsmaterialien; Lehrbücher; Notizen
- Wissenschaftlichen Taschenrechner (keine Smartphones!)

Sie können Notizen mitbringen, aber…

- Zeit ist knapp bemessen: Sie müssen schnell wissen, **was** gefragt wird; **wo** Sie ggf. weitere Informationen finden; **was** Sie rechnen müssen, etc.
- Sie haben **nicht** die Zeit, jede einzelne Lösung bzw. Rechenwege im Detail „nachzuschauen“

„Open Book“-Klausur: Was heißt das?

Herausforderung I:

Verstehen, welches Wissen gefragt ist (Sitzungsnummern bzw. um welchen Test es geht, steht nicht immer explizit in der jeweiligen Aufgabe)

Herausforderung II

Rechnen bzw. Interpretation unter Zeitdruck schaffen

„Open Book“-Klausur: Wie bereite ich mich vor?

- Folien der Vorlesung & Übung studieren:
  - **Definitionen kennen**: Was ist z. B. ein „Standardfehler“?
  - **Methoden kennen**: Voraussetzungen, Anwendung, Interpretation
  - **Aufgaben rechnen**:
    - Wissen, wann welche Methode gefragt ist und welche Informationen „gegeben sind“
    - Formeln verstehen und anwenden können
    - Rechenschritte belegen
    - Interpretationen „ausschreiben“

„Open Book“-Klausur: Wie bereite ich mich vor?

- Zusammenfassende Notizzettel machen
  - Wie erkenne ich (z.B. anhand einer Aufgabenstellung), welches Verfahren gemeint ist? Was sind Voraussetzungen dieser Verfahren? → z. B. je Kennwert/Test notieren
  - Wie berechne ich statistische Kennwerte: Was sind die relevanten Formeln? Was setze ich dort ein? z. B. je Kennwert/Test notieren
  - Wie interpretiere ich beispielhaft „typische“ Tests (z.B. t-test, Korrelation, Regression, etc.): Welche Kennwerte muss ich angeben? Wann werden Hypothesen abgelehnt/angenommen? → z. B. je Kennwert/Test notieren

2. Antworten auf FAQs zu R

Unterschied tidyverse vs. tidycomm

Tidyverse und **tidycomm** sind R-Packages, die jeweils verschiedene Funktionen bieten. Beide Pakete folgen derselben Schreiblogik für den Code, der gut zu lesen ist.

Die Funktionen von Tidyverse und tidycomm kann man kombinieren.

Tidyverse

- Meta-Package: enthält viele unterschiedliche Pakete (z.B. dyplr und ggplot)
- Verkettet Funktion mit dem **Pipe-Operator**
- Funktioniert nach einer intiutiven „tidy“ Logik, gut von oben nach unten zu lesen
- Beispiele für Funktionen aus dem dyplr-Paket: `select()`, `filter()`, `mutate()`, `summarize()`, `group_by()`

Tidycomm

- Kommunikationswissenschaftliches Paket
- Verkettet Funktionen mit dem **Pipe-Operator**
- Gleiche Schreiblogik wie tidyverse
- Eigene Funktionen, die für KW relevant sind: z.B. Häufigkeiten/Lagemaße/Streuungsmaße berechnen und veranschaulichen, Skalen verändern
- Beispiele für Funktionen: `describe()`, `describe_cat()`, `tab_frequencies()`, `tab_percentiles()`, `_scale()`-Funktionen

Objekte vs. Konsole

Objekte in R

- Ein **Objekt** in R kann alles mögliche sein – z.B. eine Variable, ein Vektor oder ein ganzer Datensatz
- Im **Source**–Fenster wird ein Objekt erstellt, indem ihm mit dem Zuweisungspfeil ein Wert zugewiesen wird
- Das erstellte Objekt wird in der **Environment** gespeichert und kann dort abgerufen werden
- In der **Konsole** kann man den Wert von einem Objekt auch abrufen, ohne dass es im Skript gespeichert wird, indem Sie den Objektnamen eintippen und auf Enter drücken

Alles, was exstiert, ist ein R-Objekt
Alles, was passiert, ist ein Funktionsaufruf

Objekte vs. Konsole

Konsolenausgabe

- Die Konsole gibt Ihnen das Ergebnis Ihres Codes aus, den Sie im Skript ausführen.
- Dies kann der Wert eines Objekts sein oder die Ergebnisse von Berechnungen, Funktionen oder anderen Befehlen.
- In der Konsole können Sie nach dem >-Symbol auch Code eingeben, z.B. um ihn zu testen, denn:

Was Sie in die Konsole eingeben, wird nicht gespeichert!

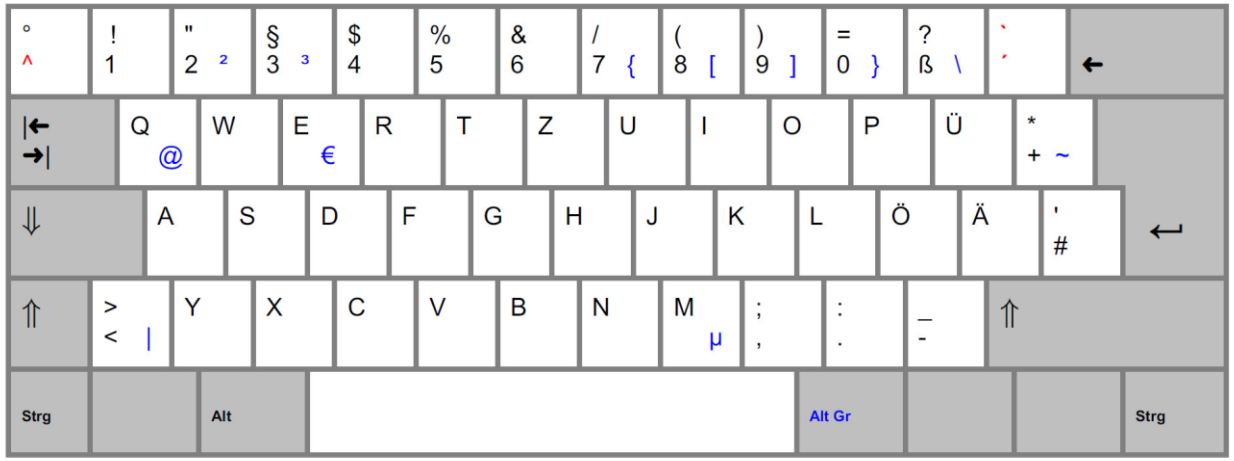
Stoppzeichen/Befehl abbrechen

- Das Stoppzeichen wird verwendet, um die Ausführung eines laufenden Befehls abrupt abzubrechen (Roter Punkt in der Konsole oder Esc-Taste).

> - Symbol

- Das **>-Symbol** in der Konsole zeigt, dass R bereit ist, einen Befehl entgegenzunehmen.
- Nachdem Sie einen Befehl eingegeben haben und die Eingabetaste gedrückt haben, wird der Befehl ausgeführt. Das Ergebnis wird unterhalb des >-Symbols in der Konsole mit Zeilenangaben in eckigen Klammern angezeigt.

Nützliche Shortcuts auf der Windows-Tastatur



Funktion/Zeichen	Shortcut Windows
Code ausführen	Strg + Enter
Befehl abbrechen	Esc
| (logisches Oder)	Alt Gr + <
Pipe Operator %>%	Strg + Shift + M
Zuweisungspfeil <-	Alt + - (Minus)
Konsole leeren	Strg + L
Neues R-Skript	Strg + Shift + N
Zeile Code zu Kommentar & andersherum	Strg + Shift + C
Zum Zeilenanfang/-ende springen	Alt + Pfeil links/rechts

Nützliche Shortcuts auf der Mac-Tastatur

Funktion/Zeichen	Shortcut Mac
Code ausführen	Cmd + Enter
Befehl abbrechen	Esc
| (logisches Oder)	Opt + 7
Pipe Operator %>%	Cmd + Shift + M
Zuweisungspfeil <-	Option + - (Minus)
Konsole leeren	Ctrl + L
Neues R-Skript	Cmd + Shift + N
Zeile Code zu Kommentar & andersherum	Cmd + Shift + C
Zum Zeilenanfang/-ende springen	Cmd + Pfeil links/rechts

Wichtige dyplr-Befehle (Tidyverse)

- `select()`: Wählt Spalten aus einem Datensatz aus. Kann Spalten nach Namen oder nach einem bestimmten Muster (z.B. mit `starts_with, ends_with, contains`) auswählen.
- `filter()`: Filtert Zeilen basierend auf angegebenen Bedingungen. Kann mehrere Bedingungen kombinieren (logisches UND: `&`, logisches ODER: `|`).
- `mutate()`: Fügt neue Spalten hinzu oder verändert bestehende Spalten.
- `summarize()`: Berechnet wichtige statistische Kennzahlen, zum Beispiel Mittelwert und Standardabweichung. In Kombination mit `group_by()` können Kennzahlen für verschiedene Gruppen berechnet werden.

Pipe Operator %>%

= Und mach damit das Folgende...

Der **Pipe-Operator %>%** (= “Rohr”, dass die Daten transportiert) verkettet eine Folge von Funktionen in einer klaren und lesbaren Weise.

Er ermöglicht es, Funktionen nacheinander auf **dasselbe Ausgangsobjekt** anzuwenden, so dass Schritt-für-Schritt-Transformationen auf die Daten angewendet werden können.

Daher rufen wir immer zuerst das Ausgangsobjekt auf und fügen dann jeden Transformationschritt separat durch den %>%-Operator hinzu.

Pipe Operator %>%

Beispiel:

```
25  WoJ %>%
26    select(employment, country) %>%
27    filter(country == "Germany") %>%
28    filter(employment == "Full-time")
```

Nimm mein R-Objekt WoJ und übergib es der Pipeline %>%
Wähle zwei Variablen aus dem Datensatz aus und übergib sie der Pipeline
Filtere nur deutsche Journalist*innen heraus und übergib den Output der Pipeline
Und filtere zusätzlich Vollzeitangestellte heraus

```
25  WoJ %>%
26    select(employment) %>%
27    filter(country == "Germany") %>%
28    filter(employment == "Full-time")
```

```
i In argument: `country == "Germany"`.
Caused by error:
! object 'country' not found
```

Probeweise wird die Variable *country* im select-Befehl nicht ausgewählt.
Da sich die filter-Funktion in der folgenden Zeile durch den Pipe-Operator auf den Output der Zeile 26 bezieht und nicht auf den gesamten Datensatz, kommt es zu einer Fehlermeldung.

Streuungsmaße in R

- Streuungsmaße: statistische Kennzahlen, die beschreiben, wie die Datenpunkte innerhalb eines Datensatzes ausgebreitet / „gestreut“ sind
- Spannweite (Range, R): Größe des Bereichs, in dem die Daten liegen
- Interquartils-Abstand bzw. IQR: Größe des Bereichs, in dem die mittleren 50% der Fälle liegen
- Varianz (s^2): Mittlere quadratische Abweichung vom arithmetischen Mittel, erst ab Intervallskalen anwendbar
- Standardabweichung (s): Wurzel aus der Varianz (Größenordnung der Werte), erst ab Intervallskalen anwendbar
- Variationskoeffizient (v): Normierte Standardabweichung (nur für Verhältnis- bzw. Ratio-Skalen)

Streuungsmaße in R (Tidyverse/Tidycomm)

Die Streuungsmaße können in R mit `dplyr::summarize()` oder `tidycomm::describe()` berechnet werden.

Streuungsmaße in R berechnen:

- `tidycomm::describe()`: berechnet per Default *M*, *SD*, *Q25*, *Q75* & *Range*, in Kombination mit `dplyr::mutate()` auch *IQR*, *Varianz* und den *Variationskoeffizient*
- `dplyr::summarize()`: Subfunktionen müssen angegeben werden (z.B. `mean()`, `var()`, uvm.)
  - flexibler als `tidycomm::describe()`, benötigt aber mehr Subfunktionen
- `tidycomm::tab_percentiles()` berechnet Perzentile

Vor den Funktionsaufrufen wird zur besseren Lesbarkeit der Paketname angegeben, aus dem die Funktion stammt.

Stichprobenziehung

- Wenn Stichproben durch eine echten Zufallsziehung gezogen werden, dann unterliegen sie (und ihre Kennwerte) einem **Stichprobenfehler**.
- Der **Standardfehler** ist die Standardabweichung für Kennwerte (z.B. Mittelwert, Anteilswert) über mehrere (gedachte) Stichproben hinweg.
- Das **Konfidenzintervall** ist ein Bereich um den beobachteten Kennwert herum, in welchem der wahre Kennwert (z.B. Mittelwert oder Anteilswert) wahrscheinlich liegt.
- Die akzeptierte **Irrtumwahrscheinlichkeit** (auch Signifikanzniveau oder Alpha-Niveau) gibt an, mit welcher Wahrscheinlichkeit die Aussage falsch ist, dass der wahre Kennwert außerhalb des Konfidenzintervalls liegt.
  - Zum Beispiel bedeutet ein Alpha-Niveau von 0.05 (5%), dass man in 5% der Fälle, in denen man die Stichprobe zieht und das Konfidenzintervall berechnet, erwarten kann, dass das Konfidenzintervall den wahren Kennwert nicht enthält.

Auswahl der statistischen Verfahren

AV / UV	Nominalskala	Ordinalskala	Metrische Skala (Intervall oder Verhältnis)
Nominalskala	**Chi-Quadrat-Test**	**Chi-Quadrat-Test**	**t-Test (2 Gruppen) / ANOVA (> 2 Gruppen)**
Ordinalskala	**Chi-Quadrat-Test**	Spearman-Korrelation	Spearman-Korrelation / **Lineare Regression**
Metrische Skala (Intervall oder Verhältnis)	(Multinominale) Logistische Regression	Spearman-Korrelation	Pearson-Korrelation / **Lineare Regression**

Achtung: Vor der Berechnung des Tests sollten immer erst die spezifischen Voraussetzungen des Tests geprüft werden!

Befehlsstruktur der statistischen Verfahren

Chi-Quadrat-Test

```
data %>%
 crosstab(variable1, variable2)


data %>%
 crosstab(variable1, variable2,
  add_total = TRUE,
  percentages = TRUE,
  chi_square = TRUE)
```

t-Test für unabhängige Stichproben

```
data %>%
 t_test(independent_variable,
      dependent_variable)
```

t-Test für abhängige Stichproben

```
data %>%
 t_test(independent_variable,
      dependent_variable,
      paired = TRUE,
      case_var = variable)
```

Befehlsstruktur der statistischen Verfahren

ANOVA

```
data %>%
unianova(independent_variable,
      dependent_variable,
      post_hoc = TRUE)
```

Pearson-Korrelation

```
data %>%
   correlate(variable1,
             variable2)
```

Lineare Regression

```
data %>%
   regress(dependent_variable,
        independent_variable1,
        independent_variable2,
        …,
       check_independenterrors = TRUE,
       check_multicollinearity = TRUE,
       check_homoscedasticity = TRUE)
```

3. Das WARUM der Datenvisualisierung

Das WARUM der Datenvisualisierung

Wiederholung:

- Alle erhobenen Daten bilden zusammen die sog. **Urliste**
- Für die **Rangwertreihe** werden die erhobenen Daten sortiert
- Die **Häufigkeitsverteilung** gibt für jeden Merkmalswert die Häufigkeit an, mit der dieser in den erhobenen Daten vorkommt

Das WARUM der Datenvisualisierung

Wiederholung:

- Datensätze in der Kommunikationswissenschaft umfassen jedoch oft **viele Fälle** bei einer **begrenzten Anzahl von Merkmalsausprägungen**.
- Je mehr Fälle ein Datensatz umfasst, desto schwieriger wird es, sich in der Urliste oder Rangwertreihe zurechtzufinden.

➔ Die Häufigkeitsverteilung kann hier helfen. Sie lässt sich **tabellarisch** oder **grafisch** darstellen.

Das WARUM der Datenvisualisierung

Wiederholung:

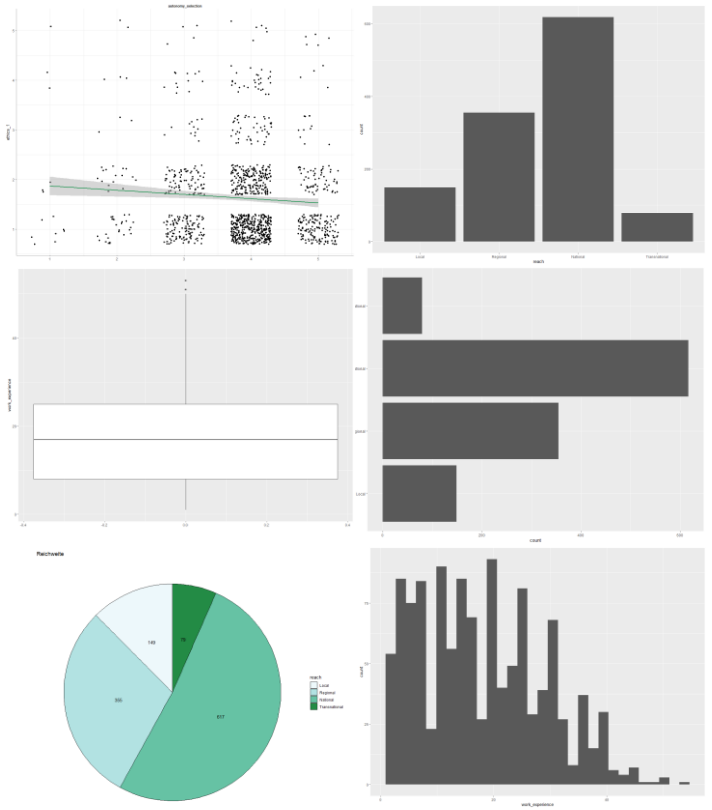
- Für viele Menschen ist eine Visualisierung der Daten mit einem Bild einfacher begreifbar als mit einer Tabelle
- Es haben sich unterschiedliche Darstellungsformen eingebürgert, die jeweilige Form hängt auch vom Skalenniveau des darzustellenden Merkmals ab

Nominale und ordinale Merkmale

- Kreisdiagramm (nur für nominalskalierte Merkmale!)
- Säulen-/Balkendiagramm

Metrische Merkmale

- Streudiagramm
- Histogramm
- Boxplot

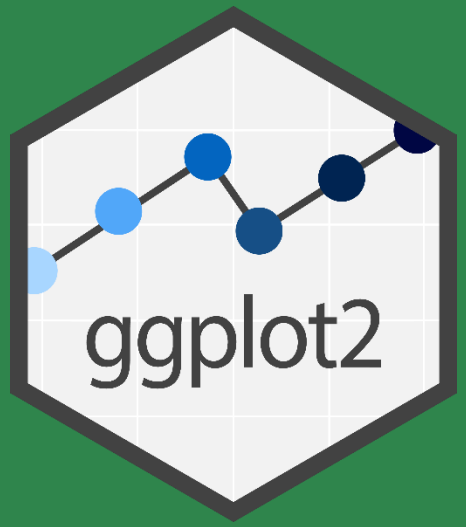


4. Das WIE der Datenvisualisierung

Das WIE der Datenvisualisierung

- Die Datenvisualisierung mit Base R ist möglich, jedoch sehr aufwendig
- Das ggplot2-Package aus dem tidyverse-Metapackage macht die Datenvisualisierung viel einfacher
- ggplot2 ist ein mächtiger Baukasten, um statistische Diagramme zu erstellen

Quelle: https://datasciencepartners.nl/tidyverse-r/



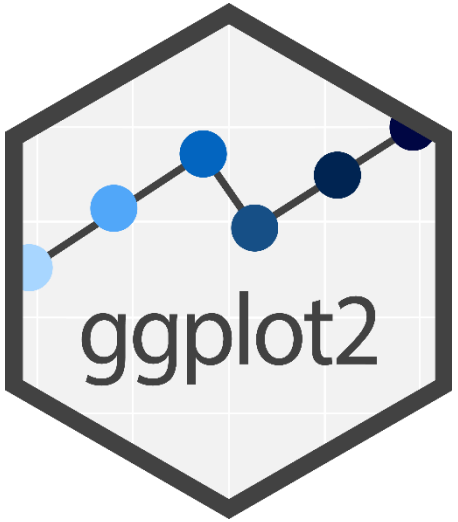
5. Datenvisualisierung mit ggplot: Grammar of Graphics

Grammar of Graphics

- Das gg in ggplot2 steht für **Grammar of Graphics**, was bedeutet, dass wir jede Komponente eines Graphen Schicht für Schicht und Komponente für Komponente beschreiben können.

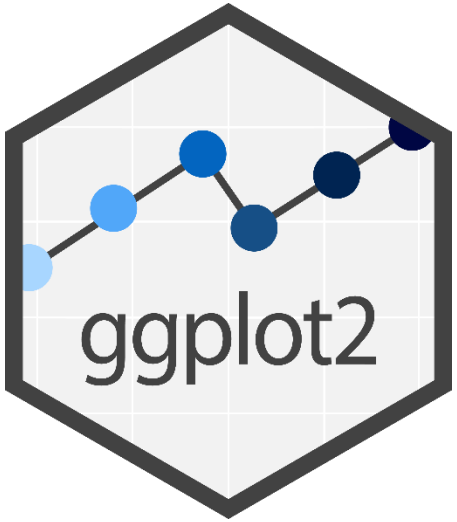
1. Ebene : Häufig ein Koordinatensystem
2. Ebene : z.B. Datenwerte
3. Ebene : z.B. Trendlinie mit Konfidenzintervall
4. Ebene : …

➔ ggplot2 weist den Variablen Objekte (Punkte, Linien, Balken) mit ästhetischen Attributen (Farbe, Form, Größe) zu.



Quelle: https://raw.githubusercontent.com/rstudio/hex-stickers/master/PNG/ggplot2.png

Grammar of Graphics



Notwendige Bestandteile sind:

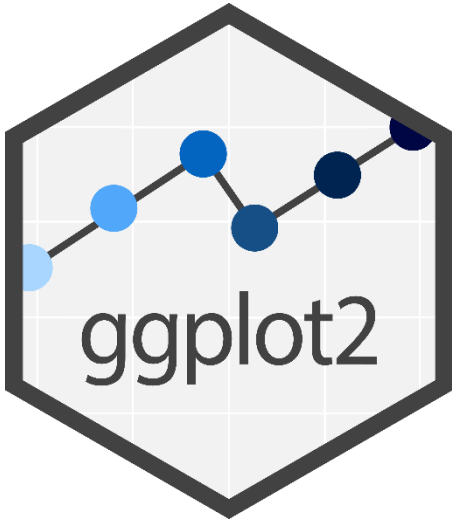
- Quellobjekt („data“): Daten, die visualisiert werden sollen
- Geometries („geom_“): Geometrieoptionen, um festzulegen, welche geometrischen Objekte die Daten darstellen sollen (z. B. Punkte, Balken, Linien und vieles mehr)
- Aesthetics („aes()“): Spezifikationen der Objekte (d. h. Position, Farbe, Größe, Form, Linientyp, Transparenz, etc.)

Quelle: https://raw.githubusercontent.com/rstudio/hex-stickers/master/PNG/ggplot2.png

Grammar of Graphics

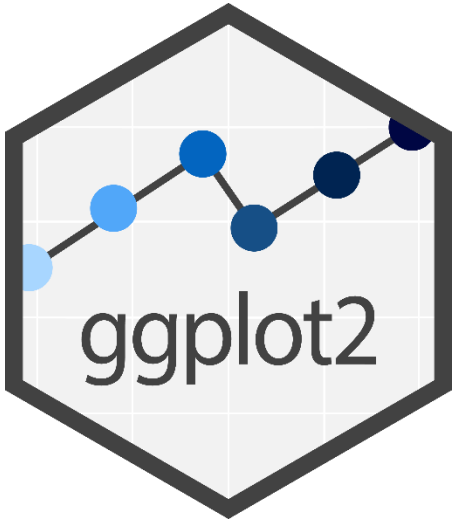
Optionale Bestandteile sind:

- Scales („scale_“): Skalierungsoptionen, um das Mapping von den Variablen bis zur Ästhetik abzustimmen (z.B. Achsenbegrenzungen, Teilstriche, etc.
- Statistical transformations („stat_“): Ergänzung statistischer Zusammenfassungen der Daten für die Visualisierung (z. B. Mittelwerte und Standardabweichungen
- Coordinate system („coord_“): Anpassung des Erscheinungsbilds des Koordinatensystems (z.B. um Koordinaten umzudrehen



Quelle: https://raw.githubusercontent.com/rstudio/hex-stickers/master/PNG/ggplot2.png

Grammar of Graphics



Ergänzende Bestandteile sind:

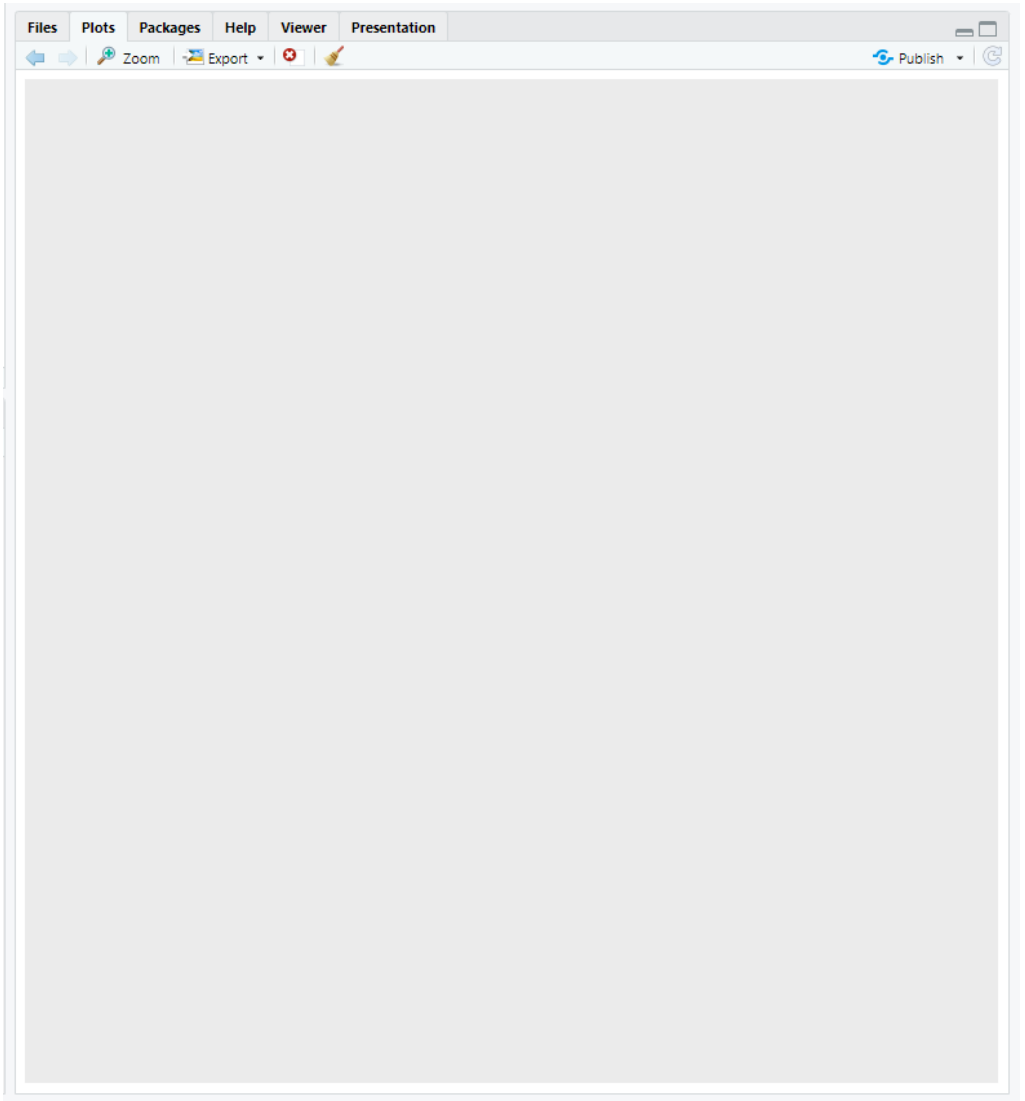
- Position: Anpassung überlappender Objekte (z. B. Stapeln)
- Facets („facet_“): Unterteilung des Hauptplots in mehrere Unterplots
- Visual themes („theme()“): Veränderung der visuellen Grundlagen eines Plots (z. B. Hintergrund, Schriftart, Größen und Farben)
- Axis labels („labs()“): Anpassung der Achsenbeschriftung

Quelle: https://raw.githubusercontent.com/rstudio/hex-stickers/master/PNG/ggplot2.png

Grammar of Graphics

Im **ersten Schritt** werden die **Daten geladen** (hier der WoJ-Datensatz)

```
4  WoJ %>%
5    ggplot()
```



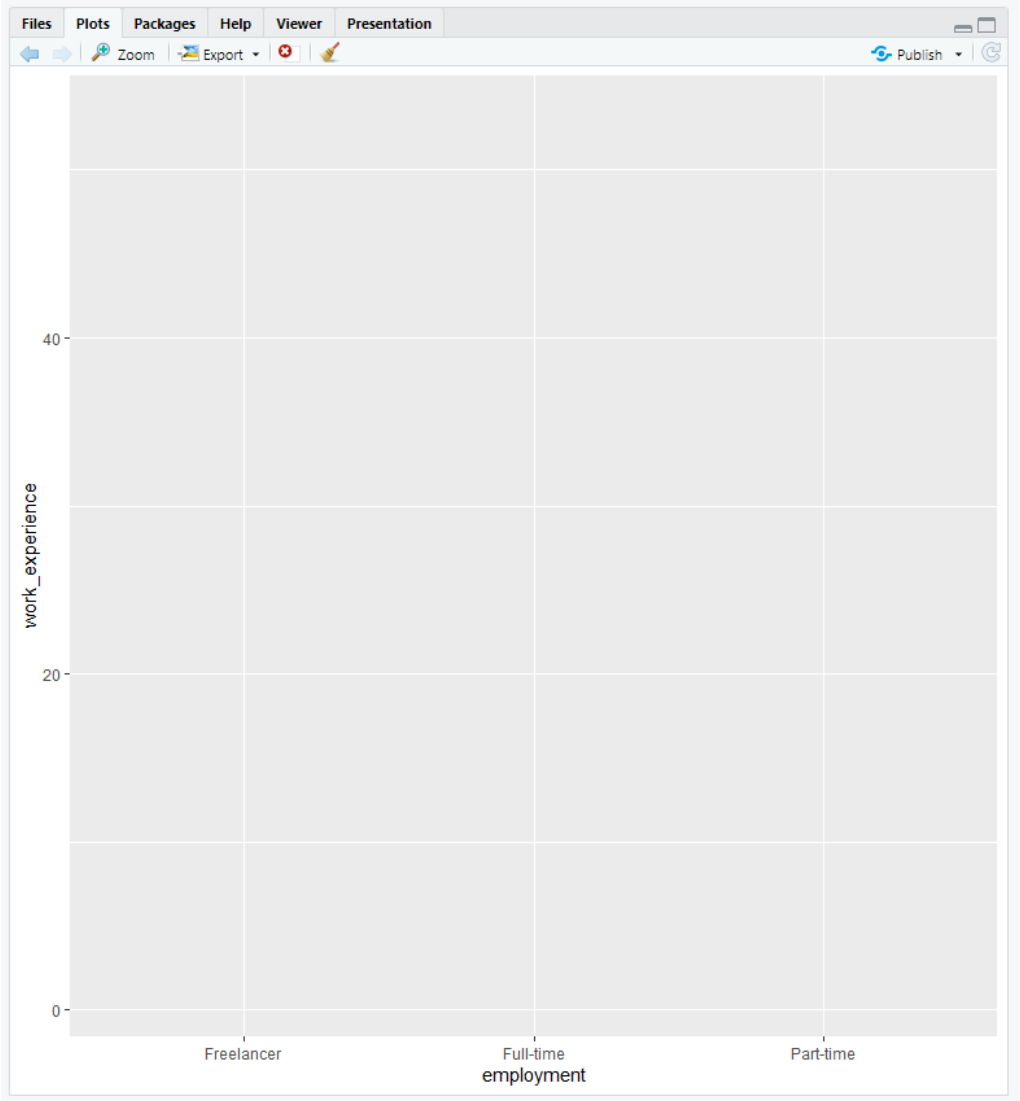
Die Funktion **ggplot()** erstellt jetzt eine leere Leinwand (d. h. die erste Ebene).

Grammar of Graphics

Um diese leere Leinwand zu füllen, muss der Funktion **ggplot()** mittgeteilt werden, welche **Variablen** der x- und y-Achse zugewiesen werden sollen.

Dafür wird die Funktion **aes()** verwendet.

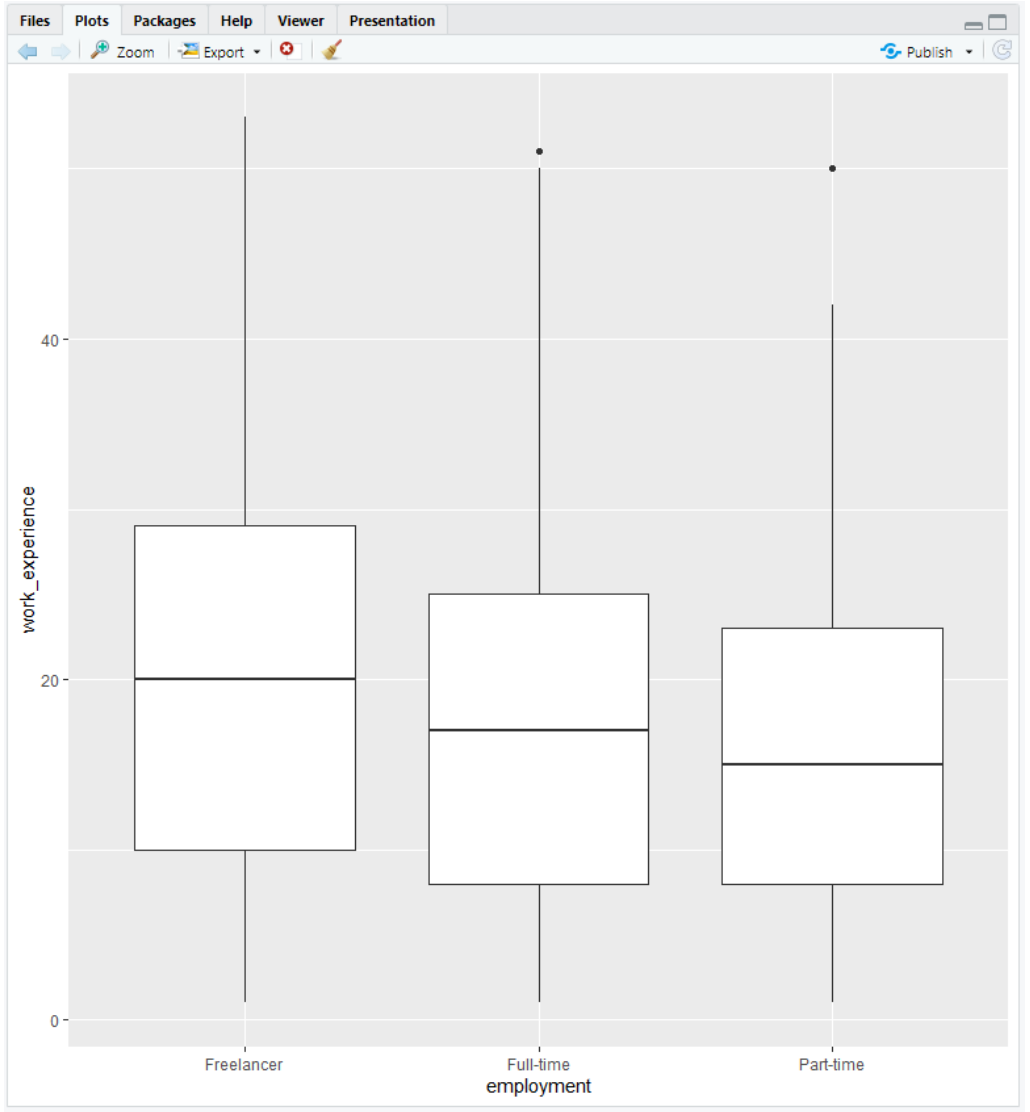
```
7  WoJ %>%
8    ggplot(aes(x = employment, y = work_experience))
```



Grammar of Graphics

Danach muss noch das **geometrische Objekt** bestimmt werden.

```
10  WoJ %>%
11    ggplot(aes(x = employment, y = work_experience)) +
12    geom_boxplot()
```

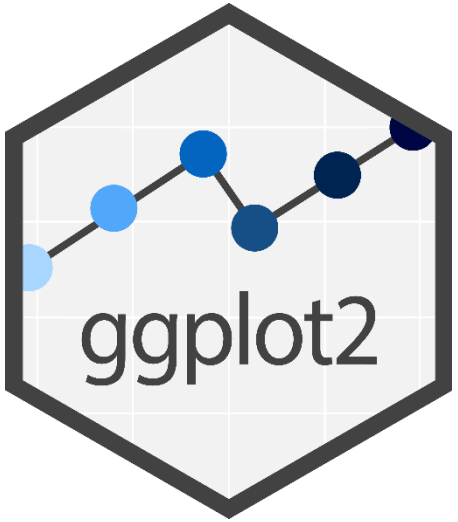


Grammar of Graphics

Mit der **geom_** Komponente der Funktion **ggplot()** können viele Diagrammtypen erstellt werden.

Besonders prominent:

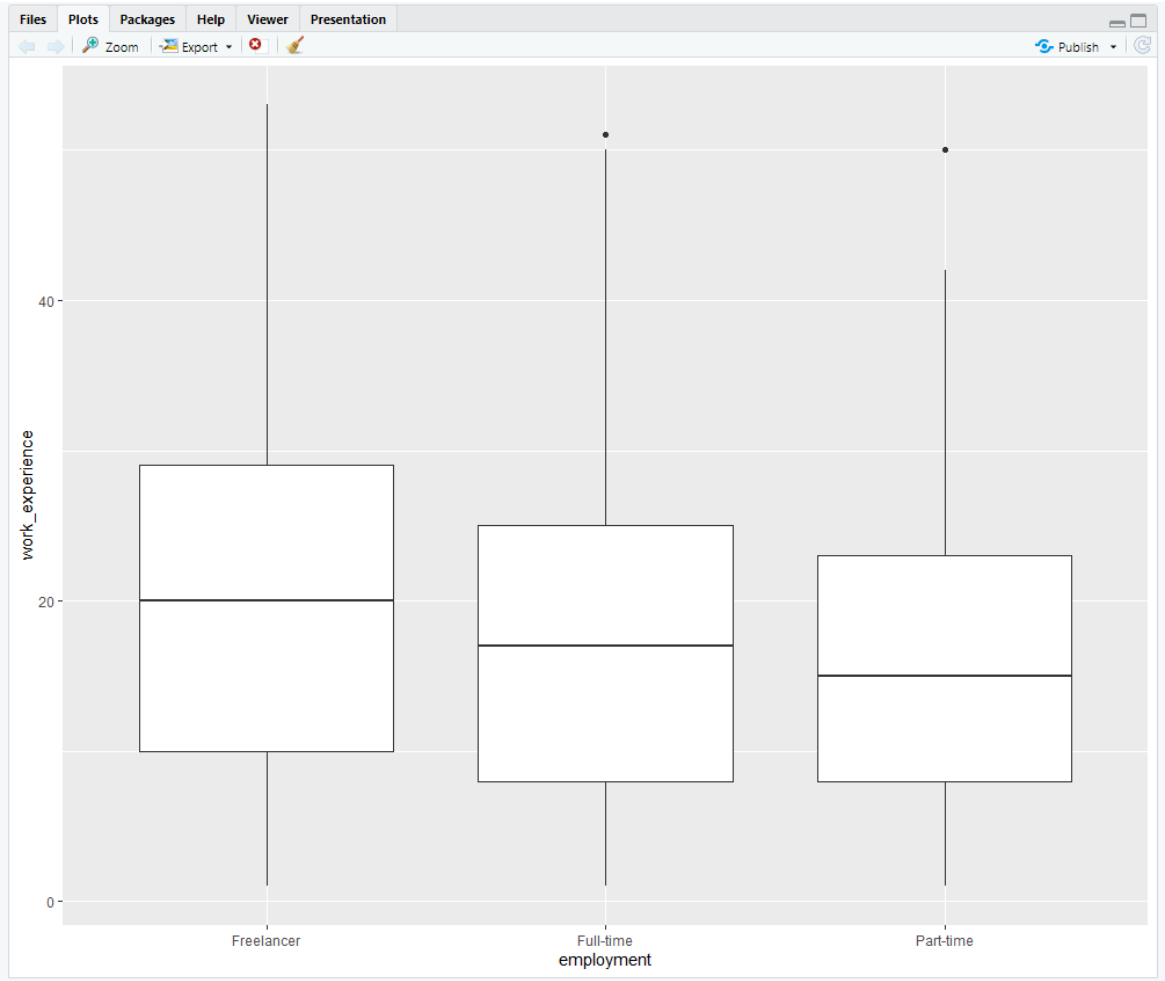
- **geom_boxplot()**: um einen Boxplot zu erstellen
- **geom_bar()**: um ein Balkendiagramm zu erstellen
- **geom_histogram**: um ein Histogramm zu erstellen
- **geom_point()**: um ein Streu- oder Blasendiagramm zu erstellen



Quelle: https://raw.githubusercontent.com/rstudio/hex-stickers/master/PNG/ggplot2.png

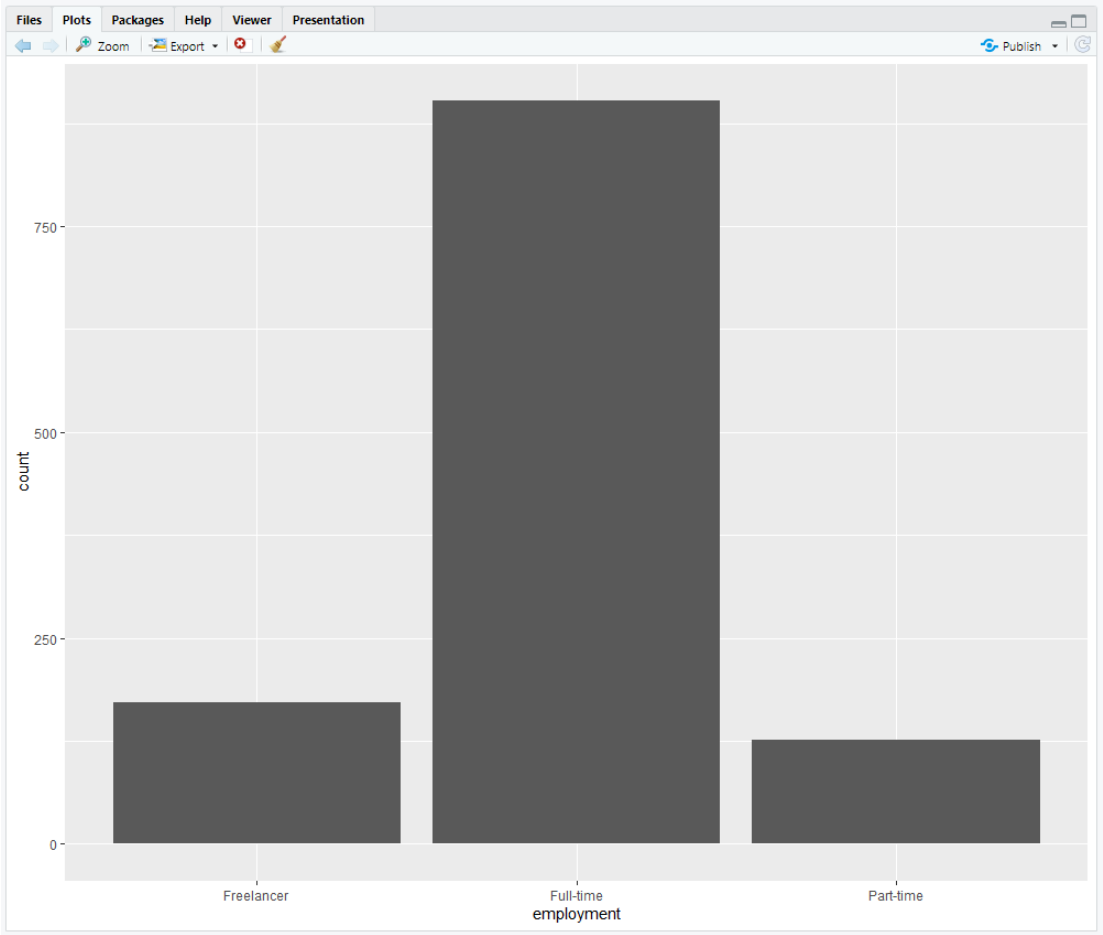
Grammar of Graphics: geom_boxplot()

```
14  # library(tidyverse)
15  # library(tidycomm)
16
17  WoJ %>%
18    ggplot(aes(x = employment, y = work_experience)) +
19    geom_boxplot()
```



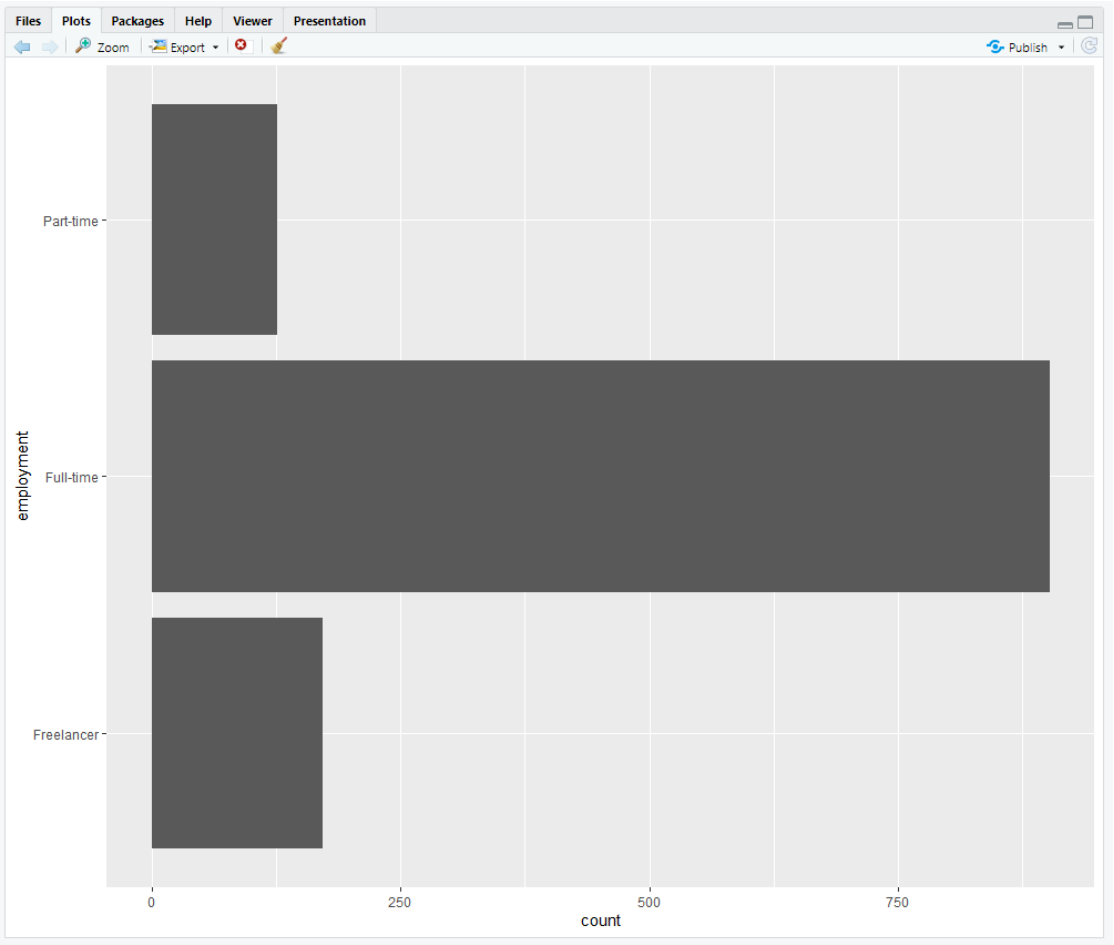
Grammar of Graphics: geom_bar()

```
21  # library(tidyverse)
22  # library(tidycomm)
23
24  WoJ %>%
25    ggplot(aes(x = employment)) +
26    geom_bar()
```



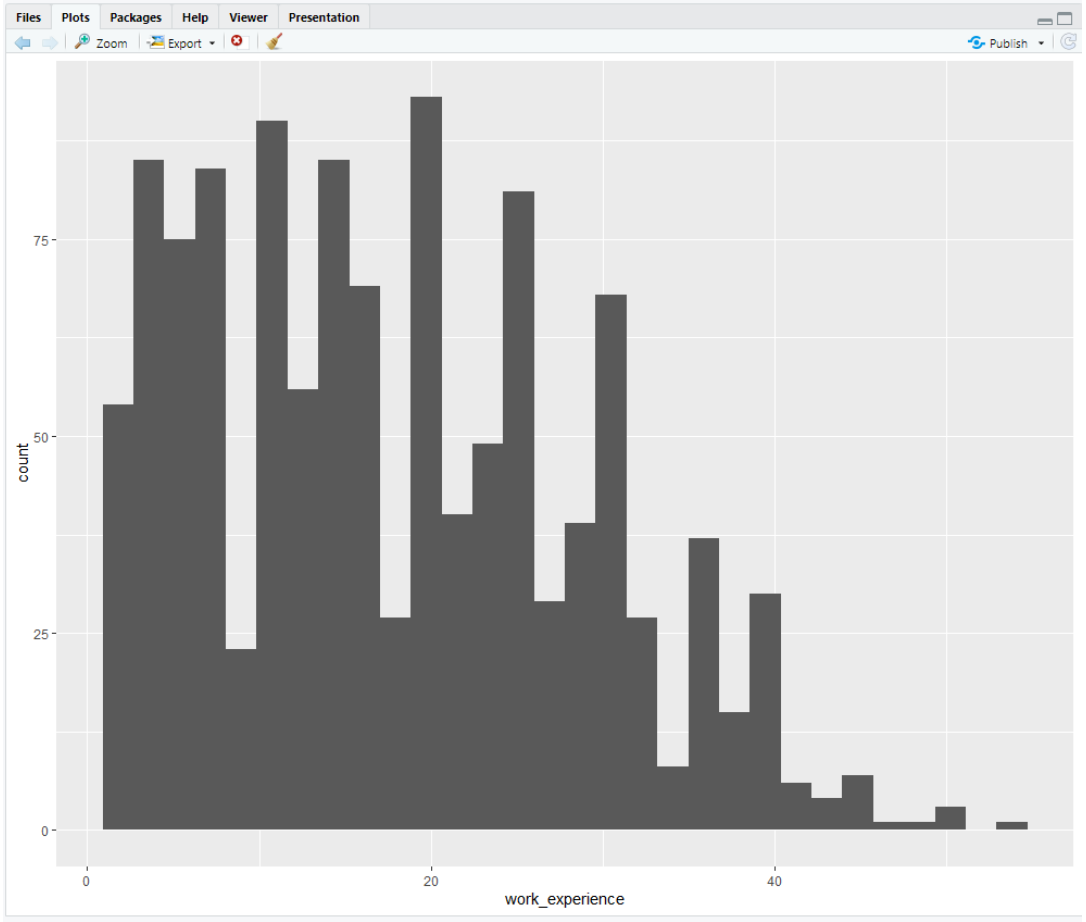
Grammar of Graphics: geom_bar()

```
31  WoJ %>%
32    ggplot(aes(x = employment)) +
33    geom_bar() +
34    coord_flip()
```



Grammar of Graphics: geom_histogram()

```
36  # library(tidyverse)
37  # library(tidycomm)
38
39  WoJ %>%
40    ggplot(aes(x = work_experience)) +
41    geom_histogram()
```



Grammar of Graphics: geom_point()

```
43  # library(tidyverse)
44  # library(tidycomm)
45
46  WoJ %>%
47    ggplot(aes(x = work_experience, y = ethics_1)) +
48    geom_point(shape = 21,
49               fill = "black",
50               alpha = 0.25,
51               color = "black")
```

6. DATENVISUALISIERUNG MIT TIDYCOMM

Datenvisualisierung mit tidycomm

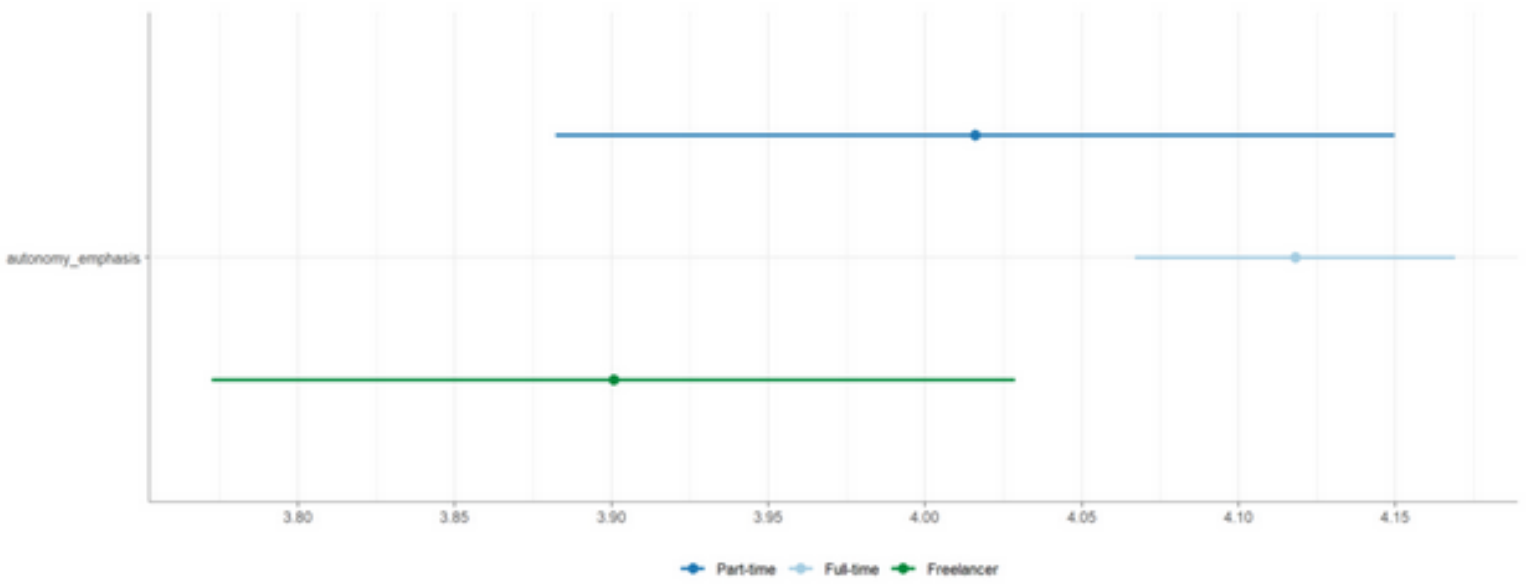
Um statistische Standardtestverfahren noch einfacher zu visualisieren, gibt es im **tidycomm-Package** den **visualize()-**Befehl.

Hier ein paar Beispiele zur Wiederholung…

Befehl:

```
WoJ %>%
  unianova(employment, autonomy_emphasis) %>%
  visualize()
```

Ausgabe:

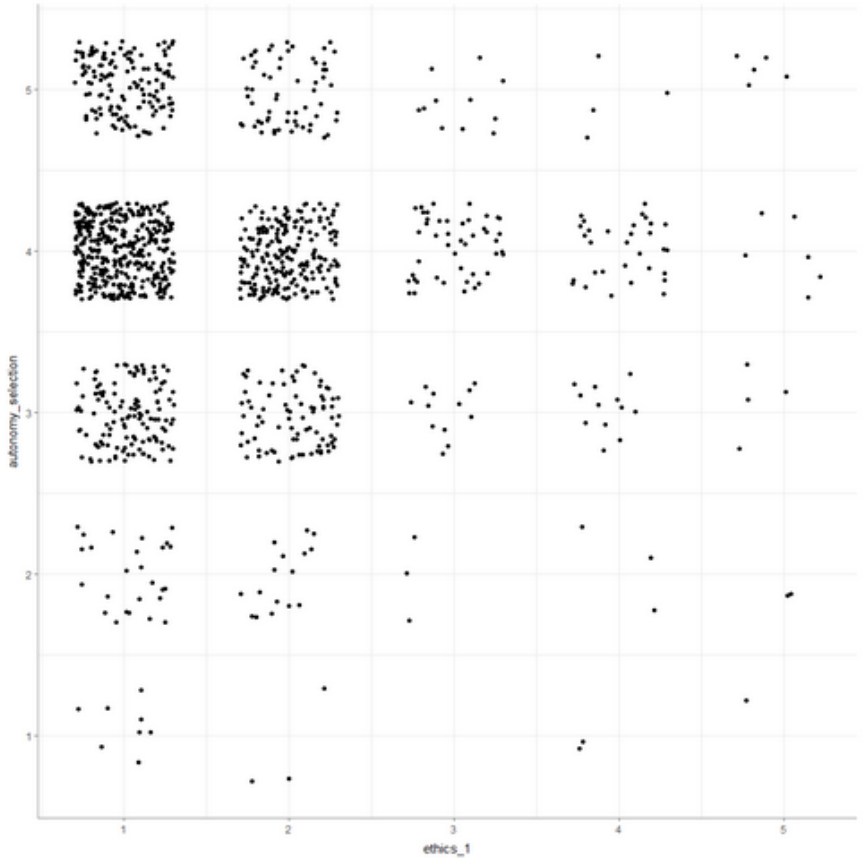


Visualisierung einer einfaktoriellen ANOVA in R

Befehl:

```
WoJ %>%
  correlate(ethics_1, autonomy_selection) %>%
  visualize()
```

Ausgabe:

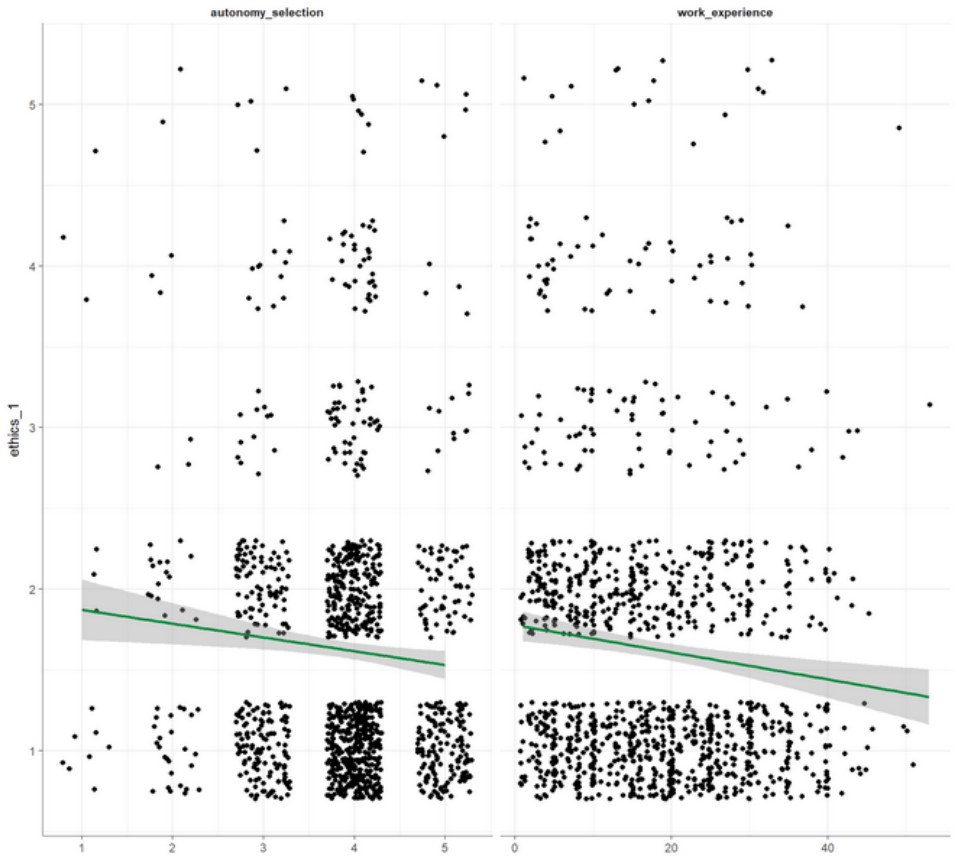


Visualisierung der Korrelation von zwei Variablen

Befehl:

```
WoJ %>%
  regress(ethics_1, autonomy_selection, work_experience) %>%
  visualize()
```

Ausgabe:

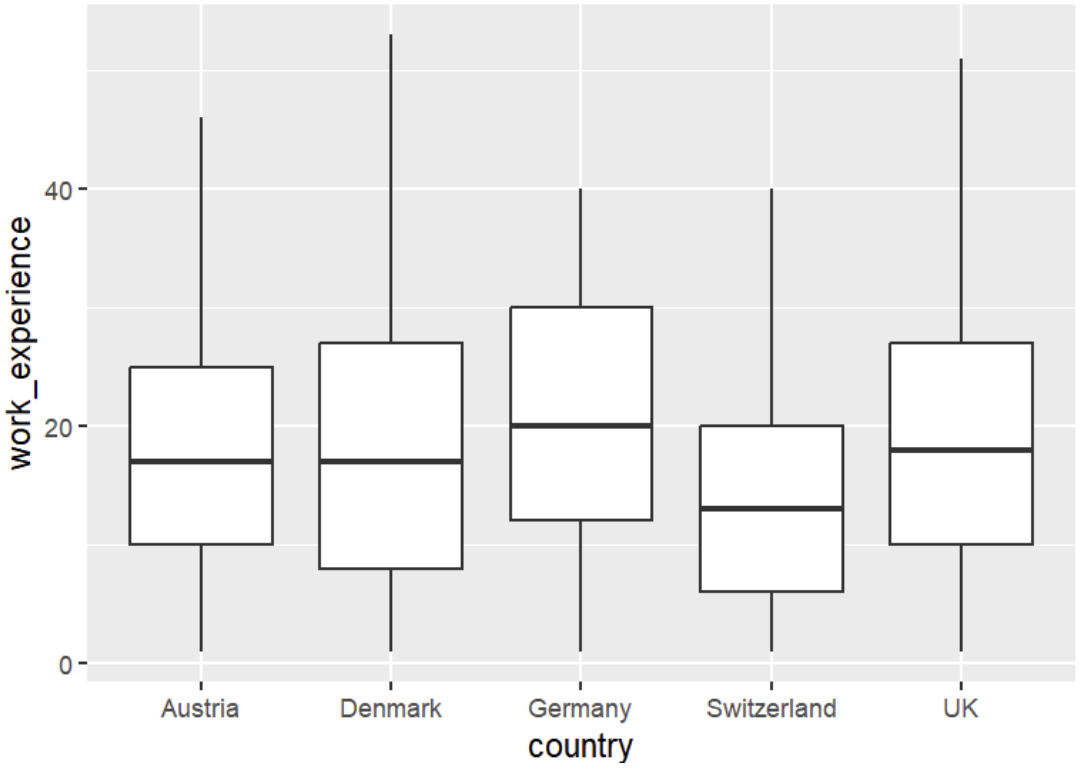


Basis-Visualisierung einer multiplen linearen Regression mit zwei Prädiktoren in R

7. Übung 1: Datenvisualisierung

Übung 1

Erstellen Sie für den WoJ-Datensatz einen Boxplot, der die Verteilung der Berufserfahrung (work_experience) der Journalist*innen getrennt für jedes Land darstellt. Weisen Sie dem Land die x-Achse und der Berufserfahrung die y-Achse zu.



Lösung für Übung 1

```
65  # Übung 1
66  WoJ %>%
67    ggplot(aes(x = country, y = work_experience)) +
68    geom_boxplot()
```