Tutorial 8

Lars Abt

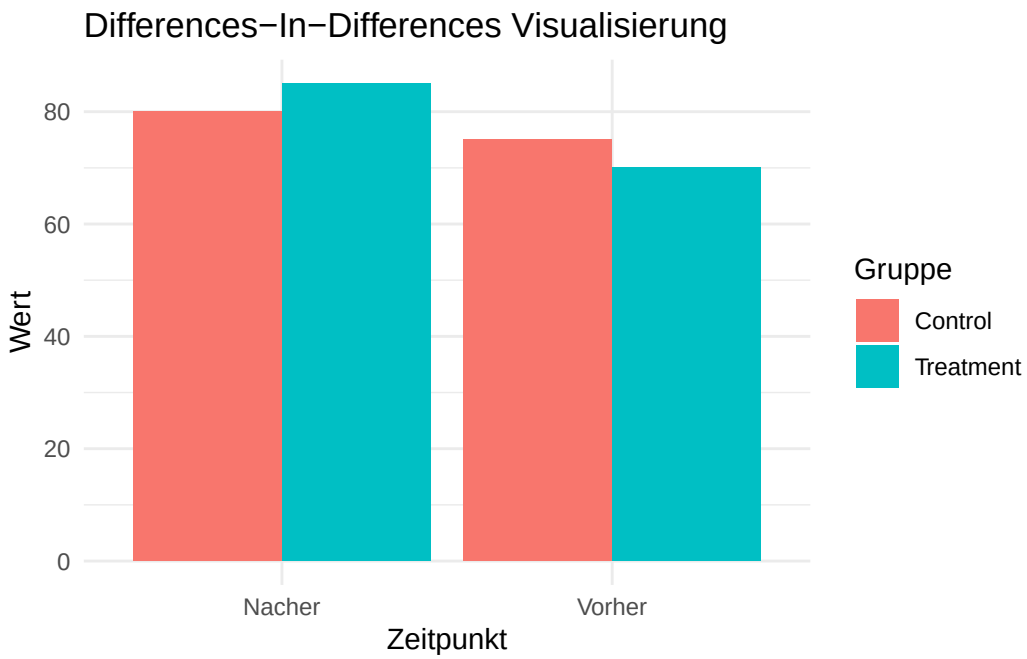
Differences-In-Differences

```
# Pakete laden
pacman::p_load(data.table, ggplot2)

# Data.Table anlegen
did_data <- data.table(
    Gruppe = c("Treatment", "Treatment", "Control", "Control"),
    Zeit = c("Vorher", "Nacher", "Vorher", "Nacher"),
    Wert = c(70, 85, 75, 80)
    )

# Kopie der Daten erstellen, als Sicherungskopie (und für spätere Replizierbarkeit, da kein (
did_data_backup <- copy(did_data)

# Daten visualisieren
did_plot <- ggplot(did_data, aes(x = Zeit, y = Wert, fill = Gruppe)) +
    geom_bar(stat = "identity", position = "dodge") +
    labs(title = "Differences-In-Differences Visualisierung", x = "Zeitpunkt", y = "Wert") +
    theme_minimal()
did_plot
```



Wir sehen, dass die Werte für beide Gruppen nach dem Treatment ansteigen, aber der Anstieg bei der Treatment-Gruppe deutlich stärker ist. Dies ist der Effekt, den wir mit einem Differences-In-Differences Schätzer messen wollen.

Differences-In-Differences Schätzer

```
# Wir nutzen unsere Sicherungskopie der Daten, sodass wir diesen Code-Chunk immer wieder aus
did_data <- copy(did_data_backup)

# Zeitwerte in numerische Werte übersetzen
did_data[, Zeit := ifelse(Zeit == "Vorher", 0, 1)]

# Gruppenwerte in numerische Werte übersetzen
did_data[, Gruppe := ifelse(Gruppe == "Treatment", 1, 0)]

# Berechnung des Effekts mit unserer "normalen" Formel über die Differenzen
diff1 <- did_data[Gruppe == 1 & Zeit == 1]$Wert - did_data[Gruppe == 1 & Zeit == 0]$Wert
diff2 <- did_data[Gruppe == 0 & Zeit == 1]$Wert - did_data[Gruppe == 0 & Zeit == 0]$Wert

diff_effect <- diff1 - diff2
diff_effect
```

```
[1] 10
```

Diff-In-Diff Effekt beträgt 10.

Jetzt als Lineare Regression

```
# Erstellen unseres Interaktionsterms
did_data[, Interaktion := Gruppe * Zeit]
# Wird nur zu 1, wenn wir in Treatment-Gruppe und nach Verabreichung des Treatments

# Lineares Regressionsmodell schätzen
fit_model <- lm(Wert ~ Gruppe + Zeit + Interaktion, data = did_data)
summary(fit_model)
```

```
Call:
lm(formula = Wert ~ Gruppe + Zeit + Interaktion, data = did_data)

Residuals:
ALL 4 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)       75        NaN     NaN      NaN
Gruppe            -5        NaN     NaN      NaN
Zeit               5        NaN     NaN      NaN
Interaktion       10        NaN     NaN      NaN

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:    NaN
F-statistic:   NaN on 3 and 0 DF,  p-value: NA
```

```
# alternativ zu Interaktion: Gruppe * Zeit
fit_model <- lm(Wert ~ Gruppe + Zeit + Gruppe*Zeit, data = did_data)
summary(fit_model)
```

```
Call:
lm(formula = Wert ~ Gruppe + Zeit + Gruppe * Zeit, data = did_data)
```

```
Residuals:
ALL 4 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)       75        NaN     NaN      NaN
Gruppe            -5        NaN     NaN      NaN
Zeit               5        NaN     NaN      NaN
Gruppe:Zeit       10        NaN     NaN      NaN

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:    NaN
F-statistic:   NaN on 3 and 0 DF,  p-value: NA
```

Wir sehen, dass unsere Diff-in-Diff Schätzung mit Hilfe eines linearen Regressionsmodells den gleichen Effekt schätzt, wie unsere manuelle Berechnung. Dieser ist in unserem Interaktionsterm wiederzufinden, da dieser genau den Fall des isolierten Treatment-Effekts darstellt. Zudem schätzen wir auch zeitliche Effekte in unserer `post`-Variable und Gruppeneffekte in unserer `group`Variable, welch hier aber keine Bedeutung besitzen.

Bootstrapping

```r
# Working Directory überprüfen
# getwd()

# Daten herunterladen
download.file("https://statistik.julianhinz.com/tutorials/tutorial7/avocado.csv", destfile =

# Daten einlesen
avocado_data <- fread("../input/avocado.csv")
```

Wir laden unseren Paketmanager `pacman`, und laden unseren Datensatz programmatisch herunter, und speichern ihn in dem dafür vorgesehen Ordner. Anschließend laden wir die Daten mit Hilfe von `fread` in R.

Mit dem Datensatz arbeiten

```r
# Daten ansehen
head(avocado_data)
```

```
      V1       Date AveragePrice Total Volume    4046      4225   4770
   <int>     <IDat>        <num>        <num>   <num>     <num>  <num>
1:     0 2015-12-27         1.33     64236.62 1036.74  54454.85  48.16
2:     1 2015-12-20         1.35     54876.98  674.28  44638.81  58.33
3:     2 2015-12-13         0.93    118220.22  794.70 109149.67 130.50
4:     3 2015-12-06         1.08     78992.15 1132.00  71976.41  72.58
5:     4 2015-11-29         1.28     51039.60  941.48  43838.39  75.78
6:     5 2015-11-22         1.26     55979.78 1184.27  48067.99  43.61
   Total Bags Small Bags Large Bags XLarge Bags         type  year region
        <num>      <num>      <num>       <num>       <char> <int> <char>
1:    8696.87    8603.62      93.25           0 conventional  2015 Albany
2:    9505.56    9408.07      97.49           0 conventional  2015 Albany
3:    8145.35    8042.21     103.14           0 conventional  2015 Albany
4:    5811.16    5677.40     133.76           0 conventional  2015 Albany
5:    6183.95    5986.26     197.69           0 conventional  2015 Albany
6:    6683.91    6556.47     127.44           0 conventional  2015 Albany
```

```r
str(avocado_data)
```

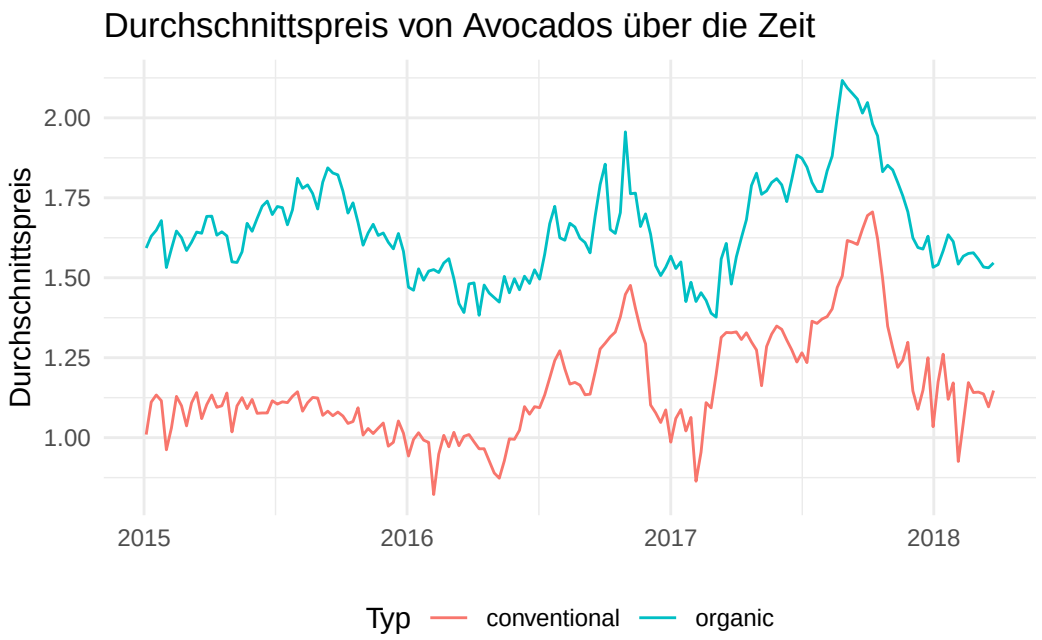
```
Classes 'data.table' and 'data.frame':  18249 obs. of  14 variables:
 $ V1          : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Date        : IDate, format: "2015-12-27" "2015-12-20" ...
 $ AveragePrice: num  1.33 1.35 0.93 1.08 1.28 1.26 0.99 0.98 1.02 1.07 ...
 $ Total Volume: num  64237 54877 118220 78992 51040 ...
 $ 4046        : num  1037 674 795 1132 941 ...
 $ 4225        : num  54455 44639 109150 71976 43838 ...
 $ 4770        : num  48.2 58.3 130.5 72.6 75.8 ...
 $ Total Bags  : num  8697 9506 8145 5811 6184 ...
 $ Small Bags  : num  8604 9408 8042 5677 5986 ...
 $ Large Bags  : num  93.2 97.5 103.1 133.8 197.7 ...
 $ XLarge Bags : num  0 0 0 0 0 0 0 0 0 0 ...
 $ type        : chr  "conventional" "conventional" "conventional" "conventional" ...
 $ year        : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ region      : chr  "Albany" "Albany" "Albany" "Albany" ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
# Variablen auswählen und vernünftig umbenennen
data = avocado_data[, .(date = Date,
                 average_price = AveragePrice,
                 total_volume = `Total Volume`,
                 region = region,
                 type = type)]
```

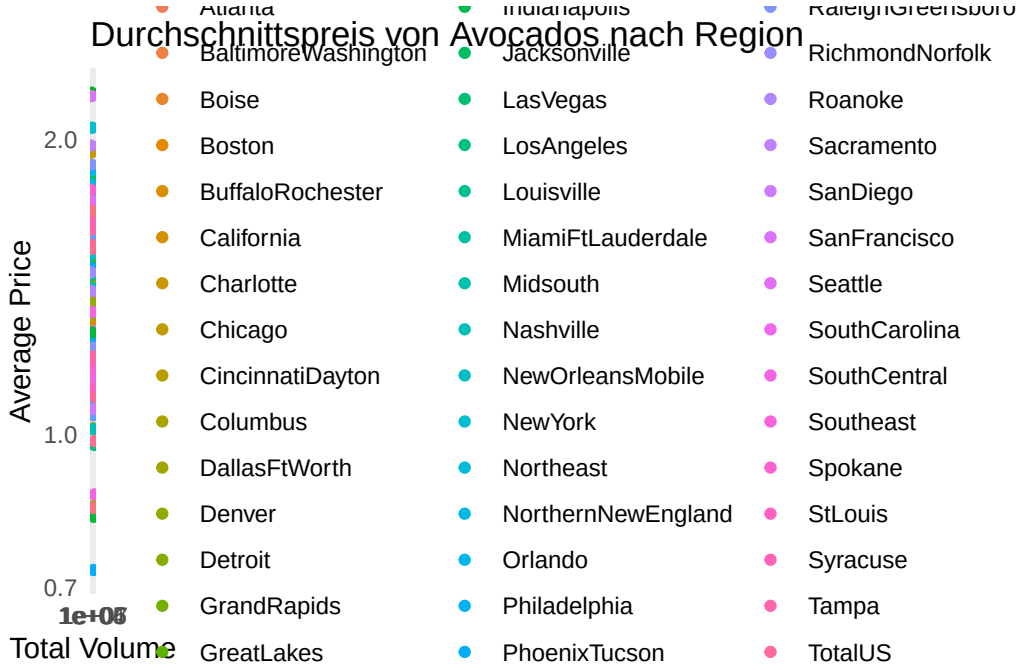
Das Umbennen der Variablen und die Selektion ist nicht zwingend nötig, erlaubt uns aber hier, mit unseren Namenskonventionen konstant zu bleiben, und unnötigen Ballast zu entfernen. Dies ist nur ratsam, wenn wir sicher sind, dass wir die anderen Variablen nicht mehr benötigen.

Plotten

```
# Variation über die Zeit
plot_data_time = data[, .(average_price = mean(average_price)), by = .(date, type)]
ggplot(plot_data_time) +
    theme_minimal() +
    geom_line(aes(x = date, y = average_price, color = type)) +
    scale_x_date(NULL) +
    labs(title = "Durchschnittspreis von Avocados über die Zeit",
        y = "Durchschnittspreis",
        color = "Typ") +
    theme(legend.position = "bottom")
```




```
## Variation über die Regionen
plot_data_region = data[, .(average_price = mean(average_price),
                      total_volume = mean(total_volume)), by = .(region, type)]
ggplot(plot_data_region) +
    theme_minimal() +
    geom_point(aes(x = total_volume,
                   y = average_price,
                   color = region)) +
    scale_x_log10("Total Volume") +
    scale_y_log10("Average Price") +
    labs(title = "Durchschnittspreis von Avocados nach Region") +
    theme(legend.position = "right")
```



Wir sehen klare Unterschiede bei den Durchschnittspreisen wenn wir nach Typ unterscheiden, und erkennen, dass Bio-Avocados immer teuer sind als klassische Avocados. Zudem sind die Verläufe insgesamt, absteigend und ansteigend, nahezu identisch. Auch die Regionen haben einen großen Einfluss auf den Preis und zeigen deutliche Unterschiede an, allerdings gibt es so viele Regionen, dass die weitere Interpretation des Plots schwierig ist.

Lineare Regression

```
# Verschiedene mögliche Gleichungen der linearen Regression
regression0 = lm(log(total_volume) ~ log(average_price), data = data)
summary(regression0)
```

```
Call:
lm(formula = log(total_volume) ~ log(average_price), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-6.2488 -1.2830 -0.0161  1.1999  6.6370

Coefficients:
```

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       12.72076    0.01968  646.31   <2e-16 ***
log(average_price) -4.68824    0.04723  -99.26   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.848 on 18247 degrees of freedom
Multiple R-squared:  0.3506,    Adjusted R-squared:  0.3506
F-statistic:  9853 on 1 and 18247 DF,  p-value: < 2.2e-16
```

```
regression1 = lm(log(total_volume) ~ log(average_price) + type, data = data)
summary(regression1)
```

```

Call:
lm(formula = log(total_volume) ~ log(average_price) + type, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.0707 -0.9661 -0.1823  0.7308  4.8455

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       13.29818    0.01515  877.54   <2e-16 ***
log(average_price) -1.28678    0.04402  -29.23   <2e-16 ***
typeorganic       -3.19332    0.02550 -125.21   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.356 on 18246 degrees of freedom
Multiple R-squared:  0.6507,    Adjusted R-squared:  0.6507
F-statistic: 1.7e+04 on 2 and 18246 DF,  p-value: < 2.2e-16
```

```
regression2 = lm(log(total_volume) ~ log(average_price) + region, data = data)
summary(regression2)
```

```

Call:
lm(formula = log(total_volume) ~ log(average_price) + region,
    data = data)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-5.4887 -0.9402  0.0519  0.8926  4.2856

Coefficients:
                          Estimate Std. Error  t value Pr(>|t|)
(Intercept)               11.72645    0.07157  163.839  < 2e-16 ***
log(average_price)        -5.26848    0.03625 -145.323  < 2e-16 ***
regionAtlanta              0.75533    0.09904    7.627 2.52e-14 ***
regionBaltimoreWashington  2.16075    0.09883   21.864  < 2e-16 ***
regionBoise               -0.94450    0.09905   -9.536  < 2e-16 ***
regionBoston               1.63722    0.09883   16.566  < 2e-16 ***
regionBuffaloRochester     0.43045    0.09883    4.355 1.34e-05 ***
regionCalifornia           3.55124    0.09894   35.892  < 2e-16 ***
regionCharlotte            1.06789    0.09883   10.806  < 2e-16 ***
regionChicago              2.37435    0.09883   24.025  < 2e-16 ***
regionCincinnatiDayton     0.03832    0.09934    0.386 0.699726
regionColumbus            -0.26924    0.09919   -2.714 0.006645 **
regionDallasFtWorth        0.43430    0.09979    4.352 1.36e-05 ***
regionDenver               1.07131    0.09928   10.791  < 2e-16 ***
regionDetroit              0.53559    0.09913    5.403 6.64e-08 ***
regionGrandRapids          0.17283    0.09884    1.749 0.080392 .
regionGreatLakes           3.06363    0.09898   30.951  < 2e-16 ***
regionHarrisburgScranton   0.92989    0.09883    9.409  < 2e-16 ***
regionHartfordSpringfield  1.97801    0.09895   19.990  < 2e-16 ***
regionHouston              0.24030    0.09999    2.403 0.016262 *
regionIndianapolis        -0.25261    0.09903   -2.551 0.010759 *
regionJacksonville         0.21497    0.09885    2.175 0.029654 *
regionLasVegas             0.53219    0.09900    5.376 7.72e-08 ***
regionLosAngeles           2.09463    0.09934   21.086  < 2e-16 ***
regionLouisville          -0.93416    0.09911   -9.426  < 2e-16 ***
regionMiamiFtLauderdale    0.64291    0.09889    6.501 8.17e-11 ***
regionMidsouth             3.15469    0.09890   31.898  < 2e-16 ***
regionNashville           -0.36324    0.09931   -3.657 0.000255 ***
regionNewOrleansMobile    -0.19577    0.09906   -1.976 0.048146 *
regionNewYork              3.33701    0.09888   33.747  < 2e-16 ***
regionNortheast            4.08066    0.09883   41.289  < 2e-16 ***
regionNorthernNewEngland   1.34572    0.09885   13.614  < 2e-16 ***
regionOrlando              0.83616    0.09884    8.459  < 2e-16 ***
regionPhiladelphia         1.83734    0.09884   18.589  < 2e-16 ***
regionPhoenixTucson        0.40016    0.09956    4.019 5.86e-05 ***
regionPittsburgh          -0.22453    0.09893   -2.270 0.023243 *
regionPlains               2.60243    0.09888   26.318  < 2e-16 ***
```

```
regionPortland             1.23474    0.09911   12.459  < 2e-16 ***
regionRaleighGreensboro    1.18903    0.09883   12.031  < 2e-16 ***
regionRichmondNorfolk      0.21328    0.09907    2.153 0.031338 *
regionRoanoke             -0.30175    0.09918   -3.043 0.002349 **
regionSacramento           1.56304    0.09883   15.816  < 2e-16 ***
regionSanDiego             1.08490    0.09897   10.962  < 2e-16 ***
regionSanFrancisco         2.87583    0.09891   29.075  < 2e-16 ***
regionSeattle              1.87160    0.09891   18.923  < 2e-16 ***
regionSouthCarolina        0.75861    0.09892    7.669 1.82e-14 ***
regionSouthCentral         2.14110    0.09970   21.474  < 2e-16 ***
regionSoutheast            2.87414    0.09892   29.055  < 2e-16 ***
regionSpokane             -0.33510    0.09891   -3.388 0.000705 ***
regionStLouis              0.32193    0.09891    3.255 0.001137 **
regionSyracuse            -0.24214    0.09883   -2.450 0.014292 *
regionTampa                0.43862    0.09891    4.434 9.28e-06 ***
regionTotalUS              5.12488    0.09903   51.750  < 2e-16 ***
regionWest                 3.23992    0.09918   32.666  < 2e-16 ***
regionWestTexNewMexico     0.68590    0.09953    6.892 5.70e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.285 on 18194 degrees of freedom
Multiple R-squared:  0.6871,    Adjusted R-squared:  0.6862
F-statistic:   740 on 54 and 18194 DF,  p-value: < 2.2e-16
```

```r
# Besonders relevant!
regression3 = lm(log(total_volume) ~ log(average_price) + type + region, data = data)
summary(regression3)
```

```
Call:
lm(formula = log(total_volume) ~ log(average_price) + type +
    region, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4086 -0.2388  0.0036  0.2555  2.2269

Coefficients:
                         Estimate Std. Error  t value Pr(>|t|)
(Intercept)             11.547369   0.023926  482.628  < 2e-16 ***
log(average_price)      -0.978088   0.016554  -59.084  < 2e-16 ***
```

```
typeorganic                -3.303655   0.008685 -380.371  < 2e-16 ***
regionAtlanta               1.515884   0.033160   45.714  < 2e-16 ***
regionBaltimoreWashington   2.237114   0.033031   67.727  < 2e-16 ***
regionBoise                -0.163731   0.033167   -4.937 8.02e-07 ***
regionBoston                1.742457   0.033033   52.749  < 2e-16 ***
regionBuffaloRochester      0.537592   0.033033   16.275  < 2e-16 ***
regionCalifornia            4.120739   0.033103  124.482  < 2e-16 ***
regionCharlotte             1.008430   0.033031   30.530  < 2e-16 ***
regionChicago               2.412701   0.033031   73.045  < 2e-16 ***
regionCincinnatiDayton      1.234517   0.033351   37.016  < 2e-16 ***
regionColumbus              0.728672   0.033254   21.912  < 2e-16 ***
regionDallasFtWorth         2.070490   0.033628   61.570  < 2e-16 ***
regionDenver                2.195506   0.033314   65.904  < 2e-16 ***
regionDetroit               1.456655   0.033221   43.848  < 2e-16 ***
regionGrandRapids           0.381429   0.033040   11.544  < 2e-16 ***
regionGreatLakes            3.724070   0.033128  112.413  < 2e-16 ***
regionHarrisburgScranton    1.077646   0.033035   32.621  < 2e-16 ***
regionHartfordSpringfield   1.393635   0.033107   42.095  < 2e-16 ***
regionHouston               2.043074   0.033755   60.527  < 2e-16 ***
regionIndianapolis          0.506389   0.033160   15.271  < 2e-16 ***
regionJacksonville          0.445170   0.033042   13.473  < 2e-16 ***
regionLasVegas              1.216328   0.033135   36.708  < 2e-16 ***
regionLosAngeles            3.282953   0.033347   98.449  < 2e-16 ***
regionLouisville           -0.053508   0.033205   -1.611    0.107
regionMiamiFtLauderdale     1.055891   0.033069   31.930  < 2e-16 ***
regionMidsouth              3.602943   0.033075  108.931  < 2e-16 ***
regionNashville             0.798255   0.033333   23.948  < 2e-16 ***
regionNewOrleansMobile      0.611929   0.033177   18.444  < 2e-16 ***
regionNewYork               2.942602   0.033065   88.994  < 2e-16 ***
regionNortheast             3.975746   0.033033  120.358  < 2e-16 ***
regionNorthernNewEngland    1.592448   0.033044   48.192  < 2e-16 ***
regionOrlando               1.058020   0.033041   32.021  < 2e-16 ***
regionPhiladelphia          1.650033   0.033038   49.943  < 2e-16 ***
regionPhoenixTucson         1.830051   0.033488   54.648  < 2e-16 ***
regionPittsburgh            0.311028   0.033095    9.398  < 2e-16 ***
regionPlains                2.995033   0.033065   90.580  < 2e-16 ***
regionPortland              2.117313   0.033205   63.764  < 2e-16 ***
regionRaleighGreensboro     1.275913   0.033032   38.627  < 2e-16 ***
regionRichmondNorfolk       1.029428   0.033180   31.026  < 2e-16 ***
regionRoanoke               0.681278   0.033247   20.491  < 2e-16 ***
regionSacramento            1.481435   0.033032   44.849  < 2e-16 ***
regionSanDiego              1.705596   0.033117   51.502  < 2e-16 ***
regionSanFrancisco          2.388737   0.033084   72.203  < 2e-16 ***
```

```
regionSeattle              2.339973   0.033080   70.738  < 2e-16 ***
regionSouthCarolina        1.254773   0.033086   37.925  < 2e-16 ***
regionSouthCentral         3.703322   0.033576  110.298  < 2e-16 ***
regionSoutheast            3.382530   0.033088  102.227  < 2e-16 ***
regionSpokane              0.134868   0.033080    4.077 4.58e-05 ***
regionStLouis              0.795652   0.033081   24.052  < 2e-16 ***
regionSyracuse            -0.154400   0.033032   -4.674 2.97e-06 ***
regionTampa                0.921008   0.033083   27.840  < 2e-16 ***
regionTotalUS              5.876608   0.033157  177.234  < 2e-16 ***
regionWest                 4.232687   0.033252  127.293  < 2e-16 ***
regionWestTexNewMexico     1.824493   0.033398   54.628  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4294 on 18193 degrees of freedom
Multiple R-squared:  0.9651,    Adjusted R-squared:  0.9649
F-statistic:  9135 on 55 and 18193 DF,  p-value: < 2.2e-16
```

```r
regression4 = lm(log(total_volume) ~ log(average_price) * type, data = data)
summary(regression4)
```

```
Call:
lm(formula = log(total_volume) ~ log(average_price) * type, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.0686 -0.9650 -0.1804  0.7358  4.8540

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   13.31509    0.01604 830.322  < 2e-16 ***
log(average_price)            -1.42676    0.06188 -23.056  < 2e-16 ***
typeorganic                   -3.27878    0.03682 -89.059  < 2e-16 ***
log(average_price):typeorganic  0.28326    0.08803   3.218  0.00129 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.355 on 18245 degrees of freedom
Multiple R-squared:  0.6509,    Adjusted R-squared:  0.6509
F-statistic: 1.134e+04 on 3 and 18245 DF,  p-value: < 2.2e-16
```

Wie interpretieren wir das ganze? Wir schauen uns `regression4`einmal genauer an:

Der Koeffizient für `log(average_price)` ist -1.43. Dies können wir so deuten, dass eine 1%ige Erhöhung des Durchschnittspreises mit einer Abnahme des Gesamtvolumens um etwa 1.43% einhergeht (da wir hier ein log-log-Modell haben). Der Koeffizient für `typeorganic` ist -3.28. Hier liegt ein log-level-Modell vor, dies können wir so deuten, dass, wenn wir `log(average_price)` konstant halten, der Wert für `log(total_volume)` (also der Gesamtabsatz) für Bio-Produkte um 3.28*100% niedriger ist als für konventionelle Produkte. Und der Interaktionsterm zwischen `log(average_price)` und `typeorganic` ist 0.28. Dies können wir so deuten, dass der negative Einfluss von steigenden Preisen auf den Absatz für Bio-Produkte um 0.28 Einheiten geringer ist als für konventionelle Produkte (also -1.43 + 0.28 = -1.15).

Bootstrap

```r
# Seed setzen
set.seed(12345)

n <- 250
bootstrap_elastizität = numeric(n)

for (i in 1:n) {
    bootstrap_data <- data[sample(1:nrow(data), size = nrow(data), replace = TRUE)]
    bootstrap_model <- lm(log(total_volume) ~ log(average_price) + type + region,
                          data = bootstrap_data)
    # type und region nicht zwingend notwendig. aber vlt. schöner
    bootstrap_elastizität[i] <- coef(bootstrap_model)[2]
    print(paste0(i, ": ", bootstrap_elastizität[i]))
    # nur für die Übersichtlichkeit
}
```

```
[1] "1: -0.969673516164367"
[1] "2: -0.998095961859883"
[1] "3: -0.939005951197449"
[1] "4: -0.968419664667269"
[1] "5: -1.00061478371613"
[1] "6: -0.972168358873977"
[1] "7: -0.959384482836069"
[1] "8: -0.990434249427722"
[1] "9: -0.968508900978095"
[1] "10: -0.980850434523678"
[1] "11: -0.954664241174224"
```

```
[1] "12: -0.963990985189253"
[1] "13: -0.981656076647602"
[1] "14: -1.00907583035466"
[1] "15: -0.94513013372944"
[1] "16: -0.976120843567589"
[1] "17: -0.993865607161841"
[1] "18: -0.993032043448355"
[1] "19: -0.994068564006097"
[1] "20: -0.980966176006796"
[1] "21: -0.985121886585841"
[1] "22: -0.94554578968297"
[1] "23: -0.992535419952877"
[1] "24: -0.974484489988464"
[1] "25: -0.972173773818143"
[1] "26: -0.960880430822915"
[1] "27: -0.967122069646993"
[1] "28: -0.963510219000144"
[1] "29: -0.977537518411463"
[1] "30: -0.998074672586454"
[1] "31: -0.989893795815324"
[1] "32: -1.00912051994213"
[1] "33: -0.951322155175248"
[1] "34: -0.988940995512063"
[1] "35: -0.987193706328388"
[1] "36: -0.975785896354413"
[1] "37: -0.962566803436722"
[1] "38: -0.997034738971481"
[1] "39: -0.982027147234291"
[1] "40: -0.981066663566436"
[1] "41: -0.948610892555995"
[1] "42: -0.987959638999373"
[1] "43: -0.986636331232031"
[1] "44: -0.966329195551314"
[1] "45: -0.968426447209649"
[1] "46: -0.978740433128252"
[1] "47: -0.975228209859694"
[1] "48: -0.986388979890596"
[1] "49: -0.9796272295263"
[1] "50: -0.963360275557301"
[1] "51: -0.974050442538481"
[1] "52: -0.992476100959748"
[1] "53: -0.991916790480981"
[1] "54: -0.989616628548826"
```

```
[1] "55: -0.982568031811553"
[1] "56: -0.985176315214366"
[1] "57: -0.970482911848964"
[1] "58: -0.941957135387675"
[1] "59: -0.962293871753837"
[1] "60: -0.952883667735302"
[1] "61: -0.941851499753112"
[1] "62: -0.948084178685472"
[1] "63: -0.986474912022546"
[1] "64: -0.957039773515345"
[1] "65: -0.981315593788554"
[1] "66: -0.972625828564522"
[1] "67: -0.969348732979286"
[1] "68: -0.969819251549193"
[1] "69: -0.969125572569668"
[1] "70: -0.956048596845359"
[1] "71: -0.97268648960742"
[1] "72: -0.997361249856871"
[1] "73: -1.00834166930684"
[1] "74: -0.996251140314311"
[1] "75: -0.999708132573745"
[1] "76: -1.01149108042974"
[1] "77: -0.977549635285935"
[1] "78: -0.962834521884159"
[1] "79: -0.990795540425657"
[1] "80: -0.976763107853033"
[1] "81: -1.00054646192183"
[1] "82: -0.994993702718019"
[1] "83: -0.97015074302896"
[1] "84: -0.98619975057401"
[1] "85: -0.955905289184495"
[1] "86: -1.0020455188356"
[1] "87: -0.975713383432436"
[1] "88: -0.997461982950519"
[1] "89: -0.94605303708255"
[1] "90: -0.982910488211776"
[1] "91: -0.962157643916623"
[1] "92: -0.986896956849235"
[1] "93: -0.969207177335364"
[1] "94: -0.997756318549554"
[1] "95: -0.969951596850817"
[1] "96: -0.995062166615775"
[1] "97: -0.938651361489644"
```

```
[1] "98: -0.982396611879977"
[1] "99: -1.01140603160203"
[1] "100: -1.0032674170942"
[1] "101: -0.972529856717716"
[1] "102: -0.952451106028493"
[1] "103: -0.9785781300515"
[1] "104: -0.984137483800601"
[1] "105: -0.974141514471299"
[1] "106: -0.959806607272328"
[1] "107: -0.970600346806833"
[1] "108: -1.00917622292574"
[1] "109: -0.967991616642768"
[1] "110: -0.981438363587475"
[1] "111: -0.996759956878831"
[1] "112: -0.966562234836263"
[1] "113: -0.978821504214539"
[1] "114: -0.963173876261099"
[1] "115: -0.970549243821362"
[1] "116: -1.00280454634847"
[1] "117: -1.00956555740325"
[1] "118: -0.980441247357418"
[1] "119: -0.970231104156498"
[1] "120: -0.951527455998395"
[1] "121: -0.971144468635633"
[1] "122: -0.953347821455635"
[1] "123: -0.974231349648538"
[1] "124: -0.984728391209212"
[1] "125: -0.965259686992638"
[1] "126: -0.981958664040799"
[1] "127: -0.970702296514616"
[1] "128: -0.941360641188651"
[1] "129: -0.968127472820836"
[1] "130: -0.955377823388754"
[1] "131: -0.958398377760474"
[1] "132: -0.987809157274696"
[1] "133: -0.966617341849387"
[1] "134: -0.982086136025036"
[1] "135: -0.962887703880996"
[1] "136: -0.984888627056705"
[1] "137: -0.967100217763622"
[1] "138: -0.963608421527358"
[1] "139: -0.988336770711036"
[1] "140: -0.997015247228098"
```

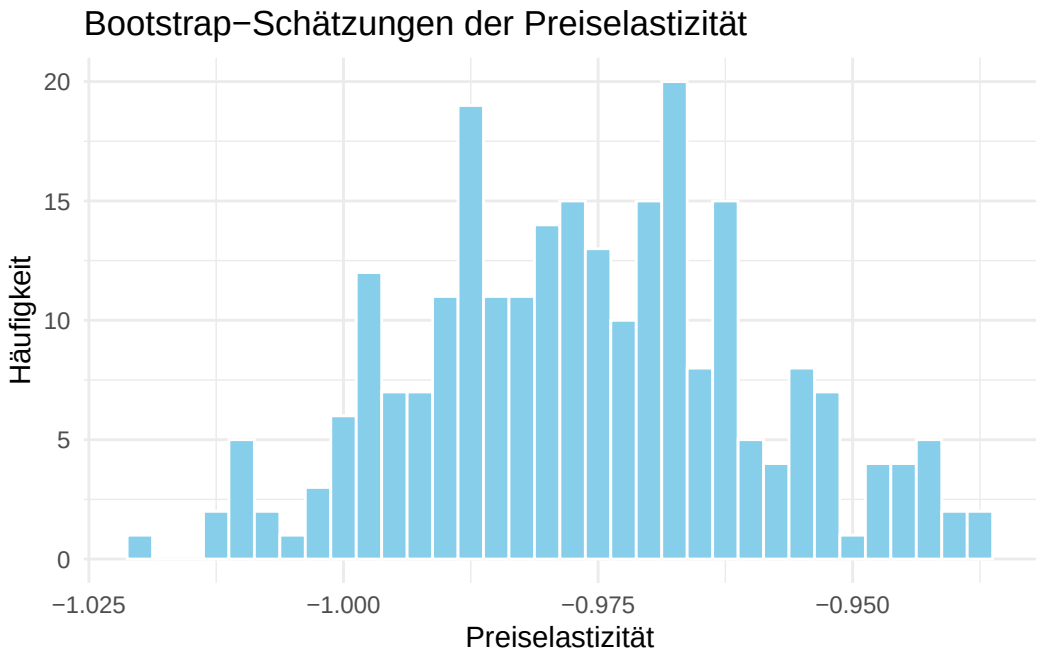
```
[1] "141: -0.962242037162242"
[1] "142: -0.94804271182187"
[1] "143: -0.987682876005877"
[1] "144: -1.00579936289414"
[1] "145: -1.00123296309192"
[1] "146: -0.962715666317734"
[1] "147: -0.961916417179089"
[1] "148: -0.988102803038353"
[1] "149: -0.987552822035592"
[1] "150: -1.00078601053638"
[1] "151: -0.959645427236212"
[1] "152: -0.971923477595026"
[1] "153: -0.947468147554072"
[1] "154: -0.978863825961521"
[1] "155: -0.965834747408714"
[1] "156: -0.978297054428174"
[1] "157: -0.985544983505552"
[1] "158: -0.986844762809482"
[1] "159: -0.945743328950989"
[1] "160: -0.974015927621696"
[1] "161: -0.982909731543127"
[1] "162: -0.950444387657035"
[1] "163: -0.996789048646788"
[1] "164: -0.974438126032873"
[1] "165: -0.959350667836062"
[1] "166: -0.955111483984079"
[1] "167: -0.962814720289746"
[1] "168: -0.98968513999801"
[1] "169: -0.976557260058653"
[1] "170: -0.936494535671749"
[1] "171: -0.96708581970603"
[1] "172: -0.966534008102023"
[1] "173: -0.977288941083373"
[1] "174: -0.979422456984673"
[1] "175: -0.979426849164042"
[1] "176: -0.989439257583809"
[1] "177: -0.979551286540642"
[1] "178: -0.957305002278556"
[1] "179: -0.989074990963872"
[1] "180: -0.991193644440372"
[1] "181: -0.957593397094416"
[1] "182: -0.987529963958672"
[1] "183: -0.984466259901984"
```

```
[1] "184: -0.998648534550322"
[1] "185: -1.02110281746278"
[1] "186: -0.971411953698632"
[1] "187: -0.97886615313371"
[1] "188: -0.953951493255626"
[1] "189: -0.955564273195618"
[1] "190: -0.96717725105963"
[1] "191: -0.999788984196158"
[1] "192: -0.976736556483124"
[1] "193: -0.981333034632516"
[1] "194: -0.943571145758864"
[1] "195: -0.986581362453005"
[1] "196: -0.976199308350931"
[1] "197: -0.98554884024415"
[1] "198: -0.953456348906701"
[1] "199: -0.991635692006662"
[1] "200: -0.968050162411491"
[1] "201: -0.975601808192935"
[1] "202: -0.985950783833544"
[1] "203: -0.967972027473878"
[1] "204: -0.952193357344384"
[1] "205: -0.955753042610033"
[1] "206: -0.968394677741138"
[1] "207: -0.994434827492686"
[1] "208: -1.00798040615671"
[1] "209: -0.970446028680622"
[1] "210: -0.964487986599085"
[1] "211: -0.976926122260962"
[1] "212: -0.971813217890043"
[1] "213: -0.987248772037937"
[1] "214: -0.976349798766977"
[1] "215: -0.978162450162162"
[1] "216: -0.966050877948019"
[1] "217: -0.997887035782867"
[1] "218: -0.968179330032835"
[1] "219: -0.939502196336215"
[1] "220: -0.972430631581329"
[1] "221: -0.976669562219033"
[1] "222: -0.962094875302316"
[1] "223: -0.967397557551146"
[1] "224: -0.977935496894783"
[1] "225: -0.965022573335234"
[1] "226: -0.99187686718488"
```

```
[1] "227: -0.987801470103923"
[1] "228: -0.973404848260776"
[1] "229: -0.989373602546248"
[1] "230: -0.991878648640432"
[1] "231: -0.965048648245697"
[1] "232: -0.968266532950016"
[1] "233: -0.9874196896686"
[1] "234: -0.994256936318991"
[1] "235: -0.990301996228811"
[1] "236: -0.961811659362211"
[1] "237: -0.984012030536918"
[1] "238: -0.976650613222543"
[1] "239: -0.980147264857819"
[1] "240: -0.980565495699639"
[1] "241: -0.988209441813063"
[1] "242: -0.941816869771374"
[1] "243: -0.979392519181992"
[1] "244: -0.974153584187677"
[1] "245: -0.968087643936417"
[1] "246: -0.986268378144932"
[1] "247: -1.00877627778987"
[1] "248: -0.969596639401388"
[1] "249: -0.965429918066697"
[1] "250: -0.995779789819656"
```

Wir erhalten variierende, immer etwas abweichende Werte, welche jedoch alle um den vermeintlich wahren Wert, welchen wir in unserer Stichprobe finden, streuen.

```
# Ergebnisse visualisieren
bootstrap_plot <- ggplot() +
    theme_minimal() +
    geom_histogram(aes(x = bootstrap_elastizität),
    binwidth = 0.0025, fill = "skyblue", color = "white") +
    labs(title = "Bootstrap-Schätzungen der Preiselastizität",
         x = "Preiselastizität", y = "Häufigkeit")
bootstrap_plot
```



Bootstrap Konfidenzintervall

Konfidenzintervall berechnen

quantile(bootstrap_elastizität, probs = c(0.025, 0.975))

Plot mit Konfidenzintervall und Schätzung

funktioniert, aber macht Probleme beim rendern zur PDF, deswegen auskommentiert

result_plot <- bootstrap_plot +

geom_vline(xintercept = mean(bootstrap_elastizität),

linetype = “dotted”, color = “purple”) +

geom_vline(xintercept = quantile(bootstrap_elastizität, probs = c(0.025, 0.975)),

linetype = “dashed”, color = “red”)

result_plot

```
Wir sehen, dass unser Konfidenzintervall um den wahren Wert streut, und dass die Schätzung i
```

```
## Vergleich mit traditionellem Konfidenzintervall

::: {.cell}

```{.r .cell-code}
# Konfidenzintervall mit normaler Verteilung
summary(regression3)
```

```
Call:
lm(formula = log(total_volume) ~ log(average_price) + type +
    region, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
```
```

```
-4.4086 -0.2388  0.0036  0.2555  2.2269

Coefficients:
                           Estimate Std. Error  t value Pr(>|t|)
(Intercept)               11.547369   0.023926  482.628  < 2e-16 ***
log(average_price)        -0.978088   0.016554  -59.084  < 2e-16 ***
typeorganic               -3.303655   0.008685 -380.371  < 2e-16 ***
regionAtlanta              1.515884   0.033160   45.714  < 2e-16 ***
regionBaltimoreWashington  2.237114   0.033031   67.727  < 2e-16 ***
regionBoise               -0.163731   0.033167   -4.937 8.02e-07 ***
regionBoston               1.742457   0.033033   52.749  < 2e-16 ***
regionBuffaloRochester     0.537592   0.033033   16.275  < 2e-16 ***
regionCalifornia           4.120739   0.033103  124.482  < 2e-16 ***
regionCharlotte            1.008430   0.033031   30.530  < 2e-16 ***
regionChicago              2.412701   0.033031   73.045  < 2e-16 ***
regionCincinnatiDayton     1.234517   0.033351   37.016  < 2e-16 ***
regionColumbus             0.728672   0.033254   21.912  < 2e-16 ***
regionDallasFtWorth        2.070490   0.033628   61.570  < 2e-16 ***
regionDenver               2.195506   0.033314   65.904  < 2e-16 ***
regionDetroit              1.456655   0.033221   43.848  < 2e-16 ***
regionGrandRapids          0.381429   0.033040   11.544  < 2e-16 ***
regionGreatLakes           3.724070   0.033128  112.413  < 2e-16 ***
regionHarrisburgScranton   1.077646   0.033035   32.621  < 2e-16 ***
regionHartfordSpringfield  1.393635   0.033107   42.095  < 2e-16 ***
regionHouston              2.043074   0.033755   60.527  < 2e-16 ***
regionIndianapolis         0.506389   0.033160   15.271  < 2e-16 ***
regionJacksonville         0.445170   0.033042   13.473  < 2e-16 ***
regionLasVegas             1.216328   0.033135   36.708  < 2e-16 ***
regionLosAngeles           3.282953   0.033347   98.449  < 2e-16 ***
regionLouisville          -0.053508   0.033205   -1.611    0.107
regionMiamiFtLauderdale    1.055891   0.033069   31.930  < 2e-16 ***
regionMidsouth             3.602943   0.033075  108.931  < 2e-16 ***
regionNashville            0.798255   0.033333   23.948  < 2e-16 ***
regionNewOrleansMobile     0.611929   0.033177   18.444  < 2e-16 ***
regionNewYork              2.942602   0.033065   88.994  < 2e-16 ***
regionNortheast            3.975746   0.033033  120.358  < 2e-16 ***
regionNorthernNewEngland   1.592448   0.033044   48.192  < 2e-16 ***
regionOrlando              1.058020   0.033041   32.021  < 2e-16 ***
regionPhiladelphia         1.650033   0.033038   49.943  < 2e-16 ***
regionPhoenixTucson        1.830051   0.033488   54.648  < 2e-16 ***
regionPittsburgh           0.311028   0.033095    9.398  < 2e-16 ***
regionPlains               2.995033   0.033065   90.580  < 2e-16 ***
regionPortland             2.117313   0.033205   63.764  < 2e-16 ***
```

```
regionRaleighGreensboro    1.275913   0.033032   38.627  < 2e-16 ***
regionRichmondNorfolk      1.029428   0.033180   31.026  < 2e-16 ***
regionRoanoke              0.681278   0.033247   20.491  < 2e-16 ***
regionSacramento           1.481435   0.033032   44.849  < 2e-16 ***
regionSanDiego             1.705596   0.033117   51.502  < 2e-16 ***
regionSanFrancisco         2.388737   0.033084   72.203  < 2e-16 ***
regionSeattle              2.339973   0.033080   70.738  < 2e-16 ***
regionSouthCarolina        1.254773   0.033086   37.925  < 2e-16 ***
regionSouthCentral         3.703322   0.033576  110.298  < 2e-16 ***
regionSoutheast            3.382530   0.033088  102.227  < 2e-16 ***
regionSpokane              0.134868   0.033080    4.077 4.58e-05 ***
regionStLouis              0.795652   0.033081   24.052  < 2e-16 ***
regionSyracuse            -0.154400   0.033032   -4.674 2.97e-06 ***
regionTampa                0.921008   0.033083   27.840  < 2e-16 ***
regionTotalUS              5.876608   0.033157  177.234  < 2e-16 ***
regionWest                 4.232687   0.033252  127.293  < 2e-16 ***
regionWestTexNewMexico     1.824493   0.033398   54.628  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4294 on 18193 degrees of freedom
Multiple R-squared:  0.9651,    Adjusted R-squared:  0.9649
F-statistic:  9135 on 55 and 18193 DF,  p-value: < 2.2e-16
```

```r
estimate <- summary(regression3)$coefficients[2, 1]
estimate_sd <- summary(regression3)$coefficients[2, 2]
ci_normal <- estimate + c(-1, 1) * qnorm(0.975) * estimate_sd
ci_normal
```

```
[1] -1.0105342 -0.9456427
```

```r
# funktioniert, aber macht Probleme beim rendern zur PDF, deswegen auskommentiert
# comparison_plot <- result_plot +
#     geom_vline(xintercept = ci_normal,
#                linetype = "dashed", color = "orange")
# comparison_plot
```

:::

Wir sehen, dass wir mit Hilfe unseres Bootstraps eine gute Schätzung für das Konfidenzintervall erhalten haben, welche dem traditionellen Konfidenzintervall ähnlich ist. Dies ist ein gutes Zeichen, dass unser Bootstrap-Verfahren funktioniert hat. Das positive daran ist, dass

unser Bootstrap völlig ohne Verteilungsannahme funktioniert, und lediglich auf unseren Daten basiert, was es zu einem sehr nützlichen Werkzeug für uns macht.