

# Aachen Report

Bhaskar Kamble

5-11-2020

## Get the data from co2online

```
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/energy_proportions_by_et.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/appendLinearTrend.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/area_proportions_by_et.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/find_proportions.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/getTotalConsumption.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/getAbsoluteEnergyShares.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/getCO2Emissions.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/getRowSums.R")
source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/getCumSums.R")

source("/home/kbhaskar/Github_Repos/co2emissions/Cologne/getSpecificConsumptionCologne.R")

source("/home/kbhaskar/Github_Repos/co2emissions/RheinNeckarKreis/getRegionData.R")
source("/home/kbhaskar/Github_Repos/visualization-project2-smurfs/cleanData.R")
gtype <- "SFH"
region <- "StXdtregion Aachen"
region_data_sfH <- getRegionData(gtype,region)

gtype <- "MFH"
region <- "StXdtregion Aachen"
region_data_mfH <- getRegionData(gtype,region)
#2019 is an outlier - remove it
region_data_sfH <- region_data_sfH[region_data_sfH$abrechnungsjahr > 2000 , ]
region_data_sfH <- region_data_sfH[region_data_sfH$abrechnungsjahr <= 2018 , ]
```

## Data Size

```
dim(region_data_sfH)
```

```
## [1] 15479    10
```

```
dim(region_data_mfH)
```

```
## [1] 2833     10
```

## Area per building for each year

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
by_year <- group_by(region_data_sf, abrechnungsjahr)
```

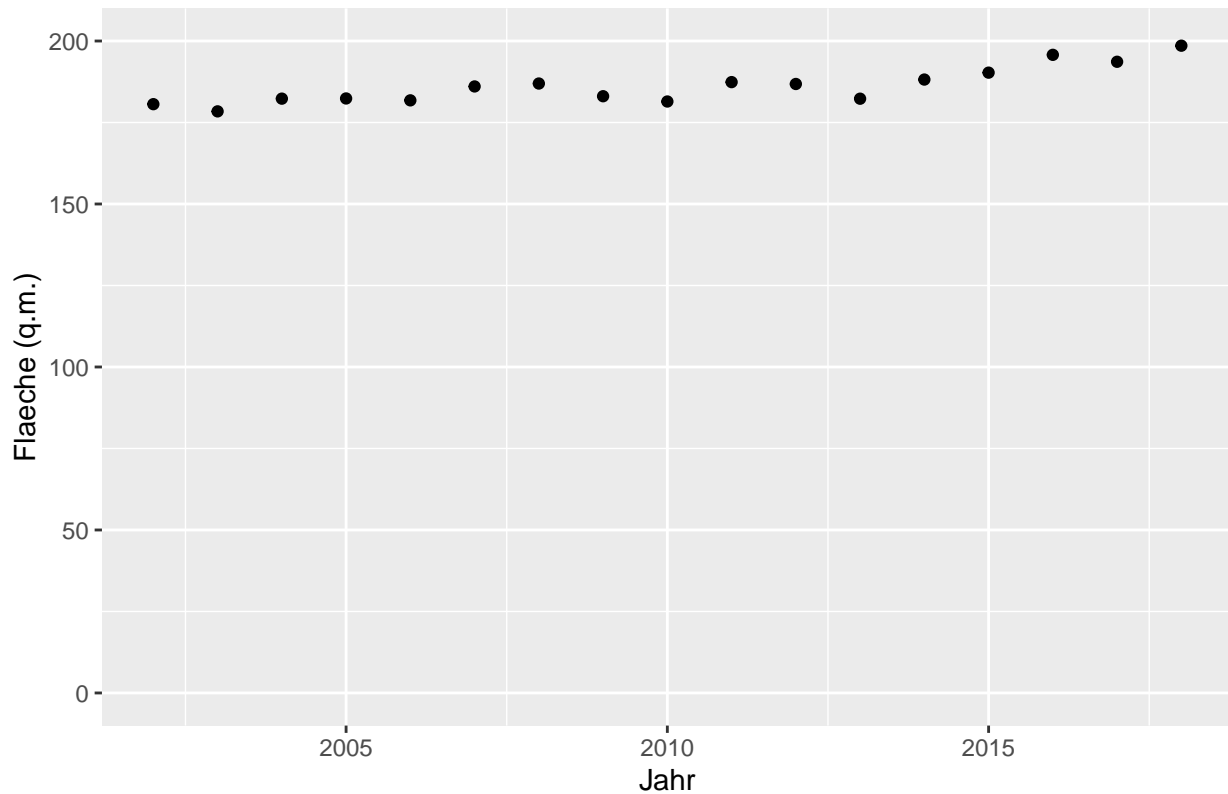
```
avgAreaSFH <- as.data.frame(summarize(by_year, mean(gebaeude_nutzflaeche)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
names(avgAreaSFH) <- c("abrechnungsjahr", "meanArea")
```

```
ggplot()+geom_point(data=avgAreaSFH, aes(x=abrechnungsjahr, y=meanArea))+scale_y_continuous(lim=c(0, 200))
```

## Fläche pro Gebäude, 1–2FH, Köln



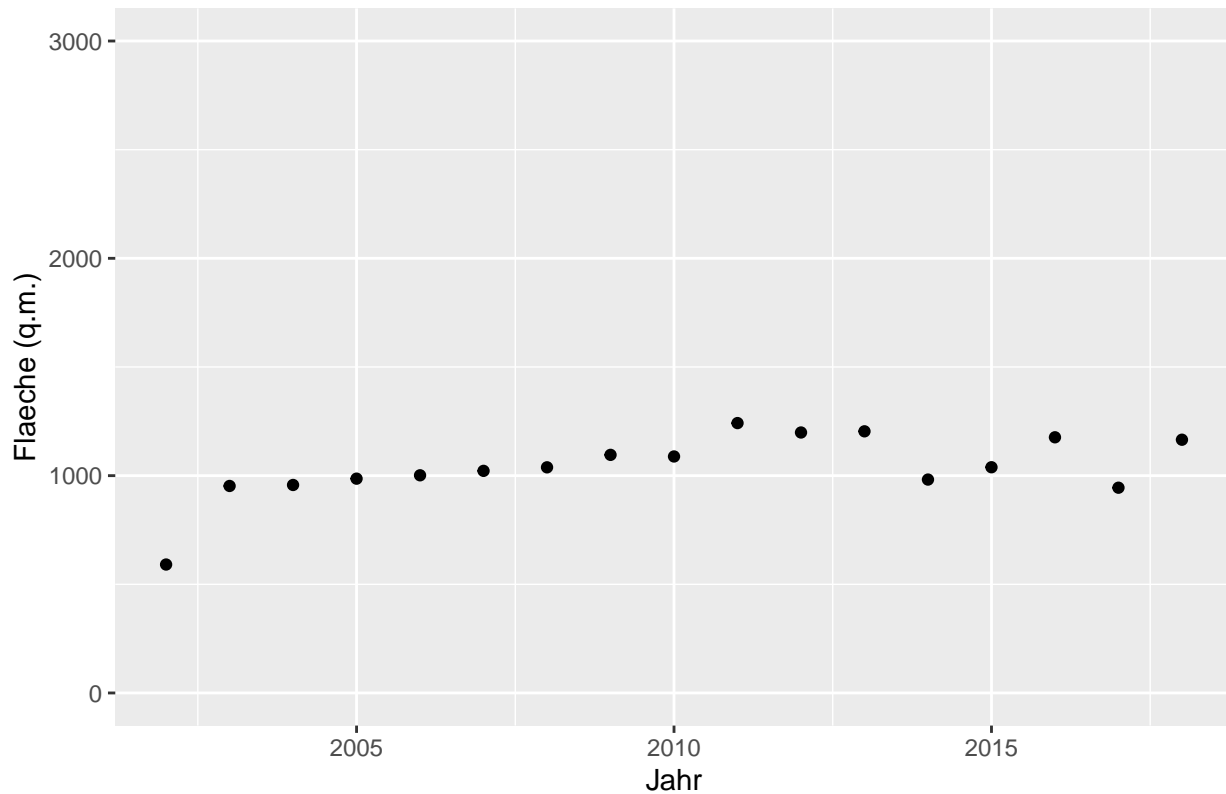
```
by_year <- group_by(region_data_mfh, abrechnungsjahr)
avgAreaMFH <- as.data.frame(summarize(by_year, mean(gebaeude_nutzflaeche)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
avgAreaMFH = avgAreaMFH[avgAreaMFH$abrechnungsjahr > 1950,]
names(avgAreaMFH) <- c("abrechnungsjahr", "meanArea")
```

```
ggplot() + geom_point(data=avgAreaMFH, aes(x=abrechnungsjahr, y=meanArea)) + scale_y_continuous(lim=c(0, 3000))
```

## Fläche pro Gebäude, MFH, Köln



```
et_list <- c("erdgas","waerme","fluessiggas","heizoel","holzpellets","strom")
region_data_mfh <- region_data_mfh[region_data_mfh$abrechnungsjahr>1950 , ]
region_data_mfh <- region_data_mfh[region_data_mfh$abrechnungsjahr<2019 , ]
energy_prop_table_mfh <- energy_proportions_by_et(region_data_mfh,et_list)
```

```
## `summarise()` regrouping output by 'abrechnungsjahr' (override with `.groups` argument)
```

```
## Loading required package: reshape2
```

```
region_data_sf <- region_data_sf[region_data_sf$abrechnungsjahr<2019 , ]
energy_prop_table_sf <- energy_proportions_by_et(region_data_sf,et_list)
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

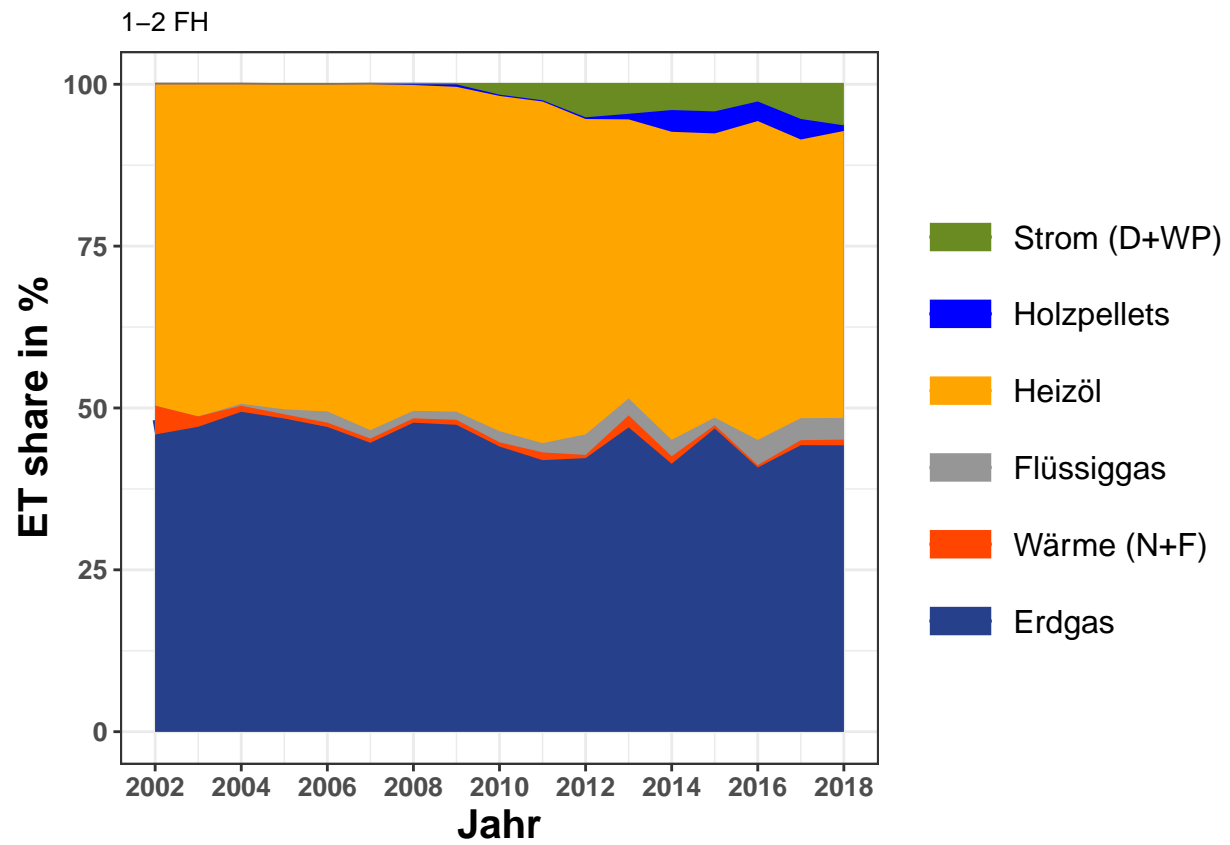
```
## `summarise()` regrouping output by 'abrechnungsjahr' (override with `.groups` argument)
```

```
## Loading required package: reshape2
```

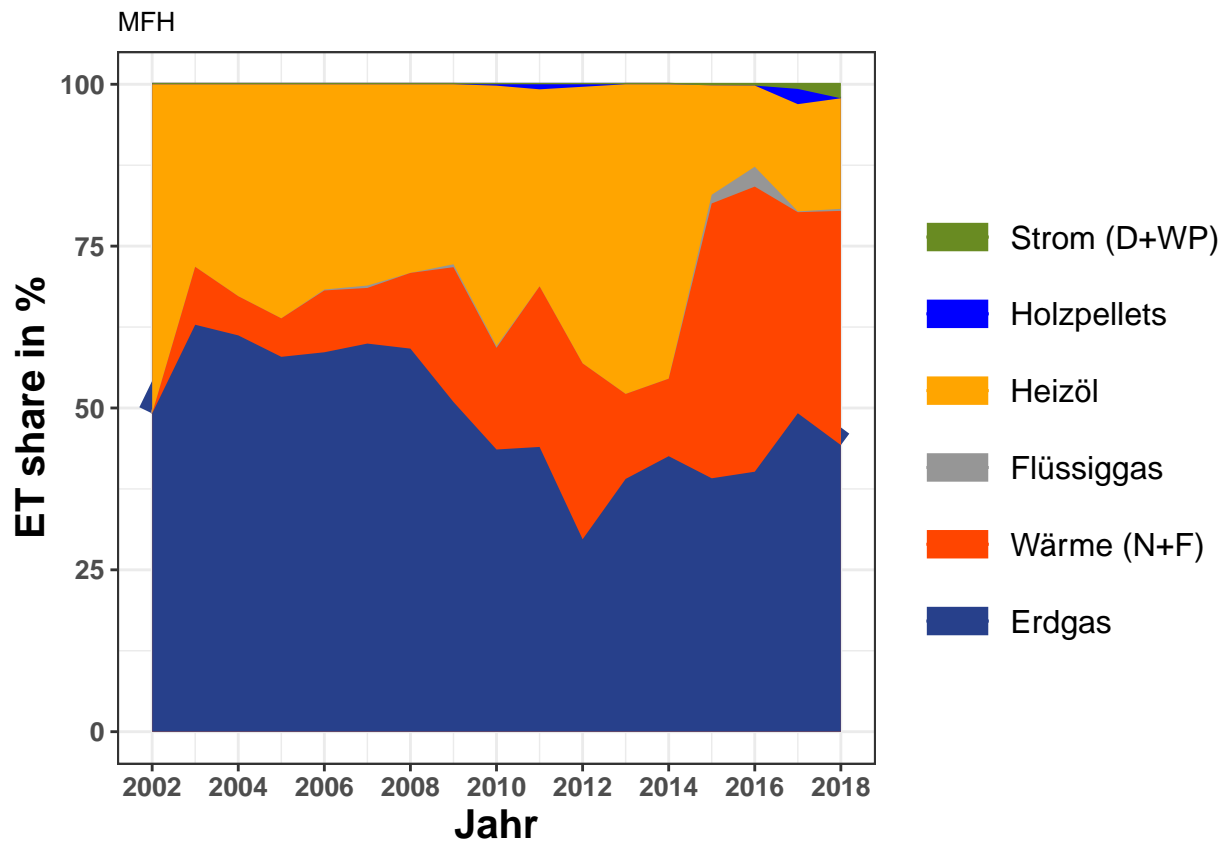
```
ET_shares_sfh <- getCumSums(energy_prop_table_sfh , "abrechnungsjahr")
ET_shares_mfh <- getCumSums(energy_prop_table_mfh , "abrechnungsjahr")
```

```
col_list <- c("royalblue4","orangered1","gray59","orange","blue","olivedrab4")
cols <- c(
  "erdgas"      = "royalblue4",
  "waerme"      = "orangered1",
  "fluessiggas" = "gray59",
  "heizael"     = "orange",
  "holzpellets" = "blue",
  "strom"       = "olivedrab4"
)
plot_title <- NULL
order_legend <- rev(c("erdgas","waerme","fluessiggas","heizael","holzpellets","strom"))
order_labels <- rev(c("Erdgas","Wärme (N+F)","Flüssiggas","Heizöl","Holzpellets","Strom (D+WP)"))
et_list <- c("erdgas","waerme","fluessiggas","heizael","holzpellets","strom")
plot_byET <- function(obj,xlabel,ylabel,plottitle,kt_to_mt=FALSE) {
  if (kt_to_mt) {
    source("/home/kbhaskar/Github_Repos/co2emissions/Berlin/BezirkAnalysis/convert_kilo_to_megaton.R")
    obj <- convert_kilo_to_megaton(obj , "abrechnungsjahr")
  }
  ggplot()+geom_line(data=obj,aes(x=abrechnungsjahr,y=get(et_list[1]),color=et_list[1]),size=5)
  +geom_line(data=obj,aes(x=abrechnungsjahr,y=get(et_list[2]),color=et_list[2]))
  +geom_line(data=obj,aes(x=abrechnungsjahr,y=get(et_list[3]),color=et_list[3]))
  +geom_line(data=obj,aes(x=abrechnungsjahr,y=get(et_list[4]),color=et_list[4]))
  +geom_line(data=obj,aes(x=abrechnungsjahr,y=get(et_list[5]),color=et_list[5]))
  +geom_line(data=obj,aes(x=abrechnungsjahr,y=get(et_list[6]),color=et_list[6]))
  +scale_color_manual(labels=order_labels,name=" ",values=cols,breaks=order_legend)
  +geom_ribbon(data=obj,aes(x=abrechnungsjahr,ymin=0,ymax=get(et_list[6])),fill=col_list[6])
  +geom_ribbon(data=obj,aes(x=abrechnungsjahr,ymin=0,ymax=get(et_list[5])),fill=col_list[5])
  +geom_ribbon(data=obj,aes(x=abrechnungsjahr,ymin=0,ymax=get(et_list[4])),fill=col_list[4])
  +geom_ribbon(data=obj,aes(x=abrechnungsjahr,ymin=0,ymax=get(et_list[3])),fill=col_list[3])
  +geom_ribbon(data=obj,aes(x=abrechnungsjahr,ymin=0,ymax=get(et_list[2])),fill=col_list[2])
  +geom_ribbon(data=obj,aes(x=abrechnungsjahr,ymin=0,ymax=get(et_list[1])),fill=col_list[1])+theme_bw()
  plot.title=element_text(size=10),
  axis.title.x=element_text(size=15, face="bold"),
  axis.title.y = element_text(size=15, face="bold"),
  legend.text = element_text(size=12),
  axis.text.x=element_text(size=10,face="bold"),
  axis.text.y=element_text(size=10,face="bold"),
  legend.key.size=unit(2, "lines")
  +scale_x_continuous(breaks=seq(2002,2018,2))
}
```

```
plot_byET(obj = ET_shares_sfh,
  xlabel = "Jahr",
  ylabel = "ET share in %",
  plottitle = "1-2 FH")
```



```
plot_byET(obj = ET_shares_mfh,  
  xlabel = "Jahr",  
  ylabel = "ET share in %",  
  plottitle = "MFH")
```



```
spz_verbrauch_mean_mfh <- getSpecificConsumptionCologne(region_data_mfh , FALSE)
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
spz_verbrauch_mean_sfH <- getSpecificConsumptionCologne(region_data_sfH , FALSE)
```

```
## Loading required package: dplyr
```

```
##
```

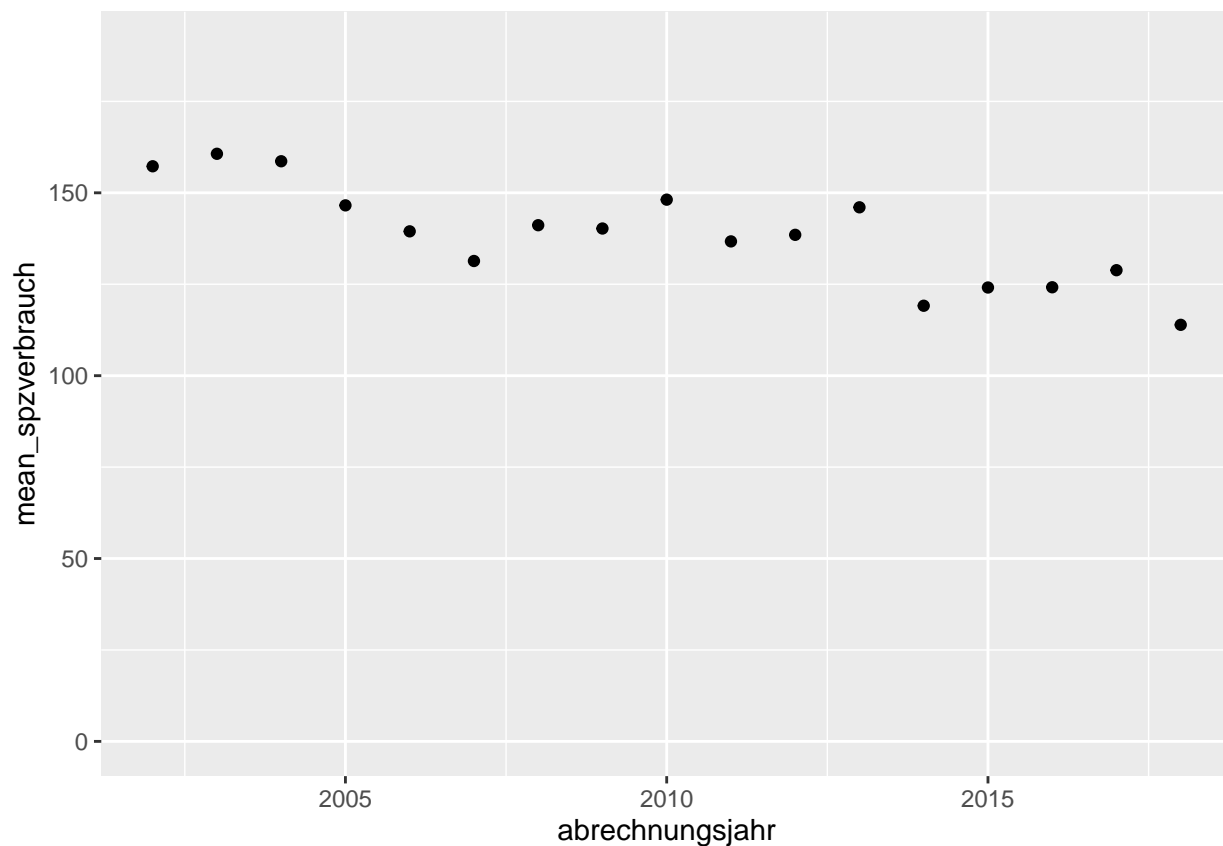
```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

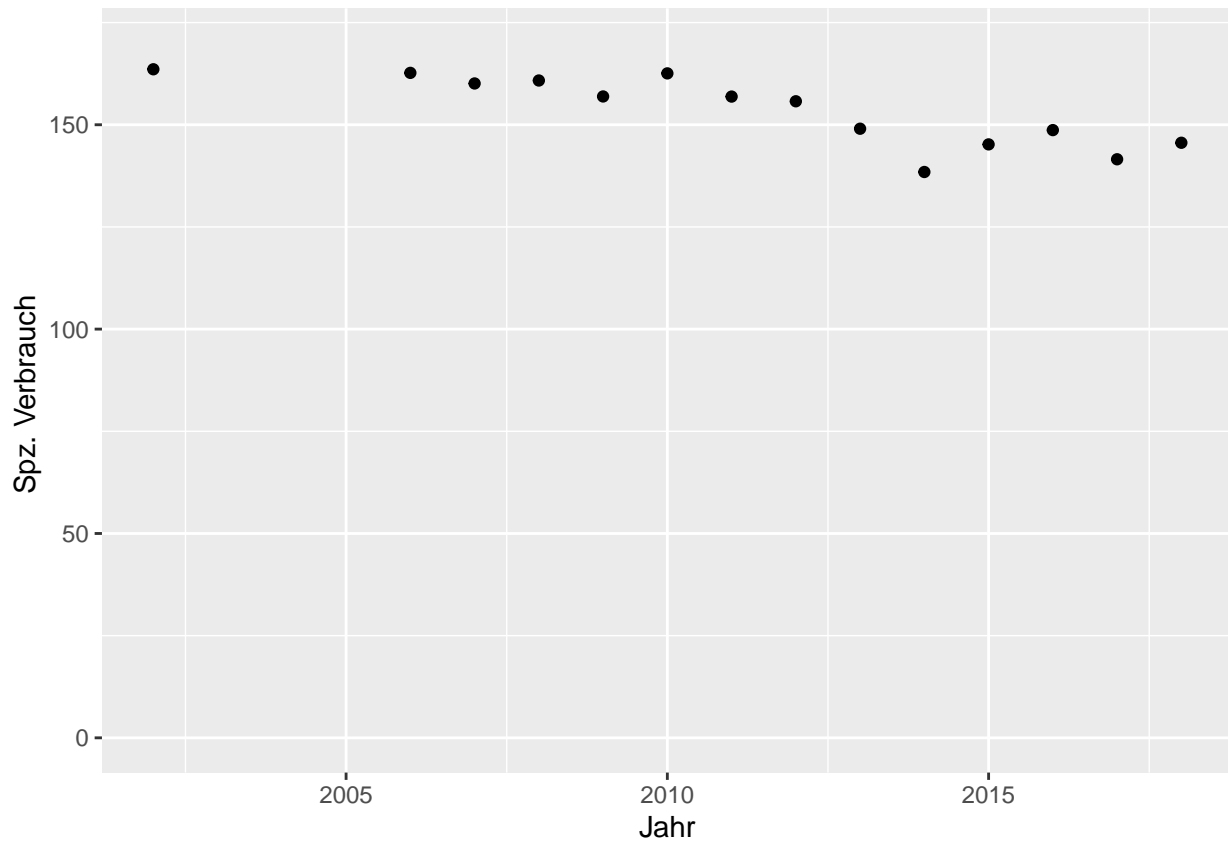
```
ggplot(spz_verbrauch_mean_mfh , aes(x=abrechnungsjahr,y=mean_spzverbrauch))+geom_point()+ylim(c(0,190))
```



```
ggplot(spz_verbrauch_mean_sfh , aes(x=abrechnungsjahr,y=mean_spzverbrauch))+geom_point()+ylim(c(0,170))
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```





## Gebaeude Baujahr

```
mfhbaujahr <- region_data_mfh$gebaeude_baujahr
sfhbaujahr <- region_data_sf$gebaeude_baujahr
```

```
as.data.frame(t(t(table(cut(mfhbaujahr,breaks=10)))))
```

```
##           Var1 Var2 Freq
## 1  (1.8e+03,1.82e+03]    A    2
## 2  (1.82e+03,1.84e+03]    A    0
## 3  (1.84e+03,1.87e+03]    A   22
## 4  (1.87e+03,1.89e+03]    A   18
## 5  (1.89e+03,1.91e+03]    A  235
## 6  (1.91e+03,1.93e+03]    A  176
## 7  (1.93e+03,1.95e+03]    A  234
## 8  (1.95e+03,1.98e+03]    A 1313
## 9   (1.98e+03,2e+03]     A  701
## 10 (2e+03,2.02e+03]     A  131
```

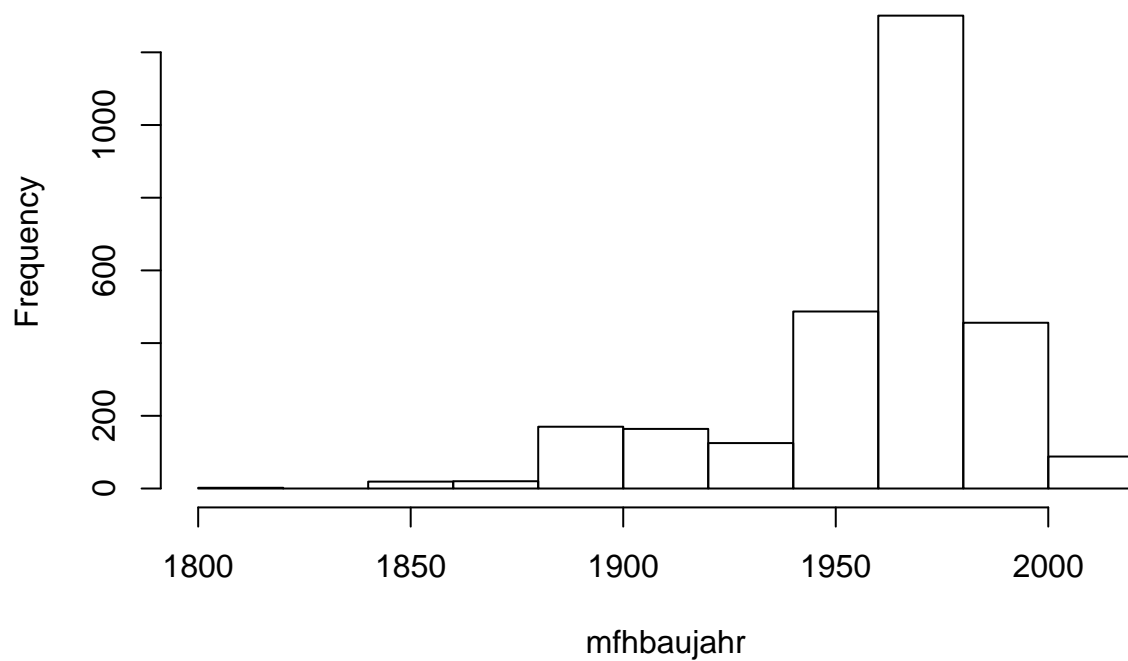
```
as.data.frame(t(t(table(cut(sfhbaujahr,breaks=seq(from=min(sfhbaujahr,na.rm=T),to=max(sfhbaujahr,na.rm=T),
length.out=10))))))
```

```
##           Var1 Var2 Freq
```

```
## 1 (1.6e+03,1.65e+03] A 0
## 2 (1.65e+03,1.69e+03] A 4
## 3 (1.69e+03,1.74e+03] A 5
## 4 (1.74e+03,1.79e+03] A 7
## 5 (1.79e+03,1.83e+03] A 9
## 6 (1.83e+03,1.88e+03] A 186
## 7 (1.88e+03,1.93e+03] A 1274
## 8 (1.93e+03,1.97e+03] A 6942
## 9 (1.97e+03,2.02e+03] A 7036
```

```
hist(mfhbaujahr)
```

**Histogram of mfhbaujahr**



```
hist(sfhbaujahr)
```

**Histogram of sfhbaujahr**

