Sitzung 2: Erste Schritte (Notizen)

Jonas Schulte

2023-04-10

Benötigte Packages

Zu Beginn eines jeden R-Markdown Dokument, binden wir mit der Funktion `library()` die Pakete ein, die wir für unsere Analyse benötigen. Wir benötigen das Package `gapminder`, das uns einen Datensatz bereitstellt.

```
library(gapminder)
# falls nicht installiert, den folgenden Befehl ausführen:
# install.packages("gapminder")
```

Datensatz

Nachdem wir das Package `gapminder` mit dem `library()`-Befehl eingebunden haben, können wir den Datensatz aufrufen, in dem wir den Namen des Datensatzes in der Console oder in einem Chunk ausführen. Der Datensatz trägt - wie das Paket - den Objektnamen `gapminder`.

```
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

Gerade bei großen Datensätzen wollen wir uns nicht immer den gesamten Datensatz anschauen. Mit den Funktionen `head()` und `tail()` können wir uns nur den Anfang und das Ende des Datensatzes ausgeben lassen.

```
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

```
tail(gapminder)
```

```
## # A tibble: 6 x 6
##   country  continent  year lifeExp      pop gdpPercap
##   <fct>    <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Zimbabwe Africa     1982    60.4  7636524      789.
## 2 Zimbabwe Africa     1987    62.4  9216418      706.
## 3 Zimbabwe Africa     1992    60.4 10704340      693.
## 4 Zimbabwe Africa     1997    46.8 11404948      792.
## 5 Zimbabwe Africa     2002    40.0 11926563      672.
## 6 Zimbabwe Africa     2007    43.5 12311143      470.
```

Übersichtsstatistiken

Die Funktion `nrow()` zählt die Zeilen eines Datensatzes.

```
nrow(gapminder)
```

```
## [1] 1704
```

Häufig wollen wir Funktionen nicht auf den ganzen Datensatz anwenden, sondern auf eine Variable bzw. eine Spalte des Datensatzes. Eine einzelne Spalte können wir adressieren, in dem wir nach dem Datensatznamen ein Dollarzeichen, gefolgt vom Spaltennamen schreiben und ausführen.

```
gapminder$country
```

Die Funktion `unique()` entfernt Duplikate in einem Vektor. Sie ist hilfreich, um zu prüfen, welche Länder in unserem Datensatz enthalten sind.

```
unique(gapminder$country)
```

```
##   [1] Afghanistan              Albania                  Algeria
##   [4] Angola                   Argentina                Australia
##   [7] Austria                  Bahrain                  Bangladesh
##  [10] Belgium                  Benin                    Bolivia
##  [13] Bosnia and Herzegovina   Botswana                 Brazil
##  [16] Bulgaria                 Burkina Faso             Burundi
```

```
##  [19] Cambodia                 Cameroon                 Canada
##  [22] Central African Republic Chad                     Chile
##  [25] China                    Colombia                 Comoros
##  [28] Congo, Dem. Rep.         Congo, Rep.              Costa Rica
##  [31] Cote d'Ivoire            Croatia                  Cuba
##  [34] Czech Republic           Denmark                  Djibouti
##  [37] Dominican Republic       Ecuador                  Egypt
##  [40] El Salvador              Equatorial Guinea        Eritrea
##  [43] Ethiopia                 Finland                  France
##  [46] Gabon                    Gambia                   Germany
##  [49] Ghana                    Greece                   Guatemala
##  [52] Guinea                   Guinea-Bissau            Haiti
##  [55] Honduras                 Hong Kong, China         Hungary
##  [58] Iceland                  India                    Indonesia
##  [61] Iran                     Iraq                     Ireland
##  [64] Israel                   Italy                    Jamaica
##  [67] Japan                    Jordan                   Kenya
##  [70] Korea, Dem. Rep.         Korea, Rep.              Kuwait
##  [73] Lebanon                  Lesotho                  Liberia
##  [76] Libya                    Madagascar               Malawi
##  [79] Malaysia                 Mali                     Mauritania
##  [82] Mauritius                Mexico                   Mongolia
##  [85] Montenegro               Morocco                  Mozambique
##  [88] Myanmar                  Namibia                  Nepal
##  [91] Netherlands              New Zealand              Nicaragua
##  [94] Niger                    Nigeria                  Norway
##  [97] Oman                     Pakistan                 Panama
## [100] Paraguay                 Peru                     Philippines
## [103] Poland                   Portugal                 Puerto Rico
## [106] Reunion                  Romania                  Rwanda
## [109] Sao Tome and Principe    Saudi Arabia             Senegal
## [112] Serbia                   Sierra Leone             Singapore
## [115] Slovak Republic          Slovenia                 Somalia
## [118] South Africa             Spain                    Sri Lanka
## [121] Sudan                    Swaziland                Sweden
## [124] Switzerland              Syria                    Taiwan
## [127] Tanzania                 Thailand                 Togo
## [130] Trinidad and Tobago      Tunisia                  Turkey
## [133] Uganda                   United Kingdom           United States
## [136] Uruguay                  Venezuela                Vietnam
## [139] West Bank and Gaza       Yemen, Rep.              Zambia
## [142] Zimbabwe
## 142 Levels: Afghanistan Albania Algeria Angola Argentina Australia ... Zimbabwe
```

Viele der statistischen Operation, wie Durchschnitt, Median oder Standardabweichungen haben in R relativ eindeutige Namen. So können wir bsp. mit `mean()` das arithmetische Mittel, mit `sd()` die Standardabweichung und mit `median()` den Median berechnen. Im folgenden berechnen wir einige Übersichtsstatistiken für die Lebenserwartung.

```
mean(gapminder$lifeExp)
```

```
## [1] 59.47444
```

```
sd(gapminder$lifeExp)
```

```
## [1] 12.91711
```

```
median(gapminder$lifeExp)
```

```
## [1] 60.7125
```

```
hist(gapminder$lifeExp)
```


```
max(gapminder$lifeExp)
```

```
## [1] 82.603
```

```
gapminder[which(gapminder$lifeExp == max(gapminder$lifeExp)),]
```

```
## # A tibble: 1 x 6
##   country continent  year lifeExp       pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Japan   Asia       2007    82.6 127467972    31656.
```

```
min(gapminder$lifeExp)
```

```
## [1] 23.599
```

```
gapminder[which(gapminder$lifeExp == min(gapminder$lifeExp)),]
```

```
## # A tibble: 1 x 6
##   country continent  year lifeExp     pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>   <int>     <dbl>
## 1 Rwanda  Africa     1992    23.6 7290203      737.
```

Bist du dir unsicher, was eine Funktion genau tut, kannst du ein Fragenzeichen gefolgt vom Funktionsnamen (ohne Klammern) in die Console eingeben. Es öffnet sich dann auf der rechten Seite eine Hilfeseite.

```
?mean
```

Übungsaufgabe III

1. Berechne das arithmetische Mittel, die Standardabweichung und den Median des BIP pro Kopfs und interpretiere die Ergebnisse.

```
# Hier den Code schreiben
```

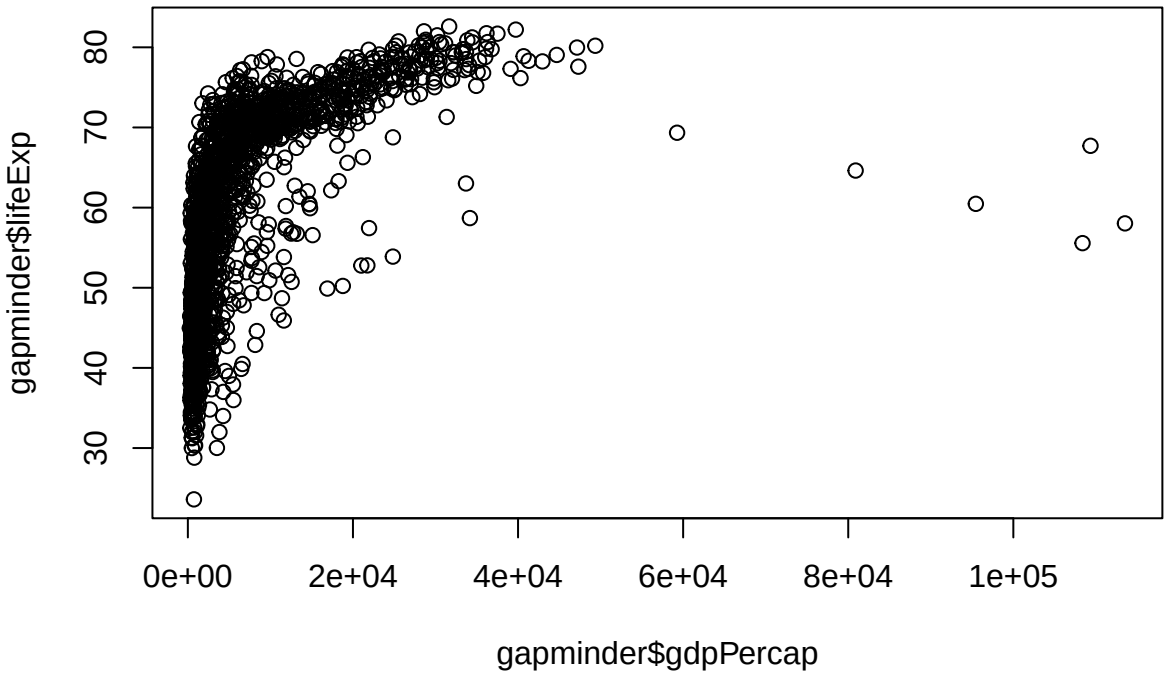
2. Was ist der höchste, was der tiefste Wert für das BIP pro Kopf im Datensatz? Bonus: In welchem Land in welchem Jahr wurde das Maximum bzw. Minimum gemessen?

3. Erstelle ein Histogram und interpretiere die Verteilung des BIP pro Kopfs

Bivariater Zusammenhang

Wie hängen das BIP pro Kopf und die Lebenserwartung zusammen? Wir vermuten, dass reichere Länder auch eine höhere Lebenserwartung haben. Um diese These zu überprüfen, ist es hilfreich, die Daten zunächst zu plotten und den Zusammenhang grafisch zu betrachten. Für ein einfaches Streudiagramm können wir die Funktion plot() verwenden. Innerhalb der Klammer müssen wir angeben, welche Daten auf der x- und welche auf der y-Achse abgebildet werden sollen. Später werden wir noch weitere Funktionen kennenlernen, die ansprechendere Grafiken erstellen können.

```
plot(x = gapminder$gdpPercap, y = gapminder$lifeExp)
```



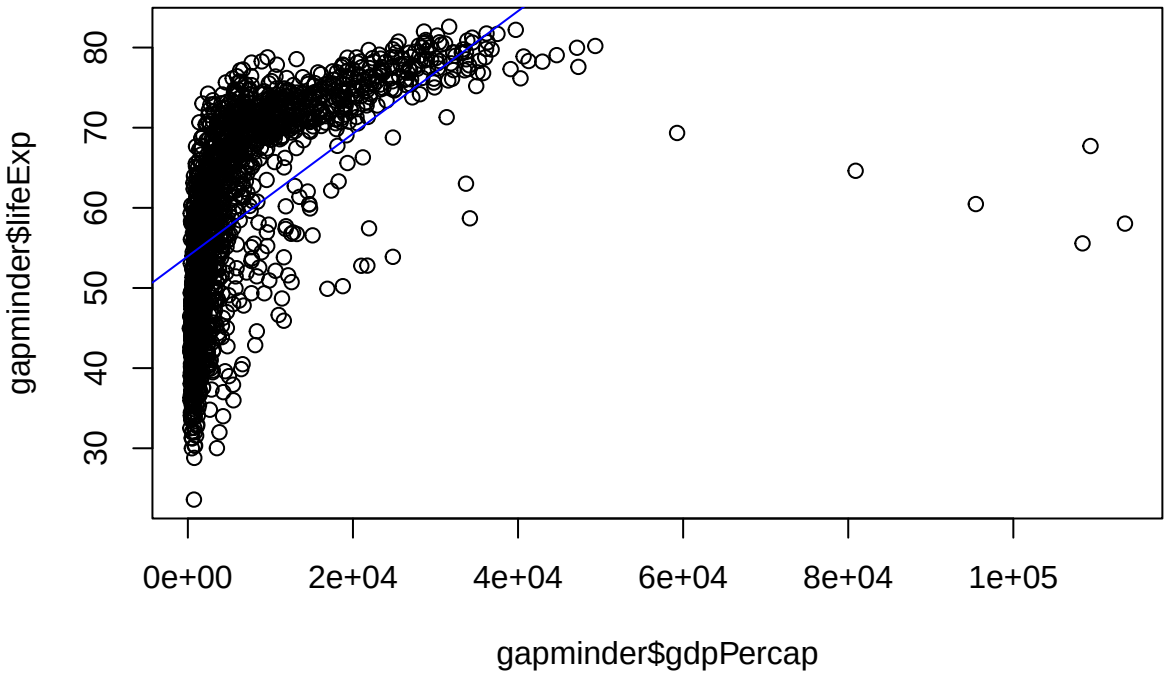
Der Korrelationskoeffizient ist ein Maß für den Grad und die Richtung des Zusammenhangs zwischen zwei Variablen misst. Werte reichen von -1 (perfekt negative Korrelation) über 0 (keine Korrelation) bis +1 (perfekt positive Korrelation). In R können wir die Korrelation mit der Funktion `cor()` bestimmen.

```
cor(gapminder$gdpPercap, gapminder$lifeExp)
```

```
## [1] 0.5837062
```

Sowohl das Streudiagramm, als auch der Korrelationskoeffizient bestätigen unsere Vermutung, dass das BIP pro Kopf und die Lebenserwartung positiv zusammenhängen. Wir können versuchen den Zusammenhang in der einfachsten Art und Weise zu modellieren, in dem wir annehmen, dass der Zusammenhang linear ist. Grafisch bedeutet das, dass wir eine Gerade durch unsere Datenpunkte legen, die den Zusammenhang "best möglichst" modeliert.

```
plot(x = gapminder$gdpPercap, y = gapminder$lifeExp)
abline(lm(gapminder$lifeExp ~ gapminder$gdpPercap), col = "blue")
```



Eine Gerade, die den Zusammenhang besonders gut annähernd beschreibt, kann mit Hilfe der linearen Regression bestimmt werden (weitere Details dazu werden in der Vorlesung und im Tutorium in den kommenden Wochen behandelt). Mit der Funktion lm() können wir den Achsenabschnitt und die Steigung dieser Geraden bestimmen.

```
lm(formula = gapminder$lifeExp ~ gapminder$gdpPercap)
```

```
##
## Call:
## lm(formula = gapminder$lifeExp ~ gapminder$gdpPercap)
##
## Coefficients:
##         (Intercept)  gapminder$gdpPercap
##           5.396e+01            7.649e-04
```

Übungsaufgabe IV

1. Berechne den Korrelationskoeffizienten zwischen dem BIP pro Kopf und der Lebenserwartung und interpretiere das Ergebnis.
2. Berechne mit Hilfe des `lm()`-Befehls den Achsenabschnitt und die Steigung der Geraden.
3. Interpretiere die Steigung der Geraden. Tipp: Zeichne ein Steigungsdreieck und multipliziere die Werte mit 1000.