lab4

shichenh

9/25/2017

Mean Centering X

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
reg1 <- lm(mpg ~ disp, data = mtcars)

mtcars.mean <- data.frame(scale(mtcars, scale = FALSE))

reg2 <- lm(mpg ~ disp, data = mtcars.mean)

print(reg1)
```

```
##
## Call:
## lm(formula = mpg ~ disp, data = mtcars)
##
## Coefficients:
## (Intercept)         disp
##    29.59985     -0.04122
```

```
# Recovering Beta0
beta0 <- mean(mtcars$mpg) - reg2$coefficients[2] * mean(mtcars$disp)
beta0 - reg1$coefficients[1]
```

```
##         disp
## 7.105427e-15
```

Standardizing X

```
#setting up the variables
mu.y = mean(mtcars$mpg)
mu.x = mean(mtcars$disp)
sd.y = sd(mtcars$mpg)
sd.x = sd(mtcars$disp)
```

```r
mtcars.std <- scale(mtcars)
reg3 <- lm(mpg ~ disp, data = data.frame(mtcars.std))

beta0.from.sd <- mu.y - sd.y/sd.x*reg3$coefficients[2]*mu.x
beta1.from.sd <- sd.y/sd.x * reg3$coefficients[2]
c(beta0.from.sd, beta1.from.sd) - reg1$coefficients
```

```
##          disp          disp
##  7.105427e-15 -6.938894e-18
```

To fit a linear model with intercept, use ‘lm(y ~ x + 0’)

```r
lm(mpg~disp + 0, data = data.frame(mtcars.std))
```

```
##
## Call:
## lm(formula = mpg ~ disp + 0, data = data.frame(mtcars.std))
##
## Coefficients:
##    disp
## -0.8476
```

Fitting a linear model with a subset of data

```r
lm(mpg ~ disp, data = subset(mtcars, am  == 1))
```

```
##
## Call:
## lm(formula = mpg ~ disp, data = subset(mtcars, am == 1))
##
## Coefficients:
## (Intercept)         disp
##    32.86614     -0.05904
```

Summary of lm

```r
summary(reg1)
```

```
##
## Call:
## lm(formula = mpg ~ disp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8922 -2.2022 -0.9631  1.6272  7.2305
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.599855   1.229720  24.070  < 2e-16 ***
## disp        -0.041215   0.004712  -8.747 9.38e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.251 on 30 degrees of freedom
## Multiple R-squared:  0.7183, Adjusted R-squared:  0.709
## F-statistic: 76.51 on 1 and 30 DF,  p-value: 9.38e-10
```

```r
typeof(reg1)
```

```
## [1] "list"
```

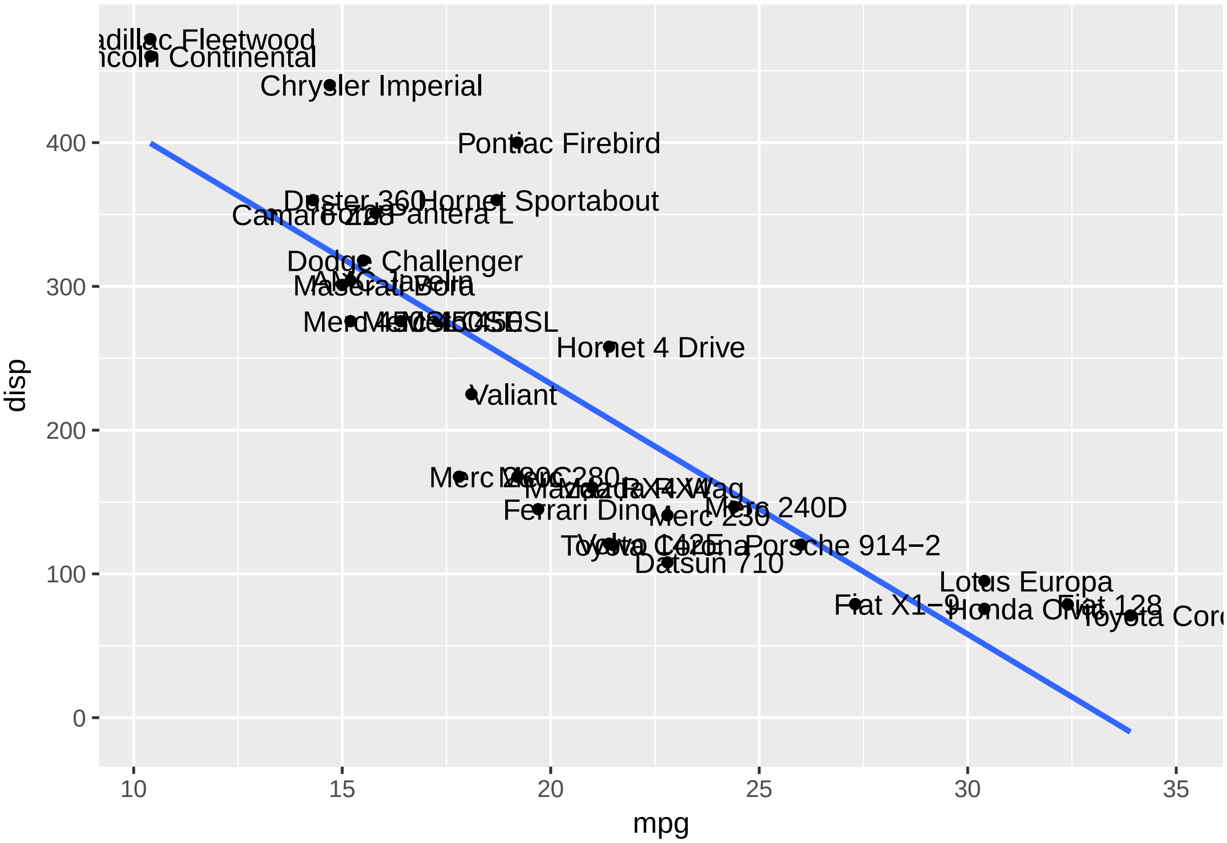
```r
#it is a list

names(reg1)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

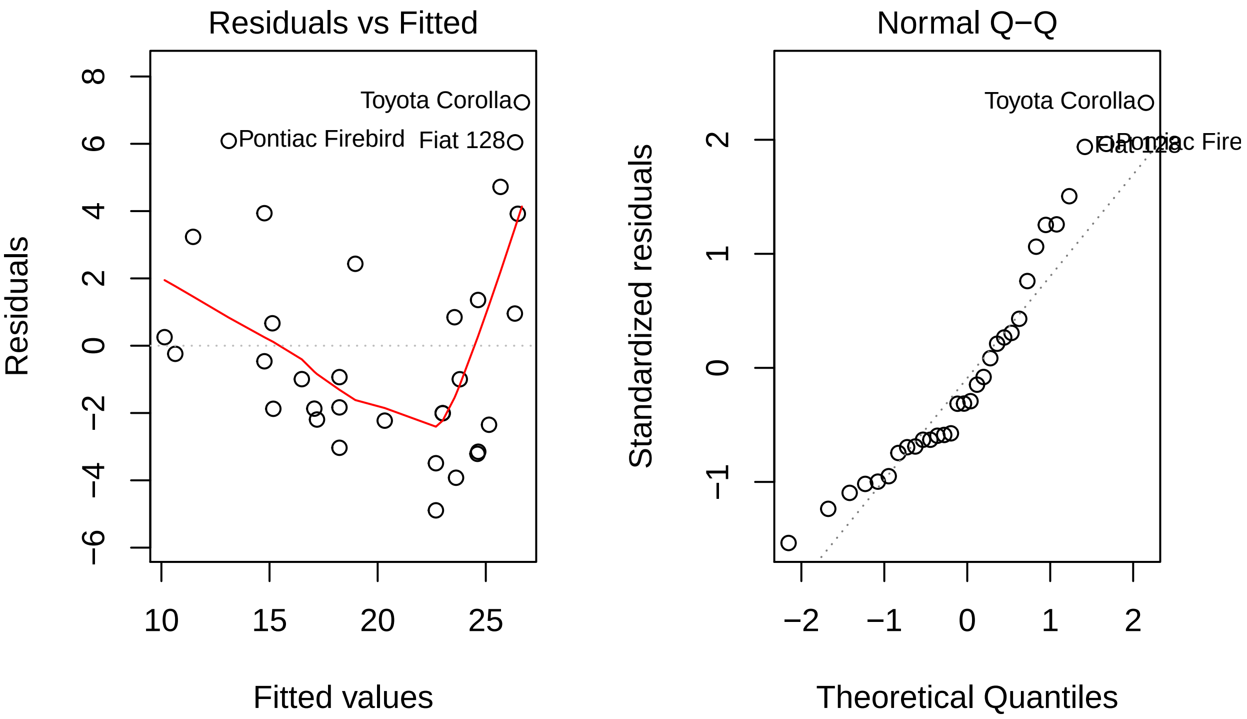
Plotting the regression line

```r
ggplot(mtcars, aes(x = mpg, y = disp)) +
  geom_point() +
  stat_smooth(method = "lm", se = F) +
  geom_text(aes(x = mpg, y = disp, label = rownames(mtcars)), nudge_x = 1)
```



## Plots checking linear fit

```r
par(mfrow=c(1, 2))
plot(reg1, which = 1)
plot(reg1, which = 2)
```



Thre residuals are not normal at all. The linear fit is not so good.

Using QR for lm

```
qr <- qr(cbind(mtcars$mpg, 1))
q <- qr.Q(qr)
r <- qr.R(qr)

#f <- t(q %*% t(matrix(mtcars$disp)))
#solve(r, f)
```