WZB

Wissenschaftszentrum Berlin
für Sozialforschung

Einführung in die Quantitative Datenanalyse

Sitzung 4: Lineare Regression II

Proseminar an der Freien Universität Berlin
22.05.2017 - Marcus Spittler

Inhalt der 4. Sitzung

- **Lineare Regression - Fortsetzung**
  - Grundlegende Begriffe, wie *Effektstärke* und *Richtung des Zusammenhangs*
  - Interpretation
  - Grundlagen der Modellbildung
- Drei **tidyverse**-Verben

Wiederholung zur linearen Regression

- Regression: Rückführung einer **abhängigen** Variablen auf eine oder mehrere **unabhängige Variablen**
- Verwendet man bei **Zusammenhangshypothesen**
- Die abhängige Variable muss mindestens **intervallskaliert** sein
- Unterstellt wird ein **linearer** Zusammenhang

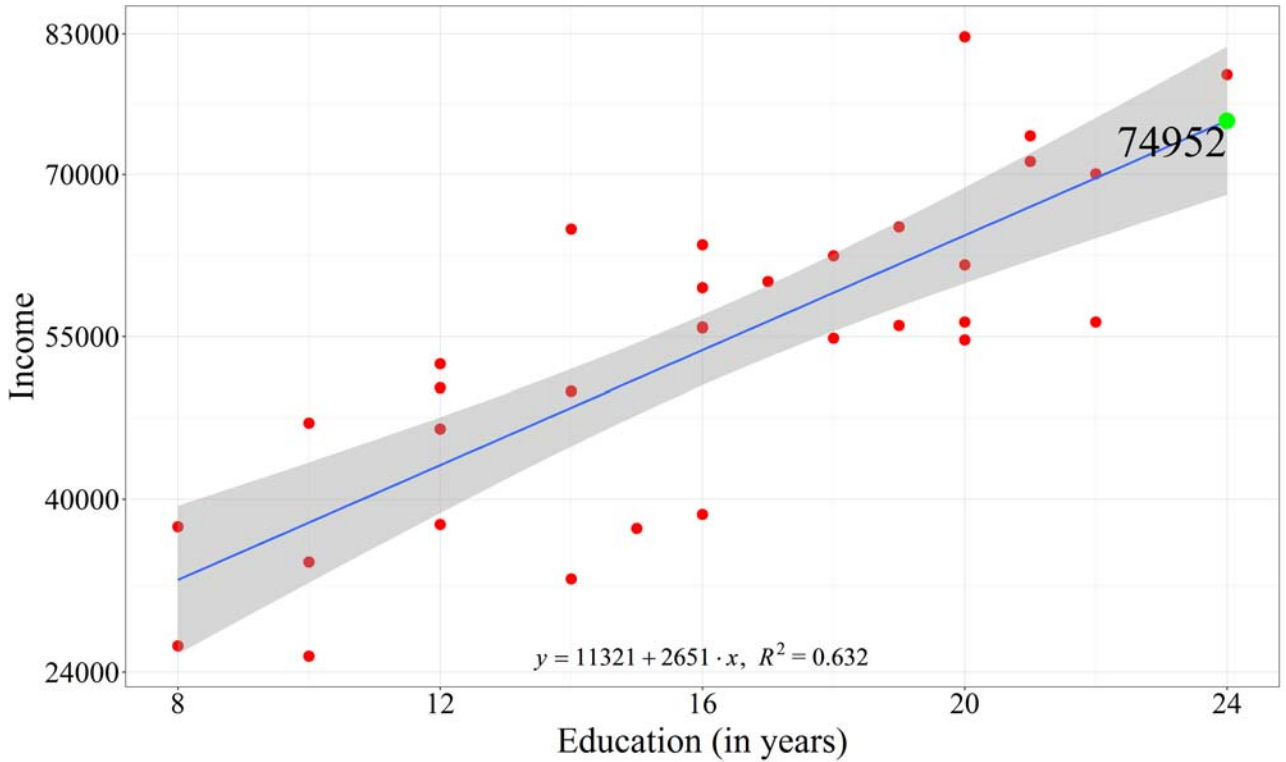
Regressionsgerade

- Die Regressionsanalyse versucht eine **Gerade** zu ermitteln, die den linearen Zusammenhang zwischen der unabhängigen Variablen **X** und der abhängigen Variablen **Y** beschreibt. Auf dieser Geraden liegen **nicht** notwendigerweise die Punkte mit den tatsächlichen, beobachteten Werten von **Y**, sondern die durch die Regressionsgleichung **geschätzten**. Die Werte auf dieser **Geraden** lassen sich durch Einsetzen der X-Werte wie folgt ermitteln.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- Wie etwa im **Riverview** Beispiel für eine Person die 24 Bildungsjahre hat:

$$74952 = 11321 + 2651 * 24$$


83000
70000
55000
40000
24000
Income
74952
$y = 11321 + 2651 \cdot x,\ R^2 = 0.632$
8
12
16
20
24
Education (in years)

Unstandardisierter Steigungskoeffizient

- β_1 (sprich: beta 1) ist der **unstandardisierte Steigungskoeffizient** bei der Regression von *Y* auf *X*, der die **Steigung der Regressionsgeraden** angibt. Er ergibt sich, wenn wir die **Kovarianz** von X und Y durch die **Varianz** von X dividieren.

- Anders ausgedrückt: Der **unstandardisierte Steigungskoeffizient** gibt an, um wieviel sich Y *im Mittel ändert*, wenn wir den Wert von X um eine Einheit verändern. Im Beispiel beträgt β_1 2651.

- Das Vorzeichen von β_1 gibt an, ob es sich um einen **positiven** oder **negativen** Zusammenhang handelt.

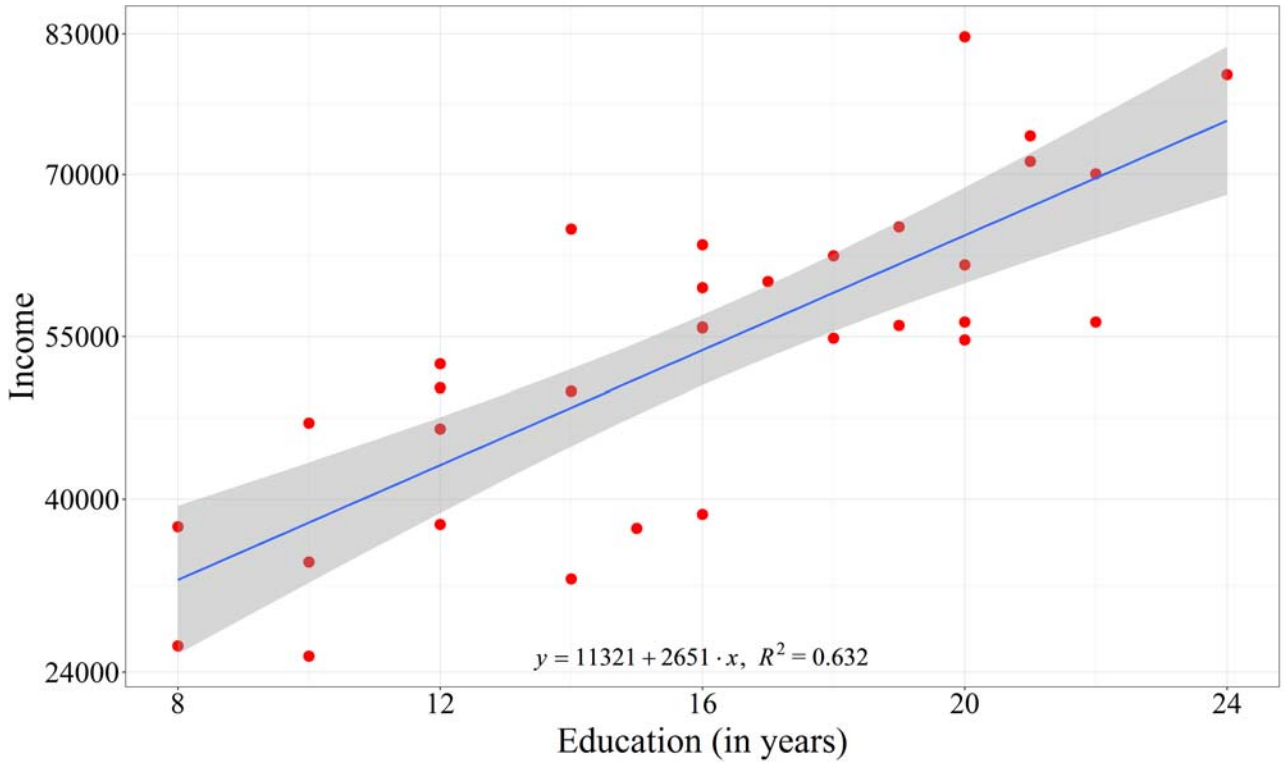
- **Interpretation**: Das Einkommen erhöht sich im Mittel um 2651 Euro, wenn der Befragte **1 Jahr** länger in Ausbildung war.

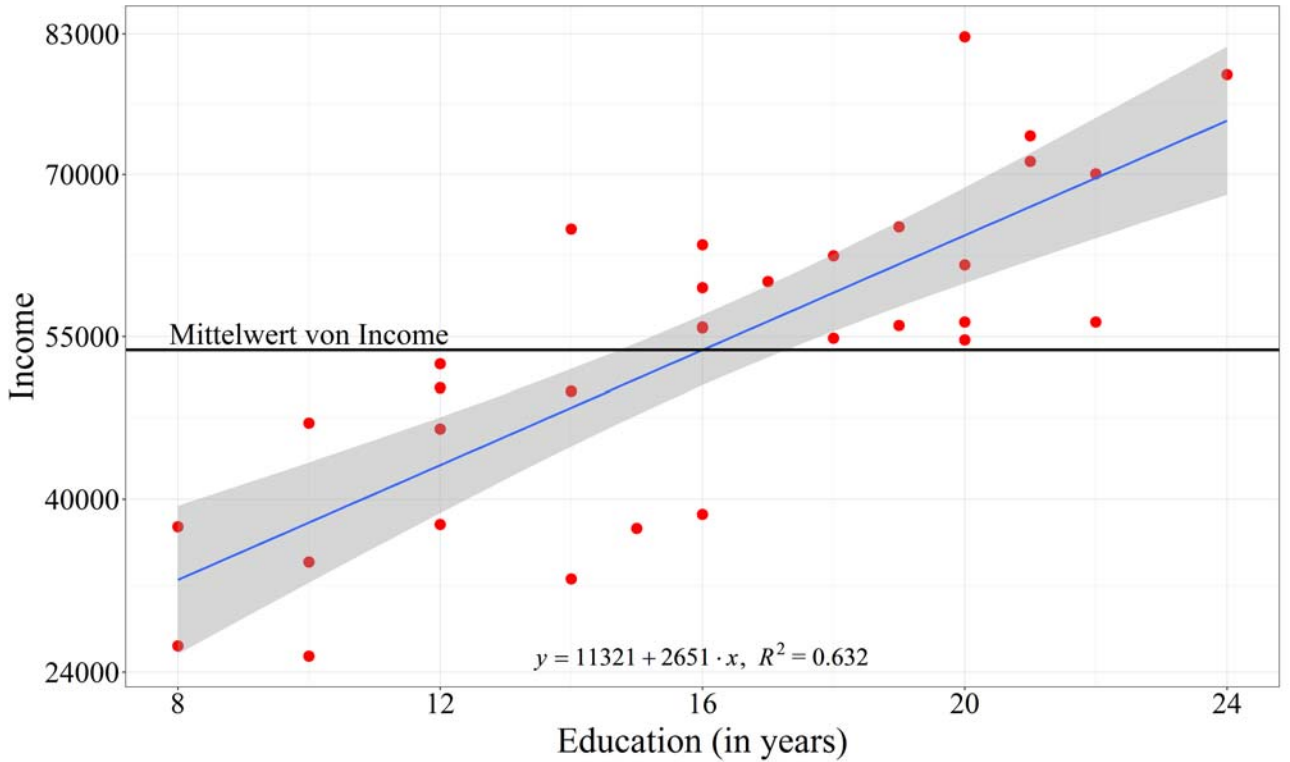
Residuen

- In der Realität wird man annehmen, dass die nach der **Geradengleichung** vorherzusagenden Y-Werte $\hat{y}$ nicht exakt mit den tatsächlichen *Y-Werten* y_i übereinstimmen.
- Diese Abweichungen zwischen tatsächlichen und nach der Regressionsgleichung vorherzusagenden Y-Werten, die **Vorhersagefehler** also, bezeichnen wir als Residuen e_i der konkreten Stichprobe.

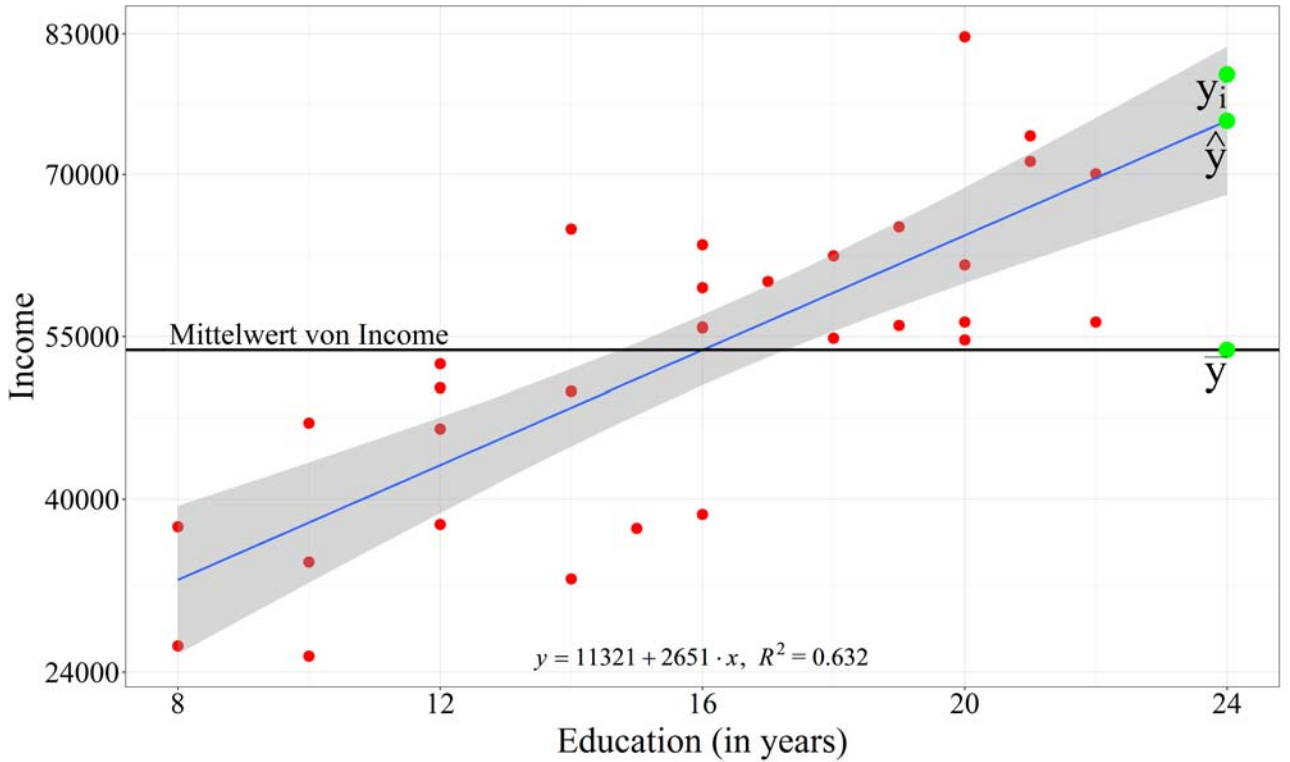
$$e_i = y_i - \hat{y}$$

- Wie ermittelt man aber nun die **Regressionsgerade**, die die Beziehung zwischen X und Y am besten wiedergibt?

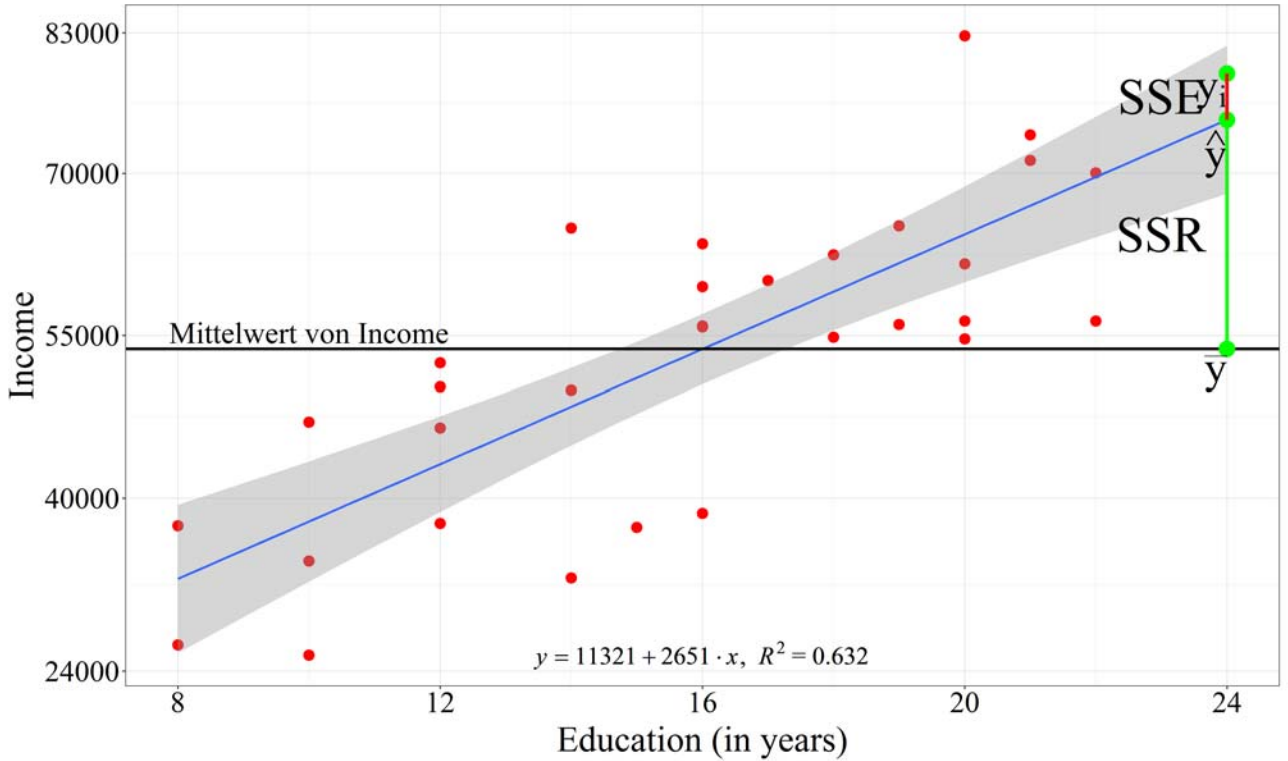

83000
70000
55000
40000
24000
Income
8
12
16
20
24
Education (in years)
$y = 11321 + 2651 \cdot x,\ R^2 = 0.632$



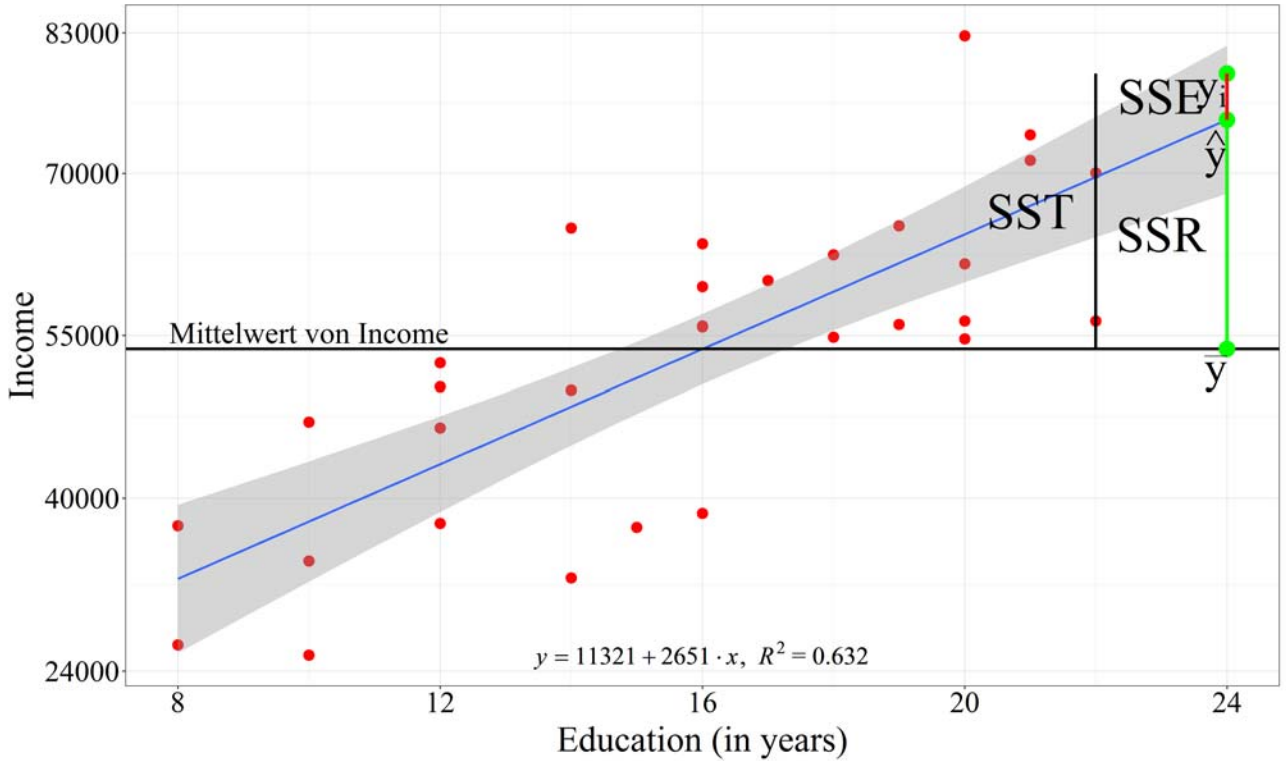
- Der Mittelwert von Income $\bar{y}$ ist unser **Erwartungswert**, wenn wir nichts über die Bildungsdauer einer Person wissen.



- $\bar{y}$ ist das mittlere Einkommen. y_i ist der Wert der Person mit 24 Jahren Bildungsdauer. $\hat{y}$ ist unser vorhergesagter Wert.



- Das heißt: Eine bestimmte Abweichung vom Mittelwert haben wir vorhergesagt (**Regression Sum of Squares = SSR**) und einen anderen Teil nicht (**Summed Squared of Errors = SSE**).



- **SSE** und **SSR** bilden zusammen die **Total Sum of Squares = SST**

Methode der kleinsten Quadrate

- Wie soll nun die Gerade ermittelt werden, die die Beziehung zwischen *Y* und *X* am besten wiedergibt?

- Anforderung: Die gesuchte Gerade soll die **Punktewolke** möglichst *gut* beschreiben. Eine zweite Forderung ist: Die Residuen sollen in der Summe **null** ergeben. Die **quadrierte Summe** der Residuen soll minimal sein. Jedoch erfüllen *sehr viele mögliche Geraden* diese Kriteren.

- Lösung: Die **Methode der kleinsten Quadrate** (Kleinst-Quadrat-Verfahren, KQ-Verfahren, **Ordinary Least Squares-Verfahren**, OLS). Mit ihr wird die Summe der **quadrierten Residuen** minimiert.

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

Drei Eigenschaften der **Methode der kleinsten Quadrate**

- Die **Summe und der Mittelwert der Residuen** ist null. Die Regressionsgerade ist also eine ausgleichende Gerade, da die Abweichungen zwischen Vorhersagewerten und tatsächlichen Werten sich aufheben.
- Die Regressionsgerade verläuft durch den Schwerpunkt der Punktewolke (Genauer durch die Mittelwerte $\bar{x}$ und $\bar{y}$).
- X-Variable und (nach dem KQ-Verfahren geschätzte) Residualvariable sind unkorreliert.
- Dynamische Beispiele:
  - https://www.geogebra.org/m/xC6zq7Zv
  - http://www.miabella-llc.com/demo.html

Gleichung der **Regressionsgeraden**

- Man kann zeigen, dass die Regressionsgerade dann die kleinste Residuenquadratsumme hat, wenn gilt:

$$\hat{\beta}_1 = \frac{Kovarianz}{Varianz} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{cov(x,y)}{s_x^2}$$

- Und für β_0:

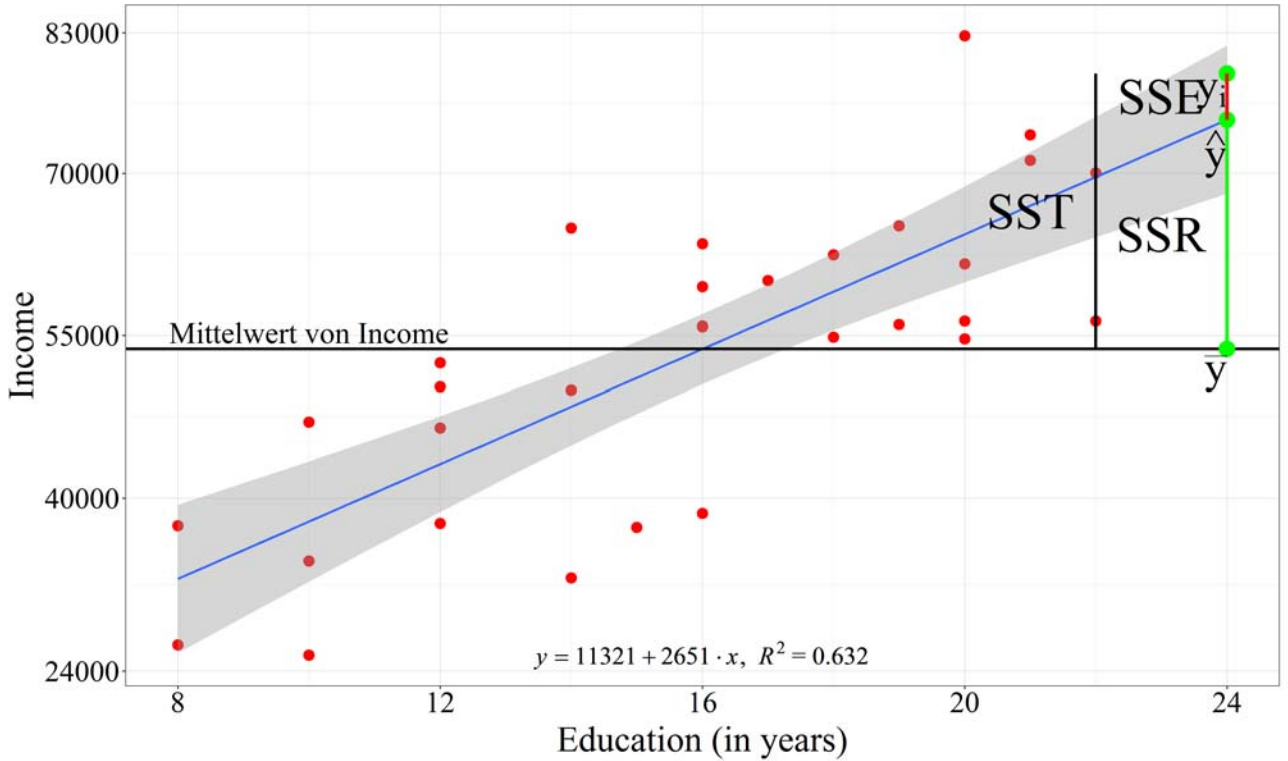
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Güte der Regression

- Bis jetzt haben wir vor allem β_1, die **Effektstärke**, betrachtet. Daneben wollen wir wissen, **wie gut** die Regressionsgerade die Punktewolke beschreibt.
- Ist die Streuung um die Regressionsgerade groß, so ist unser Regressionsmodell nur schlecht geeignet, die Daten zu beschreiben. Liegen dagegen im Extremfall alle Punkte auf der Geraden, sind also alle Residuen null, so haben wir es mit einem **perfekten linearen Zusammenhang** zwischen X und Y zu tun. Nur im Falle einer nicht zu großen Streuung um die Regressionsgerade werden wir guten Gewissens Vorhersagewerte verwenden wollen.

Güte der **Regression**

- Begriffe:
  - **SST** = Total Sum of Squares
  - **SSR** = Regression Sum of Squares
  - **SSE** = Error Sum Squares
- Es gilt: **SST = SSR + SSE**
- **SSR** repräsentiert den Effekt der X-Variablen auf die Variation von Y, **SSE** den geschätzten Effekt des Fehlers e auf die Variation von Y.



- Für das **Bestimmtheitsmaß** teilt man den erklärten Varianzanteil durch den Gesamtanteil. Dies geschieht für **alle** Beobachtungen.

Bestimmtheitsmaß

$$R^2 = \frac{SSR = Regression Sum of Squares}{SST = Total Sum of Squares} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

- Dividiert man also die Variation der Vorhersagewerte durch die Gesamtvariation von Y, so erhalten wir das **Bestimmtheitsmaß** (Determinationskoeffizient).

- Wertebereich: $0 <= R^2 <= 1$

Bestimmtheitsmaß

- **Interpretation**: In unserem Riverview Beispiel beträgt das **Bestimmtheitsmaß 0,63**. Damit erklärt unser einfaches Modell immerhin schon 63 Prozent der Variation des Einkommens.
- Der Wert gibt an, wie hoch der Anteil der Varianz von Y ist, der durch die unabhängige Variable X erklärt wird. Maximal beträgt der Wert 1, dann würden 100 Prozent der Varianz durch das Regressionsmodell erklärt. In diesem Fall wäre die Residualvarianz null, alle Datenpunkte würden exakt auf der Regressionsgeraden liegen. Kann das lineare Regressionsmodell dagegen überhaupt nichts erklären, so ist das Bestimmtheitsmaß null.
- R^2 ist dann null, wenn kein linearer Zusammenhang gegeben ist. Dann verläuft die Regressionsgerade parallel zur X-Achse.
- Wichtig: Varianzerklärung bedeutet nicht **kausale** Erklärung. Man muss im Grunde immer mit dem Problem der **Scheinkausalität** rechnen! Vgl. zu Scheinkausalitäten: Spurious Correlations

Lineare Regression in R: Ein **lineares Modell (lm)** hat in *R* die Form y ~ x

```
lm( income ~ edu, data=Riverview ) %>% summary()
```

```
##
## Call:
## lm(formula = income ~ edu, data = Riverview)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -15808  -5783   2088   5127  18379
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11321.4     6123.2   1.849   0.0743 .
## edu           2651.3      369.6   7.173 5.56e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8978 on 30 degrees of freedom
## Multiple R-squared:  0.6317,    Adjusted R-squared:  0.6194
## F-statistic: 51.45 on 1 and 30 DF,  p-value: 5.562e-08
```

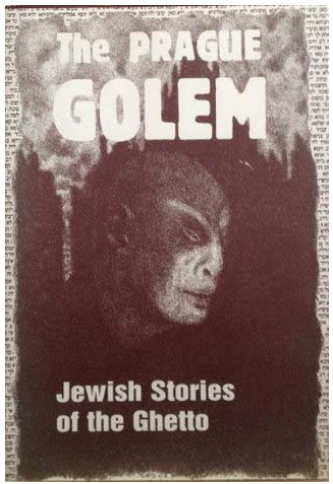
Der Golem von Prag

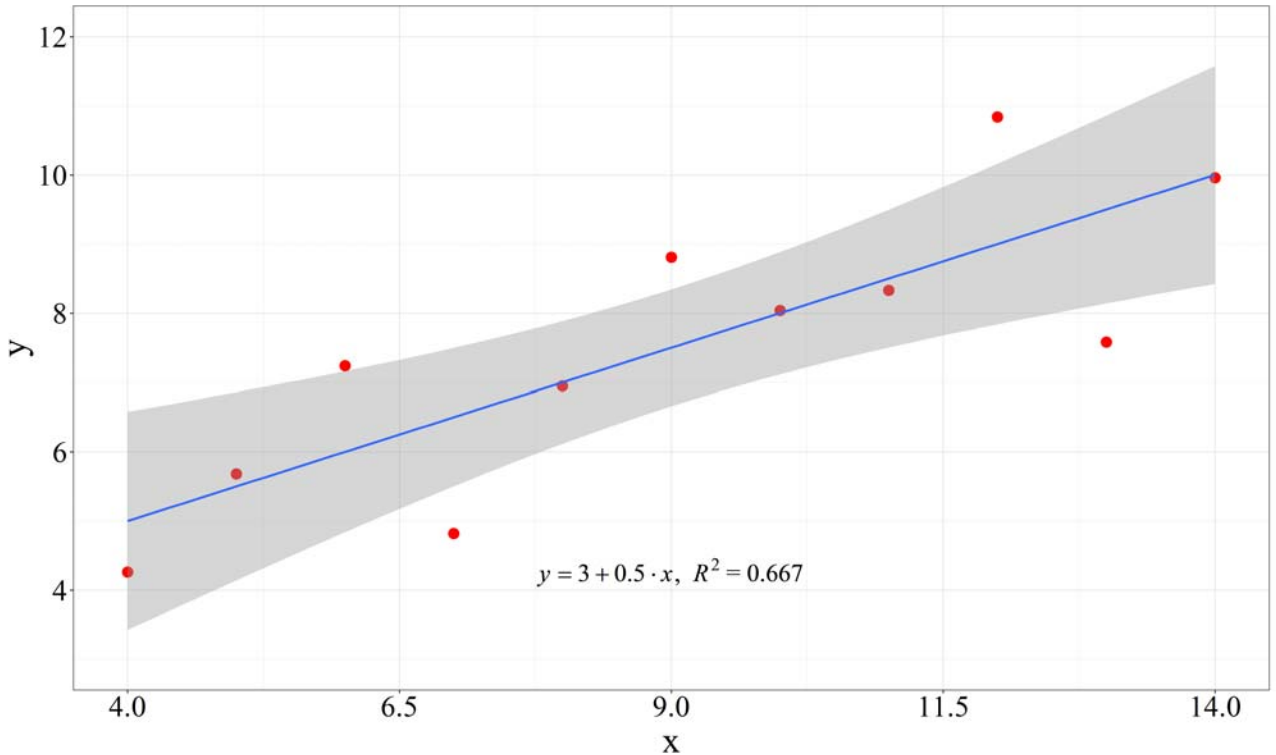
A statistical tale by Richard McElreath

- Ein Golem ist nach jüdischer Sage ein aus Lehm oder Ton künstlich erschaffenes, stummes menschliches Wesen, das oft **gewaltige Größe und Kraft** besitzt.
- Bekannt vor allem durch die Legende von **Rabbi Löw**, der um 1580 in Prag eine von ihm geknetete Tonfigur für einige Zeit belebt haben soll.

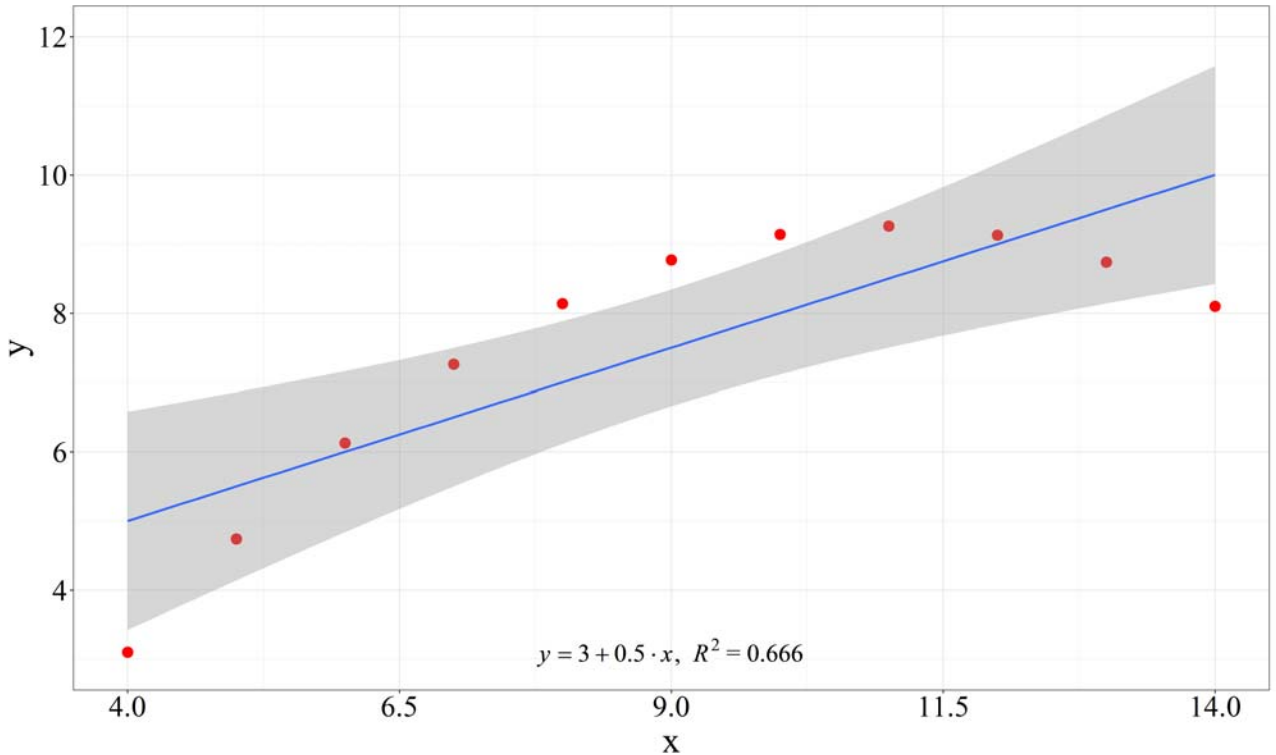
How to Golem-Erschaffung:

```
1. Nimm eine Tonne Lehm
2. Forme einen Menschenkörper
3. Schreib auf die Stirn: *emeth*
("Wahrheit")
4. Gib Befehle... vorsichtig!
```

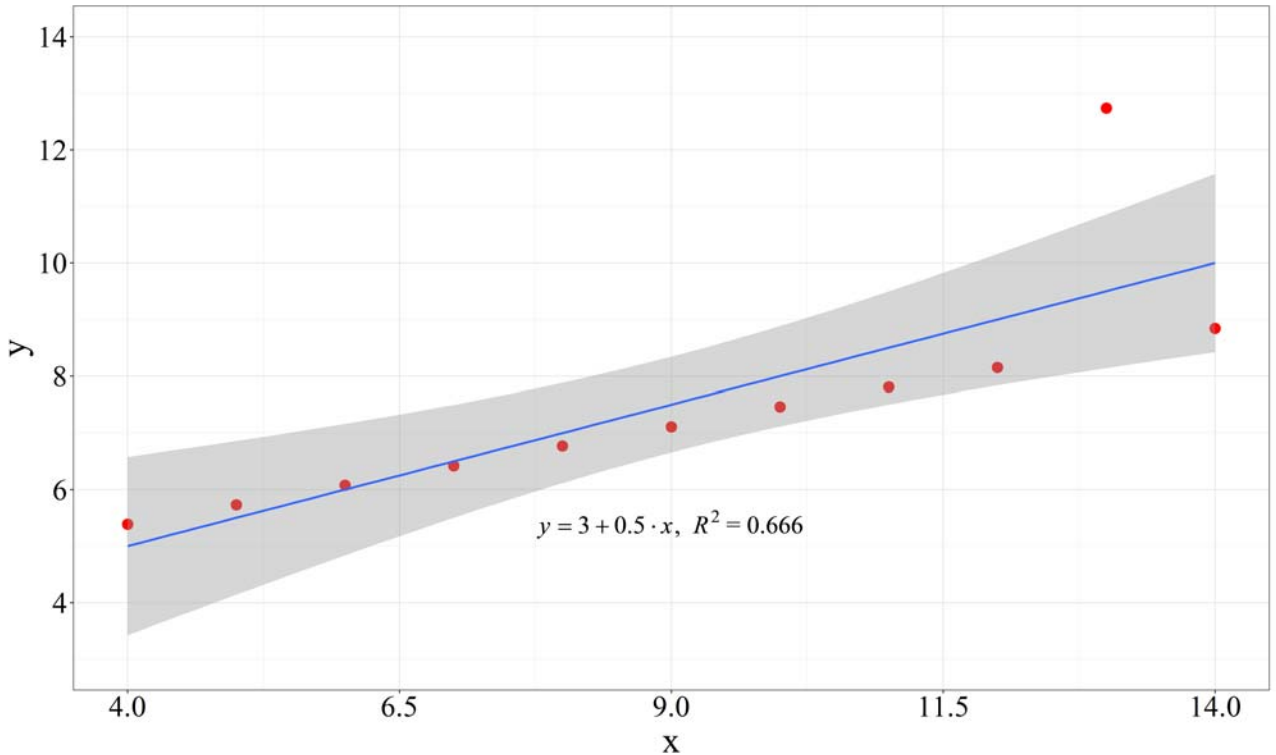




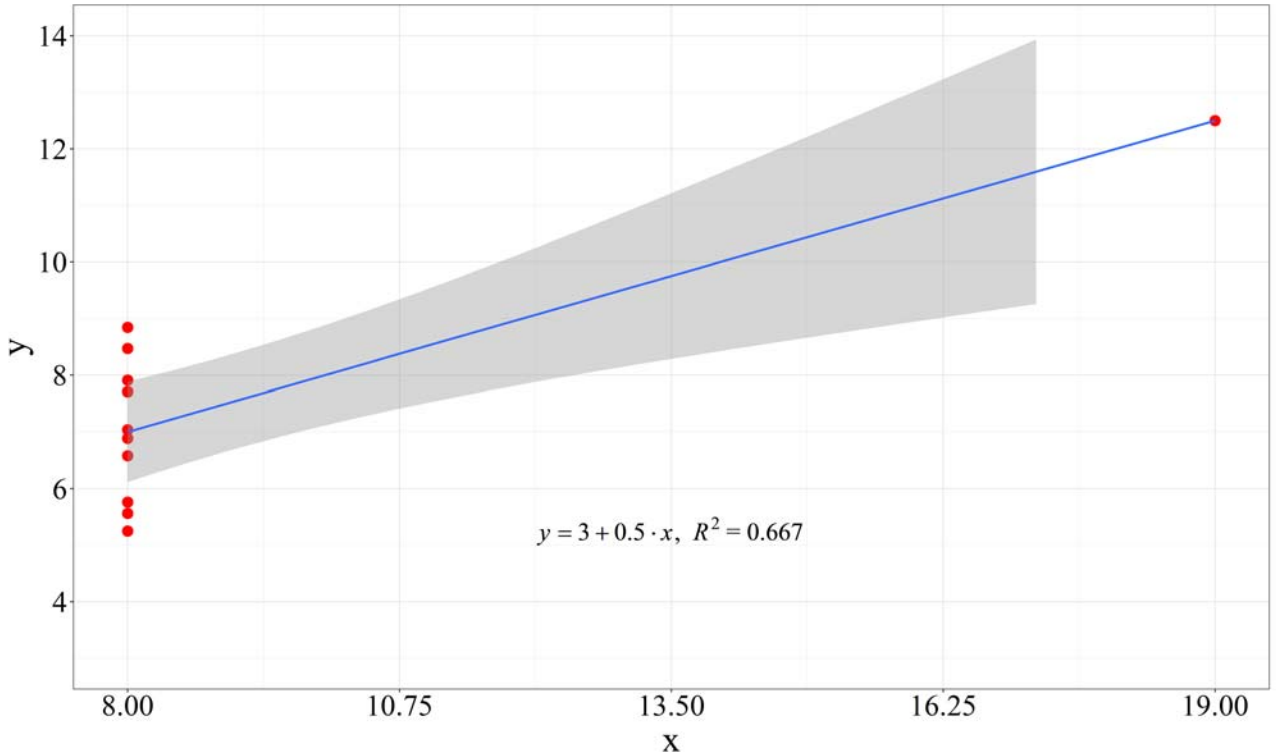
- Wo der **Golem** scheitert: Das **Anscombe-Quartett** besteht aus vier Mengen von Datenpunkten, die nahezu identische einfache statistische Eigenschaften haben, aber aufgetragen sehr verschieden aussehen



- Jede dieser vier Mengen besteht aus elf (x,y)-Punkten.



- Diese vier Mengen wurden im Jahre 1973 von dem englischen Statistiker Francis Anscombe konstruiert, um die Bedeutung einer graphischen Datenanalyse herauszustellen und die Effekte von Ausreißern zu demonstrieren.


14
12
10
8
6
4
y
8.00
10.75
13.50
16.25
19.00
x
$y = 3 + 0.5 \cdot x,\ R^2 = 0.667$

Data wrangling

- Um Daten zu manipulieren können wir drei **dplyr-Funktionen** nutzen. Eine Dokumentation findet sich hier: http://dplyr.tidyverse.org/ und eine interaktive Hilfe hier: https://www.rdocumentation.org/

Filter

- **filter**: Mit `filter` können wir den Datensatz so reduzieren, dass wir nur Zeilen behalten/nutzen, die die Bedingung innerhalb der Filterfunktion `(...)` erfüllen.

```
imdb %>% filter(show.title == "The Wire")
```

- Bei numerischen Werten können wir auch Vergleichsoperatoren verwenden, z.B. <, >, <=, >=, ==, !=.

```
imdb %>% filter(episode.rating >= 9.9)
```

Filter

- Um eine bestimme Gruppe auszuschließen verwenden wir ein `!`.

```
imdb %>% filter(!(show.title == "The Wire"))
```

- Zeilen die 2 (oder mehr) Bedingungen gleichzeitig erfüllen sollen, kann man mit einem **UND** & ansprechen. Möglich auch mit `,`.

```
imdb %>% filter(show.title=="Game of Thrones" & season.no <= 2)
imdb %>% filter(show.title=="Game of Thrones", season.no <= 2)
```

- Zeilen die eine **ODER** eine andere Bedingungen erfüllen sollen, kann man mit einem `|` ansprechen.

```
imdb %>% filter(episode.rating >= 9 | episode.rating.count > 10000)
```

Filter

- Mehrere Bedingungen kann man auch mit dem `%in%`-Befehl formulieren. Die Funktion `nrow()` gibt die Zeilenzahl.

```
imdb %>%
  filter( show.title == "The Wire" | show.title == "Friends" ) %>%
  nrow()
```

```
## [1] 254
```

```
imdb %>%
  filter( show.title %in% c("The Wire", "Friends")) %>%
  nrow()
```

```
## [1] 254
```

mutate

- Mit **mutate** kann man einzelne Spalten innerhalb eines Datensatzes verändern, manipulieren oder neue Berechnungen vornehmen.
- Beispielsweise möchte man einen Laufindex aus `season.no` und `episode.no` erstellen. Dazu kann man ähnlich wie bei der Vergabe von Zimmernummern im Hotel vorgehen in dem man die `season.no` mit 100 multipliziert und die `episode.no` addiert.

```
imdb <- imdb %>% mutate(
  laufindex = (season.no * 100) + episode.no
)
imdb %>%
  select(show.title, season.no, episode.no, laufindex) %>%
  head(3)
```

```
##                   show.title season.no episode.no laufindex
## 1 The Man in the High Castle         1          1       101
## 2 The Man in the High Castle         1          2       102
## 3 The Man in the High Castle         1          3       103
```

mutate

- **Beispiel 2:** Man möchte die Distanz zwischen dem Mittelwert der Episoden und einer einzelnen Episode berechnen, um etwa zu erfahren, ob eine bestimmte Episode über- oder unterdurchschnittlich war. Dazu berechnet man zunächst den Mittelwert aller `episode.rating` (dadurch steht in der Spalte `episode.mittelwert` in jeder Zeile der gleiche Wert) und zieht davon anschließend das individuelle `episode.rating` ab.

```
imdb <- imdb %>%
  mutate(episode.mittelwert = mean(episode.rating),
         episode.distanz   = episode.mittelwert - episode.rating
        )
imdb %>% select(show.title, episode.rating, episode.mittelwert, epis
```

```
##                   show.title episode.rating episode.mittelwert
## 1 The Man in the High Castle            8.1           8.607558
## 2 The Man in the High Castle            8.3           8.607558
## 3 The Man in the High Castle            7.9           8.607558
##   episode.distanz
## 1       0.5075581
## 2       0.3075581
## 3       0.7075581
```

- `select` wird hier genutzt um einzelne Spalten auszuwählen.

group_by

- Mit group_by lassen sich die Daten in Gruppen einteilen. Dadurch werden Berechnungen nur **gruppenweise** vorgenommen.

```
imdb %>% group_by(show.title) %>%
  summarize(mittelwert=mean(episode.rating))
```

```
## # A tibble: 9 × 2
##                    show.title mittelwert
##                        <fctr>      <dbl>
## 1       House of Cards (1990)   8.625000
## 2                     Friends   8.542268
## 3             Game of Thrones   9.040000
## 4              House of Cards   8.705769
## 5                Prison Break   8.740351
## 6             Stranger Things   8.900000
## 7               The Blacklist   8.410448
## 8 The Man in the High Castle   8.405000
## 9                    The Wire   8.493333
```