Example regression analysis (Data: Official Results of the 2021 German Federal Election)

Susumu Shikano

Last compiled at 13. Juli 2022

Preparing data

We download the official results of 2021 German Federal Election.

```
library(foreign)

url.result <- "https://www.bundeswahlleiter.de/dam/jcr/fc2afe29-c086-43eb-bc34-a48356dee154/btw21_kerg.
# reading the results
rawdata <- read.csv(url.result,
                    skip=5,sep=";",header = FALSE,encoding="UTF-8")

# reading the labels
main.labels <- read.csv(url.result,
                        skip=2,nrows=1,sep=";",header = FALSE,
                        colClasses = "character",encoding="UTF-8")

# reading the second line of the labels
first.second.votes <- read.csv(url.result,
                               skip=3,nrows=1,sep=";",header = FALSE)
first.second.votes[1:3] <- "aaa"

# delete previous results and state-level results
result21 <- rawdata[,!is.na(first.second.votes)]
result21 <- result21[result21[,3]!=99 & !is.na(result21[,3]),]

# keep previous results and delete stat-level results
result17 <- rawdata[,is.na(first.second.votes)|first.second.votes=="aaa"]
result17 <- result17[result17[,3]!=99 & !is.na(result17[,3]),]
result17 <- result17[,1:ncol(result21)]

# make the labels
main.labels <- main.labels[!is.na(first.second.votes)]
main.labels[seq(5,length(main.labels),by=2)] <-    main.labels[seq(4,length(main.labels)-1,by=2)] # add

main.labels <- paste(main.labels,first.second.votes[!is.na(first.second.votes)])
main.labels <- gsub("aaa","",main.labels)

main.labels <- gsub("ä","ae",main.labels)
main.labels <- gsub("Ä","Ae",main.labels)
main.labels <- gsub("ö","oe",main.labels)
main.labels <- gsub("Ö","Oe",main.labels)
```

```
main.labels <- gsub("ü","ue",main.labels)
main.labels <- gsub("Ü","Ue",main.labels)

colnames(result21) <- colnames(result17) <- main.labels
```

After data cleaning, the dataset contains 299 electoral districts as unit.

We further download socio-demographic data for the districts.

```
url.sociodem <- "https://www.bundeswahlleiter.de/dam/jcr/b1d3fc4f-17eb-455f-a01c-a0bf32135c5d/btw21_stru
rawdata <- read.csv(url.sociodem,
                    skip=8,sep=";",dec = ",",
                    header = TRUE,encoding="UTF-8")

# delete the state-level results
sociodem <- rawdata[rawdata[,2]<500,]
```

Obtaining variables as vectors

- old.gen: percentage of the older generation (age 60+)
- gdp: GDP per capita
- eligible: number of eligible citizens to vote
- valid: number of valid votes
- tunrout: turnout rate (in %)
- union.pr: vote share of CDU/CSU (in %)

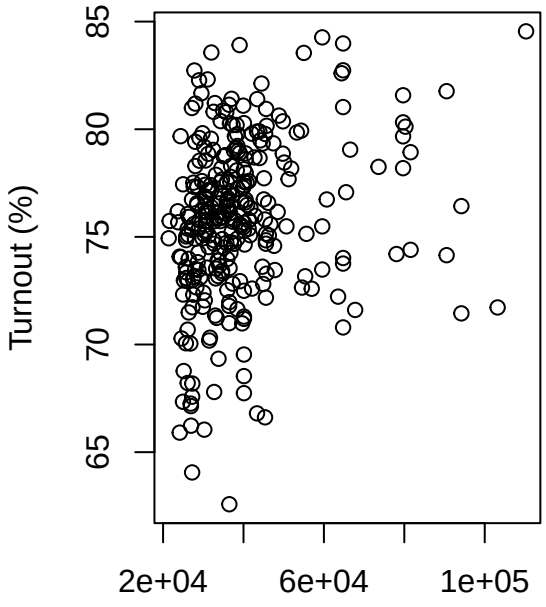
```
gdp <- sociodem[,grep("Bruttoinlandsprodukt",names(sociodem))]

old.gen <- sociodem[,grep("Alter",names(sociodem))]
old.gen <- old.gen[,5] + old.gen[,6]
young.gen <- sociodem[,grep("Alter",names(sociodem))]
young.gen <- young.gen[,2]

abi <- sociodem[,grep("allgemeiner.und.Fachhochschulreife",names(sociodem))]



unemployment <- sociodem[,grep("Arbeitslosenquote",names(sociodem))]
unemp.total <- unemployment[,grep("insgesamt",names(unemployment))]
unemp.young <- unemployment[,grep("15.bis.24.Jahre",names(unemployment))]

pkw <- sociodem[,grep("PKW",names(sociodem))]
pkw.e <- pkw[,grep("Elektro",names(pkw))]
pkw <- pkw[,grep("insgesamt",names(pkw))]

company <- sociodem[,grep("Unternehmen.insgesamt",names(sociodem))]

## from here election results

eligible <- result21[,grep("Wahlberechtigte Zweitstimmen",names(result21))]
eligible.17 <- result17[,grep("Wahlberechtigte Zweitstimmen",names(result17))]
valid <- result21[,grep("Gueltige Stimmen Zweitstimmen",names(result21))]
valid.17 <- result17[,grep("Gueltige Stimmen Zweitstimmen",names(result17))]

turnout <- valid *100/ eligible
```

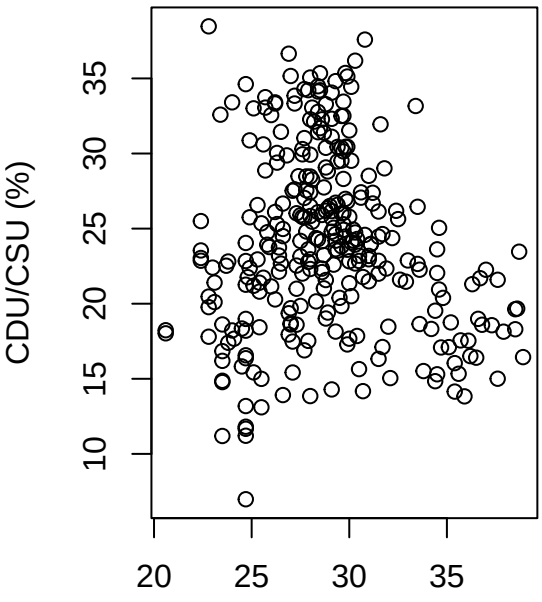
```
turnout.17 <- valid.17 *100/ eligible.17

cdu.pr <- result21[,grep("Christlich Demokratische Union Deutschlands Zweitstimmen",names(result21))]

csu.pr <- result21[,grep("Christlich-Soziale Union in Bayern e.V. Zweitstimmen",names(result21))]

union.pr <- cdu.pr
union.pr[is.na(union.pr)] <- csu.pr[is.na(union.pr)]

union.pr <- union.pr*100/valid

spd.pr <- result21[,grep("Sozialdemokratische Partei Deutschlands Zweitstimmen",names(result21))]
spd.pr <- spd.pr *100/valid


gru.pr <- result21[,grep("BUeNDNIS 90/DIE GRUeNEN Zweitstimmen",names(result21))]
gru.pr <- gru.pr *100/valid

fdp.pr <- result21[,grep("Freie Demokratische Partei Zweitstimmen",names(result21))]
fdp.pr <- fdp.pr *100/valid

afd.pr <- result21[,grep("Alternative fuer Deutschland Zweitstimmen",names(result21))]
afd.pr <- afd.pr *100/valid

afd.pr.17 <- result21[,grep("Alternative fuer Deutschland Zweitstimmen",names(result17))]
afd.pr.17 <- afd.pr.17 *100/valid.17


gru.smd <- result21[,grep("BUeNDNIS 90/DIE GRUeNEN Erststimmen",names(result21))]
gru.smd <- gru.smd *100/valid

gru.smd.17 <- result17[,grep("BUeNDNIS 90/DIE GRUeNEN Erststimmen",names(result17))]
gru.smd.17 <- gru.smd.17 *100/valid.17
```

Joint distributions

We observe two joint distributions:

```
par(mfrow=c(1,2))
plot(gdp,turnout,xlab="GDP per capita",ylab="Turnout (%)")
plot(old.gen,union.pr,xlab="Age 60+ (%)",ylab="CDU/CSU (%)")
```





Variances and covariances

```
cov(cbind(gdp,old.gen,turnout,union.pr))
```

```
##                   gdp       old.gen      turnout     union.pr
## gdp      209743186.521 -33559.517877 12568.976282 -2528.212814
## old.gen     -33559.518     13.322507    -4.998972    -2.216658
## turnout      12568.976     -4.998972    14.221570     9.328584
## union.pr     -2528.213     -2.216658     9.328584    35.588193
```

The diagonal elements are variances. The off-diagonal elements are covariances. Here, you have to be able to calculate the slope of the regression lines. Calculate the regression slopes (Turnout on GDP and CDU/CSU on older generation).

Calculating the regression slopes

To obtain the regression slope, we can divide covariance of the dependent and independent variable by the variance of independent variable. We now estimate the slop for the following regression model:

- turnout = β_0 + β_1 gdp + u

```
cov(turnout,gdp)/var(gdp)
```

```
## [1] 5.992555e-05
```

- union.pr = β_0 + β_1 old.gen + u

```
cov(union.pr,old.gen)/var(old.gen)
```

```
## [1] -0.1663845
```

Simple regression models (Turnout on GDP)

```
lm.out.1 <- lm(turnout ~ gdp)
summary(lm.out.1)
```

```
##
## Call:
## lm(formula = turnout ~ gdp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0464  -2.0181   0.3747   2.3607   8.2044
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.344e+01  6.170e-01 119.021  < 2e-16 ***
## gdp         5.993e-05  1.470e-05   4.075  5.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.676 on 297 degrees of freedom
## Multiple R-squared:  0.05296,    Adjusted R-squared:  0.04977
## F-statistic: 16.61 on 1 and 297 DF,  p-value: 5.9e-05
```

You can check whether your hand-calculated regression slope corresponds to the output.

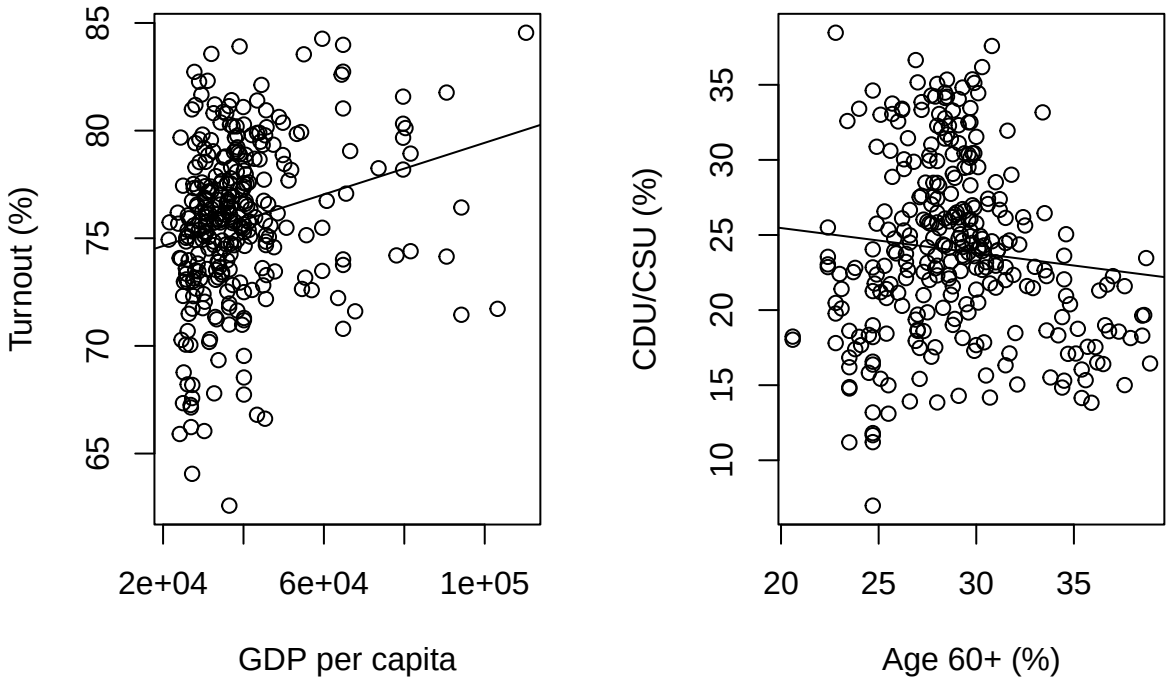
Simple regression models (Vote for CDU/CSU on age 60+)

```
lm.out.2 <- lm(union.pr ~ old.gen)
summary(lm.out.2)
```

```
##
## Call:
## lm(formula = union.pr ~ old.gen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.7003  -4.1639  -0.3729   3.7572  13.9209
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.78957    2.74481  10.489   <2e-16 ***
## old.gen     -0.16638    0.09435  -1.764   0.0788 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.945 on 297 degrees of freedom
## Multiple R-squared:  0.01036,    Adjusted R-squared:  0.007031
## F-statistic:  3.11 on 1 and 297 DF,  p-value: 0.07883
```

Simple regression models (graphical presentation)

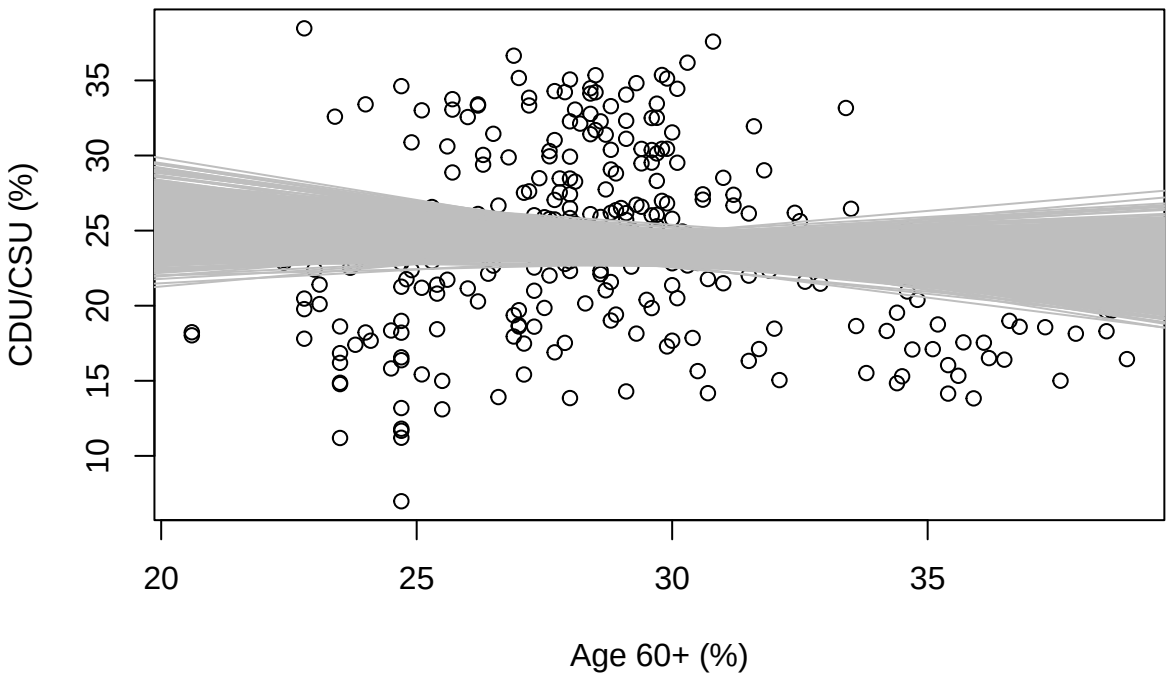
```
par(mfrow=c(1,2))
plot(gdp,turnout,xlab="GDP per capita",ylab="Turnout (%)")
abline(lm.out.1)
plot(old.gen,union.pr,xlab="Age 60+ (%)",ylab="CDU/CSU (%)")
abline(lm.out.2)
```



Regression parameters as random parameters

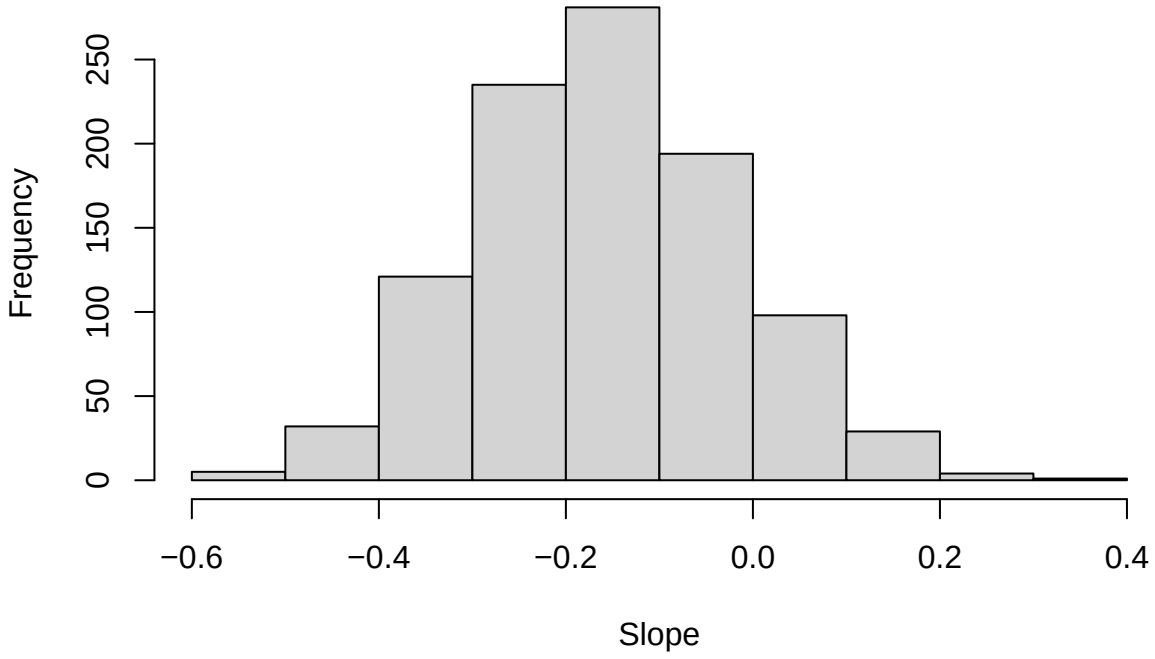
Now, we change the perspective and treat our data as population, from which we draw random samples. More concretely, we randomly sample 100 districts from 299 all districts and draw the regression line. This is repeated for 1000 times.

```
n.iter <- 1000
sample.size <- 100

estimates <- matrix(NA,ncol=2,nrow=n.iter)
plot(old.gen,union.pr,xlab="Age 60+ (%)",ylab="CDU/CSU (%)")
for (i in 1:n.iter){
  this.sample <- sample(1:length(union.pr),size = sample.size)
  this.lm.out <- lm(union.pr[this.sample] ~ old.gen[this.sample])
  estimates[i,] <- coefficients(this.lm.out)
  abline(this.lm.out,col="grey")
}
```



We can observe the distribution of slopes based on 1000 samples.

```
hist(estimates[,2],xlab="Slope",main="")
```

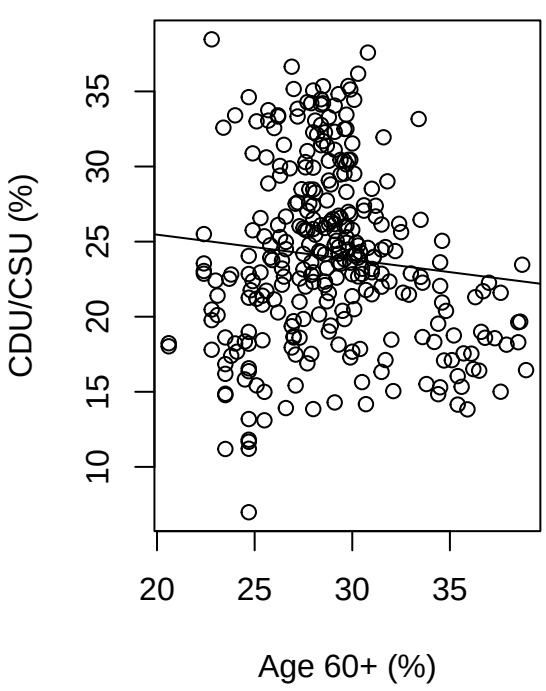


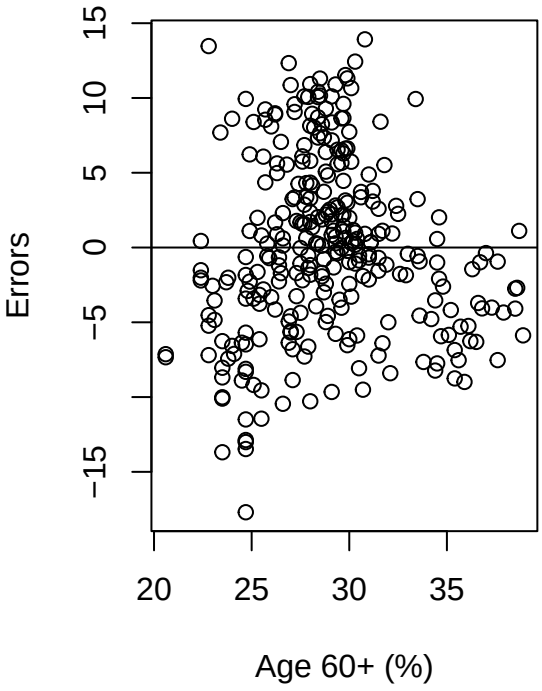
The mean of this distribution is -0.1598.

This estimator (estimand: the regression line based on all 299 districts) is likely to be biased due to the violation of the zero conditional mean assumption.

To see this, we can check the joint distribution of the errors and the independent variable (Age 60+).

```
par(mfrow=c(1,2))
plot(old.gen,union.pr,xlab="Age 60+ (%)",ylab="CDU/CSU (%)")
abline(lm.out.2)

errors <- lm.out.2$residuals

plot(old.gen,errors,xlab="Age 60+ (%)",ylab="Errors")
abline(h=0)
```



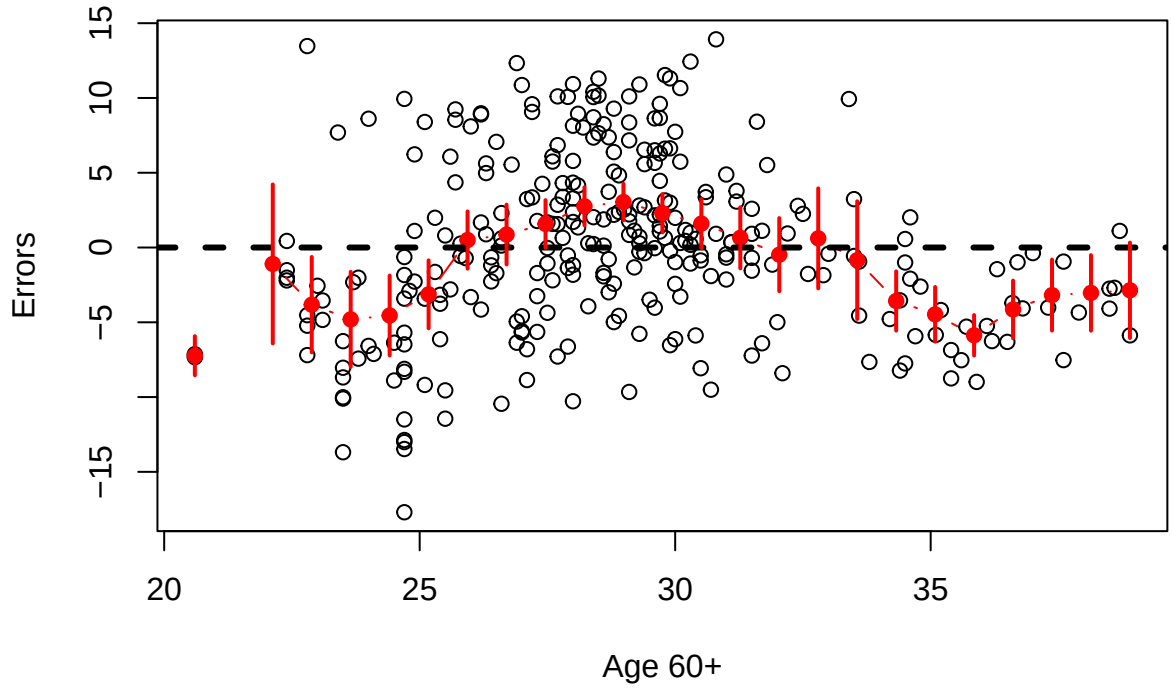


If we calculate the mean errors for different x values, it is clearly to see that the estimated regression line tends to over-predict the CDU/CSU share for the districts with about 25% and those with 35% or more.

```
y.range <- range(errors)
x.range <- range(old.gen)

plot(errors ~ old.gen,ylab="Errors",xlab="Age 60+",
     xlim=x.range,ylim=y.range)

x.values <- seq(min(old.gen),max(old.gen),length=25)
x.interval <- x.values[2] -x.values[1]

conditional.mean <- lower.b <- upper.b <- rep(NA,length(x.values))
for (i in 1:length(conditional.mean)){

  selected.error <- errors[(old.gen>(x.values[i]-x.interval)) &
                                          (old.gen<(x.values[i]+x.interval)) ]
  conditional.mean[i] <- mean(selected.error)
  this.ci <- ci.sample.mean(selected.error)
  lower.b[i] <- this.ci$lower.b
  upper.b[i] <- this.ci$upper.b

}
```

```
## Warning in qt(bounds.prob, df = length(x) - 1): NaNs wurden erzeugt
```

```
par(new=T)
plot(conditional.mean ~ x.values,ann=F,axes=F,
     xlim=x.range,ylim=y.range,
     col="red",pch=19,type="b")
abline(h=0,lty=2,lwd=3)
for (i in 1:length(conditional.mean)){
  lines(rep(x.values[i],2),c(upper.b[i],lower.b[i]),col="red",lwd=2)
}
```



Furthermore, the homoscedasticity assumption seems to be violated, as well. The red points in the below figure show the standard deviations of the errors for different x values with their 95%-confidence intervals (vertical red lines). Accordingly, the error variance is larger for smaller x values.

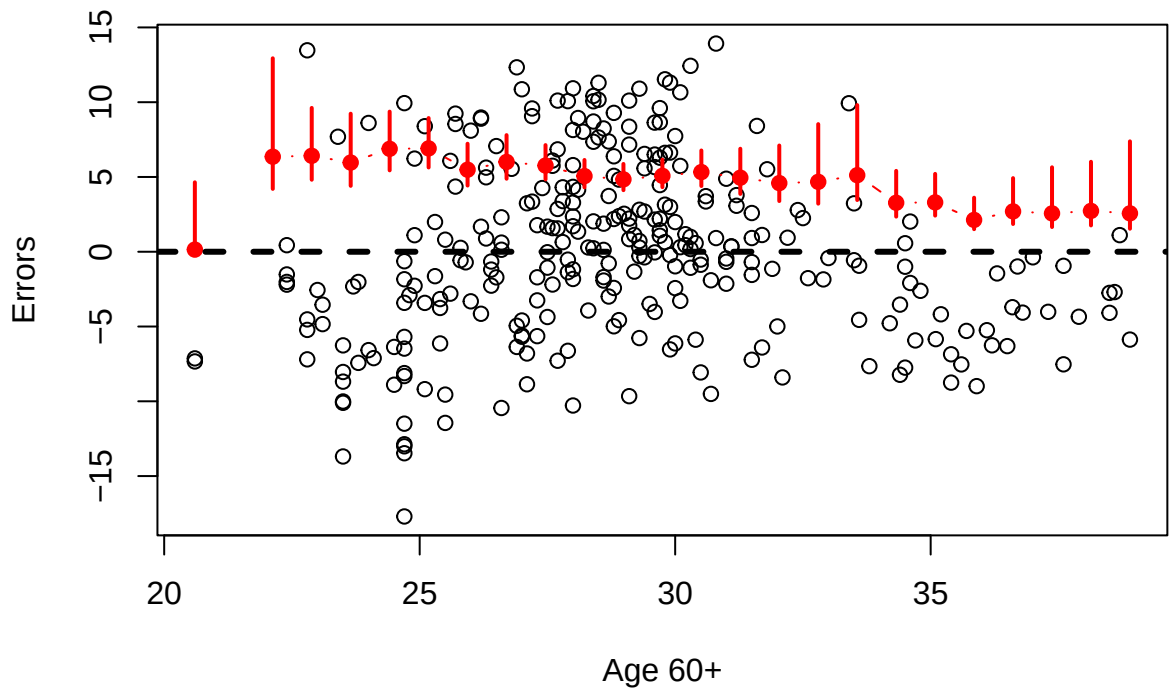
```
y.range <- range(errors)
x.range <- range(old.gen)

plot(errors ~ old.gen,ylab="Errors",xlab="Age 60+",
     xlim=x.range,ylim=y.range)

x.values <- seq(min(old.gen),max(old.gen),length=25)
x.interval <- x.values[2] -x.values[1]

conditional.sd <- lower.b <- upper.b <- rep(NA,length(x.values))
for (i in 1:length(conditional.sd)){

  selected.error <- errors[(old.gen>(x.values[i]-x.interval)) &
```

```
                                                (old.gen<(x.values[i]+x.interval)) ]
  conditional.sd[i] <- sd(selected.error)

  dof <- length(selected.error)-1

  lower.b[i] <- sqrt(conditional.sd[i]^2*dof/qchisq(0.975,df=dof))
  upper.b[i] <- sqrt(conditional.sd[i]^2*dof/qchisq(0.025,df=dof))

}
```

```
## Warning in qchisq(0.975, df = dof): NaNs wurden erzeugt
```

```
## Warning in qchisq(0.025, df = dof): NaNs wurden erzeugt
```

```
par(new=T)
plot(conditional.sd ~ x.values,ann=F,axes=F,
     xlim=x.range,ylim=y.range,
     col="red",pch=19,type="b")
abline(h=0,lty=2,lwd=3)
for (i in 1:length(conditional.sd)){
  lines(rep(x.values[i],2),c(upper.b[i],lower.b[i]),col="red",lwd=2)
}
```



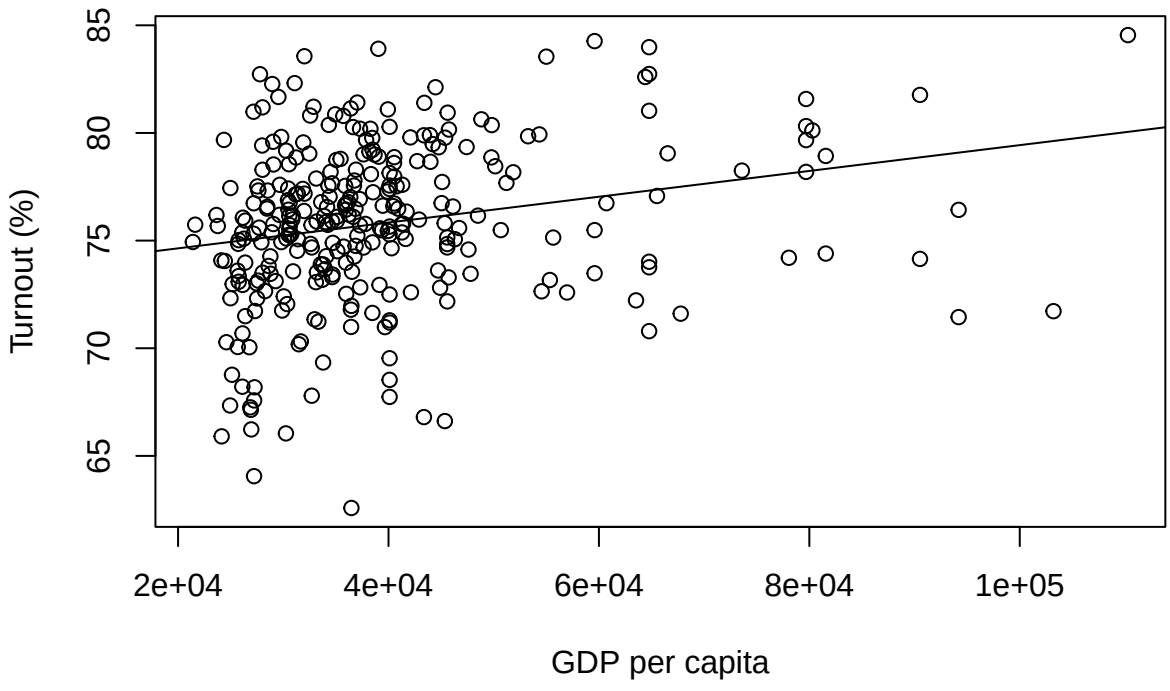
Variance decomposition and Goodness of Fit

To discuss the variance decomposition and goodness of fit, we come back to the first regression model:

```r
summary(lm.out.1)
```

```
##
## Call:
## lm(formula = turnout ~ gdp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0464  -2.0181   0.3747   2.3607   8.2044
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.344e+01  6.170e-01 119.021  < 2e-16 ***
## gdp         5.993e-05  1.470e-05   4.075  5.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.676 on 297 degrees of freedom
## Multiple R-squared:  0.05296,    Adjusted R-squared:  0.04977
## F-statistic: 16.61 on 1 and 297 DF,  p-value: 5.9e-05
```

```r
plot(gdp,turnout,xlab="GDP per capita",ylab="Turnout (%)")
abline(lm.out.1)
```



Based on the regression result above, we can calculate the following three sum of squares

```
SST <- mean((turnout - mean(turnout))^2)
SST
```

```
## [1] 14.17401
```

```
SSE <- mean((lm.out.1$fitted.values - mean(turnout))^2)
SSE
```

```
## [1] 0.7506838
```

```
SSR <- mean((lm.out.1$residuals )^2)
SSR
```

```
## [1] 13.42332
```

```
r.squared <- SSE/SST
r.squared
```

```
## [1] 0.052962
```

Obviously, the total sum of squares of y can be decomposed to SSE and SSR. And the share of SSE in SST is the r-squared, which is a goodness of fit measure of the model.

Estimating variances of errors and OLS estimates

```
SSR <- sum(lm.out.1$residuals^2) # residual square sum
sigma2.hat <- SSR/lm.out.1$df.residual # estimated variance of errors

lm.out.1$df.residual
```

```
## [1] 297
```

```
sqrt(sigma2.hat)
```

```
## [1] 3.676101
```

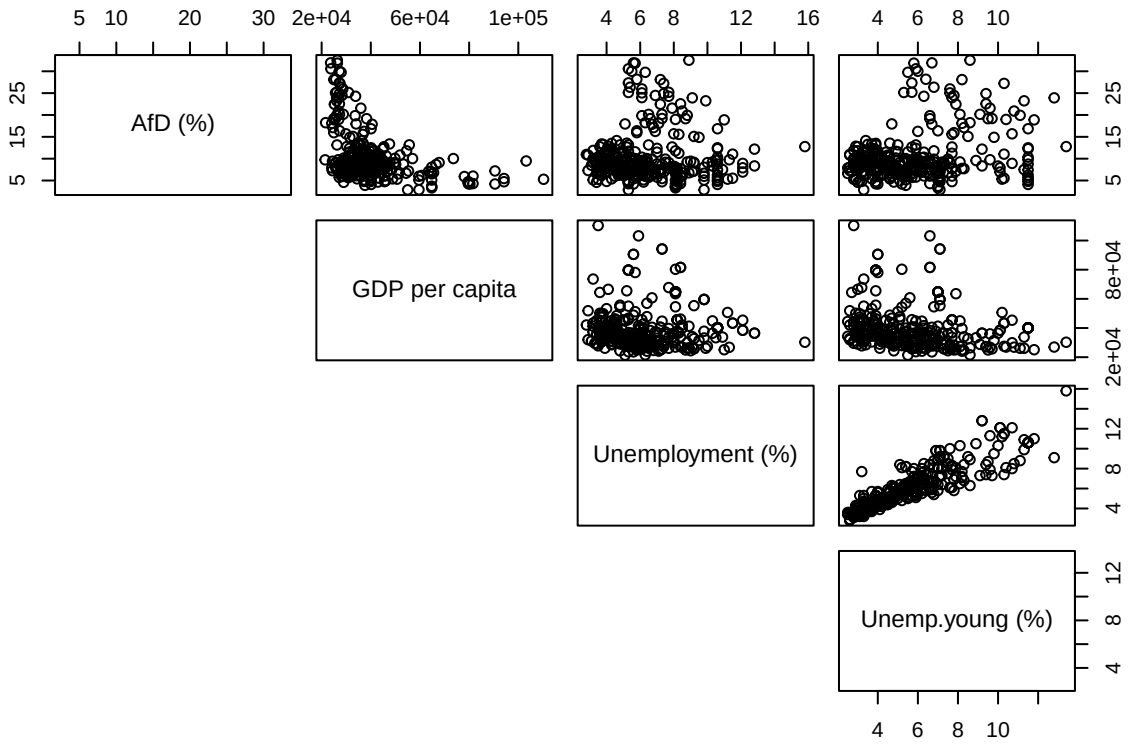
```
SST.x <- sum((gdp-mean(gdp))^2) # total square sum of X (GDP)

sigma2.beta1 <- sigma2.hat/SST.x
sqrt(sigma2.beta1) # standard error of beta_1
```

```
## [1] 1.470399e-05
```

Check this result with the output above.

Multiple regression analysis

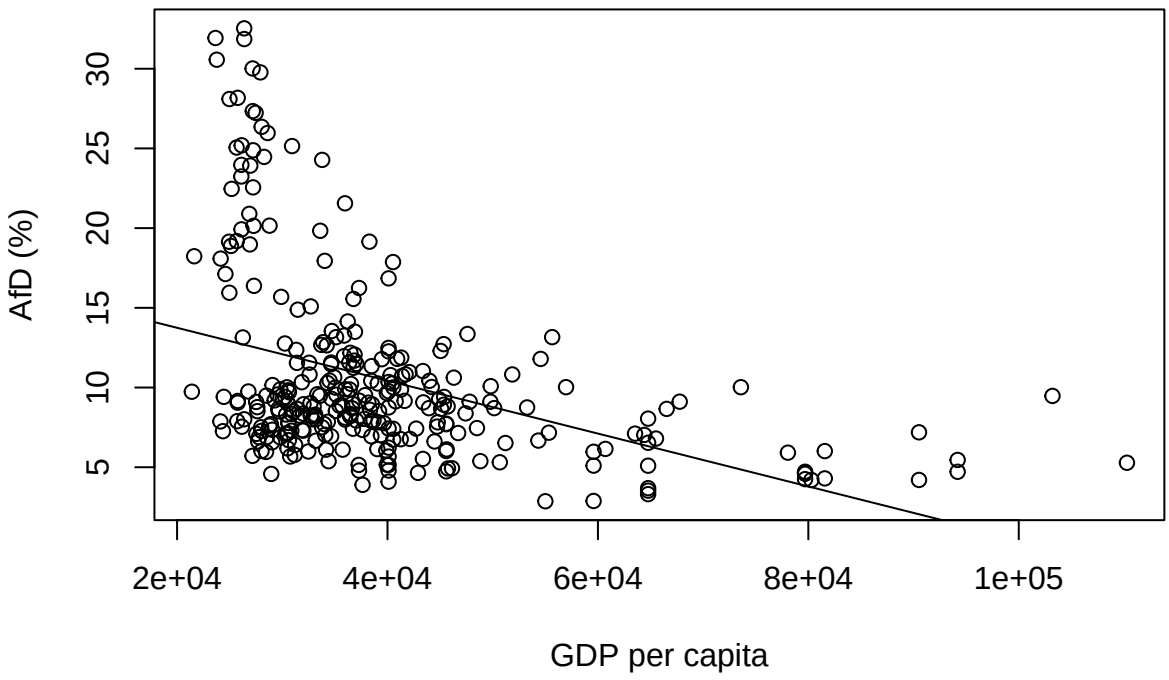
```
pairs(cbind(afd.pr,gdp,unemp.total,unemp.young),
      labels=c("AfD (%)","GDP per capita","Unemployment (%)","Unemp.young (%)"),
      lower.panel = NULL)
```




```
srm.out <- lm(afd.pr ~ gdp)
summary(srm.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.681 -3.554 -1.446  1.463 19.845
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.707e+01  9.026e-01  18.906  < 2e-16 ***
## gdp         -1.659e-04  2.151e-05  -7.713 1.87e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.377 on 297 degrees of freedom
## Multiple R-squared:  0.1669, Adjusted R-squared:  0.1641
## F-statistic: 59.49 on 1 and 297 DF,  p-value: 1.875e-13
```

```
plot(gdp,afd.pr,xlab="GDP per capita",ylab="AfD (%)")
abline(srm.out)
```



The slope estimate has a very small effect size (-1.66×10^{-4}), which is due to the large number of the independent variable (GDP per capita). That is, one Euro increase in GDP leads to only a small change in the AfD vote share.

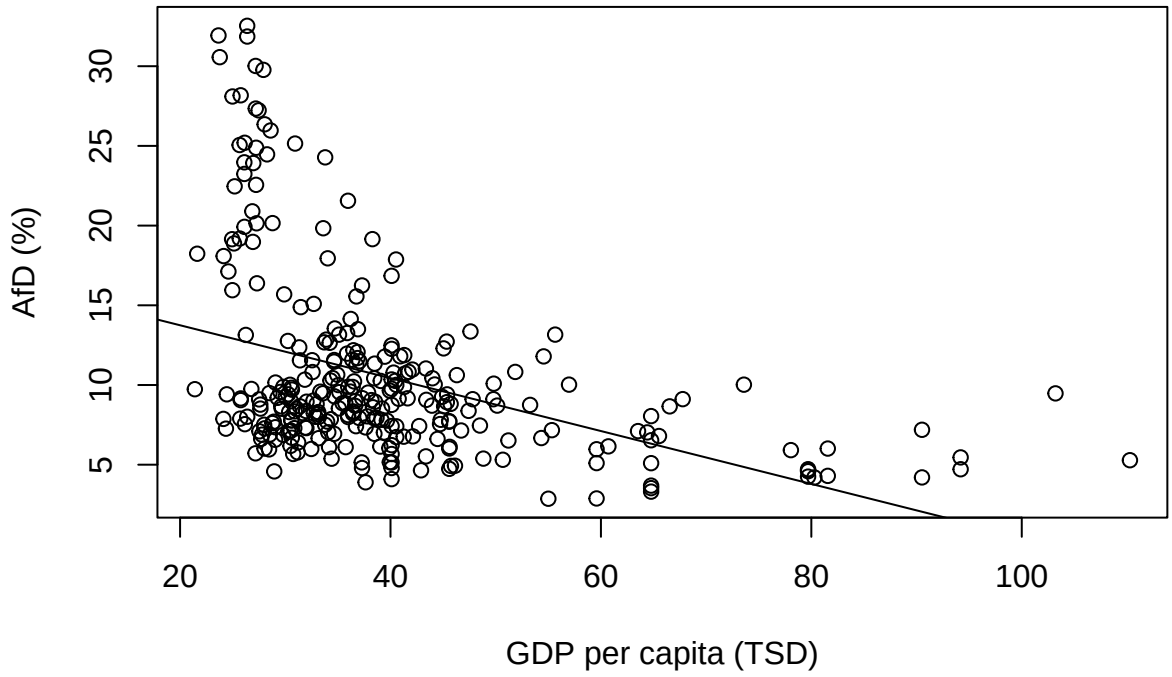
We can now change the scale of the independent variable to thousand Euro and estimate the same model.

```
gdp.tsd <- gdp/1000

srm.out <- lm(afd.pr ~ gdp.tsd)
summary(srm.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.tsd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.681 -3.554 -1.446  1.463 19.845
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.06501    0.90262  18.906  < 2e-16 ***
## gdp.tsd     -0.16589    0.02151  -7.713 1.87e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.377 on 297 degrees of freedom
```

```
## Multiple R-squared:  0.1669, Adjusted R-squared:  0.1641
## F-statistic: 59.49 on 1 and 297 DF,  p-value: 1.875e-13
```

```
plot(gdp.tsd,afd.pr,xlab="GDP per capita (TSD)",ylab="AfD (%)")
abline(srm.out)
```



Now the slope estimate is much larger since it is about the change of AfD vote share for 1000 Euro increase in GDP.

If we add another variable (unemployment of younger citizens), the point estimate of the GDP becomes closer to zero.

```
mrm.out <- lm(afd.pr ~ gdp.tsd + unemp.young)
summary(mrm.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.tsd + unemp.young)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.215 -3.745 -1.230  2.043 19.414
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.45193    1.24560  10.800  < 2e-16 ***
## gdp.tsd     -0.15154    0.02125  -7.131 7.66e-12 ***
## unemp.young  0.51612    0.12599   4.096 5.42e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.24 on 296 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2062
## F-statistic: 39.71 on 2 and 296 DF,  p-value: 5.262e-16
```

To see why this differences in the estimates between two models, we can regress gdp on unemployment.

```
gdp.unemp <- lm(gdp.tsd ~ unemp.young)
summary(gdp.unemp)
```

```
##
## Call:
## lm(formula = gdp.tsd ~ unemp.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.388  -9.031  -4.164   4.407  67.852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  45.1683     2.1677  20.837  < 2e-16 ***
## unemp.young  -0.9777     0.3393  -2.881  0.00425 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.31 on 297 degrees of freedom
## Multiple R-squared:  0.0272, Adjusted R-squared:  0.02392
## F-statistic: 8.303 on 1 and 297 DF,  p-value: 0.004247
```

GDP is negatively correlated with the youth unemployment rate. We can now regress the share of AfD on the residual (part of GDP which cannot be predicted by the youth unemployment rate.)

```
residuals <- gdp.unemp$residuals
afd.resid <- lm(afd.pr ~ residuals)
summary(afd.resid)
```

```
##
## Call:
## lm(formula = afd.pr ~ residuals)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.961 -3.462 -1.461  1.415 20.431
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.52959    0.31679  33.238  < 2e-16 ***
## residuals   -0.15154    0.02222  -6.821 5.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.478 on 297 degrees of freedom
## Multiple R-squared:  0.1355, Adjusted R-squared:  0.1325
## F-statistic: 46.53 on 1 and 297 DF,  p-value: 5.054e-11
```

Compare with the result of the multiple regression model. The point estimate is identical with that of GDP.

The omitted variable bias

```
simple <- lm(afd.pr ~ gdp.tsd)
summary(simple)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.tsd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.681 -3.554 -1.446  1.463 19.845
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.06501    0.90262  18.906  < 2e-16 ***
## gdp.tsd     -0.16589    0.02151  -7.713 1.87e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.377 on 297 degrees of freedom
## Multiple R-squared:  0.1669, Adjusted R-squared:  0.1641
## F-statistic: 59.49 on 1 and 297 DF,  p-value: 1.875e-13
```

```
multiple <- lm(afd.pr ~ gdp.tsd + unemp.young)
summary(multiple)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.tsd + unemp.young)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.215 -3.745 -1.230  2.043 19.414
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.45193    1.24560  10.800  < 2e-16 ***
## gdp.tsd     -0.15154    0.02125  -7.131 7.66e-12 ***
## unemp.young  0.51612    0.12599   4.096 5.42e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.24 on 296 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2062
## F-statistic: 39.71 on 2 and 296 DF,  p-value: 5.262e-16
```

If the model with two independent variables is true, the first one is misspecified. The bias can be calculated as follows:

```
# extracting the relevant estimates
beta.1.s <- simple$coefficients[2]
beta.1.m <- multiple$coefficients[2]
beta.2.m <- multiple$coefficients[3]
```

```r
# regress the omitted variable on the other variable
unemp.gdp <- lm(unemp.young ~ gdp)

# extracting the coefficient
delta.1 <- unemp.gdp$coefficients[2]

# biased estimate in SRM
beta.1.s
```

```
##    gdp.tsd
## -0.1658944
```

```r
# true estimate in MRM
beta.1.m
```

```
##    gdp.tsd
## -0.1515387
```

```r
# reconstructing the biased estimate by using the MRM estimates
beta.1.m + beta.2.m*delta.1
```

```
##   gdp.tsd
## -0.151553
```

```r
# bias
beta.2.m*delta.1
```

```
##   unemp.young
## -1.435573e-05
```

The bias constitutes the effect of the omitted variable and its relationship with the other variable.

Homoskedasticity

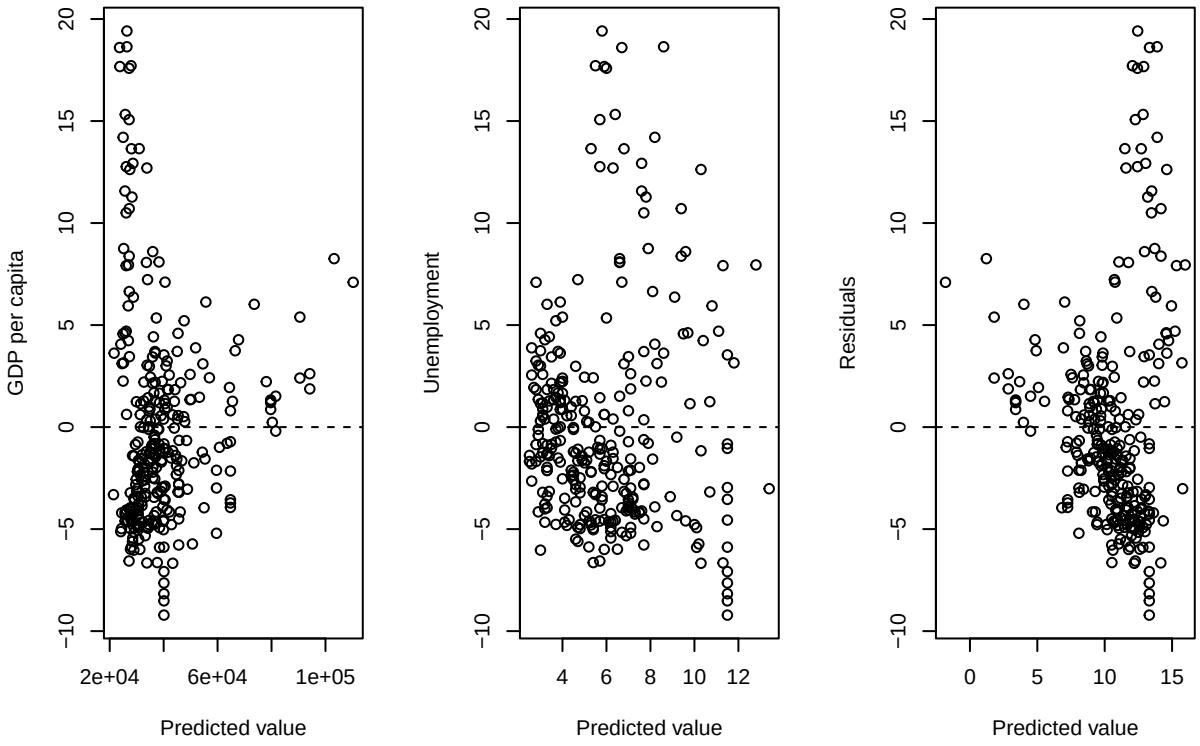
The one of the Gauss-Markov-assumptions is about the variance of errors. While we cannot directly observe the errors, we can obtain certain information about the errors by observing residuals and comparing them with independent variables/fitted values.

```r
# extracting the relevant estimates
lm.out <- lm(afd.pr ~ gdp + unemp.young)
summary(lm.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp + unemp.young)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.215 -3.745 -1.230  2.043 19.414
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.345e+01  1.246e+00  10.800  < 2e-16 ***
## gdp         -1.515e-04  2.125e-05  -7.131 7.66e-12 ***
## unemp.young  5.161e-01  1.260e-01   4.096 5.42e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5.24 on 296 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2062
## F-statistic: 39.71 on 2 and 296 DF,  p-value: 5.262e-16
```

```
par(mfrow=c(1,3))
plot(gdp,lm.out$residuals,
     xlab="Predicted value",
     ylab="GDP per capita")
abline(h=0,lty=2)
plot(unemp.young,lm.out$residuals,
     xlab="Predicted value",
     ylab="Unemployment")
abline(h=0,lty=2)
plot(lm.out$fitted.values,lm.out$residuals,
     xlab="Predicted value",
     ylab="Residuals")
abline(h=0,lty=2)
```



The distributions indicate violence of the homoskedasticity assumption, thus the biased estimates of the regression coefficients' variances.

Model selection

You regress the vote share of FDP on the following variables:

- Number of companies per 1000 inhabitants
- Number of registered cars per 1000 inhabitants
- Number of registered electronic/hybrid cars per 1000 inhabitants

- Share of the older generation
- Share of the younger generation

```
summary(lm.out <- lm(fdp.pr ~ company + pkw + pkw.e  + young.gen +  old.gen))
```

```
##
## Call:
## lm(formula = fdp.pr ~ company + pkw + pkw.e + young.gen + old.gen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6371 -1.0257 -0.1073  1.0282  5.5884
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.605231   3.289842   0.792 0.429059
## company      0.050926   0.020569   2.476 0.013858 *
## pkw          0.014153   0.001477   9.582  < 2e-16 ***
## pkw.e        1.221298   0.314111   3.888 0.000125 ***
## young.gen    0.044609   0.136001   0.328 0.743142
## old.gen     -0.123897   0.064380  -1.924 0.055266 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.876 on 293 degrees of freedom
## Multiple R-squared:  0.3362, Adjusted R-squared:  0.3248
## F-statistic: 29.67 on 5 and 293 DF,  p-value: < 2.2e-16
```

Since the last two variables concerning the generation are not significant, you estimated another model without these variables:

```
summary(lm.out.res <- lm(fdp.pr ~ company + pkw + pkw.e  ))
```

```
##
## Call:
## lm(formula = fdp.pr ~ company + pkw + pkw.e)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1759 -1.0699  0.0001  0.9187  5.9206
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.316346   1.180173  -1.115  0.26559
## company      0.058199   0.020190   2.883  0.00423 **
## pkw          0.013698   0.001497   9.154  < 2e-16 ***
## pkw.e        1.671881   0.277963   6.015 5.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.91 on 295 degrees of freedom
## Multiple R-squared:  0.3074, Adjusted R-squared:  0.3003
## F-statistic: 43.64 on 3 and 295 DF,  p-value: < 2.2e-16
```

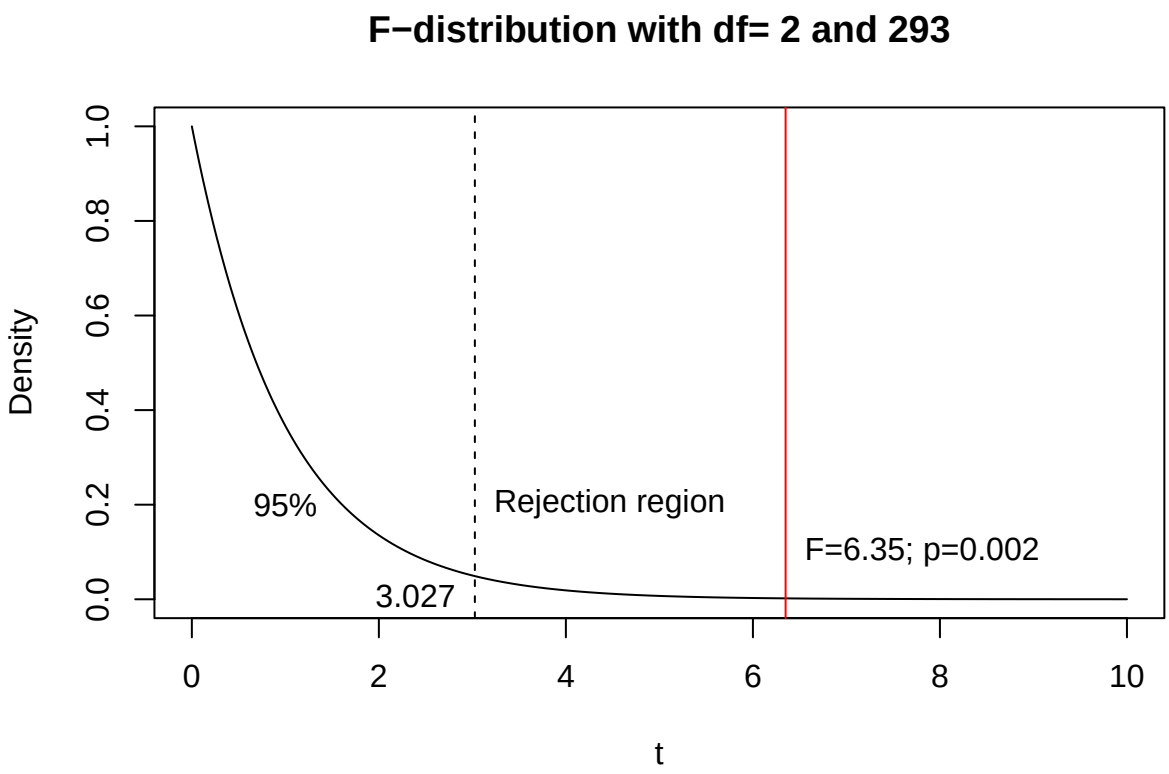
Now you wish to decide for one of the two models. In such a situation, the typical approach is to compute the F-values based on both outcomes:

```
    SSR.ur <- sum(lm.out$residuals^2)
    SSR.r <-  sum(lm.out.res$residuals^2)
    q <- lm.out.res$df.residual - lm.out$df.residual

    F <-  ((SSR.r - SSR.ur)/q)/(SSR.ur/lm.out$df.residual)

    F
```

```
## [1] 6.349564
```

The computed F-value is then compared with the corresponding F-distribution:

```
x.scale <- seq(0,10,by=0.01)
this.y <- df(x.scale,df1=q,df2=lm.out$df.residual)

plot(x.scale,this.y,type="l",xlab="t",ylab="Density",
       main=paste("F-distribution with df=",q,"and",lm.out$df.residual))

this.95 <- qf(c(.95),df1=q,df2=lm.out$df.residual)
abline(v=this.95,lty=2)
text(1,0.2,"95%")
text(this.95,0,pos=2,round(this.95,3))
text(this.95,0.2,"Rejection region",pos=4)

emp.sig <- 1-pf(F,df1=q,df2=lm.out$df.residual)

abline(v=F,col="red")
text(F,0.1,
     paste0("F=",round(F,3),"; p=",round(emp.sig,5)),
     pos=4)
```



Accordingly, we can reject the null-hypothesis that both models are equivalent and decide for the first unrestricted model.

An alternative approach is based on the Lagrange Multiplier statistic:

```
r.tilde <- residuals(lm.out.res)

summary(lm.out.lagr <- lm(r.tilde ~ company + pkw + pkw.e  + young.gen +  old.gen))
```

```
##
## Call:
## lm(formula = r.tilde ~ company + pkw + pkw.e + young.gen + old.gen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6371 -1.0257 -0.1073  1.0282  5.5884
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.9215770  3.2898422   1.192   0.2342
## company     -0.0072725  0.0205694  -0.354   0.7239
## pkw          0.0004551  0.0014770   0.308   0.7582
## pkw.e       -0.4505825  0.3141111  -1.434   0.1525
## young.gen    0.0446090  0.1360009   0.328   0.7431
## old.gen     -0.1238971  0.0643803  -1.924   0.0553 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

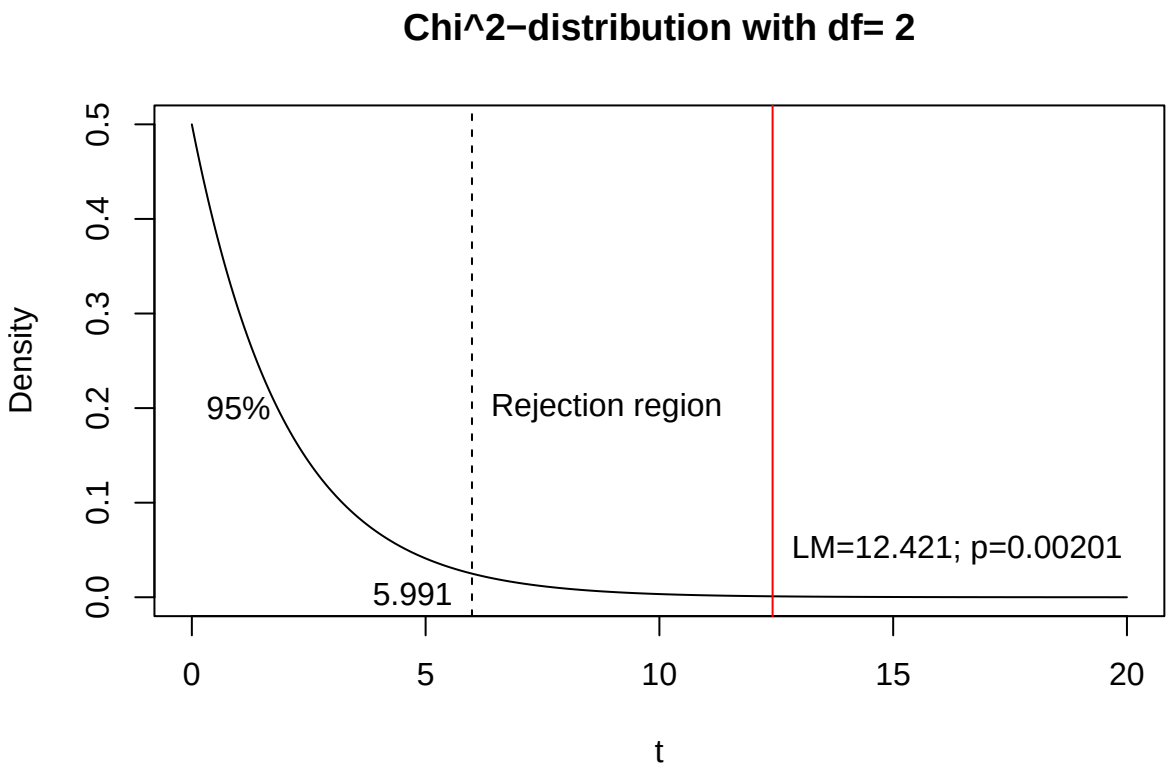
```
##
## Residual standard error: 1.876 on 293 degrees of freedom
## Multiple R-squared:  0.04154,    Adjusted R-squared:  0.02519
## F-statistic:  2.54 on 5 and 293 DF,  p-value: 0.0286
```

```
r.square.u <- summary(lm.out.lagr)$r.squared

LM <- r.square.u * nrow(lm.out.lagr$model)

LM
```

```
## [1] 12.42084
```

The calculated LM-value is compared with the corresponding chi-square distribution:

```
x.scale <- seq(0,20,by=0.01)
this.y <- dchisq(x.scale,df=q)

plot(x.scale,this.y,type="l",xlab="t",ylab="Density",
       main=paste("Chi^2-distribution with df=",q))

this.95 <- qchisq(c(.95),df=q)
abline(v=this.95,lty=2)
text(1,0.2,"95%")
text(this.95,0,pos=2,round(this.95,3))
text(this.95,0.2,"Rejection region",pos=4)

emp.sig <- 1-pchisq(LM,df=q)

abline(v=LM,col="red")
text(LM,0.05,
     paste0("LM=",round(LM,3),"; p=",round(emp.sig,5)),
     pos=4)
```



Accordingly, you can obtain the same result of the above F-test.

Using logarithm

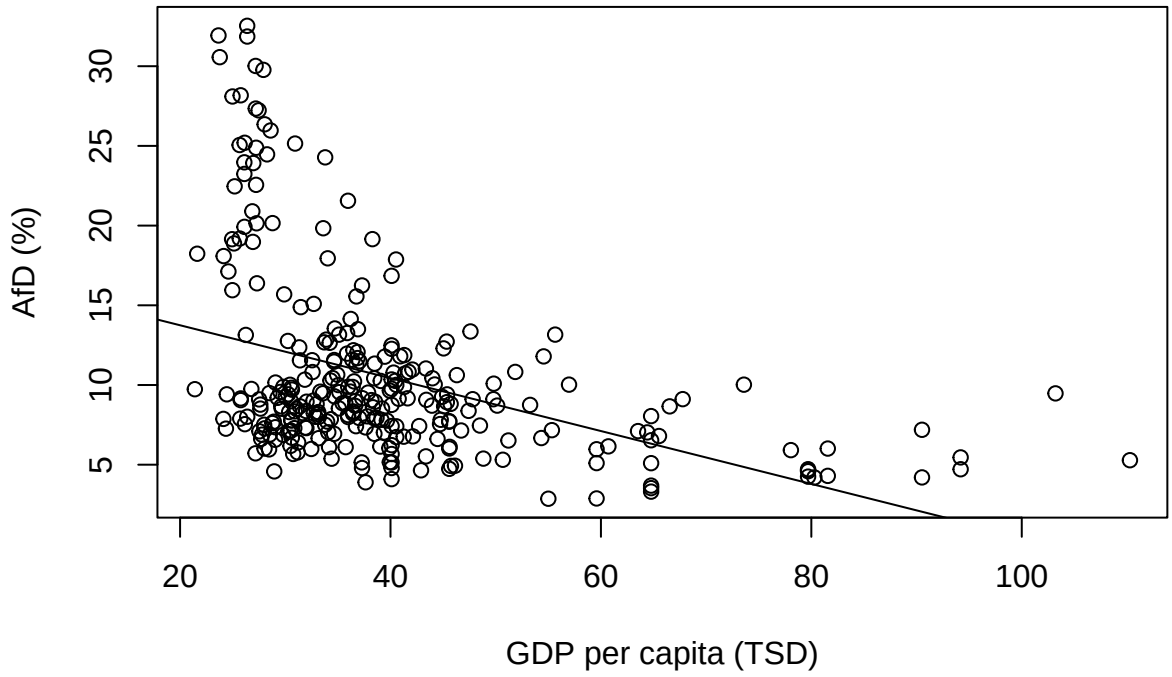
Recall the following regression model:

```
gdp.tsd <- gdp/1000

srm.out <- lm(afd.pr ~ gdp.tsd)
summary(srm.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.tsd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.681 -3.554 -1.446  1.463 19.845
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.06501    0.90262  18.906  < 2e-16 ***
## gdp.tsd     -0.16589    0.02151  -7.713 1.87e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.377 on 297 degrees of freedom
```

```
## Multiple R-squared:  0.1669, Adjusted R-squared:  0.1641
## F-statistic: 59.49 on 1 and 297 DF,  p-value: 1.875e-13
```

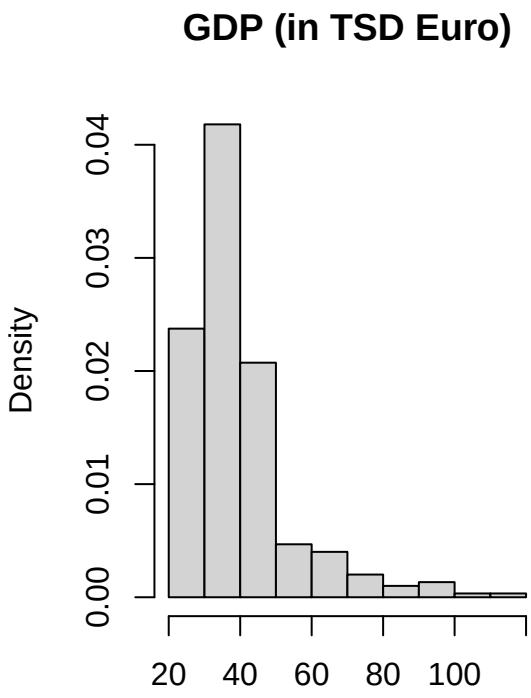
```
plot(gdp.tsd,afd.pr,xlab="GDP per capita (TSD)",ylab="AfD (%)")
abline(srm.out)
```

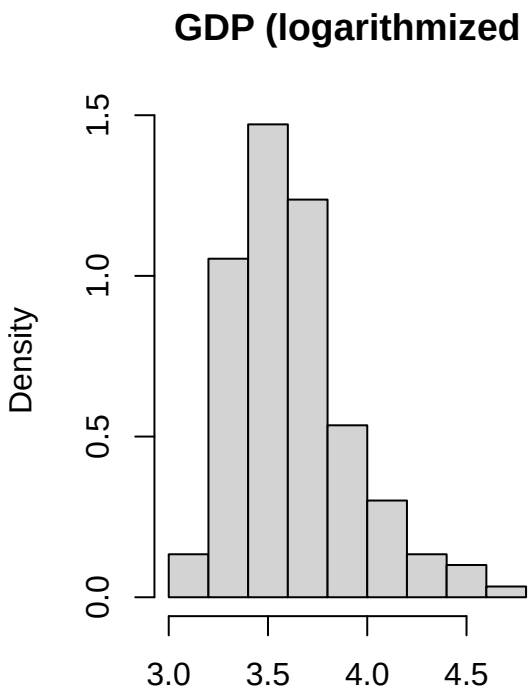


The joint distribution of both variables shows certain outlier districts whose GDP is very high.

In such cases, one can consider to transform the variable by using the logartihmic function.

```
gdp.tsd.log <- log(gdp.tsd)

par(mfrow=c(1,2))

hist(gdp.tsd,
     main="GDP (in TSD Euro)",xlab="",
     freq=FALSE)

hist(gdp.tsd.log,
     main="GDP (logarithmized",xlab="",
     freq=FALSE)
```

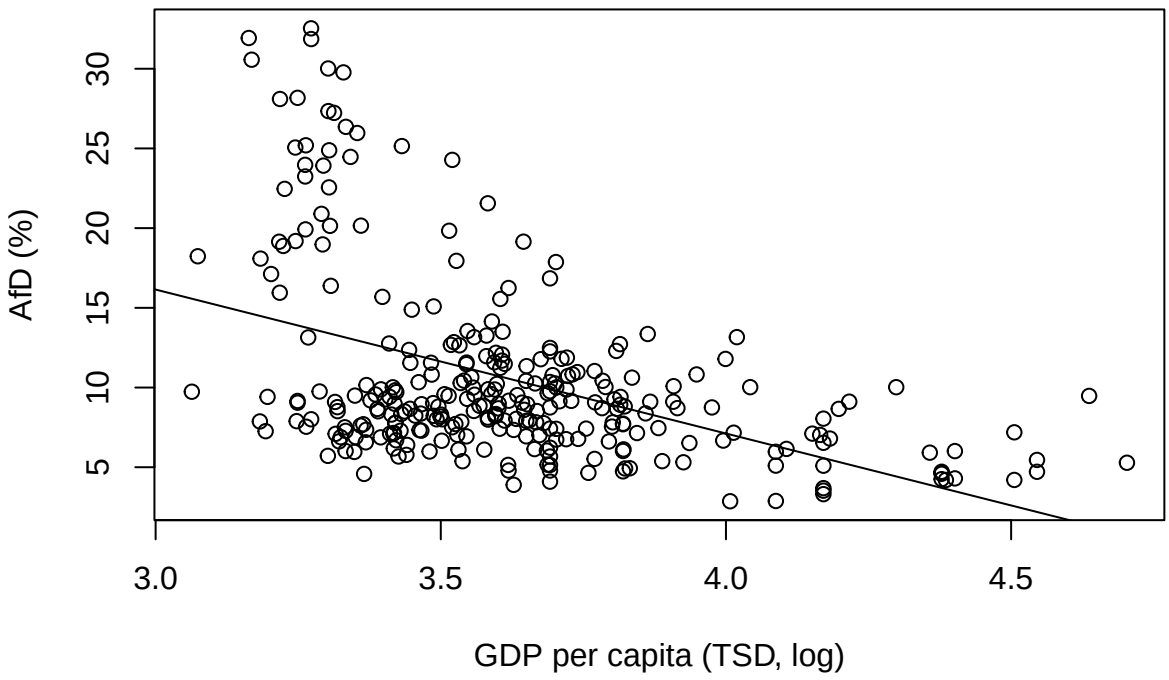





```
srm.out.2 <- lm(afd.pr ~ gdp.tsd.log)
summary(srm.out.2)
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.tsd.log)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.264 -3.593 -1.034  1.926 18.848
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43.2336     3.6028  12.000   <2e-16 ***
## gdp.tsd.log  -9.0291     0.9912  -9.109   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.209 on 297 degrees of freedom
## Multiple R-squared:  0.2184, Adjusted R-squared:  0.2157
## F-statistic: 82.98 on 1 and 297 DF,  p-value: < 2.2e-16
```

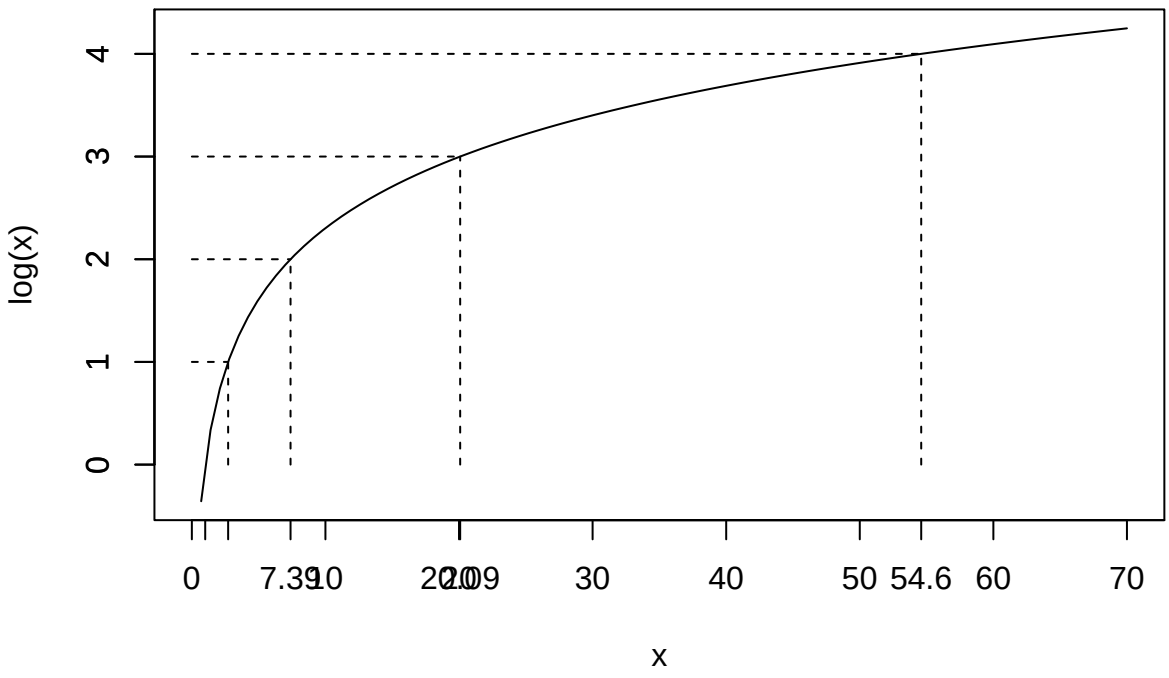
```
plot(gdp.tsd.log,afd.pr,xlab="GDP per capita (TSD, log)",ylab="AfD (%)")
abline(srm.out.2)
```



To interpret the slope coefficient of a logarithmized variable is tricky. According to the regression result above, unit increase of log GDP is associated with increase of -9.029074. However, a unit increase of the log GDP is not constant on the scale of the raw GDP.

To see what a unit increase of the log GDP means, we can first check the logarithm function.

```
curve(log,0,70,axes=F)
axis(1,at=exp(seq(0,5)),
     c("",round(exp(seq(1,5)),2))
     )
axis(2,at=seq(0,5))

for (i in 1:4){
  lines(c(0,exp(i)),rep(i,2),lty=2)
  lines(rep(exp(i),2),c(0,i),lty=2)
}
```



Note that the opposite function of log() is the exponential function (exp()). To solve e.g. log(x)=2 for x, we can calculate just exp(2)=7.3890561.

The above figure clearly demonstrates that increase from log(2) to log(3) and that from log(3) to log(4) correspond different increase in the raw scale. If you however take the growth, it is constant. That is:

- exp(3)/exp(2) = 20.09/7.39 = 2.7182818
- exp(4)/exp(3) = 54.6/20.09 = 2.7182818

That is, we can interpret the coefficient of the log GDP (-9.029074) as change in Y given X grows constantly. More specifically, given a district's GDP is 1% higher, the the AfD share will change with -9.029074/100.

Consequently, the result will not change if we take the log of GDP in Euro instead of in thousand Euro:

```
gdp.log <- log(gdp)

head(cbind(gdp,gdp.log,gdp.tsd,gdp.tsd.log),n=10)
```

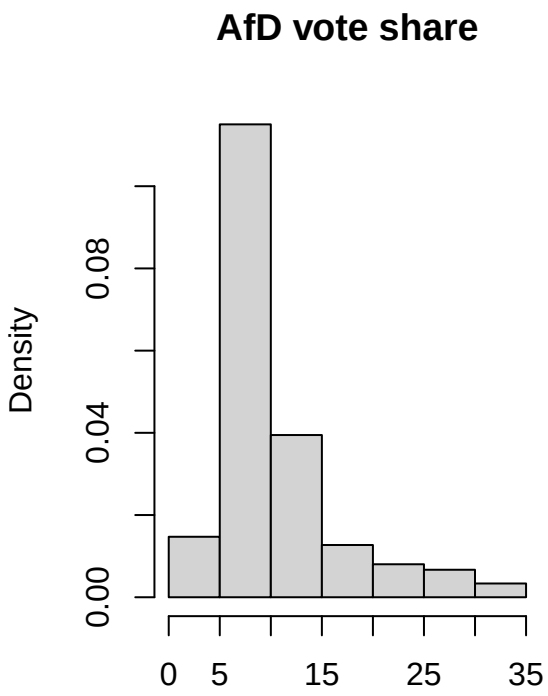
```
##         gdp  gdp.log gdp.tsd gdp.tsd.log
##  [1,] 31178 10.34747  31.178    3.439713
##  [2,] 34160 10.43881  34.160    3.531055
##  [3,] 31977 10.37277  31.977    3.465017
##  [4,] 29036 10.27629  29.036    3.368536
##  [5,] 46128 10.73918  46.128    3.831420
##  [6,] 29053 10.27688  29.053    3.369122
##  [7,] 29803 10.30236  29.803    3.394609
##  [8,] 31902 10.37042  31.902    3.462669
##  [9,] 27539 10.22336  27.539    3.315603
## [10,] 27988 10.23953  27.988    3.331776
```

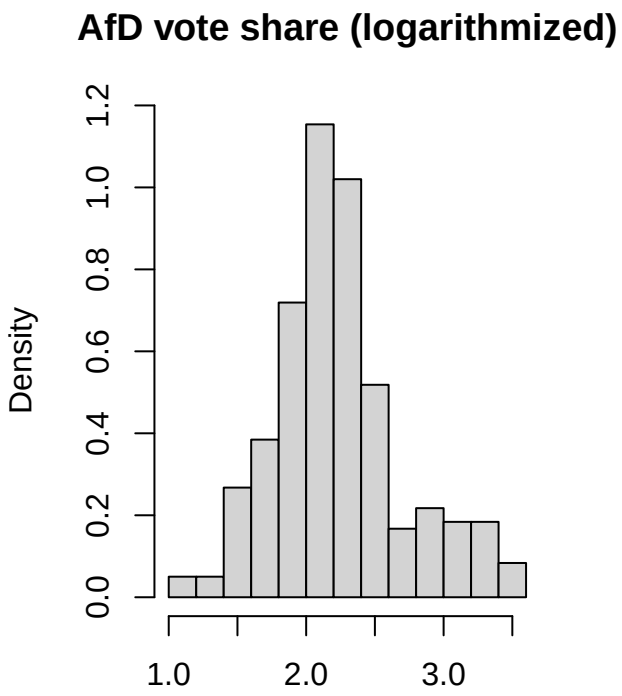
```r
summary(srm.out.3 <-
          lm(afd.pr ~ gdp.log))
```

```
##
## Call:
## lm(formula = afd.pr ~ gdp.log)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.264 -3.593 -1.034  1.926 18.848
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 105.6043    10.4416  10.114   <2e-16 ***
## gdp.log      -9.0291     0.9912  -9.109   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.209 on 297 degrees of freedom
## Multiple R-squared:  0.2184, Adjusted R-squared:  0.2157
## F-statistic: 82.98 on 1 and 297 DF,  p-value: < 2.2e-16
```

Thus far, we logarithmized the independent variable, but the distribution of the dependent variable also has some outliers.

```r
afd.pr.log <- log(afd.pr)

par(mfrow=c(1,2))

hist(afd.pr,
     main="AfD vote share",xlab="",
     freq=FALSE)

hist(afd.pr.log,
     main="AfD vote share (logarithmized)",xlab="",
     freq=FALSE)
```






```
srm.out.4 <- lm(afd.pr.log ~ gdp.tsd.log)
summary(srm.out.4)
```

```
##
## Call:
## lm(formula = afd.pr.log ~ gdp.tsd.log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91563 -0.26935 -0.03682  0.24233  0.97087
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.10573    0.27909   18.29   <2e-16 ***
## gdp.tsd.log -0.79274    0.07678  -10.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4035 on 297 degrees of freedom
## Multiple R-squared:  0.2641, Adjusted R-squared:  0.2616
## F-statistic: 106.6 on 1 and 297 DF,  p-value: < 2.2e-16
```

Having learned about the properties of logarithmized variables, the interpretation of this result is now straighforward. The estimated slope give information about what growth for AfD vote share in a district we expect for one-percent growth of the district's GDP.

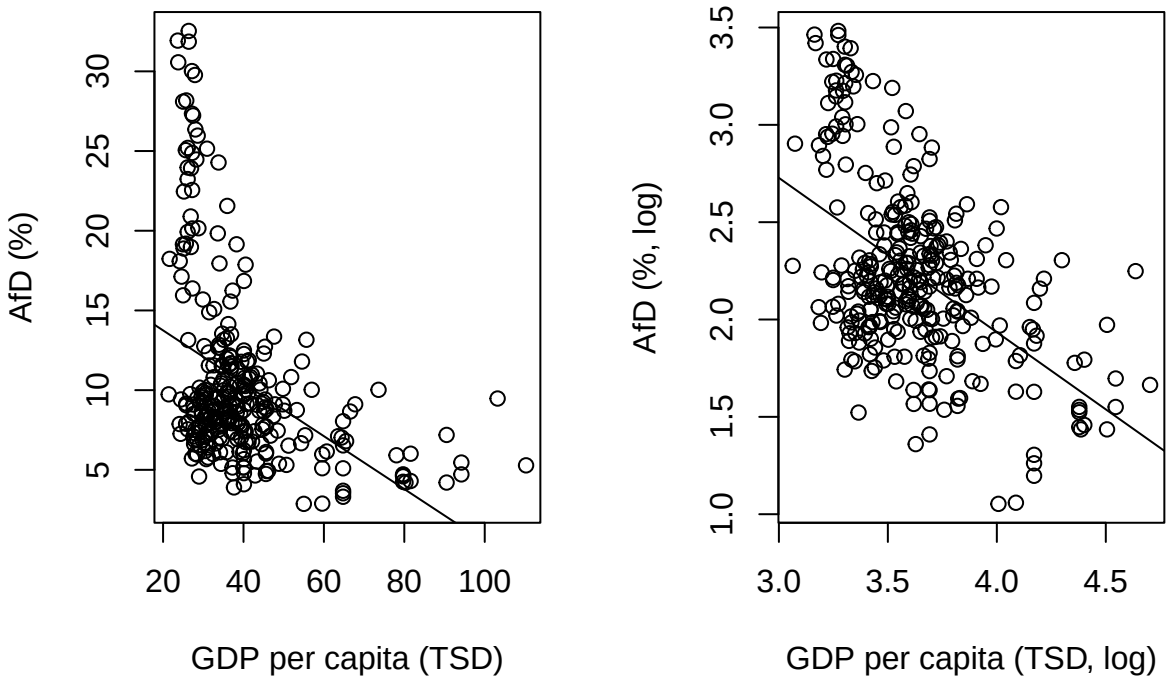
In this model, the percentage change of AfD vote share for the percent change of GDP is constant: $\frac{\%\Delta y}{\%\Delta x} = \beta_1$.

This is called elasticity (see Appendix A) and the model above is called the constant elasticity model.

Elasticity of y in respect to x: $\frac{\%\Delta y}{\%\Delta x}$.

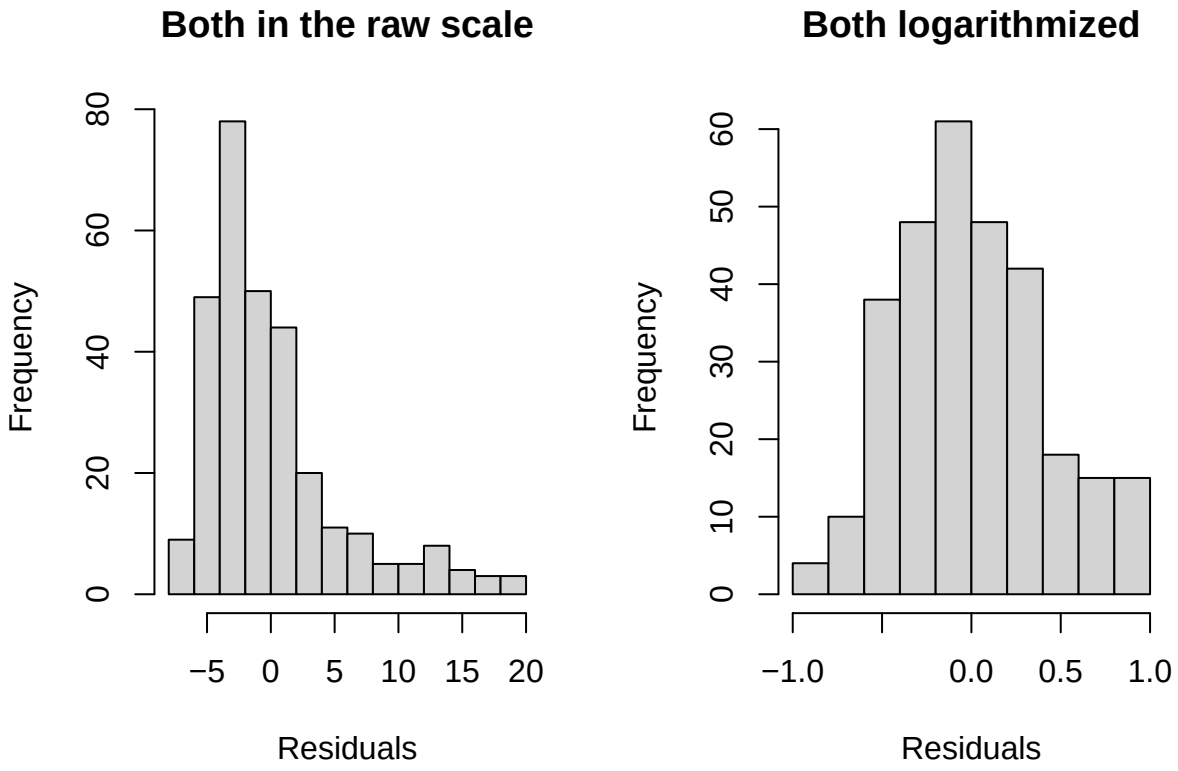
Besides reducing the effect of outliers on the estimated results, using logarithmized variables has another important advantage. To see this, we compare the joint distributions based on the raw variables and logarithmized variables:

```
par(mfrow=c(1,2))

plot(gdp.tsd,afd.pr,xlab="GDP per capita (TSD)",ylab="AfD (%)")
abline(srm.out)

plot(gdp.tsd.log,afd.pr.log,xlab="GDP per capita (TSD, log)",ylab="AfD (%, log)")
abline(srm.out.4)
```



In the left-hand panel, the joint distribution seems to violate a series of the GM-assumptions such like linearity, zero conditional mean and heteroskedasticity. In the right-hand panel with the logarithmized variables, the joint distribution seems to be more conform with the GM-assumptions.

```
par(mfrow=c(1,2))

hist(srm.out$residuals,xlab="Residuals",main="Both in the raw scale",
     freq=F)

hist(srm.out.4$residuals,xlab="Residuals",main="Both logarithmized",
     freq=F)
```



The distributions of the residuals also shows that the model with the logarithmized dependent and independent variables much better resembles the normal distribution, even though it is not perfect.

Above, we have already seen that the logarithm function and the exponential function are the opposite function to each other.

Therefore, $\ln(y) = \beta_0 + \beta_1 x + u$ is equivalent to $y = \exp(\beta_0 + \beta_1 x + u)$. Consider the following model:

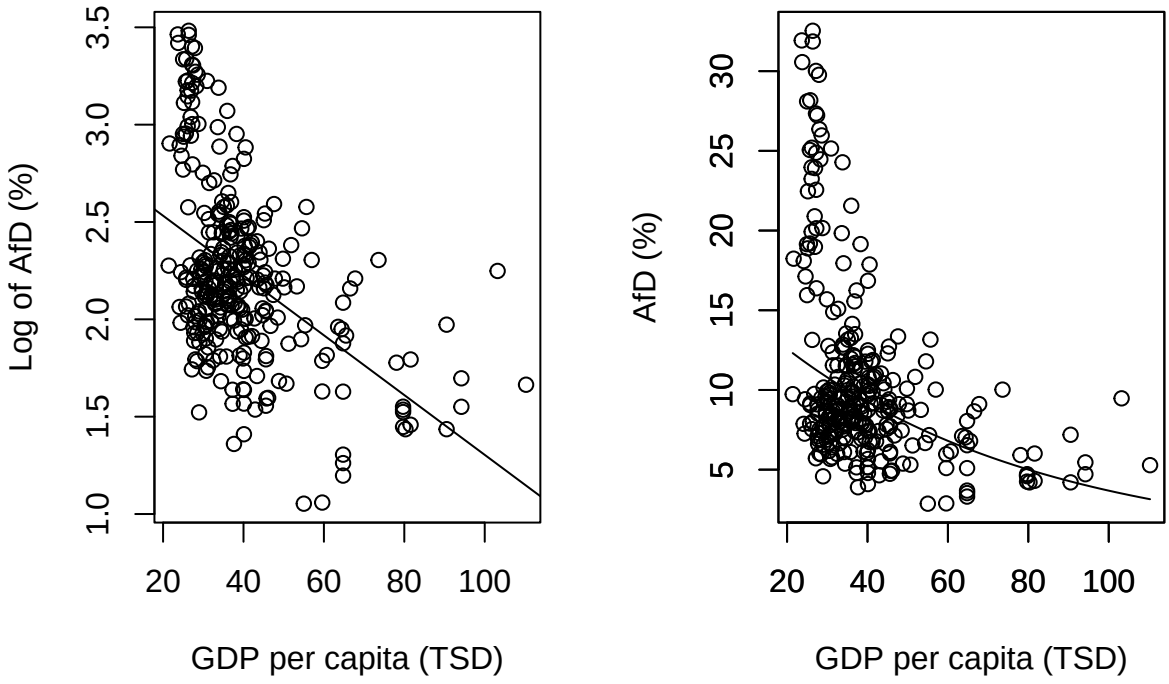
```
srm.out.5 <- lm(afd.pr.log ~ gdp.tsd)
summary(srm.out.5)
```

```
##
## Call:
## lm(formula = afd.pr.log ~ gdp.tsd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94158 -0.26999 -0.07433  0.23021  1.04810
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.839174   0.069534  40.831   <2e-16 ***
## gdp.tsd     -0.015353   0.001657  -9.266   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4143 on 297 degrees of freedom
## Multiple R-squared:  0.2242, Adjusted R-squared:  0.2216
```

```
## F-statistic: 85.85 on 1 and 297 DF,  p-value: < 2.2e-16
```

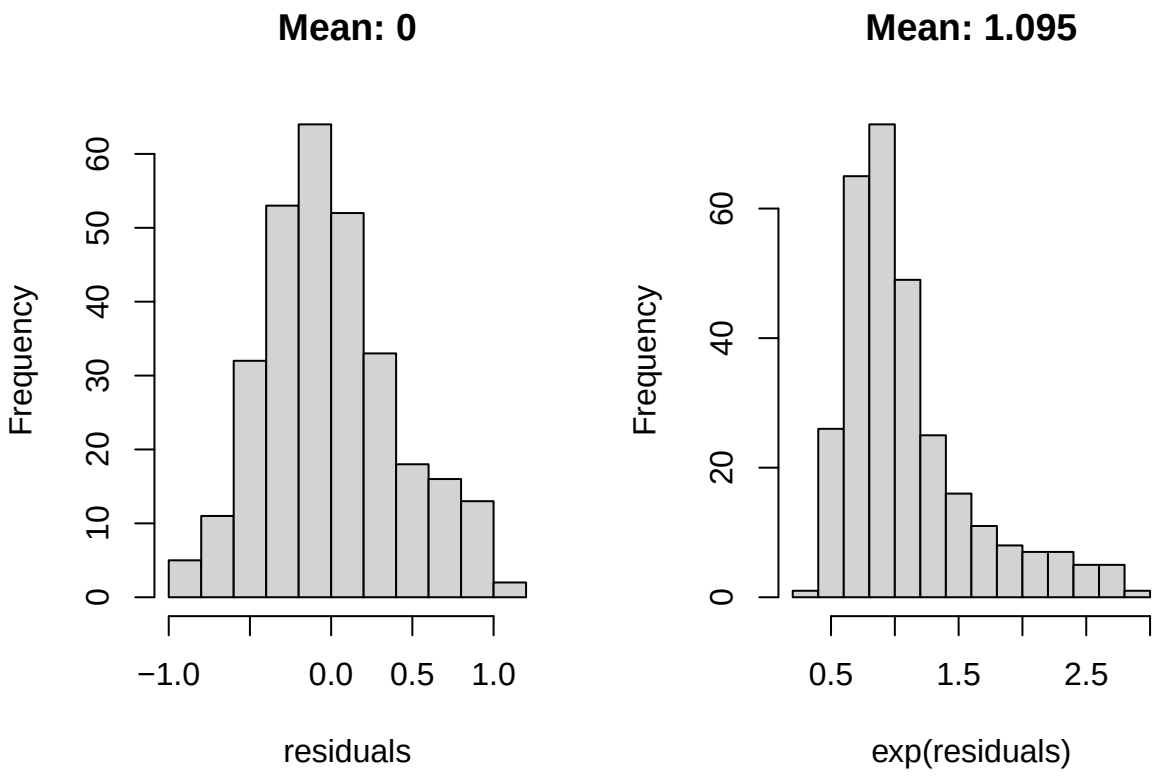
This result can be visualized as follows:

```
x.values <- seq(min(gdp.tsd),max(gdp.tsd),length=100)
predicted <- exp(coefficients(srm.out.5)[1] + coefficients(srm.out.5)[2]*x.values)

par(mfrow=c(1,2))
plot(gdp.tsd,log(afd.pr),
     xlim=range(gdp.tsd),ylim=range(log(afd.pr)),
     xlab="GDP per capita (TSD)",ylab="Log of AfD (%)")
abline(srm.out.5)

plot(gdp.tsd,afd.pr,
     xlim=range(gdp.tsd),ylim=range(afd.pr),
     xlab="GDP per capita (TSD)",ylab="AfD (%)")
par(new=T)
plot(x.values,predicted,type="l",
     xlim=range(gdp.tsd),ylim=range(afd.pr),
     axes=F,ann=F,xlab="",ylab="")
```



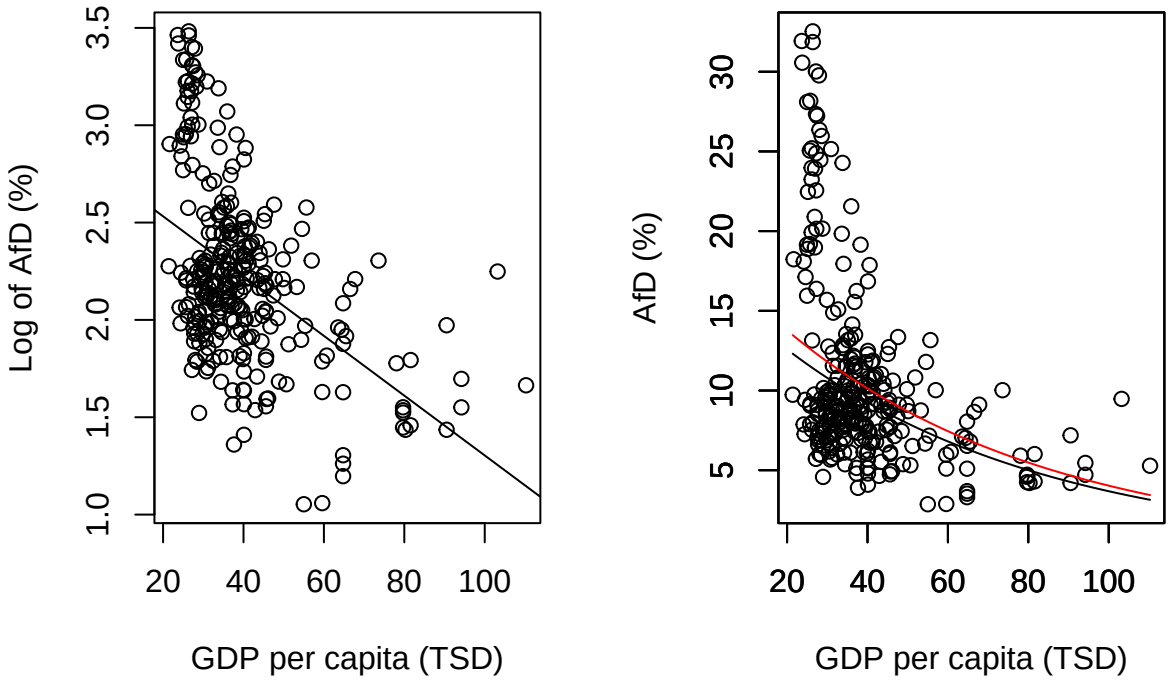
The right hand side panel is based on the raw scales. Correspondingly, the predicted value was transformed by the exponential function. This predicted values, however, are systematically underestimated and biased. This is because the estimated residuals do not have unit mean ($exp(0) = 1$) after exponentiation:

```
par(mfrow=c(1,2))
hist((srm.out.5$residuals),main=paste("Mean:", round(mean(srm.out.5$residuals),3)),xlab="residuals")
hist(exp(srm.out.5$residuals),main=paste("Mean:", round(mean(exp(srm.out.5$residuals)),3)),xlab="exp(res
```



Under the CLM-assumptions, therefore, we have to correct that bias by using the average exponentiated residuals:

```
alpha.0.hat <- mean(exp(srm.out.5$residuals))

correct.predicted <- alpha.0.hat*predicted


par(mfrow=c(1,2))
plot(gdp.tsd,log(afd.pr),
     xlim=range(gdp.tsd),ylim=range(log(afd.pr)),
     xlab="GDP per capita (TSD)",ylab="Log of AfD (%)")
abline(srm.out.5)

plot(gdp.tsd,afd.pr,
     xlim=range(gdp.tsd),ylim=range(afd.pr),
     xlab="GDP per capita (TSD)",ylab="AfD (%)")
par(new=T)
plot(x.values,predicted,type="l",
     xlim=range(gdp.tsd),ylim=range(afd.pr),
     axes=F,ann=F,xlab="",ylab="")
par(new=T)
plot(x.values,correct.predicted,type="l",
     xlim=range(gdp.tsd),ylim=range(afd.pr),col="red",
     axes=F,ann=F,xlab="",ylab="")
```



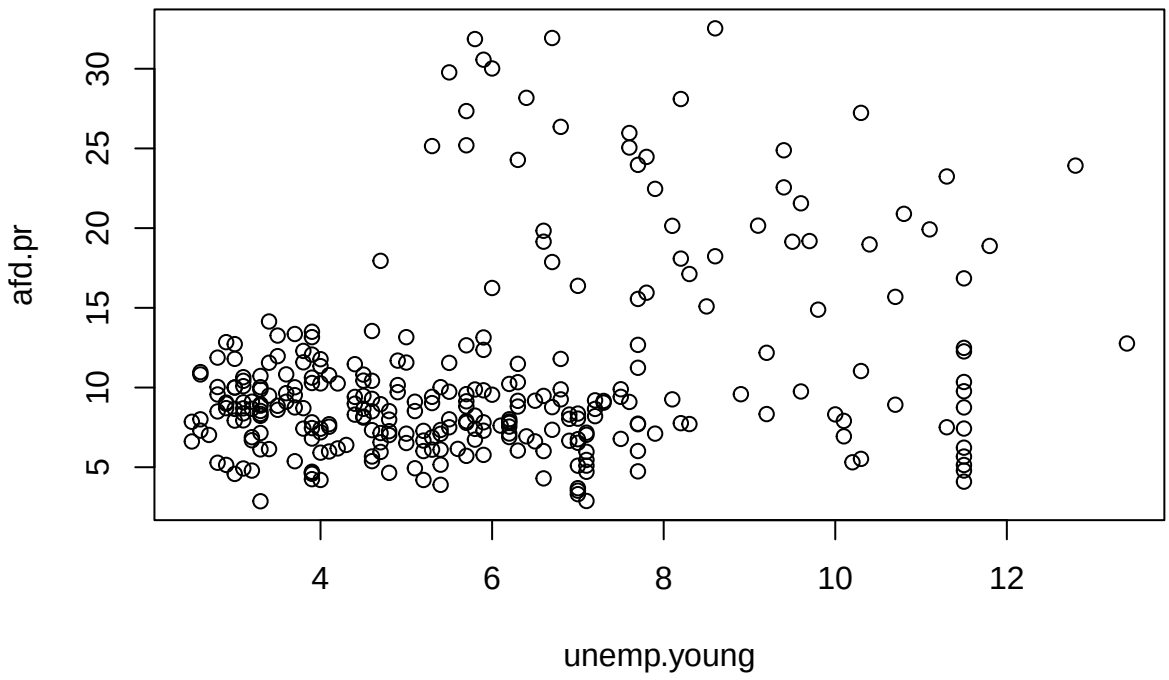
The red line in the right-hand panel is corrected prediction ($\hat{\alpha}_0 \exp(\beta_0 + \beta_1 x)$). They would never be negative.

Models with Quadratics/Interaction Terms

Suppose we are interested in the relationship between AfD vote share and the youth unemployment rate.

First, we can observe the joint distribution.

```
plot(unemp.young,afd.pr)
```



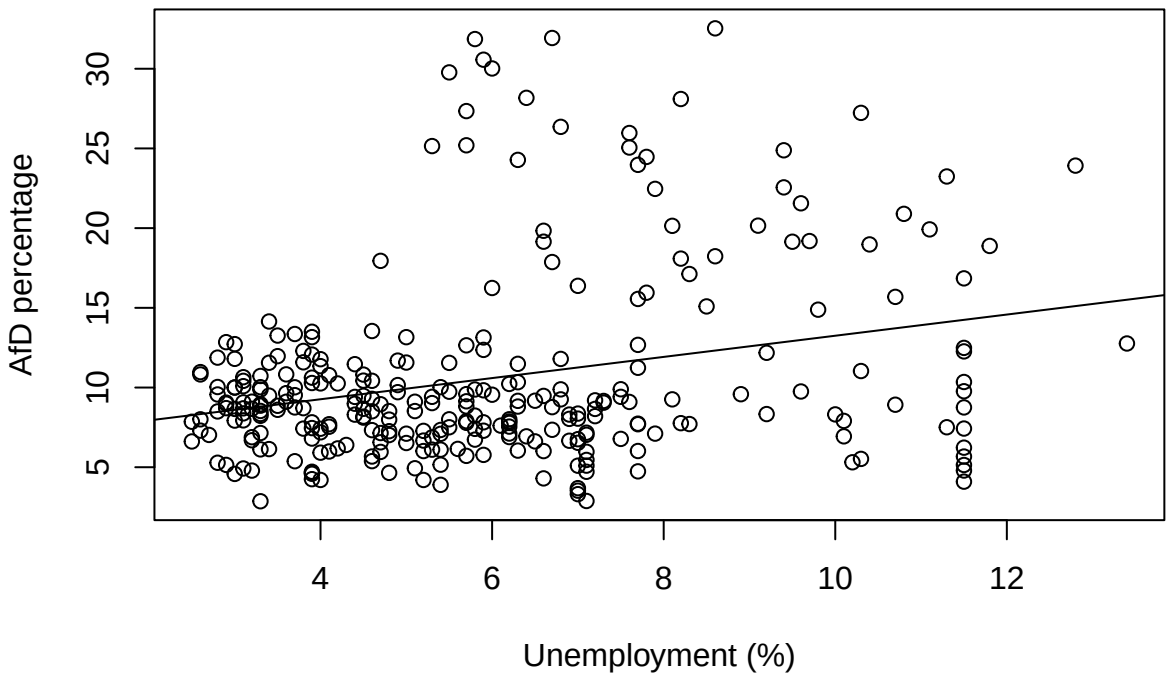
Subsequently, we can regress the AfD share on the unemployment rate:

```
srm.out <- lm(afd.pr ~ unemp.young)
summary(srm.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ unemp.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.151  -3.369  -1.258   1.760  21.401
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6072     0.8579   7.701 2.02e-13 ***
## unemp.young   0.6643     0.1343   4.947 1.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.663 on 297 degrees of freedom
## Multiple R-squared:  0.07612,    Adjusted R-squared:  0.07301
## F-statistic: 24.47 on 1 and 297 DF,  p-value: 1.266e-06
```

There seems to be a positive effect of the unemployment rate on the SPD vote share.

```
plot(unemp.young,afd.pr,xlab="Unemployment (%)",ylab="AfD percentage")
abline(srm.out)
```



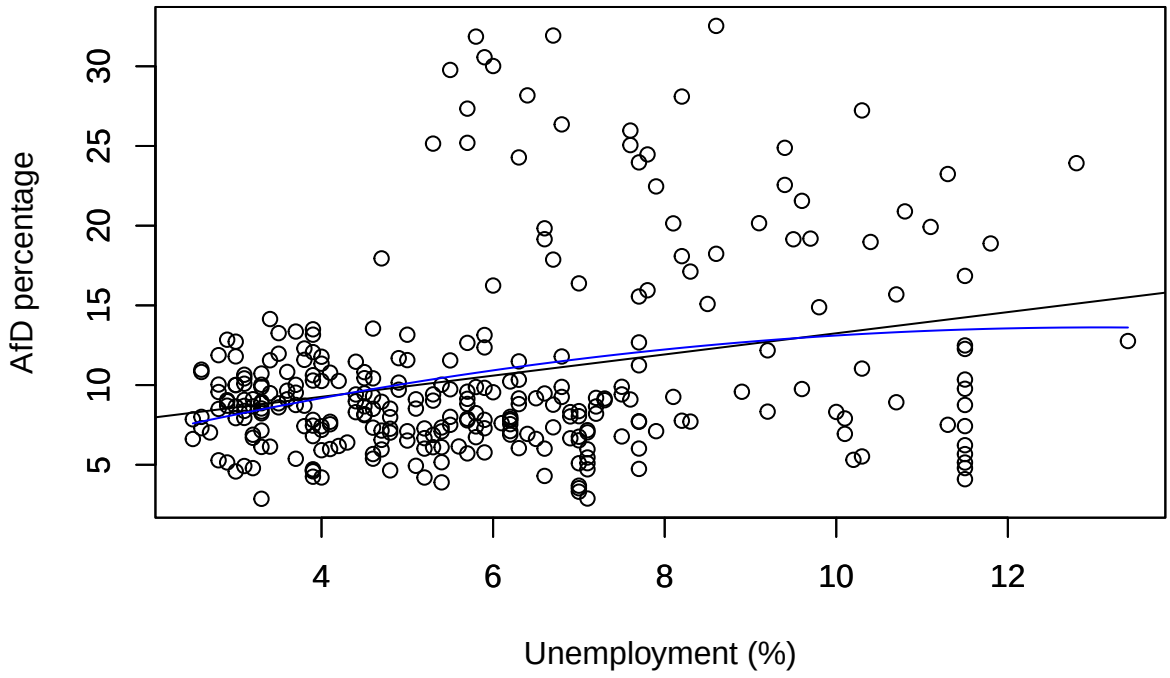
At the same time, there seems to be a curve-linear relationship. To test this, we add the quadratic term to the regression:

```
unemployment.qrd <- unemp.young^2

quadratics.out <- lm(afd.pr ~ unemp.young + unemployment.qrd)
summary(quadratics.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ unemp.young + unemployment.qrd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.390 -3.528 -1.241  1.851 21.092
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.40767    2.11591   2.083   0.0381 *
## unemp.young       1.40954    0.66901   2.107   0.0360 *
## unemployment.qrd -0.05393    0.04743  -1.137   0.2564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.66 on 296 degrees of freedom
## Multiple R-squared:  0.08013,    Adjusted R-squared:  0.07392
## F-statistic: 12.89 on 2 and 296 DF,  p-value: 4.278e-06
```

The result shows a curve linear relationship.

```
# make predition
xvalues <- seq(min(unemp.young),max(unemp.young),length=100)
predicted <- coefficients(quadratics.out)[1] +
             coefficients(quadratics.out)[2]*xvalues +
             coefficients(quadratics.out)[3]*(xvalues^2)

# plot
plot(unemp.young,afd.pr,xlab="Unemployment (%)",ylab="AfD percentage",ylim=range(afd.pr))
abline(srm.out)
par(new=T)
plot(xvalues,predicted,type="l",axes=F,ann=F,xlab="",ylab="",
     ylim=range(afd.pr),col="blue")
```



However, the quadratic term is not significant and if one compare both models by using F-statistic, the curve linear model is not superior to the simple linear model.

Still, in the mid of x-value range, the data tend to have positive residuals than the other ranges. Now, we take another approach with an interaction term. That is, we assume the effect of unemployment can vary depending on another independent variable "West or East Germany".

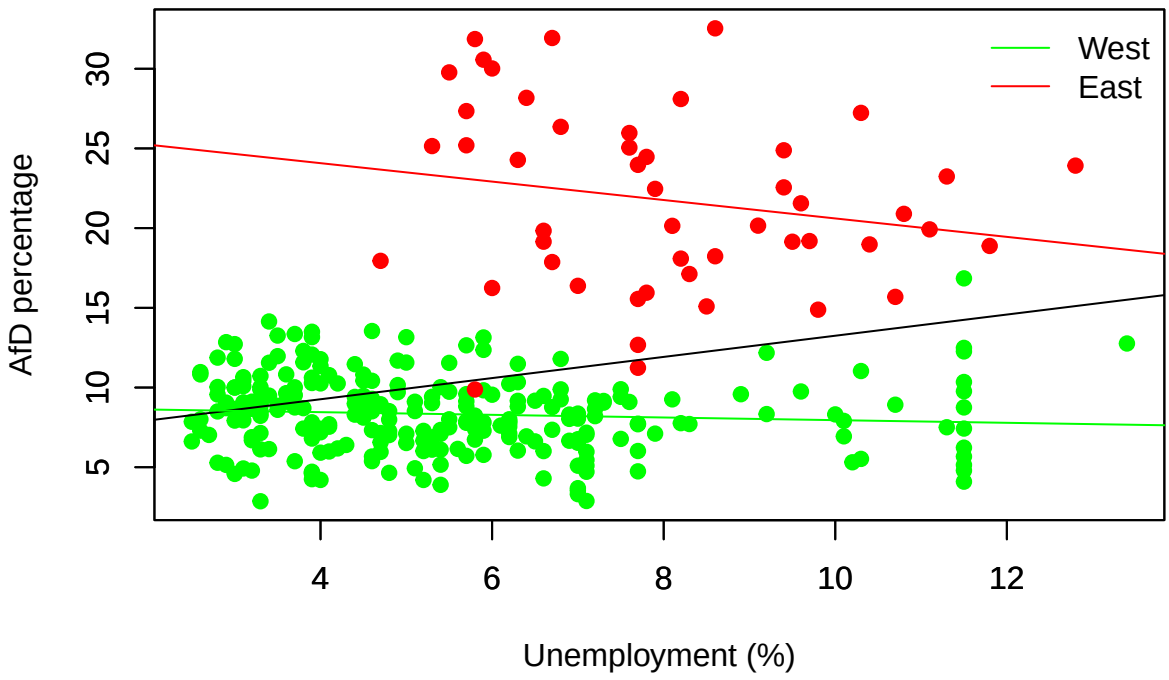
```
bundesland <- sociodem[,"Land"]
east.index <- c(grep("Mecklenburg",bundesland),
                grep("Brandenburg",bundesland),
                grep("Sachsen",bundesland),
                grep("ringen",bundesland))
```

```r
east.dummy <- rep(0,length(bundesland))
east.dummy[east.index] <- 1

interaction.out <- lm(afd.pr ~ unemp.young*east.dummy)
summary(interaction.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ unemp.young * east.dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1574  -1.6771  -0.0505   1.6751  11.1162
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              8.78873    0.50695  17.336   <2e-16 ***
## unemp.young             -0.08349    0.08509  -0.981   0.3273
## east.dummy              17.59871    2.00955   8.758   <2e-16 ***
## unemp.young:east.dummy -0.49446    0.25077  -1.972   0.0496 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.115 on 295 degrees of freedom
## Multiple R-squared:  0.7223, Adjusted R-squared:  0.7194
## F-statistic: 255.7 on 3 and 295 DF,  p-value: < 2.2e-16
```

The result is better interpreted by visualizing it:

```r
reg.coef.west <- coefficients(interaction.out)[c(1,2)]
reg.coef.east <- coefficients(interaction.out)[c(1,2)] + coefficients(interaction.out)[c(3,4)]

# plot
plot(unemp.young[east.dummy==0],afd.pr[east.dummy==0],
     xlab="Unemployment (%)",ylab="AfD percentage",
     col="green",pch=19,
     xlim=range(unemp.young),ylim=range(afd.pr))
par(new=T)
plot(unemp.young[east.dummy==1],afd.pr[east.dummy==1],
     col="red",pch=19,
     ann=F,axes=F,xlab="",ylab="",
     xlim=range(unemp.young),ylim=range(afd.pr))
abline(srm.out)
par(new=T)
#plot(xvalues,predicted,type="l",axes=F,ann=F,xlab="",ylab="",
#     ylim=range(afd.pr),col="blue")
abline(reg=reg.coef.west,col="green",lty=1)
abline(reg=reg.coef.east,col="red",lty=1)
legend("topright",lty=1,col=c("green","red"),
       c("West","East"),
       bty="n"
)
```



Suprizingly, the positive effect of youth unemployment above was due to the heterogeneity of West and East German districts. According to the current model with the interaction, yourh unemployment has no significant result in West and even a significant negative effect in East.

The effect of unemployment can depend on further factors, e.g. the GDP level:

```
interaction.2.out <- lm(afd.pr ~ unemp.young * gdp.tsd)
summary(interaction.2.out)
```

```
##
## Call:
## lm(formula = afd.pr ~ unemp.young * gdp.tsd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0004 -3.2636 -0.7657  1.5573 19.4828
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.38703    2.48634   0.156  0.87641
## unemp.young          2.85499    0.40965   6.969 2.08e-11 ***
## gdp.tsd              0.18668    0.06013   3.104  0.00209 **
## unemp.young:gdp.tsd -0.06203    0.01039  -5.968 6.89e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.958 on 295 degrees of freedom
```

```
## Multiple R-squared:  0.2965, Adjusted R-squared:  0.2893
## F-statistic: 41.44 on 3 and 295 DF,  p-value: < 2.2e-16
```

The partial effect of the unemployment rate on the AfD vote share is $2.85499 - 0.06203 \times gdp.tsd$. By considering the range of the GDP (in Tsd), we can obtain the range of the partial effect:

```
range.gdp.tsd <- range(gdp.tsd)
range.gdp.tsd
```

```
## [1]  21.403 110.283
```

```
range.partial.effect <- coefficients(interaction.2.out)[2] +
                        coefficients(interaction.2.out)[4]*range.gdp.tsd
range.partial.effect
```

```
## [1]  1.527277 -3.986321
```

For better interpretation, we can reparametrize the model:

```
unemployment.center <- unemp.young-mean(unemp.young)
gdp.tsd.center <- gdp.tsd - mean(gdp.tsd)

interaction.3.out <- lm(spd.pr ~ unemp.young + gdp.tsd +
                          unemployment.center:gdp.tsd.center)
summary(interaction.3.out)
```

```
##
## Call:
## lm(formula = spd.pr ~ unemp.young + gdp.tsd + unemployment.center:gdp.tsd.center)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9460  -3.5425   0.0841   3.6467  15.9990
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        23.20105    1.32606  17.496  < 2e-16 ***
## unemp.young                         0.87965    0.13310   6.609 1.81e-10 ***
## gdp.tsd                            -0.05710    0.02281  -2.503   0.0128 *
## unemployment.center:gdp.tsd.center  0.02726    0.01148   2.374   0.0182 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.476 on 295 degrees of freedom
## Multiple R-squared:  0.1773, Adjusted R-squared:  0.169
## F-statistic:  21.2 on 3 and 295 DF,  p-value: 1.84e-12
```

The main effect of the unemployment rate corresponds to the partial effect of the unemployment rate if the GDP in Tsd is its average value:

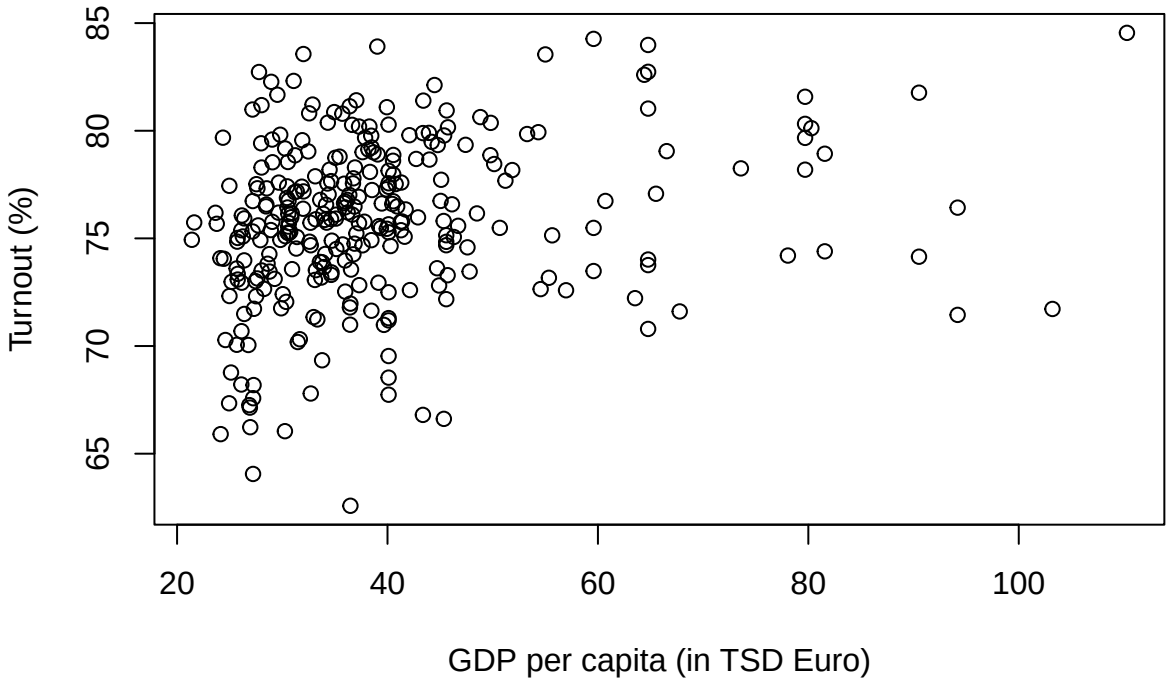
```
coefficients(interaction.2.out)[2] +
                        coefficients(interaction.2.out)[4]*mean(gdp.tsd)
```

```
## unemp.young
##    0.411155
```

Heteroskedasticity

Let's turn to the very first model which we estimated:

```
plot(gdp.tsd,turnout,xlab="GDP per capita (in TSD Euro)",ylab="Turnout (%)")
```



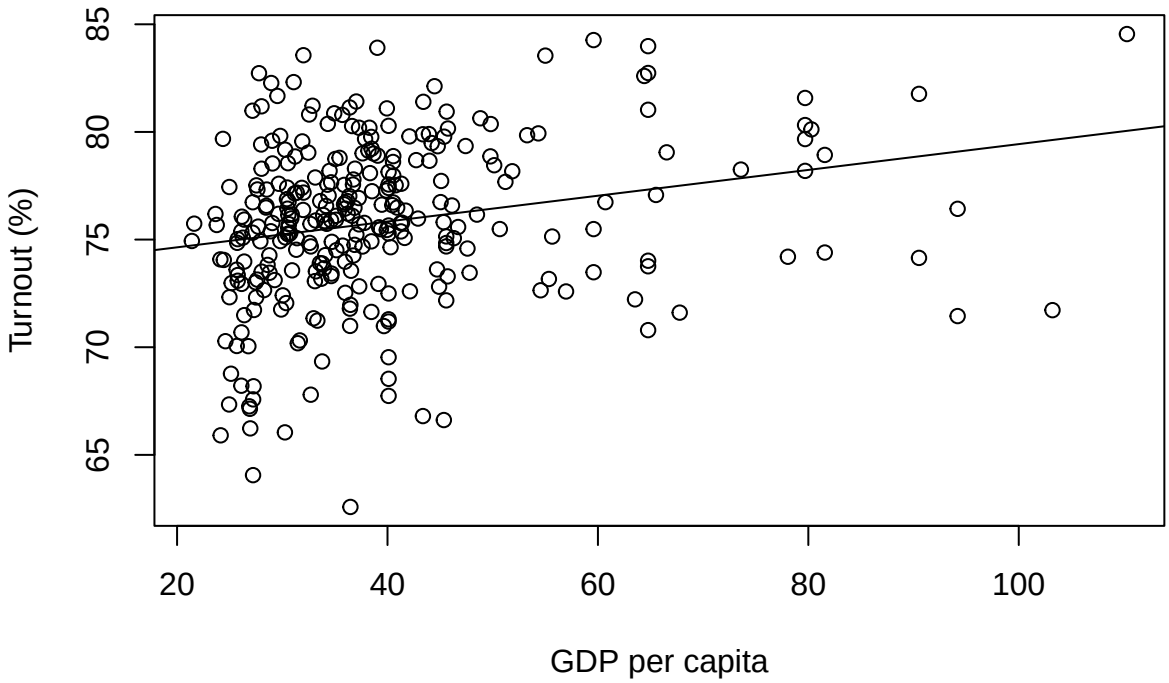
Simple regression models (Turnout on GDP)

```
lm.out.1 <- lm(turnout ~ gdp.tsd)
summary(lm.out.1)
```

```
##
## Call:
## lm(formula = turnout ~ gdp.tsd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0464  -2.0181   0.3747   2.3607   8.2044
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 73.44137    0.61704 119.021  < 2e-16 ***
## gdp.tsd      0.05993    0.01470   4.075  5.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.676 on 297 degrees of freedom
## Multiple R-squared:  0.05296,    Adjusted R-squared:  0.04977
## F-statistic: 16.61 on 1 and 297 DF,  p-value: 5.9e-05
```

Graphical presentation

```
plot(gdp.tsd,turnout,xlab="GDP per capita",ylab="Turnout (%)")
abline(lm.out.1)
```



Heteroskedasticity-Robust Inference after OLS Estimation

We first calculate the variance of the slope under the homoskedasticity assumption according to Eqn 8.2 but with σ^2 instead of σ_i^2.

For the first model (Turnout ~ GDP), we first obtain the variance of the slope and its square rood (the standard error):

```
var.slope.1 <- sum((gdp.tsd - mean(gdp.tsd))^2*(summary(lm.out.1)$sigma^2))/((sum((gdp.tsd - mean(gdp.ts

var.slope.1
```

```
## [1] 0.0002162074
```

```
sqrt(var.slope.1)
```

```
## [1] 0.01470399
```

The calculated standard error corresponds to that in the output before.

Now, we consider the different residual size of the individual observations. That is, we consider $\hat{u}_i^2$ as in Equation 8.3:

```
var.slope.1.het <- sum((gdp.tsd - mean(gdp.tsd))^2*(lm.out.1$residuals^2))/((sum((gdp.tsd - mean(gdp.tsd

var.slope.1.het
```

```
## [1] 0.00030328
```

```
sqrt(var.slope.1.het)
```

```
## [1] 0.01741494
```

This **heteroskedasticity-robust** standard error is larger than the previous naive one. If one checks the previous graphic of the joint distribution with the regression line, we can observe a larger variance of residuals in the districts with lower GDP than in those with higher GDP. This heteroskedasticity lead to the higher robust standard error.

Testing for Heteroskedasticity

Our null-hypothesis is that the homoskedasticity is true: $H_0 : Var(u|x_1, ..., x_k) = \sigma^2$, which is equvalent to: $H_0 : E(u^2|x_1, ..., x_k) = \sigma^2$.

This can be tested by regressing the squared residuals on the same independent variables of the regression model of interest:

```
resid.sq <- (lm.out.1$residuals)^2

resid.lm.out <- lm(resid.sq ~ gdp.tsd )
summary(resid.lm.out)
```

```
##
## Call:
## lm(formula = resid.sq ~ gdp.tsd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.947 -12.191  -8.323   2.442 156.968
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.95586    3.50777   3.123  0.00196 **
## gdp.tsd      0.06263    0.08359   0.749  0.45427
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.9 on 297 degrees of freedom
## Multiple R-squared:  0.001887,   Adjusted R-squared:  -0.001474
## F-statistic: 0.5615 on 1 and 297 DF,  p-value: 0.4543
```

In the output, we can just check the overall significance of regression. In this case, F=0.561 is not in the rejection region (for p=0.05). Therefore, we cannot reject the null-hypothesis.

Above, we tested the original homoskedastiticy assumption that $E(u|x) = 0$. We can instead the weaker version of the assumption $Cov(x, u) = 0$ (see Chapter 5). To do this, we can apply the White test, which regresses the squared residuals on the predicted values and the squared predicted values:

```
predict <- lm.out.1$fitted.values
predict.sq <- lm.out.1$fitted.values^2

white.lm.out <- lm(resid.sq ~ predict + predict.sq)
summary(white.lm.out)
```

```
##
## Call:
```

```
## lm(formula = resid.sq ~ predict + predict.sq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.629 -11.811  -8.650   3.832 157.694
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10486.1452  5732.6940   1.829   0.0684 .
## predict      -274.1607   149.4959  -1.834   0.0677 .
## predict.sq      1.7939     0.9745   1.841   0.0666 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.81 on 296 degrees of freedom
## Multiple R-squared:  0.01319,    Adjusted R-squared:  0.006518
## F-statistic: 1.978 on 2 and 296 DF,  p-value: 0.1402
```

Also here, we cannot reject the null hypothesis.

Below we repeat the same exercise for a multiple regression : turnout ~ gdp + young unemployment:

```
lm.out.3 <- lm(turnout ~ gdp.tsd + unemp.young )
summary(lm.out.3)
```

```
##
## Call:
## lm(formula = turnout ~ gdp.tsd + unemp.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9306 -1.7517 -0.0971  1.6969  9.8823
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 80.22770    0.67518 118.824  < 2e-16 ***
## gdp.tsd      0.03296    0.01152   2.862  0.00452 **
## unemp.young -0.96942    0.06829 -14.195  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.84 on 296 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.4327
## F-statistic: 114.7 on 2 and 296 DF,  p-value: < 2.2e-16
```

First, we regress the suqared residuals on all independent variables:

```
resid.sq <- (lm.out.3$residuals)^2

resid.lm.out <- lm(resid.sq ~ gdp.tsd + unemp.young)
summary(resid.lm.out)
```

```
##
## Call:
## lm(formula = resid.sq ~ gdp.tsd + unemp.young)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -21.193  -6.540  -3.351   2.796  85.756
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.88756    3.23950  -3.052 0.002478 **
## gdp.tsd      0.20479    0.05527   3.705 0.000252 ***
## unemp.young  1.66080    0.32767   5.068 7.08e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.63 on 296 degrees of freedom
## Multiple R-squared:  0.1035, Adjusted R-squared:  0.09739
## F-statistic: 17.08 on 2 and 296 DF,  p-value: 9.572e-08
```

In this case, we reject the null hypothesis for 5% significance level.

The White test gives the following result:

```
predict <- lm.out.3$fitted.values
predict.sq <- lm.out.3$fitted.values^2

white.lm.out <- lm(resid.sq ~ predict + predict.sq)
summary(white.lm.out)
```

```
##
## Call:
## lm(formula = resid.sq ~ predict + predict.sq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.895  -6.075  -4.060   1.479  87.025
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1402.9578   614.2330   2.284   0.0231 *
## predict      -36.0766    16.4105  -2.198   0.0287 *
## predict.sq     0.2329     0.1095   2.127   0.0343 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.98 on 296 degrees of freedom
## Multiple R-squared:  0.05673,    Adjusted R-squared:  0.05035
## F-statistic:   8.9 on 2 and 296 DF,  p-value: 0.0001764
```

Here, we can also reject the null-hypothesis. Note that we are testing two slightly different null-hypothesis. The first test with the original homoskedasticity assumption and the other the weaker version of that.

Weighted Least Squares Estimation

In the above, analysis, we saw the squared residuals are positively correlated with the share of youth unemployment (the variable *unemp.young*). Now we assume that the residual variance is proportional to this variable: $Var(u_i|unemp.young_i) = \sigma^2 unemp.young_i$.

Under this assumption, we divide all the variables by the square root of the "unemp.young" variable and estimate the coefficients:

```r
turnout.star <- turnout/sqrt(unemp.young)
interc.star <- 1/sqrt(unemp.young)
gdp.star <- gdp.tsd/sqrt(unemp.young)
unemp.young.star <- unemp.young/sqrt(unemp.young)



wls.out <- lm(turnout.star ~ 0 + interc.star + gdp.star + unemp.young.star )
summary(wls.out)
```

```
##
## Call:
## lm(formula = turnout.star ~ 0 + interc.star + gdp.star + unemp.young.star)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3300 -0.6998 -0.0597  0.7207  3.4313
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## interc.star      80.79445    0.62617 129.030   <2e-16 ***
## gdp.star          0.02721    0.01054   2.581   0.0103 *
## unemp.young.star -1.02706    0.07378 -13.920   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.184 on 296 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9988
## F-statistic: 8.349e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

This is the result of the Weighted Least Squares (WLS) estimation. To compare, we will check the OLS results again.

```r
summary(lm.out.3)
```

```
##
## Call:
## lm(formula = turnout ~ gdp.tsd + unemp.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9306 -1.7517 -0.0971  1.6969  9.8823
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 80.22770    0.67518 118.824  < 2e-16 ***
## gdp.tsd      0.03296    0.01152   2.862  0.00452 **
## unemp.young -0.96942    0.06829 -14.195  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.84 on 296 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.4327
## F-statistic: 114.7 on 2 and 296 DF,  p-value: < 2.2e-16
```

8.4 Feasible Generalized Least Squares esimation

WLS required a certain heteroskedasticity function ($Var(u_i|unemp.young_i) = \sigma^2 unemp.young_i$ in the above example).

In many cases, the exact form of the heteroskedasticity is not known. In such case, we can estimate the heteroskedasticity function.

We assume the following function:

$Var(u|x) = \sigma^2 \exp(\delta_0 + \delta_1 x_1 + ... + \delta_k x_k)$

with $\delta_0 ... \delta_k$ being unknown parameters.

The above equation can be reformulated as follows:

$Var(u|x) = E(u^2|x) = \sigma^2 \exp(\delta_0 + \delta_1 x_1 + ... + \delta_k x_k)$

By assuming ν a random variable with unit mean:

$u^2 = \sigma^2 \exp(\delta_0 + \delta_1 x_1 + ... + \delta_k x_k)\nu$.

Taking the log of both sides of the equation:

$\ln(u^2) = \ln(\sigma^2) + \delta_0 + \delta_1 x_1 + ... + \delta_k x_k + e$.

The parameter of this model can be obtained by regressing the log of the squared residuals on the independent variables:

```
resid.sq.log <- log((lm.out.3$residuals^2))

heterosk.func <- lm(resid.sq.log ~ gdp.tsd + unemp.young)
summary(heterosk.func)
```

```
##
## Call:
## lm(formula = resid.sq.log ~ gdp.tsd + unemp.young)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7097  -0.8900   0.3808   1.5960   3.7061
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.78696    0.55510  -1.418   0.1573
## gdp.tsd      0.02086    0.00947   2.202   0.0284 *
## unemp.young  0.09906    0.05615   1.764   0.0787 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.335 on 296 degrees of freedom
## Multiple R-squared:  0.02268,    Adjusted R-squared:  0.01607
## F-statistic: 3.434 on 2 and 296 DF,  p-value: 0.03355
```

Here, the estimated parameters are not important, but the predicted values since we are interested in the weights of individual observations. They can be obtained by exponentiating the predicted values:

```
weights <- exp(heterosk.func$fitted.values)
```

Now we can conduct WLS by using the obtained weights:

```
turnout.star <- turnout/sqrt(weights)
interc.star <- 1/sqrt(weights)
gdp.star <- gdp.tsd/sqrt(weights)
unemp.young.star <- unemp.young/sqrt(weights)


fgls.out <- lm(turnout.star ~ 0 + interc.star + gdp.star + unemp.young.star )
summary(fgls.out)
```

```
##
## Call:
## lm(formula = turnout.star ~ 0 + interc.star + gdp.star + unemp.young.star)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2296 -1.1604 -0.0848  1.2928  5.6690
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## interc.star      80.64508    0.70786 113.928   <2e-16 ***
## gdp.star          0.03179    0.01402   2.267   0.0241 *
## unemp.young.star -1.03545    0.06937 -14.927   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.967 on 296 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9988
## F-statistic: 8.494e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

To check, we can compare the results with those based on WLS and OLS.

```
summary(lm.out.3)
```

```
##
## Call:
## lm(formula = turnout ~ gdp.tsd + unemp.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9306 -1.7517 -0.0971  1.6969  9.8823
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 80.22770    0.67518 118.824  < 2e-16 ***
## gdp.tsd      0.03296    0.01152   2.862  0.00452 **
## unemp.young -0.96942    0.06829 -14.195  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.84 on 296 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.4327
## F-statistic: 114.7 on 2 and 296 DF,  p-value: < 2.2e-16
```

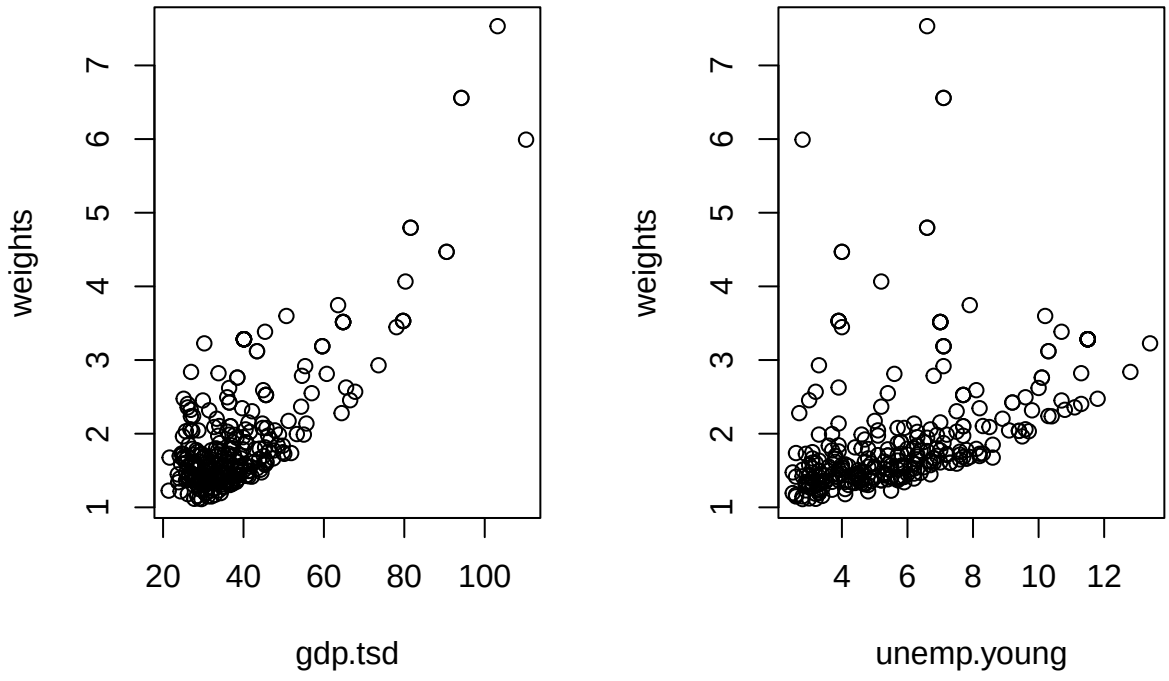
```
summary(wls.out)
```

```
##
```

```
## Call:
## lm(formula = turnout.star ~ 0 + interc.star + gdp.star + unemp.young.star)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3300 -0.6998 -0.0597  0.7207  3.4313
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## interc.star      80.79445    0.62617 129.030   <2e-16 ***
## gdp.star          0.02721    0.01054   2.581   0.0103 *
## unemp.young.star -1.02706    0.07378 -13.920   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.184 on 296 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9988
## F-statistic: 8.349e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

Another possible comparison can be done based on the weights.

```
par(mfrow=c(1,2))
plot(weights ~ gdp.tsd)

plot(weights ~ unemp.young)
```



The weights used in FGLS is strongly correlated with the GDP variable, which was not used for the WLS

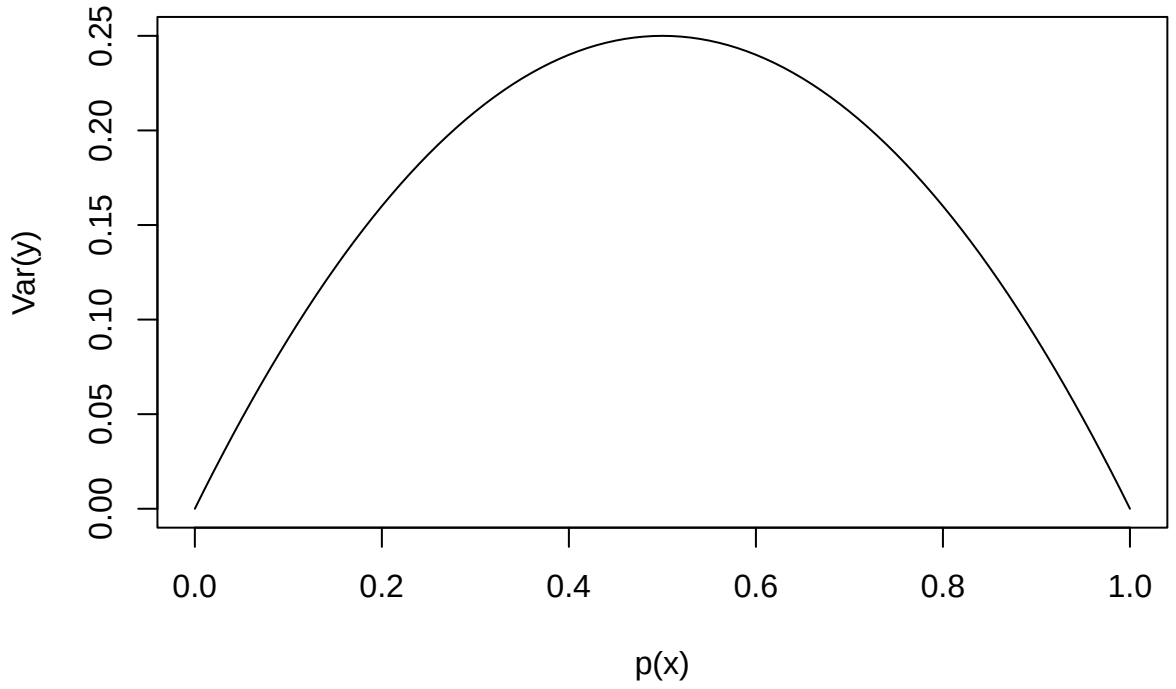
estimation.

8.5 The Linear Probability Model Revisited

The linear probability model must contain heteroskedasticity (except $\beta_j = 0$ for all j):

$Var(y|x) = p(x)[1 - p(x)]$ with $p(x) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$.

```
var.function <- function(p) p*(1-p)
curve(var.function,0,1,xlab="p(x)",ylab="Var(y)")
```



Two possible remedies:

- OLS-estimates + robust standard errors.
- WLS-estimates with the weights $\hat{h} = \hat{y}(1 - \hat{y})$.

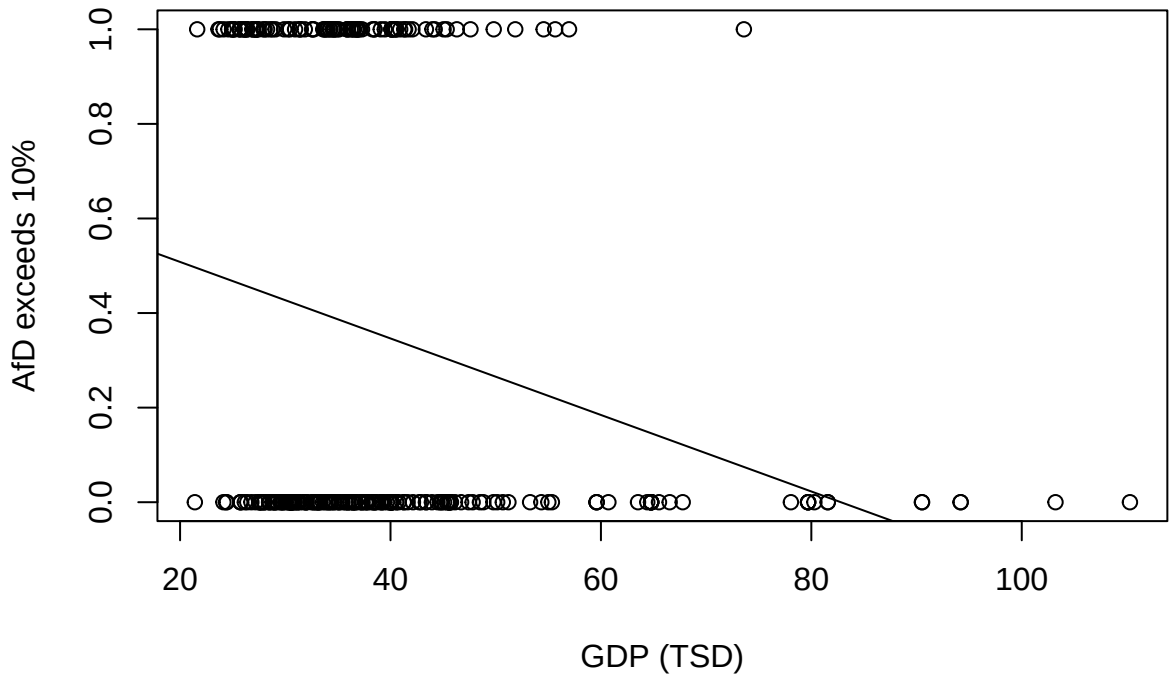
For example, we regress whether AfD exceeded 10% valid votes on GDP:

```
afd.10 <- ifelse(afd.pr>10,1,0)

lpm.out <- lm(afd.10 ~ gdp.tsd)
summary(lpm.out)
```

```
##
## Call:
## lm(formula = afd.10 ~ gdp.tsd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -0.4967 -0.3918 -0.3000  0.5542  0.9254
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.669870   0.077941   8.595 4.85e-16 ***
## gdp.tsd     -0.008090   0.001857  -4.356 1.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4643 on 297 degrees of freedom
## Multiple R-squared:  0.06004,    Adjusted R-squared:  0.05688
## F-statistic: 18.97 on 1 and 297 DF,  p-value: 1.829e-05
```

```
plot(afd.10 ~ gdp.tsd,xlab="GDP (TSD)",ylab="AfD exceeds 10%")
abline(lpm.out)
```



The (heteroskedasticits-) robust standard error is:

```
var.slope.lpm.het <- sum((gdp.tsd - mean(gdp.tsd))^2*(lpm.out$residuals^2))/((sum((gdp.tsd - mean(gdp.ts

var.slope.lpm.het
```

```
## [1] 1.610914e-06
```

```
sqrt(var.slope.lpm.het)
```

```
## [1] 0.001269218
```

The WLS estimates are:

```
predicted <- lpm.out$fitted.values
h.hat <- predicted * (1-predicted)

wls.lpm.out <- lm(afd.10 ~ gdp.tsd,weights = 1/h.hat)
```

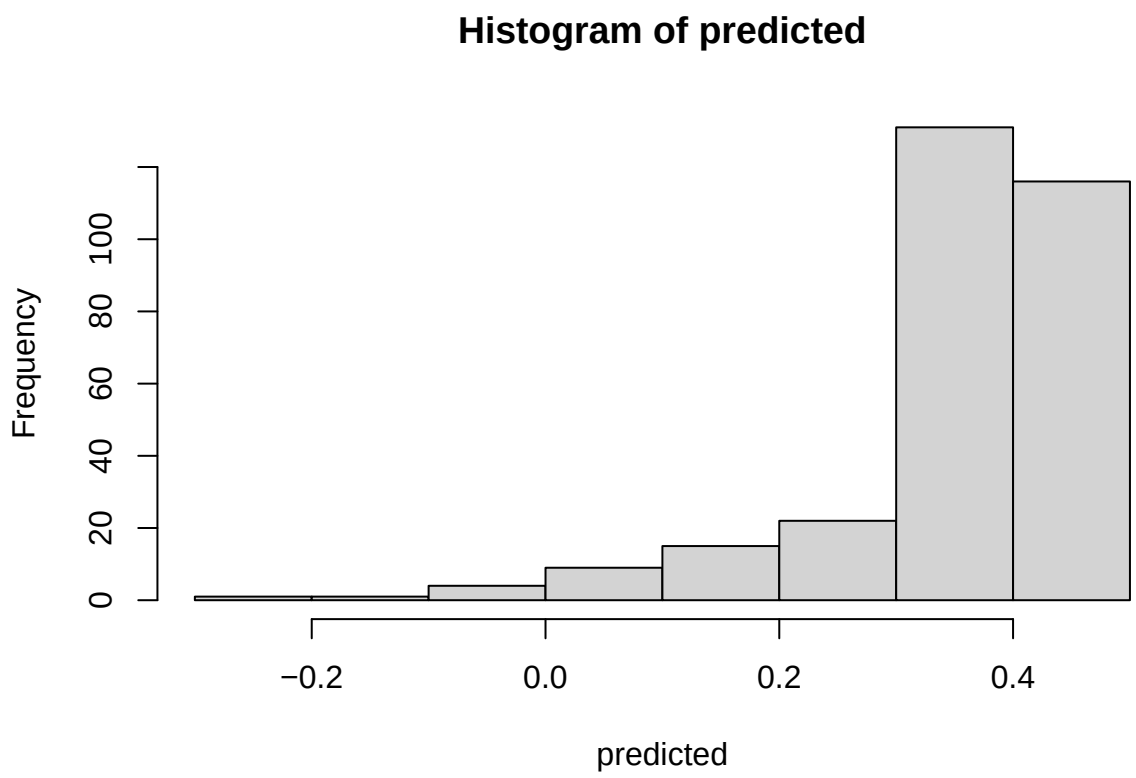
```
## Error in lm.wfit(x, y, w, offset = offset, singular.ok = singular.ok, : missing or negative weights n
```

```
summary(wls.lpm.out)
```

```
## Error in summary(wls.lpm.out): Objekt 'wls.lpm.out' nicht gefunden
```

This does not work since some predicted values are not in the range $(0, 1)$.

```
predicted <- lpm.out$fitted.values
hist(predicted)
```



The figure shows that some predictions are under zero. This makes $\hat{h}$ negative values, which cannot be used as weight.

One possibility is to adjust those predicted values. We can replace the predicted values below zero with 0.01:

```
predicted <- lpm.out$fitted.values
predicted[predicted<=0] <- 0.01

h.hat <- predicted * (1-predicted)

wls.lpm.out <- lm(afd.10 ~ gdp.tsd,
                  weights = 1/h.hat)
summary(wls.lpm.out)
```

```
##
## Call:
## lm(formula = afd.10 ~ gdp.tsd, weights = 1/h.hat)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9301 -0.7763 -0.6626  1.1647  3.3557
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6071074  0.0500989  12.118   <2e-16 ***
## gdp.tsd     -0.0066383  0.0007395  -8.977   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9894 on 297 degrees of freedom
## Multiple R-squared:  0.2134, Adjusted R-squared:  0.2108
## F-statistic: 80.58 on 1 and 297 DF,  p-value: < 2.2e-16
```

The replaced value 0.01 was arbitrarily chosen. We can also use 0.001:

```
predicted <- lpm.out$fitted.values
predicted[predicted<=0] <- 0.001

h.hat <- predicted * (1-predicted)

wls.lpm.out <- lm(afd.10 ~ gdp.tsd,
                  weights = 1/h.hat)
summary(wls.lpm.out)
```

```
##
## Call:
## lm(formula = afd.10 ~ gdp.tsd, weights = 1/h.hat)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1577 -0.7292 -0.6332  1.2280  3.3037
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5480434  0.0444869   12.32   <2e-16 ***
## gdp.tsd     -0.0056503  0.0004971  -11.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.008 on 297 degrees of freedom
## Multiple R-squared:  0.3031, Adjusted R-squared:  0.3008
## F-statistic: 129.2 on 1 and 297 DF,  p-value: < 2.2e-16
```

Here, the result is different since we have a significant amount of predicted values outside of $(0, 1)$. In such cases, we should better use the heteroskedasticity-robust statistics.

```
summary(lpm.out)
```

```
##
## Call:
```

```
## lm(formula = afd.10 ~ gdp.tsd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4967 -0.3918 -0.3000  0.5542  0.9254
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.669870   0.077941   8.595 4.85e-16 ***
## gdp.tsd     -0.008090   0.001857  -4.356 1.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4643 on 297 degrees of freedom
## Multiple R-squared:  0.06004,    Adjusted R-squared:  0.05688
## F-statistic: 18.97 on 1 and 297 DF,  p-value: 1.829e-05
```

```
robust.se.srm(lpm.out)
```

```
## [1] 0.001269218
```

Pooled-cross sectional data

At the Bundestag election 2021, the Greens were not allowed to run for PR (Zweitstimme) in Saarland due to a formal defect, while they could run candidates for SMDs (Erststimme). We are now interested whether this incident had a negative effect on the Greens' results in SMDs in Saarland.

If we regress the Greens' results in SMDs on the dummy variable for Saarland, we obtain a significant negative effect.

```
saar.dummy.21 <- ifelse(result21$`gehoert zu `==10,1,0)

summary(lm(gru.smd ~ saar.dummy.21))
```

```
##
## Call:
## lm(formula = gru.smd ~ saar.dummy.21)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6032  -4.2075  -0.9798   3.0042  26.1693
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.7407     0.3931  34.952   <2e-16 ***
## saar.dummy.21  -8.3096     3.3989  -2.445   0.0151 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.752 on 297 degrees of freedom
## Multiple R-squared:  0.01973,    Adjusted R-squared:  0.01643
## F-statistic: 5.977 on 1 and 297 DF,  p-value: 0.01507
```

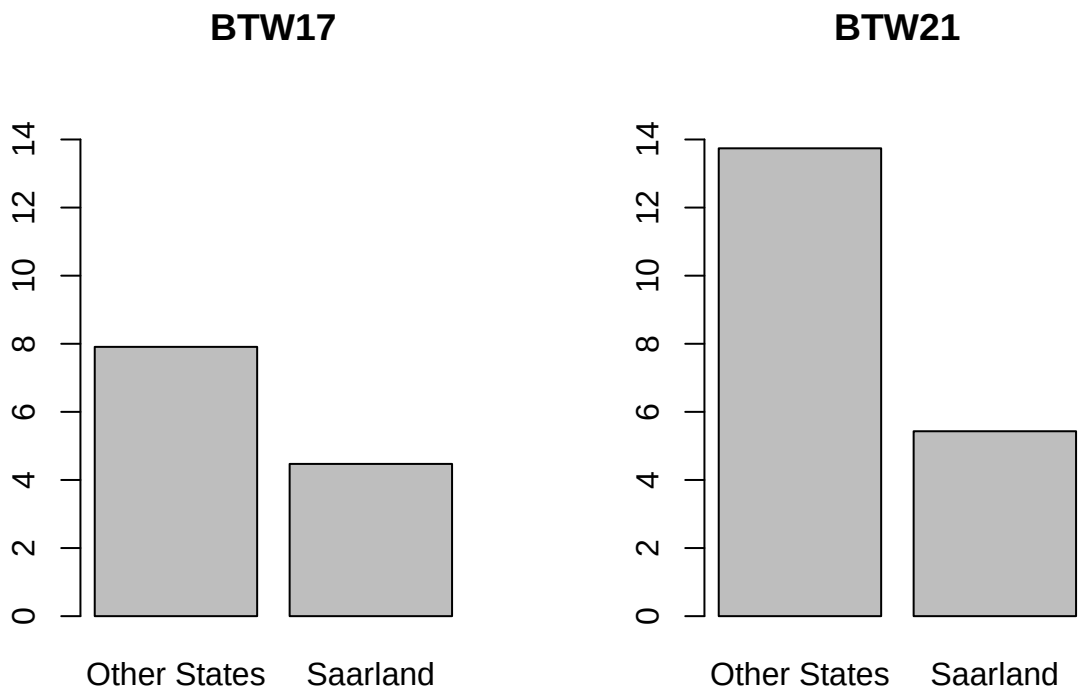
This result can be however misleading since the Greens may have obtained less support in Saarland for a longer time. To check this, we can estimate the same model by using the 2017 election result:

```r
saar.dummy.17 <- ifelse(result17$`gehoert zu `==10,1,0)

summary(lm(gru.smd.17 ~ saar.dummy.17))
```

```
##
## Call:
## lm(formula = gru.smd.17 ~ saar.dummy.17)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9479 -2.8680 -0.9802  1.6892 21.7376
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.9089     0.2421  32.661   <2e-16 ***
## saar.dummy.17  -3.4377     2.0901  -1.645    0.101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.152 on 296 degrees of freedom
##   (1 Beobachtung als fehlend gelöscht)
## Multiple R-squared:  0.009057,   Adjusted R-squared:  0.005709
## F-statistic: 2.705 on 1 and 296 DF,  p-value: 0.1011
```

Indeed, the Greens had less support in Saarland than in other federal states, while the difference is not significant.

```r
par(mfrow=c(1,2))

for (i.fig in 1:2){

  if (i.fig ==1 ) means <- tapply(gru.smd.17,as.factor(saar.dummy.17),mean,na.rm=T)
  if (i.fig ==2 ) means <- tapply(gru.smd,as.factor(saar.dummy.21),mean,na.rm=T)

  names(means) <- c("Other States","Saarland")

  barplot(means,main=c("BTW17","BTW21")[i.fig],ylim=c(0,15))

}
```



Now, we wonder whether the significant negative effect in the first model is larger than the latter negative effect. This can be checked by pooling both data and estimate the following model:

$$y = \beta_0 + \delta_0 d2 + \beta_1 dT + \delta_1 d2dT$$

, where $d2$ denotes the dummy variable for the 2021 election and dT denotes the dummy variable for the districts in Saarland:

```
gru.pooled <- rbind(cbind(result17$`Nr `,0,saar.dummy.17,gru.smd.17),
                    cbind(result21$`Nr `,1,saar.dummy.21,gru.smd))


colnames(gru.pooled) <- c("Dist","BTW21","Saar","gru.smd.pooled")
gru.pooled <- as.data.frame(gru.pooled)

summary(lm(gru.smd.pooled ~ BTW21 * Saar,data=gru.pooled))
```

```
##
## Call:
## lm(formula = gru.smd.pooled ~ BTW21 * Saar, data = gru.pooled)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6032  -3.3683  -0.9798   2.0209  26.1693
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9089     0.3270  24.185   <2e-16 ***
```

```
## BTW21         5.8318     0.4621  12.621   <2e-16 ***
## Saar         -3.4377     2.8225  -1.218    0.224
## BTW21:Saar   -4.8719     3.9916  -1.221    0.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.607 on 593 degrees of freedom
##   (1 Beobachtung als fehlend gelöscht)
## Multiple R-squared:  0.2208, Adjusted R-squared:  0.2168
## F-statistic: 56.01 on 3 and 593 DF,  p-value: < 2.2e-16
```

Obviously, the last coefficient corresponds to the difference between both negative effects estimated above, which is the difference-in-difference estimator. This effect is not significant, which means that the incident in Saarland at the 2021 election had no significant effect on their SMD results.

In the above analysis, we only analyzed the pooled data and did not consider that the heterogeneity of individual districts. That is, each of 299 districts has different characteristics which can affect the election outcomes. To account these factors, we just add the dummy variable for each district:

```
summary(fixed.lm.out <- lm(gru.smd.pooled ~ BTW21 * Saar + as.factor(Dist),
          data=gru.pooled))
```

```
##
## Call:
## lm(formula = gru.smd.pooled ~ BTW21 * Saar + as.factor(Dist),
##     data = gru.pooled)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.441 -1.201  0.000  1.201  7.441
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         16.38653    1.92001   8.535 7.44e-16 ***
## BTW21                5.85501    0.22357  26.188  < 2e-16 ***
## Saar               -11.94339    2.87730  -4.151 4.33e-05 ***
## as.factor(Dist)2    -7.44750    2.71070  -2.747 0.006375 **
## as.factor(Dist)3    -9.78055    2.71070  -3.608 0.000362 ***
## as.factor(Dist)4    -7.43825    2.71070  -2.744 0.006439 **
## as.factor(Dist)5     1.87145    2.71070   0.690 0.490488
## as.factor(Dist)6    -6.99314    2.71070  -2.580 0.010367 *
## as.factor(Dist)7    -6.75040    2.71070  -2.490 0.013313 *
## as.factor(Dist)8    -8.30857    2.71070  -3.065 0.002377 **
## as.factor(Dist)9    -9.13958    2.71070  -3.372 0.000846 ***
## as.factor(Dist)10   -5.76788    2.71070  -2.128 0.034178 *
## as.factor(Dist)11   -2.17171    2.71070  -0.801 0.423680
## as.factor(Dist)12  -13.52434    2.71070  -4.989 1.04e-06 ***
## as.factor(Dist)13  -14.46325    2.71070  -5.336 1.90e-07 ***
## as.factor(Dist)14  -11.66798    2.71070  -4.304 2.28e-05 ***
## as.factor(Dist)15  -14.11722    2.71070  -5.208 3.58e-07 ***
## as.factor(Dist)16  -16.03176    2.71070  -5.914 9.19e-09 ***
## as.factor(Dist)17  -14.96122    2.71070  -5.519 7.44e-08 ***
## as.factor(Dist)18    0.07522    2.71070   0.028 0.977880
## as.factor(Dist)19    2.66717    2.71070   0.984 0.325948
## as.factor(Dist)20    3.04942    2.71070   1.125 0.261519
## as.factor(Dist)21    0.27606    2.71070   0.102 0.918951
```

```
## as.factor(Dist)22   -8.08320    2.71070  -2.982 0.003102 **
## as.factor(Dist)23   -7.73892    2.71070  -2.855 0.004608 **
## as.factor(Dist)24  -11.20037    2.71070  -4.132 4.69e-05 ***
## as.factor(Dist)25  -12.23988    2.71070  -4.515 9.14e-06 ***
## as.factor(Dist)26  -11.49404    2.71070  -4.240 2.99e-05 ***
## as.factor(Dist)27   -3.46236    2.71070  -1.277 0.202499
## as.factor(Dist)28   -8.94615    2.71070  -3.300 0.001084 **
## as.factor(Dist)29  -10.78978    2.71070  -3.980 8.66e-05 ***
## as.factor(Dist)30   -9.22538    2.71070  -3.403 0.000757 ***
## as.factor(Dist)31  -10.53913    2.71070  -3.888 0.000125 ***
## as.factor(Dist)32  -11.83442    2.71070  -4.366 1.75e-05 ***
## as.factor(Dist)33   -8.74549    2.71070  -3.226 0.001395 **
## as.factor(Dist)34   -8.67175    2.71070  -3.199 0.001528 **
## as.factor(Dist)35  -12.87430    2.71070  -4.749 3.18e-06 ***
## as.factor(Dist)36   -7.28396    2.71070  -2.687 0.007615 **
## as.factor(Dist)37    0.56602    2.71070   0.209 0.834740
## as.factor(Dist)38   -8.13339    2.71070  -3.000 0.002925 **
## as.factor(Dist)39   -2.82280    2.71070  -1.041 0.298560
## as.factor(Dist)40   -9.08677    2.71070  -3.352 0.000906 ***
## as.factor(Dist)41   -3.87867    2.71070  -1.431 0.153521
## as.factor(Dist)42   -0.48742    2.71070  -0.180 0.857422
## as.factor(Dist)43   -9.82986    2.71070  -3.626 0.000338 ***
## as.factor(Dist)44  -10.45136    2.71070  -3.856 0.000142 ***
## as.factor(Dist)45  -11.74530    2.71070  -4.333 2.02e-05 ***
## as.factor(Dist)46  -11.56859    2.71070  -4.268 2.66e-05 ***
## as.factor(Dist)47  -10.23408    2.71070  -3.775 0.000193 ***
## as.factor(Dist)48   -8.35458    2.71070  -3.082 0.002249 **
## as.factor(Dist)49  -11.90124    2.71070  -4.390 1.58e-05 ***
## as.factor(Dist)50   -5.17684    2.71070  -1.910 0.057128 .
## as.factor(Dist)51  -12.69819    2.71070  -4.684 4.28e-06 ***
## as.factor(Dist)52  -11.34706    2.71070  -4.186 3.75e-05 ***
## as.factor(Dist)53   -2.10559    2.71070  -0.777 0.437915
## as.factor(Dist)54   -2.65893    2.71070  -0.981 0.327442
## as.factor(Dist)55   -8.18148    2.71070  -3.018 0.002764 **
## as.factor(Dist)56  -15.49285    2.71070  -5.715 2.67e-08 ***
## as.factor(Dist)57  -14.56054    2.71070  -5.372 1.58e-07 ***
## as.factor(Dist)58  -11.73544    2.71070  -4.329 2.05e-05 ***
## as.factor(Dist)59  -13.27445    2.71070  -4.897 1.60e-06 ***
## as.factor(Dist)60  -14.23210    2.71070  -5.250 2.90e-07 ***
## as.factor(Dist)61   -5.81720    2.71070  -2.146 0.032685 *
## as.factor(Dist)62  -13.58990    2.71070  -5.013 9.22e-07 ***
## as.factor(Dist)63  -14.52934    2.71070  -5.360 1.68e-07 ***
## as.factor(Dist)64  -15.81135    2.71070  -5.833 1.43e-08 ***
## as.factor(Dist)65  -16.11714    2.71070  -5.946 7.74e-09 ***
## as.factor(Dist)66  -15.44379    2.71070  -5.697 2.94e-08 ***
## as.factor(Dist)67  -16.01487    2.71070  -5.908 9.50e-09 ***
## as.factor(Dist)68  -15.43197    2.71070  -5.693 3.01e-08 ***
## as.factor(Dist)69  -12.69084    2.71070  -4.682 4.34e-06 ***
## as.factor(Dist)70  -13.66297    2.71070  -5.040 8.10e-07 ***
## as.factor(Dist)71  -16.66488    2.71070  -6.148 2.54e-09 ***
## as.factor(Dist)72  -13.24024    2.71070  -4.884 1.70e-06 ***
## as.factor(Dist)73  -16.38804    2.71070  -6.046 4.47e-09 ***
## as.factor(Dist)74  -16.54043    2.71070  -6.102 3.28e-09 ***
## as.factor(Dist)75    4.89365    2.71070   1.805 0.072042 .
```

```
## as.factor(Dist)76    0.47221    2.71070   0.174 0.861825
## as.factor(Dist)77   -8.37569    2.71070  -3.090 0.002193 **
## as.factor(Dist)78  -10.30091    2.71070  -3.800 0.000176 ***
## as.factor(Dist)79   -1.85348    2.71070  -0.684 0.494658
## as.factor(Dist)80   -0.31719    2.71070  -0.117 0.906928
## as.factor(Dist)81    2.68030    2.71070   0.989 0.323576
## as.factor(Dist)82   -3.94953    2.71070  -1.457 0.146172
## as.factor(Dist)83   12.71451    2.71070   4.690 4.17e-06 ***
## as.factor(Dist)84  -11.64101    2.71070  -4.294 2.38e-05 ***
## as.factor(Dist)85  -14.59257    2.71070  -5.383 1.49e-07 ***
## as.factor(Dist)86   -9.56884    2.71070  -3.530 0.000482 ***
## as.factor(Dist)87    0.42794    2.71070   0.158 0.874665
## as.factor(Dist)88  -11.13962    2.71070  -4.110 5.14e-05 ***
## as.factor(Dist)89  -11.50542    2.71070  -4.244 2.94e-05 ***
## as.factor(Dist)90  -11.91294    2.71070  -4.395 1.55e-05 ***
## as.factor(Dist)91  -10.01841    2.71070  -3.696 0.000261 ***
## as.factor(Dist)92   -9.90724    2.71070  -3.655 0.000304 ***
## as.factor(Dist)93   -2.50883    2.71070  -0.926 0.355446
## as.factor(Dist)94    5.26132    2.71070   1.941 0.053214 .
## as.factor(Dist)95    1.36607    2.71070   0.504 0.614669
## as.factor(Dist)96   -2.51682    2.71070  -0.928 0.353917
## as.factor(Dist)97   -9.06542    2.71070  -3.344 0.000931 ***
## as.factor(Dist)98   -7.69145    2.71070  -2.837 0.004862 **
## as.factor(Dist)99  -10.26189    2.71070  -3.786 0.000186 ***
## as.factor(Dist)100  -6.89102    2.71070  -2.542 0.011527 *
## as.factor(Dist)101 -10.82656    2.71070  -3.994 8.20e-05 ***
## as.factor(Dist)102  -9.10601    2.71070  -3.359 0.000884 ***
## as.factor(Dist)103  -9.95765    2.71070  -3.673 0.000284 ***
## as.factor(Dist)104  -4.45386    2.71070  -1.643 0.101431
## as.factor(Dist)105  -9.05996    2.71070  -3.342 0.000938 ***
## as.factor(Dist)106  -4.53099    2.71070  -1.672 0.095675 .
## as.factor(Dist)107  -4.50587    2.71070  -1.662 0.097520 .
## as.factor(Dist)108 -11.14641    2.71070  -4.112 5.09e-05 ***
## as.factor(Dist)109  -9.59058    2.71070  -3.538 0.000468 ***
## as.factor(Dist)110  -8.13974    2.71070  -3.003 0.002903 **
## as.factor(Dist)111  -8.95798    2.71070  -3.305 0.001068 **
## as.factor(Dist)112  -9.94515    2.71070  -3.669 0.000289 ***
## as.factor(Dist)113 -10.31293    2.71070  -3.805 0.000173 ***
## as.factor(Dist)114  -9.61210    2.71070  -3.546 0.000455 ***
## as.factor(Dist)115  -9.40170    2.71070  -3.468 0.000601 ***
## as.factor(Dist)116 -11.63133    2.71070  -4.291 2.41e-05 ***
## as.factor(Dist)117 -10.54825    2.71070  -3.891 0.000123 ***
## as.factor(Dist)118  -9.19449    2.71070  -3.392 0.000788 ***
## as.factor(Dist)119 -10.58119    2.71070  -3.903 0.000117 ***
## as.factor(Dist)120  -5.75794    2.71070  -2.124 0.034486 *
## as.factor(Dist)121 -11.52612    2.71070  -4.252 2.84e-05 ***
## as.factor(Dist)122 -11.60858    2.71070  -4.283 2.50e-05 ***
## as.factor(Dist)123 -11.93773    2.71070  -4.404 1.49e-05 ***
## as.factor(Dist)124 -10.66058    2.71070  -3.933 0.000105 ***
## as.factor(Dist)125 -12.01818    2.71070  -4.434 1.31e-05 ***
## as.factor(Dist)126  -9.76332    2.71070  -3.602 0.000371 ***
## as.factor(Dist)127  -6.79594    2.71070  -2.507 0.012709 *
## as.factor(Dist)128  -8.30012    2.71070  -3.062 0.002401 **
## as.factor(Dist)129   3.21988    2.71070   1.188 0.235848
```

```
## as.factor(Dist)130 -10.09593    2.71070  -3.724 0.000234 ***
## as.factor(Dist)131 -10.05332    2.71070  -3.709 0.000249 ***
## as.factor(Dist)132  -3.64870    2.71070  -1.346 0.179321
## as.factor(Dist)133 -11.43630    2.71070  -4.219 3.27e-05 ***
## as.factor(Dist)134 -11.61092    2.71070  -4.283 2.49e-05 ***
## as.factor(Dist)135  -8.11547    2.71070  -2.994 0.002987 **
## as.factor(Dist)136 -12.03025    2.71070  -4.438 1.28e-05 ***
## as.factor(Dist)137  -9.81348    2.71070  -3.620 0.000346 ***
## as.factor(Dist)138 -11.13016    2.71070  -4.106 5.21e-05 ***
## as.factor(Dist)139  -7.07720    2.71070  -2.611 0.009492 **
## as.factor(Dist)140  -6.27175    2.71070  -2.314 0.021369 *
## as.factor(Dist)141 -10.69747    2.71070  -3.946 9.92e-05 ***
## as.factor(Dist)142  -7.17984    2.71070  -2.649 0.008514 **
## as.factor(Dist)143  -8.77451    2.71070  -3.237 0.001345 **
## as.factor(Dist)144  -9.18367    2.71070  -3.388 0.000800 ***
## as.factor(Dist)145 -12.06725    2.71070  -4.452 1.21e-05 ***
## as.factor(Dist)146 -10.35237    2.71070  -3.819 0.000163 ***
## as.factor(Dist)147 -13.18274    2.71070  -4.863 1.88e-06 ***
## as.factor(Dist)148 -12.01249    2.71070  -4.432 1.32e-05 ***
## as.factor(Dist)149 -13.16570    2.71070  -4.857 1.93e-06 ***
## as.factor(Dist)150 -12.94621    2.71070  -4.776 2.82e-06 ***
## as.factor(Dist)151 -15.49921    2.71070  -5.718 2.64e-08 ***
## as.factor(Dist)152 -10.64092    2.71070  -3.926 0.000108 ***
## as.factor(Dist)153  -5.17077    2.71070  -1.908 0.057418 .
## as.factor(Dist)154 -14.85268    2.71070  -5.479 9.14e-08 ***
## as.factor(Dist)155 -15.06931    2.71070  -5.559 6.05e-08 ***
## as.factor(Dist)156 -16.99013    2.71070  -6.268 1.29e-09 ***
## as.factor(Dist)157 -15.64221    2.71070  -5.771 1.99e-08 ***
## as.factor(Dist)158 -15.58782    2.71070  -5.750 2.22e-08 ***
## as.factor(Dist)159 -11.29862    2.71070  -4.168 4.04e-05 ***
## as.factor(Dist)160  -8.24810    2.71070  -3.043 0.002554 **
## as.factor(Dist)161 -15.91180    2.71070  -5.870 1.17e-08 ***
## as.factor(Dist)162 -13.64505    2.71070  -5.034 8.37e-07 ***
## as.factor(Dist)163 -15.56881    2.71070  -5.743 2.30e-08 ***
## as.factor(Dist)164 -16.73929    2.71070  -6.175 2.18e-09 ***
## as.factor(Dist)165 -15.06887    2.71070  -5.559 6.06e-08 ***
## as.factor(Dist)166 -15.09320    2.71070  -5.568 5.78e-08 ***
## as.factor(Dist)167 -11.45831    2.71070  -4.227 3.16e-05 ***
## as.factor(Dist)168  -6.01348    2.71070  -2.218 0.027285 *
## as.factor(Dist)169 -14.20403    2.71070  -5.240 3.06e-07 ***
## as.factor(Dist)170 -12.07622    2.71070  -4.455 1.19e-05 ***
## as.factor(Dist)171  -9.45263    2.71070  -3.487 0.000562 ***
## as.factor(Dist)172 -11.85653    2.71070  -4.374 1.69e-05 ***
## as.factor(Dist)173  -8.51711    2.71070  -3.142 0.001848 **
## as.factor(Dist)174 -12.24061    2.71070  -4.516 9.12e-06 ***
## as.factor(Dist)175 -11.91473    2.71070  -4.395 1.54e-05 ***
## as.factor(Dist)176  -7.23076    2.71070  -2.667 0.008063 **
## as.factor(Dist)177  -8.25191    2.71070  -3.044 0.002543 **
## as.factor(Dist)178  -8.65075    2.71070  -3.191 0.001568 **
## as.factor(Dist)179  -5.28817    2.71070  -1.951 0.052018 .
## as.factor(Dist)180 -10.10229    2.71070  -3.727 0.000232 ***
## as.factor(Dist)181  -6.32577    2.71070  -2.334 0.020284 *
## as.factor(Dist)182  -4.50620    2.71070  -1.662 0.097496 .
## as.factor(Dist)183   1.89058    2.71070   0.697 0.486067
```

```
## as.factor(Dist)184 -10.09514    2.71070  -3.724 0.000235 ***
## as.factor(Dist)185  -6.48111    2.71070  -2.391 0.017430 *
## as.factor(Dist)186  -0.37853    2.71070  -0.140 0.889037
## as.factor(Dist)187  -9.31723    2.71070  -3.437 0.000672 ***
## as.factor(Dist)188  -8.58395    2.71070  -3.167 0.001703 **
## as.factor(Dist)189 -16.10146    2.71070  -5.940 7.99e-09 ***
## as.factor(Dist)190 -15.97816    2.71070  -5.894 1.02e-08 ***
## as.factor(Dist)191 -12.05151    2.71070  -4.446 1.24e-05 ***
## as.factor(Dist)192 -15.35431    2.71070  -5.664 3.50e-08 ***
## as.factor(Dist)193  -9.88266    2.71070  -3.646 0.000315 ***
## as.factor(Dist)194 -16.37198    2.71070  -6.040 4.62e-09 ***
## as.factor(Dist)195 -15.71342    2.71070  -5.797 1.73e-08 ***
## as.factor(Dist)196 -16.92555    2.71070  -6.244 1.48e-09 ***
## as.factor(Dist)197 -12.08328    2.71070  -4.458 1.18e-05 ***
## as.factor(Dist)198 -11.15482    2.71070  -4.115 5.02e-05 ***
## as.factor(Dist)199 -10.05374    2.71070  -3.709 0.000249 ***
## as.factor(Dist)200 -12.08143    2.71070  -4.457 1.18e-05 ***
## as.factor(Dist)201 -13.22871    2.71070  -4.880 1.74e-06 ***
## as.factor(Dist)202 -15.32425    3.32179  -4.613 5.91e-06 ***
## as.factor(Dist)203  -9.59409    2.71070  -3.539 0.000466 ***
## as.factor(Dist)204 -12.41548    2.71070  -4.580 6.85e-06 ***
## as.factor(Dist)205  -4.64645    2.71070  -1.714 0.087554 .
## as.factor(Dist)206 -11.11193    2.71070  -4.099 5.36e-05 ***
## as.factor(Dist)207 -11.01992    2.71070  -4.065 6.15e-05 ***
## as.factor(Dist)208  -9.48565    2.71070  -3.499 0.000538 ***
## as.factor(Dist)209 -12.41994    2.71070  -4.582 6.80e-06 ***
## as.factor(Dist)210 -14.20715    2.71070  -5.241 3.04e-07 ***
## as.factor(Dist)211  -9.73693    2.71070  -3.592 0.000384 ***
## as.factor(Dist)212 -11.97454    2.71070  -4.418 1.40e-05 ***
## as.factor(Dist)213  -6.79702    2.71070  -2.507 0.012695 *
## as.factor(Dist)214  -8.38803    2.71070  -3.094 0.002160 **
## as.factor(Dist)215  -8.16244    2.71070  -3.011 0.002827 **
## as.factor(Dist)216 -11.29168    2.71070  -4.166 4.08e-05 ***
## as.factor(Dist)217  -0.66789    2.71070  -0.246 0.805550
## as.factor(Dist)218  -0.75876    2.71070  -0.280 0.779739
## as.factor(Dist)219   1.26631    2.71070   0.467 0.640734
## as.factor(Dist)220   2.28605    2.71070   0.843 0.399717
## as.factor(Dist)221  -2.27752    2.71070  -0.840 0.401476
## as.factor(Dist)222  -7.77801    2.71070  -2.869 0.004409 **
## as.factor(Dist)223  -4.87340    2.71070  -1.798 0.073221 .
## as.factor(Dist)224  -3.46333    2.71070  -1.278 0.202373
## as.factor(Dist)225  -9.96934    2.71070  -3.678 0.000279 ***
## as.factor(Dist)226  -8.85523    2.71070  -3.267 0.001216 **
## as.factor(Dist)227 -14.34462    2.71070  -5.292 2.36e-07 ***
## as.factor(Dist)228 -10.43190    2.71070  -3.848 0.000146 ***
## as.factor(Dist)229 -12.09244    2.71070  -4.461 1.16e-05 ***
## as.factor(Dist)230 -13.04289    2.71070  -4.812 2.39e-06 ***
## as.factor(Dist)231 -14.26280    2.71070  -5.262 2.74e-07 ***
## as.factor(Dist)232 -11.82767    2.71070  -4.363 1.77e-05 ***
## as.factor(Dist)233  -6.98056    2.71070  -2.575 0.010504 *
## as.factor(Dist)234 -15.10515    2.71070  -5.572 5.65e-08 ***
## as.factor(Dist)235 -15.02915    2.71070  -5.544 6.54e-08 ***
## as.factor(Dist)236  -7.07129    2.71070  -2.609 0.009551 **
## as.factor(Dist)237 -10.31173    2.71070  -3.804 0.000173 ***
```

```
## as.factor(Dist)238 -12.10096    2.71070  -4.464 1.14e-05 ***
## as.factor(Dist)239 -13.76694    2.71070  -5.079 6.73e-07 ***
## as.factor(Dist)240 -13.39531    2.71070  -4.942 1.30e-06 ***
## as.factor(Dist)241  -9.95996    2.71070  -3.674 0.000283 ***
## as.factor(Dist)242  -4.18207    2.71070  -1.543 0.123947
## as.factor(Dist)243  -7.58798    2.71070  -2.799 0.005458 **
## as.factor(Dist)244  -1.69980    2.71070  -0.627 0.531095
## as.factor(Dist)245  -9.38759    2.71070  -3.463 0.000613 ***
## as.factor(Dist)246  -8.73482    2.71070  -3.222 0.001413 **
## as.factor(Dist)247  -8.01090    2.71070  -2.955 0.003375 **
## as.factor(Dist)248 -11.03211    2.71070  -4.070 6.04e-05 ***
## as.factor(Dist)249 -10.73780    2.71070  -3.961 9.35e-05 ***
## as.factor(Dist)250 -10.89795    2.71070  -4.020 7.38e-05 ***
## as.factor(Dist)251  -2.51843    2.71070  -0.929 0.353610
## as.factor(Dist)252  -2.10780    2.71070  -0.778 0.437434
## as.factor(Dist)253  -9.47741    2.71070  -3.496 0.000544 ***
## as.factor(Dist)254 -12.55222    2.71070  -4.631 5.47e-06 ***
## as.factor(Dist)255  -9.25243    2.71070  -3.413 0.000731 ***
## as.factor(Dist)256  -7.01014    2.71070  -2.586 0.010184 *
## as.factor(Dist)257  -8.86212    2.71070  -3.269 0.001205 **
## as.factor(Dist)258  15.46422    2.71070   5.705 2.82e-08 ***
## as.factor(Dist)259   0.44997    2.71070   0.166 0.868271
## as.factor(Dist)260  -5.57896    2.71070  -2.058 0.040453 *
## as.factor(Dist)261  -2.51235    2.71070  -0.927 0.354771
## as.factor(Dist)262  -2.95058    2.71070  -1.088 0.277262
## as.factor(Dist)263  -7.34889    2.71070  -2.711 0.007098 **
## as.factor(Dist)264  -5.72171    2.71070  -2.111 0.035629 *
## as.factor(Dist)265  -2.07047    2.71070  -0.764 0.445586
## as.factor(Dist)266  -5.24399    2.71070  -1.935 0.053998 .
## as.factor(Dist)267  -8.88726    2.71070  -3.279 0.001168 **
## as.factor(Dist)268  -5.45127    2.71070  -2.011 0.045230 *
## as.factor(Dist)269  -7.93654    2.71070  -2.928 0.003678 **
## as.factor(Dist)270  -8.43361    2.71070  -3.111 0.002045 **
## as.factor(Dist)271   4.44844    2.71070   1.641 0.101845
## as.factor(Dist)272  -5.27878    2.71070  -1.947 0.052434 .
## as.factor(Dist)273  -7.36414    2.71070  -2.717 0.006982 **
## as.factor(Dist)274   4.09622    2.71070   1.511 0.131822
## as.factor(Dist)275  -1.49729    2.71070  -0.552 0.581115
## as.factor(Dist)276 -10.36207    2.71070  -3.823 0.000161 ***
## as.factor(Dist)277  -7.09930    2.71070  -2.619 0.009273 **
## as.factor(Dist)278  -7.82794    2.71070  -2.888 0.004166 **
## as.factor(Dist)279  -8.13407    2.71070  -3.001 0.002923 **
## as.factor(Dist)280  -9.09076    2.71070  -3.354 0.000901 ***
## as.factor(Dist)281   7.92152    2.71070   2.922 0.003742 **
## as.factor(Dist)282  -1.54325    2.71070  -0.569 0.569573
## as.factor(Dist)283  -6.73801    2.71070  -2.486 0.013481 *
## as.factor(Dist)284  -6.01002    2.71070  -2.217 0.027373 *
## as.factor(Dist)285  -8.39827    2.71070  -3.098 0.002134 **
## as.factor(Dist)286  -8.38538    2.71070  -3.093 0.002167 **
## as.factor(Dist)287  -3.62909    2.71070  -1.339 0.181663
## as.factor(Dist)288  -5.72502    2.71070  -2.112 0.035523 *
## as.factor(Dist)289  -3.84297    2.71070  -1.418 0.157329
## as.factor(Dist)290   3.04042    2.71070   1.122 0.262925
## as.factor(Dist)291  -3.97669    2.71070  -1.467 0.143428
```

```
## as.factor(Dist)292  -5.79027    2.71070  -2.136 0.033493 *
## as.factor(Dist)293  -3.85158    2.71070  -1.421 0.156404
## as.factor(Dist)294   1.30528    2.71070   0.482 0.630496
## as.factor(Dist)295  -4.49923    2.71070  -1.660 0.098013 .
## as.factor(Dist)296   2.06245    2.71070   0.761 0.447349
## as.factor(Dist)297  -0.80595    2.71070  -0.297 0.766431
## as.factor(Dist)298  -1.14410    2.71070  -0.422 0.673283
## as.factor(Dist)299        NA         NA      NA       NA
## BTW21:Saar          -4.89512    1.92975  -2.537 0.011705 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.711 on 296 degrees of freedom
##   (1 Beobachtung als fehlend gelöscht)
## Multiple R-squared:  0.9091, Adjusted R-squared:  0.817
## F-statistic: 9.867 on 300 and 296 DF,  p-value: < 2.2e-16
```

Since the output is too long to print due to the large number of dummy variables, we can extract the last rows of the output:

```
coefficients(summary(fixed.lm.out))[290:301,]
```

```
##                       Estimate Std. Error    t value   Pr(>|t|)
## as.factor(Dist)288 -5.7250227   2.710698 -2.1120106 0.03552315
## as.factor(Dist)289 -3.8429682   2.710698 -1.4177044 0.15732890
## as.factor(Dist)290  3.0404233   2.710698  1.1216386 0.26292522
## as.factor(Dist)291 -3.9766944   2.710698 -1.4670371 0.14342761
## as.factor(Dist)292 -5.7902664   2.710698 -2.1360796 0.03349314
## as.factor(Dist)293 -3.8515804   2.710698 -1.4208815 0.15640380
## as.factor(Dist)294  1.3052820   2.710698  0.4815299 0.63049575
## as.factor(Dist)295 -4.4992274   2.710698 -1.6598041 0.09801288
## as.factor(Dist)296  2.0624491   2.710698  0.7608554 0.44734912
## as.factor(Dist)297 -0.8059452   2.710698 -0.2973202 0.76643061
## as.factor(Dist)298 -1.1440950   2.710698 -0.4220666 0.67328279
## BTW21:Saar         -4.8951161   1.929748 -2.5366609 0.01170522
```

We can see here the almost same effect size, however, with a significant result. The effect size is slightly different since one district was omitted from the analysis since the Greens had no district candidate (District Bitburg).

While the effect size is almost same, its standard error is much smaller than in the previous analysis. This is because the error variance is reduced by taking the district heterogeneity into account.

This result is equivalent for the estimate of the interested effect if we regress the first difference of the dependent variable on the dummy variable for the districts in Saarland.

```
gru.smd.diff <- gru.smd - gru.smd.17

summary(lm(gru.smd.diff ~ saar.dummy.21))
```

```
##
## Call:
## lm(formula = gru.smd.diff ~ saar.dummy.21)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3570 -2.8098 -0.2997  1.7328 14.8823
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.8550     0.2236  26.188   <2e-16 ***
## saar.dummy.21  -4.8951     1.9297  -2.537   0.0117 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.834 on 296 degrees of freedom
##   (1 Beobachtung als fehlend gelöscht)
## Multiple R-squared:  0.02128,    Adjusted R-squared:  0.01797
## F-statistic: 6.435 on 1 and 296 DF,  p-value: 0.01171
```

To be precise, the above data does not constitute the panel data since before the 2021 elections 17 districts were redistricted due to their population changes or municipality area changes. We can therefore construct the new data without these districts and repeate the analysis:

```
gru.pooled.wo.redist <- gru.pooled[!(gru.pooled$Dist %in%
                                       c(60,61,135:137,228,230,233,234,242,243,253,254,
                                         190,192,195,196)),]


fixed.lm.out <- lm(gru.smd.pooled ~ BTW21 * Saar + as.factor(Dist),
           data=gru.pooled.wo.redist)

coefficients(summary(fixed.lm.out))[280:284,]
```

```
##                       Estimate Std. Error    t value   Pr(>|t|)
## as.factor(Dist)295 -4.4992274   2.712497 -1.6587033 0.09829933
## as.factor(Dist)296  2.0624491   2.712497  0.7603508 0.44768700
## as.factor(Dist)297 -0.8059452   2.712497 -0.2971230 0.76659369
## as.factor(Dist)298 -1.1440950   2.712497 -0.4217867 0.67350553
## BTW21:Saar         -5.0109767   1.931824 -2.5939098 0.00999023
```

The result is almost identical with the previous one.

Instrumental variable and 2SLS

It has been often said that lower voter turnout benefits AfD since the party is successful in mobilizing their core supporters so that it obtains certain stable amount of votes at different elections.

To test the hypothesis, we can start with a naive analysis based on the following simple regression model:

```
summary(lm( afd.pr ~ turnout ))
```

```
##
## Call:
## lm(formula = afd.pr ~ turnout)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.723 -3.549 -1.212  1.622 21.635
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 56.68004    6.32339   8.964  < 2e-16 ***
## turnout     -0.60883    0.08332  -7.307 2.53e-12 ***
```

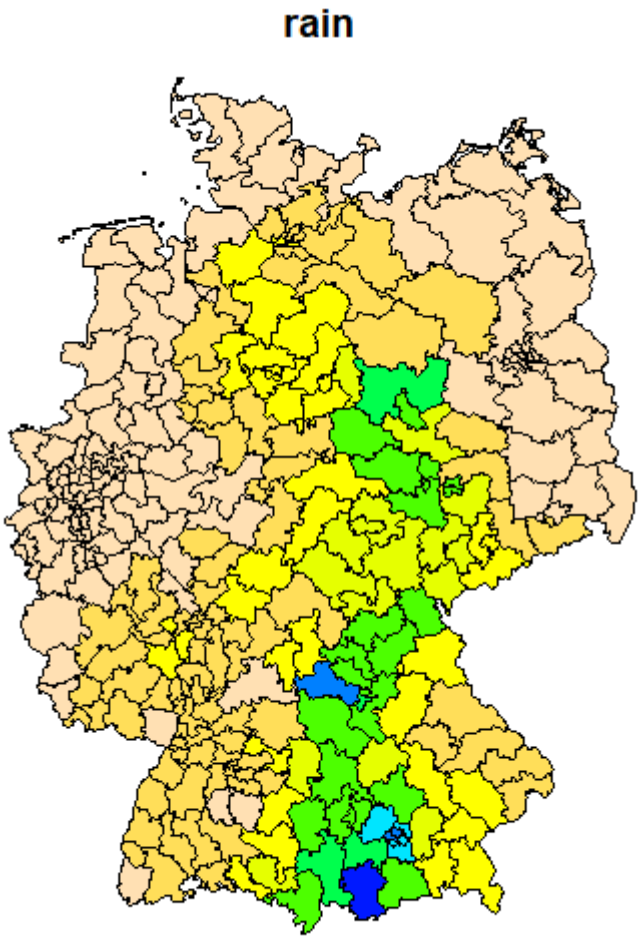
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.424 on 297 degrees of freedom
## Multiple R-squared:  0.1524, Adjusted R-squared:  0.1495
## F-statistic:  53.4 on 1 and 297 DF,  p-value: 2.529e-12
```

While the result supports the hypothesis above, we can also question whether the turnout variable is endogenous since both the vote share of AfD and voter turnout were generated simultaneously by individual voter decisions.

For this reason, we employ the instrumental variable estimator. As the instrumental variable, we use rainfall at the election day. While we can reasonably assume that weather including rainfall is exogenous to the vote decisions, it can affect the turnout level.

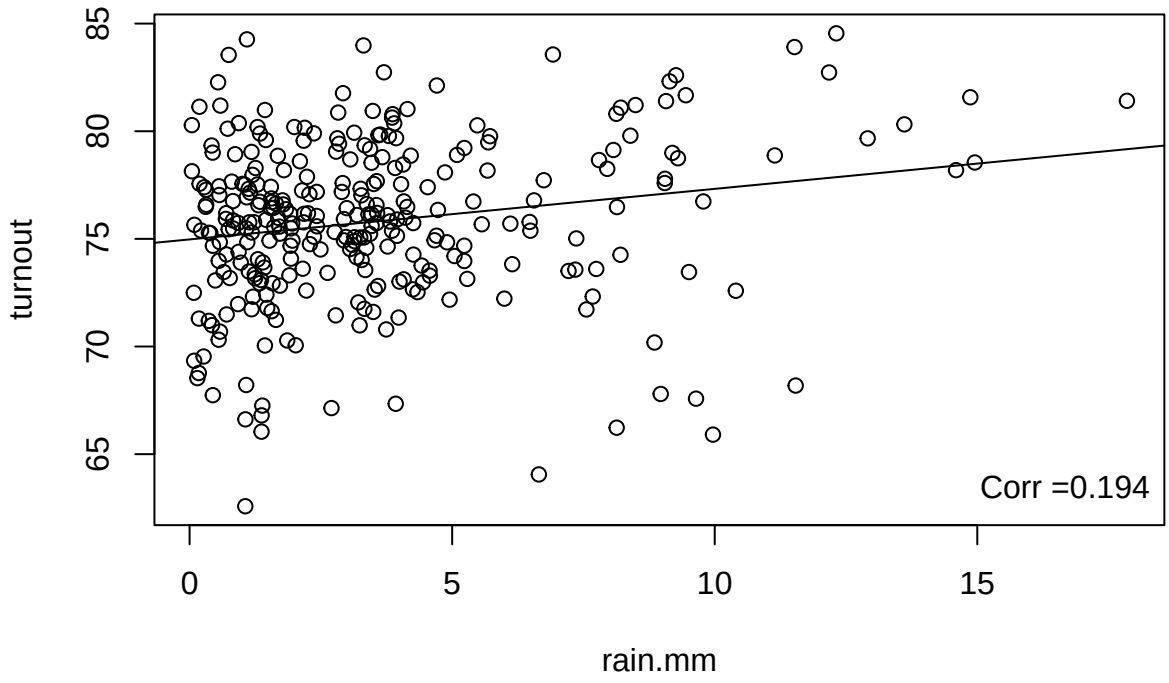
The figure below displays different rainfall in the electoral districts. Accordingly, the rainiest district was Weilheim in Bavaria with 17.8mm. The driest district was Berlin-Steglitz-Zelendorf with 0.04mm. The rainfall of individual districts was reconstructed by using the historical data of 83 weather stations (source: https://www.wetterkontor.de/). Based on the coordinates of the weather stations (source: https://www.dwd.de/) and individual districts (source: https://www.bundeswahlleiter.de), weather in each districts was estimated by interpolating all 83 stations (weighted by inverse squared distance on the coordinate system).

```
load("data/BTW21weather.RData")

rain.mm <- dist.weather$rain

knitr::include_graphics("image/BTW21_Rain.png")
```



While we can reasonably assume the exogeneity of rainfall, we should check its relationship with our independent variable (turnout).

```
plot(turnout ~ rain.mm)
lm.out <- lm(turnout ~ rain.mm )
abline(lm.out)
legend("bottomright",paste0("Corr =" , round(cor(turnout ,rain.mm),3)),bty="n")
```




```
summary(lm.out)
```

```
##
## Call:
## lm(formula = turnout ~ rain.mm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6424  -2.1058   0.2827   2.3404   9.0365
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 74.97406    0.32353 231.737  < 2e-16 ***
## rain.mm      0.23484    0.06874   3.416 0.000723 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.705 on 297 degrees of freedom
## Multiple R-squared:  0.03781,    Adjusted R-squared:  0.03457
## F-statistic: 11.67 on 1 and 297 DF,  p-value: 0.0007232
```

Interestingly, the above results show a positive effect of rainfall on turnout level. That is, the more rainfall, the higher turnout. While its effect is very small, it is significant at 5% level and F-statistics of the model also signals that the model is superior to the model without the independent variable.

Now, we can compute the estimate by using IV estimator:

```
    y <- afd.pr
    x1 <- turnout
    z <- rain.mm

    slope.est <- cov(y,z)/cov(x1,z)
    intercept.est <- mean(y) - slope.est*mean(x1)

    resid <- y - (intercept.est + slope.est*x1)
```

```
## Warning in slope.est * x1: Recycling-Array mit Länge 1 in Array-Vektor-Rechnungen ist veraltet.
##  Bitte stattdessen c() oder as.vector() nutzen.
```

```
## Warning in intercept.est + slope.est * x1: Recycling-Array mit Länge 1 in Array-Vektor-Rechnungen is
##  Bitte stattdessen c() oder as.vector() nutzen.
```

```
    sigma2.hat <- sum(resid^2)/(length(y)-2)
    SSTx <- sum((x1 - mean(x1))^2)
    R2.xz <- summary(lm(x1 ~ z))$r.squared

    slope.se <- sqrt(sigma2.hat/(SSTx*R2.xz))


    slope.est
```

```
##           [,1]
## [1,] 0.9450308
```

```
    slope.se
```

```
## [1] 0.6313369
```

```
    intercept.est
```

```
##           [,1]
## [1,] -61.10576
```

Now the point estimate of the effect of turnout on the AfD vote share becomes positive, while its standard error is quite large so that we cannot reject the null hypothesis of no effect.

The same result can be also obtained by using a ready made package and its function:

```
library(ivreg)
```

```
## Warning: Paket 'ivreg' wurde unter R Version 4.1.3 erstellt
```

```
summary(iv.reg.out <- ivreg( afd.pr ~ turnout | rain.mm))
```

```
##
## Call:
## ivreg(formula = afd.pr ~ turnout | rain.mm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.646  -4.936  -1.875   2.299  23.727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -61.1058    47.8589  -1.277    0.203
## turnout       0.9450     0.6313   1.497    0.135
```

```
##
## Diagnostic tests:
##                  df1 df2 statistic  p-value
## Weak instruments   1 297     11.67 0.000723 ***
## Wu-Hausman         1 296     14.28 0.000190 ***
## Sargan             0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.992 on 297 degrees of freedom
## Multiple R-Squared: -0.8403, Adjusted R-squared: -0.8465
## Wald test: 2.241 on 1 and 297 DF,  p-value: 0.1355
```

While this package uses the 2sls estimator, the estimates are identical with the results above. That is, the instrumental variable estimator for a single instrumental variable is a special case of 2SLS.

This output also includes some diagnostics. The first one about "weak instruments" test against the null hypothesis that the instrumental variable is not correlated with the endogenous variable. This can be here rejected, which was also obvious from the above regression result.

The second one "Wu-Hausman" test against the null hypothesis that the OLS estimator is consistent, therefore the results are similar to that of the IV estimator. Also this null-hypothesis is rejected, which is also obvious from the very different point estimate here from the first OLS regression.

If you like to consider possible heteroskedasticity, here we can also obtain the robust standard errors:

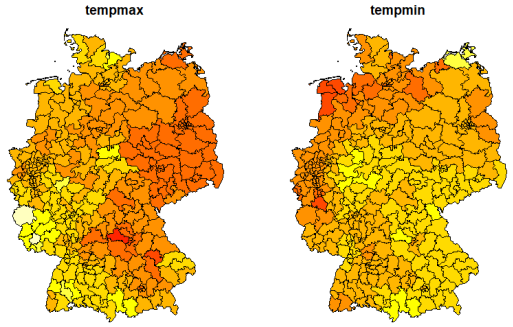
```
summary(iv.reg.out , vcov =sandwich::sandwich)
```

```
##
## Call:
## ivreg(formula = afd.pr ~ turnout | rain.mm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.646  -4.936  -1.875   2.299  23.727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -61.1058    57.0637  -1.071    0.285
## turnout       0.9450     0.7541   1.253    0.211
##
## Diagnostic tests:
##                  df1 df2 statistic  p-value
## Weak instruments   1 297      8.55 0.003721 **
## Wu-Hausman         1 296     13.98 0.000221 ***
## Sargan             0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.992 on 297 degrees of freedom
## Multiple R-Squared: -0.8403, Adjusted R-squared: -0.8465
## Wald test:  1.57 on 1 and 297 DF,  p-value: 0.2111
```

Accordingly, we have larger standard errors, while conclusion about the interested hypothesis remains same, that is no clear effect of turnout on the AfD vote share.

Now, you may wish to use further weather data as instrumental variables. For example, the maximum and minimum temperature at the election day:

```
temp.max <- dist.weather$tempmax
temp.min <- dist.weather$tempmin

knitr::include_graphics(c("image/BTW21_tempmax.png","image/BTW21_tempmin.png"))
```



We can check the correlation of the relevant variables:

```
cor.out <- cor(cbind(afd.pr , turnout , rain.mm , temp.min, temp.max))
colnames(cor.out) <- rownames(cor.out) <- c("afd.pr","turnout","rain.mm","temp.min","temp.max")
print(cor.out,digits=3)
```

```
##          afd.pr turnout rain.mm temp.min temp.max
## afd.pr    1.000  -0.390   0.118   -0.308    0.383
## turnout  -0.390   1.000   0.194   -0.131   -0.135
## rain.mm   0.118   0.194   1.000   -0.471    0.216
## temp.min -0.308  -0.131  -0.471    1.000   -0.128
## temp.max  0.383  -0.135   0.216   -0.128    1.000
```

```
summary(lm(turnout ~ rain.mm + temp.min + temp.max))
```

```
##
## Call:
## lm(formula = turnout ~ rain.mm + temp.min + temp.max)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4795  -2.1095   0.2243   2.2713   9.0574
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 90.78238    5.21550  17.406  < 2e-16 ***
## rain.mm      0.25162    0.07796   3.228  0.00139 **
## temp.min    -0.22265    0.25040  -0.889  0.37464
## temp.max    -0.57340    0.17552  -3.267  0.00122 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.649 on 295 degrees of freedom
## Multiple R-squared:  0.07329,    Adjusted R-squared:  0.06387
## F-statistic: 7.777 on 3 and 295 DF,  p-value: 5.151e-05
```

Besides to rainfall, maximum temperature seems to have an impact on turnout.

```
summary(iv.reg.out.2 <- ivreg( afd.pr ~ turnout | rain.mm +  temp.max))
```

```
##
## Call:
## ivreg(formula = afd.pr ~ turnout | rain.mm + temp.max)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0727  -3.7636  -0.5473   2.2011  21.7719
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  83.4702    24.4444   3.415 0.000727 ***
## turnout      -0.9623     0.3224  -2.984 0.003079 **
##
## Diagnostic tests:
##                  df1 df2 statistic  p-value
## Weak instruments   2 296    11.278 1.90e-05 ***
## Wu-Hausman         1 296     1.373    0.242
## Sargan             1  NA    40.361 2.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.586 on 297 degrees of freedom
## Multiple R-Squared: 0.101,   Adjusted R-squared: 0.09801
## Wald test: 8.905 on 1 and 297 DF,  p-value: 0.003079
```

Interestingly, the result became again reversed and the effect of turnout on the AfD vote becomes negative and significant. Correspondingly, the second diagnostic (Wu-Hausman) does not reject the null hypothesis of the consistency of the OLS estimator.

In this model, however, we also have the third diagnostic (Sargan), which tests overidentification Restrictions. Here, the null hypothesis is that all instrumental variables are uncorrelated with the error. According to the output above, this null hypothesis is rejected at 5% level, which means at least one of the instrumental variable is not exogenous. Therefore, we should better not take this model and rely on the model with one instrumental variable.

Alternatively, we can now exclude the rainfall variable from the model:

```
summary(iv.reg.out.3 <- ivreg( afd.pr ~ turnout |  temp.max))
```

```
##
## Call:
## ivreg(formula = afd.pr ~ turnout | temp.max)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -56.7366 -10.1328   0.9476  11.6366  33.3672
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  345.275    132.172   2.612  0.00945 **
## turnout       -4.416      1.744  -2.533  0.01183 *
##
## Diagnostic tests:
##                  df1 df2 statistic  p-value
```

```
## Weak instruments   1 297     5.548   0.0192 *
## Wu-Hausman         1 296    44.749 1.12e-10 ***
## Sargan             0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.37 on 297 degrees of freedom
## Multiple R-Squared: -5.807,  Adjusted R-squared: -5.83
## Wald test: 6.415 on 1 and 297 DF,  p-value: 0.01183
```

Here, we obtained a very large negative effect of turnout on the AfD vote share. Here, however, due to the first diagnostic, we seem to have a weaker instrument than the rainfall variable. For this reason, it is better to take the rainfall variable as instrument.

Logit and Probit Models for Binary Response

```
# vote results in SMD tier

all.smd <- result21[,grep("Erststimmen",names(result21))]
all.smd <- all.smd[,-c(1:4)]

max.party <- apply(all.smd,1,which.max)

spd.win <- ifelse(max.party==2,1,0)
union.win <- ifelse(max.party==1|max.party==7,1,0)
gru.win <- ifelse(max.party==6,1,0)
```

We model the dependent variable whether CDU/CSU candidate won the district by using the variables: GDP, and abi.

We first estimate the linear probability model:

```
lpm.out <- lm(union.win ~ gdp + abi )
summary(lpm.out)
```

```
##
## Call:
## lm(formula = union.win ~ gdp + abi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8364 -0.3976 -0.1008  0.3849  0.9071
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.209e+00  1.323e-01   9.140  < 2e-16 ***
## gdp          6.589e-06  1.937e-06   3.402 0.000762 ***
## abi         -2.848e-02  3.786e-03  -7.521 6.52e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4593 on 296 degrees of freedom
## Multiple R-squared:  0.1632, Adjusted R-squared:  0.1575
## F-statistic: 28.86 on 2 and 296 DF,  p-value: 3.537e-12
```

Now, we estimate the logit model:

```
logit.out <- glm(union.win ~ gdp + abi ,family=binomial(link="logit"))
summary(logit.out)
```

```
##
## Call:
## glm(formula = union.win ~ gdp + abi, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8655  -0.9872  -0.5392   0.9617   2.0081
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.284e+00  6.749e-01   4.866 1.14e-06 ***
## gdp          2.957e-05  9.365e-06   3.158  0.00159 **
## abi         -1.309e-01  2.022e-02  -6.472 9.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 413.94  on 298  degrees of freedom
## Residual deviance: 361.83  on 296  degrees of freedom
## AIC: 367.83
##
## Number of Fisher Scoring iterations: 4
```

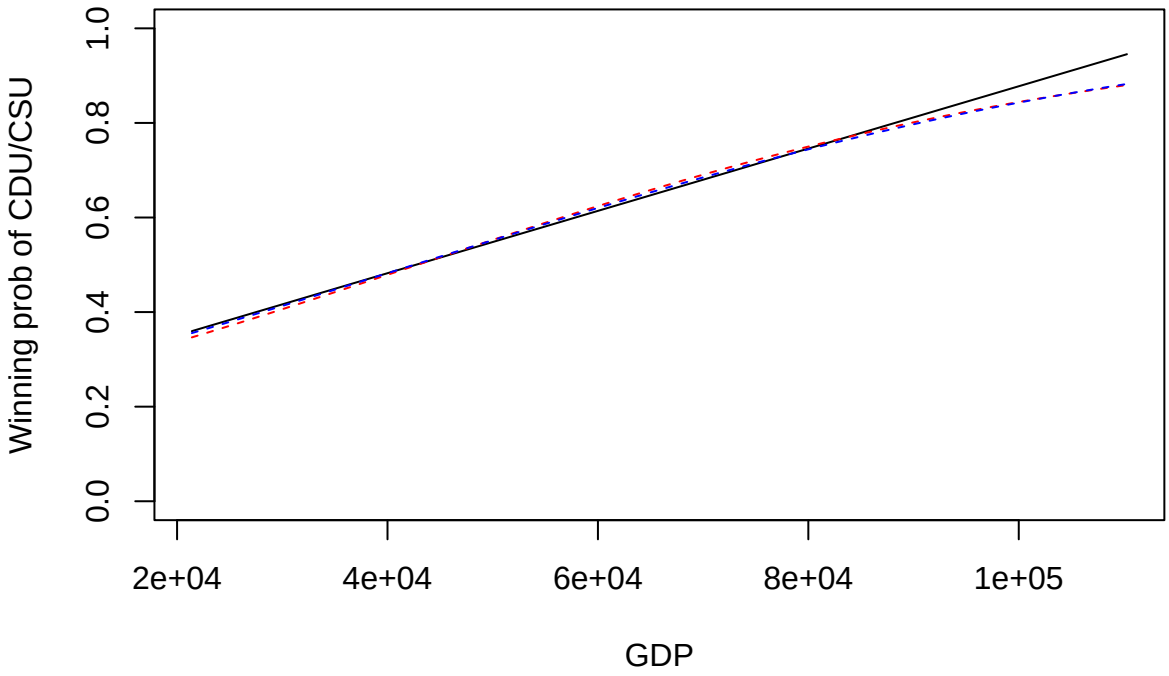
and the probit model:

```
probit.out <- glm(union.win ~ gdp + abi ,family=binomial(link="probit"))
summary(probit.out)
```

```
##
## Call:
## glm(formula = union.win ~ gdp + abi, family = binomial(link = "probit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8761  -0.9998  -0.5275   0.9618   2.0212
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.012e+00  3.980e-01   5.055 4.29e-07 ***
## gdp          1.752e-05  5.621e-06   3.117  0.00183 **
## abi         -7.929e-02  1.172e-02  -6.768 1.31e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 413.94  on 298  degrees of freedom
## Residual deviance: 362.12  on 296  degrees of freedom
## AIC: 368.12
##
## Number of Fisher Scoring iterations: 4
```

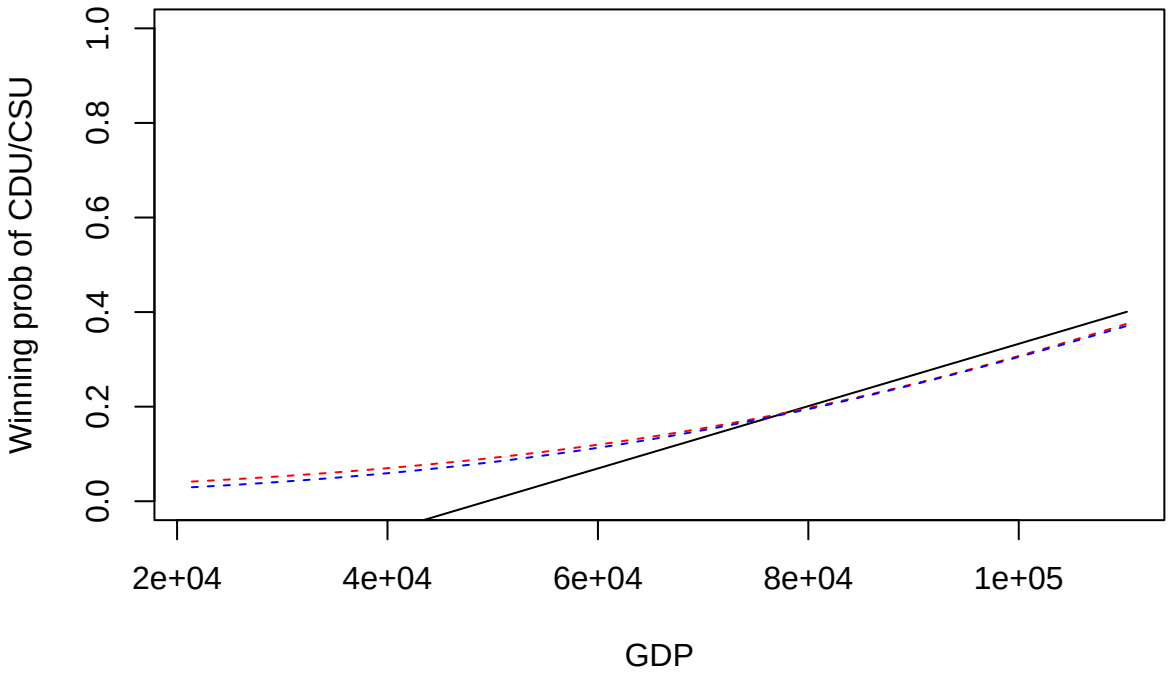
We compare the predicted values based on three different models. Below, we predict the winning probability of the CDU/CSU candidates for different levels of GDP, while keeping the other variables (abi) being the average:

```
# average values for abi
abi.bar <- mean(abi)

# generate different values for the GDP variable between its minimum and maximum value
gdp.values <- seq(min(gdp),max(gdp),length=100)

# predict for different catholic shares based on different model
## LPM
predict.lpm <- coefficients(lpm.out)[1] +
               coefficients(lpm.out)[2] *gdp.values +
               coefficients(lpm.out)[3] *abi.bar

predict.logit <- coefficients(logit.out)[1] +
               coefficients(logit.out)[2] *gdp.values +
               coefficients(logit.out)[3] *abi.bar
predict.logit <- exp(predict.logit)/(1+exp(predict.logit))

predict.probit <- coefficients(probit.out)[1] +
               coefficients(probit.out)[2] *gdp.values +
               coefficients(probit.out)[3] *abi.bar
predict.probit <- pnorm(predict.probit)

plot(gdp.values,predict.lpm,ylim=c(0,1),type="l",
     xlab="GDP",ylab="Winning prob of CDU/CSU")
lines(gdp.values, predict.logit, col="red",lty=2)
lines(gdp.values, predict.probit, col="blue",lty=2)
```



Now, we change the average value for the abi (20) variable to the maximum value (53.9 in Hamburg Mitte):

```
# average values for GDP, but the maximum value for abi
abi.max <- max(abi)


# predict for different catholic shares based on different model
## LPM
predict.lpm <- coefficients(lpm.out)[1] +
               coefficients(lpm.out)[2] *gdp.values +
               coefficients(lpm.out)[3] *abi.max

predict.logit <- coefficients(logit.out)[1] +
               coefficients(logit.out)[2] *gdp.values +
               coefficients(logit.out)[3] *abi.max
predict.logit <- exp(predict.logit)/(1+exp(predict.logit))

predict.probit <- coefficients(probit.out)[1] +
               coefficients(probit.out)[2] *gdp.values +
               coefficients(probit.out)[3] *abi.max
predict.probit <- pnorm(predict.probit)

plot(gdp.values,predict.lpm,ylim=c(0,1),type="l",
     xlab="GDP",ylab="Winning prob of CDU/CSU")
lines(gdp.values, predict.logit, col="red",lty=2)
lines(gdp.values, predict.probit, col="blue",lty=2)
```



It is apparent that the partial effect of the same level of GDP differ depending on the share of voters with University entrance diploma (Abi).

17.1 Logit and Probit Models for Binary Response: LR-test

We focus here only on the probit model:

```
probit.out <- glm(union.win ~ gdp + abi ,family=binomial(link="probit"))
summary(probit.out)
```

```
##
## Call:
## glm(formula = union.win ~ gdp + abi, family = binomial(link = "probit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8761  -0.9998  -0.5275   0.9618   2.0212
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.012e+00  3.980e-01   5.055 4.29e-07 ***
## gdp          1.752e-05  5.621e-06   3.117  0.00183 **
## abi         -7.929e-02  1.172e-02  -6.768 1.31e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 413.94  on 298  degrees of freedom
## Residual deviance: 362.12  on 296  degrees of freedom
## AIC: 368.12
##
## Number of Fisher Scoring iterations: 4
```

Concerning this model, we test whether all the included independent variables are relevant as a whole to explain the winning probability of CDU/CSU candidates. To do this, we can estimate the restricted model without any independent variables:

```
probit.out.res <- glm(union.win ~ 1,family=binomial(link="probit"))
summary(probit.out.res)
```

```
##
## Call:
## glm(formula = union.win ~ 1, family = binomial(link = "probit"))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.141  -1.141  -1.141   1.215   1.215
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.05452    0.07252  -0.752    0.452
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 413.94  on 298  degrees of freedom
## Residual deviance: 413.94  on 298  degrees of freedom
## AIC: 415.94
##
## Number of Fisher Scoring iterations: 3
```

We can obtain the log likelihood of both models:

```
logLik(probit.out)
```

```
## 'log Lik.' -181.058 (df=3)
```

```
logLik(probit.out.res)
```

```
## 'log Lik.' -206.9683 (df=1)
```

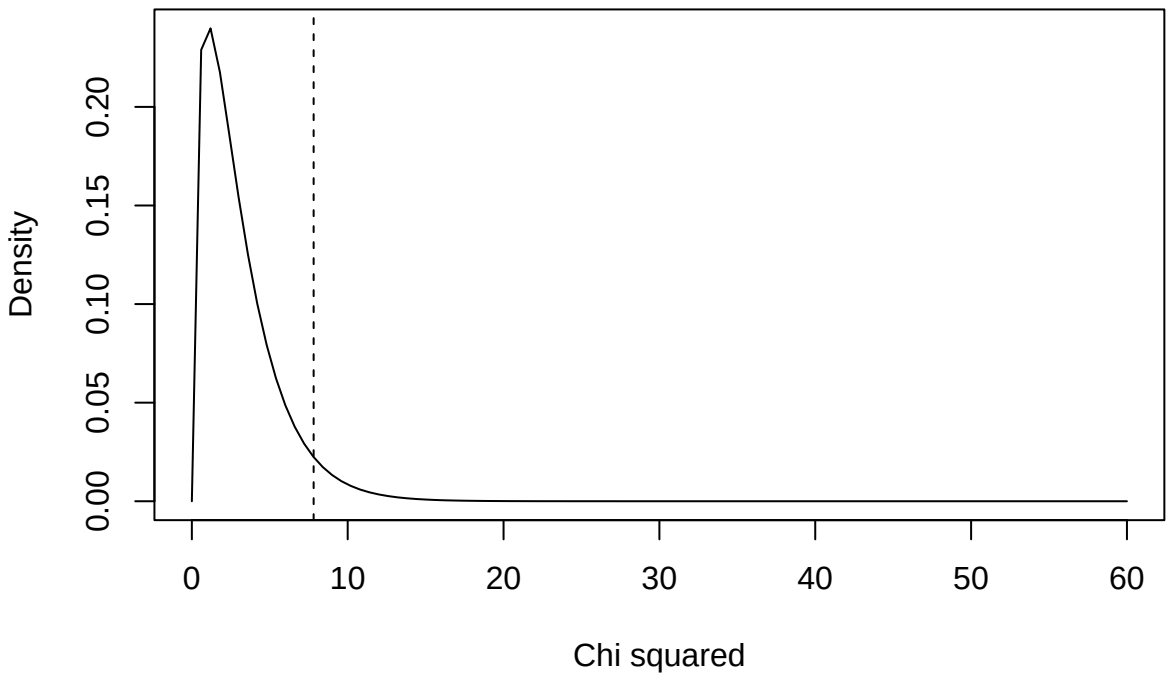
... and the likelihood ration, as well:

```
LR <- as.numeric(2 * (logLik(probit.out) - logLik(probit.out.res)))
LR
```

```
## [1] 51.8207
```

This value follows the chi-square distribution with 2 df:

```
chisq.func <- function(x) dchisq(x,df=3)

curve(chisq.func,0,60,ylab="Density",xlab="Chi squared")
abline(v=qchisq(0.95,df=3),lty=2)
```



At the significance level of 5%, we can reject the Null-hypothesis, "the restricted and unrestricted models do not differ".

```
knitr::knit_exit()
```