Exercise 8

Tobias Raidl, 11717659

2023-12-18

```
library(dplyr)
```

```
## Warning: Paket 'dplyr' wurde unter R Version 4.3.2 erstellt

##
## Attache Paket: 'dplyr'

## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ISLR)
```

```
## Warning: Paket 'ISLR' wurde unter R Version 4.3.2 erstellt
```

```
library(splines)
data(Auto)

df = select(Auto, -name)
df_trans = df
df_trans[, 1] = log(df[, 1])
df_trans[, 3] = log(df[, 3])
df_trans[, 4] = log(df[, 4])
df_trans[, 5] = log(df[, 5])



set.seed(11717659)
sample <- sample(c(TRUE, FALSE), nrow(df), replace = TRUE, prob = c(0.7, 0.3))
train <- df[sample, ]
test <- df[!sample, ]
```

Initial

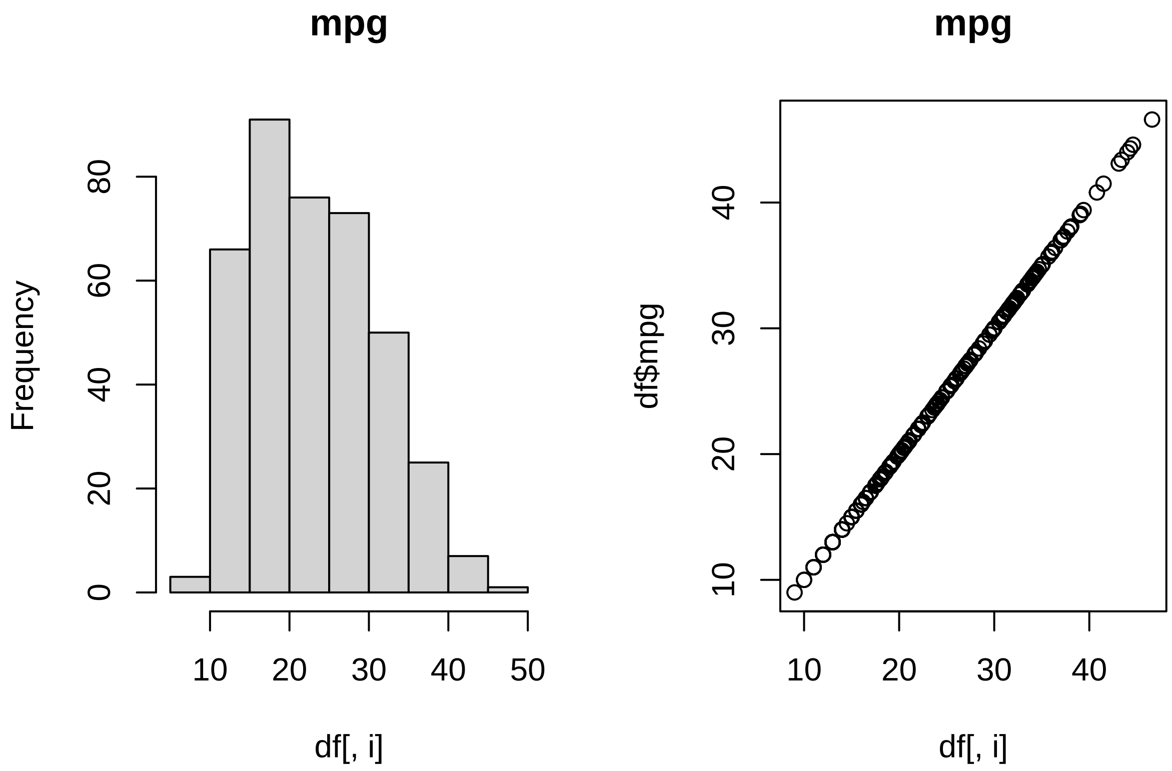
```r
library(ggplot2)
```

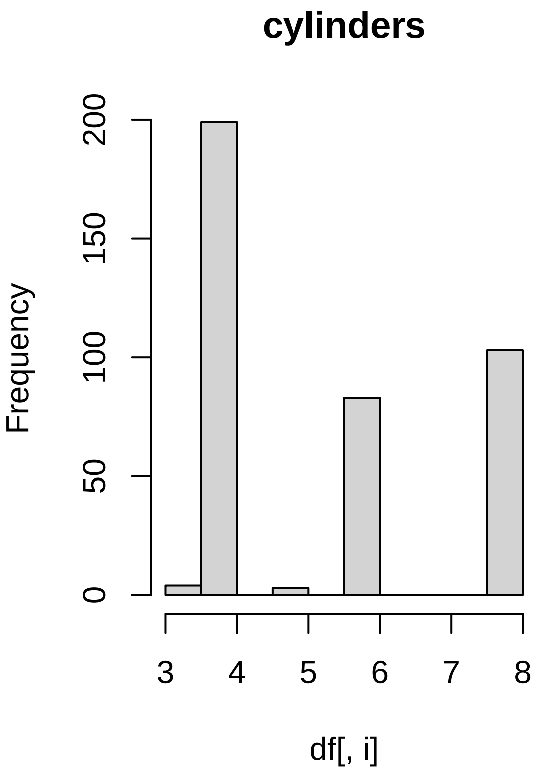
```
## Warning: Paket 'ggplot2' wurde unter R Version 4.3.2 erstellt
```

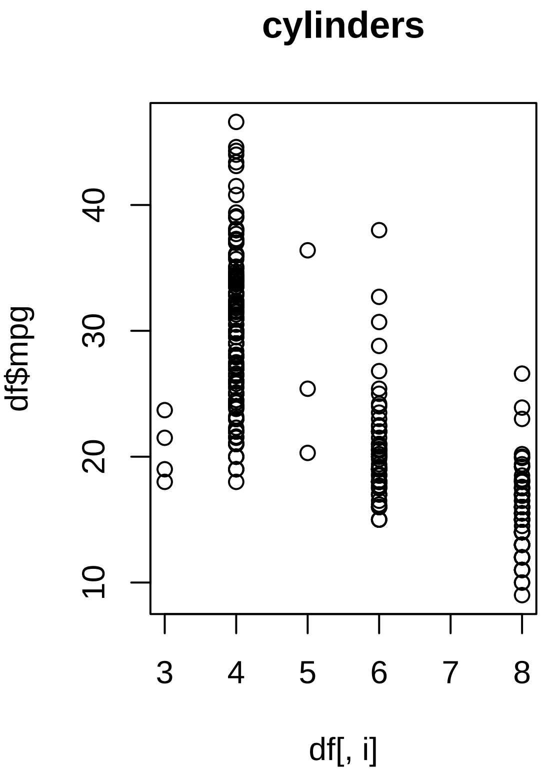
```r
library(tidyr)
```

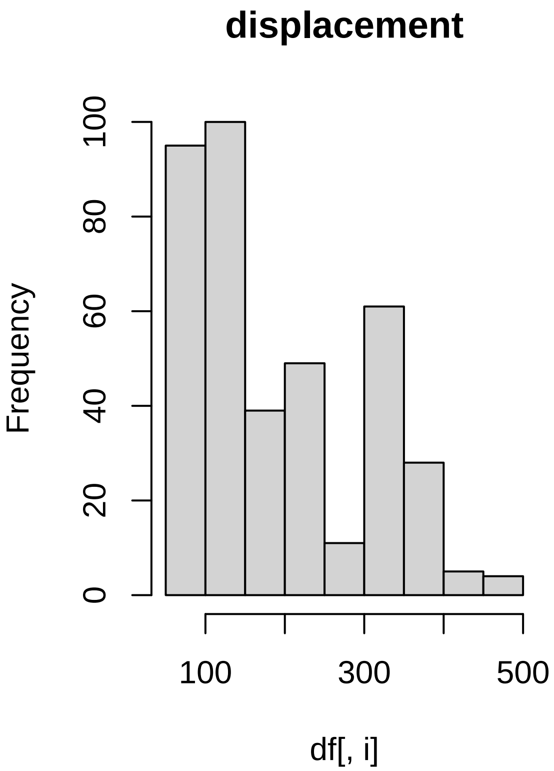
```
## Warning: Paket 'tidyr' wurde unter R Version 4.3.2 erstellt
```

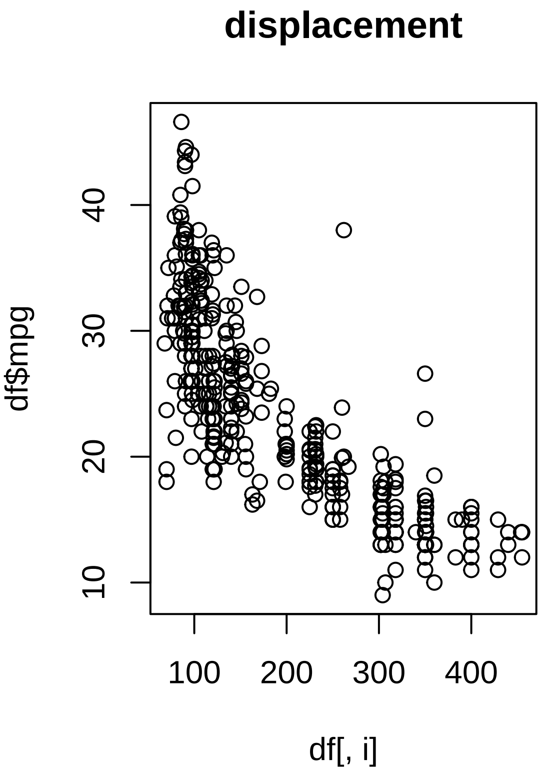
```r
for (i in 1:ncol(df)) {
    par(mfrow = c(1, 2))
    hist(df[, i], main = colnames(df)[i])
    plot(df[, i], df$mpg, main = colnames(df)[i])
}
```

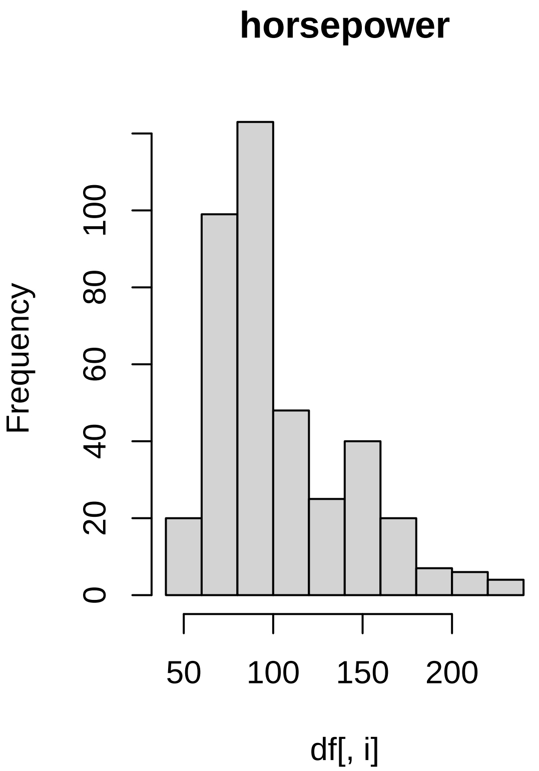


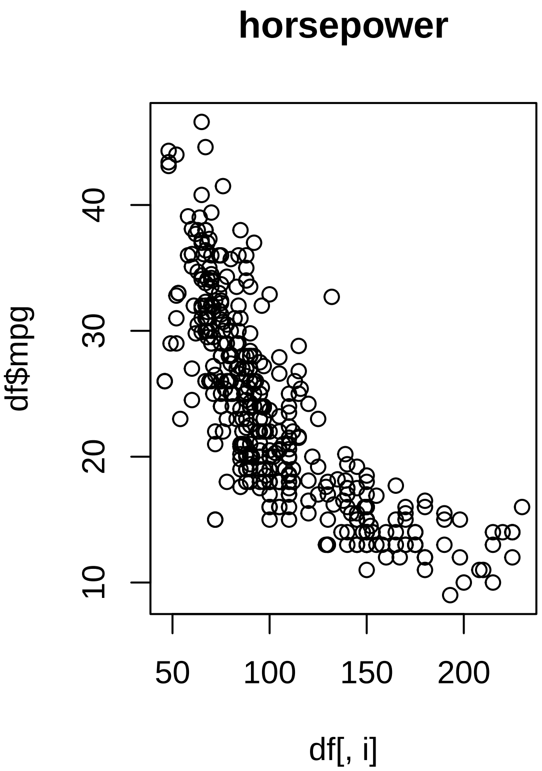

cylinders
Frequency
0
50
100
150
200
3
4
5
6
7
8
df[, i]

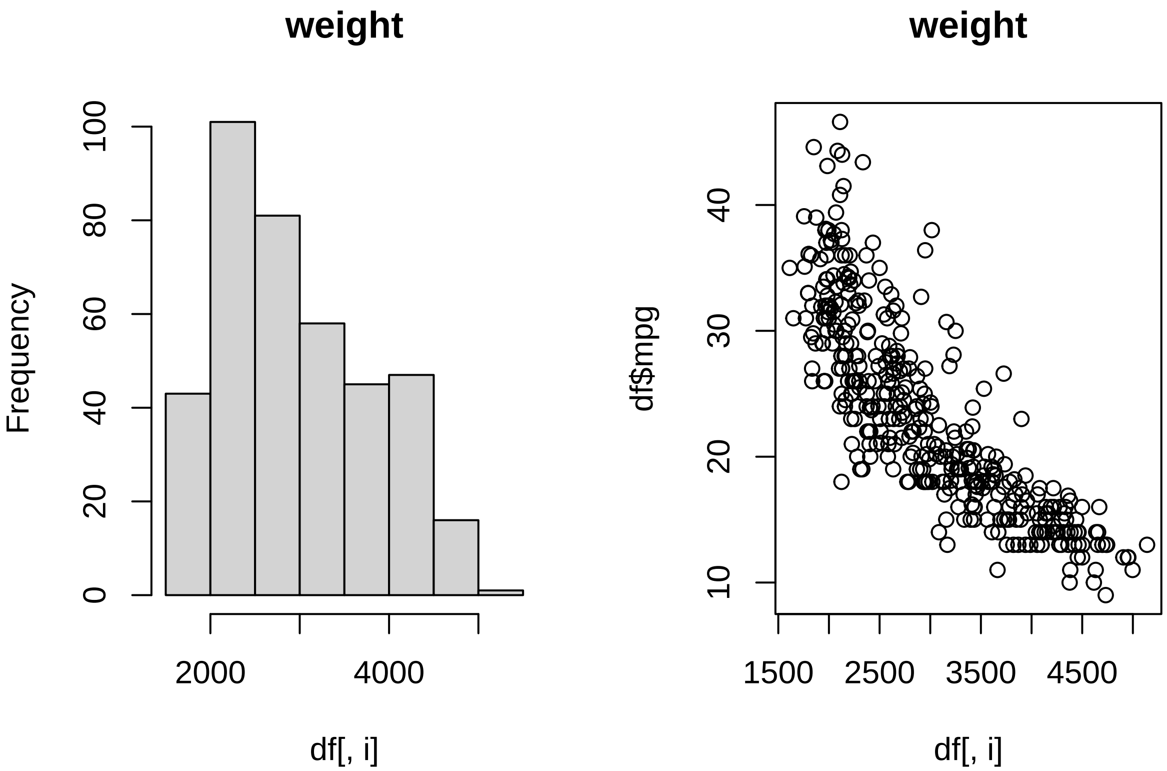

cylinders
df$mpg
10
20
30
40
3
4
5
6
7
8
df[, i]

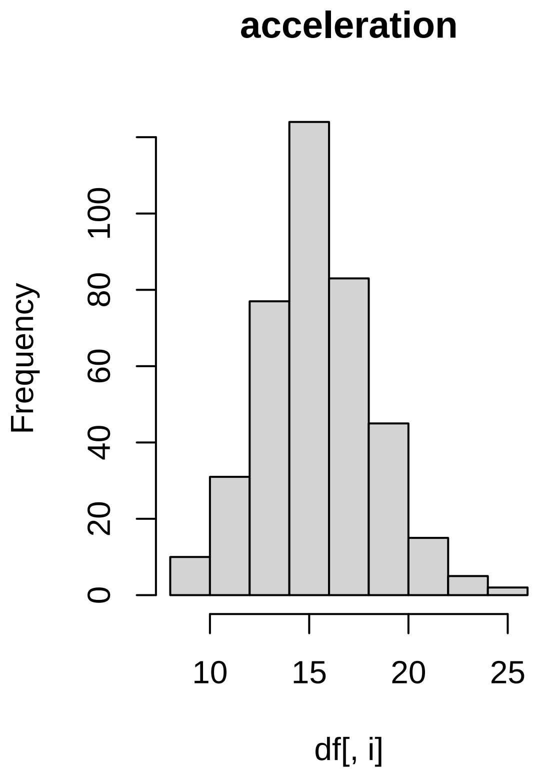

displacement
Frequency
0
20
40
60
80
100
100
300
500
df[, i]

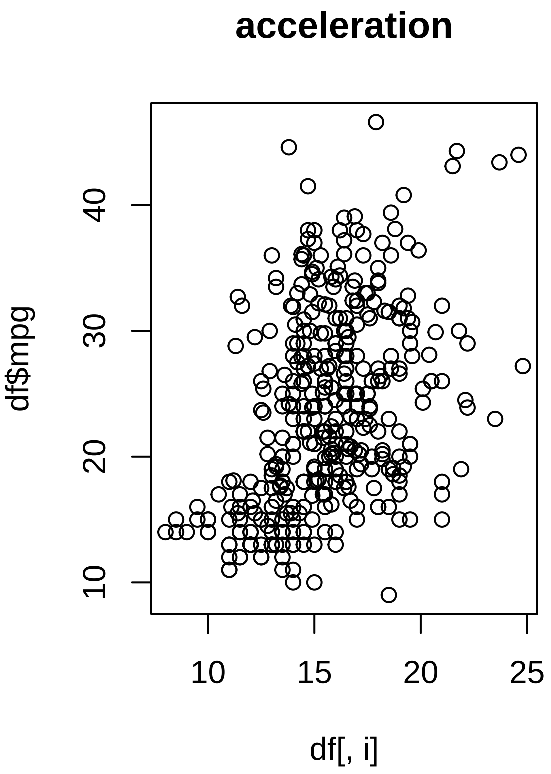

displacement
df$mpg
10
20
30
40
100
200
300
400
df[, i]

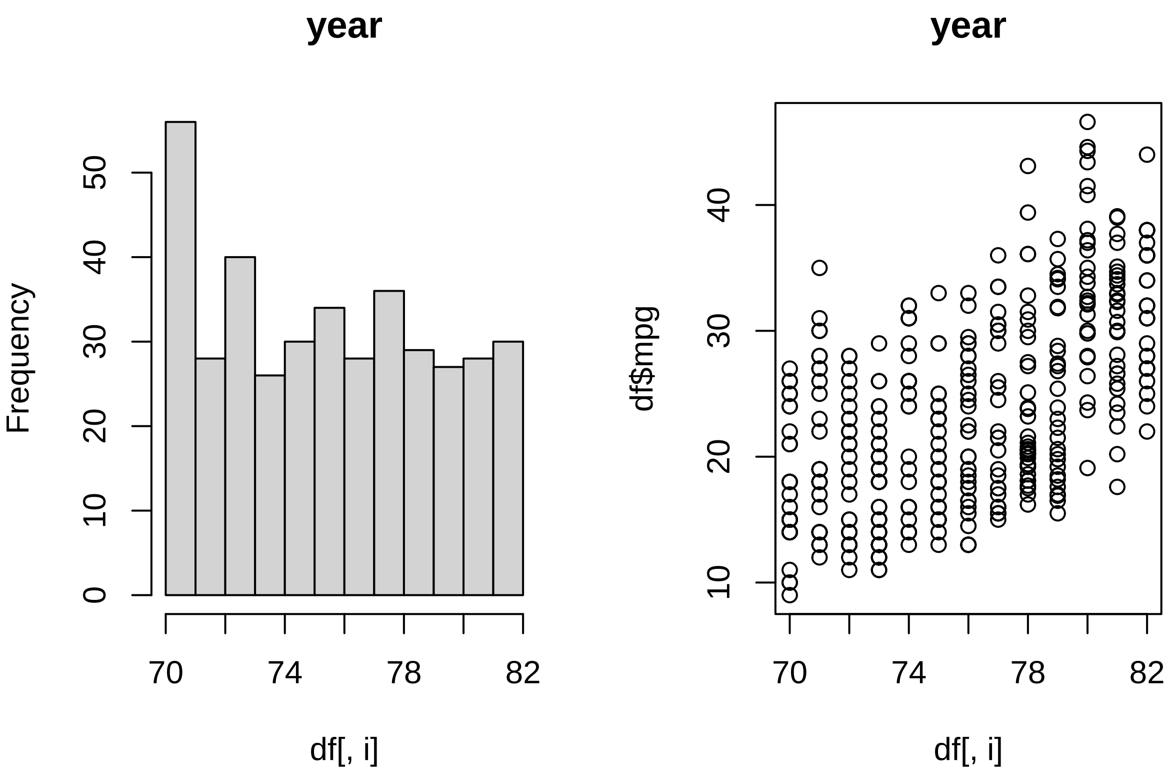

horsepower
Frequency
0
20
40
60
80
100
50
100
150
200
df[, i]

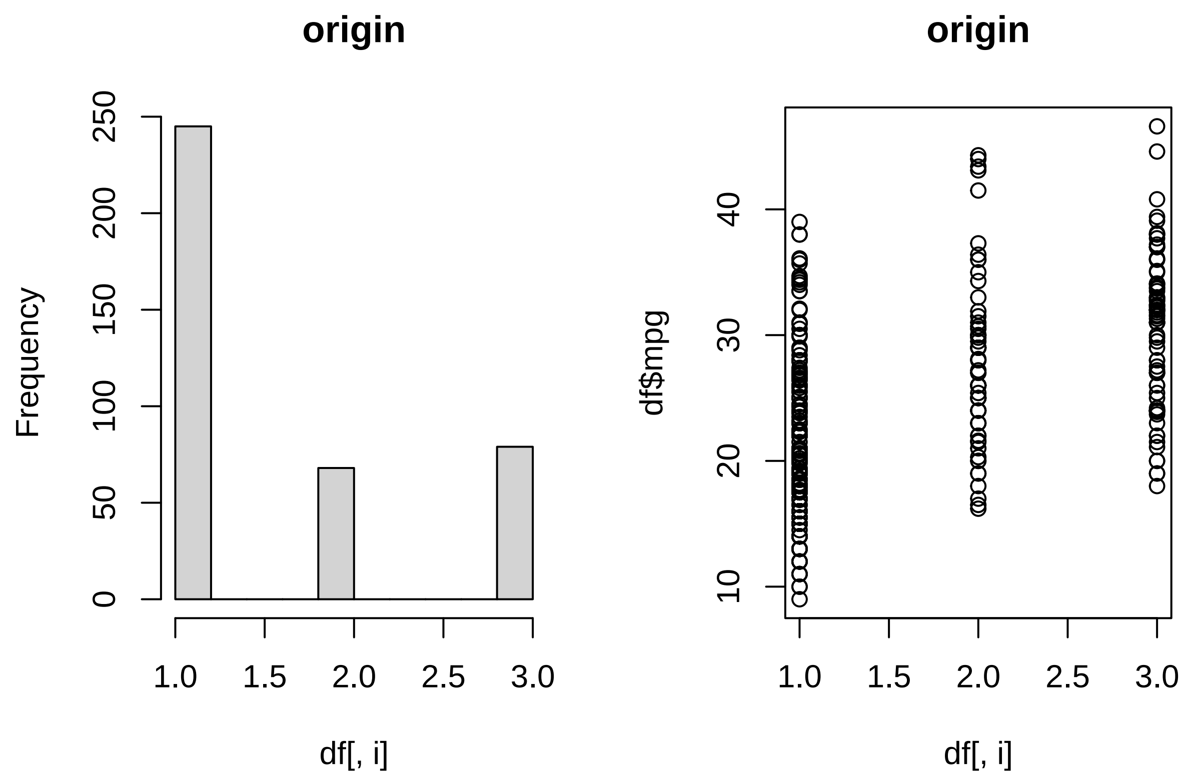

horsepower
df$mpg
10
20
30
40
50
100
150
200
df[, i]


weight
Frequency
0
20
40
60
80
100
2000
4000
df[, i]
weight
df$mpg
10
20
30
40
1500
2500
3500
4500
df[, i]


acceleration
Frequency
0
20
40
60
80
100
10
15
20
25
df[, i]

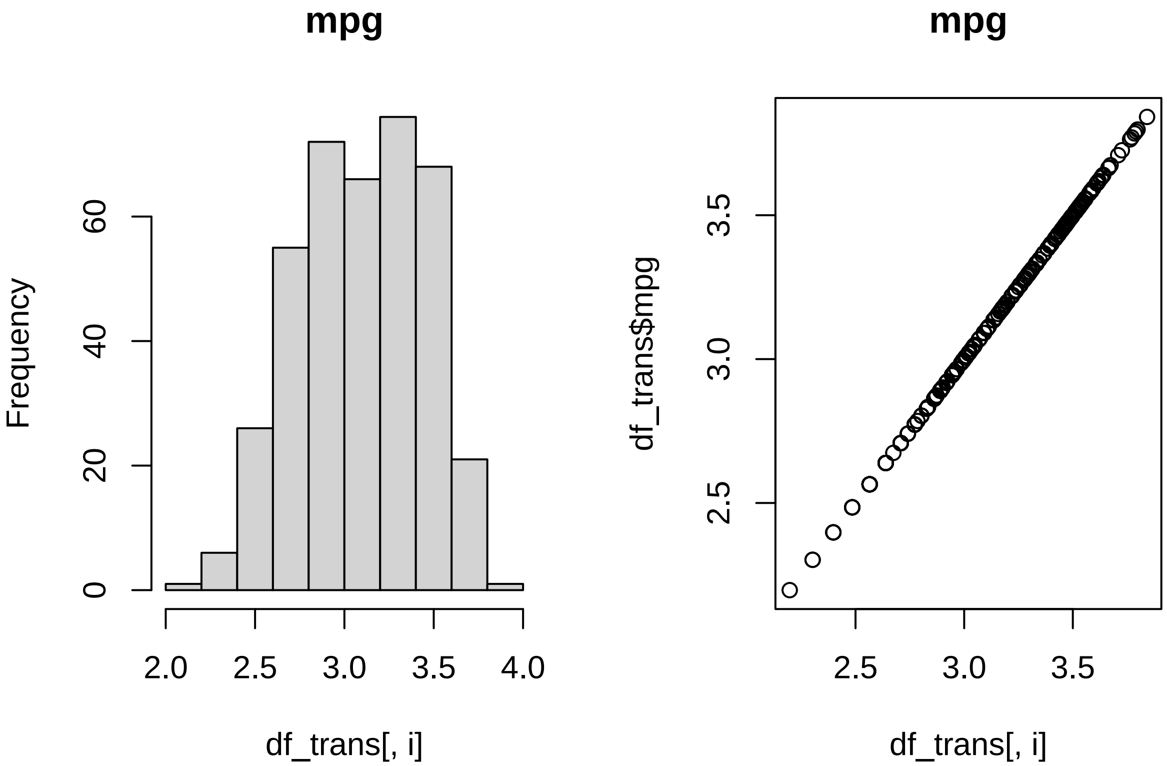

acceleration
df$mpg
10
20
30
40
10
15
20
25
df[, i]

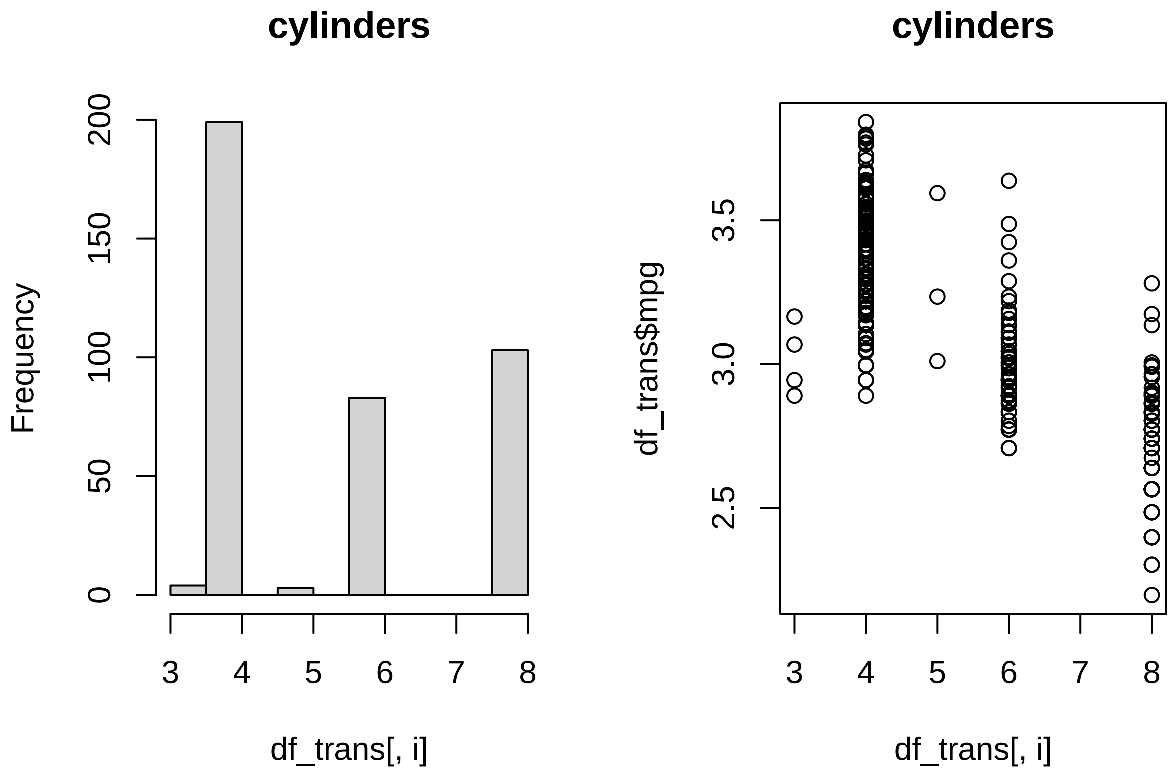

year
Frequency
0
10
20
30
40
50
70
74
78
82
df[, i]
year
df$mpg
10
20
30
40
70
74
78
82
df[, i]

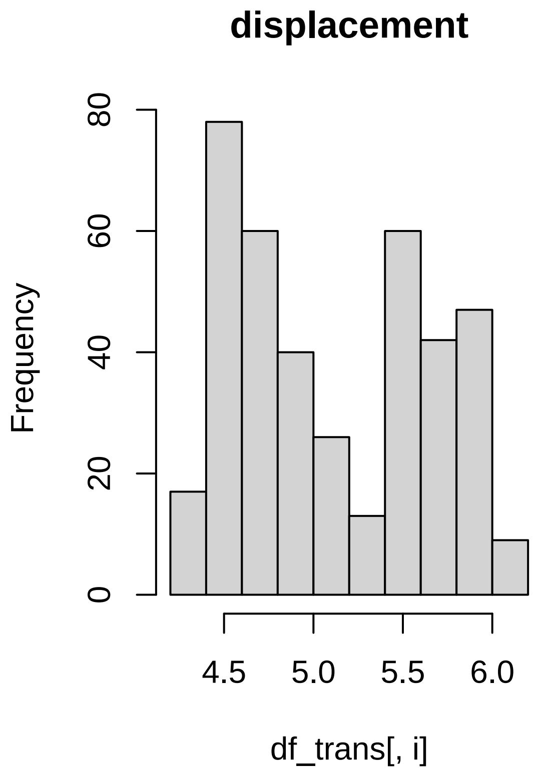


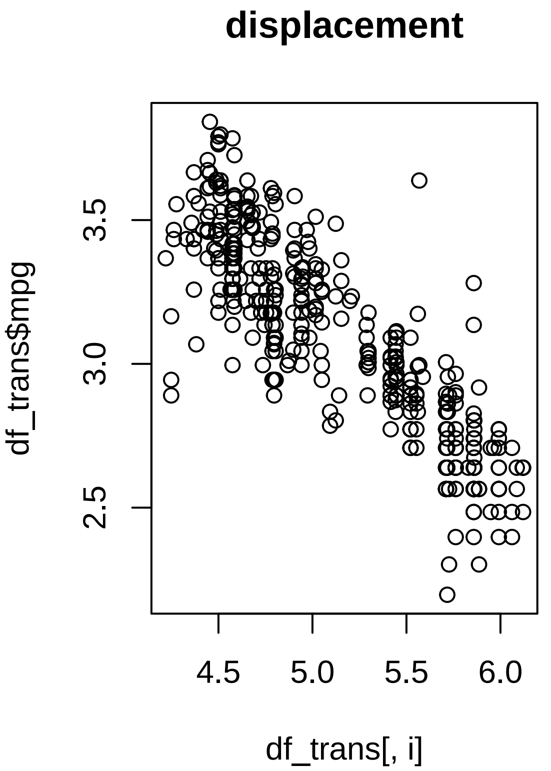
Transformed

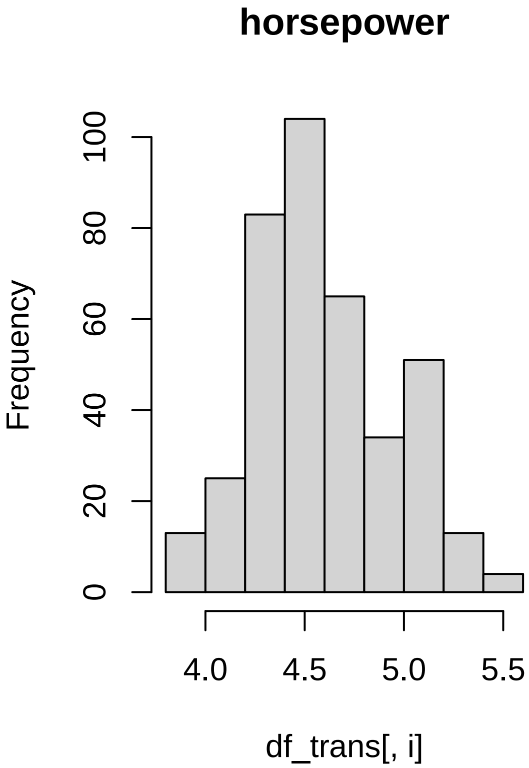
```
library(ggplot2)
library(tidyr)
for (i in 1:ncol(df_trans)) {
    par(mfrow = c(1, 2))
    hist(df_trans[, i], main = colnames(df_trans)[i])
    plot(df_trans[, i], df_trans$mpg, main = colnames(df_trans)[i])
}
```

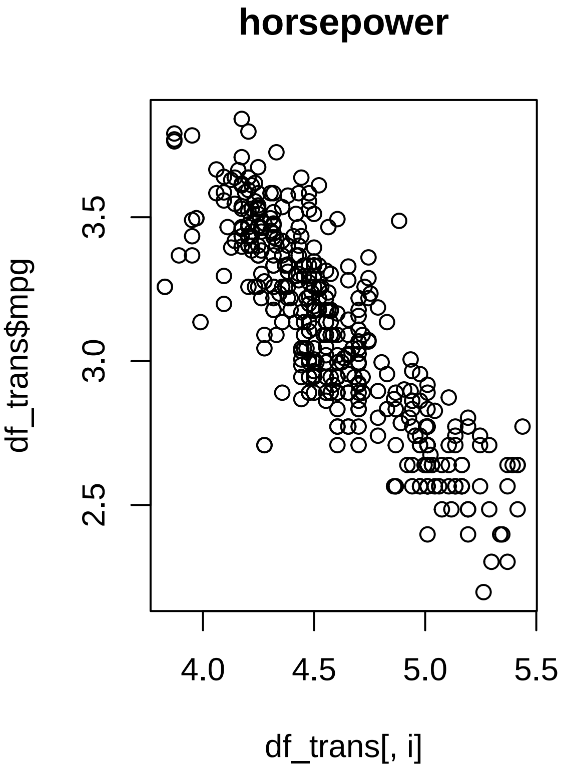

mpg
Frequency
0
20
40
60
2.0
2.5
3.0
3.5
4.0
df_trans[, i]
mpg
df_trans$mpg
2.5
3.0
3.5
2.5
3.0
3.5
df_trans[, i]

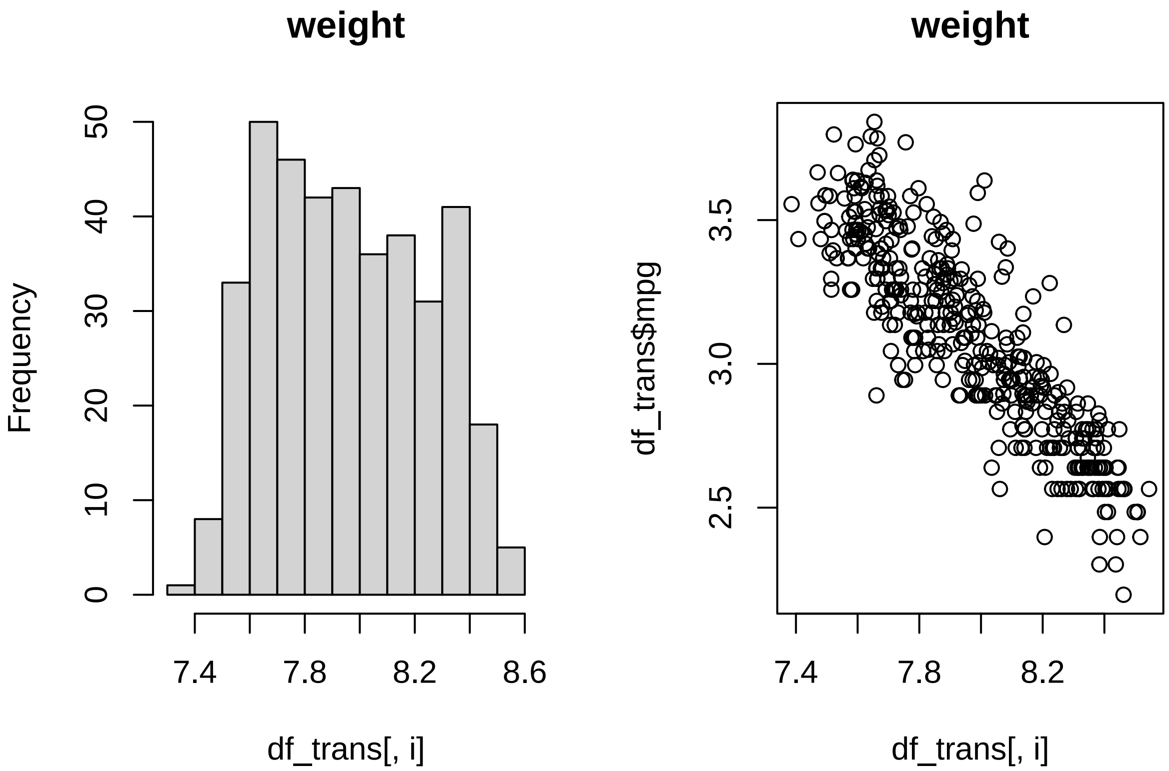

cylinders
Frequency
0
50
100
150
200
3
4
5
6
7
8
df_trans[, i]
cylinders
df_trans$mpg
2.5
3.0
3.5
3
4
5
6
7
8
df_trans[, i]

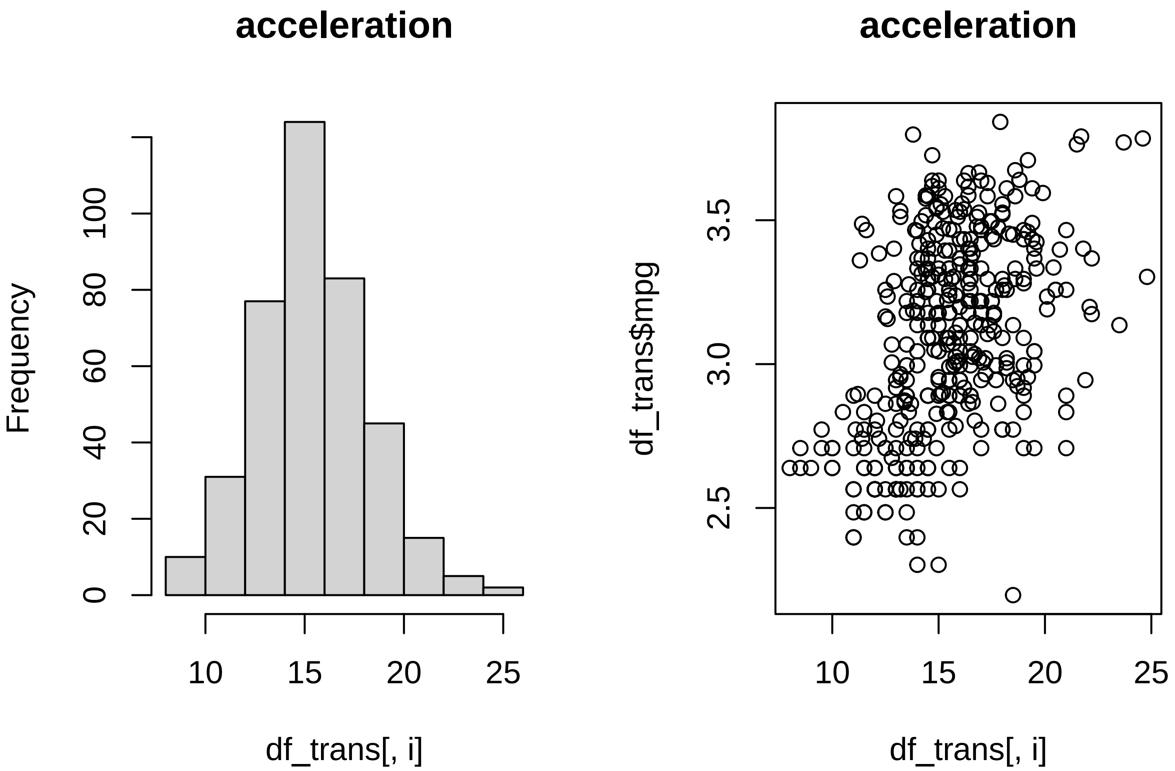

displacement
Frequency
0
20
40
60
80
4.5
5.0
5.5
6.0
df_trans[, i]

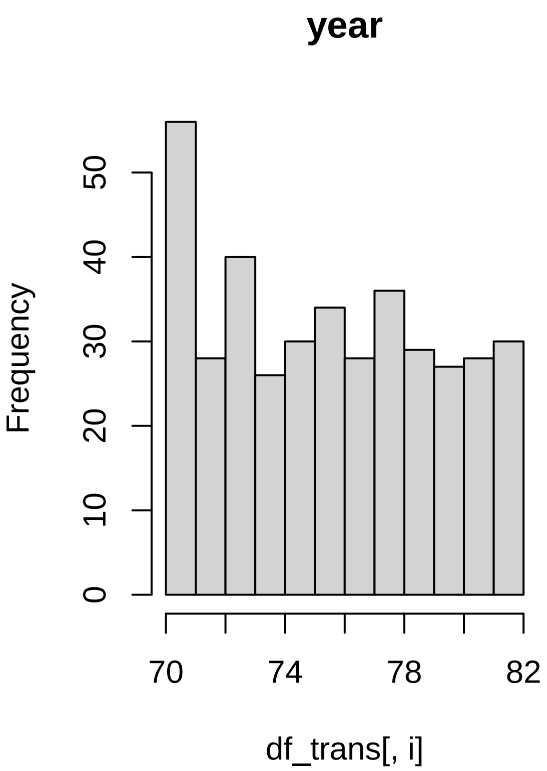

displacement
df_trans$mpg
2.5
3.0
3.5
4.5
5.0
5.5
6.0
df_trans[, i]

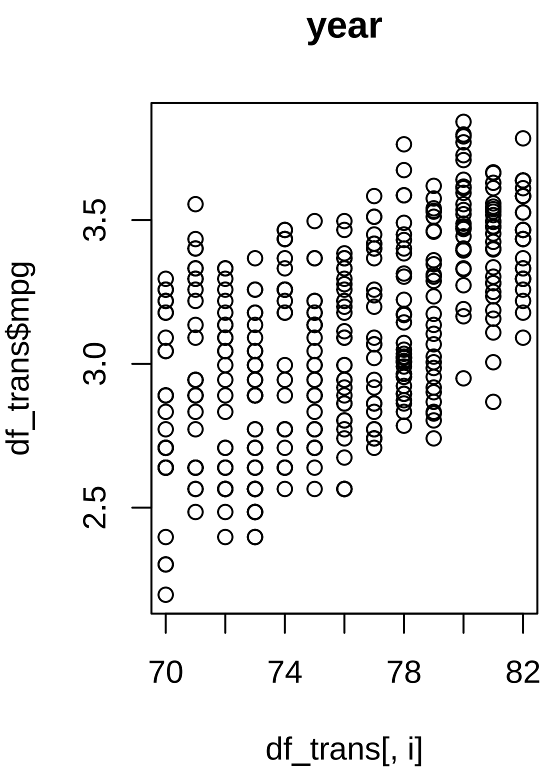

horsepower
Frequency
0
20
40
60
80
100
4.0
4.5
5.0
5.5
df_trans[, i]

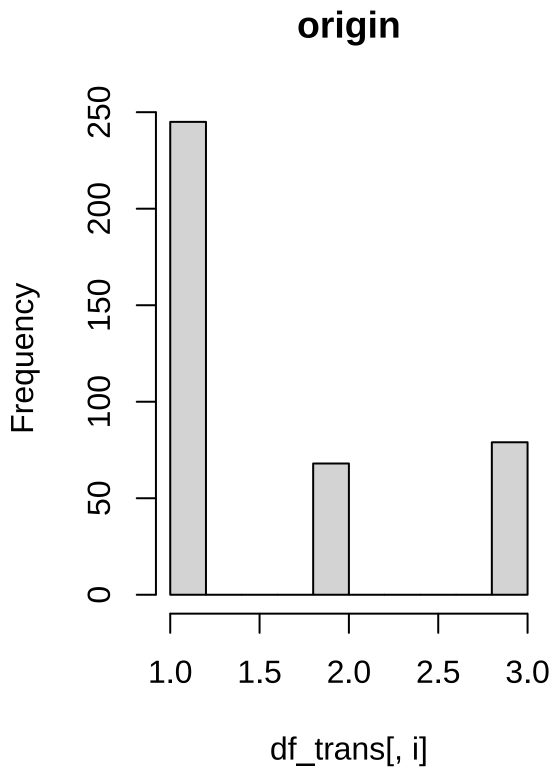

horsepower
df_trans$mpg
2.5
3.0
3.5
4.0
4.5
5.0
5.5
df_trans[, i]

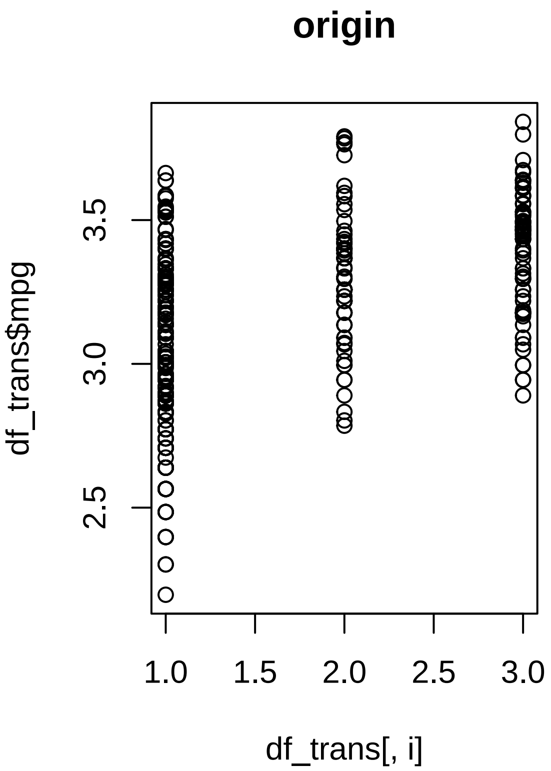

weight
Frequency
0
10
20
30
40
50
7.4
7.8
8.2
8.6
df_trans[, i]
weight
df_trans$mpg
2.5
3.0
3.5
7.4
7.8
8.2
df_trans[, i]


acceleration
Frequency
0
20
40
60
80
100
10
15
20
25
df_trans[, i]
acceleration
df_trans$mpg
2.5
3.0
3.5
10
15
20
25
df_trans[, i]


year
Frequency
0
10
20
30
40
50
70
74
78
82
df_trans[, i]


year
df_trans$mpg
2.5
3.0
3.5
70
74
78
82
df_trans[, i]





RMSE function

```r
get_rmse = function(y_pred, y_gt) {
    return(sqrt(mean((y_pred - y_gt)^2)))
}
```

1

Cylinders, year and origins should be treated as factors. Using log transformation all variables could be represented linearly. Not using log transform displacement, horsepower, weight, cylinders and year should be represented by splines.

2 & 3

```r
degfree = 2
lin_model = lm(mpg ~ ., data = train)
spl_model = lm(mpg ~ (ns(displacement, df = degfree) + ns(horsepower, df = degfree) +
    ns(weight, df = degfree) + ns(cylinders, df = degfree) + ns(year, df = degfree)),
    data = train)
summary(spl_model)
```

```
##
## Call:
## lm(formula = mpg ~ (ns(displacement, df = degfree) + ns(horsepower,
##     df = degfree) + ns(weight, df = degfree) + ns(cylinders,
##     df = degfree) + ns(year, df = degfree)), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9958 -1.5826 -0.0154  1.3109 11.8227
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      32.8342     1.1943  27.492  < 2e-16 ***
## ns(displacement, df = degfree)1  -7.4566     3.9936  -1.867  0.06300 .
## ns(displacement, df = degfree)2   1.7562     2.6124   0.672  0.50202
## ns(horsepower, df = degfree)1   -13.4381     2.7779  -4.838 2.25e-06 ***
## ns(horsepower, df = degfree)2    -6.6600     2.0873  -3.191  0.00159 **
## ns(weight, df = degfree)1       -17.5720     3.5293  -4.979 1.16e-06 ***
## ns(weight, df = degfree)2        -8.7383     1.8540  -4.713 3.96e-06 ***
## ns(cylinders, df = degfree)1      3.4320     3.1731   1.082  0.28043
## ns(cylinders, df = degfree)2      1.2350     1.5485   0.798  0.42587
## ns(year, df = degfree)1           6.9496     1.1528   6.029 5.60e-09 ***
## ns(year, df = degfree)2           9.0863     0.5933  15.315  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.906 on 262 degrees of freedom
## Multiple R-squared:  0.8707, Adjusted R-squared:  0.8658
## F-statistic: 176.4 on 10 and 262 DF,  p-value: < 2.2e-16
```

5 & 6

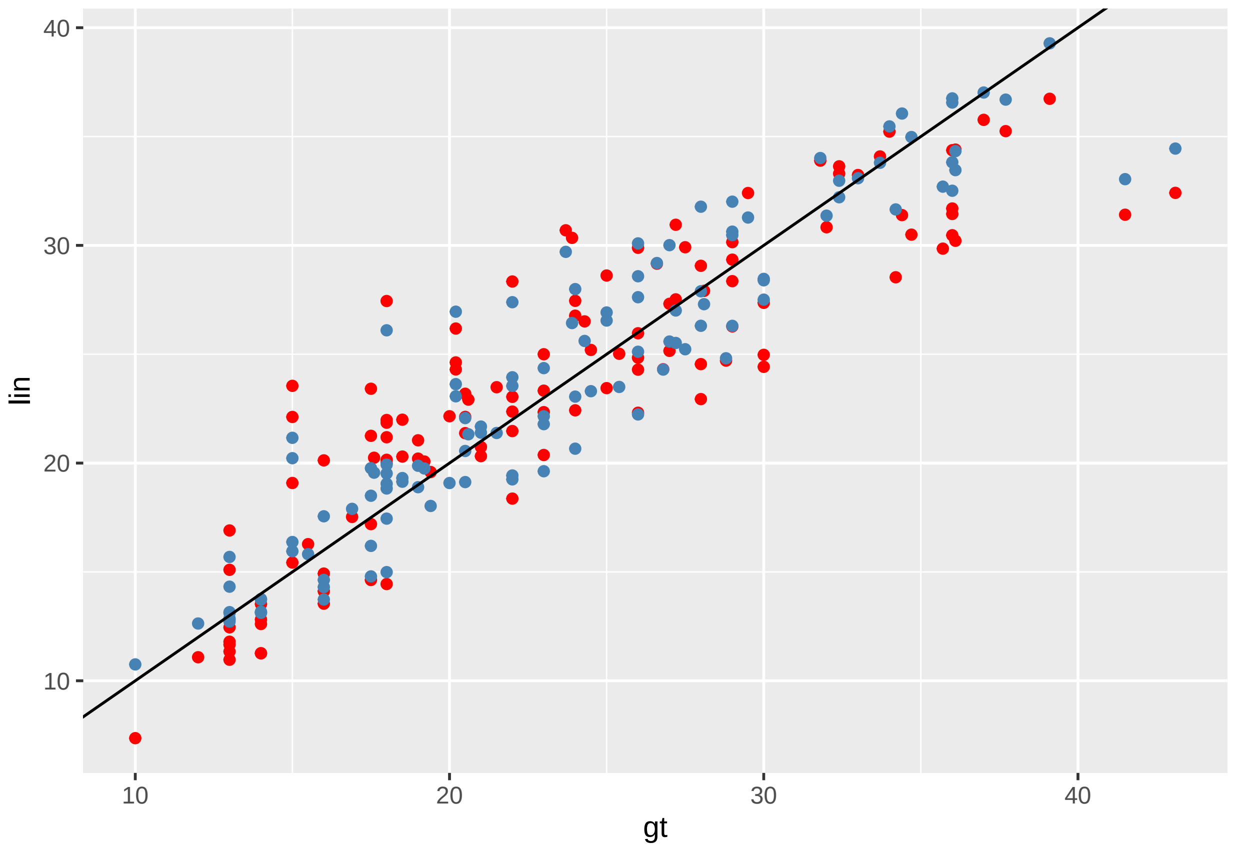
```r
y_pred_lin = predict(lin_model, test)
y_pred_spl = predict(spl_model, test)
get_rmse(y_pred_lin, test$mpg)
```

```
## [1] 3.440262
```

```r
get_rmse(y_pred_spl, test$mpg)
```

```
## [1] 2.532837
```

```r
preds = data.frame(lin = y_pred_lin, spl = y_pred_spl, gt = test$mpg)
ggplot() + geom_point(data = preds, aes(gt, lin), col = "red") + geom_point(data = preds,
    aes(gt, spl), col = "steelblue") + geom_abline(intercept = 0, slope = 1)
```


40
30
20
10
lin
10
20
30
40
gt