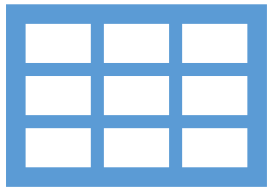
Statistik 1 – Tutorate
Sitzung 6: Univariate Statistik & Illustrationen

Marco Giesselmann, Lea Elina Hofer, Norma De Min, Mara Moos,
Rémy Blum

Lernziele dieser Sitzung



Tabellarische Darstellung

Kategoriale Variablen

Metrische Variablen

Grafische Darstellung

Barplot

Histogramm

Boxplot

0 Working directory setzen (Arbeitsverzeichnis definieren)

- Problem: R speichert Grafiken und Tabellen an den Orten, auf die R zuletzt zugegriffen hat (z.B. Daten- oder Skriptordner). Das ist aber nicht immer erwünscht und führt zu unübersichtlichen Ordnerstrukturen
- Lösung: Definition einer «Working Directory»: R greift dann immer automatisch auf den Ordner zu, der als «Working directory» definiert wurde
- Eventuell ebenfalls hilfreich: Einleseprozess im Skript auch über Working Directory steuern

```
# working directory setzen über Konsole:
# "Set Working Directory" --> "Choose Directory"
# working directory anzeigen lassen
getwd()
# Working directory in Skript kopieren
setwd("mein_laufwerk/mein_datenverzeichnis")
# Daten aus der working directory einlesen
library(haven)
kursdata_anon <- read_dta("kursdata_anon.dta")
# Neues Working Directory für Grafiken etc definieren
setwd("mein_laufwerk/mein_Ergebnisverzeichnis")
```

Tabellarische Darstellung

1.1 Tabellarische Darstellung einer kategorialen Variable

Tabellen sind nützlich, um einen einfachen Überblick zur Verteilung von **kategorialen** oder **kategorisierten** bzw. **klassifizierten** Variablen zu gewinnen.
Für Häufigkeiten und Kreuztabellen verwenden wir den **freq()** Befehl aus dem **summarytools** Package.

```
Label: interesse: Migration und Integration
Type: Numeric

                Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
------------ ------ --------- -------------- --------- --------------
           1     43     58.11          58.11     56.58          56.58
           2     28     37.84          95.95     36.84          93.42
           3      3      4.05         100.00      3.95          97.37
       <NA>       2                                2.63         100.00
       Total     76    100.00         100.00    100.00         100.00
```

```
library(summarytools)
freq(kursdata_anon$intmig)
```

```
Label: interesse: Migration und Integration
Type: Factor

                 Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
-------------- ------ --------- -------------- --------- --------------
          sehr     43     58.11          58.11     56.58          56.58
        etwas      28     37.84          95.95     36.84          93.42
  gar nicht         3      4.05         100.00      3.95          97.37
       <NA>         2                                2.63         100.00
       Total       76    100.00         100.00    100.00         100.00
```

```
kursdata_anon$intmig <- as_factor(kursdata_anon$intmig)
freq(kursdata_anon$intmig)
```

Fragen:

- Was zeigen die einzelnen Spalten der Tabelle?
- Was ist an dieser Tabelle kritikwürdig?

Vor Verwendung von **freq()** sollten wir die Variable faktorisieren. Warum ist hier unsere Standardvariante *as_factor()* sinnvoll?

1.1 Tabellarische Darstellung einer kategorialen Variable

Tabellen sind nützlich, um einen einfachen Überblick zur Verteilung von **kategorialen** oder **kategorisierten** bzw. **klassifizierten** Variablen zu gewinnen.
Für Häufigkeiten und Kreuztabellen verwenden wir den **freq()** Befehl aus dem **summarytools** Package.

```
Label: interesse: Migration und Integration
Type: Numeric

              Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
---------- ------ --------- -------------- --------- --------------
         1     43     58.11          58.11     56.58          56.58
         2     28     37.84          95.95     36.84          93.42
         3      3      4.05         100.00      3.95          97.37
      <NA>      2                                2.63         100.00
     Total     76    100.00         100.00    100.00         100.00
```

```
library(summarytools)
freq(kursdata_anon$intmig)
```

```
Label: interesse: Migration und Integration
Type: Factor

              Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
-------------- ------ --------- -------------- --------- --------------
          sehr     43     58.11          58.11     56.58          56.58
        etwas     28     37.84          95.95     36.84          93.42
   gar nicht      3      4.05         100.00      3.95          97.37
      <NA>        2                                2.63         100.00
     Total       76    100.00         100.00    100.00         100.00
```

```
kursdata_
freq(kursda
```

Die **Ausprägungen** unsere Variablen sind meist **codierte Zahlenwerte**, die **eigentlichen Kategorien** aber im **Wertelabel** abgelegt. ***as_factor*** ersetzt die Zahlenwerte durch die Label und bringt somit direkt die **Bedeutung der Kategorien in den R-Output**.

Fragen:
- Was zeigen die einzelnen Spalten der Tabelle?
- Was ist an dieser Tabelle kritikwürdig?

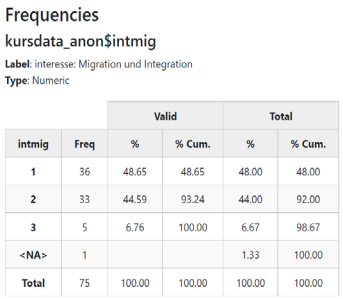
Vor Verwendung von **freq()** [illegible]ten wir die Variable faktorisieren. Warum ist hier unsere Standardvariante *as_factor()* sinnvoll?

1.1 … *publikationswürdige Aufbereitung*

- Tabelle mit „<-“ als Objekt definieren
- Objekt mit „print“ als html speichern
- Öffnen, nach Word kopieren, dort weiterbearbeiten

```
tab_intmig <- freq(kursdata_anon$intmig)
print(tab_intmig, method ="browser", file="intmig.html")
```

intmig.html



Frequencies

kursdata_anon$intmig

Label: interesse: Migration und Integration

Type: Numeric

		Valid		Total	
intmig	Freq	%	% Cum.	%	% Cum.
1	36	48.65	48.65	48.00	48.00
2	33	44.59	93.24	44.00	92.00
3	5	6.76	100.00	6.67	98.67
<NA>	1			1.33	100.00
Total	75	100.00	100.00	100.00	100.00

1.1 *… publikationswürdige Aufbereitung*

- Tabelle mit „<-“ als Objekt definieren
- Objekt mit „print“ als html speichern
- Öffnen, nach Word kopieren, dort weiterbearbeiten

Interesse an Migration und Integration

Wie interessiert bist du an den Themen Migration und Integration?

Ausprägung	Anzahl	%	Kumulierte %
sehr	36	48.65	48.65
etwas	33	44.59	93.24
gar nicht	5	6.76	100.00
keine Angabe	1		
Total	75	100.00	100.00

Quelle: Kursbefragung Statistik I 2022 (n=75)

Dabei achten auf:

- Titel
- Ggf. Fragestellung in Untertitel
- Klare Zeilen- und Spaltennamen
- Quellenangabe der Daten
- Kumulierte Prozente nur bei mindestens ordinalen Variablen
- Ein oder Zwei Nachkommastellen
- Totale Prozente in der Regel nicht ausweisen

1.2 Tabellarische Darstellung einer metrischen Variable

Aufgabe: Erstellt mit **freq()** eine Tabelle für die metrische Variable **lezufr**. Dazu müsst ihr die Variable zuerst sinnvoll klassifizieren bzw. kategorisieren.

Ist das sinnvoll? Was spricht dagegen?

```
kursdata_anon$lezufr_kls[kursdata_anon$lezufr >= 0 & kursdata_anon$lezufr <= 33] <- "unzufrieden"
kursdata_anon$lezufr_kls[kursdata_anon$lezufr >= 34 & kursdata_anon$lezufr <= 66] <- "zufrieden"
kursdata_anon$lezufr_kls[kursdata_anon$lezufr >= 67] <-"sehr zufrieden"
```

```
freq(kursdata_anon$lezufr_kls)
```

Problem: Die Ausprägungen sind alphabetisch (und somit nicht sinnvoll) geordnet. Auch hier hilft die Faktorisierung der Variable. Welche Faktorisierungsvariante ist hier sinnvoll?

```
kursdata_anon$lezufr_kls
Type: Character

                     Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
-------------------- ------ --------- -------------- --------- --------------
      sehr zufrieden     51     68.00          68.00     67.11          67.11
         unzufrieden      5      6.67          74.67      6.58          73.68
           zufrieden     19     25.33         100.00     25.00          98.68
                <NA>      1                               1.32         100.00
               Total     76    100.00         100.00    100.00         100.00
```

1.2 Tabellarische Darstellung einer metrischen Variable

Nutzt ***factor(…, levels(…))*** um die Kategorien einer Variable neu zu ordnen:

```
kursdata_anon$lezufr_kls <- factor(kursdata_anon$lezufr_kls,
      levels = c("unzufrieden", "zufrieden", "sehr zufrieden"))
freq(kursdata_anon$lezufr_kls)
```

```
Type: Factor

                     Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
-------------------- ------ --------- -------------- --------- --------------
        unzufrieden      5      6.67           6.67      6.58           6.58
          zufrieden     19     25.33          32.00     25.00          31.58
      sehr zufrieden     51     68.00         100.00     67.11          98.68
                <NA>      1                               1.32         100.00
               Total     76    100.00         100.00    100.00         100.00
```

1.3 Tabellen mit Verteilungseigenschaften – «Stichprobenstatistik»

Tabellen sind ebenfalls nützlich, um die Verteilungseigenschaften mehrerer (metrischer) Variablen kompakt darzustellen. Hierfür verwenden wir das **stargazer** Package.

```
library(dplyr)
kursdata_T <- select(kursdata_anon, lezufr, llezufr, leftright, alter)
```

Erstellt Teildatensatz mit den interessierenden metrischen Variablen

```
library(stargazer)
stargazer(as.data.frame(kursdata_T), median = T, type = "text",
          title = "Übersichtstabelle: Univariate Statistik", digits = 2)
```

Bereitet die Verteilungscharakterisika der ausgewählten Variablen kompakt auf

```
Übersichtstabelle: Univariate Statistik
=============================================================
Statistic N  Mean  St. Dev. Min  Pctl(25) Median Pctl(75)  Max
-------------------------------------------------------------
lezufr    75 66.11  28.24   -99     60      72      80     100
llezufr   75 58.17  24.53    0      40.5    63      75     100
leftright 74 26.36  19.87   0.00   10.00   25.00   40.00  90.00
alter     75 21.71   2.05    19     20      21      23     25
-------------------------------------------------------------
```

Achtung: Daten aus 2022!

Zuerst müssen wir die Missings richtig kodieren. Wie?

```
kursdata_anon$lezufr[kursdata_anon$lezufr == -99] <- NA
summary(kursdata_anon$lezufr)
```

1.3 Tabellen mit Verteilungseigenschaften – «Stichprobenstatistik»

Tabellen sind ebenfalls nützlich, um die Verteilungseigenschaften mehrerer (metrischer) Variablen kompakt darzustellen. Hierfür verwenden wir das **stargazer** Package.

```
library(dplyr)
kursdata_T <- select(kursdata_anon, lezufr, llezufr, leftright, alter)
```

Erstellt Teildatensatz mit den interessierenden metrischen Variablen

```
library(stargazer)
stargazer(as.data.frame(kursdata_T), median = T, type = "text",
          title = "Übersichtstabelle: Univariate Statistik", digits = 2)
```

Bereitet die Verteilungscharakterisika der ausgewählten Variablen kompakt auf

```
Übersichtstabelle: Univariate Statistik
=================================================================
Statistic N  Mean  St. Dev. Min  Pctl(25) Median Pctl(75)  Max
-----------------------------------------------------------------
lezufr    74 68.34  20.73   0.00  60.00   72.50   80.00   100.00
llezufr   75 58.17  24.53    0     40.5     63      75      100
leftright 74 26.36  19.87   0.00  10.00   25.00   40.00   90.00
alter     75 21.71   2.05    19    20      21      23      25
-----------------------------------------------------------------
```

Achtung: In neueren stargazer-Versionen braucht es einen Befehlszusatz zur Darstellung der Quartile. Finde ggf. heraus, welcher es ist.

Beschreibe die Entwicklung der Lebenszufriedenheit der Studierenden innerhalb des letzten Jahres

1.3 Tabellen mit Verteilungseigenschaften – «Stichprobenstatistik»

Tabellen sind ebenfalls nützlich, um die Verteilungseigenschaften mehrerer (metrischer) Variablen kompakt darzustellen. Hierfür verwenden wir das **stargazer** Package.

```
library(dplyr)
kursdata_T <- select(kursdata_anon, lezufr, llezufr, leftright, alter)
```

Erstellt Teildatensatz mit den interessierenden metrischen Variablen

```
library(stargazer)
stargazer(as.data.frame(kursdata_T), median = T, type = "text",
          title = "Übersichtstabelle: Univariate Statistik", digits = 2)
```

Bereitet die Verteilungscharakterisika der ausgewählten Variablen kompakt auf

```
Übersichtstabelle: Univariate Statistik
=================================================================
Statistic N  Mean  St. Dev. Min  Pctl(25) Median Pctl(75)  Max
-----------------------------------------------------------------
lezufr    74 68.34  20.73   0.00  60.00   72.50   80.00   100.00
llezufr   75 58.17  24.53    0     40.5     63      75      100
leftright 74 26.36  19.87   0.00  10.00   25.00   40.00   90.00
alter     75 21.71   2.05    19    20      21      23      25
-----------------------------------------------------------------
```

Warum ist es in dieser Anwendung sinnvoll, im Rahmen der „select“-Prozedur einen neuen Datensatz anzulegen?

«Select» dient hier **nicht** der Schaffung von Übersichtlichkeit, sondern ist **Teil der befehlsspezifischen Variablenauswahl** – im Rahmen des «stargazer»-Befehls können keine Variablen (sondern nur ganze Matrixen) gezielt spezifiziert werden. **Der Ausgangsdatensatz soll daher erhalten bleiben**

1.3 Tabellen mit Verteilungseigenschaften – «Stichprobenstatistik»

Tabellen sind ebenfalls nützlich, um die Verteilungseigenschaften mehrerer (metrischer) Variablen kompakt darzustellen. Hierfür verwenden wir das **stargazer** Package.

```
library(dplyr)
kursdata_T <- select(kursdata_anon, lezufr, llezufr, leftright, alter)

library(stargazer)
stargazer(as.data.frame(kursdata_T), median = T, type = "text",
          title = "Übersichtstabelle: Univariate Statistik", digits = 2)
```

```
Übersichtstabelle: Univariate Statistik
=================================================================
Statistic N  Mean  St. Dev. Min  Pctl(25) Median Pctl(75)  Max
-----------------------------------------------------------------
lezufr    74 68.34  20.73   0.00  60.00   72.50   80.00   100.00
llezufr   75 58.17  24.53    0     40.5     63      75      100
leftright 74 26.36  19.87   0.00  10.00   25.00   40.00   90.00
alter     75 21.71   2.05    19     20      21      23      25
-----------------------------------------------------------------
```

Auch diese Tabelle muss vor der Publikation noch extern bearbeitet und formatiert werden.

1.3 Tabellen mit Verteilungseigenschaften – «Stichprobenstatistik»

Tabellen sind ebenfalls nützlich, um die Verteilungseigenschaften mehrerer (metrischer) Variablen kompakt darzustellen. Hierfür verwenden wir das **stargazer** Package.

```
library(dplyr)
kursdata_T <- select(kursdata_anon, lezufr, llezufr, leftright, alter)

library(stargazer)
stargazer(as.data.frame(kursdata_T), median = T, type = "text",
          title = "Übersichtstabelle: Univariate Statistik", digits = 2)
```

```
Übersichtstabelle: Univariate Statistik
=========================================================================
Statistic N  Mean  St. Dev. Min  Pctl(25) Median Pctl(75)  Max
-------------------------------------------------------------------------
lezufr    74 68.34  20.73   0.00  60.00   72.50   80.00   100.00
llezufr   75 58.17  24.53    0     40.5     63      75      100
leftright 74 26.36  19.87   0.00  10.00   25.00   40.00   90.00
alter     75 21.71   2.05    19     20      21      23      25
-------------------------------------------------------------------------
```

oder: `type = "html", out = "stargazer.doc",`

Übersichtstabelle: Univariate Statistik

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
lezufr	74	68.34	20.73	0.00	60.00	72.50	80.00	100.00
llezufr	75	58.17	24.53	0	40.5	63	75	100
leftright	74	26.36	19.87	0.00	10.00	25.00	40.00	90.00
alter	75	21.71	2.05	19	20	21	23	25

Auch diese Tabelle muss vor der Publikation noch extern bearbeitet und formatiert werden.

- Geeignet dafür ist insb. das Ausgangsformat «html».
- Innerhalb des stargazer-Befehls kann zudem direkt ein Dokument im Zielformat angelegt werden.
- Speicherort ist per default die working directory!

1.3 Tabellen mit Verteilungseigenschaften – «Stichprobenstatistik»

Nach der Formatierung könnte eine publikationswürdige Tabelle etwa so aussehen:

Übersichtstabelle: Univariate Statistik

	Anzahl	Durchschnitt	Standard-abweichung	Min	1. Quartil	Median	3. Quartil	Max	95%- KI (Durchschnitt)
Aktuelle Lebenszufriedenheit	74	68.34	20.73	0.00	60.00	72.50	80.00	100.00	63.53/73.14
Lebenszufriedenheit vor 1 Jahr	75	58.17	24.53	0	40.5	63	75	100	52.53/63.82
Politische Position auf links-rechts Skala	74	26.36	19.87	0.00	10.00	25.00	40.00	90.00	21.76/30.97
Alter	75	21.71	2.05	19	20	21	23	25	21.24/22.18

Quelle: Kursbefragung Statistik I 2022 (n=75)

… manuell kann dann in Word oder Excel z.B. der Standardfehler oder, wie hier und auch üblicher, das 95% Konfidenzintervall für den Mittelwert angefügt werden:

```
ci_leftright <- t.test(kursdata_anon$leftright)
ci_leftright$conf.int
```

```
> ci_leftright$conf.int
[1] 21.76108 30.96865
attr(,"conf.level")
[1] 0.95
```

Der wahre Mittelwert der Population liegt mit 95% Sicherheit zwischen 21,8 und 31.

Übung

1. Erstelle einen Teildatensatz mit den metrischen Variablen zu den Schulnoten.
2. Stelle die Verteilungscharacteristika der beiden Variablen übersichtlich in einer Tabelle dar.
3. Werte die Tabelle aus (bzw. bereite einen kurzen Vortrag zur Schulperformance Soziologiestudierender vor).
4. Ermittle das 95% CI und binde es in eine extern formatierte Tabelle (wie auf Folien 15/16) ein.
5. Interpretiere die 95% CI

Grafische Darstellung

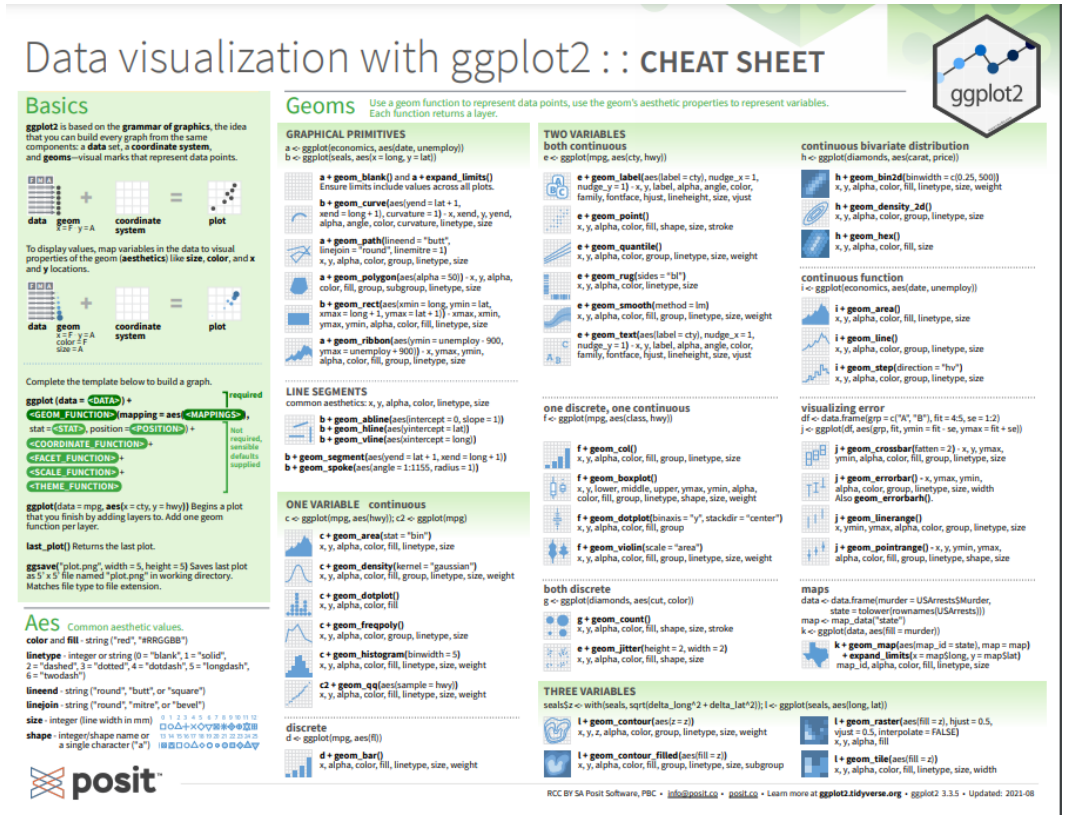
2.0 Grafische Darstellung: Einführung

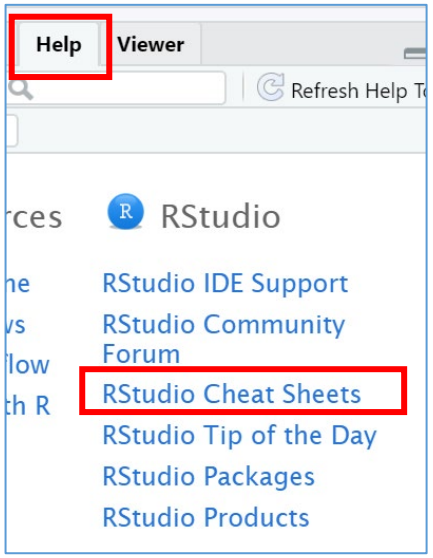
- Wir erstellen Grafiken in R/ R-Studio mit dem Package **ggplot2**.
- Das Package **ggplot2** ist im Package-Bündel **tidyverse** inkludiert.
- Ihr müsst **ggplot2** also nur installieren/aktivieren, falls ihr **tidyverse** noch nicht installiert/aktiviert habt.

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

2.0 Grafische Darstellung: Einführung

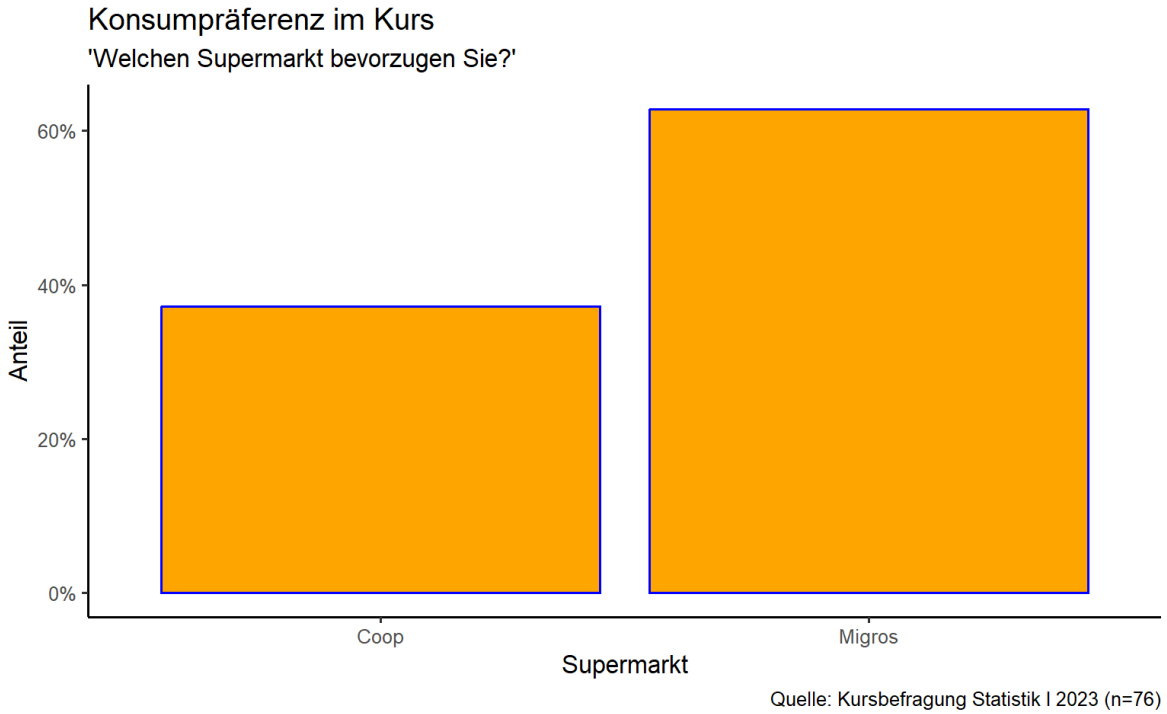
Um uns einen Überblick über die Funktionen von ggplot2 zu verschaffen, schauen wir uns das Cheat Sheet an





2.1 Grafische Darstellung: kategoriale Variablen (Barplot)

Barplots (Säulendiagramme) sind einfach zu interpretierende Darstellungen der Häufigkeitsverteilung von kategorialen Variablen. Hier ein mit **ggplot()** erstellter Barplot:



Frage:

- Was zeigen die Bars (oder Säulen bzw. Balken) an?
- Welche Aussage enthält die Abbildung?

Im folgenden Schritt-für-Schritt: Wie müssen wir den ggplot-Befehl aufbauen und ausgestalten, um zu dieser Abbildung zu kommen?

2.1 Umsetzung mit ggplot: Schritt für Schritt

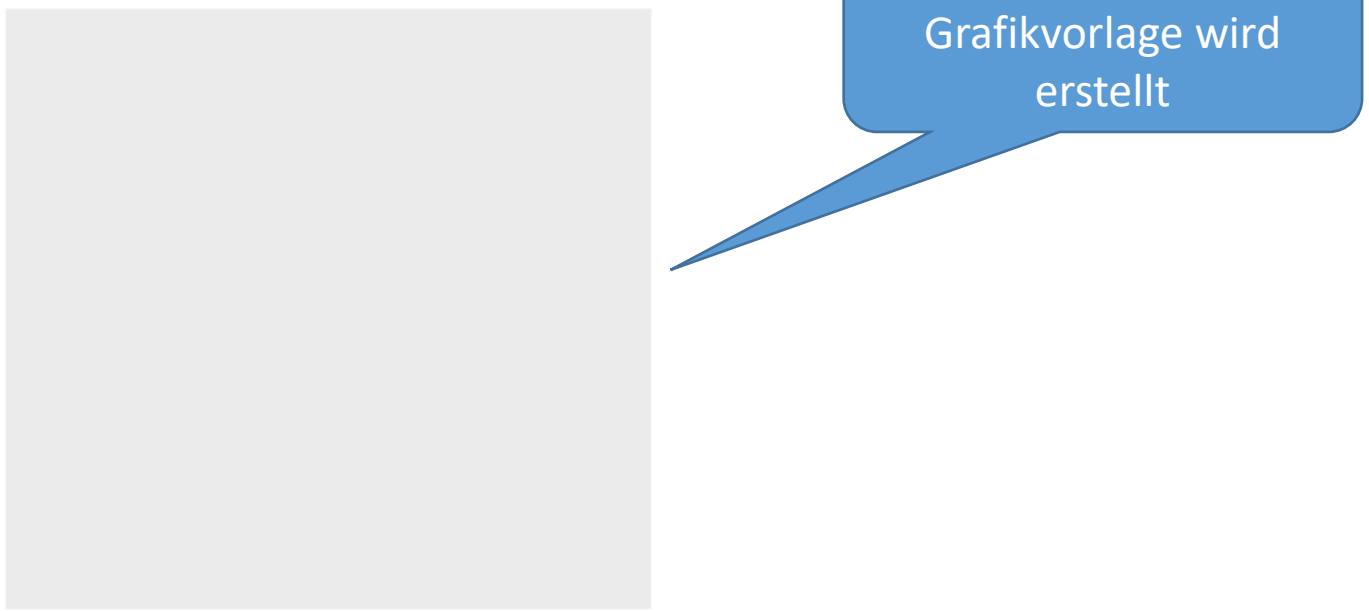
```
library(ggplot2)
plot_1 <- ggplot(kursdata_anon)
plot_1
```

Tippt jeweils den Code ab und erklärt:
Was für eine Anweisung ist das? Was ist ihr Ergebnis?

2.1 Umsetzung mit ggplot: Schritt für Schritt

```
library(ggplot2)
plot_1 <- ggplot(kursdata_anon)
plot_1
```

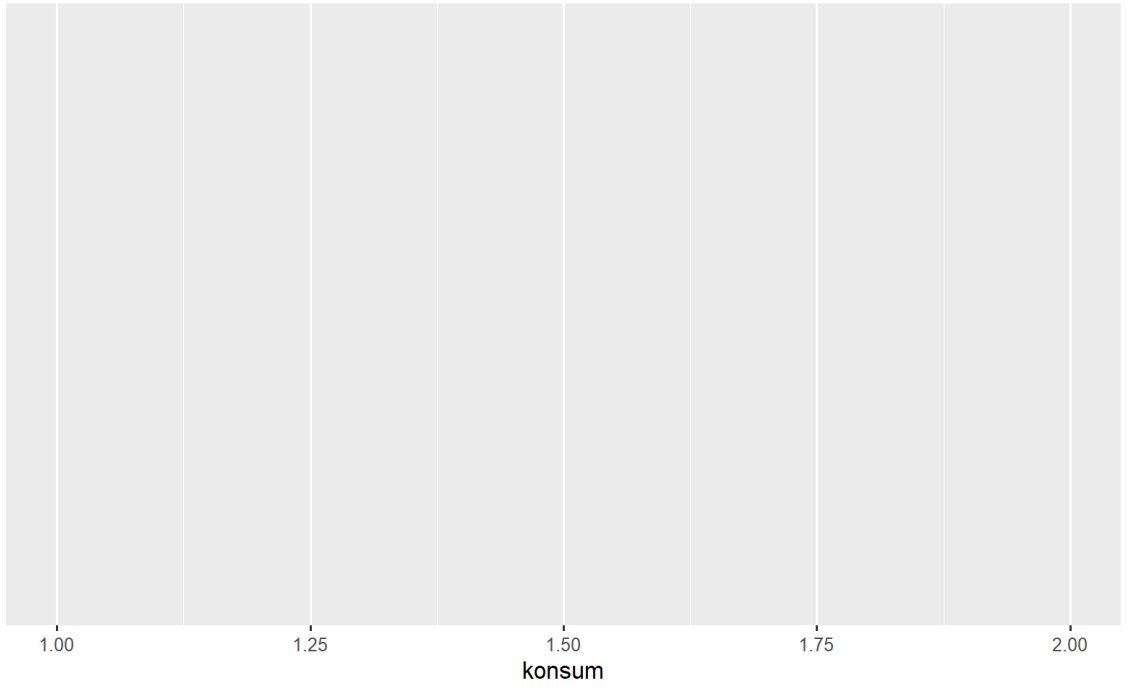
Erstelle mit ggplot eine Abbildung zum Datensatz ‚kursdata_anon', nenne diese ‚plot1'



2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon, aes(x = konsum))
plot_1
```

*…platziere Ausprägungen der Variable **konsum** auf der x- bzw. horizontalen Achse der Abbildung*



Problem?
Lösungsvorschläge?

2.1 Umsetzung mit ggplot

```
kursdata_anon$konsum<-as_factor(kursdata_anon$konsum)
plot_1 <- ggplot(kursdata_anon, aes(x = konsum))
plot_1
```

Faktorisiere die kategoriale x-Variable

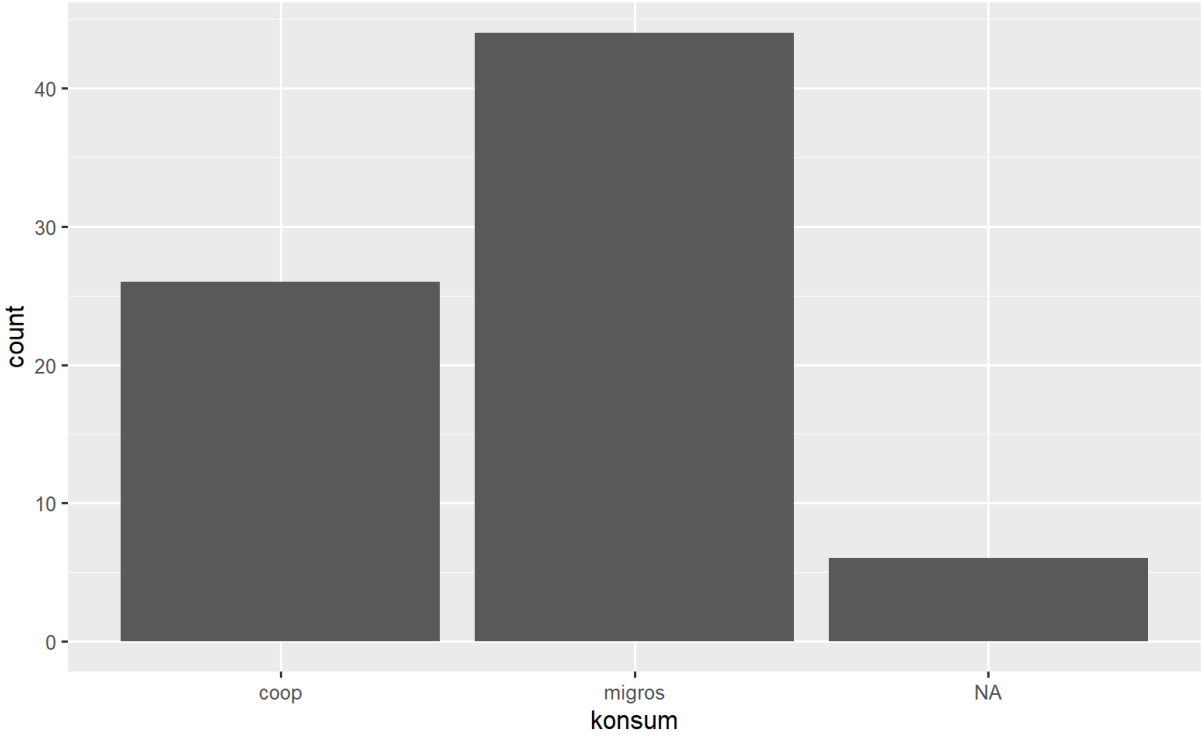
…platziere Ausprägungen der Variable konsum auf der x- bzw. horizontalen Achse der Abbildung

Durch Faktorisierung: diskrete und inhaltliche Kategorien!

2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon, aes(x = konsum))+
          geom_bar()
plot_1
```

*Stelle die Verteilung der in **aes()** spezifizierten Variable als Bar- bzw. Balkendiagramm dar*

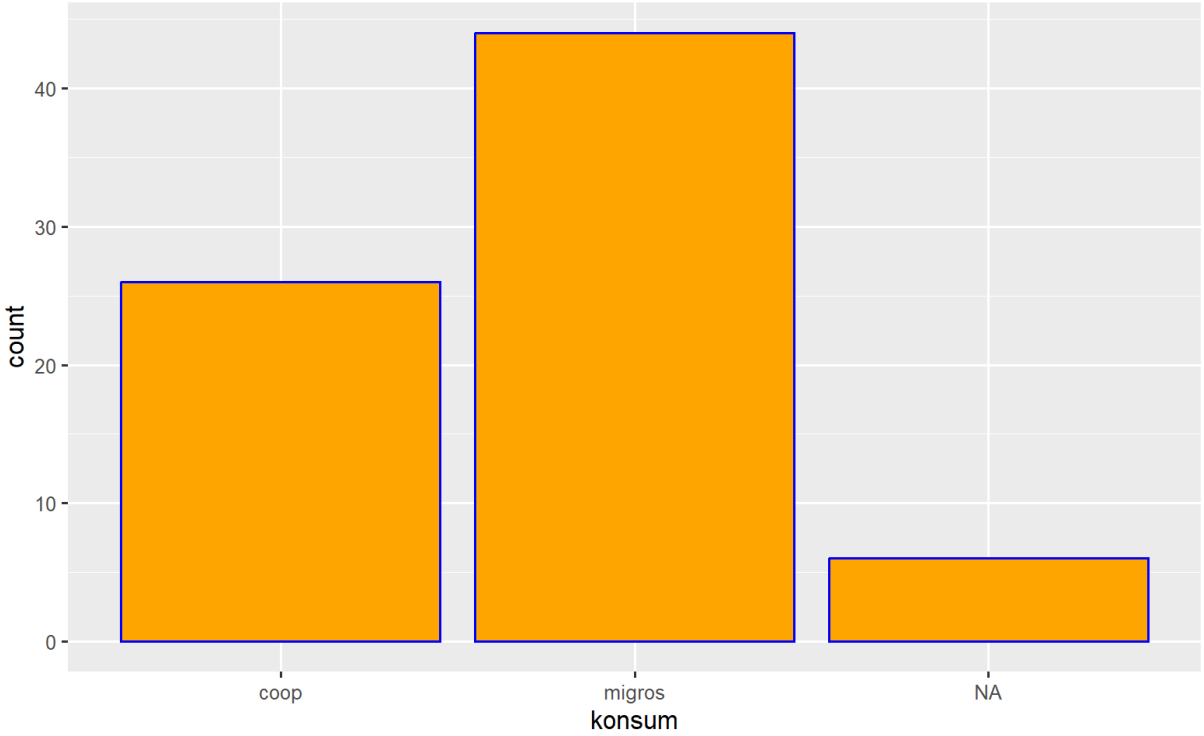


2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon, aes(x = konsum))+
          geom_bar(colour = "blue", fill = "orange")
plot_1
```

Färbe die Balken orange und umrande sie blau.

Gestalte nun die Bars entsprechend deiner Lieblingsfarben!



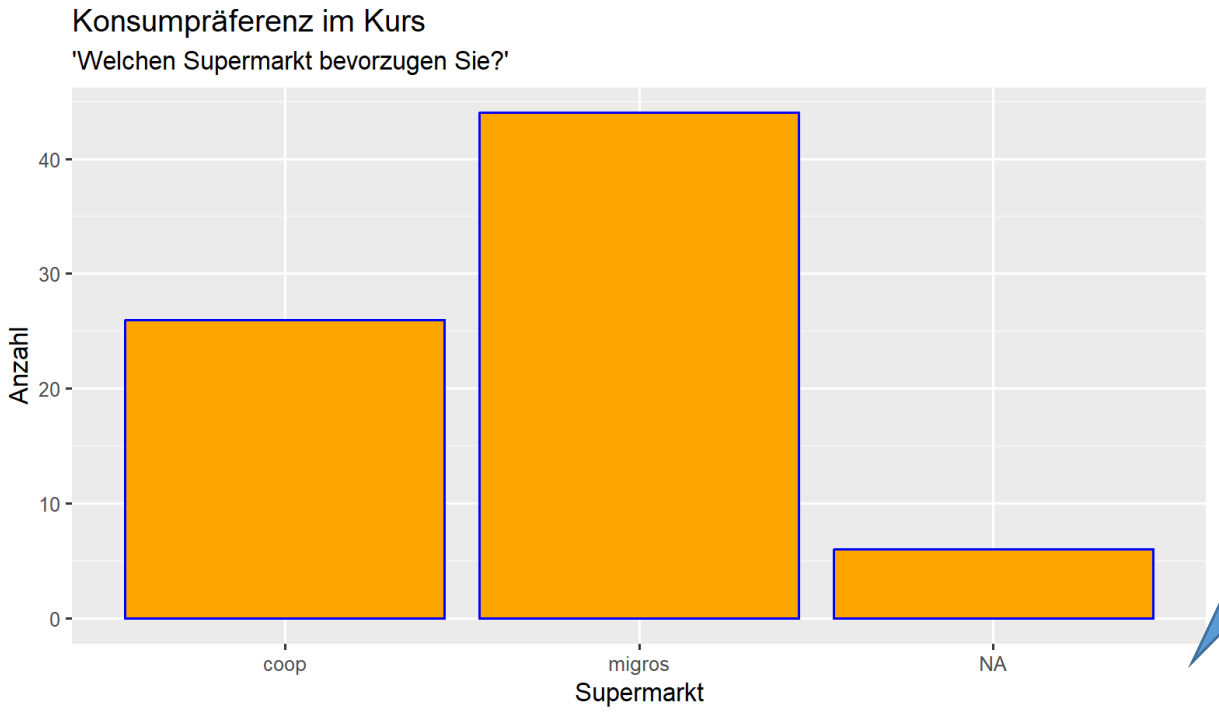
white	aliceblue	antiquewhite	antiquewhite1	antiquewhite2
antiquewhite3	antiquewhite4	aquamarine	aquamarine1	aquamarine2
aquamarine3	aquamarine4	azure	azure1	azure2
azure3	azure4	beige	bisque	bisque1
bisque2	bisque3	bisque4		blanchedalmond
blue	blue1	blue2	blue3	blue4
blueviolet	brown	brown1	brown2	brown3
brown4	burlywood	burlywood1	burlywood2	burlywood3
burlywood4	cadetblue	cadetblue1	cadetblue2	cadetblue3
cadetblue4	chartreuse	chartreuse1	chartreuse2	chartreuse3
chartreuse4	chocolate	chocolate1	chocolate2	chocolate3
chocolate4	coral	coral1	coral2	coral3
coral4	cornflowerblue	cornsilk	cornsilk1	cornsilk2
cornsilk3	cornsilk4	cyan	cyan1	cyan2
cyan3	cyan4	darkblue	darkcyan	darkgoldenrod
darkgoldenrod1	darkgoldenrod2	darkgoldenrod3	darkgoldenrod4	darkgray
darkgreen	darkgrey	darkkhaki	darkmagenta	darkolivegreen
darkolivegreen1	darkolivegreen2	darkolivegreen3	darkolivegreen4	darkorange
darkorange1	darkorange2	darkorange3	darkorange4	darkorchid
darkorchid1	darkorchid2	darkorchid3	darkorchid4	darkred
darksalmon	darkseagreen	darkseagreen1	darkseagreen2	darkseagreen3
darkseagreen4	darkslateblue	darkslategray	darkslategray1	darkslategray2
darkslategray3	darkslategray4	darkslategrey	darkturquoise	darkviolet
deeppink	deeppink1	deeppink2	deeppink3	deeppink4
deepskyblue	deepskyblue1	deepskyblue2	deepskyblue3	deepskyblue4

2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  labs(x = "Supermarkt", y = "Anzahl", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")
 plot_1
```

Muss nicht 1:1 abgetippt werden!

Beschrifte die Abbildung, so dass sie selbsterklärend ist!



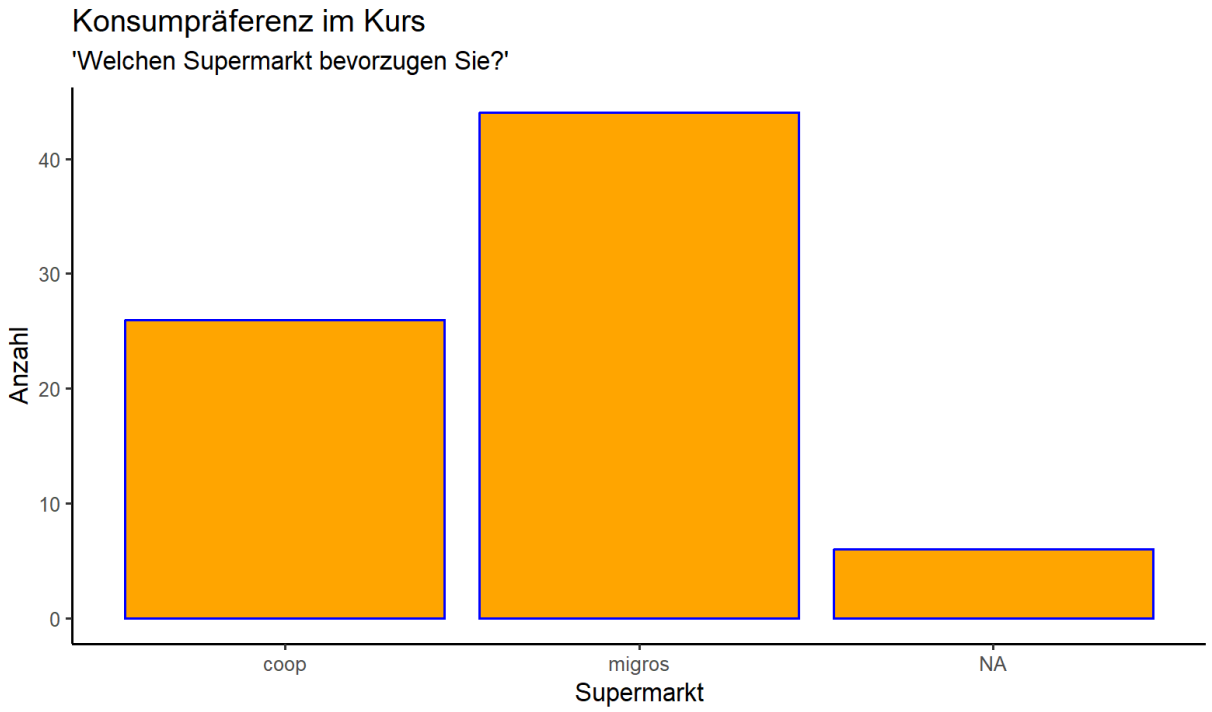
Die Achsen werden beschriftet, ein Titel hinzugefügt und eine Note mit Quellenangaben integriert

2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  labs(x = "Supermarkt", y = "Anzahl", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  theme_classic()
plot_1
```

*Verwende den neutralen Hintergrund **theme_classic()**. Weitere Möglichkeiten:*

```
theme_bw        {ggplot2}
theme_classic   {ggplot2}
theme_dark      {ggplot2}
theme_get       {ggplot2}
theme_gray      {ggplot2}
theme_grey      {ggplot2}
theme_light     {ggplot2}
```



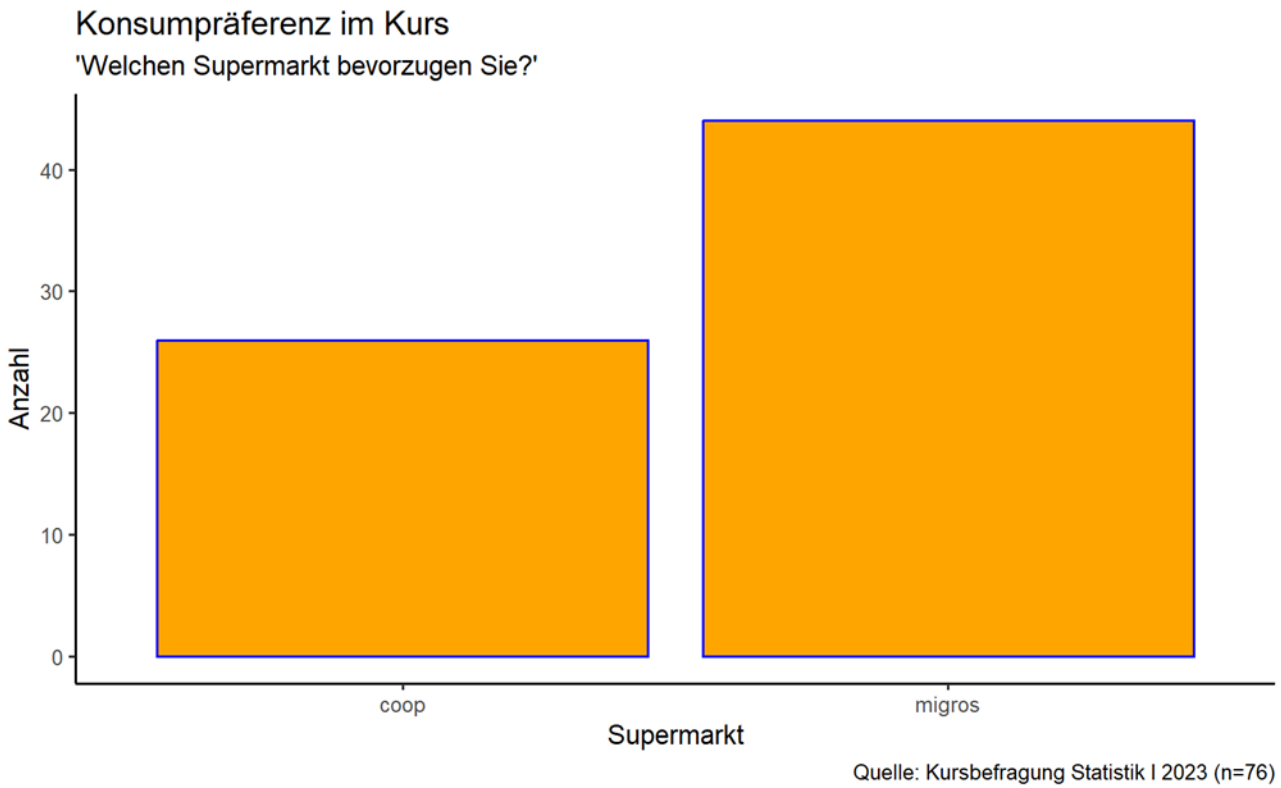
Wie können wir die Kategorie ‘NA’ aus der Abbildung entfernen?

2.1 Umsetzung mit ggplot

```
library(dplyr)
kursdata_anon_N <- filter(kursdata_anon, konsum != "NA")
```

Filtere vorab Merkmalsträger mit NA aus

```
plot_1 <- ggplot(kursdata_anon_N, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  labs(x = "Supermarkt", y = "Anzahl", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  theme_classic()
plot_1
```

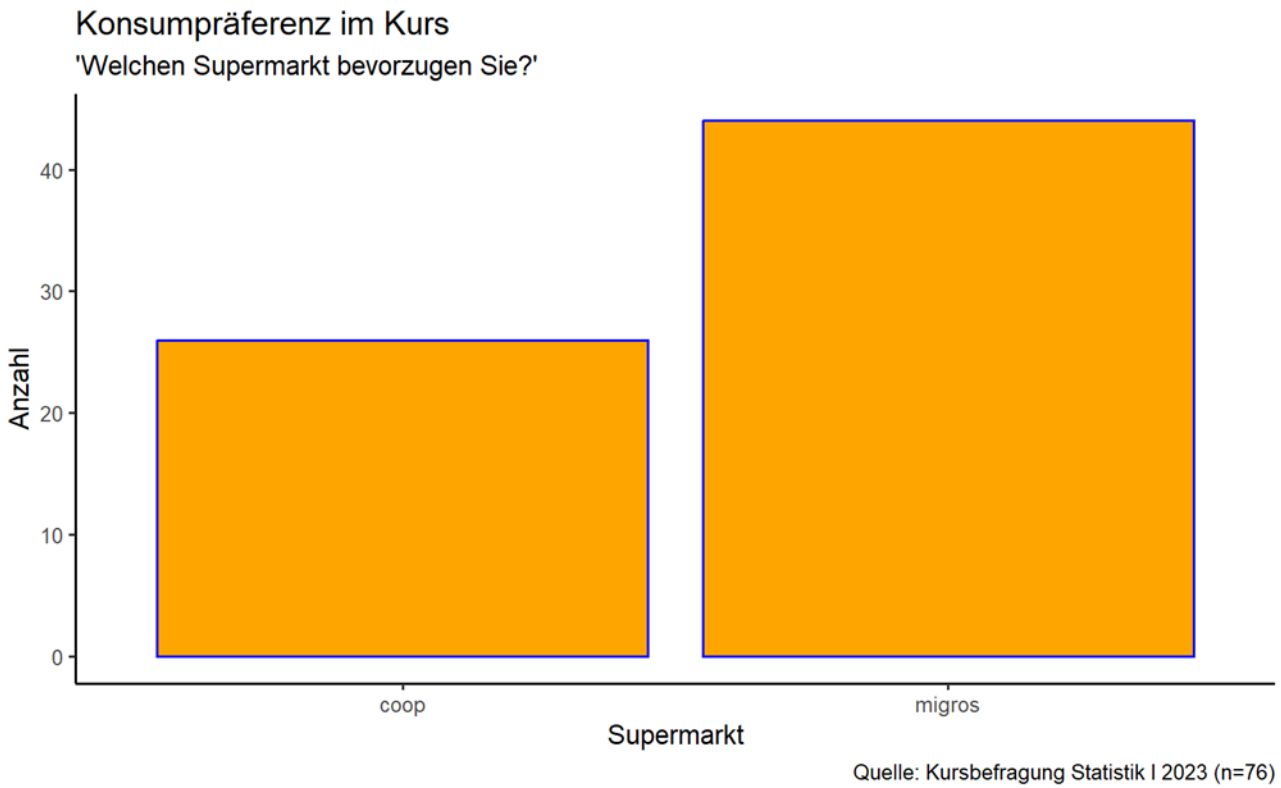


Wie können wir die Kategorie ‘NA’ aus der Abbildung entfernen?

2.1 Umsetzung mit ggplot

```
library(dplyr)
kursdata_anon_N <- filter(kursdata_anon, konsum != "NA")

plot_1 <- ggplot(kursdata_anon_N, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  labs(x = "Supermarkt", y = "Anzahl", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  theme_classic()
plot_1
```

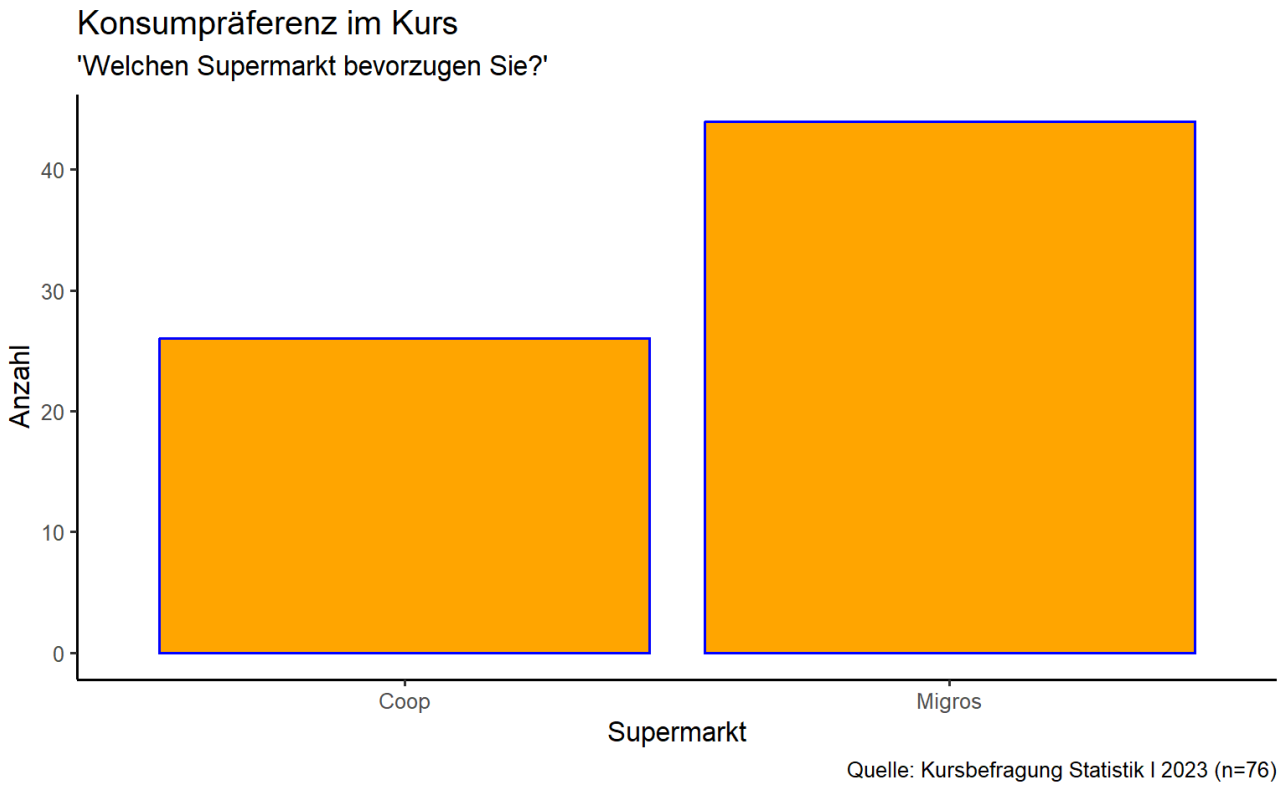


Die Kategoriennamen sollen gross geschrieben werden. Wie kann das umgesetzt werden?

2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon_N, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  scale_x_discrete(labels = c("Coop", "Migros"))+
  labs(x = "Supermarkt", y = "Anzahl", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  theme_classic()
plot_1
```

modifiziere die Label auf der (diskreten) x-Achse



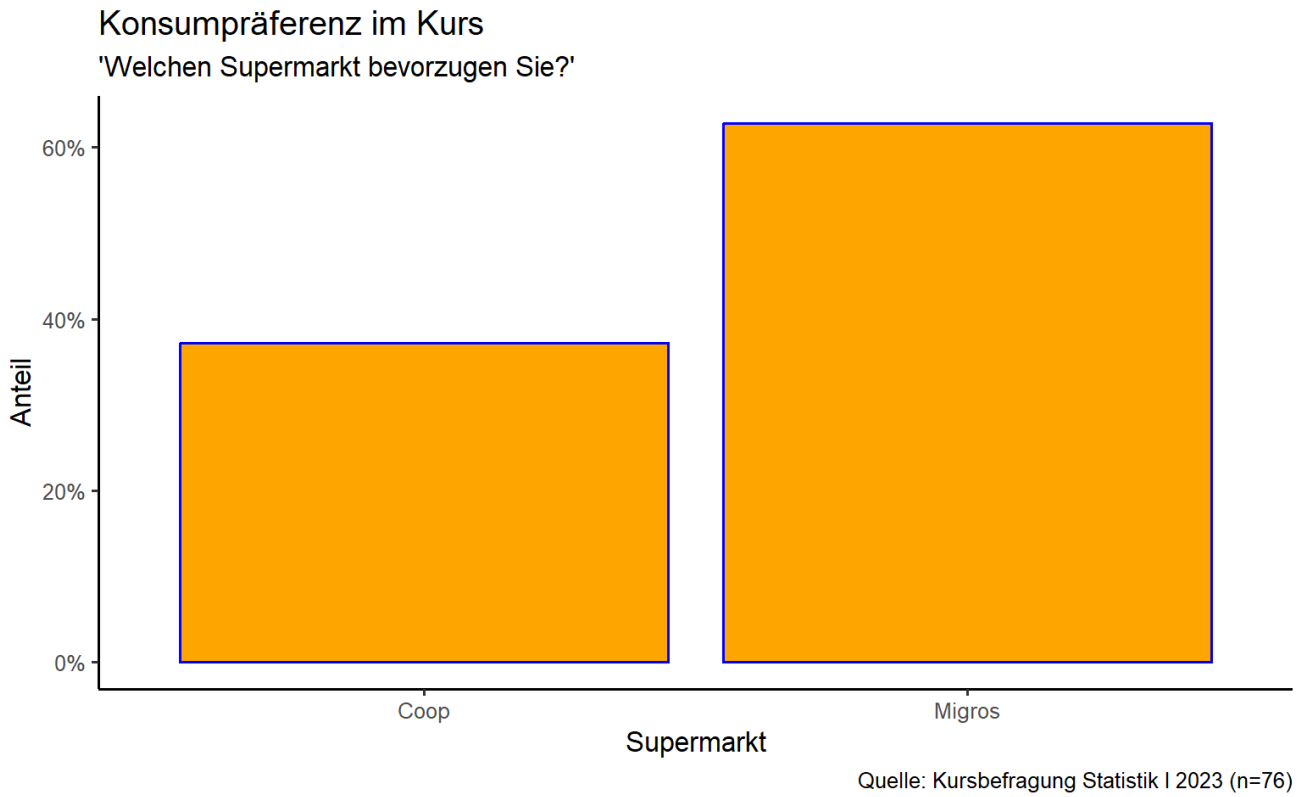
Die Kategoriennamen sollen gross geschrieben werden. Wie kann das umgesetzt werden?

2.1 Umsetzung mit ggplot

```
plot_1 <- ggplot(kursdata_anon_N, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  scale_x_discrete(labels = c("Coop", "Migros"))+
  aes(y = after_stat(count / sum(count)))+
  scale_y_continuous(labels = scales::percent)+
  labs(x = "Supermarkt", y = "Anteil", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  theme_classic()
plot_1
```

Wandle absolute Werte auf der stetigen y-Achse in Anteilswerte

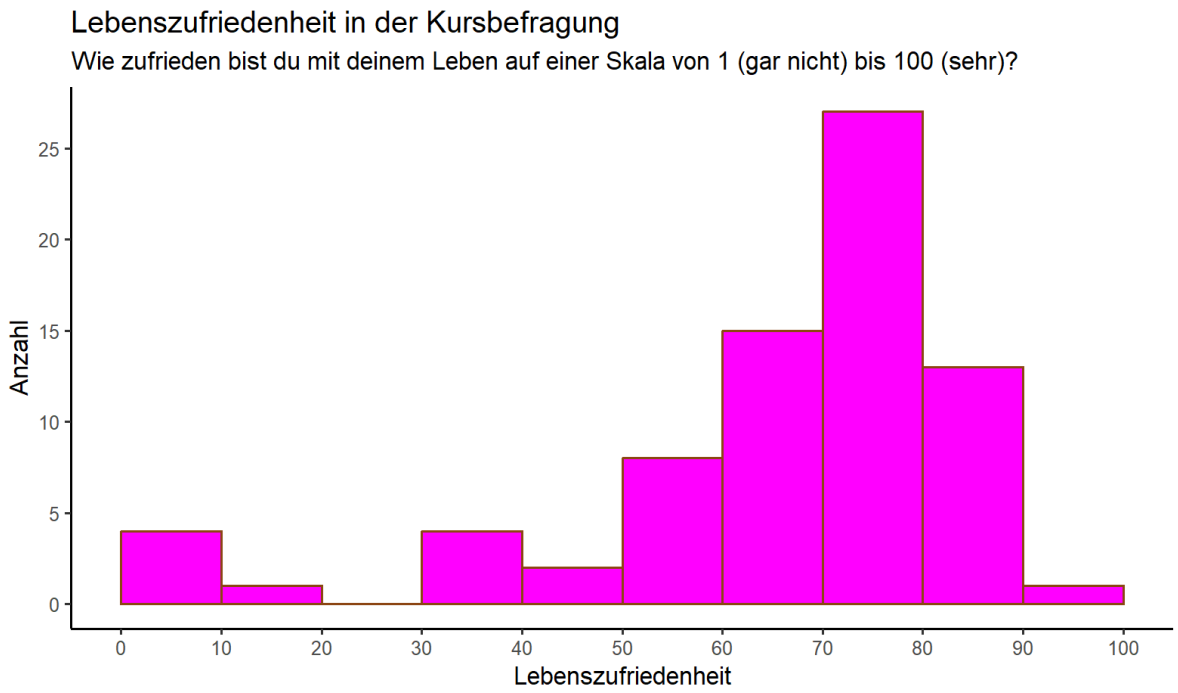
Schreibe Anteilswerte als Prozente



Fine-tuning (per chatgpt):

- Die Schrift soll ingesamt etwas grösser
- Der Untertitel soll kursiv gesetzt sein
- Zusatzaufgabe: Die Reihenfolge der Balken soll vertauscht werden

2.2 Grafische Darstellung: metrischer Variablen (Histogramm)

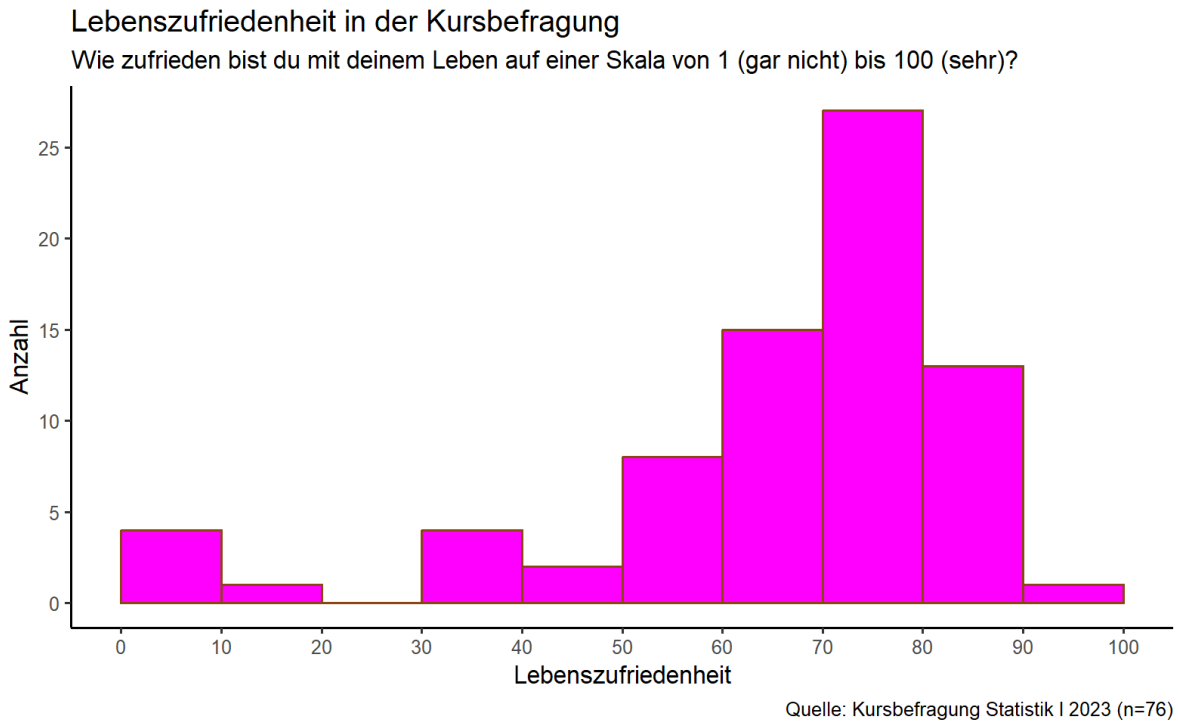


Quelle: Kursbefragung Statistik I 2023 (n=76)

Diskussion:

- Was ist hier dargestellt?
- Worin unterscheidet sich diese Grafik vom Barplot?
- Wieso sind Histogramme für kontinuierliche metrische Variablen geeignet?

2.2 Grafische Darstellung: metrischer Variablen (Histogramm)



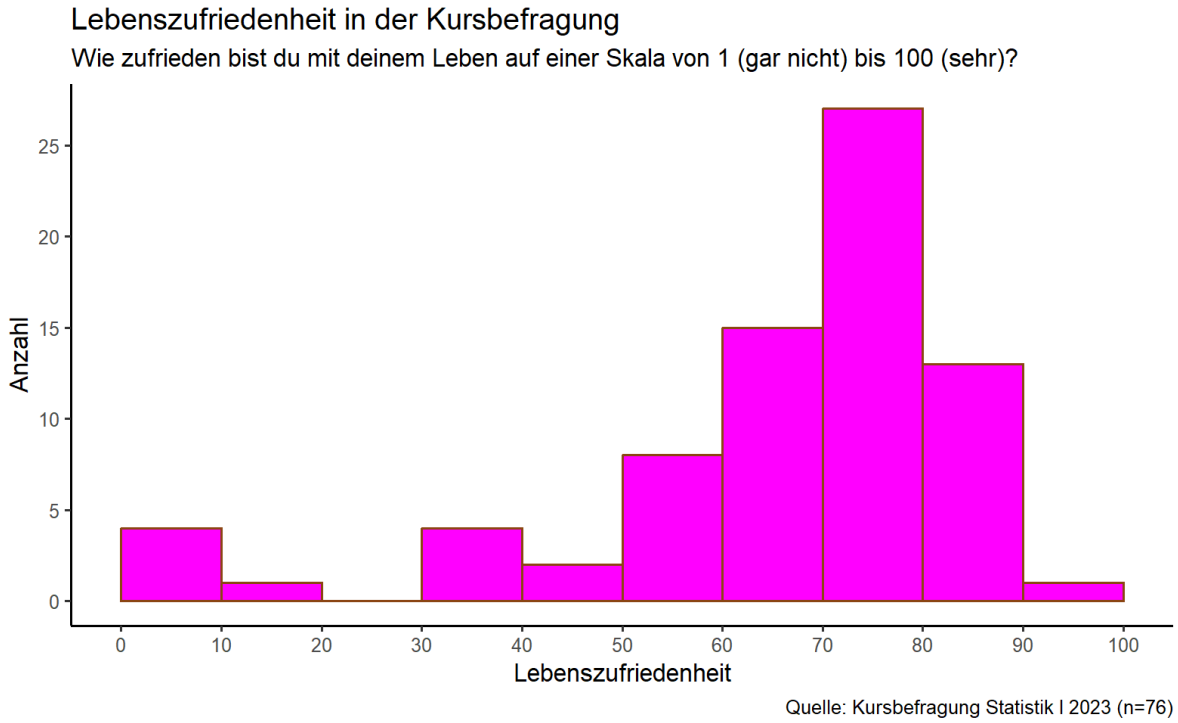
Welche Elemente aus dem vorherigen Befehl müssen modifiziert werden um das Histogramm links zu erzeugen?

```
plot_1 <- ggplot(kursdata_anon_N, aes(x = konsum))+
  geom_bar(colour = "blue", fill = "orange")+
  scale_x_discrete(labels = c("Coop", "Migros"))+
  aes(y = after_stat(count / sum(count)))+
  scale_y_continuous(labels = scales::percent)+
  labs(x = "Supermarkt", y = "Anteil", title = "Konsumpräferenz im Kurs",
       subtitle = "'Welchen Supermarkt bevorzugen Sie?'",
       caption ="Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  theme_classic()
plot_1
```

2.2 Grafische Darstellung: metrischer Variablen (Histogramm)

Dies ist der Befehl zur Erstellung dieses Histogramms:

```
plot_2 <- ggplot(kursdata_anon, aes(x = lezufr))+
  geom_histogram (fill = "magenta", color="chocolate4", breaks = c(0,10,20,30,40,50,60,70,80,90,100))+
  labs(x = "Lebenszufriedenheit",y ="Anzahl",
       title = "Lebenszufriedenheit in der Kursbefragung",
       subtitle = "Wie zufrieden bist du mit deinem Leben auf einer Skala von 1 (gar nicht) bis 100 (sehr)?",
       caption = "Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  scale_x_continuous(breaks = seq(0,100,10))+
  scale_y_continuous(breaks = seq(0,25,5))+
  theme_classic()
plot_2
```

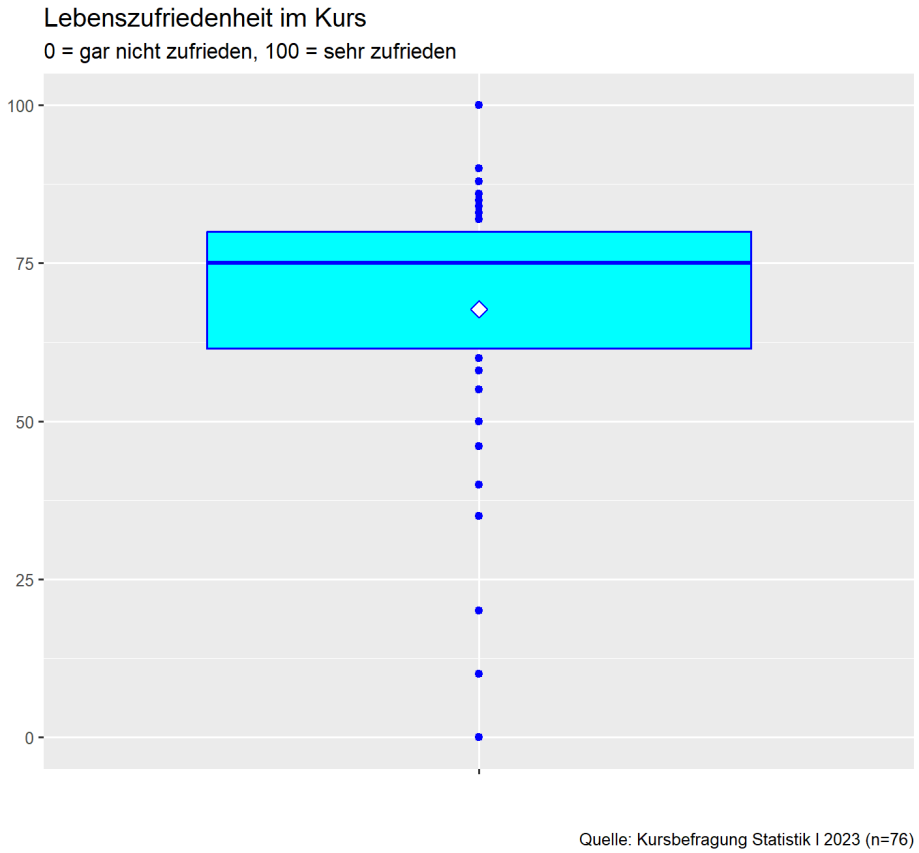


Aufgabe:

Variiert den Befehl:

- **breaks**:
  - Benutzt **seq()** anstatt **c()**
  - Ersetzt durch **bins** und **binwidth**
- Varriert die Spezifikation von **scale_x_continuous()** und **scale_y_continuous()**
- Ändert die Farben
- Variiert den theme

2.3 Grafische Darstellung: metrischer Variablen (Boxplot)



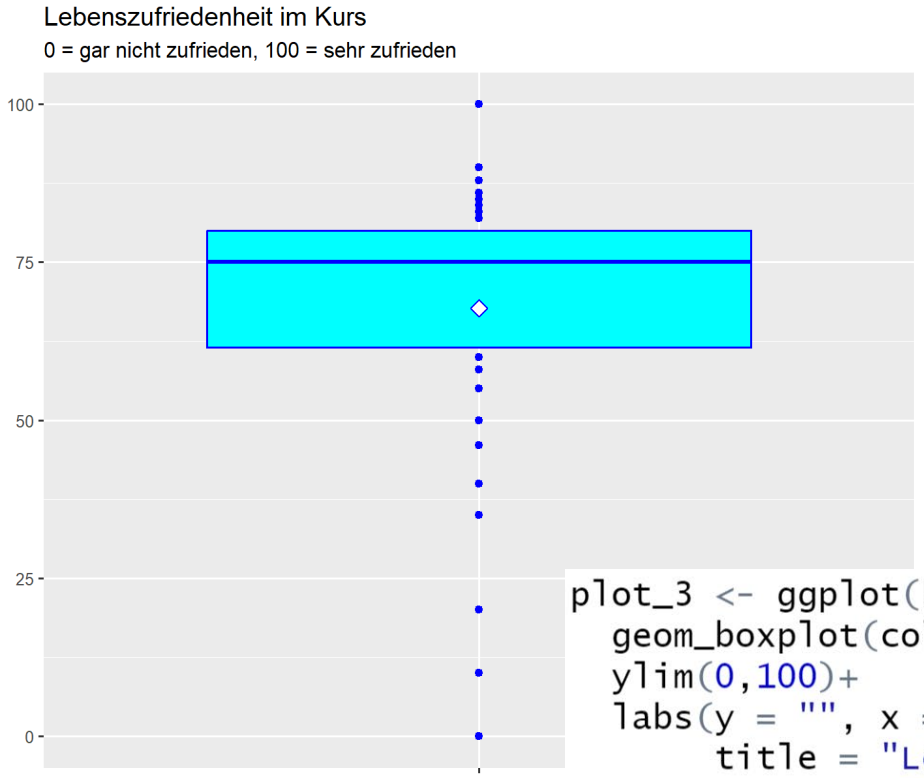
Quelle: Kursbefragung Statistik I 2023 (n=76)

Diskussion:

- Was seht ihr hier?
- Was zeigen die verschiedenen Segmente der Box?
- Welche Befehlselemente müssen geändert werden?
- Passt die Befehlssegmente entsprechend an. Beauftragt ggf. chatgpt, die Mittelwertsraute etc. einzufügen

```
plot_2 <- ggplot(kursdata_anon, aes(x = lezufr))+
  geom_histogram (fill = "magenta", color="chocolate4", breaks = c(0,10,20,30,40,50,60,70,80,90,100))+
  labs(x = "Lebenszufriedenheit",y ="Anzahl",
       title = "Lebenszufriedenheit in der Kursbefragung",
       subtitle = "Wie zufrieden bist du mit deinem Leben auf einer Skala von 1 (gar nicht) bis 100 (sehr)?",
       caption = "Quelle: Kursbefragung Statistik I 2023 (n=76)")+
  scale_x_continuous(breaks = seq(0,100,10))+
  scale_y_continuous(breaks = seq(0,25,5))+
  theme_classic()
plot_2
```

2.3 Grafische Darstellung: metrischer Variablen (Boxplot)

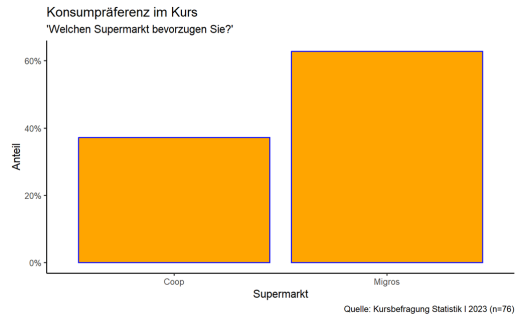


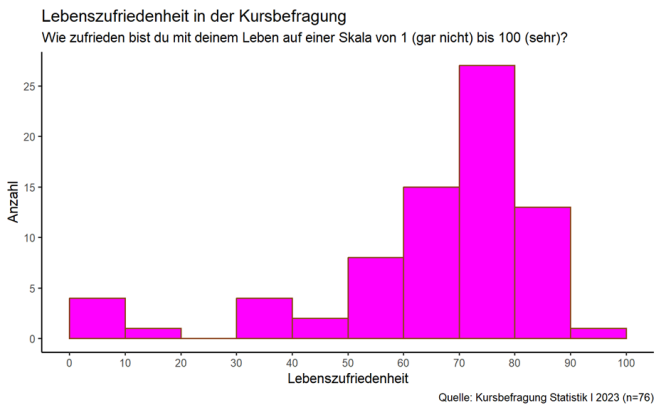
Aufgaben:

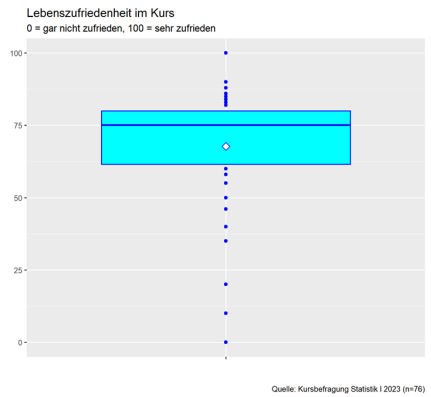
- Was seht ihr hier?
- Was zeigen die verschiedenen Segmente der Box?
- Welche Befehlselemente müssen geändert werden?
- Passt die Befehlssegmente entsprechend an. Beauftragt ggf. chatgpt, die Mittelwertsraute etc. einzufügen

```
plot_3 <- ggplot(kursdata_anon, aes(x = "",y = lezufr))+
  geom_boxplot(colour = "blue", fill = "cyan", coef=0) +
  ylim(0,100)+
  labs(y = "", x = "",
       title = "Lebenszufriedenheit im Kurs",
       subtitle = "0 = gar nicht zufrieden, 100 = sehr zufrieden",
       caption = "Quelle: Kursbefragung Statistik I 2023 (n=76)") +
  stat_summary(fun=mean, geom="point", shape=23, size=3, color="blue", fill="white")
plot_3
```

2.4 Grafische Darstellung: Abschluss







Kommentar: Grafiken mit **ggplot()** zu erstellen ist nicht einfach. Die Struktur gleicht einem Baukasten mit vielen Gestaltungsoptionen und erfordert zu Beginn Geduld, aber auch Kreativität von eurer Seite. Es ist unmöglich, hier abschliessend alle Möglichkeiten aufzuzeigen. Um ein besseres Gefühl dafür zu entwickeln ist es hilfreich, wenn ihr die Befehle an anderen Variablen ausprobiert und verändert.

Export von Grafiken

3.1 Export von Grafiken

Entweder über das Menü („Plots“->“Export“, dann rumprobieren) oder über das Skript:

Format

Name für den exportierten Graph

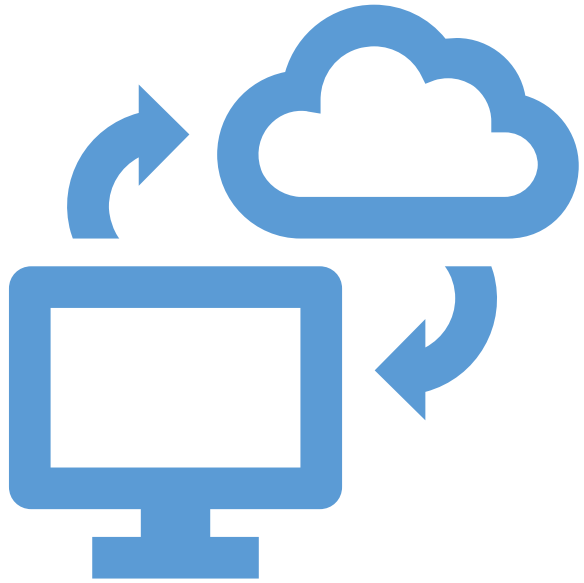
Masse und Auflösung
(ausprobieren und ggf. auf Vorgaben achten)

```
png("boxplot.png", width = 10, height = 12, units = "cm", res = 300)
plot(plot_3)
dev.off()
```

Welcher zuvor erstellte Graph soll exportiert werden?

«device off»: Schliesse den Exportprozess ab

Hypothesentest mit R



4 Hypothesentest mit R

Wir wollen wissen, wie viele Soziologiestudierende aus einem Akadmikerhaushalt stammen. Verwende Sie dazu die Variable **akback** aus den Kursdaten.

- Inspiziere die Variable: Was sind die Kategorien, wie sind sie gelabelt? `attributes (kursdata_anon$akback)`

- Wie ist das Verhältnis? `freq(as_factor(kursdata_anon$akback))`

```
Frequencies

              Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
---------- ------ --------- -------------- --------- --------------
      nein     29     38.16          38.16     38.16          38.16
        ja     47     61.84         100.00     61.84         100.00
      <NA>      0                                0.00         100.00
     Total     76    100.00         100.00    100.00         100.00
```

4 Hypothesentest mit R

Wir wollen wissen, wie viele Soziologiestudierende aus einem Akadmikerhaushalt stammen. Verwende Sie dazu die Variable **akback** aus den Kursdaten.

- Inspiziere die Variable: Was sind die Kategorien, wie sind sie gelabelt?

```
attributes (kursdata_anon$akback)
```

- Wie ist das Verhältnis?

```
freq(as_factor(kursdata_anon$akback))
```

- Ermittle zur Annahme, dass die Herkunftsanteile in der Population ausgeglichen sind, den p-Wert. Interpretiere diesen.

```
prop.test(table(kursdata_anon$akback), p=0.5)
```

```
data:  table(kursdata_anon$akback), null probability 0.5
X-squared = 3.8026, df = 1, p-value = 0.05117
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.2746526 0.5005695
sample estimates:
        p
0.3815789
```

- Binde den p-Wert in einen Test der Hypothese, dass sich die Anteile an Pro-/Contra-Stimmen in der Population unterscheiden, ein.

Übung:

- https://www.suz.uzh.ch/dataforstat/univueb.html

Übung:

- https://www.suz.uzh.ch/dataforstat/univueb.html