Applied Data Analysis, Übung 3

Henry Haustein

Vorbereitung

```r
install.packages("dplyr")
library(dplyr)
install.packages("readxl")
library(readxl)
data = read_excel("data.xlsx", na = "NA")

data$gender = factor(data$gender)
levels(data$gender) = c("male", "female", "diverse")
data$employment = factor(data$employment)
levels(data$employment) = c("student", "employed", "unemployed")
data$education = factor(data$education)
levels(data$education) = c("no degree", "secondary", "intermediate", "high school", "academic")
data$play_frequency = factor(data$play_frequency)
levels(data$play_frequency) = c("never", "every few months", "every few weeks", "1-2 days a week", "3-5 days a week", "daily")
data$treatment = factor(data$treatment)
levels(data$treatment) = c("control", "lootbox in task reward", "lootbox picture", "badge")
data$age = sapply(data$age, function(year) {2016-year})
data$rt6 = as.numeric(data$rt6)
data$rt7 = as.numeric(data$rt7)
data$rt8 = as.numeric(data$rt8)
data$rt9 = as.numeric(data$rt9)
data$rt10 = as.numeric(data$rt10)
data$rt11 = as.numeric(data$rt11)
data$rt12 = as.numeric(data$rt12)
data$rt13 = as.numeric(data$rt13)
data$rt14 = as.numeric(data$rt14)

subsetControl = data %>% filter(treatment == "control")
subsetLootPic = data %>% filter(treatment == "lootbox picture")
subsetLootInTask = data %>% filter(treatment == "lootbox in task reward")
subsetBadge = data %>% filter(treatment == "badge")
subsetYoung = data %>% filter(old == FALSE)
subsetOld = data %>% filter(old == TRUE)
```

Task 1

```r
t.test(data$`total time`, mu = 320000, alternative = "two.sided", conf.level = 0.95)
t.test(subsetYoung$`total time`, mu = 320000, alternative = "two.sided", conf.level = 0.95)     t.test(subsetOld$`total time`, mu = 320000, alternative = "two.sided", conf.level = 0.95)

t.test(subsetYoung$tasks_completed, mu = mean(subsetOld$tasks_completed), alternative = "two.sided", conf.level = 0.95)

pairwise.t.test(data$tasks_completed, data$treatment, p.adjust.method = "none", pool.sd = FALSE)
```

Bei allen Personen und den jungen Personen ist der p-value unter 5%, das heißt wir können H_0 nicht ablehnen. Bei den alten Personen ist der p-value bei 50%, wir lehnen H_0 ab.

Bei allen andern Tests sind die p-values unter 5%, auch da lehnen wir H_0 ab. Die Mittelwerte sind also verschieden voneinander.

Task 2

```r
install.packages("ggplot2")
library(ggplot2)

ggplot(data, aes(x = `total time`))
+ geom_histogram(aes(y = ..density..))
+ stat_function(fun = dnorm, args = list(mean = mean(data$`total time`), sd = sd(data$`total time`)), color = "red")
ggplot(data, aes(x = log(`total time`)))
+ geom_histogram(aes(y = ..density..))
+ stat_function(fun = dnorm, args = list(mean = mean(log(data$`total time`)), sd = sd(log(data$`total time`))), color = "red")
ggplot(data, aes(sample = `total time`))
+ geom_qq()
+ geom_qq_line()
ggplot(data, aes(sample = log(`total time`)))
+ geom_qq()
+ geom_qq_line()

ggplot(data, aes(x = tasks_completed))
+ geom_histogram(aes(y = ..density..))
+ stat_function(fun = dnorm, args = list(mean = mean(data$tasks_completed), sd = sd(data$tasks_completed)), color = "red")
ggplot(data, aes(x = log(tasks_completed)))
+ geom_histogram(aes(y = ..density..))
+ stat_function(fun = dnorm, args = list(mean = mean(log(data$tasks_completed)), sd = sd(log(data$tasks_completed))), color = "red")
ggplot(data, aes(sample = tasks_completed))
```

```
+ geom_qq()
+ geom_qq_line()
ggplot(data, aes(sample = log(tasks_completed)))
+ geom_qq()
+ geom_qq_line()
```

Man sieht, dass die Daten auf keinen Fall normalverteilt sind, höchstens die Variable `total time` ist etwas log-normalverteilt.

Es gibt einen Test auf Normalverteilung, den Shapiro-Wilk-Test. Nullhypothese dabei ist, dass die Daten normalverteilt sind.

```
shapiro.test(data$`total time`)
shapiro.test(data$tasks_completed)
shapiro.test(log(data$`total time`))
shapiro.test(log(data$tasks_completed))
```

Liefert uns p-values von $< 2.2 \cdot 10^{-16}$ (bzw. $4.767 \cdot 10^{-15}$ beim Logarithmus von `total time`), die Nullhypothese wird also angelehnt. Normalverteilt sind die Daten nicht.

Task 3

Der Kern des Skriptes ist der folgende Ablauf, der 10 000 mal wiederholt wird:

1. Wähle aus dem gegebenen Datensatz 90 Beobachtungen aus.
2. Bilde den Mittelwert von `tasks_completed` aus den ausgewählten Beobachtungen.
3. Speichere den Mittelwert ab.

Das Central Limit Theorem (Zentraler Grenzwertsatz ↗ *Statistik 2*) sagt nun, dass diese 10 000 Mittelwerte normalverteilt sind, was dann in dem Skript mit einigen Abbildungen gezeigt wird.

Der Prozess des mehrfachen Auswählens aus vorhandenen Daten heißt *Bootstrapping* und ist mit Vorsicht zu genießen. Man erweckt damit den Eindruck als hätte man 10 000 Personen untersucht, aber in Wahrheit kommen die Daten nur von 99 Personen (für das Bootstrapping der Kontrollgruppe) bzw. 114 Personen (für das Bootstrapping der Lootbox-Bild-Gruppe). Wenn also die Datengrundlage zu klein ist, kann man mittels Bootstrapping es so aussehen lassen, als sei die Datengrundlage riesig.

Anschließend wird versucht einen t-Test durchzuführen, indem man die 10 000 Mittelwerte aus der Kontrollgruppe mit den 10 000 Mittelwerten aus der Lootbox-Bild-Gruppe vergleicht. Das 95%-Konfidenzintervall für den Unterschied wird mittels Quantilen bestimmt und ergibt $[-2.1, 0.1]$. Führt man den richtigen t-Test durch, so erhält man:

```
t.test(subControl$tasks_completed, subLootPic$tasks_completed,
    paired = FALSE)
```

als Konfidenzintervall $[-2.0282492, -0.0100283]$.

Task 4

```
ggplot(data, aes(x = age, y = `total time`))
+ geom_point()
+ geom_smooth(method = "lm")
```

```
summary(lm(`total time` ~ age, data = data))
ggplot(subsetControl, aes(x = age, y = `total time`))
+ geom_point()
+ geom_smooth(method = "lm")
summary(lm(`total time` ~ age, data = subsetControl))

ggplot(data, aes(x = tasks_completed, y = `total time`))
+ geom_point()
+ geom_smooth(method = "lm")
summary(lm(`total time` ~ tasks_completed, data = data))
ggplot(subsetControl, aes(x = tasks_completed, y = `total time`))
+ geom_point()
+ geom_smooth(method = "lm")
summary(lm(`total time` ~ tasks_completed, data = subsetControl))

ggplot(data, aes(x = rt0, y = `total time`))
+ geom_point()
+ geom_smooth(method = "lm")
summary(lm(`total time` ~ rt0, data = data))
ggplot(subsetControl, aes(x = rt0, y = `total time`))
+ geom_point()
+ geom_smooth(method = "lm")
summary(lm(`total time` ~ rt0, data = subsetControl))
```

Der Residual Standard Error ist schwer von der Größe her zu interpretieren, aber der R^2 sollte nahe bei 1 liegen. Nur für das Modell `total time` $\sim$ `rt0` ist das so, dort liegt er bei 0.9139 bzw. 0.9906 für die Kontrollgruppe. Für die anderen beiden linearen Modelle verbessert sich durch die Einschränkung auf die Kontrollgruppe nichts.