MATH3431- Practical Class Sheets 1

魏上傑

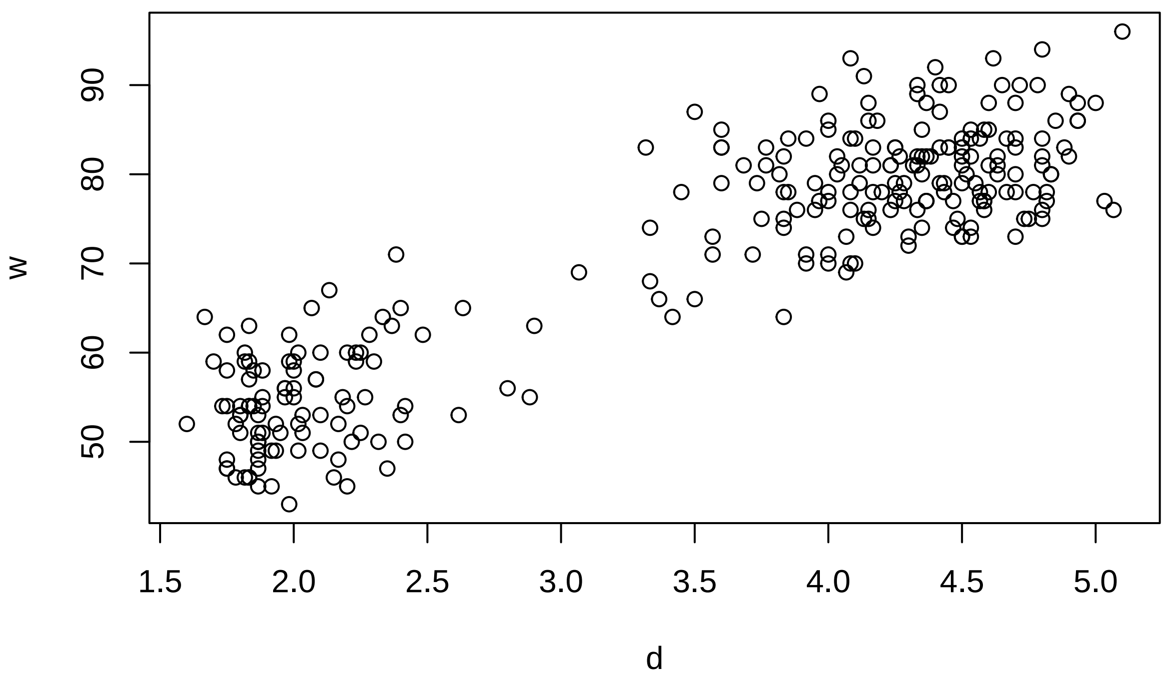
2023-03-23

1.1 Observing the faithful data set

```
# load the faithful data set
data("faithful")

w <- faithful$waiting
d <- faithful$eruptions
```

```
plot(x= d, y=w)
```



```r
# Pearson correlation coefficient
cor(x=w, y=d, method = "pearson")
```

```
## [1] 0.9008112
```

Obviously, there is evidence of a linear relationship. Let's consider fitting a linear regression model.

1.2 Fitting simple linear regression

```r
model <- lm(w~d, data = faithful)
model
```

```
##
## Call:
## lm(formula = w ~ d, data = faithful)
##
## Coefficients:
## (Intercept)            d
##       33.47        10.73
```

1.3

```r
coef(model)
```

```
## (Intercept)           d
##    33.47440    10.72964
```

```r
beta1hat <- coef(model)[2]
beta1hat
```

```
##        d
## 10.72964
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.1.0
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.4      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
lsq.Q <- resid(model)^2 %>% sum()
lsq.Q
```

```
## [1] 9443.387
```

1.4 The regression summary

```r
summ <- summary(model)
```

```r
summ$coefficients
```

```
##             Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 33.47440  1.1548735 28.98534  7.136015e-85
## d           10.72964  0.3147534 34.08904 8.129959e-100
```

Small p-value indicates significant coefficients!!

```r
# regression standard error
se <- summ$sigma
se
```

```
## [1] 5.914009
```

Let's see the relationship between R-squared and Pearson correlation coefficient.

```r
rsq <- summ$r.squared
rsq
```

```
## [1] 0.8114608
```

```r
cor(w,d)^2
```

```
## [1] 0.8114608
```

```r
# same as resid(model)
summ$residuals %>% head()
```

```
##         1         2         3         4         5         6
##  6.898894  1.212248  4.763708  4.029832  2.888139 -9.407953
```

1.5 Inference on the coefficients

```
summm$coefficients
```

```
##              Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 33.47440  1.1548735 28.98534  7.136015e-85
## d           10.72964  0.3147534 34.08904 8.129959e-100
```

```
se.beta1 <- summm$coefficients[2,2]
se.beta1
```

```
## [1] 0.3147534
```

```
# H_0: beta_1=0
# calculate t-statistics
t.beta1 <- (beta1hat-0)/se.beta1
t.beta1
```

```
##        d
## 34.08904
```

```
t.beta1 <- unname(t.beta1)
t.beta1
```

```
## [1] 34.08904
```

```
# use the pt function to find p-value
n <- length(w)
2*(1-pt(t.beta1, df=n-2))  # df=n-2
```

```
## [1] 0
```

```
# pt function will give the cdf of t distribution
?pt
```

```
# use the qt function to find confindence interval
# qt function will give the quantile of t distribution
beta1hat+c(-1,1)*qt(0.95,n-2)*se.beta1
```

```
## [1] 10.21014 11.24915
```

```r
# use confint to do the same thing
confint(model, level = 0.95)
```

```
##                2.5 %   97.5 %
## (Intercept) 31.20069 35.74810
## d           10.10996 11.34932
```

```r
# for the slope parameter
confint(model, parm = "d", level = 0.95)
```

```
##      2.5 %   97.5 %
## d 10.10996 11.34932
```

```r
# or
confint(model, level = 0.95)[2,]
```

```
##    2.5 %   97.5 %
## 10.10996 11.34932
```

1.6 Estimation and prediction

```r
# confidence interval
# this simply concerns
# the location of the regression line
newdata1 <- data.frame(d=3)
predict(model, newdata = newdata1, interval = "confidence", level=0.95)
```

```
##        fit      lwr     upr
## 1 65.66332 64.89535 66.4313
```

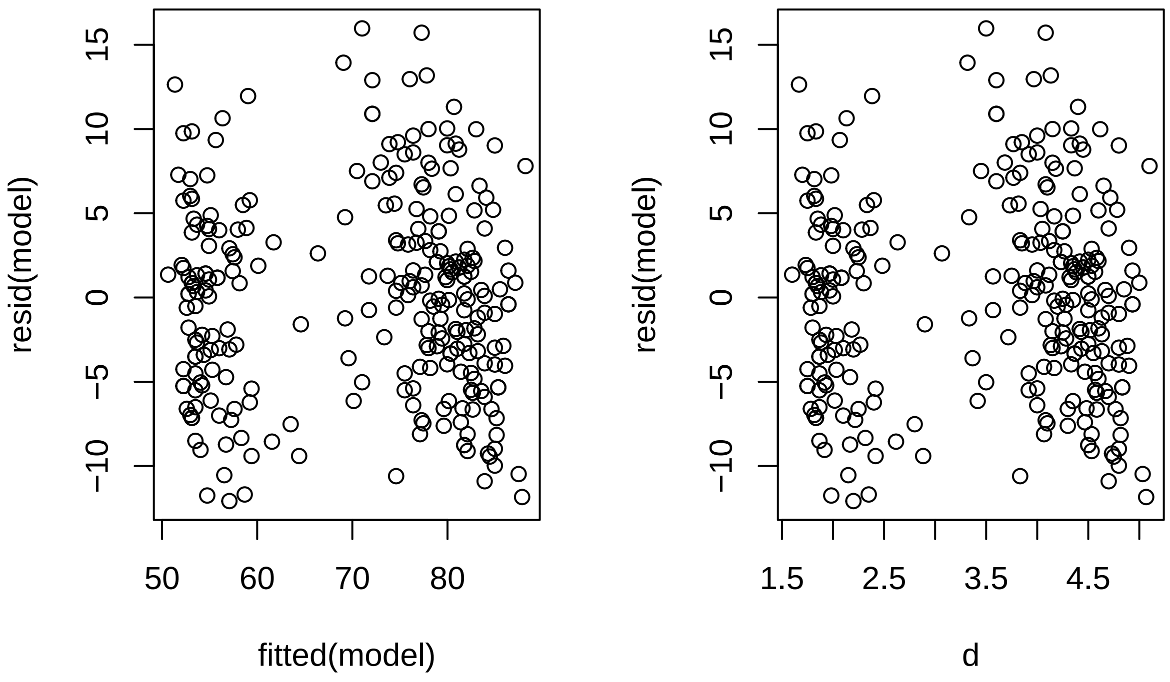
```r
# prediction interval
# This concerns both the location of the regression line
# and the regression error about that point
predict(model, newdata = newdata1, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 65.66332 53.99458 77.33206
```

1.7 Residual Analysis

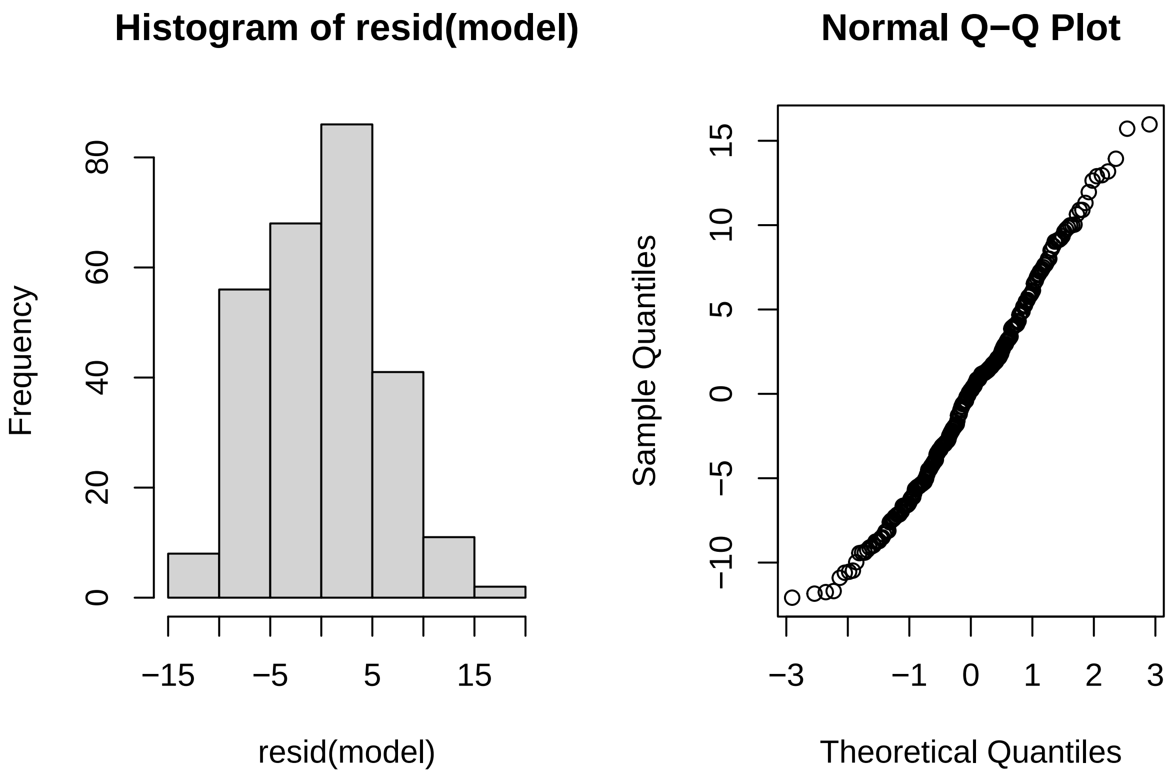
What do you want to see from the residual plots? Well, Nothing!!

```
par(mfrow=c(1,2))
plot(y=resid(model), x=fitted(model))
plot(y=resid(model), x=d)
```



1.8

```
par(mfrow=c(1,2))
hist(resid(model))
qqnorm(resid(model))
```



From either the QQ plot or the histogram, we can find that the Normal assumption holds.