

# Übung 02

## Import und erste Auswertung eines Datensatzes **R**

### INFI-IS

### 5xHWII

Albert Greinöcker  
Thomas Kefer

March 29, 2022



CCA – COMPETENCE CENTRE

**HTL Anichstraße**

## 1 Importieren Aufbereiten des Datensatzes **Wintertourismus**

### 1.1 Import und Kontrolle

- Es soll der Datensatz *Zeitreihe Wintertourismus 2000 2019* aus Moodle in R importiert werden. In diesem Fall ist die Beschriftung der Spalten schon vorhanden, allerdings wurde der Beschriftung der Jahre immer ein "X" vorne angestellt. Bitte die Namen so ändern dass daraus hervorgeht dass es sich um Jahre handelt.
- Wenn man als ersten Schritt die importierten Daten grundsätzlich kontrollieren möchte, bietet sich der Befehl (*summary*) an. Was ist aus diesem ersichtlich bzw. warum ist er für die Kontrolle wichtig?

```
1 setwd("/home/baumbart13/Desktop/Mitschrift_Lokal/Schule/GITHUB-ONLINE/Mitschrift/5/INFI/IS/Aufgabe02/")
2 d <- read.csv("../datasets/Zeitreihe_Winter_2000_2019.csv", sep=";", encoding="UTF-8")
3
4
5 sz <- d[d$Bez == "SZ",4:23]
6 new_names <- paste("Jahr ", 2000:2019, sep="")
7 names(d)[4:23] <- new_names
8 summary(d)
```

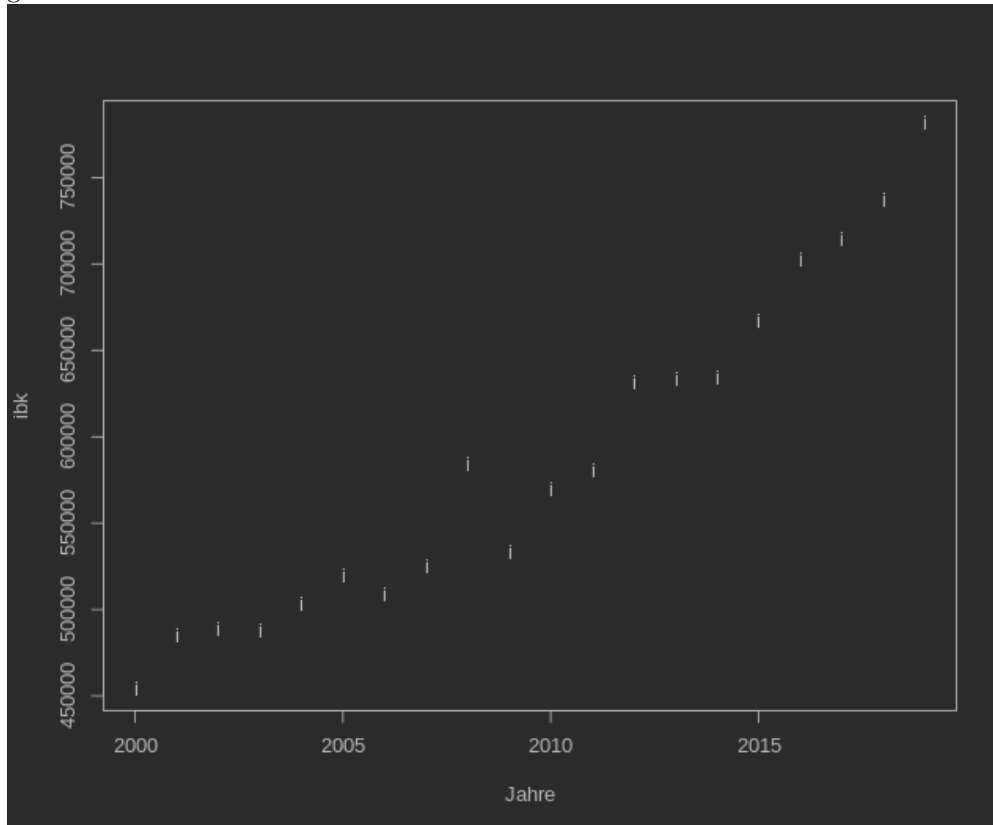
*Ergebnis:*

```
|=====|
|  Jahr 2019  |
| Min.   :    0 |
| 1st Qu.: 7102 |
| Median : 22202 |
| Mean   : 98518 |
| 3rd Qu.: 91615 |
| Max.   :2067037 |
|=====|
```

## 2 Erste Auswertung

### 2.1 Wachstum darstellen

Hole die Zahlenwerte zu den einzelnen Jahren in Innsbruck und stelle den zeitlichen Verlauf als Punktdiagramm dar. Als Zeichen soll ein `i` verwendet werden.



```
1 ibk <- as.numeric(d[d$Bez == "I",4:23])  
2 plot(2000:2019,ibk,pch="i", xlab="Jahre")
```

### 2.2 Wachstum des eigenen Bezirks (Falls Innsbruck, bitte einen anderen auswählen)

- Dazu ist zuerst die Auswahl der Gemeinden im Bezirk notwendig
- Diese Werte müssen aufsummiert werden (Hinweis: Verwende den Befehl `apply`)
- Bitte wieder einerseits die Zahlenwerte in der Konsole als auch ein Liniendiagramm ausgeben

Hier kommt das Liniendiagramm hin:

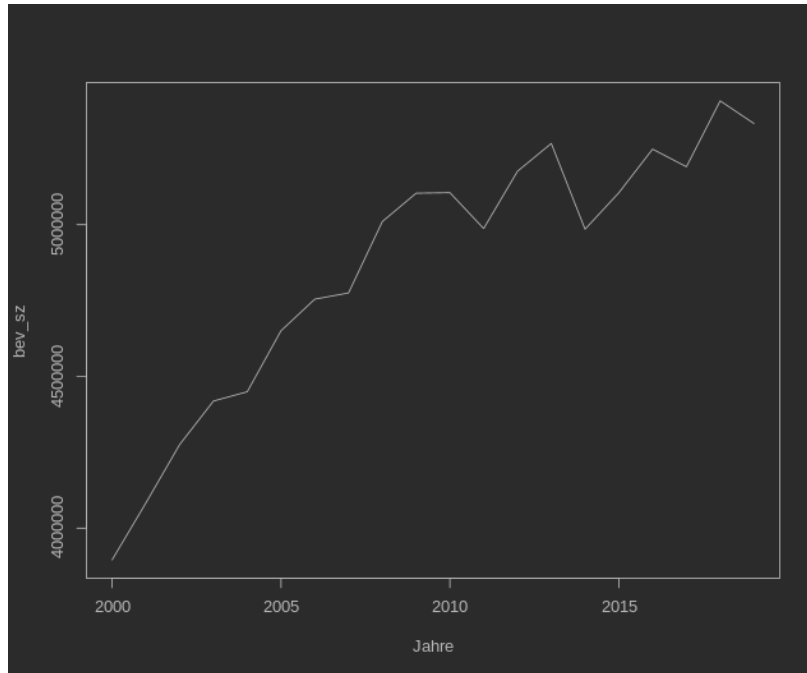


Figure 1: Figure caption.

```
1 bev_sz <- apply(d[d$Bez=="SZ",4:23],2, FUN =sum)
2 plot(2000:2019,bev_sz,xlab="Jahre",type="l")
```

*Ergebnis:*

```
|=====|=====|=====|=====|=====|
|Jahr 2000|Jahr 2001|Jahr 2002|Jahr 2003|Jahr 2004|
| 3895490| 4083001| 4275421| 4418925| 4449479|
|=====|=====|=====|=====|=====|
|Jahr 2005|Jahr 2006|Jahr 2007|Jahr 2008|Jahr 2009|
| 4650490| 4754511| 4775033| 5010371| 5103519|
|=====|=====|=====|=====|=====|
|Jahr 2010|Jahr 2011|Jahr 2012|Jahr 2013|Jahr 2014|
| 5105361| 4987087| 5175933| 5267590| 4984930|
|=====|=====|=====|=====|=====|
|Jahr 2015|Jahr 2016|Jahr 2017|Jahr 2018|Jahr 2019|
| 5104987| 5248776| 5190775| 5407888| 5333006|
|=====|=====|=====|=====|=====|
```

## 3 Berechnen von Werten

### 3.1 Min, Max, Range, Avg

Zu den einzelnen Gemeinden sollen das Minimum, Maximum, Range (also Maximum - Minimum) und der Durchschnitt berechnet werden. Diese sollen in einer eigenen Spalten-Variable abgelegt werden.

```
1 d$durchschnitt <- apply(d[,4:23],1,FUN=mean)
2 d$minimum <- apply(d[,4:23],1,FUN=min)
```

```
3 d$maximum <- apply(d[,4:23],1,FUN=max)
4 d$range <- d$maximum - d$minimum
```

## 3.2 Gesamtzahl an Touristen

- Die Gesamtzahl an Touristen pro Jahr soll berechnet werden (Befehl: `apply`)
- Diese Werte sollen weiterverarbeitet werden, so dass man die Gesamtzahl über alle Jahre bekommt (also 1 Wert). Hinweis: Es gibt eine Funktion `sum`.
- Wie kann die Zusammenfassung nach Bezirken gemacht werden?

```
1 bev_sum_bezirk <- aggregate(d[,4:23], by=list(d$Bez), FUN =sum)
2 bev_pro_jahr <- apply(bev_sum_bezirk[2:21],2, FUN =sum)
3 bev_alle_jahre <- sum(bev_pro_jahr)
```

### 3.2.1 Standardisierung

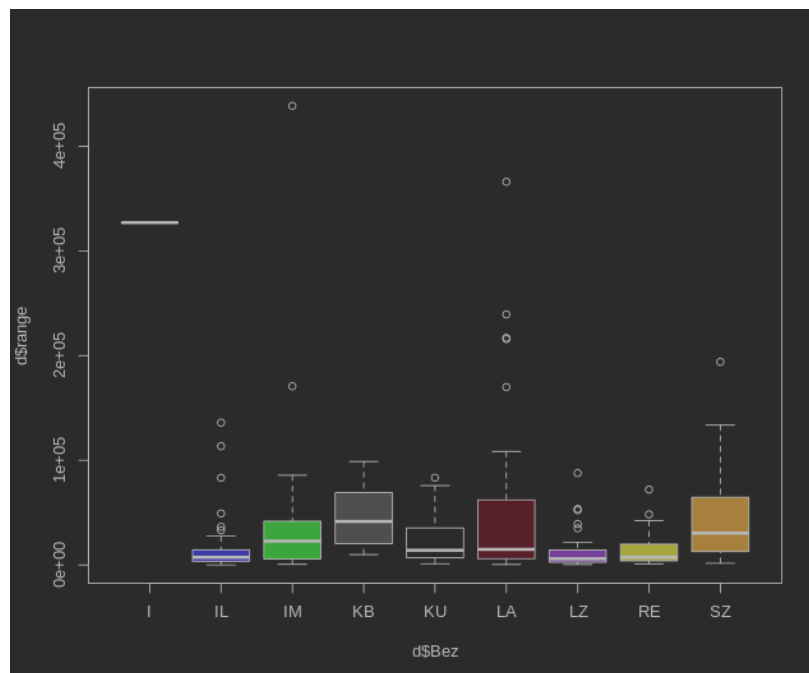
Der Range ist jetzt absolut. Wie könnte man diesen standardisieren?

```
1 numbers <- d[4:23]
2 rangestnd <- numbers / d$range
```

## 4 Gegenüberstellung von Bezirken

### 4.1 Boxplots

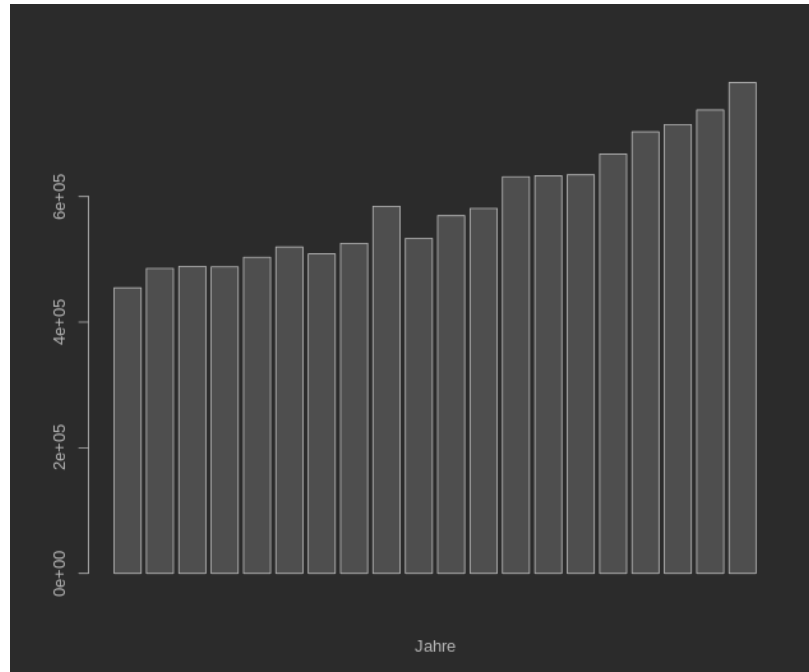
Stelle die (standardisierten) Ranges der einzelnen Bezirke als Boxplot gegenüber. Jeder Bezirk soll eine eigene Farbe haben



```
1 boxplot(d$range~d$Bez, col = c("red","blue","green","grey",
2   "white","pink","purple","yellow","orange"))
```

## 4.2 Barplot

Stelle die die Jahreswerte für Innsbruck als barplot dar. Achtung: Die Werte müssen vorher eventuell mit `as.numeric` umgewandelt werden



```
1 barplot(ibk,xlab = "Jahre")
```

## 5 Gegenüberstellung mit den Einwohnerzahlen

Hier soll noch der Datensatz über die Einwohnerzahlen, den wir zur gemeinsamen Besprechung verwendet haben, miteinbezogen werden.

Man kann relativ einfach die beiden Datensätze "joinen":

```
1 m <- merge(w,d, by="Gemnr")
```

Um ein wenig aufzuräumen bitte die Spalten wieder umbenennen und die Spalten, die doppelt sind, löschen, das geht so:

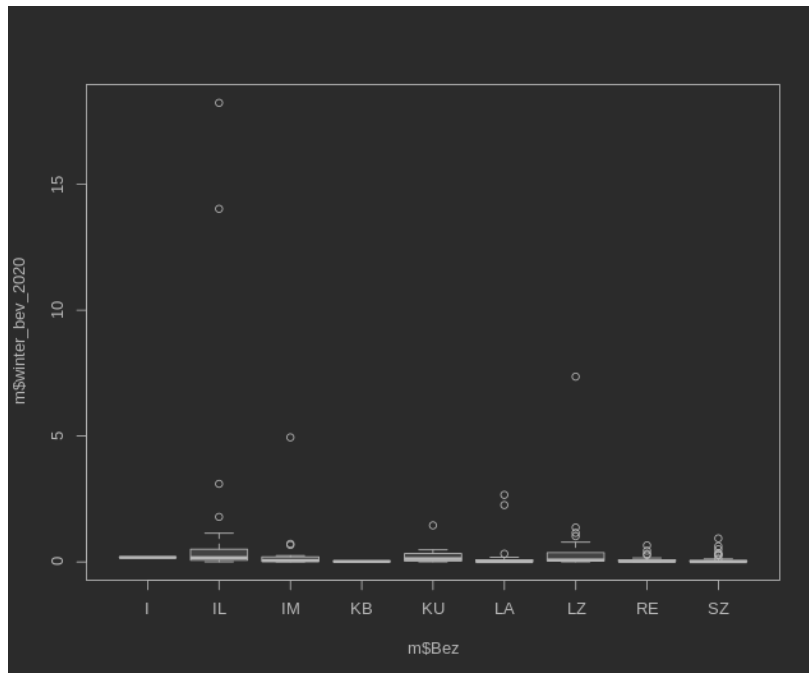
```
1 m$Bezirk <- NULL;  
2 m$Gemeinde.y <- NULL;
```

Bitte mit diesem neu erzeugten Datensatz folgende Fragen beantworten:

- Standardisiere die Anzahl Nächtigungen im Jahr 2018 mit der Bevölkerung pro Gemeinde (auch Jahr 2020). Es soll also berechnet werden, auf wie viele Einwohner kommen wie viele Touristen

```
1 w <- read.csv("../datasets/bev_meld.csv", sep=";", encoding="UTF-8")  
2 getwd()  
3 dim(w)  
4 schwaz <- apply(sz,2, FUN =sum)  
5 m <- merge(w,d, by="Gemnr")  
6 m$Bezirk <- NULL;  
7 m$Gemeinde.y <- NULL;  
8 names(m)
```

- b. Stelle diese Zahl als Boxplot, gruppiert nach Bezirk, dar und versuche, dieses Ergebnis zu interpretieren.



```
1 m$winter_bev_2020 <- m$X2018/m$`Jahr 2018`
2 boxplot(m$winter_bev_2020~m$Bez)
```

- c. Interessant wären jetzt noch die 10 Gemeinden, wo diese Verhältniszahl am größten ist und die, wo es am kleinsten ist. Dazu gibt es folgende Möglichkeit:

```
1 sub <- data.frame(gem=m$Gemeinde.x, m$winter_bev_2020)
2 ord <- order(m$winter_bev_2020, decreasing = T)
3 ord[269:279]
4 ord[1:10]
5 sub[ord[1:10],]
6 sub[ord[269:279],]
```

- d. Wie sieht das Verhältnis in der Heimatgemeinde aus?

```
1 sub[sub$gem == "Stans",]
2 #      gem m.winter_bev_2020
3 # 266 Stans      0.05370013
```