Statistik 5

Übung 6

Einleitung und Ziele der Übung

Herzlich Willkommen zur Übung Statistik 5. In dieser Übung wollen wir einfache Poweranalysen mit G*Power und in R mit Package pwr lösen.

Agenda

- Installation/Vorbereitung
- Poweranalysen (Software-analytisch)
- Funktionen schreiben

1. Installation/Vorbereitung - Einführung

Bitte führen Sie folgende Schritte durch:

1. Installieren Sie das Package pwr in R und laden Sie es.
2. Installieren Sie G*Power über folgenden Link Website G*Power

2. Poweranalysen - Einführung

Auf den nächsten Seiten finden Sie der Reihe nach Aufgaben zur Poweranalyse mit G*und R. Diese sind (grob) der Schwierigkeit nach geordnet. Bitte versuchen Sie jede Aufgabe mit beiden Möglichkeiten zu lösen.

2. Poweranalysen - Übung 1 (sehr leicht)

Bestimmen Sie die Stichprobengrösse für einen Einstichproben *t*-Test:

- ungerichtet
- $\alpha = .01$
- $\beta = .10$
- $d = .25$
- $n =$

2. Poweranalysen - Übung 2 (leicht)

Bestimmen Sie die notwendige Effektgrösse (3 NKS, Werte zwischen 0 und 1) für eine einfaktorielle Varianzanalyse (ohne Messwiederholung).

- $\alpha = .05$
- $Faktorstufen = 7$
- $Teststaerke = .85$
- $n_{Gesamt} = 280$
- $f =$

2. Poweranalysen - Übung 3 (mittel)

Eine Geschäftsfrau möchte einen neuen Imbiss im Quartier eröffnen. Um zu erfahren, welches Fastfood am besten funktioniert, befragt Sie 100 Personen, nach Ihrer Präferenz (Anzahl Antworten in Klammern):

- Pizza (33)
- Döner (20)
- Burger (15)
- Schnitzel mit Pommes (10)
- Salat (22)

Ihre ursprüngliche Hypothese war, dass alles gleich beliebt ist. Wie gross war Ihr Signifikanzniveau, wenn Sie eine Teststärke von .70 erzielt haben? Um die Effektstärke ω (0-1, 3NKS) zu ermitteln nutzen Sie bitte beide der möglichen Wege: G*Power und die folgende Formel:

- $\omega = \sqrt{\sum \frac{(p_{beob} - p_{erw})^2}{p_{erw}}} =$
- $df = k - 1 =$
- $\alpha =$

2. Poweranalysen - Übung 4 (mittel)

Bestimmen Sie die Teststärke (3 NKS, Werte zwischen 0 und 1) für einen Interaktionsterm einer reinen Messwiederholungs-Anova, diesmal nur mit G*Power. Gehen Sie von perfekter Sphärizität aus

- $\alpha = .05$
- $n = 50$
- $f = .10$
- $Design = 2x5$
- $\bar{\bar{r}} = .75$
- $Teststaerke =$

3. Funktionen schreiben - Einführung

Jetzt wollen wir noch eine Übungsweise eine Gruppe von Funktionen schreiben. Das Ziel ist Unser Beispiel 3 aus der Vorlesung zum Bootstrap als (mehrere Funktionen) umzusetzen.

3. Funktionen schreiben - Übung 1 (leicht)

Schreiben Sie eine Funktion *tStat_AS* mit den zwei Argumenten *T1* und *T2*. Die Funktion soll einen *t*-Wert für einen Mittelwertsvergleich zweier abhängiger Stichproben ermitteln.

- Orientieren Sie sich am Codebeispiel der Vorlesung zum Bootstrap
- *T1* soll die Werte der ersten Stichprobe übergeben bekommen (also ein Vektor)
- *T2* soll die Werte der zweiten Stichprobe übergeben bekommen (also ebenfalls ein Vektor)
- Nutzen Sie length() um die Stichprobengrösse zu ermitteln
- Nutzen Sie zusätzlich die Befehle mean(), sd() und sqrt() aber nicht t.test()
- Sie dürfen davon ausgehen, dass keine fehlenden Werte vorhanden sind
- Der Rückgabewert soll der Wert der t-Statistik sein
- Testen Sie Ihre Funktion mit folgenden Werten :

```r
Gruppe1 <- c(15,14,13,17)
Gruppe2 <- c(10,8,9,15)
# Ergebnis sollte 4.97709 sein
```

3. Funktionen schreiben - Übung 2 (leicht-mittel)

Erweitern Sie Ihre Funktion tStat_AS mit einem dritten Argument *Boot*:

- *Boot* erhält entweder den Wert “ja” oder “nein”
- bei “nein” funktioniert eigentlich alles wie gehabt
- bei “ja”
  - Bestimmen Sie zunächst die Personen, die in der Bootstrap-Stichprobe enthalten sein sollen (N Mal Ziehen mit Zurücklegen aus 1:N)
  - Denken Sie daran: Sie müssen die Personen der Bootstrap-Stichprobe von T1 und T2 nur 1x bestimmen, es sind die gleichen Personen in der gleichen Reihenfolge für beide Zeitpunkte
  - Ersetzen Sie die Werte von T1 und T2 mit den Werten von T1 und T2 für die ermittelten Personen
  - Danach so weiter wie immer (Sie brauchen also am restlichen Code nichts ändern!)
- Testen Sie Ihre Funktion mit folgenden Werten (Vergessen Sie nicht den Code vorher nochmals auszuführen:

```r
Gruppe1 <- c(15,14,13,17)
Gruppe2 <- c(10,8,9,15)
set.seed(123)
#1x mit "nein" sollte 4.97709 sein
#2x mit "ja" sollte das 1. Mal 9 sein und das zweite Mal 11 ergeben
```

3. Funktionen schreiben - Übung 3 (leicht)

Schreiben Sie eine neue Funktion tStat_AS_Boot mit den Argumenten *T1*,*T2* und *Bootstraps*

- *T1* und *T2* sind genauso wie zuvor
- *Bootstraps* enthält die Anzahl an Bootstraps
- Die Funktion soll der Reihe nach folgendes tun:
  - Die vorherige Funktion aufrufen um den originalen t-Wert zu bestimmen und in t_Orig zu speichern
  - replicate() mit folgenden Argumenten ausführen und das Ergebnis in t_Verteilung speichern:
    - Bootstraps
    - unsere vorherige Funktion mit Boot = “ja”
  - t_Verteilung zurückgeben
- Testen Sie Ihre Funktion mit folgenden Werten:

```
Gruppe1 <- c(15,14,13,17)
Gruppe2 <- c(10,8,9,15)
Wiederholungen <- 5000
set.seed(234)
# erste 4 Werte sollten 10.966892  5.018570  4.333333  8.660254 sein
```

3. Funktionen schreiben - Übung 4 (leicht)

Bevor wir die Signifikanzprüfung in unsere Funktion einbauen, wollen wir uns die Signifikanzprüfung per Konfidenzintervall “analog” überlegen.

- Bei einer ungerichteten Hypothese für Mittelwertsunterschiede (Bootstrap) muss
  - welche Grenze bestimmt werden? [ ]
  - das Signifikanzniveau [ ]
  - ist das Ergebnis signifikant, wenn 0 [ ] liegt
- Bei einer gerichten Hypothese (T1 kleinere Werte) für Mittelwertsunterschiede (Bootstrap) muss
  - welche Grenze bestimmt werden? [ ]
  - das Signifikanzniveau [ ]
  - ist das Ergebnis signifikant, wenn 0 [ ] liegt
- Bei einer ungerichteten Hypothese (T1 grössere Werte) für Mittelwertsunterschiede (Bootstrap) muss
  - welche Grenze bestimmt werden? [ ]
  - das Signifikanzniveau [ ]
  - ist das Ergebnis signifikant, wenn 0 [ ] liegt

4. Funktionen schreiben - Übung 5 (mittel)

Erweitern Sie die Funktion tStat_AS_Boot um zwei weitere Argumente *Gerichtet* und *Alpha*

- *Gerichtet* erhält enweder den Wert “u” (ungerichtet), “k” (kleiner) oder “g” (grösser)
- *Alpha* sei unser Signifikanzniveau (Wert zwischen 0 und 1)
- Passen Sie Ihren Code zusätzlich wie folgt an:
  - Entfernen Sie die Rückgabe der t_Verteilung
  - Bestimmen Sie, ob Ihre t-Statistik signifikant ist:
    - Wenn Gerichtet = “u” ist, gibt es zwei Grenzen und 0 muss kleiner als die untere Grenze oder grösser als die obere Grenze sein
    - Wenn Gerichtet = “k” ist, gibt es nur eine obere Grenze
    - Wenn Gerichtet = “g” ist, gibt es nur eine untere Grenze
    - Die Grenze(n) können Sie mit quantile(), Ihrer *t_Verteilung*-Variable und *Alpha* bestimmen
    - Verwenden Sie bei quantile() als drittes Argument bitte jeweils **na.rm = TRUE**
    - Geben Sie FALSE für nicht signifikant oder TRUE für signifikant zurück (ohne Anführungszeichen).
- Die Testwerte finden Sie auf der nächsten Seite

4. Funktionen schreiben - Übung 6

Fasst geschafft, testen Sie Ihre Funktion mit folgenden Werten (andere Werte!):

```r
G1 <- c(14,12,11,13)
G2 <- c(12,11,12,13)
Wiederholungen <- 5000
Signifikanzniveau = .3
set.seed(234)
# 3 Tests der Reihe nach
# ungerichtet (nicht signifikant --> FALSE)
# gerichtet kleiner (nicht signifikant --> FALSE)
# gerichtet grösser (signifikant --> TRUE)
```