R_MSE

Xiaojia Zhang

2/3/2021

Simple Linear Regression

```
#independent variable
x <- 1:20

#for reproducibility
set.seed(1)

#dependent variable; function of x with random error
y <- 2 + 0.5*x + rnorm(20,0,3)

#simple linear regression
mod <- lm(y~x)

#r-sqaure
summary(mod)$r.squared
```

```
## [1] 0.6026682
```

R-Square

The sum of squared fitted-value deviations divided by the sum of squared original-value deviations.

$R^2 = \frac{\sum(\hat{y}-\overline{\hat{y}})^2}{\sum(y-\overline{y})^2}$

$MSS = \sum(\hat{y} - \overline{\hat{y}})$

$TSS = \sum(y - \overline{y})^2$

```
# extract fitted (or predicted) values from model
y_pred <- mod$fitted.values

# sum of squared fitted-value deviations
mss <- sum((y_pred - mean(y_pred))^2)

# sum of squared original-value deviations
tss <- sum((y - mean(y))^2)

# r-squared
mss/tss
```

```
## [1] 0.6026682
```

1. R-Squared says nothing about prediction error, even with variance exactly the same, and no change in the coefficient.

```
# range from 1~10
set.seed(1)
x <- seq(1,10,length.out = 100)
y <- 2+1.2*x+rnorm(100,0,sd=0.9)
mod1 <- lm(y~x)

# r-squared
print(paste('The R^2 is: ', summary(mod1)$r.squared))
```

```
## [1] "The R^2 is:  0.938337867294955"
```

```
# Mean Square Error
print(paste('The MSE is: ', sum((fitted(mod1) - y)^2)/100))
```

```
## [1] "The MSE is:  0.646805203236385"
```

```
# range from 1~2
set.seed(1)
x <- seq(1,2,length.out = 100)
y <- 2+1.2*x+rnorm(100,0,sd=0.9)
mod1 <- lm(y~x)

# r-squared
print(paste('The R^2 is: ', summary(mod1)$r.squared))
```

```
## [1] "The R^2 is:  0.150244841054036"
```

```
# Mean Square Error
print(paste('The MSE is: ', sum((fitted(mod1) - y)^2)/100))
```

```
## [1] "The MSE is:  0.646805203236384"
```

The R-squared falls from 0.94 to 0.15 but the MSE remains the same. In other words the predictive ability is the same for both data sets, but the R-squared would lead you to believe the first example somehow had a model with more predictive power.

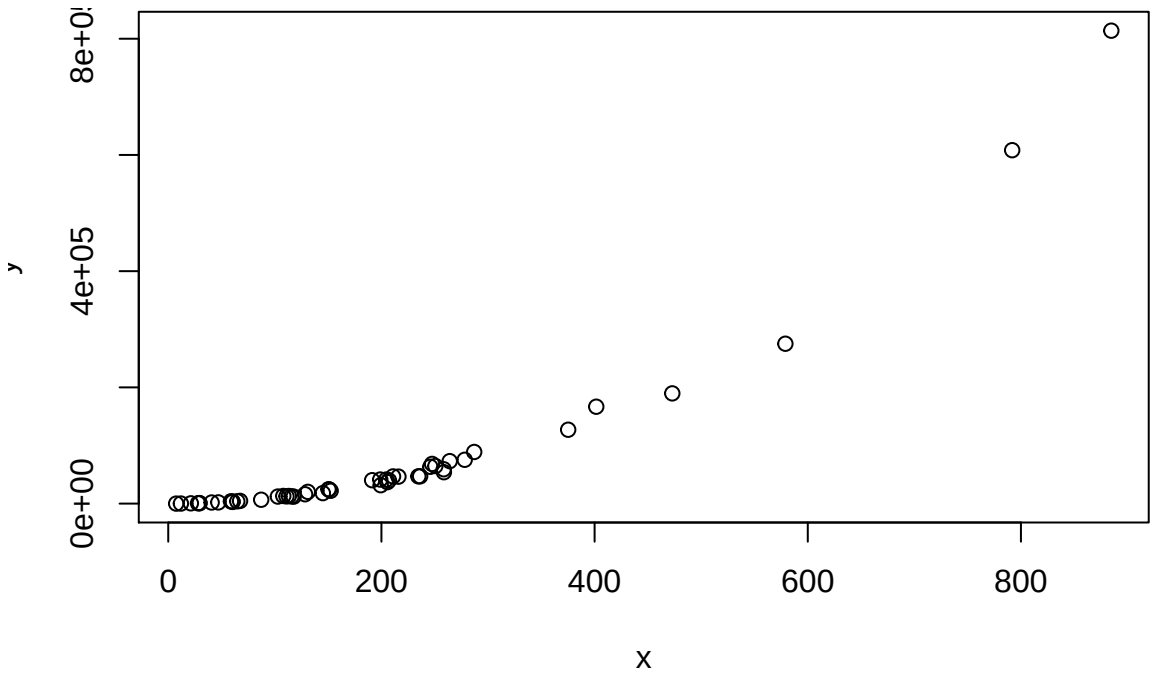
2. R-Sqaured can be closed to 1 when the model is totally wrong.

```
set.seed(1)

# our predictor is data from an exponential distribution
x <- rexp(50, rate = 0.005)

# non-linear data generated
y <- (x-1)^2*runif(50, min = 0.8, max = 1.2)

# clearly non-linear
plot(x,y)
```




```
# check the r-squared
summary(lm(y~x))$r.squared
```

```
## [1] 0.8485146
```

```
# check MSE
sum((fitted(lm(y~x))-y)^2)/50
```

```
## [1] 3087971372
```

It's very high at about 0.85, but the model is completely wrong. Using R-squared to justify the "goodness" of our model in this instance would be a mistake. However, the MSE is super large which means it's not a good model.