Lageparameter: Modus, Median, Mittelwert

Michael Linke

Table of contents

Laden wir zunächst wieder unsere Daten: Da wir uns drei verschiedene Skalenniveaus ansehen wollen, verwenden wir Daten, die jeweils ein solches repräsentieren. Wir laden daher Daten zu Geschlecht ("sex"), Vertrauen in die Bundesregierung ("pt12") und Netto-Einkommen ("di01a").

Im Folgenden werden wir aus dem Allbus-2021-Datensatz ein paar Beispiele herausgreifen, um die Berechnung und Visualisierung von Häufigkeiten und Parametern zu demonstrieren.

> Video
>
> https://nc.uni-bremen.de/index.php/s/jYBQrNFKZwPWzGX/download/%236%20Lageparameter.mp4

1 Data Management

Dazu installieren und laden wir zunächst die nötigen Pakete mit Hilfe von Pacman und dem `p_load`-Befehl:

Figure 1: Überall Daten, Bild generiert von Midjourney

```
if(!require("pacman")) {install.packages("pacman");library(pacman)}
```

```
Lade nötiges Paket: pacman
```

```
Warning: Paket 'pacman' wurde unter R Version 4.3.1 erstellt
```

```
p_load(tidyverse, ggplot2, haven, dplyr)
theme_set(theme_classic())
```

```
daten = haven::read_dta("Datensatz/Allbus_2021.dta")
```

```
allbus_df <- daten %>%
  select("sex", "pt12", "di01a") %>%
  mutate(across(c("pt12", "di01a"), ~ as.numeric(.))) %>%
  mutate(across(c("pt12", "di01a"), ~ ifelse(.%in% c(-7, -9, -11, -15, -42, -50 ), NA,.))) %
  na.omit()

colnames(allbus_df) <- c("Geschlecht", "VertrauenBR", "Einkommen")
```

2 Der Modus

Der Modus, auch Modalwert genannt, gibt an, welche Ausprägung eines gemessenen Merkmals am häufigsten vorkommt.

Wir müssen einfach nur zählen, wie oft jede Merkmalsausprägung vorkommt. Diejenige mit dem höchsten Wert (bzw. der größten absoluten Häufigkeit) ist der Modus. In R gibt es keine vorgefertigte Funktion, die diesen Parameter berechnet. Das folgende Code-Beispiel zeigt eine mögliche Lösung speziell für unseren "allbus.df$Geschlecht"-DataFrame.

```
get_mode = function(vector){

  # Häufigkeitstabelle erstellen:
  frequencies = table(vector)

  # Höhe der größten Häufigkeit ermitteln:
  max_freq = max(frequencies)

  # Teiltabelle erstellen, die nur die Spalten mit der höchsten Häufigkeit enthält:
  where_max =frequencies == max_freq

  # Namen der verbliebenen Spalten (= Modus) ermitteln:
  modus = names(frequencies[where_max])
  return(modus)
}
#Ausgabe des Modus:
cat("Der Modus lautet ", get_mode(allbus_df$Geschlecht), ".")
```

```
Der Modus lautet  1 .
```

3 Der Median

Für mindestens ordinal skalierte Messwerte empfielt sich neben dem Modus zusätzlich der Median. Einen der Größe nach aufsteigend sortierten Datensatz teilt der Median genau in der Mitte, es liegen also genauso viele Elemente links wie rechts davon.

Für den Fall, dass die Anzahl an Elementen im Datensatz n ungerade ist, entspricht der Median dem Messwert, der genau in der Mitte liegt:

$$x_{Med} = x_{\frac{n+1}{2}}$$

Ist n gerade, kann jedes der beiden Elemente, die in der Mitte liegen, als Median verwendet werden. Es ist aber eher üblich, beide zu addieren und dann durch zwei zu teilen:

$$x_{Med} = \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n}{2}+1})$$

R stellt eine Funktion zur Berechnung des Medians bereit. Wir schauen uns als Beispiel das Vertrauen der Allbus-Befragten in die Bundesregierung an: Im Gegensatz zum Geschlecht können wir hier eine Rangfolge festlegen, jedoch nicht die Abstände dazwischen exakt messen. Wir haben es folglich mit ordinalen Daten zu tun.

Wir verschaffen uns zunächst wieder einen Überblick mit der `table`-Funktion und sehen sieben verschiedene Werte.

```
table(allbus_df$VertrauenBR)
```

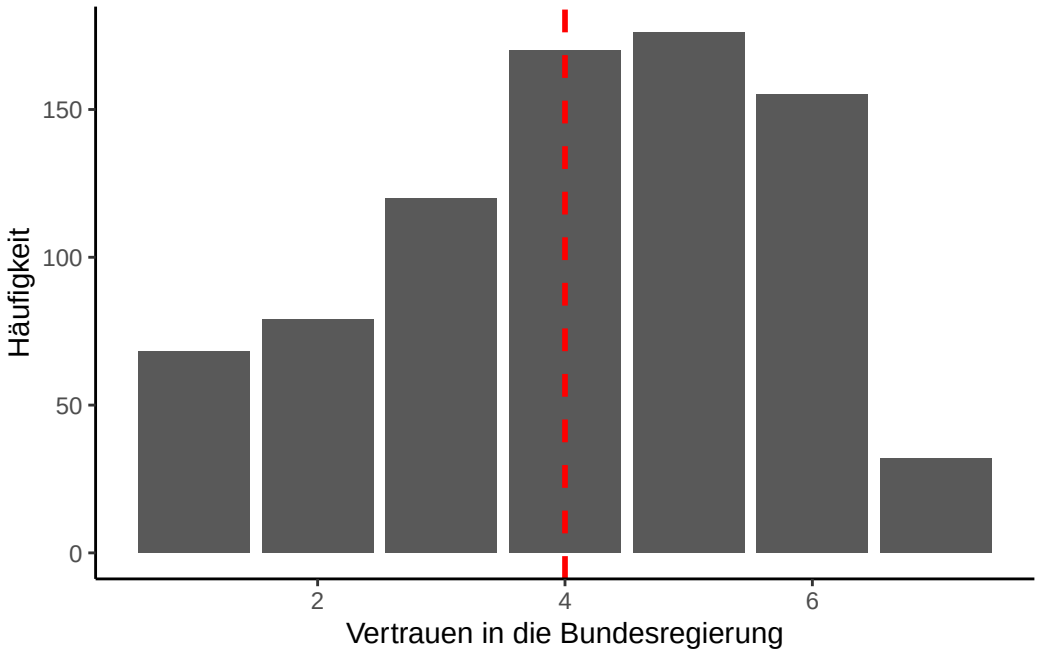
```
  1   2   3   4   5   6   7 
 68  79 120 170 176 155  32 
```

Die Daten sind bereits von "gar nicht" zu "sehr hoch" sortiert. Wir wenden die `median`-Funktion an und erhalten "4" als Ausgabe:

```
median_vertrauen = median(allbus_df$VertrauenBR)
```

Wenn man sich die Daten als Säulendiagramm bzw. Barplot ausgeben lässt und den Median einzeichnet (im folgenden Codebeispiel die rote gestrichelte Linie), kann man erahnen, dass der Median die sieben Klassen so teilt, dass beidseitig gleich viele abgegebene Stimmen liegen:

```
ggplot(data=allbus_df, aes(x=VertrauenBR)) +
  geom_bar() +
  labs(x="Vertrauen in die Bundesregierung", y="Häufigkeit") +
  geom_vline(xintercept = median_vertrauen, color = "red", linetype = "dashed", linewidth = 
```



4 Quantile

Ein Quantil legt fest, wie viele Werte über bzw. unter einer bestimmten Grenze liegen und teilt den Datensatz damit in zwei Teile. Den bekanntesten Spezialfall haben wir mit dem Median bereits kennengelernt. Die Grenze lag in dem Fall genau in der Mitte, es liegen also 50% unterhalb der Grenze und 50% darüber. Bei einem 31%-Quantil würden hingegen 31% der Werte unter der Grenze liegen und 69% darüber. Wichtige Quantile sind die sogenannten Quartile, zu denen das 25%-Quantil, der Median und das 75%-Quantil zusammengefasst werden. Sie teilen die Gesamtmenge an Messwererten in vier gleich große Teile.

Das folgende R-Beispiel gibt die drei Quartile des Vertrauens-Datensatzes aus:

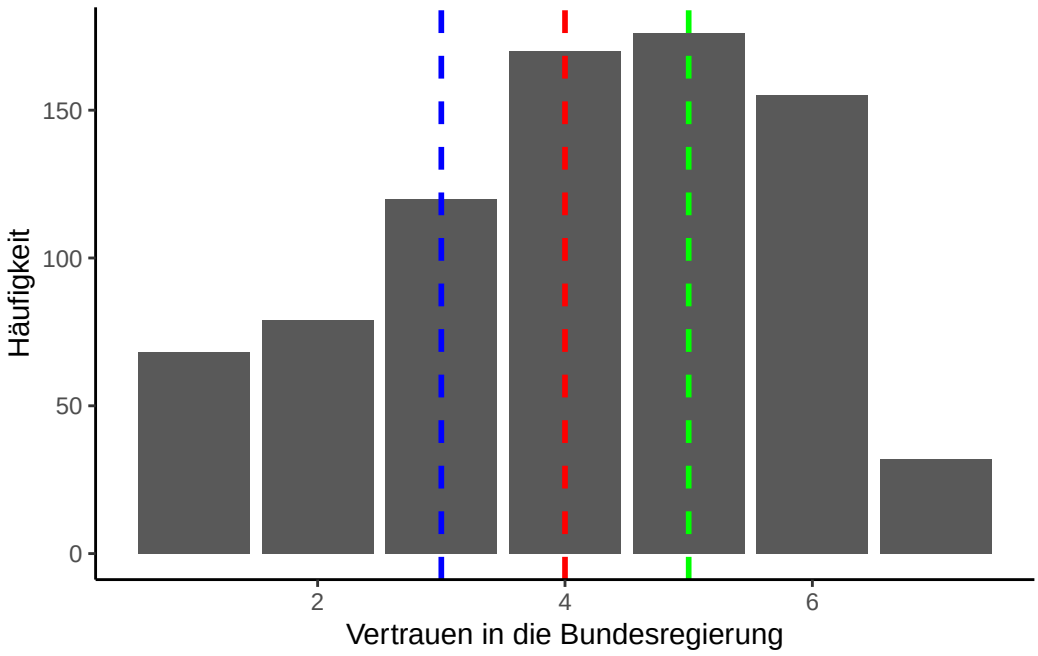
```
quartile = quantile(allbus_df$VertrauenBR, probs = c(0.25, 0.5, 0.75))
quartile
```

```
25% 50% 75%
  3   4   5
```

Das folgende Codebeispiel zeichnet neben dem Median (rot) auch das 25%-Quantil (blau) und das 75%-Quantil (grün) in das Säulendiagramm ein:

```r
quantile25 = quartile["25%"]  #quantile(allbus_df$VertrauenBR, probs=c(0.25))
quantile75 = quartile["75%"]


ggplot(data=allbus_df, aes(x=VertrauenBR)) +
  geom_bar() +
  labs(x="Vertrauen in die Bundesregierung", y="Häufigkeit") +
  geom_vline(xintercept = quantile25, color = "blue", linetype = "dashed", linewidth = 1) +
  geom_vline(xintercept = median_vertrauen, color = "red", linetype = "dashed", linewidth =
  geom_vline(xintercept = quantile75, color = "green", linetype = "dashed", linewidth = 1)
```



5 Der Mittelwert

Der Begriff "Mittelwert" ist etwas ungenau, da es mehrere verschiedene Mittelwerte gibt. Oft ist damit das arithmetische Mittel gemeint. Es lässt sich nur bei mindestens kardinal skalierten Daten anwenden und bezieht die Gewichte der jeweiligen Merkmalsausprägungen mit ein.

Das arithmetische Mittel erhält man, indem man alle Messwerte addiert und durch die Gesamtzahl an Messwerten teilt:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Schauen wir uns das am Beispiel des Netto-Einkommens der Befragten im Allbus-Datensatz an: Wir können dazu die bereits vorhandene Funktion `mean` nutzen:

```
mean_einkommen = mean(allbus_df$Einkommen)
mean_einkommen
```

```
[1] 2442.545
```

Vergleichen wir das mit dem Median, fällt auf, dass zwischen beiden Lageparametern über 200 Euro Differenz bestehen:

```
median_einkommen = median(allbus_df$Einkommen)
median_einkommen
```

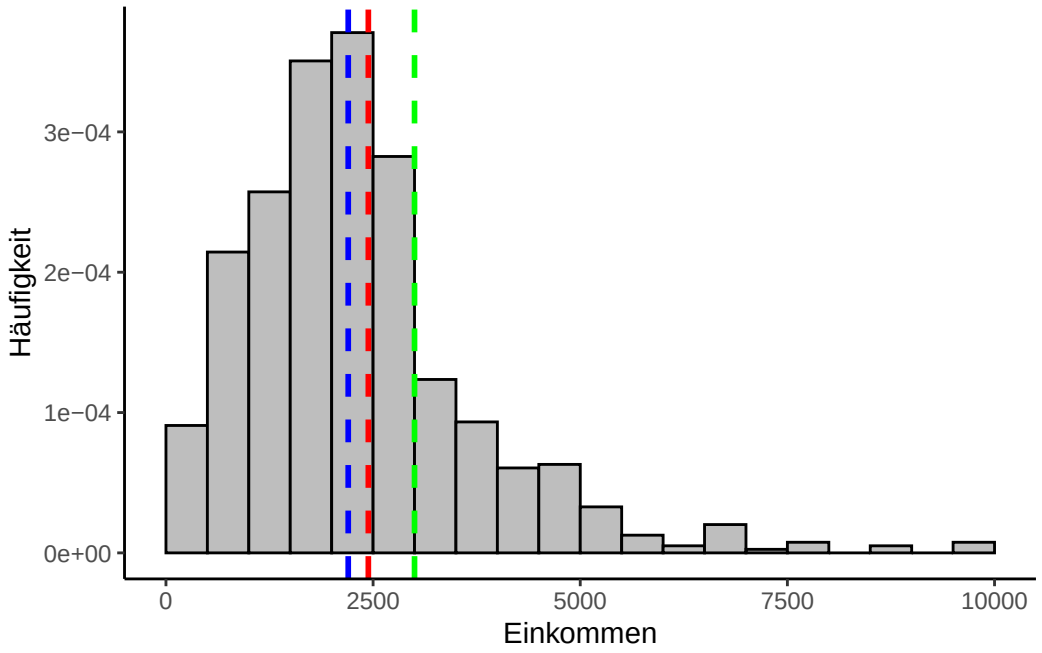
```
[1] 2200
```

Der Modus liegt noch weiter weg:

```
modus_einkommen = get_mode(allbus_df$Einkommen)
modus_einkommen
```

```
[1] "3000"
```

Das liegt daran, dass die Einkommensdaten kontinuierlich sind und es keinen homogenen An- und Abstieg der Häufigkeitsverteilung gibt. Der Einkommenswert, den am meisten Personen exakt gleich angegeben haben, ist deshalb wenig aussagekräftig und der Modus macht nur Sinn, nachdem man die Daten in Form von Einkommensklassen diskretisiert hat.



6 Geometrisches Mittel

Nicht unerwähnt bleiben sollte das geometrische Mittel, das bei der Berechnung des Mittelwerts von prozentualen Veränderungen angewendet wird. Dabei werden die einzelnen Messwerte multipliziert und die n-te Wurzel aus dem Ergebnis gezogen, wobei n die Gesamtzahl an Messwerten ist:

$$x_{Geom} = (x_1 \cdot x_2 \cdot ... \cdot x_n)^{\frac{1}{n}}$$

In R gibt es dafür keine eigenständige Funktion, man kann aber die Gleichung umstellen und mit Hilfe einiger anderer eingebauter Funktionen eine simple Alternative erstellen, indem man einen kleinen Trick mit der Exponentialfunktion und dem natürlichen Logarithmus anwendet:

$$\begin{aligned} x_{Geom} &= (x_1 \cdot x_2 \cdot ... \cdot x_n)^{\frac{1}{n}} \\ &= e^{ln(x_1 \cdot x_2 \cdot ... \cdot x_n)^{\frac{1}{n}}} \\ &= e^{\frac{1}{n} ln(x_1 \cdot x_2 \cdot ... \cdot x_n)} \\ &= e^{\frac{1}{n} \sum_{i=1}^{n} ln(x_i)} \end{aligned}$$

Auch, wenn die resultierende Formel wenig ansprechend aussieht, kann man bei genauerem Hinsehen das versteckte arithmetische Mittel erkennen und den ganzen Ausdruck in folgenden R-Code umsetzen:

```
geom_mean = function(vector){
  exp(mean(log(vector)))
}
```

```
min(allbus_df$Einkommen)
```

```
[1] -41
```

```
geom_einkommen <- geom_mean(allbus_df$Einkommen + 42)
geom_einkommen
```

```
[1] 1994.776
```