

Tut-8

Diff in Diff

Im folgenden haben Sie diese Tabelle gegeben

```
library(data.table)
```

Warning: Paket 'data.table' wurde unter R Version 4.2.3 erstellt

```
data <- data.table(  
  Gruppe = c("Treatment", "Treatment", "Control", "Control"),  
  Zeit = c("Vorher", "Nacher", "Vorher", "Nacher"),  
  Wert = c(70, 85, 75, 80)  
)  
  
data
```

	Gruppe	Zeit	Wert
1:	Treatment	Vorher	70
2:	Treatment	Nacher	85
3:	Control	Vorher	75
4:	Control	Nacher	80

Berechnen Sie den Difference-in-Differences Schätzer.

```
Diff1 <- data$Wert[data$Gruppe == "Treatment" & data$Zeit == "Nacher"] -  
  data$Wert[data$Gruppe == "Treatment" & data$Zeit == "Vorher"]  
  
Diff2 <- data$Wert[data$Gruppe == "Control" & data$Zeit == "Nacher"] -  
  data$Wert[data$Gruppe == "Control" & data$Zeit == "Vorher"]
```

```
DiD <- Diff1 - Diff2
```

```
DiD
```

```
[1] 10
```

Antwort: Der Diff in Diff Schätzer zeigt einen kausalen Effekt von 10 Einheiten.

Erstellen Sie ein data.table Objekt mit den Daten in R und transformieren Sie die Daten geeignet, sodass Sie die lm() Funktion verwenden können, um den Effekt zu bestimmen.

$$Y_{it} = \beta_1 \text{Behandlung}_i + \beta_2 \text{Nach}_t + \beta_3 (\text{Behandlung}_i \times \text{Nach}_t)$$

Siehe Foliensatz 31.

```
# dummy variable erzeugen
data[, Dummy_Nach := Zeit == "Nacher"]
data[, Dummy_Treat := Gruppe == "Treatment"]

mod <- lm(Wert ~ Dummy_Treat * Dummy_Nach, data = data)

summary(mod) # Interaktionskoeffizient ist DiD Effekt
```

Call:

```
lm(formula = Wert ~ Dummy_Treat * Dummy_Nach, data = 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
Dummy_TreatTRUE	-5	NaN	NaN	NaN
Dummy_NachTRUE	5	NaN	NaN	NaN
Dummy_TreatTRUE:Dummy_NachTRUE	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

```
mod$coef[4]
```

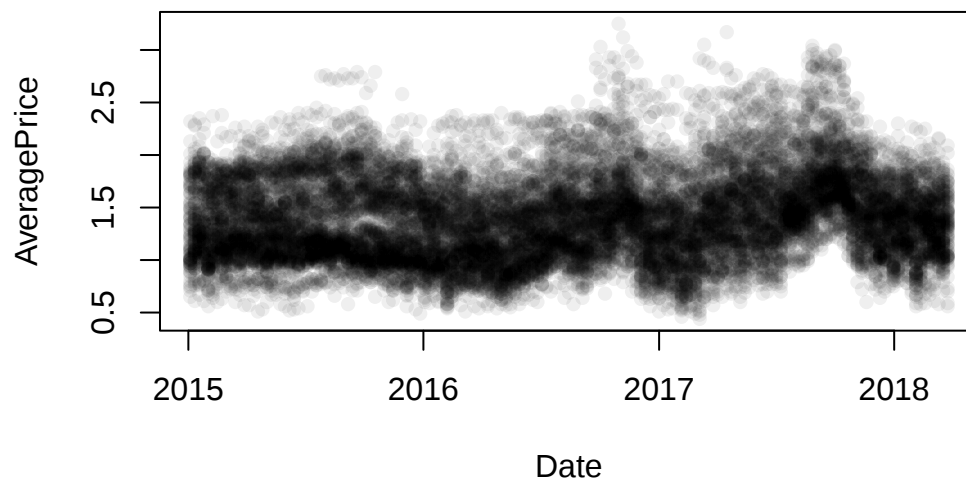
```
Dummy_TreatTRUE:Dummy_NachTRUE  
10
```

Koeffizient 4 ist unser Diff in Diff Effekt.

Bootstrapping

Laden Sie den avocado-Datensatz programmatisch hier herunter: <https://statistik.julianhinz.com/tutorials/tuto>.
Inspizieren Sie die Struktur und visualisieren Sie die Variation über die Zeit. Gibt es große geografische Unterschiede in Preisen und nachgefragten Mengen?

```
avocado <- read.csv("avocado.csv")  
  
attach(avocado)  
  
Date <- as.Date(Date)  
  
plot(Date, AveragePrice, pch = 16, col = "#00000010")
```



Data Vis

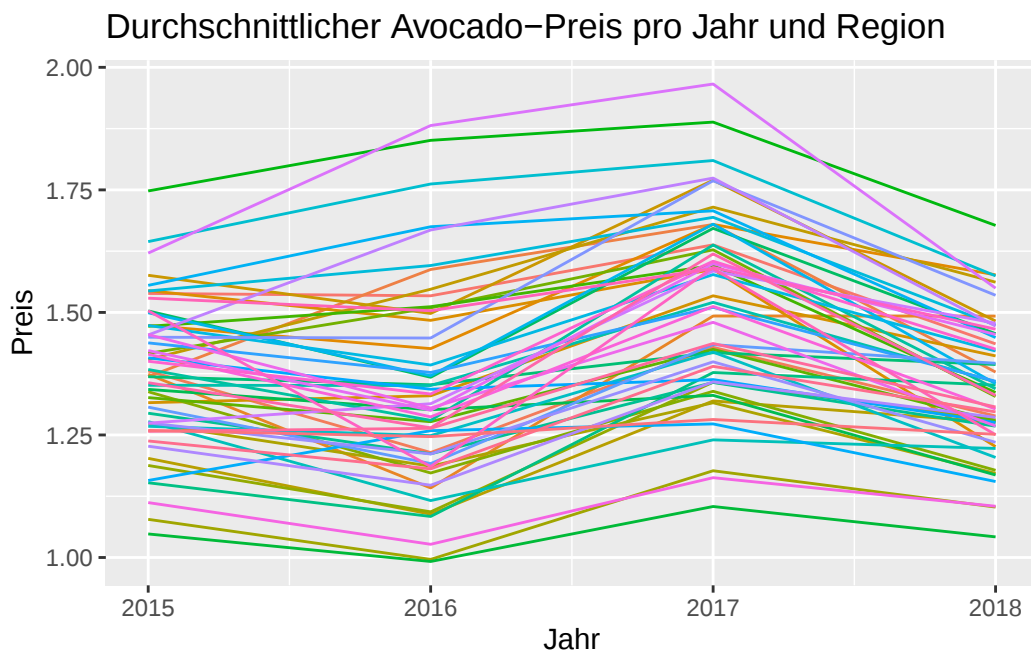
```
library(ggplot2)

avg_yearly <- aggregate(cbind(AveragePrice, Total.Volume) ~ region + year,
                        data = avocado, FUN = mean)

avg_yearly$Total.Volume <- avg_yearly$Total.Volume / 1000

avg_yearly$year <- as.numeric(avg_yearly$year)

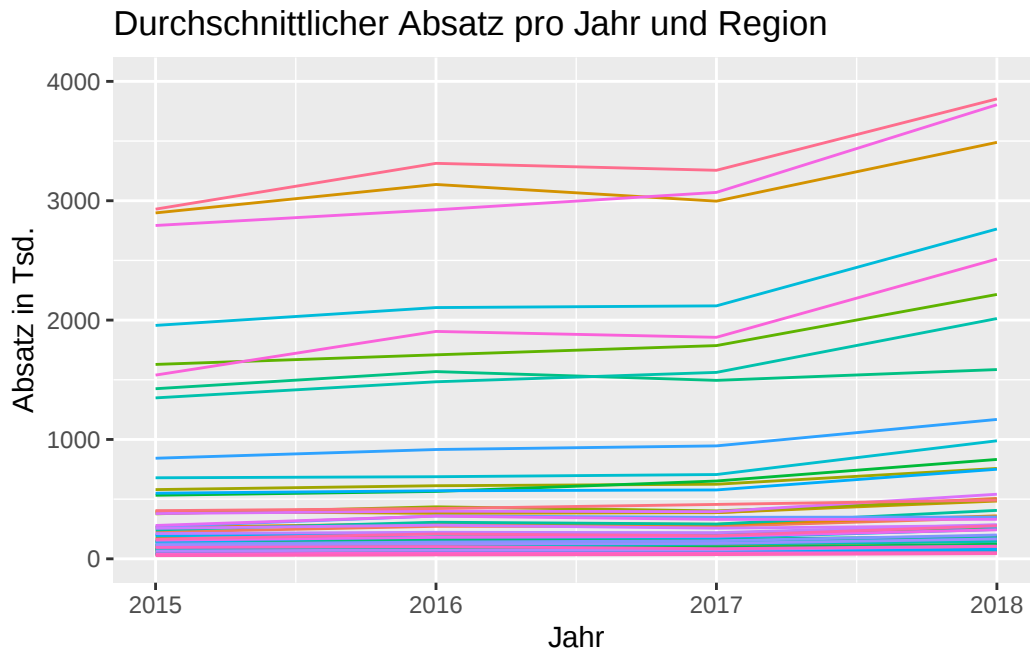
ggplot(avg_yearly, aes(x = year, y = AveragePrice, color = region)) +
  geom_line() +
  theme(legend.position = "none") +
  labs(title = "Durchschnittlicher Avocado-Preis pro Jahr und Region",
       x = "Jahr", y = "Preis")
```



```
ggplot(avg_yearly, aes(x = year, y = Total.Volume, color = region)) +
  geom_line() +
  ylim(0, 4000) +
```

```
theme(legend.position = "none") +
labs(title = "Durchschnittlicher Absatz pro Jahr und Region",
     x = "Jahr", y = "Absatz in Tsd.")
```

Warning: Removed 4 rows containing missing values or values outside the scale range (`geom_line()`).



Sowohl bei den Preisen, als auch beiden abgesetzten Mengen gibt es teilweise große Abweichungen in den unterschiedlichen Regionen.

Regression

```
mod1 <- lm(log(Total.Volume) ~ log(AveragePrice))
summary(mod1)
```

Call:

```
lm(formula = log(Total.Volume) ~ log(AveragePrice))
```

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(AveragePrice)	-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

```
mod2 <- lm(log(Total.Volume) ~ log(AveragePrice) * type)
summary(mod2)
```

Call:

```
lm(formula = log(Total.Volume) ~ log(AveragePrice) * type)
```

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(AveragePrice)	-1.42676	0.06188	-23.056	< 2e-16 ***
typeorganic	-3.27878	0.03682	-89.059	< 2e-16 ***
log(AveragePrice):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

Durch logarithmieren können die Koeffizienten als Elastizität interpretiert werden.

Mod 1: Ein 1% Anstieg des Durchschnittspreises führt zu einem 4,6% Abfall im Verkaufsvolumen.

Mod 2: Organische Avocado haben eine niedrigere Baseline als konventionelle Avocado (Koeffizient 3). Die Preiselastizität für konventionelle Avocado liegt bei -1,42 (Koeffizient 2) und die der organischen Avocado bei -1,14 (Koeffizient 2 + Koeffizient 4). Die Nachfrage der organischen Avocado ist also weniger Stark von einer Preis Erhöhung betroffen als die konventionelle.

Wir sehen außerdem, dass Mod 2 mehr Variation im Preis erklärt als Mod 1 (Siehe R-squared).

Bootstrap

```
set.seed(95)

n <- 100
bootstrap_elastizität = numeric(n)

# Ziehen mit zurücklegen für n = nrow(avocado) Datenpunkte
for (i in 1:n) {
  bootstrap_data <- avocado[sample(1:nrow(avocado), size = nrow(avocado), replace = TRUE),
  bootstrap_model <- lm(log(Total.Volume) ~ log(AveragePrice) * type,
                        data = bootstrap_data)
  bootstrap_elastizität[i] <- coef(bootstrap_model)[2]
  print(paste0(i, ": ", bootstrap_elastizität[i]))
}
```

```
[1] "1: -1.42808586330744"
[1] "2: -1.48388965210751"
[1] "3: -1.3902764498311"
[1] "4: -1.48550142670875"
[1] "5: -1.38131906041093"
[1] "6: -1.3760882984717"
[1] "7: -1.37642372188468"
[1] "8: -1.55854171397748"
[1] "9: -1.39303146504359"
[1] "10: -1.41617560428037"
[1] "11: -1.41708120325651"
[1] "12: -1.43693395627114"
[1] "13: -1.34379386317336"
[1] "14: -1.39276592432598"
[1] "15: -1.41446560044372"
[1] "16: -1.41181140382497"
```

[1] "17: -1.36986112132787"
[1] "18: -1.38098922203334"
[1] "19: -1.36237307472744"
[1] "20: -1.4293711360502"
[1] "21: -1.45169962700112"
[1] "22: -1.48982818506612"
[1] "23: -1.46462059857713"
[1] "24: -1.4935608600037"
[1] "25: -1.36737608549089"
[1] "26: -1.46576358465333"
[1] "27: -1.43897052621772"
[1] "28: -1.45412781620722"
[1] "29: -1.45284920526047"
[1] "30: -1.40683093509057"
[1] "31: -1.40363112664124"
[1] "32: -1.44511808821665"
[1] "33: -1.41094630451115"
[1] "34: -1.42511894451115"
[1] "35: -1.40082456971728"
[1] "36: -1.51106636279743"
[1] "37: -1.4505202401178"
[1] "38: -1.45166773009232"
[1] "39: -1.45311581199148"
[1] "40: -1.44693090708329"
[1] "41: -1.47181278357003"
[1] "42: -1.41114533367884"
[1] "43: -1.3595756615391"
[1] "44: -1.54117941707179"
[1] "45: -1.4790922117276"
[1] "46: -1.40154685377996"
[1] "47: -1.36826136033006"
[1] "48: -1.45061045100008"
[1] "49: -1.45371111936894"
[1] "50: -1.5174250968345"
[1] "51: -1.4777234086795"
[1] "52: -1.38585359441526"
[1] "53: -1.31351172433486"
[1] "54: -1.41521132463993"
[1] "55: -1.40736039207315"
[1] "56: -1.38412695465463"
[1] "57: -1.32510040823436"
[1] "58: -1.36064713554169"
[1] "59: -1.47367637915937"

[1] "60: -1.44887486761214"
[1] "61: -1.38957674872286"
[1] "62: -1.39933880160836"
[1] "63: -1.43558286476006"
[1] "64: -1.40390194103394"
[1] "65: -1.38599616795311"
[1] "66: -1.41854976606364"
[1] "67: -1.46676561863168"
[1] "68: -1.38127528628081"
[1] "69: -1.40184788720132"
[1] "70: -1.39761070413646"
[1] "71: -1.45689261904899"
[1] "72: -1.44584558747154"
[1] "73: -1.45386562484011"
[1] "74: -1.3757094267707"
[1] "75: -1.43457091679939"
[1] "76: -1.39506286154902"
[1] "77: -1.4778889381282"
[1] "78: -1.37639631394593"
[1] "79: -1.4532537074521"
[1] "80: -1.33400082574505"
[1] "81: -1.48755102242315"
[1] "82: -1.38566965571817"
[1] "83: -1.48998398640302"
[1] "84: -1.42031772878384"
[1] "85: -1.41598210005628"
[1] "86: -1.47779633602968"
[1] "87: -1.43859689261384"
[1] "88: -1.40688902552009"
[1] "89: -1.47834942927921"
[1] "90: -1.32470304087947"
[1] "91: -1.39531737051606"
[1] "92: -1.45326946993124"
[1] "93: -1.42554165727478"
[1] "94: -1.40746359089555"
[1] "95: -1.37377278040811"
[1] "96: -1.44908650190793"
[1] "97: -1.49657836302242"
[1] "98: -1.40048554023677"
[1] "99: -1.42864405351597"
[1] "100: -1.43933281455555"

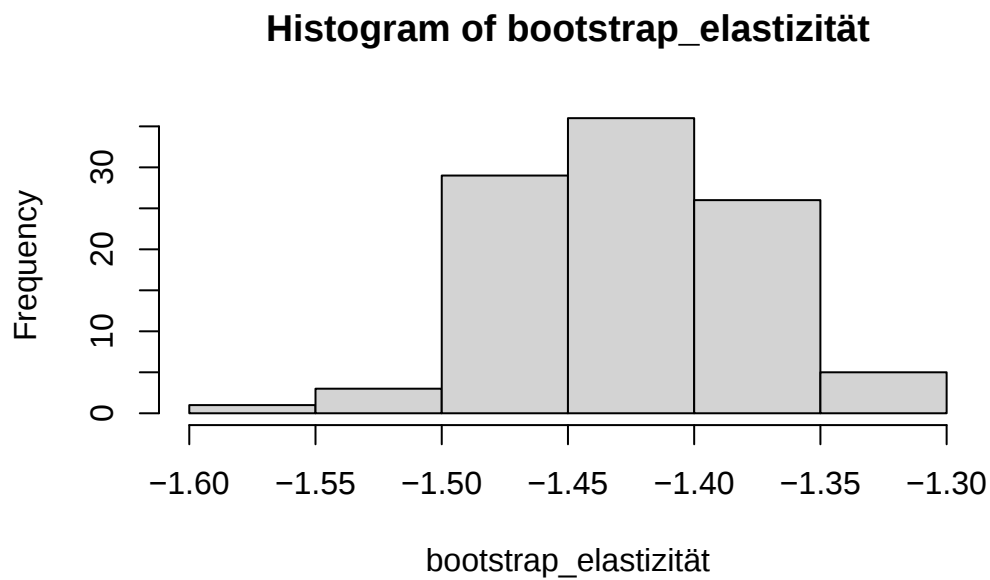
```
mod2$coef[2]
```

```
log(AveragePrice)  
-1.426764
```

```
mean(bootstrap_elastizität)
```

```
[1] -1.424591
```

```
hist(bootstrap_elastizität)
```



Der Bootstrap bestätigt unsere Modellelastizität.

Konfidenzintervall

Berechnen Sie das 95%-Konfidenzintervall für die Preiselastizität unter der Annahme einer Normalverteilung. Vergleichen Sie dieses Intervall mit dem Bootstrap-Konfidenzintervall.

```
# Bootstrap  
BS <- quantile(bootstrap_elastizität, probs = c(0.025, 0.975))  
BS
```

	2.5%	97.5%
	-1.514405	-1.329328

```
# Normal
```

```
CI <- c(mod2$coef[2] - qnorm(0.975) * summary(mod2)$coefficients[2, 2], mod2$coef[2] + qnorm(0.975) * summary(mod2)$coefficients[2, 2])
```

log(AveragePrice)	log(AveragePrice)
-1.548049	-1.305478