Plot 2

Visualisierung von Datensätzen

In diesem Abschnitt sollen alle Graphiken mit *ggplot* und alle Tabellen mit `kable` erstellt werden.

Säuglingssterblichkeit und Wirtschaftsleistung

Nutzen Sie den Datensatz `carData::UN`. In der Hilfe finden Sie Hinweise, was dort gezeigt ist. Beachten Sie, dass für einige Länder Werte fehlen (`NA`); diese Länder sollen nicht berücksichtigt werden.

Wirtschaftsleistung vs. Säuglingssterblichkeit

Erzeugen Sie diese Graphik, indem Sie den nachfolgenden Chunk vervollständigen. Die gezeigten Länder sind Germany, United States, China, India, Afghanistan, Sierra Leone, Japan, Cuba.

Tipp: Sie dürfen (und sollen) weitere Libraries nutzen, wenn diese hilfreich sind.

```
# Ihre Lösung:
library(knitr) # für Darstellung in Tabellen

un = carData::UN # Datensatz laden
kable(un[1:20,]) # Datensatz als Tabelle darstellen, ersten 10 Reihen/Zeilen
```

	region	group	fertility	ppgdp	lifeExpF	pctUrban	infantMortality
Afghanistan	Asia	other	5.968	499.0	49.49	23	124.53500
Albania	Europe	other	1.525	3677.2	80.40	53	16.56100
Algeria	Africa	africa	2.142	4473.0	75.00	67	21.45800
American Samoa	NA	NA	NA	NA	NA	NA	11.29389
Angola	Africa	africa	5.135	4321.9	53.17	59	96.19100
Anguilla	Caribbean	other	2.000	13750.1	81.10	100	NA
Argentina	Latin Amer	other	2.172	9162.1	79.89	93	12.33700
Armenia	Asia	other	1.735	3030.7	77.33	64	24.27200
Aruba	Caribbean	other	1.671	22851.5	77.75	47	14.68700
Australia	Oceania	oecd	1.949	57118.9	84.27	89	4.45500
Austria	Europe	oecd	1.346	45158.8	83.55	68	3.71300
Azerbaijan	Asia	other	2.148	5637.6	73.66	52	37.56600
Bahamas	Caribbean	other	1.877	22461.6	78.85	84	14.13500
Bahrain	Asia	other	2.430	18184.1	76.06	89	6.66300
Bangladesh	Asia	other	2.157	670.4	70.23	29	41.78600
Barbados	Caribbean	other	1.575	14497.3	80.26	45	12.28400
Belarus	Europe	other	1.479	5702.0	76.37	75	6.49400
Belgium	Europe	oecd	1.835	43814.8	82.81	97	3.73900
Belize	Latin Amer	other	2.679	4495.8	77.81	53	16.20000
Benin	Africa	africa	5.078	741.1	58.66	42	76.67400

```r
# Enfernen der Zeilen, in denen NA Werte auftreten
un_cleaned = un[complete.cases(un), ] #


# Vergleichen, wie viele NA Werte herausgefallen sind
print(nrow(un))
```

```
## [1] 213
```

```r
print(nrow(un_cleaned))
```

```
## [1] 193
```

```r
library(ggplot2) # für Grafiken
library(ggrepel) # für die Ländernamen


# in der Grafik beschriftete Länder als Vektor definieren
markierte_laender = c(
  "Germany", "United States", "China", "India",
  "Afghanistan", "Sierra Leone", "Japan", "Cuba"
)

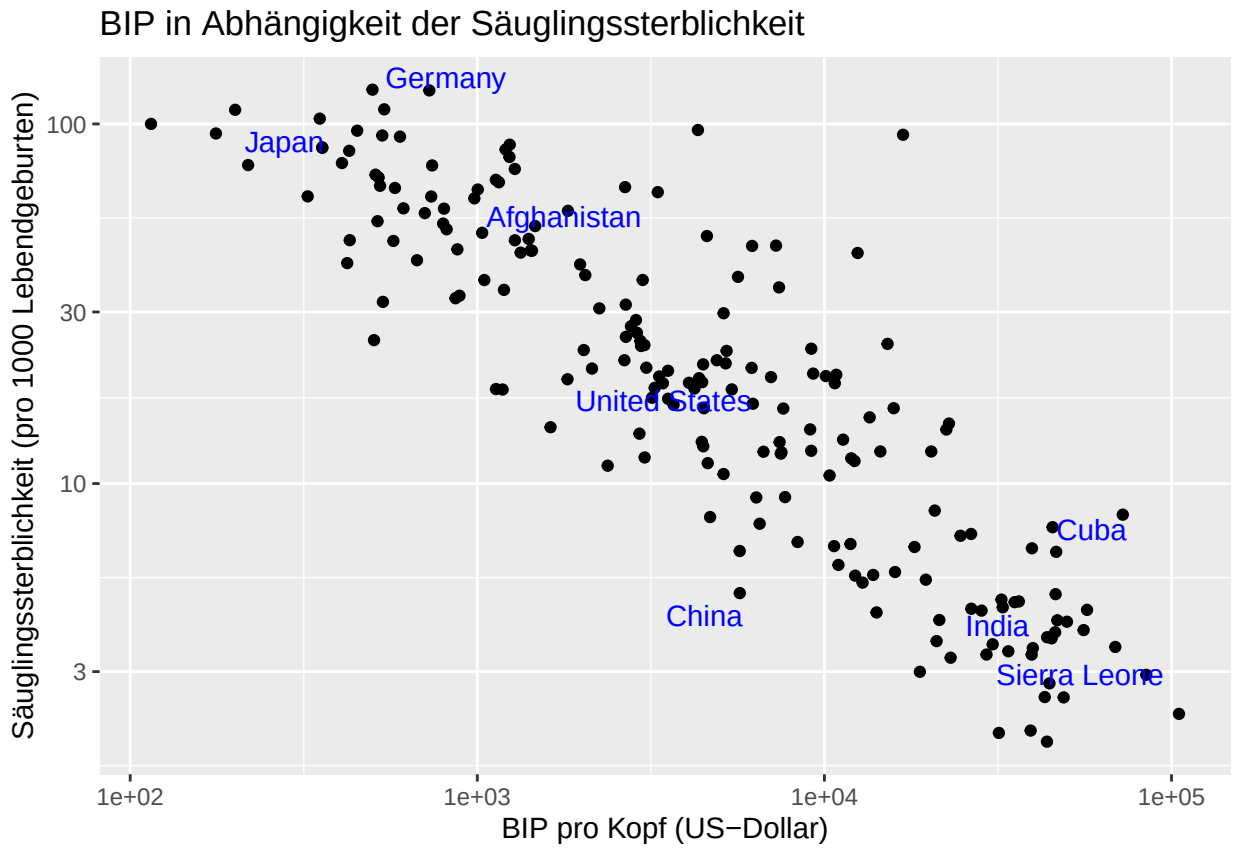
# Rückgabe df in dem nur Zeilen der markierten Länder vorkommen
# alle Spalten werden übernommen
labeldaten = un_cleaned[rownames(un_cleaned) %in% markierte_laender, ]

# gibt Liste mit TRUE und FALSE Werten zurück
print(rownames(un_cleaned) %in% markierte_laender)
```

```
##   [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE
```

```r
# eigentlichen Plot erzeugen
# df übergeben, x und y definieren
ggplot(un_cleaned, aes(x = ppgdp, y = infantMortality)) +
  geom_point() + # für Scatterplot/Punktdiagramm
  # Text in Grafik einfügen, "labeldaten" enthält nur Reihen der markierten Länder
  geom_text_repel(data = labeldaten, color="blue", aes(label = markierte_laender)) +
  scale_x_log10() + # log x Achse
  scale_y_log10() + # log y Achse
```

```
  labs(
    title= "BIP in Abhängigkeit der Säuglingssterblichkeit",
    x = "BIP pro Kopf (US-Dollar)",
    y = "Säuglingssterblichkeit (pro 1000 Lebendgeburten)"
  )
```



Säuglingssterblichkeit im Verhältnis zur Wirtschaftsleistung

Geben Sie diejenigen 10 Länder als Tabelle im Notebook aus, die das größte Verhältnis r von Säuglingssterblichkeit zu Wirtschaftsleistung haben. Dabei ist r die Säuglingssterblichkeit (pro 1000 Lebendgeburten) je 1000 US-Dollar BIP pro Kopf. Die Liste soll nach r absteigend sortiert sein und den Ländernamen und r enthalten.

Vervollständigen Sie diesen Chunk:

```
# Ihre Lösung:
library(dplyr) # für mutate
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Funktion, die Relation berechnet
func_relation = function(ppgdp, infantMortality) ppgdp / infantMortality

# neue Spalte zum df hinzufügen mit mutate, die Relationswerte enthält
un_r = un_cleaned |> mutate(relation = func_relation(ppgdp, infantMortality))

werte = data.frame(
  Land = rownames(un_r),
  Relation = un_r$relation) # df mit "Land" und "Relation" erstellen

top10 = werte |>
  # sortierte Liste der 10 größten Werte zurückgeben
  slice_max(order_by = Relation, n=10)

kable(top10)
```

Land	Relation
Luxembourg	45913.24
Norway	28771.67
Singapore	22851.30
Switzerland	19607.23
Sweden	19224.14
Iceland	19094.80
Japan	16924.64
Finland	15990.55
Hong Kong	15707.65
Denmark	14264.23

Geben Sie nun – wie eben – diejenigen 10 Länder als Tabelle aus, die das **kleinste** Verhältnis r haben. Die Liste soll nach r absteigend sortiert sein.

Vervollständigen Sie diesen Chunk:

```r
# Ihre Lösung:
bottom10 = werte |>
  # diesmal die 10 kleinsten Relation Werte zurückgeben
  slice_min(order_by = Relation, n=10) |>
  # arrange in descending order, Erabsteigend nach Relation sortieren
  arrange(desc(Relation))

kable(bottom10)
```

Land	Relation
Guinea-Bissau	4.911763
Central African Republic	4.706570
Niger	4.168026
Malawi	4.152917
Afghanistan	4.006906
Sierra Leone	3.399414
Liberia	2.844391
Burundi	1.877066
Democratic Republic of the Congo	1.832348

Land	Relation
Somalia	1.147805

Kieferwachstum bei Kindern

Nutzen Sie den Datensatz `nlme::Orthodont`. In der Hilfe finden Sie Hinweise, was dort gezeigt ist.

Überblick über Verlauf bei allen Kindern

Plotten Sie im folgenden Chunk den Verlauf des Abstands zwischen Hypophyse und Fossa pterygopalatina (y-Achse) in Abhängigkeit vom Alter (x-Achse). Dies soll in einem Diagramm mit Unterdiagrammen erfolgen: ein Unterdiagramm zeigt den Verlauf für je ein Kind (Subject) und dessen Geschlecht (Male oder Female).

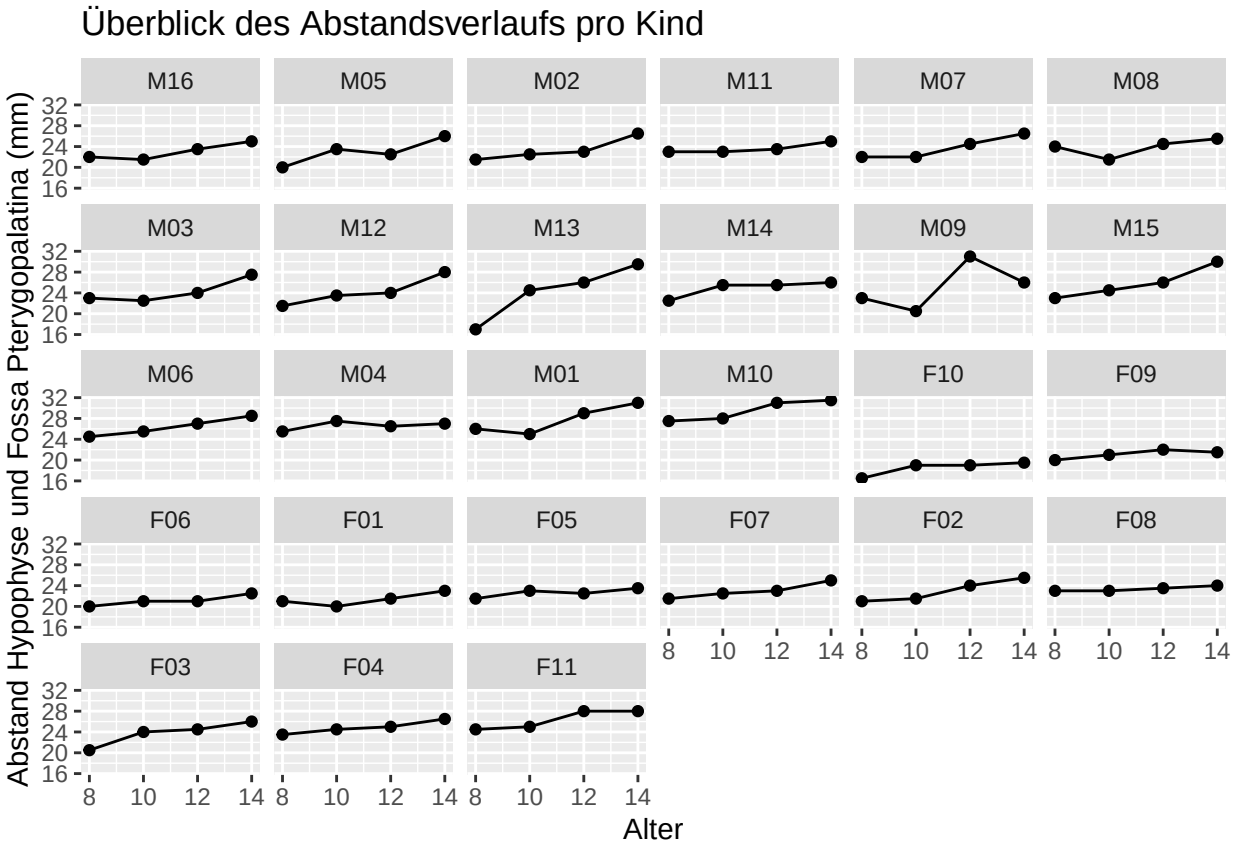
```
# Ihre Lösung:
orthodont = nlme::Orthodont # Datensatz einlesen

kable(orthodont[1:10, ]) # Datensatz als Tabelle anzeigen, die 10 ersten Zeilen
```

distance	age	Subject	Sex
26.0	8	M01	Male
25.0	10	M01	Male
29.0	12	M01	Male
31.0	14	M01	Male
21.5	8	M02	Male
22.5	10	M02	Male
23.0	12	M02	Male
26.5	14	M02	Male
23.0	8	M03	Male
22.5	10	M03	Male

```
ggplot(orthodont) +
  geom_line(aes(x=age, y=distance)) + # Liniendiagramm
  geom_point(aes(x=age, y=distance)) + # Punkte hinzufügen

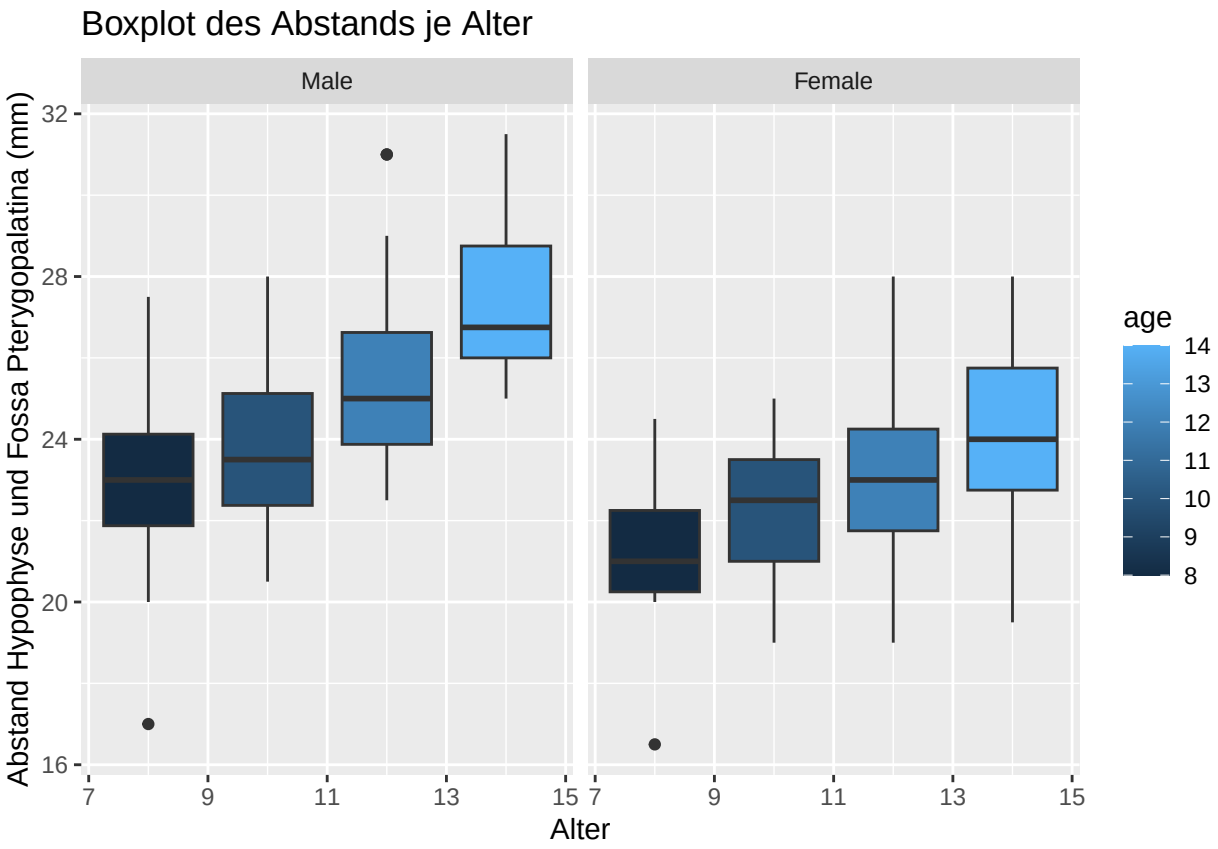
  # für jedes Subjekt ein Diagramm
  # M und F im Namen zeigt bereits, um welches Geschlecht es sich handelt
  facet_wrap(~ Subject) +
  labs(title="Überblick des Abstandsverlaufs pro Kind",
       x="Alter", y="Abstand Hypophyse und Fossa Pterygopalatina (mm)")
```



Boxplots des Abstands je Alter

Erzeugen Sie ein Diagramm, das in zwei Unterdiagrammen für die Jungen und die Mädchen Boxplots erstellt. Die Boxplots geben die Verteilung des Abstands je Alter an. In Anlehnung an das obige Diagramm sollen die Boxplots vertikal ausgerichtet sein.

```
# Ihre Lösung:
ggplot(orthodont) +
  # Boxplots für jedes Alter, fill für Farbe
  geom_boxplot(aes(x=age, y=distance, group=age, fill=age)) +
  facet_wrap(~ Sex) + # zwei Diagramme, eines für Male eines für Female
  labs(title="Boxplot des Abstands je Alter",
       x="Alter", y="Abstand Hypophyse und Fossa Pterygopalatina (mm)")
```


Boxplot des Abstands je Alter
Male
Female
Abstand Hypophyse und Fossa Pterygopalatina (mm)
32
28
24
20
16
7
9
11
13
15
Alter
age
14
13
12
11
10
9
8