

# Deskriptive Statistik

Es müssen immer alle Variablen beschrieben werden!

## Eine Variable

= STAT.DESKRIPTIV.NOMINAL(Variable)

## Visualisierung: Säulendiagramm

## Kontingenztafel mit zwei Variablen

= STAT.TABELLE(  
Variable1;  
Variable2;  
[MitBeschriftung]  
)

WAHR / FALSCH

## Visualisierung: Blasendiagramm

# Schliessende Statistik

## Korrelationen

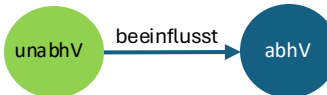
### Cramer's V

= KORREL.CRAMER(abhV; unabhV)

Cramer's V hat kein Vorzeichen!

Alle KORREL-Funktionen haben einen Korrelationskoeffizienten r als Ergebnis.

|r| < 0.25: Keine Korrelation  
|r| < 0.35: schwache Korrelation  
|r| < 0.5: Korrelation  
|r| ≥ 0.5: starke Korrelation



## Unterschiede

### X<sup>2</sup>-Unabhängigkeitstest

= CHISQ.TEST.S(  
abhV;  
unabhV;  
[MitBeschriftung]  
)

WAHR / FALSCH

= CHISQ.TEST.X(  
kT;  
[MitBeschriftung]  
)

Kontingenztafel ohne Beschriftung

WAHR / FALSCH

## z-TEST

z-Wert → p-Wert

NORM.VERT(zW; [mw]; [sd]; WAHR)

p-Wert → z-Wert

NORM.INV(pW; [mw]; [sd])

### Kendal's τ

= KORREL.KENDAL(abhV; unabhV)

### Werte für gerichtete Hypothesen:

"links", "rechts", "ungerichtet",  
"grösser", "kleiner", "ungleich"  
Ohne Angabe = "ungerichtet"

Die uV muss genau 2 Faktorstufen haben, sonst #WERT! Fehler!

= WILCOXON.TEST(  
abhV;  
unabhV;  
[gerichtet];  
[abhSP]  
)

WAHR / FALSCH

Für unabhängige Stichproben heisst der Wilcoxon-Test auch Mann U-Test.

Bei gerichteten Hypothesen wird der p-Wert automatisch angepasst.

Die uV hat mehr als 2 Faktorstufen

= KRUSKAL.WALLIS.TEST(  
abhV;  
unabhV;  
[MitBeschriftung]  
)

WAHR / FALSCH

Nur wenn der Kruskal Wallis Test ein signifikantes Ergebnis hat, dann

= DUNN.TEST(  
abhV;  
unabhV  
)

### Pearson's r

= KORREL.PEARSON(abhV; unabhV)

= KORREL(abhV; unabhV)

## Lineare Regression

$$aV = I + \beta_1 uV_1 + \beta_2 uV_2 + \dots + \beta_n uV_n$$

= REGRESSION(  
abhV;  
unabhV\_M  
)

Abhängige Variable  
Alle uV als Matrix

Ergebnis: 1. Zeile Intercept I, danach β für jede uV

= REGRESSION.RESIDUEN(  
abhV;  
unabhV\_M;  
unabhV\_Koeff  
)

Abhängige Variable  
Alle uV als Matrix

Ergebnis von REGRESSION()

Die Residuen müssen mit STAT.DESKRIPTIV() beschrieben werden!

Die uV muss genau 2 Faktorstufen haben, sonst #WERT! Fehler!

= T.TEST.S(  
abhV;  
unabhV;  
[gerichtet];  
[abhSP]  
)

WAHR / FALSCH

Bei gerichteten Hypothesen wird der p-Wert automatisch angepasst.

### Homogenitätstest

Die uV muss genau 2 Faktorstufen haben, sonst #WERT! Fehler!

= F.TEST.S(  
abhV;  
unabhV;  
[MitBeschriftung]  
)

WAHR / FALSCH

= ANOVA.TEST(  
abhV;  
unabhV;  
[MitBeschriftung]  
)

WAHR / FALSCH

Nur wenn die ANOVA ein signifikantes Ergebnis hat, dann

= PAARWEISER.T.TEST(  
abhV;  
unabhV  
)

Der Dunn Test und der paarweise t-Test können gelegentlich keine signifikanten paarweisen Unterschiede finden. In diesem Fall gilt die H<sub>0</sub>: Es gibt keinen Unterschied!

empxl

Nominal

Ordinal

Metrisch

dxi

# Ablauf deskriptive Statistik

- 1 Daten ggf. neu kodieren (s. Box)
- 2 Spalten aus der Datentabelle extrahieren

= TABELLEN.AUSWAHL(TABNAME; SPNAMEN)  
**Leere Zellen werden in #NV umgewandelt.**  
**Das Ergebnis ist immer eine Wertematrix**

## Beispiel

Alle Vektoren mit gleichem Skalenniveau mithilfe der Schemadefinition auswählen

```
= TABELLEN.AUSWAHL(  
  "Daten";  
  FILTER(Schema[Name];  
    Schema[Skalierung] = "ordinal")  
)
```

- 3 Deskriptive Statistik durchführen

## Beispiel

```
= STAT.DESKRIPTIV(  
  Wertematrix;  
  "ordinal";  
  FILTER(Schema[Name];  
    Schema[Skalierung] = "ordinal")  
)
```

## Daten neu kodieren

### Ordinalskalierte Daten in Zahlen kodieren

= SCHNELL.KODIEREN(WERTE; SORTIERUNG)

**Wandelt ordinalskalierte Werte in richtig sortierte Zahlenwerte um.**

## Beispiel

```
= SCHNELL.KODIEREN(  
  Daten[@oVariable];  
  {"a"; "b"; "d"; "c"})
```

### Nominalskalierte Daten in Zahlen kodieren

= DUMMY.CODING(Werte; [MitBeschriftung])

**Wandelt Nominalskalierte Variablen in eine binärkodierte Wertematrix um.**

**Die Beschriftung kann auch mit EINDEUTIG(Werte) erzeugt werden.**

# Ablauf schliessende Statistik

- 1 Spalten aus der Datentabelle extrahieren

## Beispiel

```
= TABELLEN.AUSWAHL("Daten";  
  {"abhV"; "unabhV"})  
  
= TABELLEN.AUSWAHL("Daten";  
  A1:B1)
```

- 2 Alle Zeilen in denen Fehlerwerte vorkommen entfernen

= ENTFERNE.FEHLER(Wertematrix)

**Nur für die schliessende Statistik verwenden!**

## Beispiel

```
= ENTFERNE.FEHLER(  
  TABELLEN.AUSWAHL("Daten";  
    {"abhV"; "unabhV"})  
)
```

- 3 Vektoren aus Wertematrix auswählen

= SPALTENWAHL(Wertematrix; SpaltenNr)

**Wählt die Spalte mit der SpaltenNr aus einer Wertematrix aus.**

## Beispiel

```
= SPALTENWAHL(A2#; 1)
```

- 3 Daten ggf. neu kodieren (s. Box)

- 4 Test für die schliessende Statistik durchführen (s. Rückseite)



Die aktuellste Version des empxl-Templates findet sich auf:

<https://github.com/phish108/empxl>

## Matrix-Funktionen

**Alle Funktionen beziehen sich auf Zahlenmatrizen.**

= SPALTENSUMME(Matrix)

= ZEILENSUMME(Matrix)

**Berechnen die Spalten-/Zeilensummen einer Matrix.**

= MEINHEIT(Länge)

**Erzeugt eine Einheits- bzw. Identitätsmatrix.**

= MDREIECK(Länge;  
 [Orientierung];  
 [Strikt])

**Erzeugt eine Dreiecksmatrix.**

**Ist Orientierung = WAHR, wird das untere Dreieck erzeugt.**

**Ist Strikt = WAHR, wird die Diagonale ausgeschlossen.**

= MDIMENSION(Matrix)

**Gibt die Dimensionen einer Matrix zurück.**

= MTRANS(Matrix)

**Transponiert eine beliebige Matrix oder Vektor.**

= MMULT(Matrix1; Matrix2)

**Berechnet das Kreuzprodukt aus zwei Matrizen.**

= MSPUR(Matrix)

**Berechnet die Spur einer Matrix.**

## Vektor-Funktionen

= VEKTOR.LÄNGE(Vektor)

**Bestimmt die Länge eines Vektors.**

= HWIEDERHOLEN(Wert; Anzahl)

= VWIEDERHOLEN(Wert; Anzahl)

**Erzeugt einen Vektor der Länge Anzahl mit dem angegebenen Wert**

**HWIEDERHOLEN() erzeugt einen Zeilenvektor.**

**VWIEDERHOLEN() erzeugt einen Spaltenvektor.**

## Beispiel

```
= HWIEDERHOLEN(2; 3)  
→ erzeugt {2;2;2}
```

```
= HWIEDERHOLEN("dxi"; 3)  
→ erzeugt {"dxi"; "dxi"; "dxi"}
```

empxl

dxi