```r
library(RColorBrewer)
library(ggplot2)
library(ggExtra)
library(ggthemes)
library(patchwork)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(vioplot)
```

```
## Loading required package: sm
## Package 'sm', version 2.2-6.0: type help(sm) for summary information
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(hrbrthemes)
```

Car information dataset analysis.

The dataset contains 399 rows of 9 features, which contains some general properties of cars. These 9 features are the following:

1. Name: Unique identifier for each automobile.
2. MPG: Fuel efficiency measured in miles per gallon.
3. Cylinders: Number of cylinders in the engine.
4. Displacement: Engine displacement, indicating its size or capacity.
5. Horsepower: Power output of the engine.
6. Weight: Weight of the automobile in pounds.
7. Acceleration: Capability to increase speed, measured in seconds to 60 miles/hour.
8. Model Year: Year of manufacture for the automobile model.
9. Origin: Country or region of origin for each automobile.

The dataset can be found via this **link**

Data exploration

```
setwd("/media/sf_SF/Fedora/R_course/Assignment/")
car_data <- read.csv("Automobile.csv")
head(car_data)
```

```
##                        name mpg cylinders displacement horsepower weight
## 1 chevrolet chevelle malibu  18         8          307        130   3504
## 2         buick skylark 320  15         8          350        165   3693
## 3        plymouth satellite  18         8          318        150   3436
## 4             amc rebel sst  16         8          304        150   3433
## 5               ford torino  17         8          302        140   3449
## 6          ford galaxie 500  15         8          429        198   4341
##   acceleration model_year origin
## 1         12.0         70    usa
## 2         11.5         70    usa
## 3         11.0         70    usa
## 4         12.0         70    usa
## 5         10.5         70    usa
## 6         10.0         70    usa
```

```
str(car_data)
```

```
## 'data.frame':    398 obs. of  9 variables:
##  $ name        : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : int  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model_year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : chr  "usa" "usa" "usa" "usa" ...
```

Let's check if there are empty values

```
missing_values <- which(is.na(car_data), arr.ind = TRUE)
print(missing_values)
```

```
##      row col
## [1,]  33   5
## [2,] 127   5
## [3,] 331   5
## [4,] 337   5
## [5,] 355   5
## [6,] 375   5
```

```
print(paste("Column 5 is",colnames(car_data)[5]))
```

```
## [1] "Column 5 is horsepower"
```

There are six missing values in the horsepower column.

The dataset contains of 3 categorical variables (name, model_year, cylinders and origin) and 6 numerical variables (mpg, cylinders, displacement, horsepower, weight and acceleration).

IMPORTANT NOTE: The amount of cylinders also falls under categorical variable, as it devides the cars in categorical groups based on their engine. Also, the miles per gallon will be ignored, because the mpg values will be calcated to liters/100k, which will be placed inside a new column

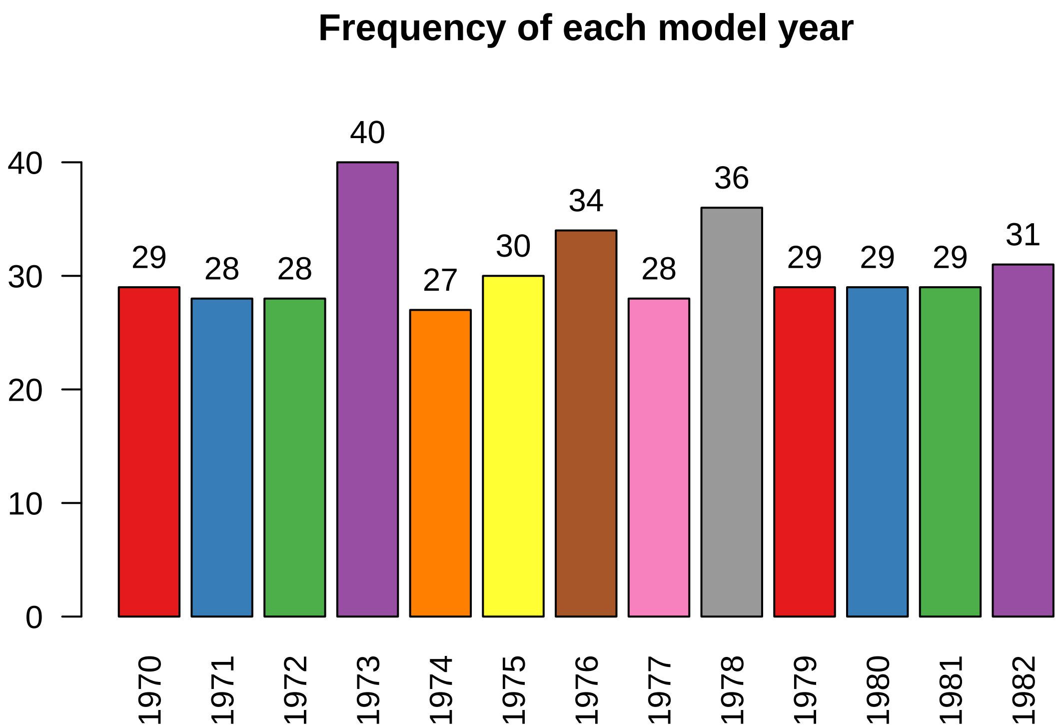
```r
car_data$model_year <- as.character(car_data$model_year)
car_data$model_year <- paste0("19",car_data$model_year)
car_data$cylinders <- as.character(car_data$cylinders)
car_data$L_100km <- 235.215 / car_data$mpg
cat_var <- c("name", "brand", "model_year", "origin", "cylinders")
num_var <- c("L_100km", "displacement", "horsepower", "weight", "acceleration")
```

Let's take a look at the frequencies of each categorical variable. Because of the huge amount of unique car models, no representative barplot can be generated. The complete dataset contains 37 car brands

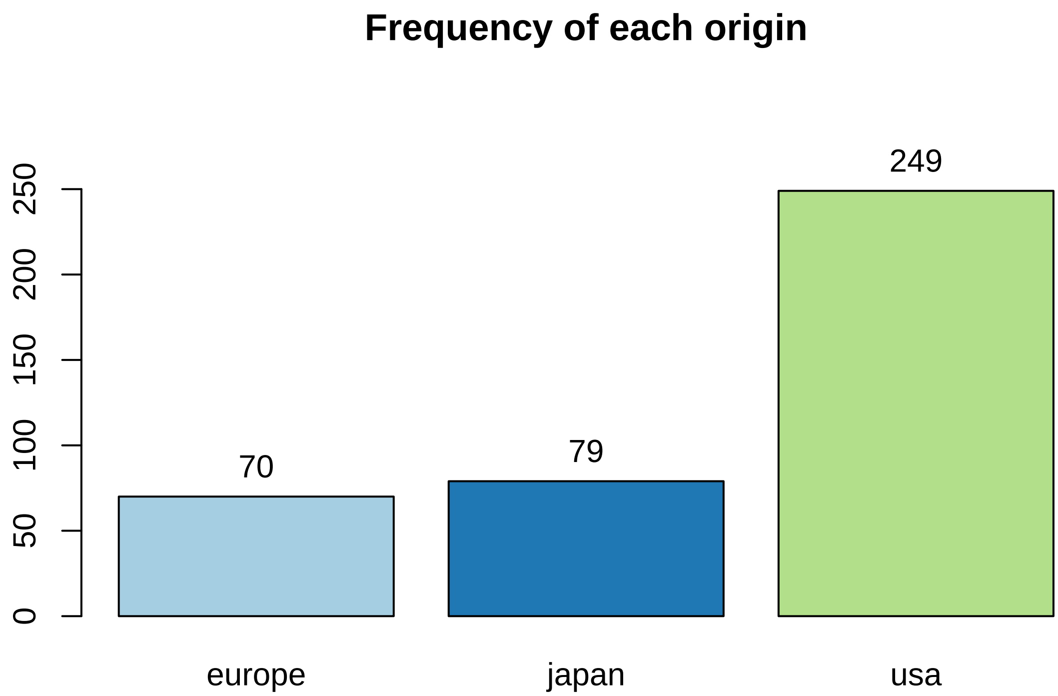
```r
car_data$brand <- sapply(strsplit(car_data$name, " "), `[`, 1)

paste0("Unique car models: ",length(unique(car_data$name)),', Unique car brands: ', length(unique(car_da
```

```
## [1] "Unique car models: 305, Unique car brands: 37"
```

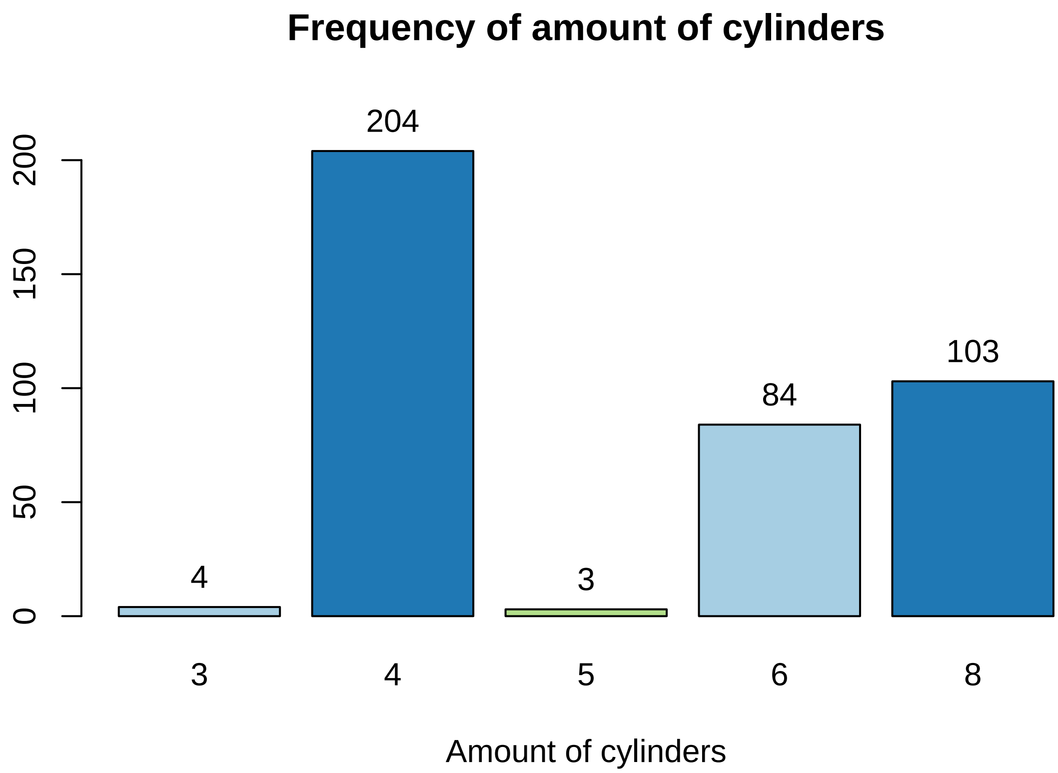
```r
bp <- barplot(table(car_data$model_year),
        main = "Frequency of each model year",
        ylim = c(0,45),
        col = brewer.pal(12, "Set1"),
        las = 2)
text(x=bp, y=table(car_data$model_year),label=table(car_data$model_year),pos=3)
```




```
bp <- barplot(table(car_data$origin),
        main = "Frequency of each origin",
        ylim = c(0,max(table(car_data$origin))+50),
        col = brewer.pal(3, "Paired"))
text(x=bp, y=table(car_data$origin),label=table(car_data$origin),pos=3)
```



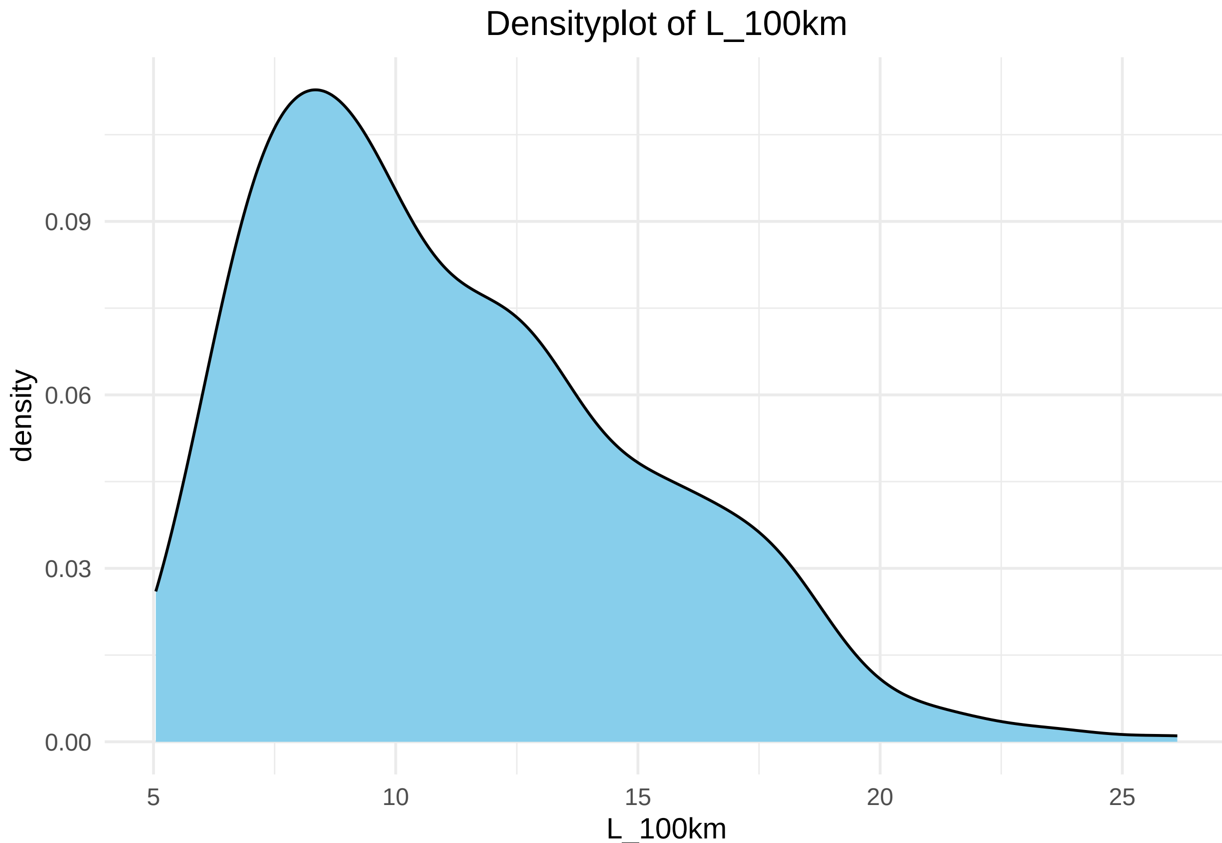

```r
bp <- barplot(table(car_data$cylinders),
        main = "Frequency of amount of cylinders",
        xlab = "Amount of cylinders",
        ylim = c(0,max(table(car_data$cylinders)+20)),
        col = brewer.pal(3, "Paired"))
text(x=bp, y=table(car_data$cylinders),label=table(car_data$cylinders),pos=3)
```

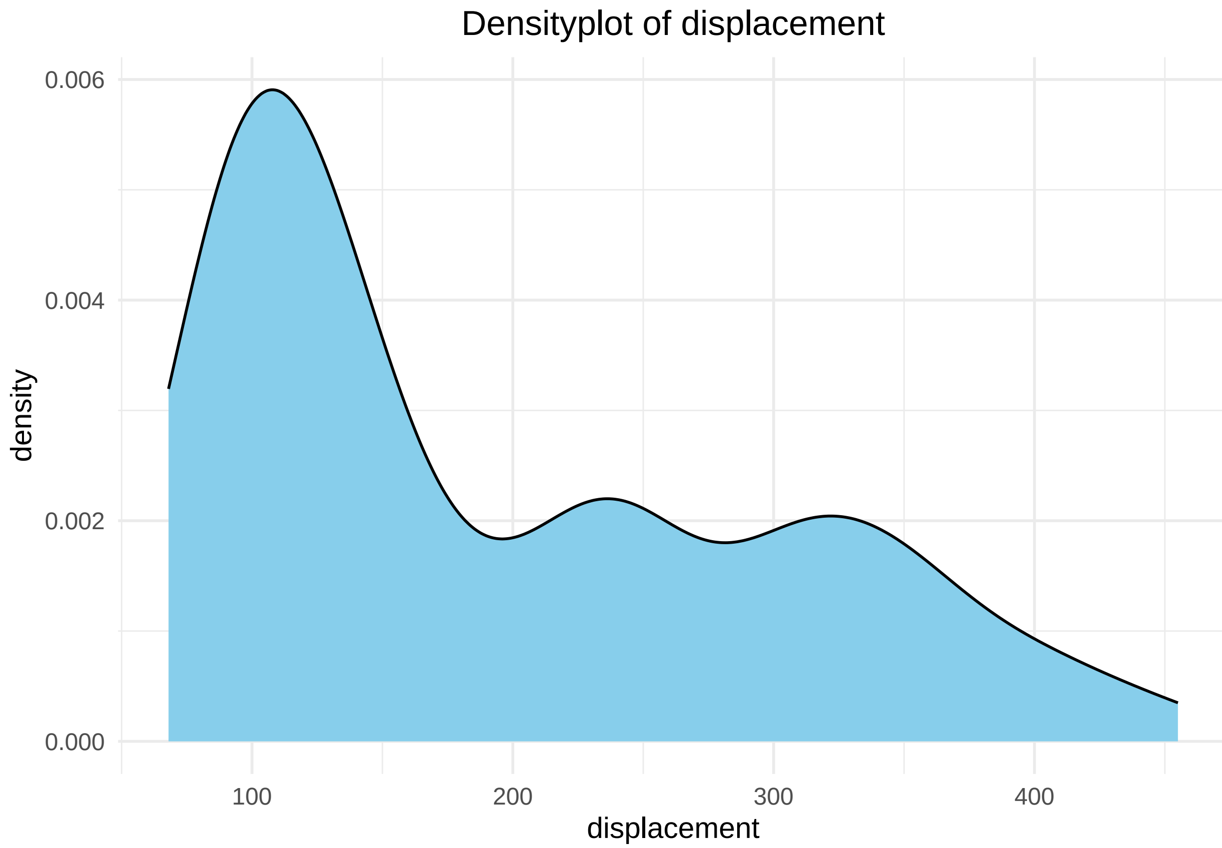


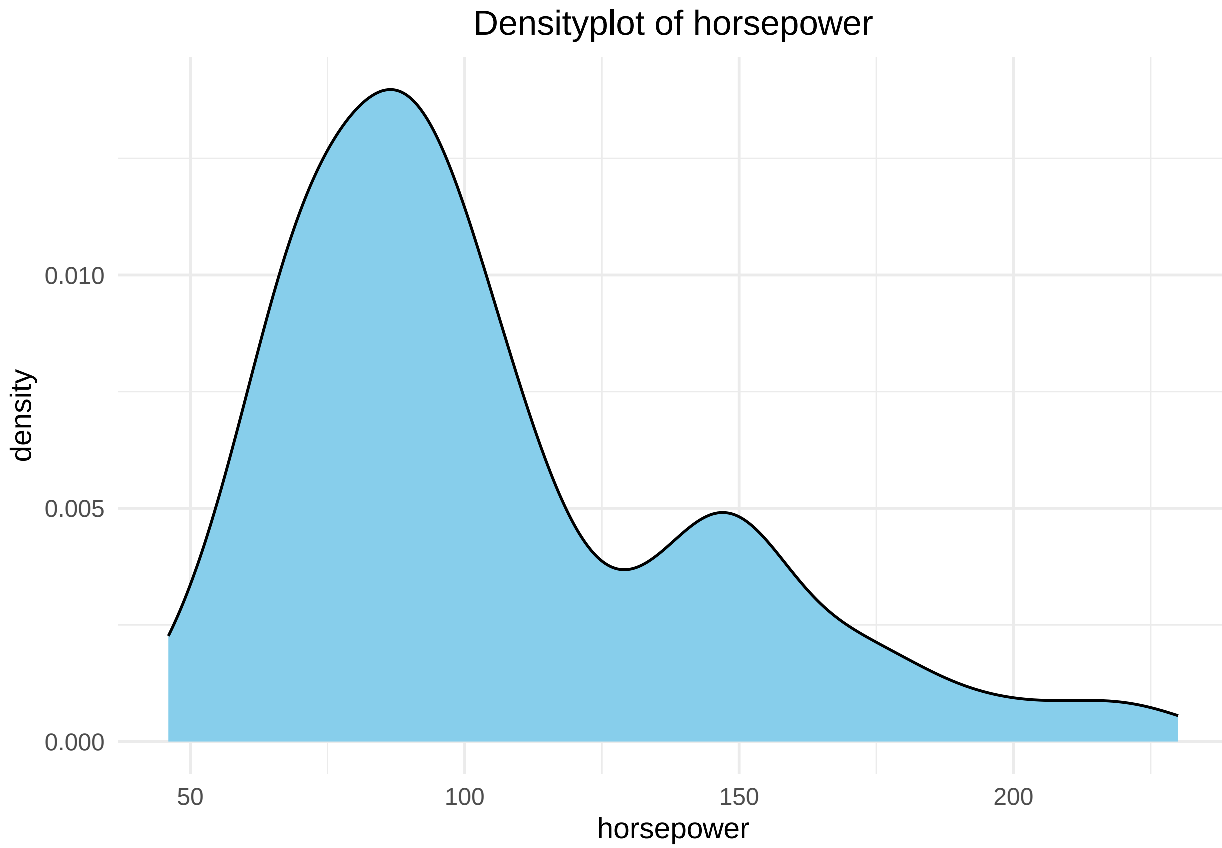
The plots above show that the model years are fairly normaliy distributed. On the other hand, the origin and amount of cylinders of the cars are not equally distributed.

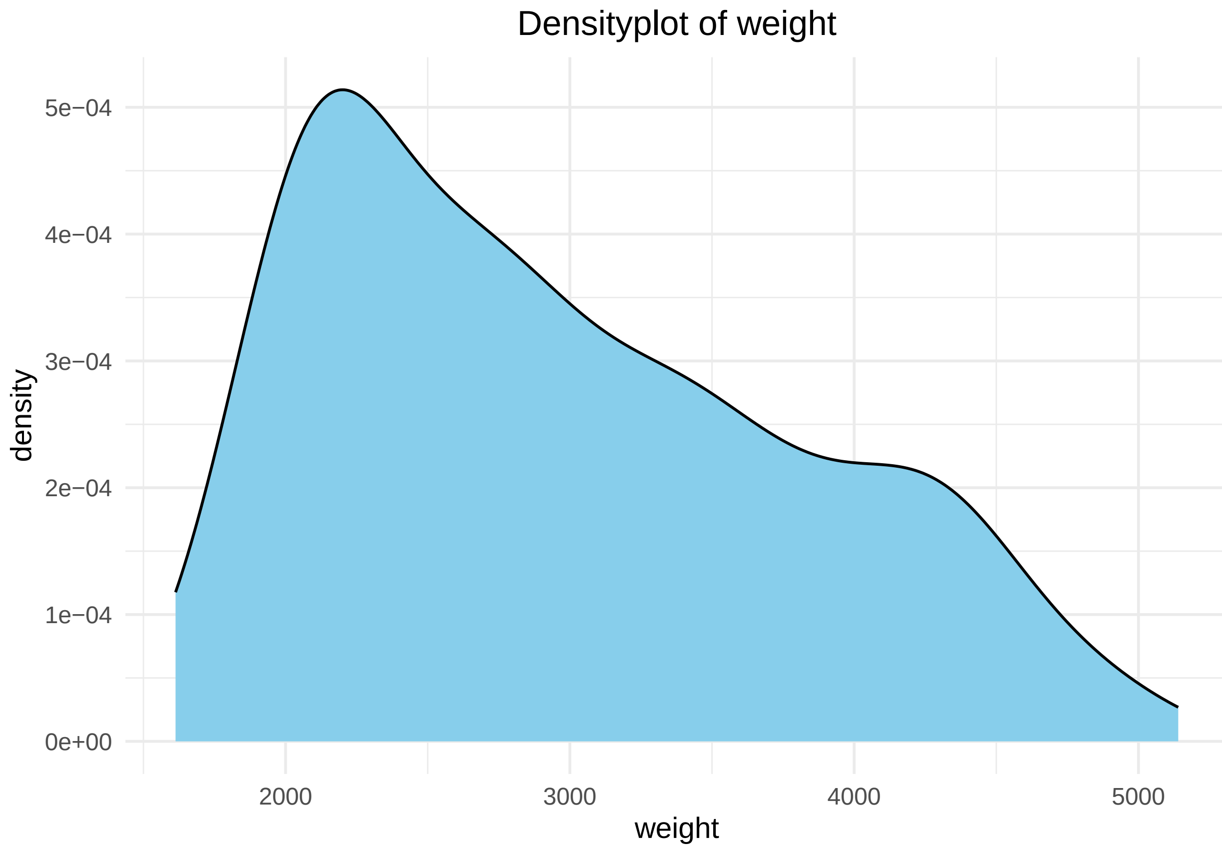
Let's take a look at density and distribution of the numerical variables now.

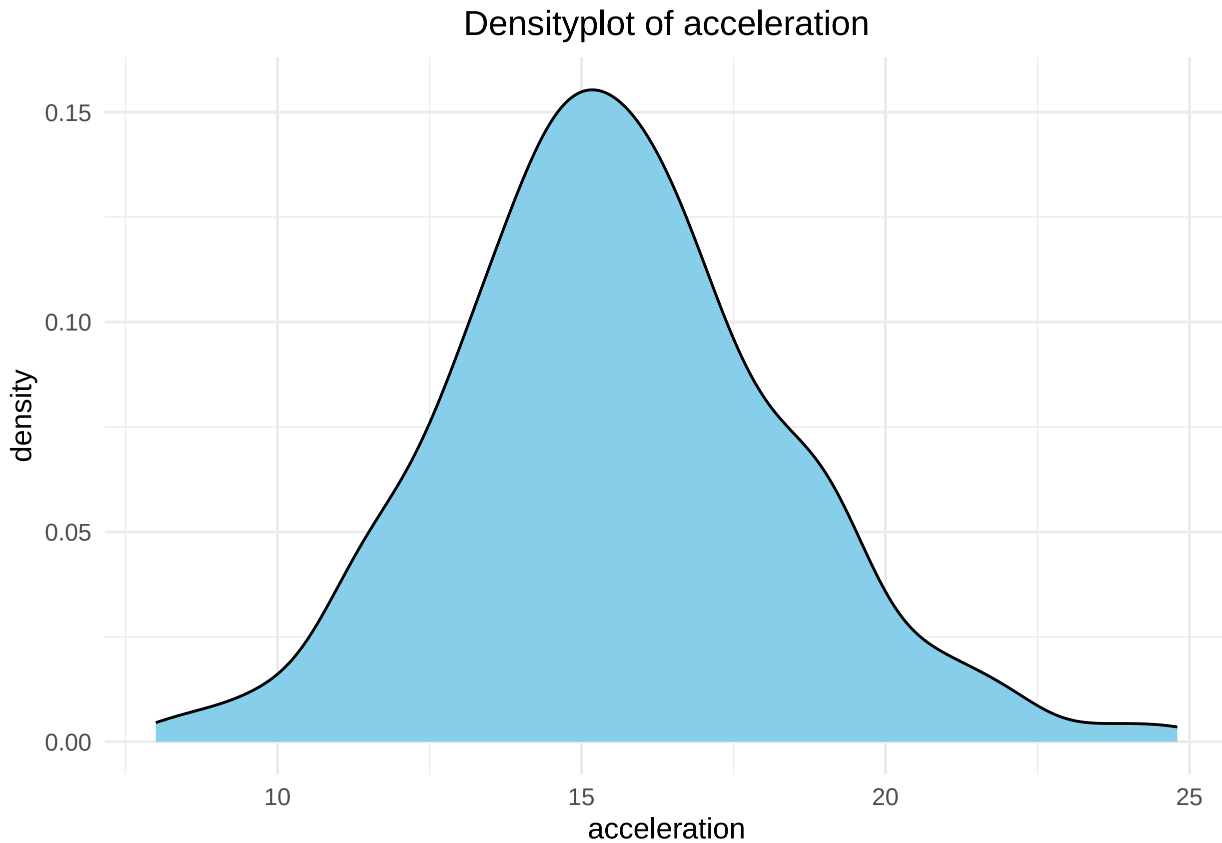
```
for (var in num_var) {
  p <- ggplot(car_data, aes(x = !!sym(var))) +
    geom_density(fill = "skyblue", color = "black") +
    labs(title = paste("Densityplot of", var)) +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))
  print(p)
}
```


Densityplot of L_100km
density
0.09
0.06
0.03
0.00
5
10
15
20
25
L_100km


Densityplot of displacement
0.006
0.004
0.002
0.000
density
100
200
300
400
displacement


Densityplot of horsepower
0.010
0.005
0.000
density
50
100
150
200
horsepower

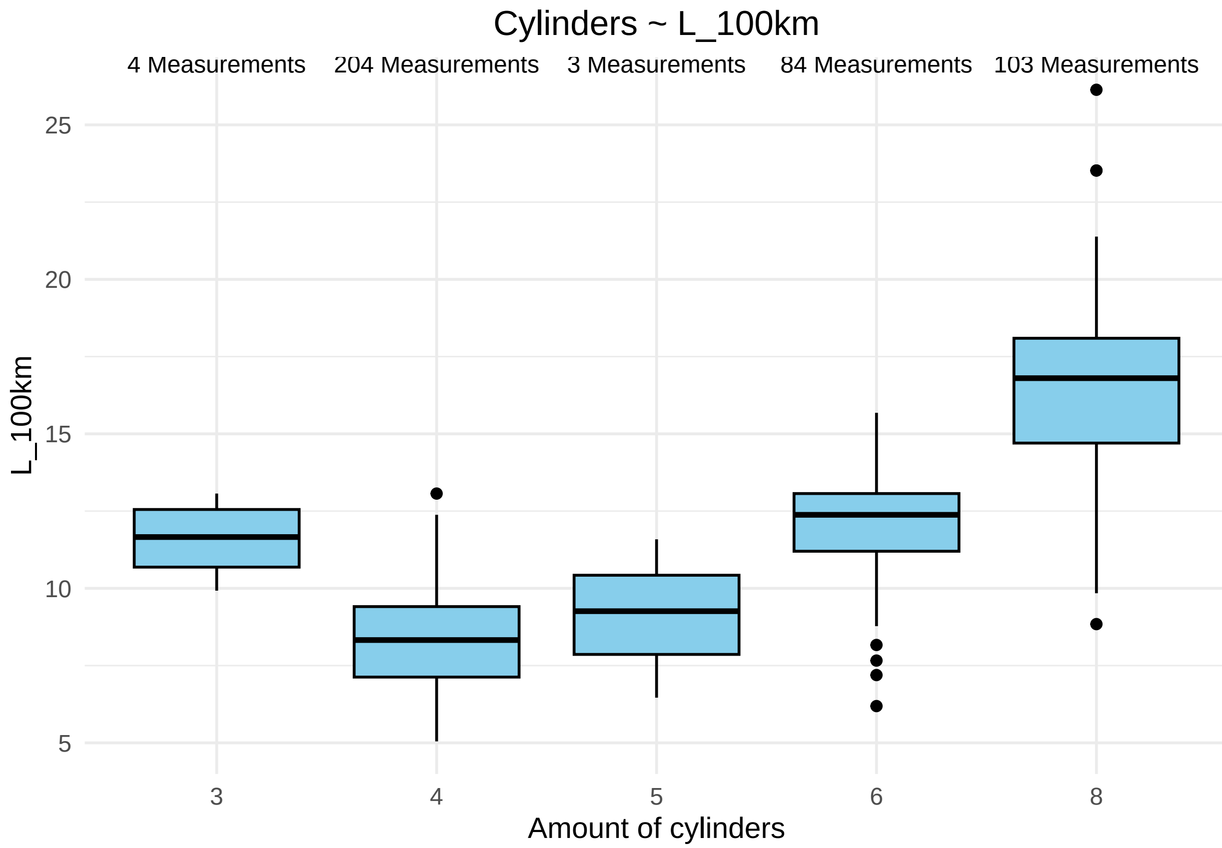

Densityplot of weight
5e−04
4e−04
3e−04
2e−04
1e−04
0e+00
density
2000
3000
4000
5000
weight

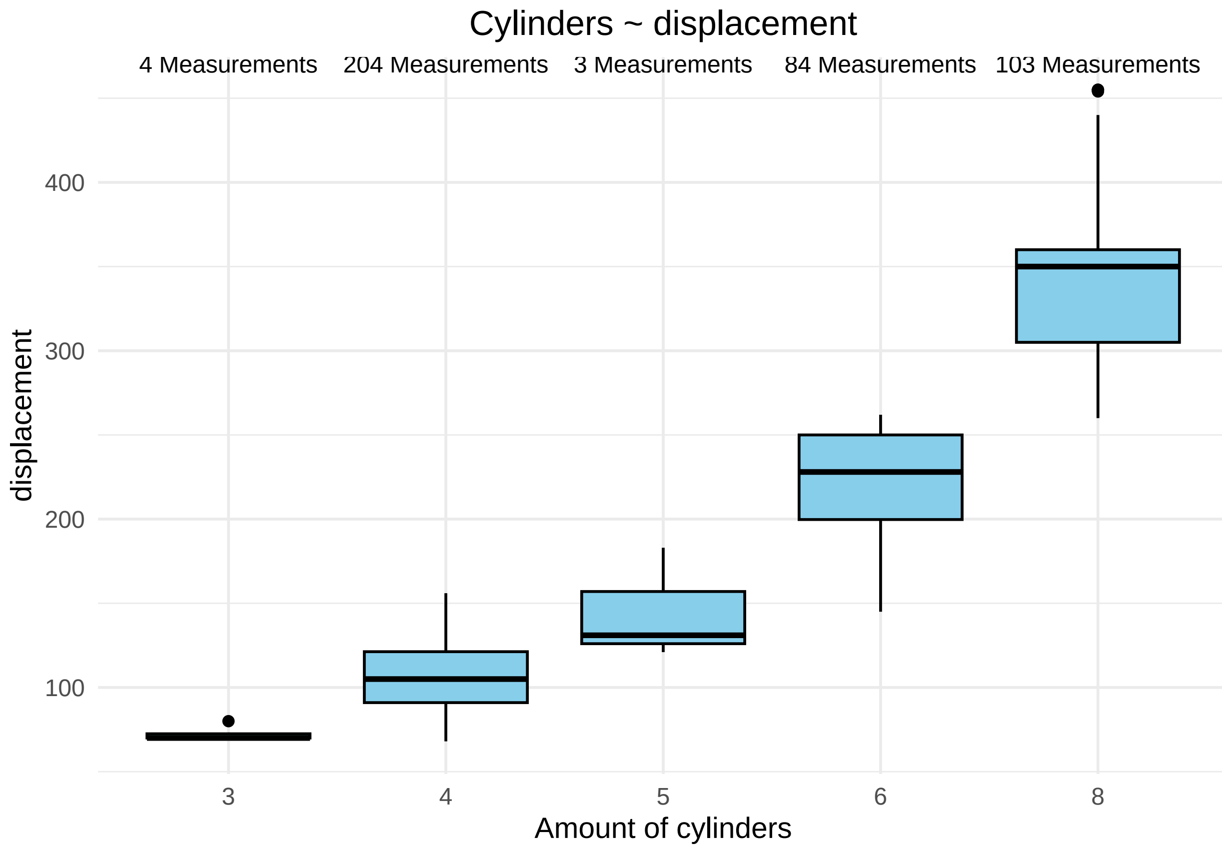


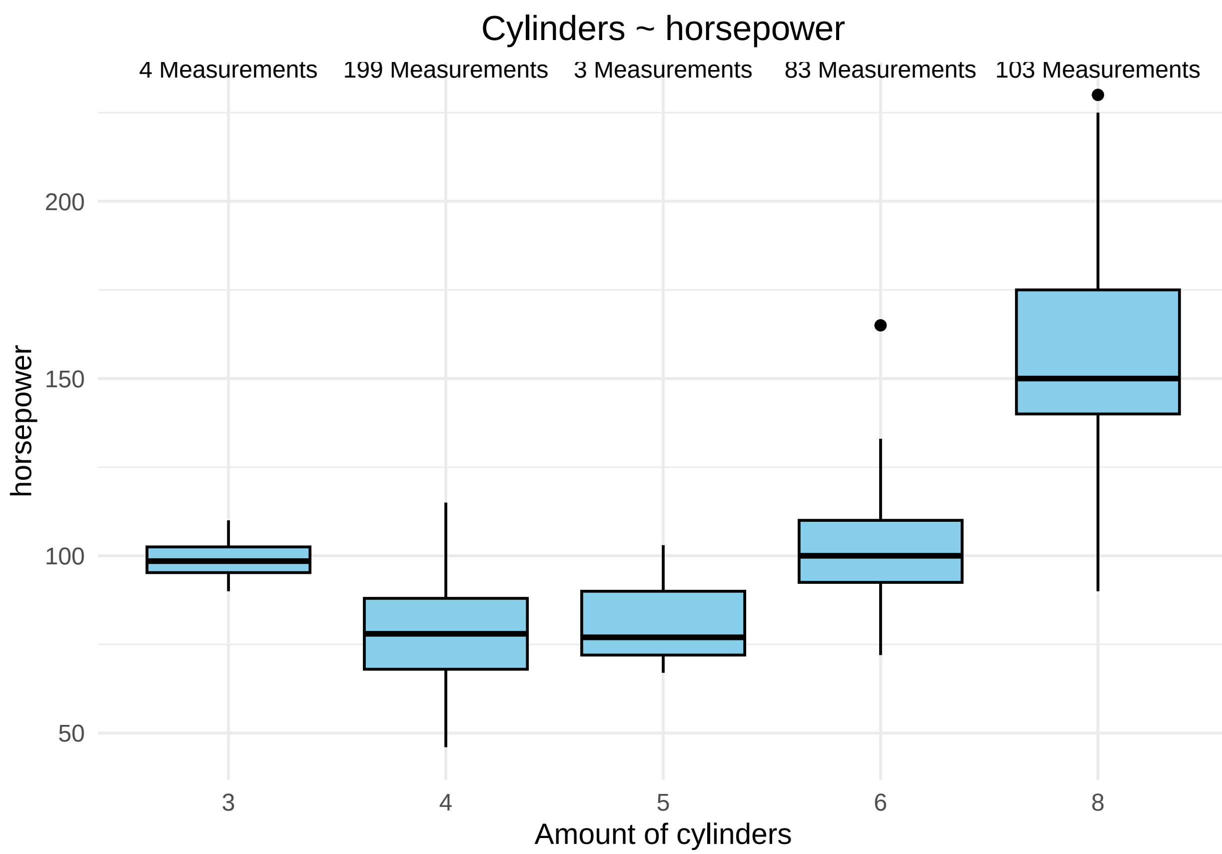
The acceleration variable has a normal distribution. The other numerical variables have a right skewness

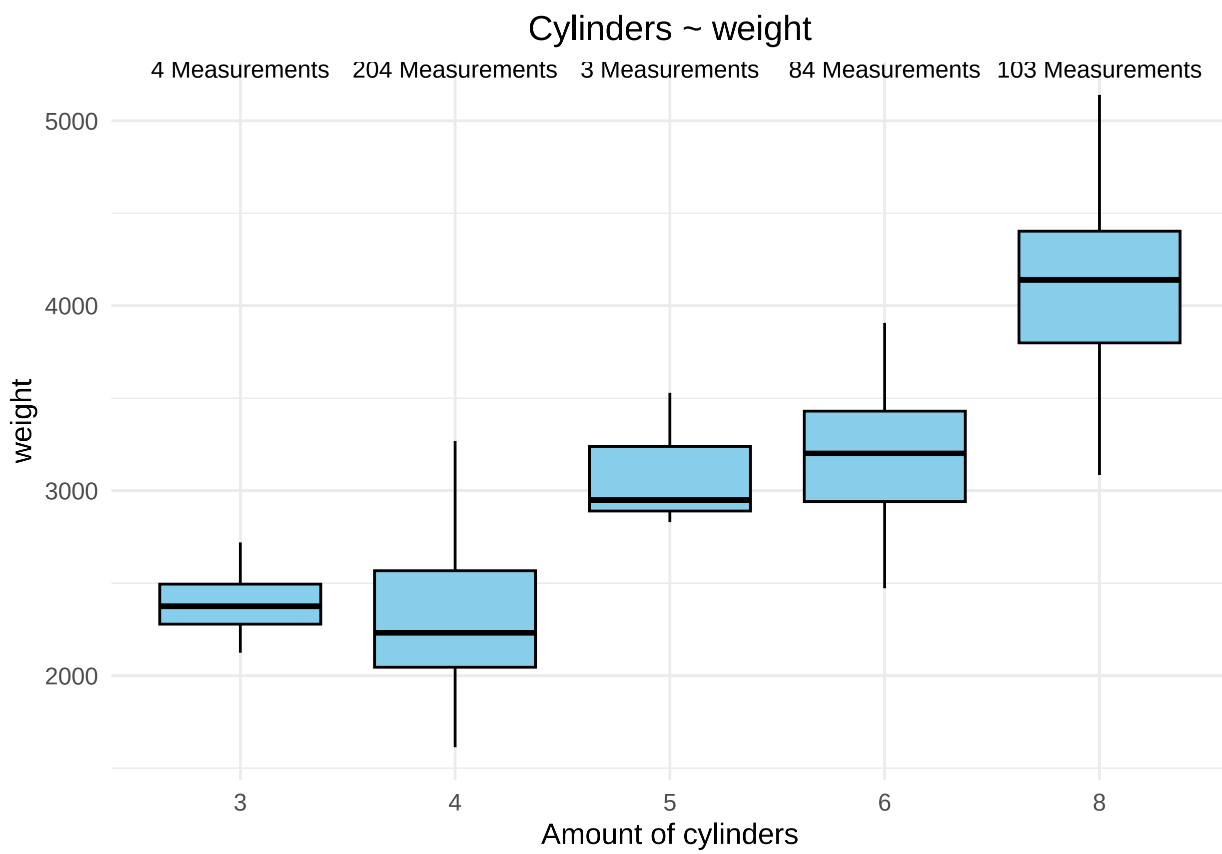
With this dataset, some genereal research questions can be asked. Firstly, do the amount of cylinders have a correlation with the numerical variables? To start, let's make some boxplots of the numerical variables, grouped by the amount of cylinders of the engine.

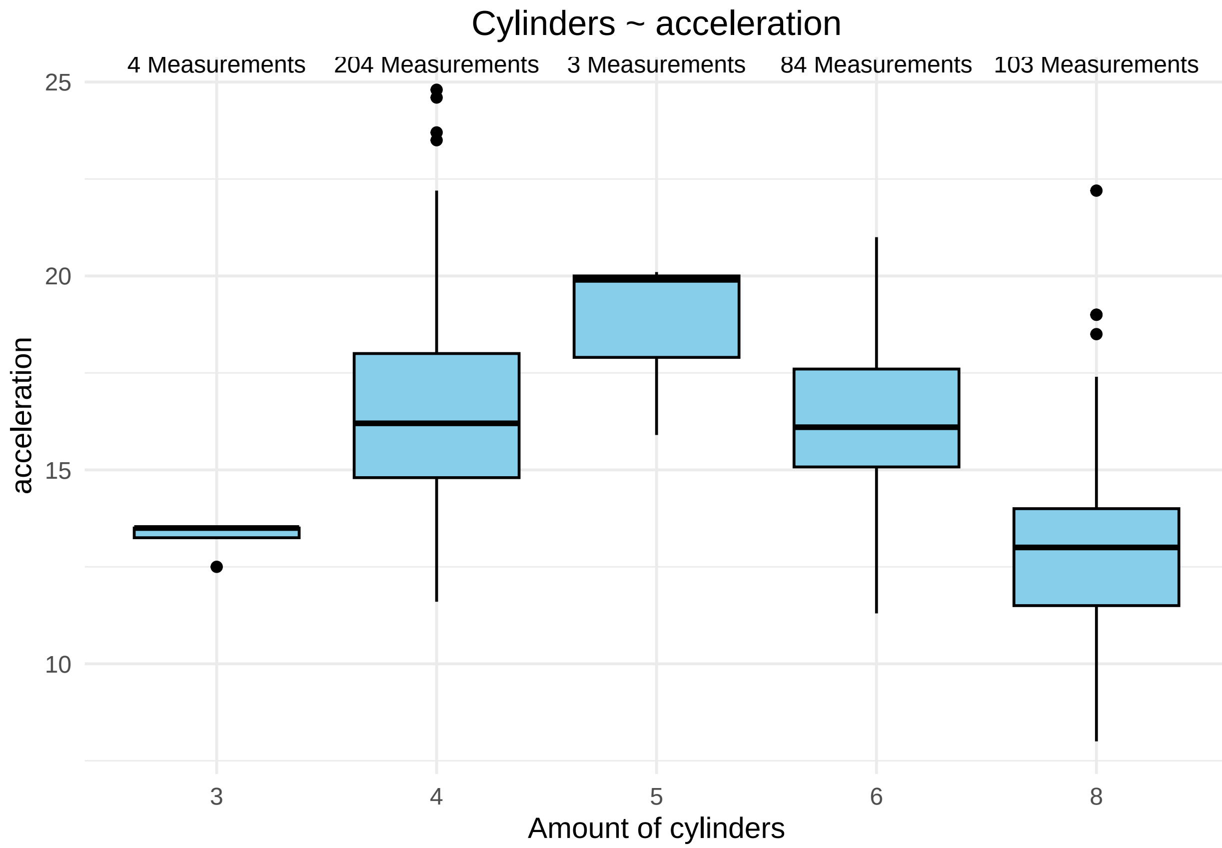
```
for (i in num_var) {
  measurements <- car_data %>%
  group_by(cylinders) %>%
  summarise(count = sum(!is.na(!!sym(i))))

  p <- ggplot(car_data, aes(x = as.factor(cylinders), y = !!sym(i))) +
    geom_boxplot(fill = "skyblue", color = "black") +
    geom_text(data = measurements,aes(label = paste(count, "Measurements"),
                  y = max(car_data[[i]], na.rm = TRUE)),
              vjust = -1, size = 3) +
    labs(title = paste("Cylinders ~", i), x = "Amount of cylinders") +
    theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5))
  print(p)
}
```


Cylinders ~ L_100km
4 Measurements
204 Measurements
3 Measurements
84 Measurements
103 Measurements
L_100km
25
20
15
10
5
3
4
5
6
8
Amount of cylinders


Cylinders ~ displacement
4 Measurements
204 Measurements
3 Measurements
84 Measurements
103 Measurements
400
300
200
100
displacement
3
4
5
6
8
Amount of cylinders

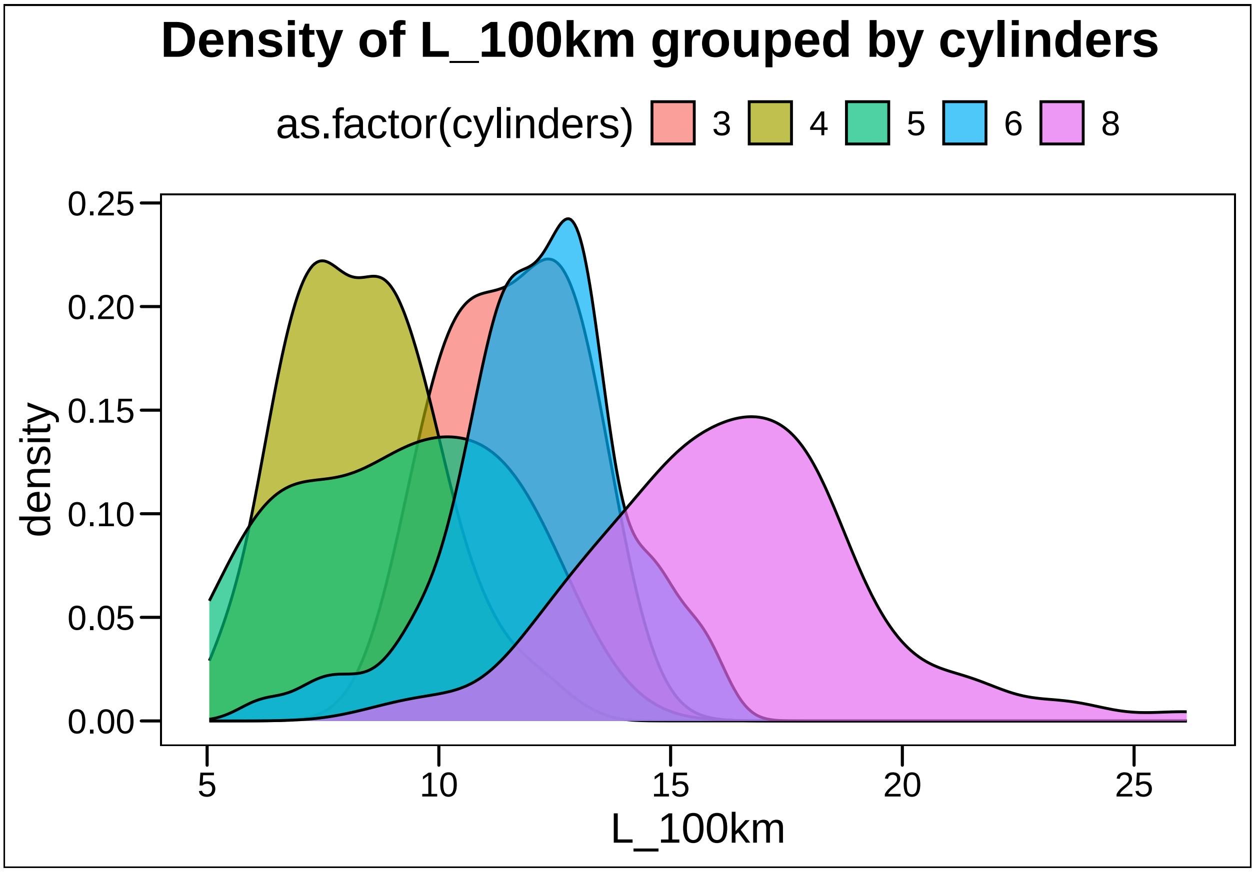

Cylinders ~ horsepower
4 Measurements
199 Measurements
3 Measurements
83 Measurements
103 Measurements
200
150
100
50
horsepower
3
4
5
6
8
Amount of cylinders

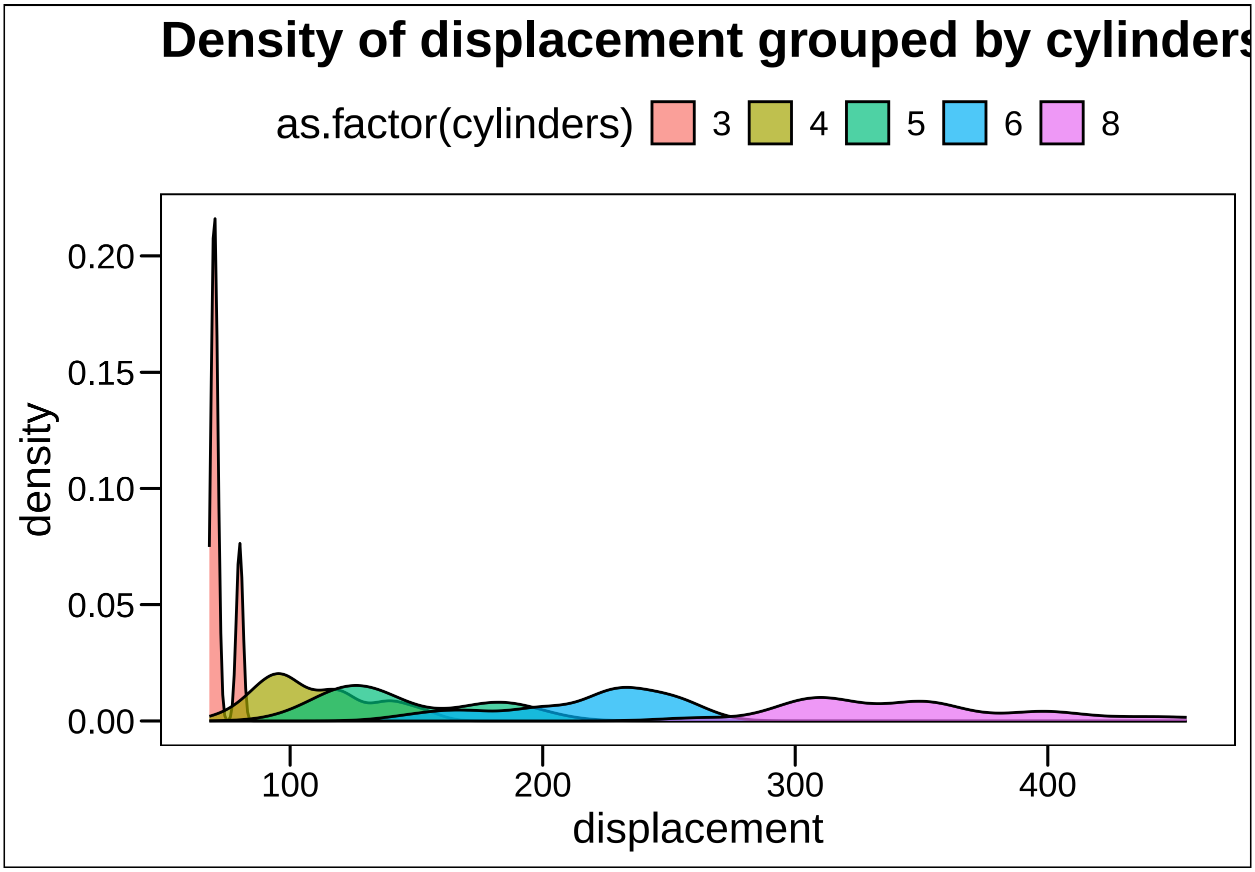

Cylinders ~ weight
4 Measurements
204 Measurements
3 Measurements
84 Measurements
103 Measurements
5000
4000
3000
2000
weight
3
4
5
6
8
Amount of cylinders

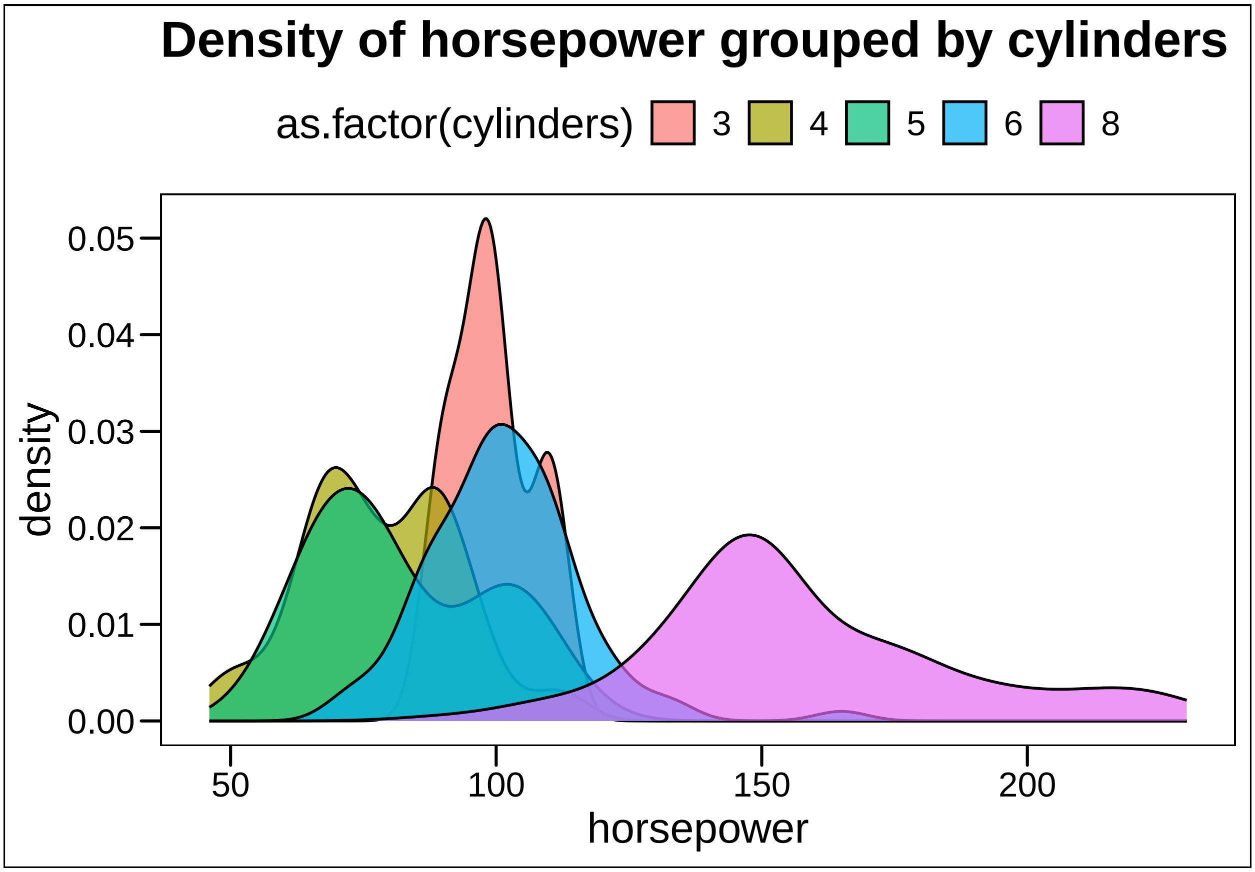


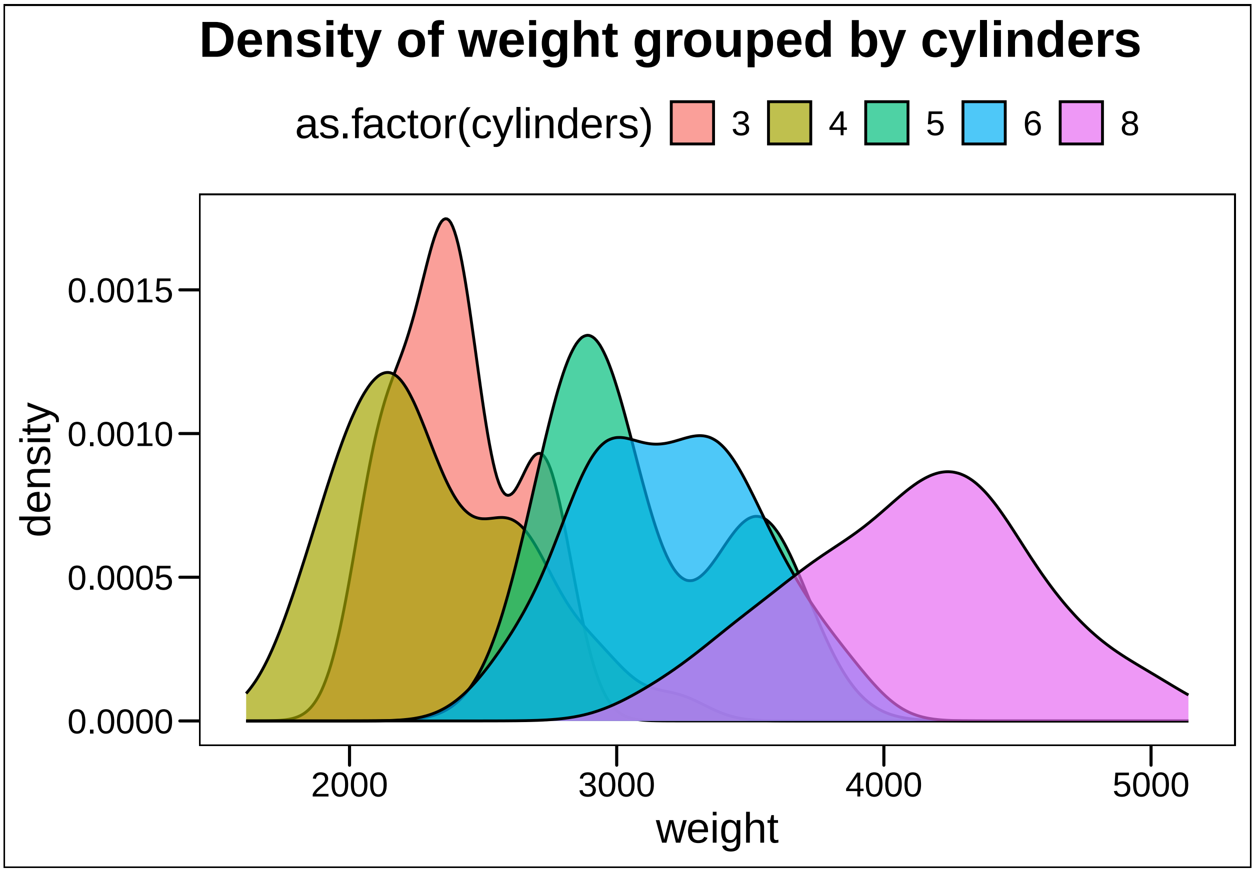
As the plots above depict, the median of fuel consumption, displacement, horsepower and weight go rises with the amount of cylinders. How about their distribution grouped by

```
for (i in num_var){
  p <- ggplot(data = car_data, aes(x = car_data[,i], fill = as.factor(cylinders))) +
    geom_density(alpha = 0.7) +
    labs(title = paste("Density of",i,"grouped by cylinders"), x = i) +
    theme_base() +
    theme(legend.position = "top")
  print(p)
}
```

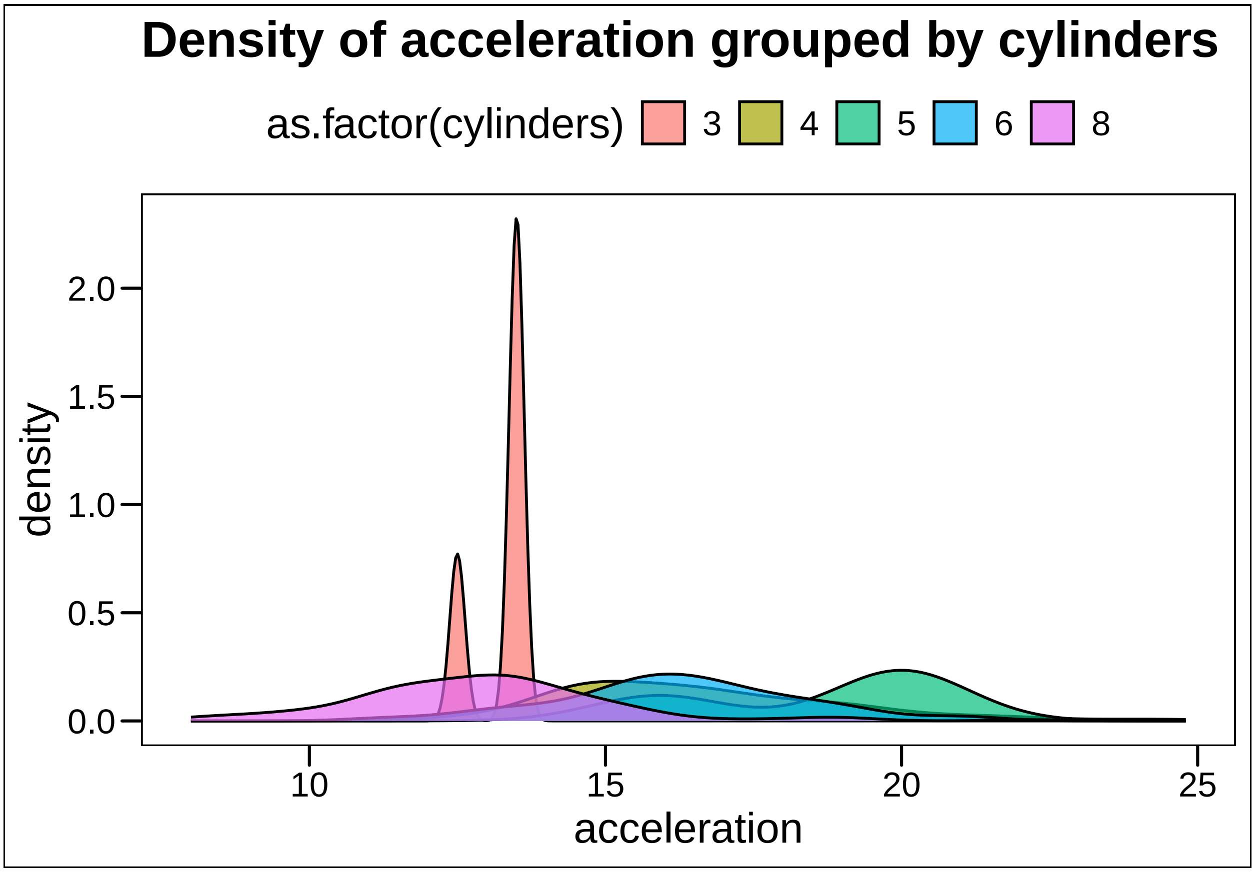

Density of L_100km grouped by cylinders
as.factor(cylinders)
3
4
5
6
8
density
0.25
0.20
0.15
0.10
0.05
0.00
5
10
15
20
25
L_100km


Density of displacement grouped by cylinders
as.factor(cylinders)
3
4
5
6
8
0.20
0.15
0.10
0.05
0.00
density
100
200
300
400
displacement


Density of horsepower grouped by cylinders
as.factor(cylinders)
3
4
5
6
8
0.05
0.04
0.03
0.02
0.01
0.00
density
50
100
150
200
horsepower



Density of weight grouped by cylinders
as.factor(cylinders)
3
4
5
6
8
0.0015
0.0010
0.0005
0.0000
density
2000
3000
4000
5000
weight

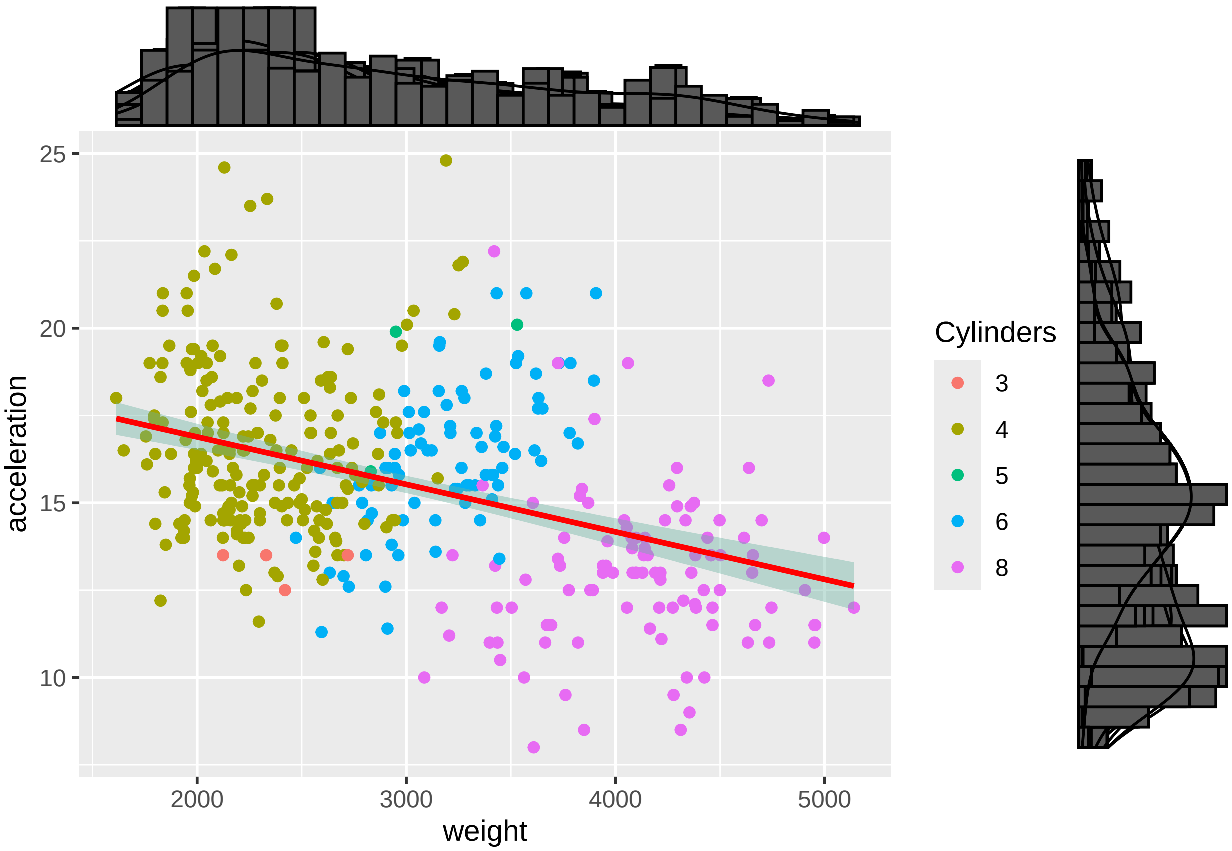


Let's make a scatterplot to further investigate the correlatations.

```
for (i in 1:length(num_var)) {
  for (j in (i + 1):length(num_var)) {  # Start from i + 1 to avoid duplicates
    if (i != j && !is.na(num_var[i]) && !is.na(num_var[j])) {
      p <- ggplot(car_data, aes_string(x = num_var[i],
                                       y = num_var[j],
                                       color = "cylinders")) +
        geom_point() +
        geom_smooth(method=lm , color="red", fill="#69b3a2", se=TRUE) +
        theme(legend.position = "right") +
        labs(color = "Cylinders")

      p2 <- ggMarginal(p, type = "densigram")
      print(p2)
    }
  }
}
```

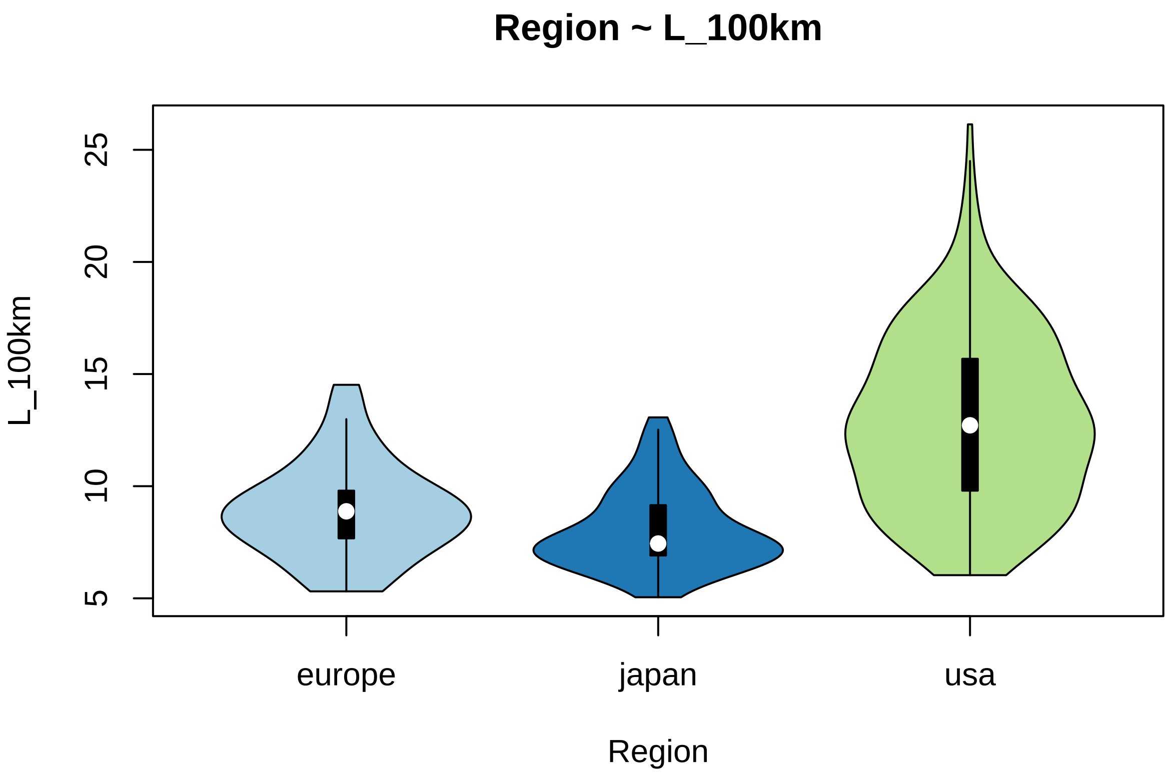
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

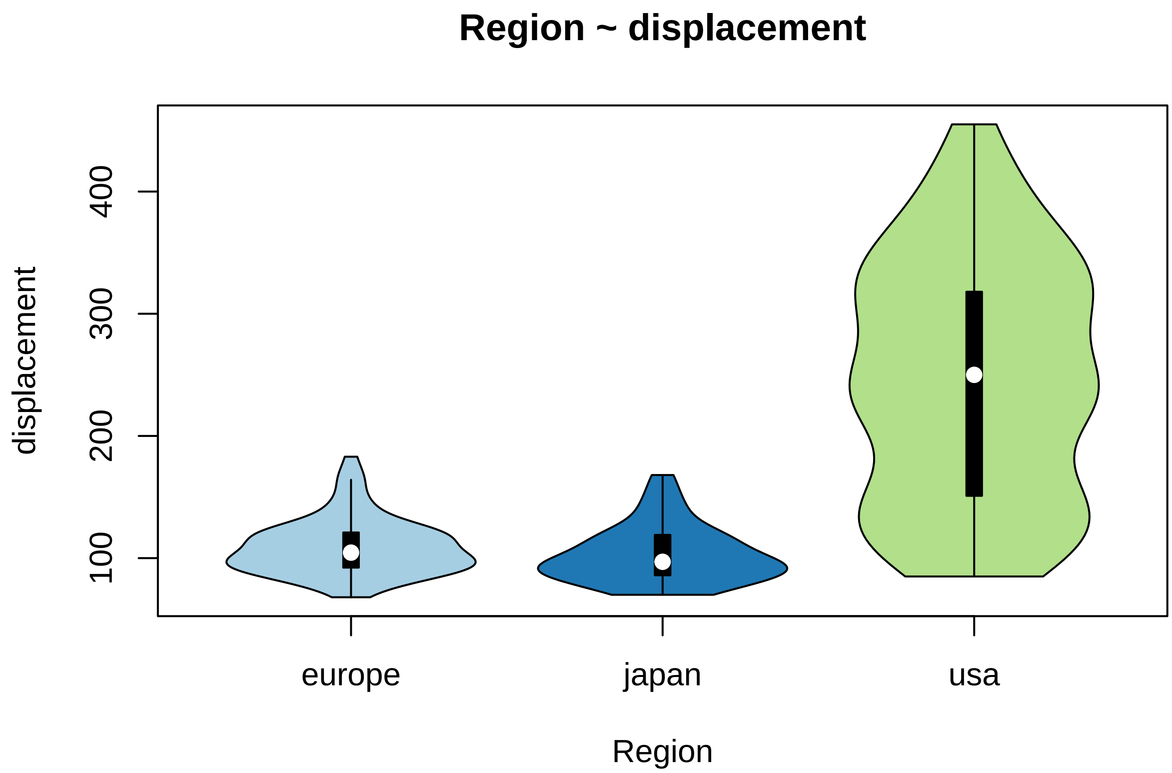
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

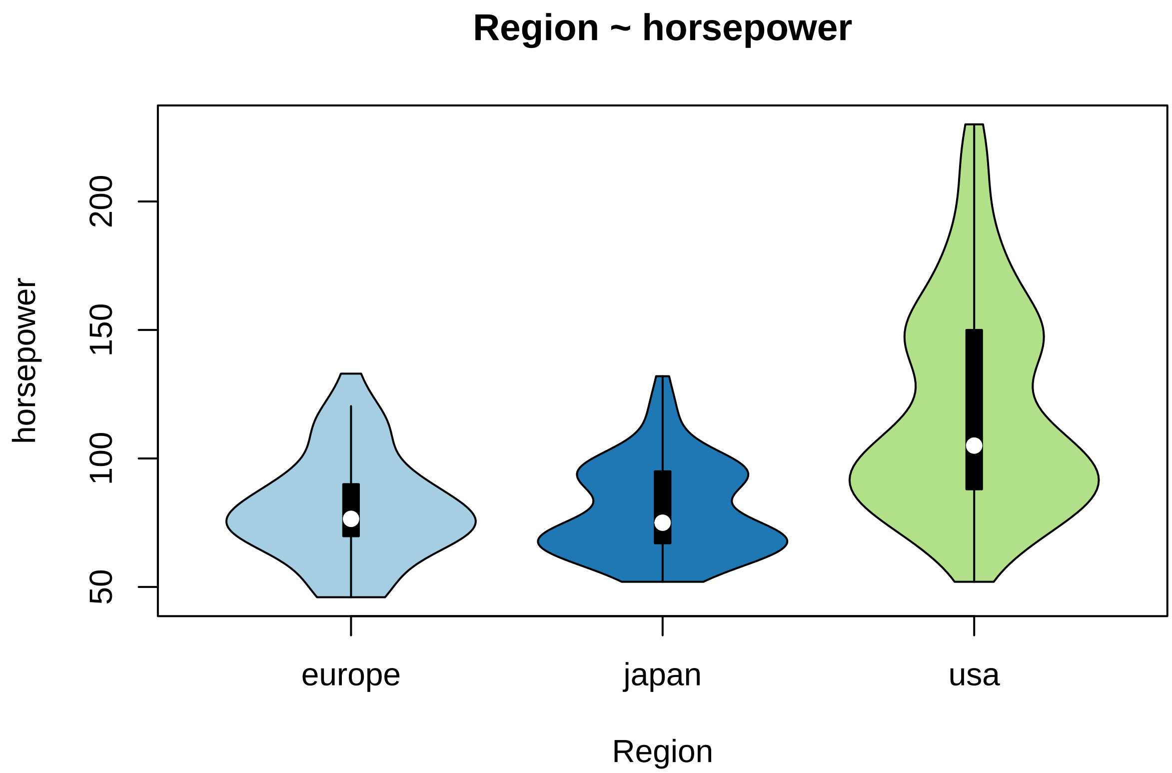


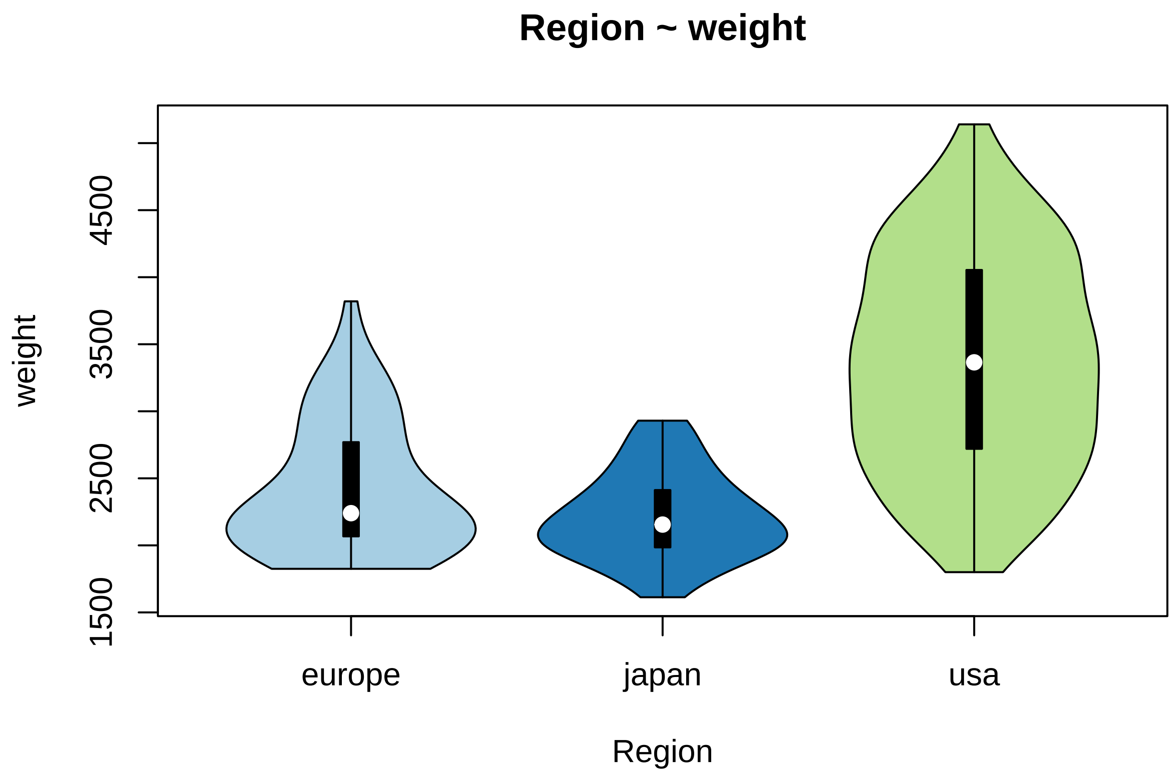
A second research question that can be asked, is if there any difference in the numerical variables between the origins of the cars? To check this, a violin plot was made.

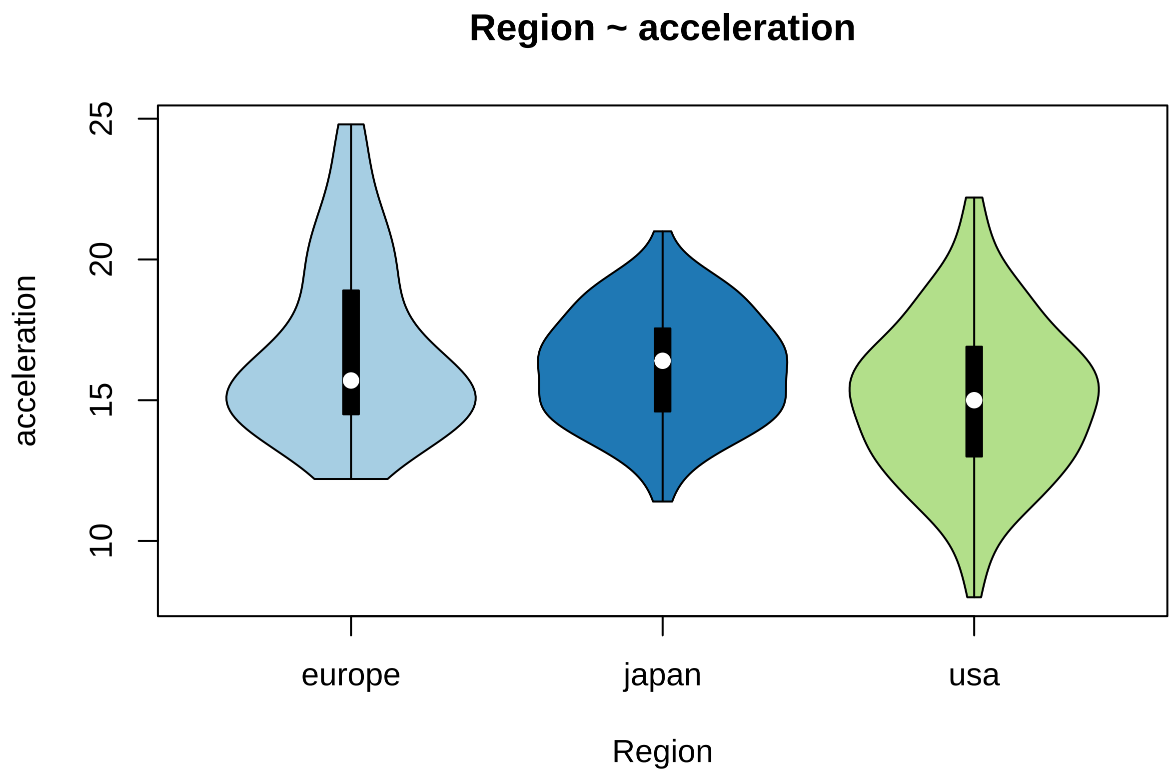
```
for (i in num_var) {
  p <- vioplot(car_data[,i] ~ car_data$origin,
               col = brewer.pal(3, "Paired"),
               xlab = "Region",
               ylab = paste(i),
               main = paste("Region ~",i))
  p
}
```


Region ~ L_100km
L_100km
5
10
15
20
25
europe
japan
usa
Region


Region ~ displacement
displacement
100
200
300
400
europe
japan
usa
Region

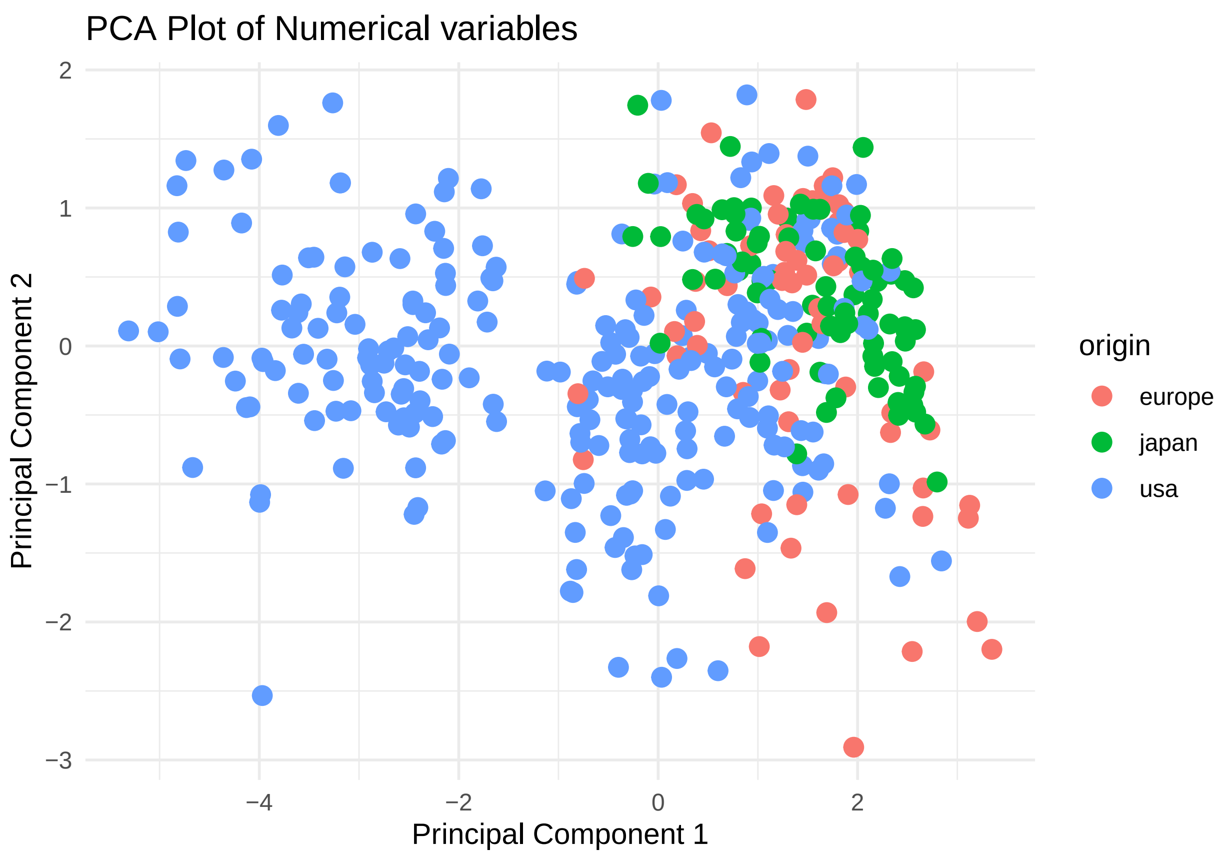

Region ~ horsepower
horsepower
50
100
150
200
europe
japan
usa
Region


Region ~ weight
weight
1500
2500
3500
4500
europe
japan
usa
Region

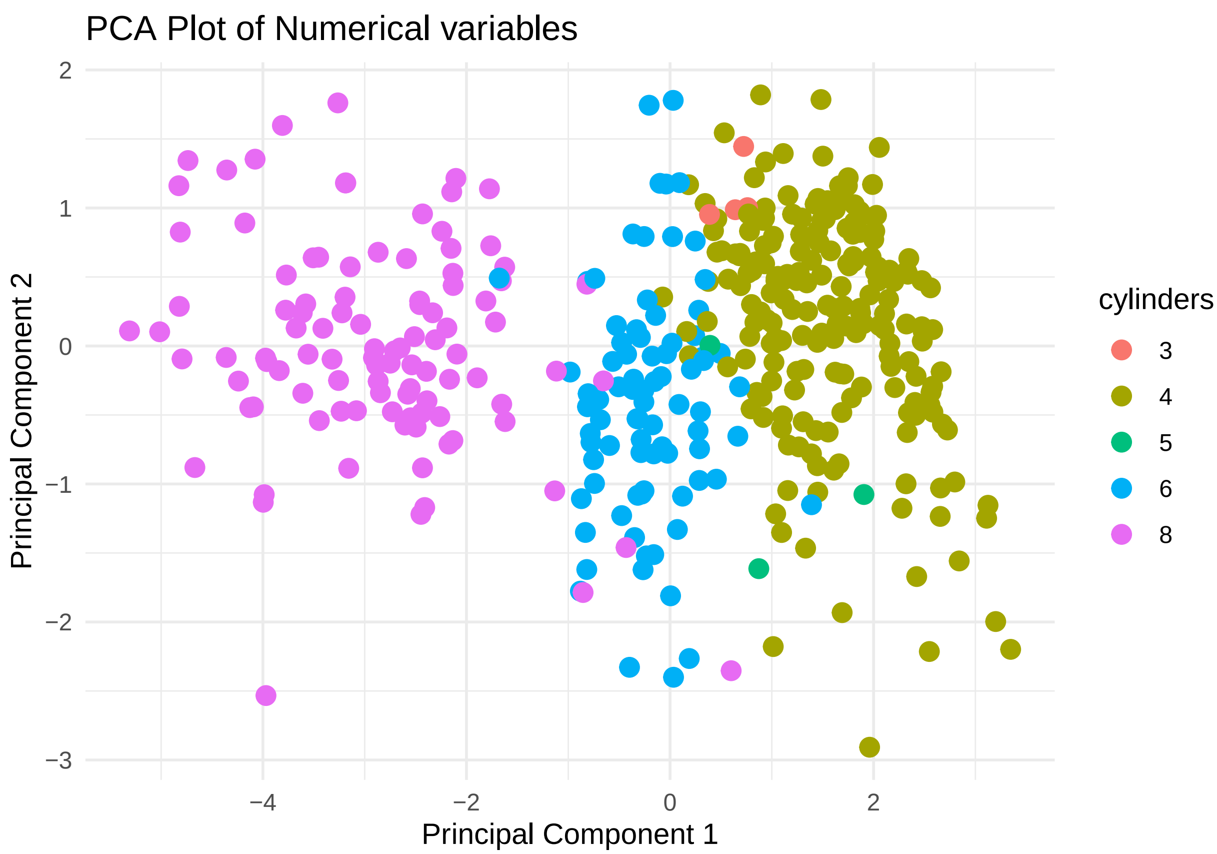


Laslty, lets's check whether the origin of the car, or the amount of cylinders cluster together. To analyse this, a principle component analysis was performed.

```
car_data_copy <- car_data
car_data_copy <- na.omit(car_data_copy)
pca_result <- prcomp(na.omit(car_data_copy[num_var]), scale. = TRUE, center = TRUE)
pca_df <- as.data.frame(pca_result$x)
pca_df <- cbind(pca_df, origin = car_data_copy$origin)

# Create the PCA plot with origin coloring
ggplot(pca_df, aes(x = PC1, y = PC2, color = origin)) +
  geom_point(size = 3) +
  labs(title = "PCA Plot of Numerical variables",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal()
```




```
pca_df <- cbind(pca_df, cylinders = car_data_copy$cylinders)
# Create the PCA plot with cylinder coloring
ggplot(pca_df, aes(x = PC1, y = PC2, color = cylinders)) +
  geom_point(size = 3) +
  labs(title = "PCA Plot of Numerical variables",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal()
```


PCA Plot of Numerical variables
Principal Component 2
Principal Component 1
2
1
0
−1
−2
−3
−4
−2
0
2
cylinders
3
4
5
6
8