

# Datenvisualisierung

---

Institut für Kommunikationswissenschaft und Medienforschung  
Ludwig-Maximilians-Universität München



# Ablauf der Sitzung

1. Das WARUM der Datenvisualisierung
2. Das WIE der Datenvisualisierung
3. Datenvisualisierung mit ggplot: Grammar of Graphics
  - Boxplot: `geom_boxplot()`
  - Histogramm.: `geom_histogram()`
  - Säulen / Balkendiagramm: `geom_bar()`
  - Streudiagramm: `geom_point()`
4. Datenvisualisierung mit tidycomm
5. Übung 1: Datenvisualisierung

# 1. 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

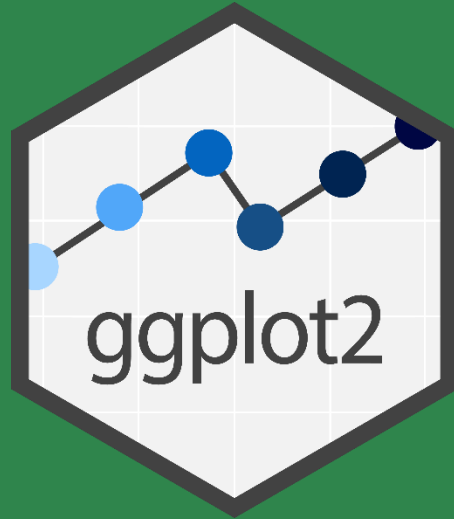


## 2. DAS WIE DER DATENVISUALISIERUNG

# 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



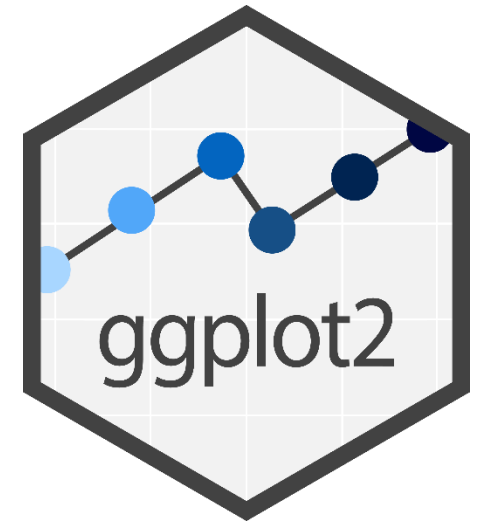


# 3. DATENVISUALISIERUNG MIT GG PLOT: GRAMMAR OF GRAPHICS

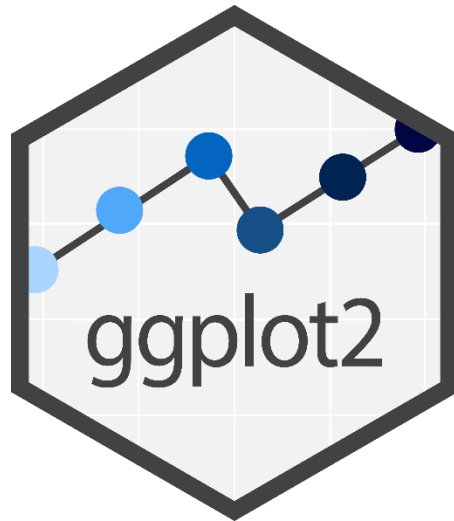
# Grammar of Graphics

- 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.



# Grammar of Graphics



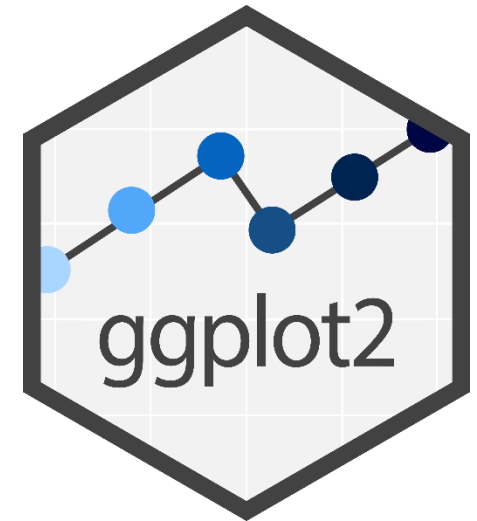
## Notwendige Bestandteile sind:

- 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.)

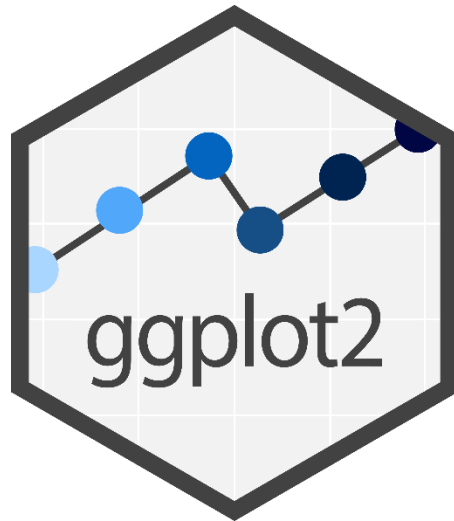
# Grammar of Graphics

## 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)



# Grammar of Graphics



## Ergänzende Bestandteile sind:

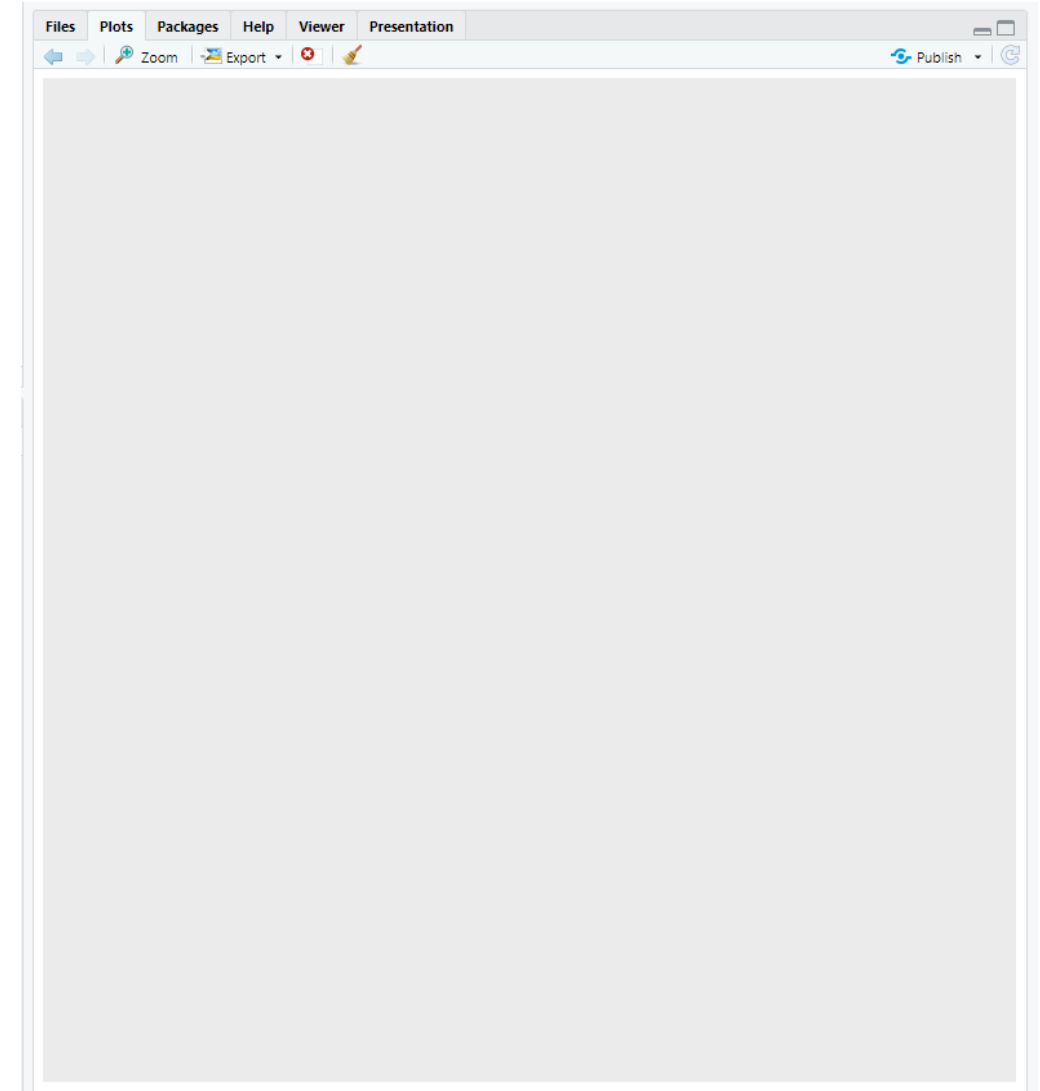
- Position: Anpassung überlappender Objekte (z. B. Stapeln)
- Facets („facet\_“): Unterteilung des Hauptplots in mehrere Unterplots
- Visual themes („theme()“): Veränderung der visuellen Grundlagen eines Plots (z. B. Hintergrund, Schriftart, Größen und Farben)
- Axis labels („labs()“): Anpassung der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

# 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).

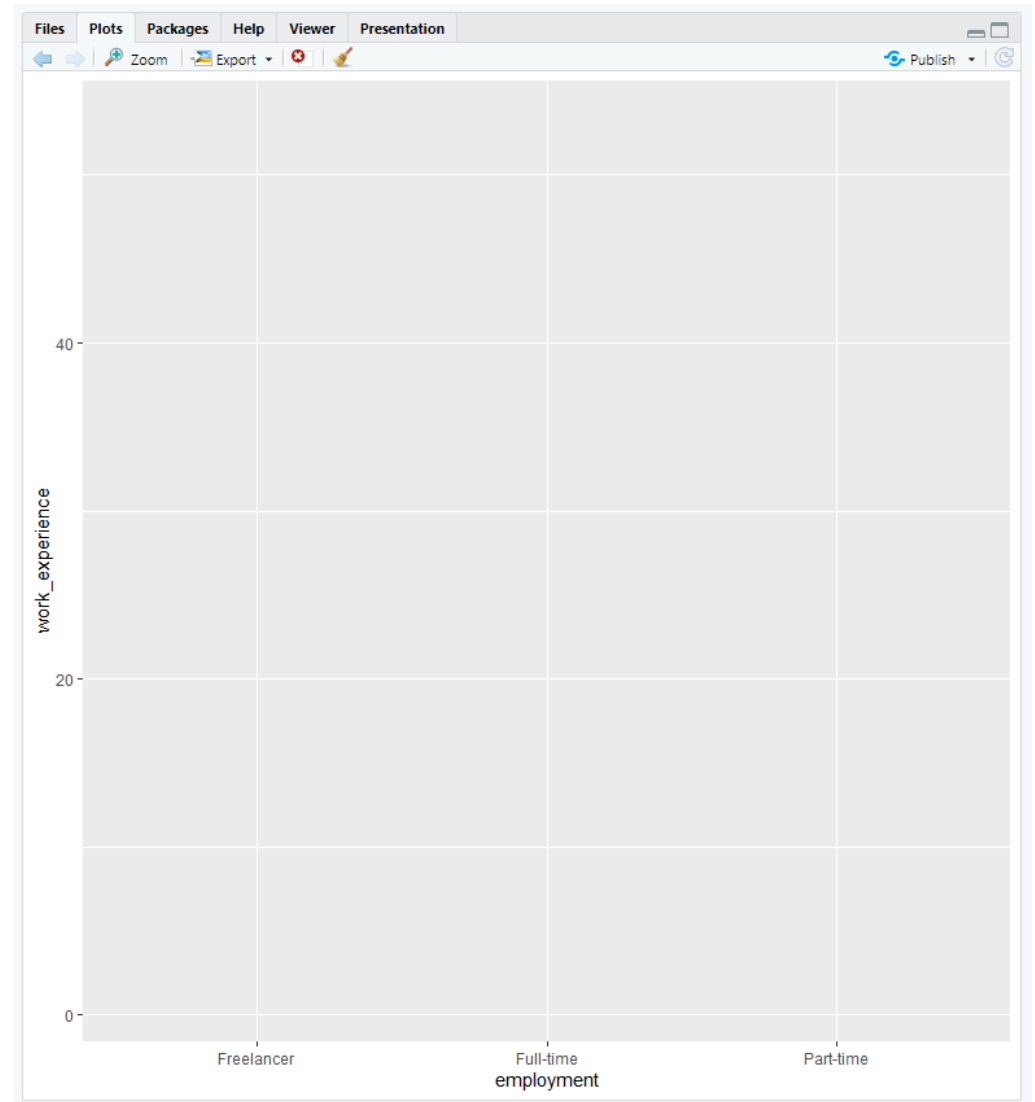


# Grammar of Graphics

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

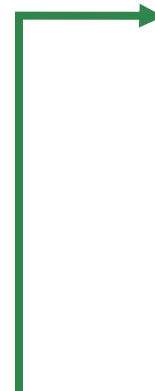
Dafür wird die Funktion `aes()` verwendet.

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



# Grammar of Graphics

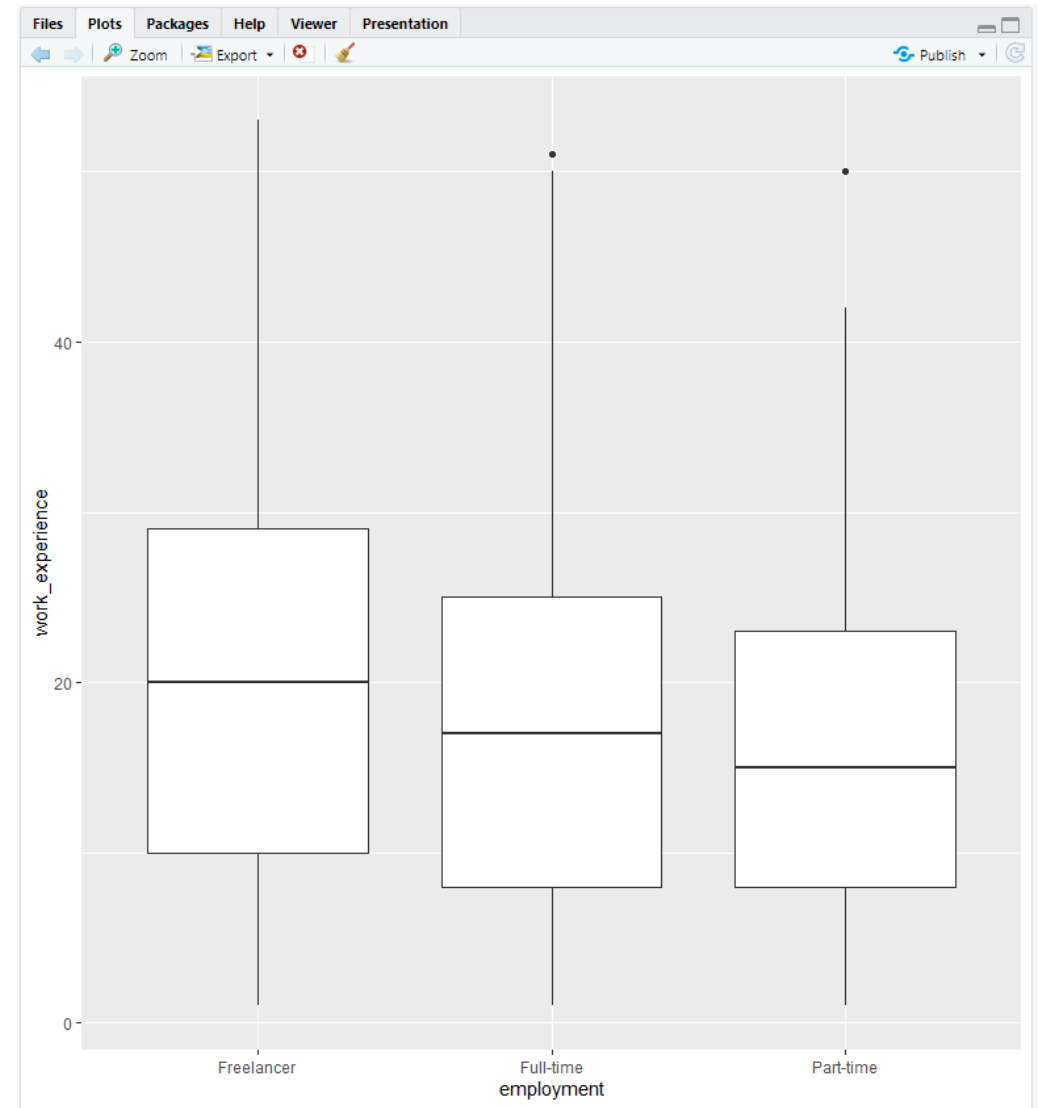
Danach muss noch das **geometrische Objekt** bestimmt werden.



```

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

```

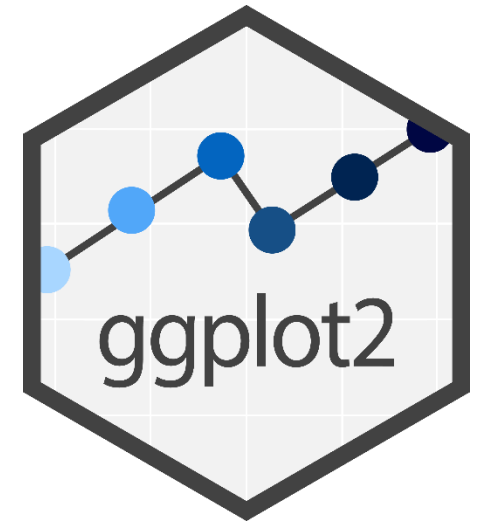


# Grammar of Graphics

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

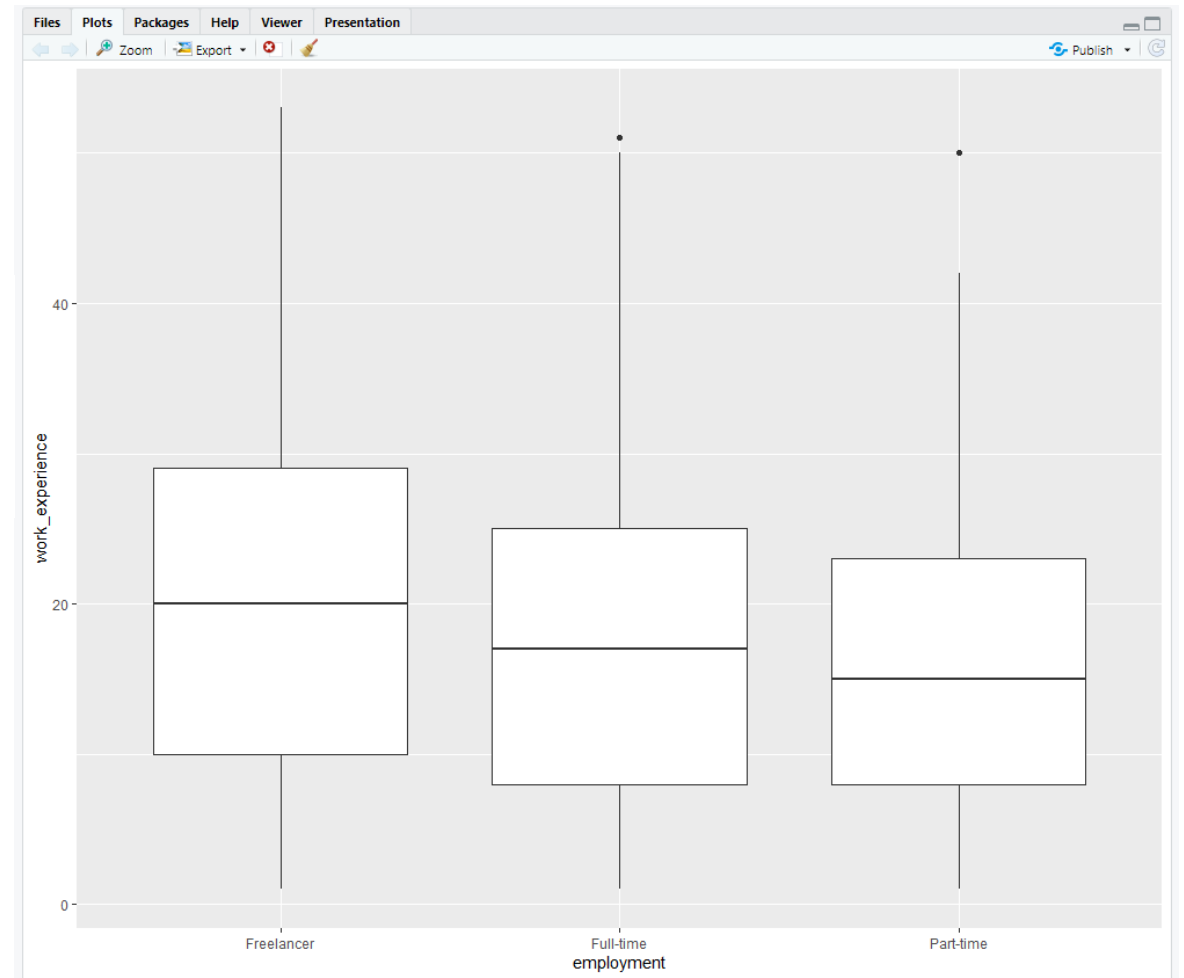


# Grammar of Graphics: geom\_boxplot()

```

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

```

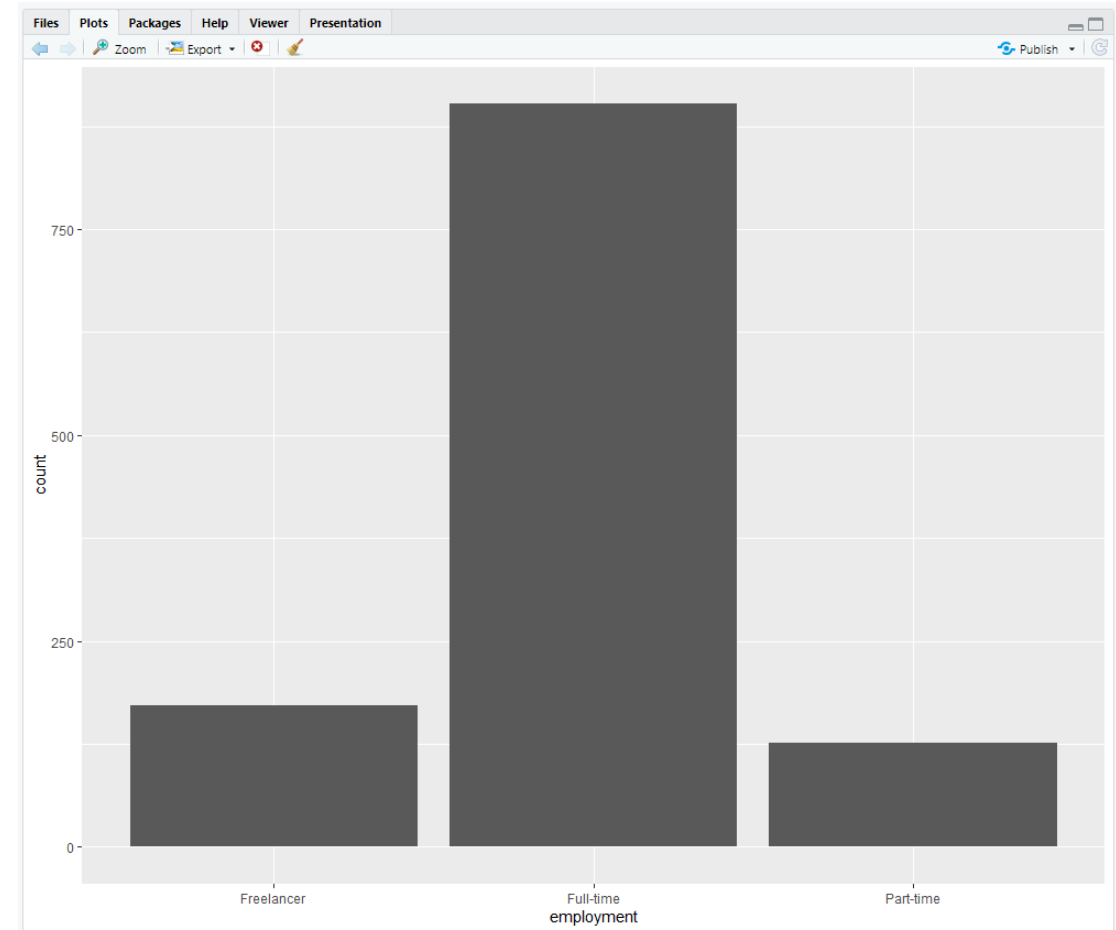


# Grammar of Graphics: geom\_bar()

```

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

```

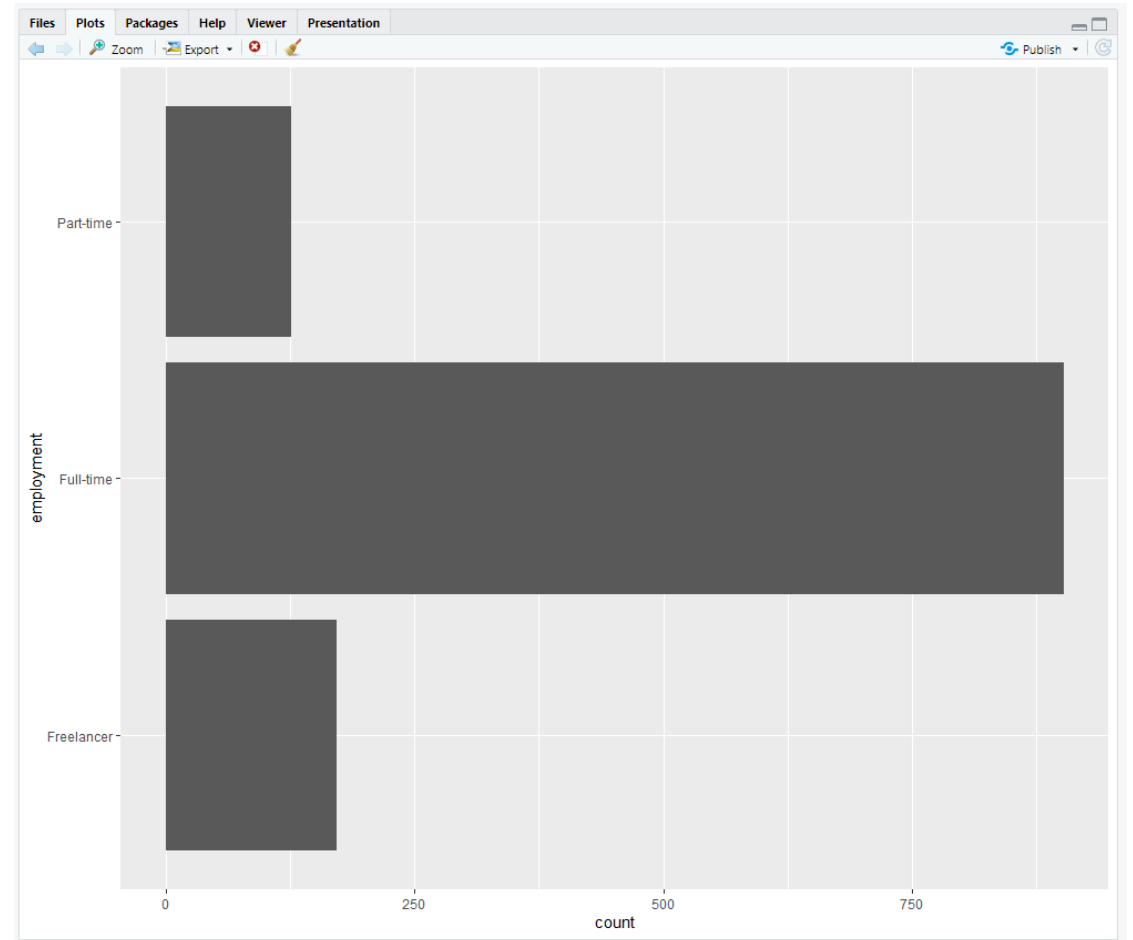


# Grammar of Graphics: geom\_bar()

```

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

```

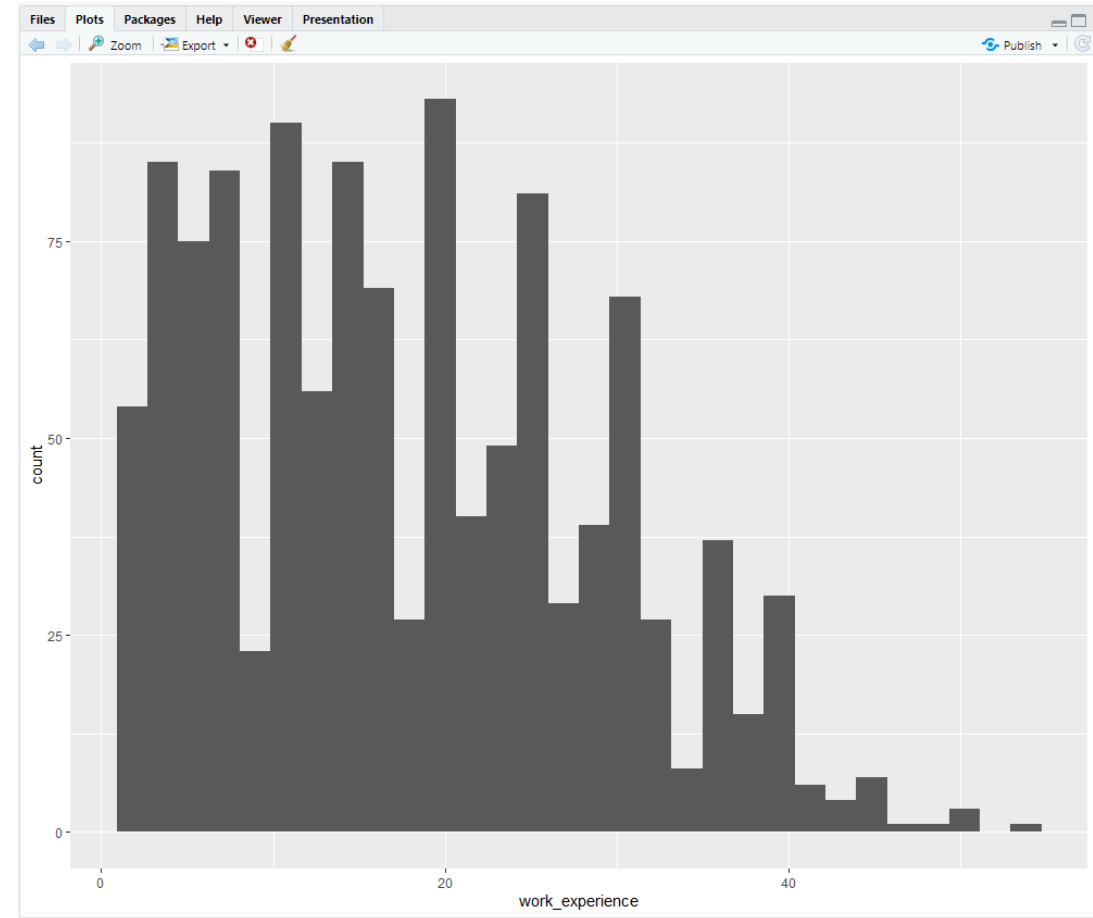


# Grammar of Graphics: geom\_histogram()

```

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

```

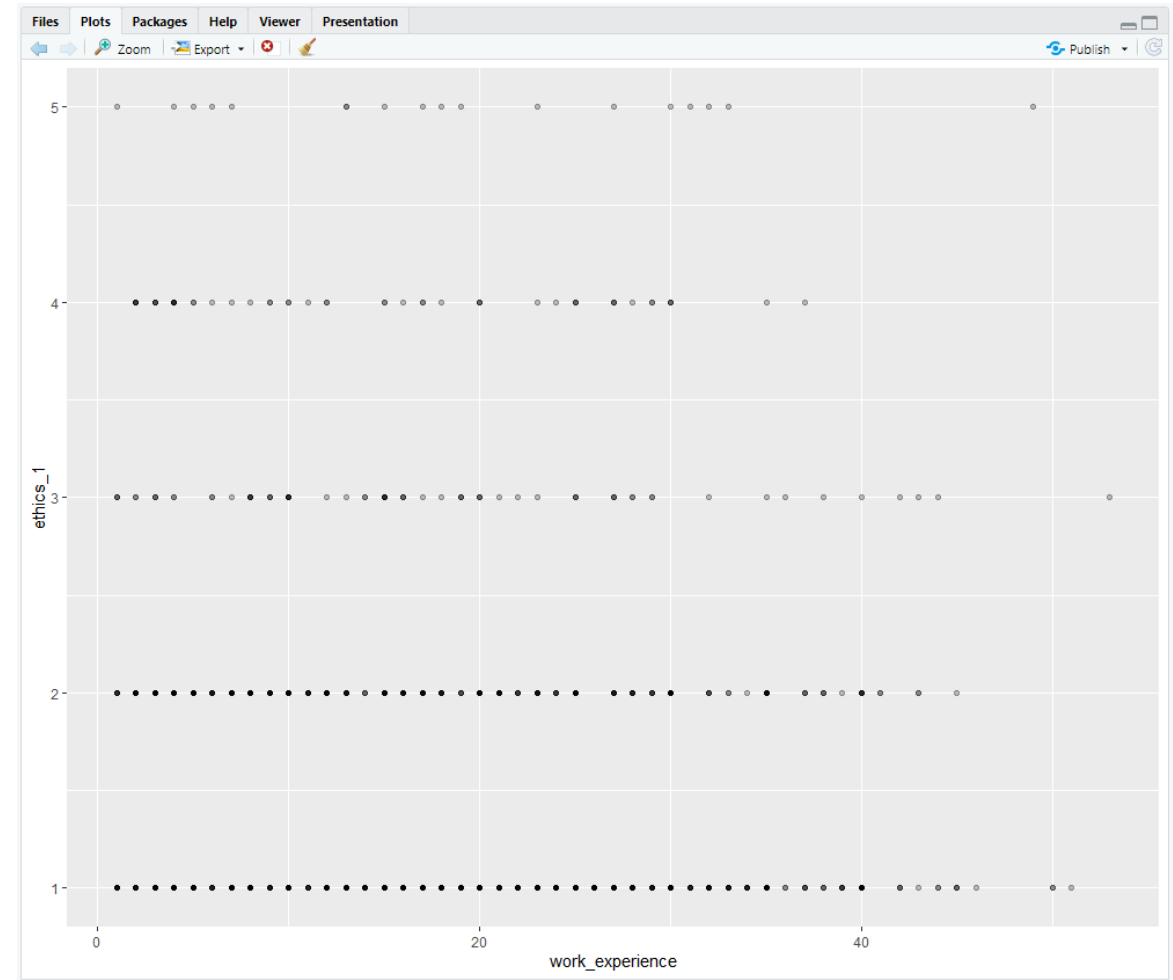


# Grammar of Graphics: geom\_point()

```

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")

```





# 4. 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...

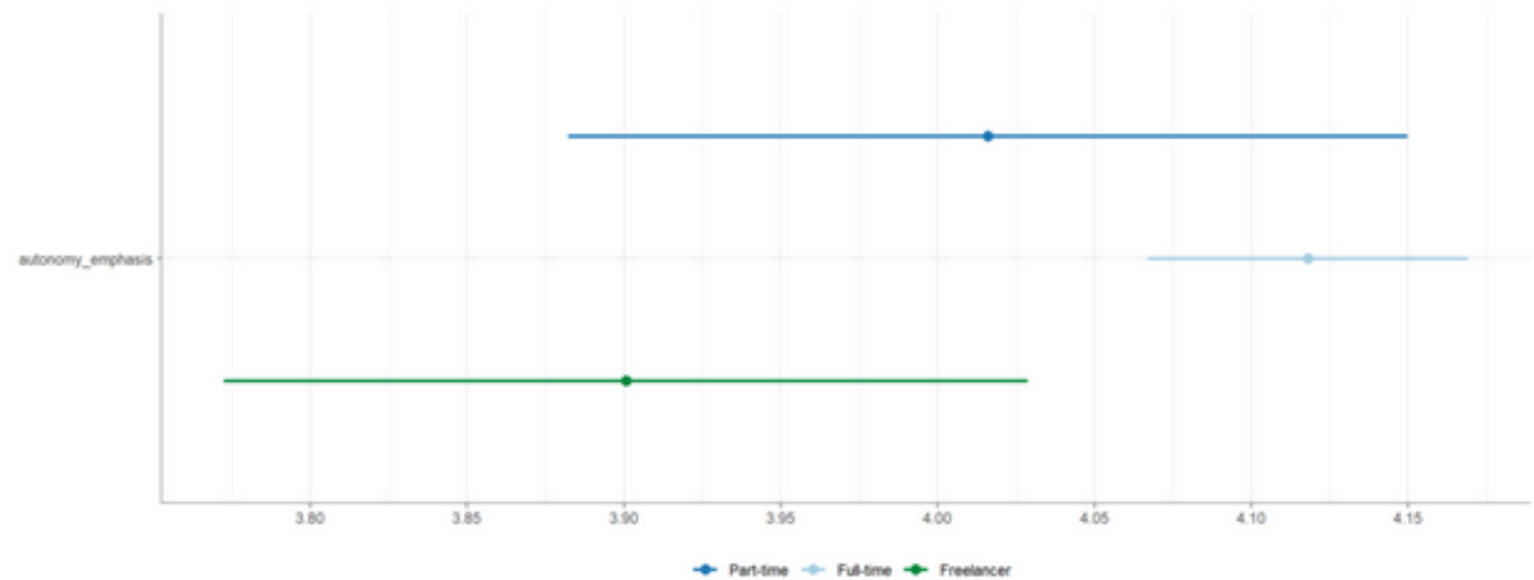


**Befehl:**

```

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

**Ausgabe:**

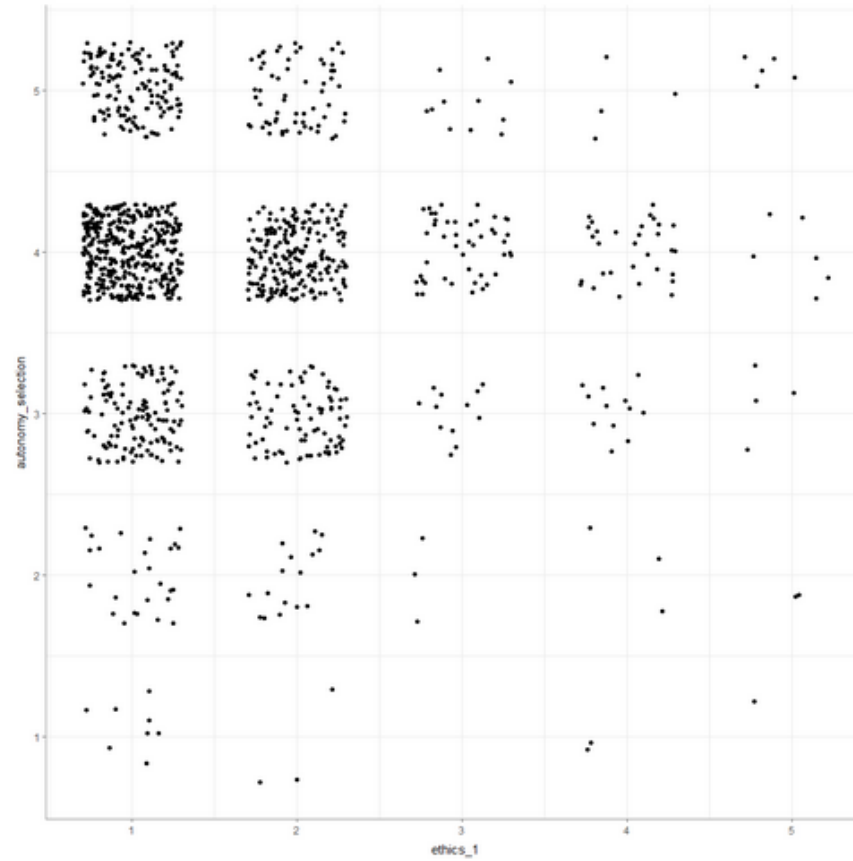


Visualisierung einer einfaktoriellen ANOVA in R

**Befehl:**

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

**Ausgabe:**



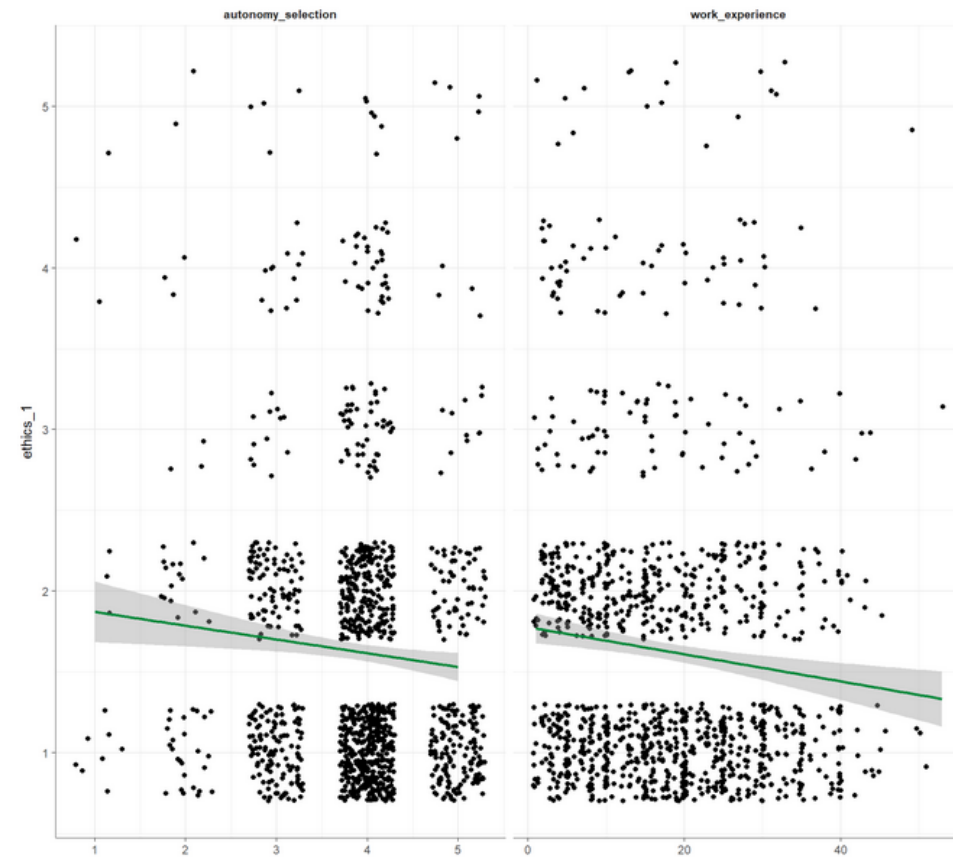
Visualisierung der Korrelation von zwei Variablen



**Befehl:**

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

**Ausgabe:**



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

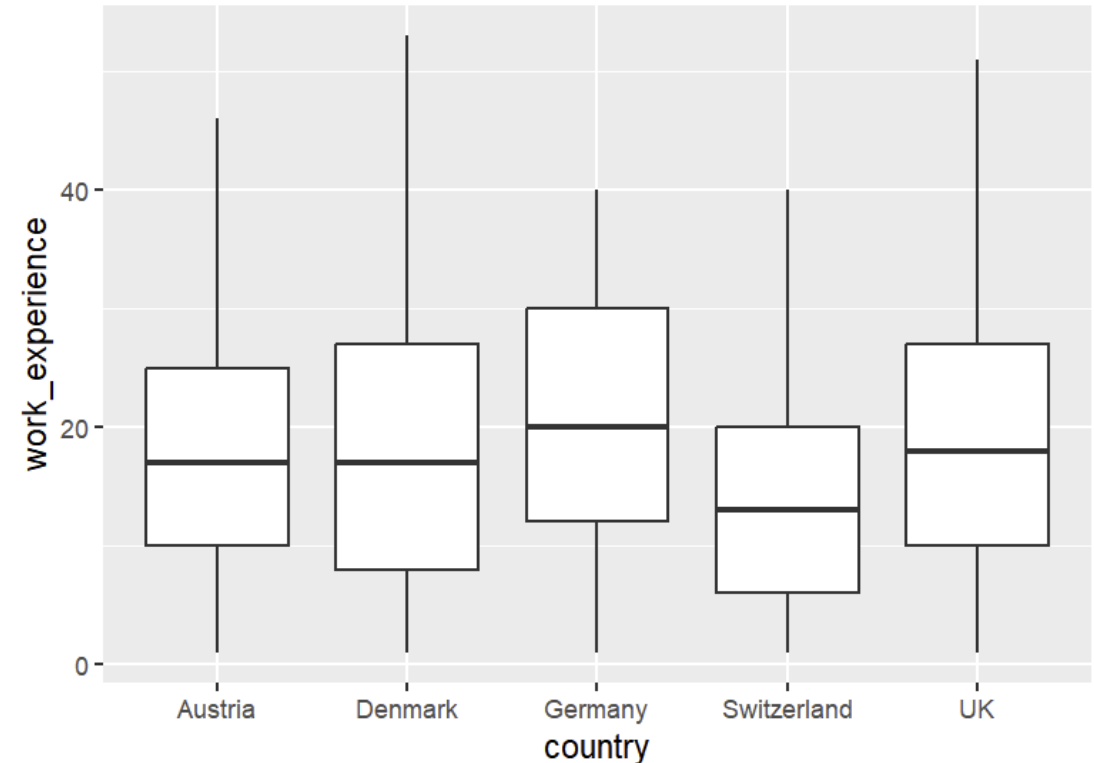




# 5. ÜBUNG 1: DATENVISUALISIERUNG

# 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()
```