ED_Project-Amazon_Fires

Jakub Bryl & Kamil Kryczka

30 10 2019

Data loading

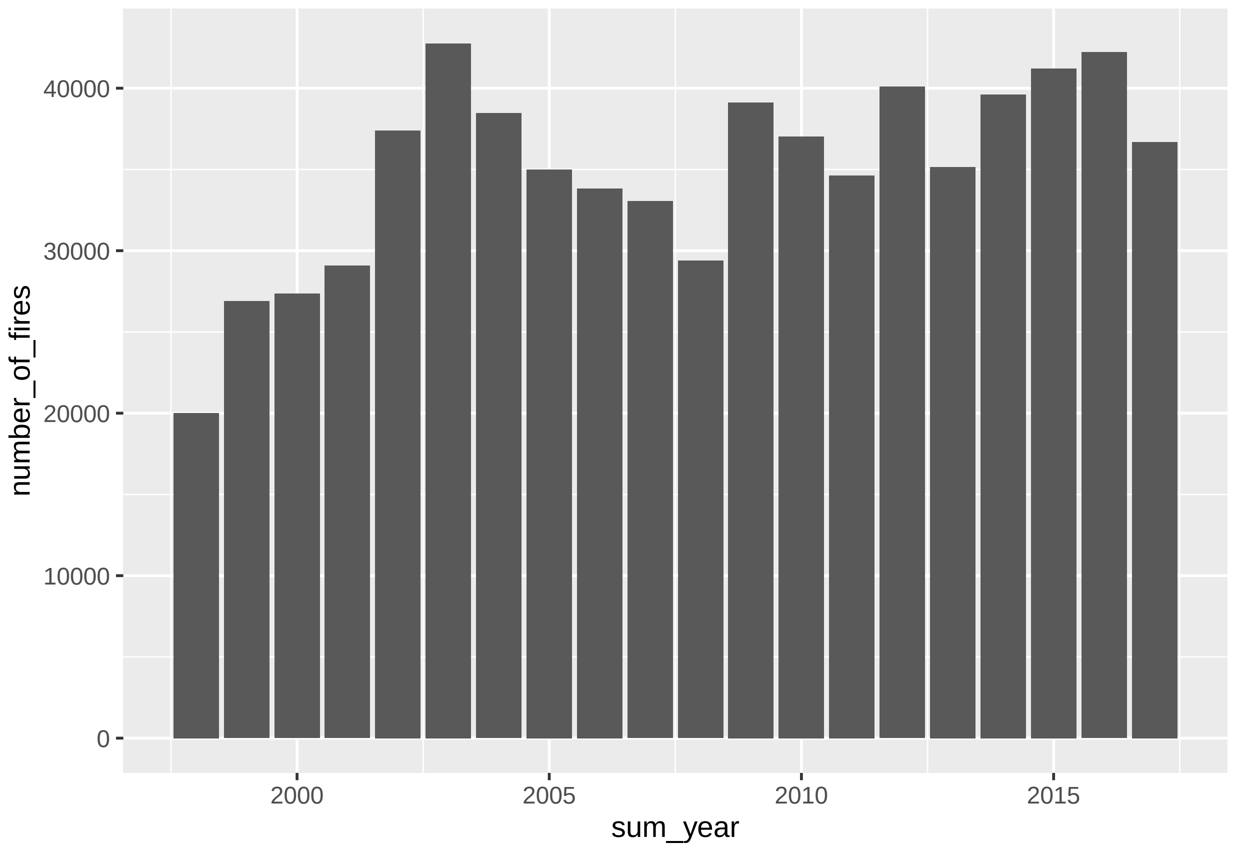
```r
data <- read.csv("amazon.csv") %>% mutate(date=as.Date(date))
```

Dividing Data Set

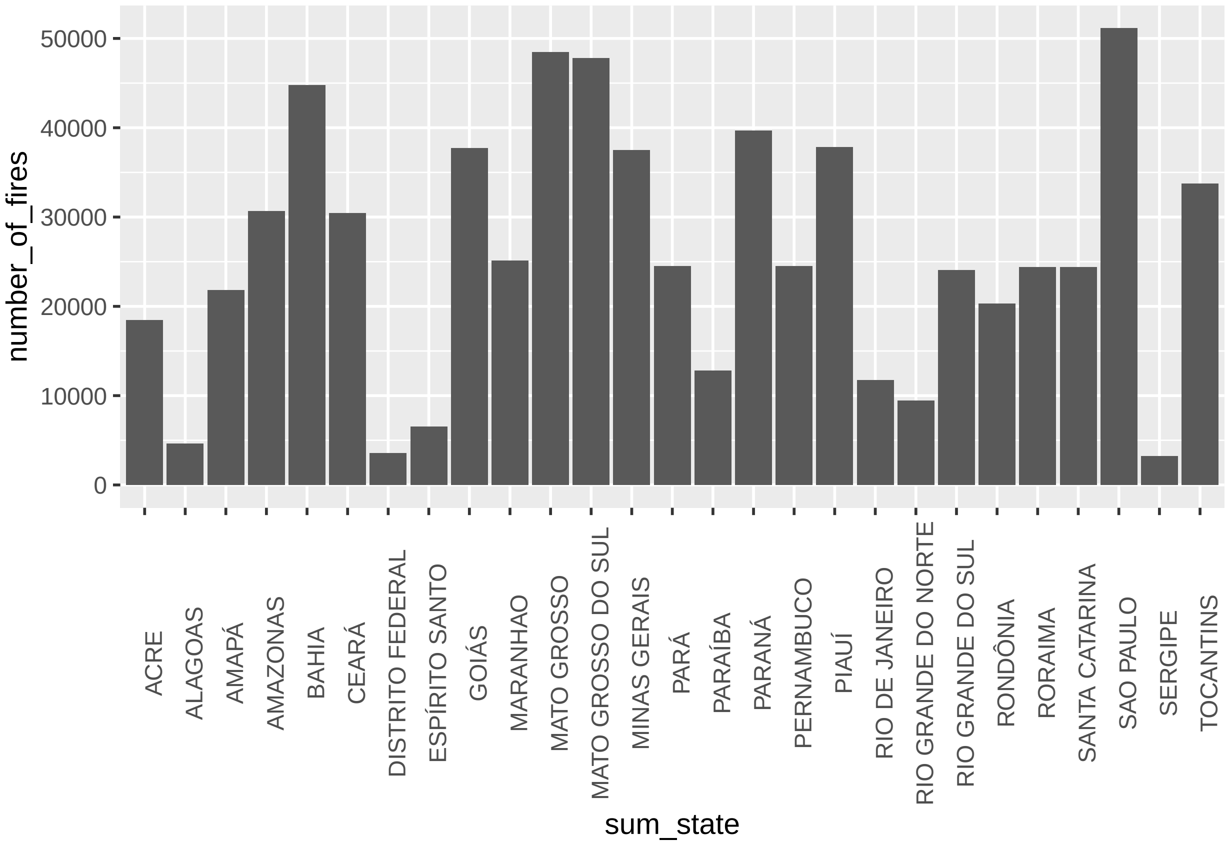
```r
data_year <- aggregate(list(number_of_fires=data$number), by=list(sum_year=data$year), FUN=sum)

data_state <- aggregate(list(number_of_fires=data$number), by=list(sum_state=data$state), FUN=sum)

data_month <- aggregate(list(number_of_fires=data$number), by=list(sum_month=data$month), FUN=sum)
data_month$sum_month <- factor(data_month$sum_month, levels = month.name)
```

Bar Charts

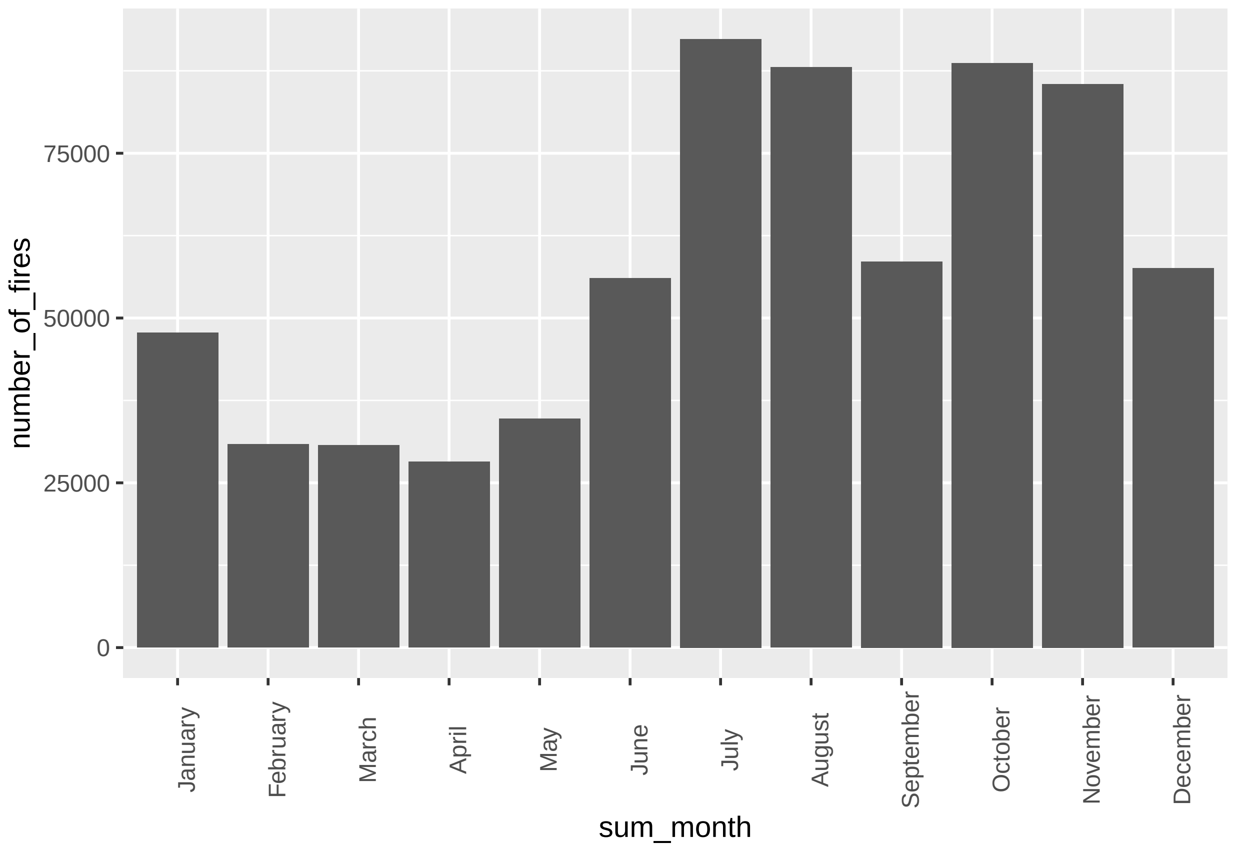
```r
ggplot(data = data_year, aes(x=sum_year, y=number_of_fires)) + geom_bar(stat = "identity")
```



```
ggplot(data = data_state, aes(x=sum_state, y=number_of_fires)) + geom_bar(stat = "identity") + theme(ax
```



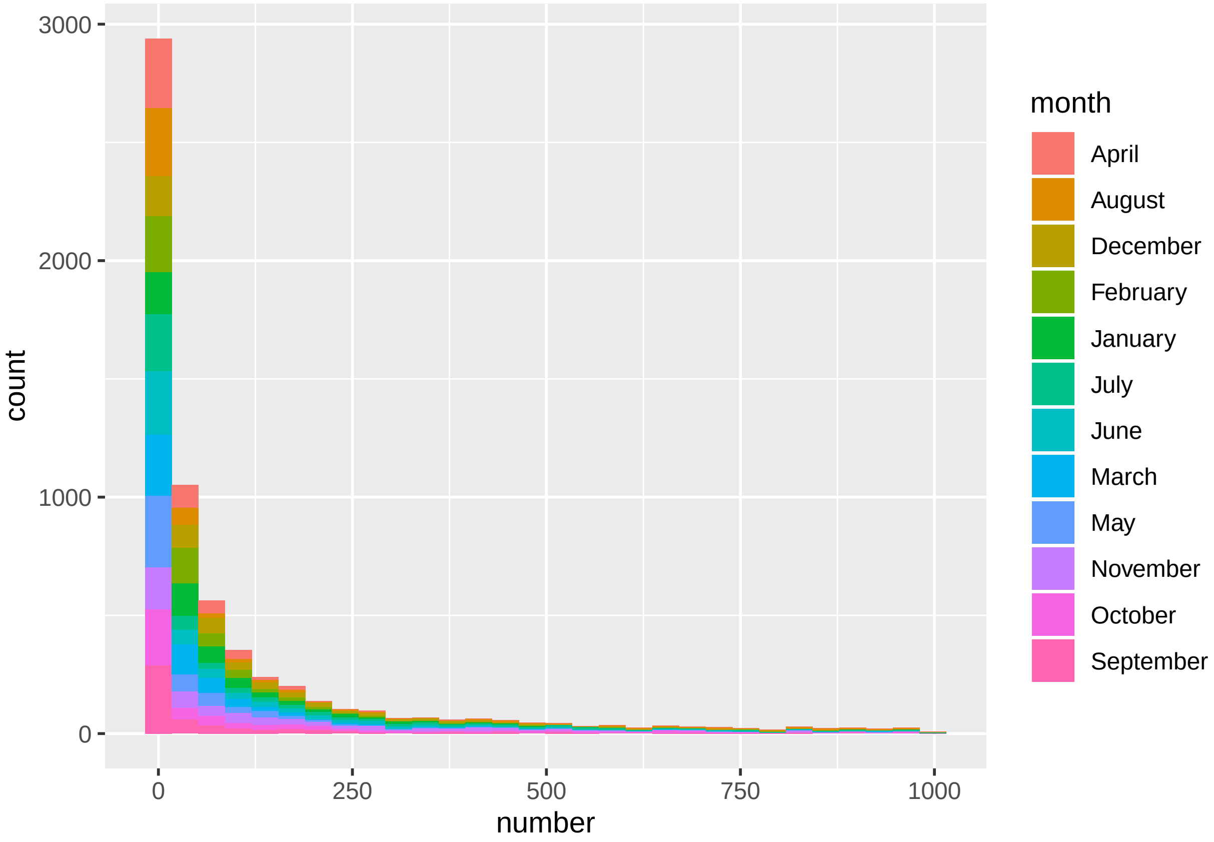
```
data_month %>% arrange(sum_month) %>% ggplot(data = . , aes(x=sum_month, y=number_of_fires)) + geom_bar
```



Histogram

```
ggplot(data = data, aes(x=number, fill=`month`)) + geom_histogram()
```

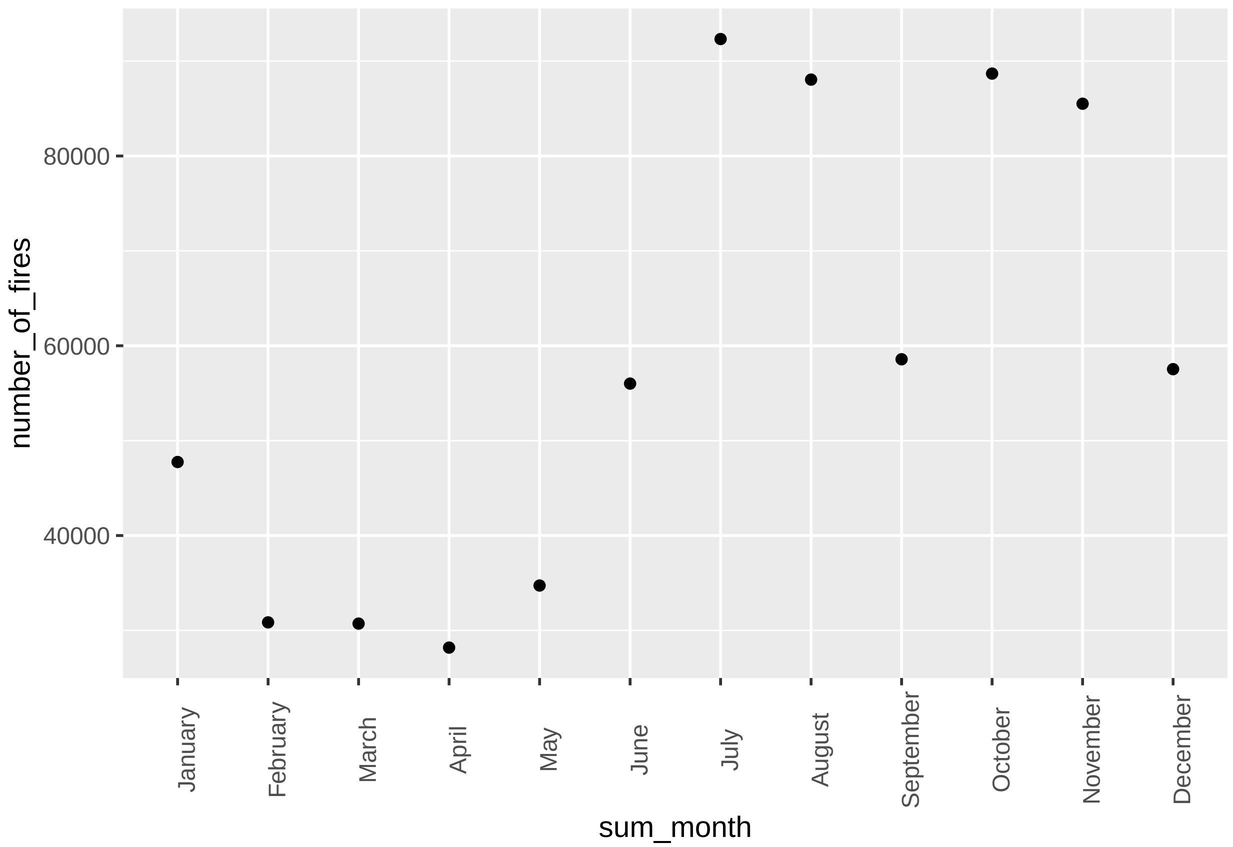
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Point Charts

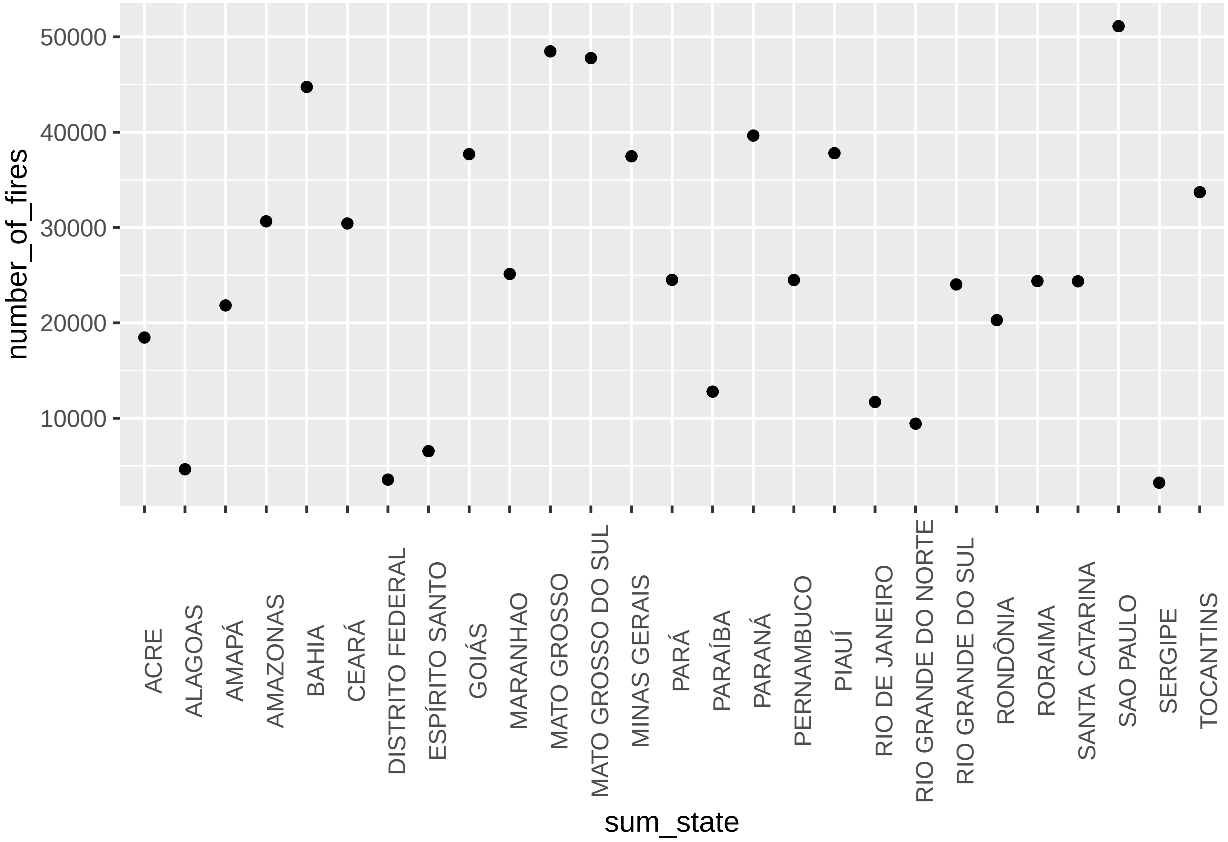
```
ggplot(data = data_month, aes(x=sum_month, y=number_of_fires)) + geom_point() + geom_line()+ stat_smoot
```

```
## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```

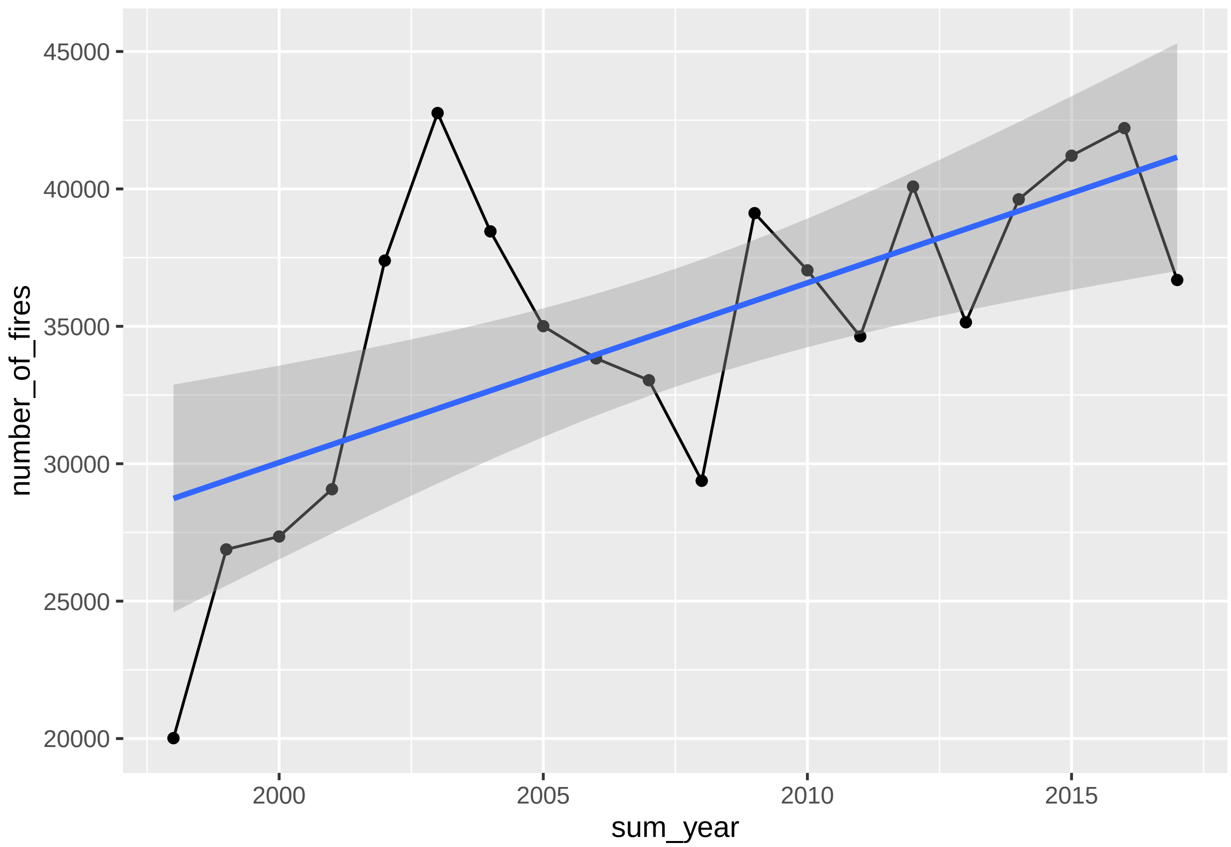


```
ggplot(data = data_state, aes(x=sum_state, y=number_of_fires)) + geom_point() + geom_line()+ stat_smoot
```

```
## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```



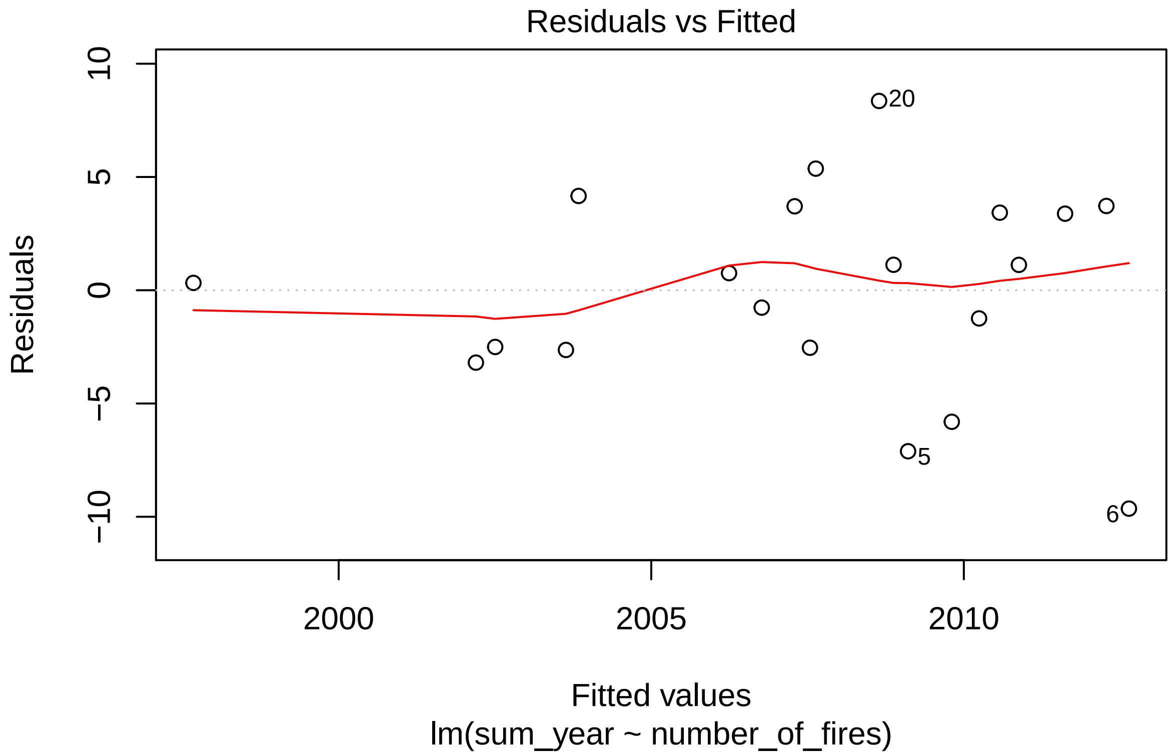
```
ggplot(data = data_year, aes(x=sum_year, y=number_of_fires)) + geom_point() + geom_line()+ stat_smooth(
```



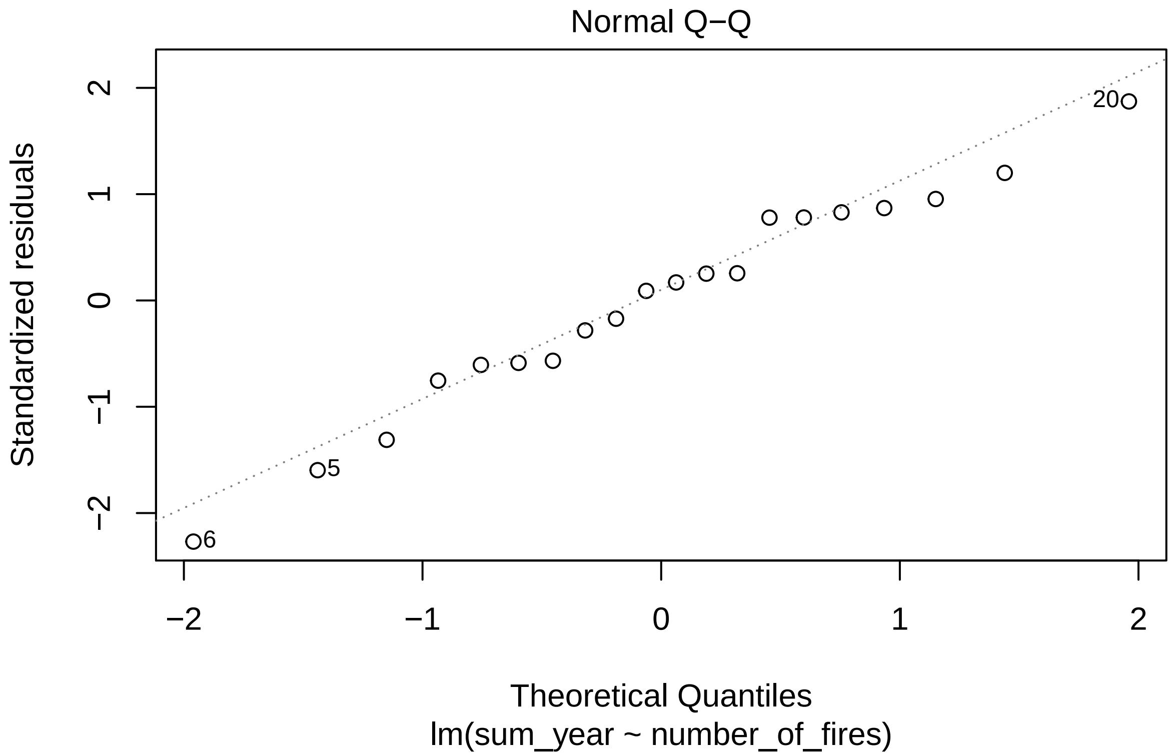
Linear Regression

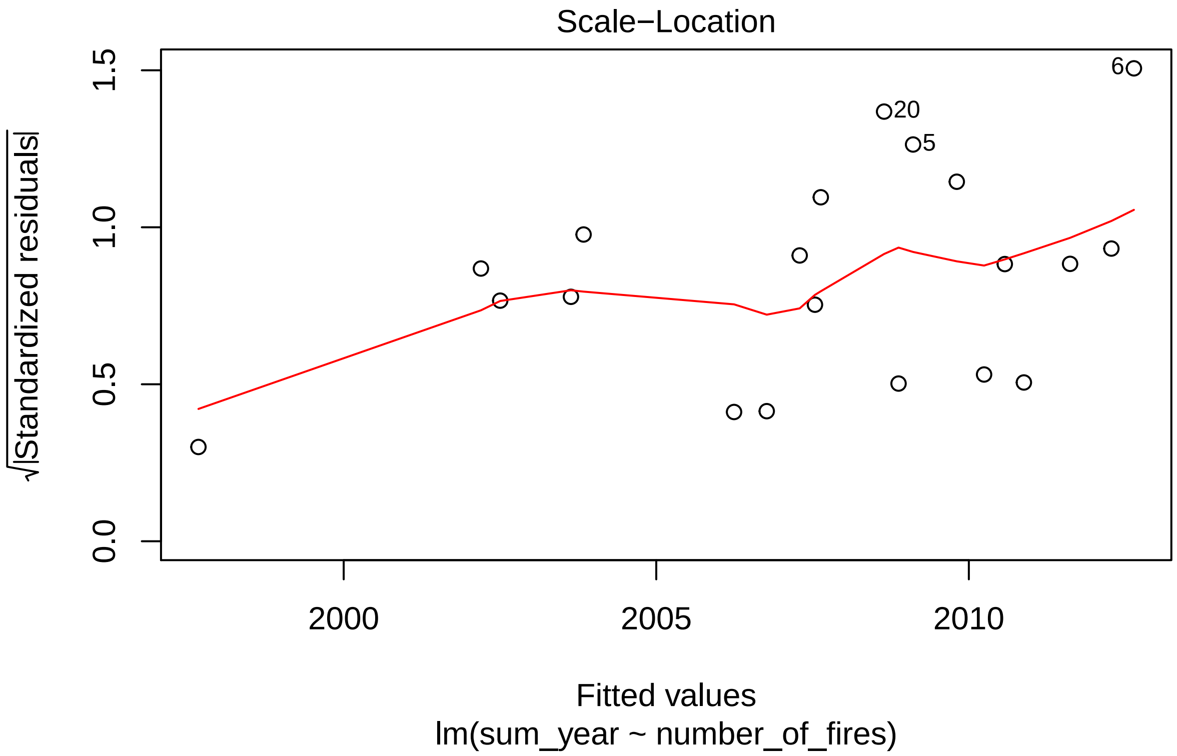
We can train our regression on the 70% of our data, and after that check if it was working on the rest.

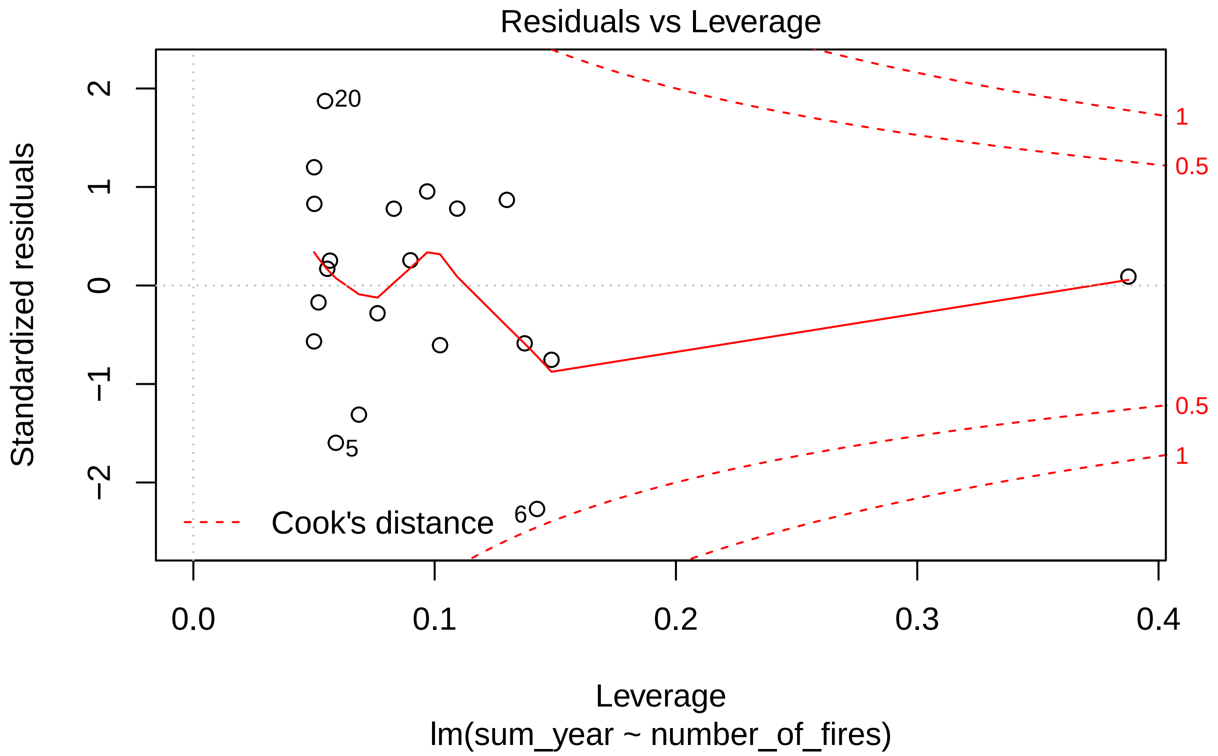
```
model <- lm(sum_year~number_of_fires, data=data_year)
data_year$prediction = predict(model, newdata = data_year)

plot(model)
```



Residuals vs Fitted
Residuals
10
5
0
−5
−10
20
5
6
2000
2005
2010
Fitted values
lm(sum_year ~ number_of_fires)


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
20
6
5
−2
−1
0
1
2
Theoretical Quantiles
lm(sum_year ~ number_of_fires)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2000
2005
2010
Fitted values
lm(sum_year ~ number_of_fires)
6
20
5



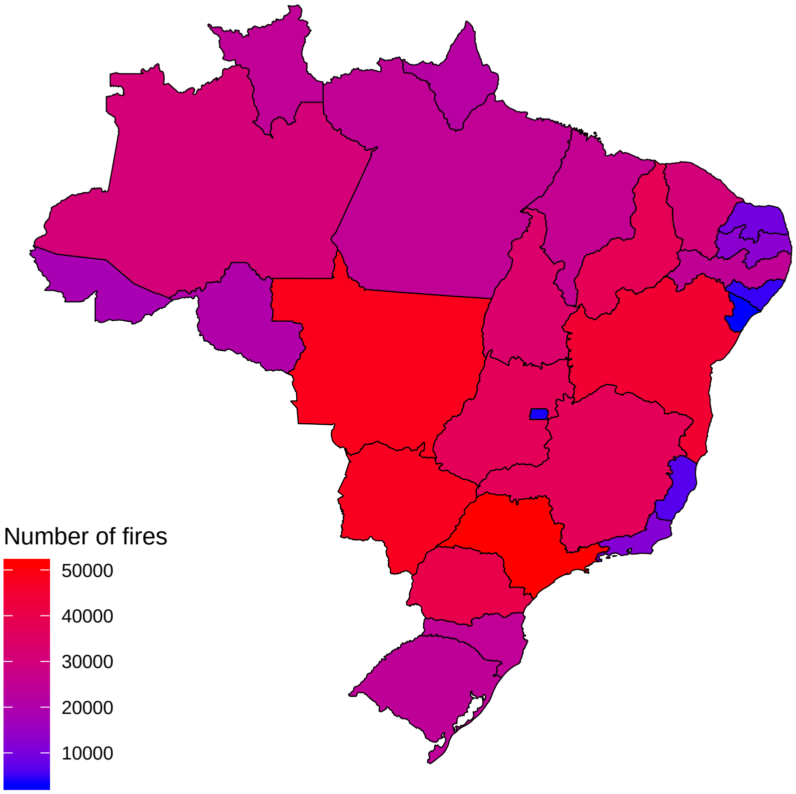
```
data_year$prediction
```

```
##  [1] 1997.676 2002.195 2002.503 2003.635 2009.108 2012.641 2009.807
##  [8] 2007.539 2006.767 2006.244 2003.837 2010.244 2008.876 2007.294
## [15] 2010.881 2007.632 2010.576 2011.620 2012.280 2008.644
```

Map visualisation

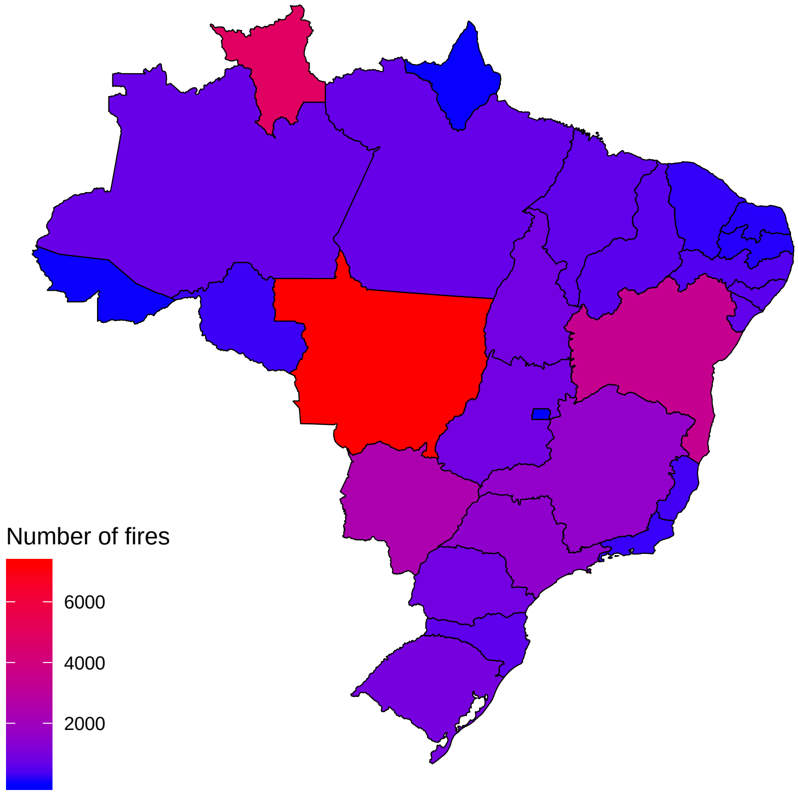
Overall number of fires

```
# Create map of Brasil states
states_map <- get_brmap(geo = "State",
                        geo.filter = NULL,
                        class = "sf") %>% arrange(nome)
# Append column with data to be visualized
states_map$number_of_fires <- data_state$number_of_fires
# Plot the created map
plot_custom_map(states_map)
```



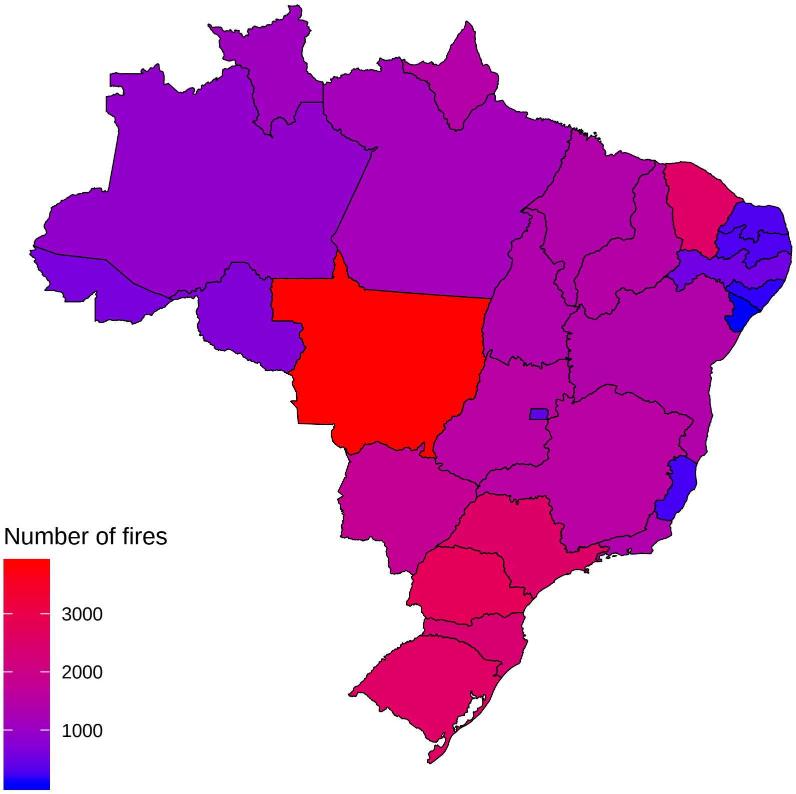
by month

```
states_map <- get_brmap(geo = "State",
                        geo.filter = NULL,
                        class = "sf") %>% arrange(nome)
monthly_data <- data[data$month=='March',]
monthly_data_agg <- aggregate(list(number_of_fires=monthly_data$number), by=list(sum_state=monthly_data
states_map$number_of_fires <- monthly_data_agg$number_of_fires
plot_custom_map(states_map)
```



by year

```
states_map <- get_brmap(geo = "State",
                        geo.filter = NULL,
                        class = "sf") %>% arrange(nome)
yearly_data <- data[data$year==2017,]
yearly_data_agg <- aggregate(list(number_of_fires=yearly_data$number), by=list(sum_state=yearly_data$st
states_map$number_of_fires <- yearly_data_agg$number_of_fires
plot_custom_map(states_map)
```


Number of fires
3000
2000
1000