DATA 609 HW5

Bin Lin

2017-9-24

Page 228: #1. Consider a model for the long-term dining behavior of the students at College USA. It is found that 25% of the students who eat at the college's Grease Dining Hall return to eat there again, whereas those who eat at Sweet Dining Hall have a 93% return rate. These are the only two dining halls available on campus, and assume that all students eat at one of these halls. Formulate a model to solve for the long-term percentage of students eating at each hall.

$Grease_{n+1} = 0.25Grease_n + 0.07Sweet_n$

$Sweet_{n+1} = 0.75Grease_n + 0.93Sweet_n$

```
#I assume each dining area has 50% of the students in the begining.
Grease <- 0.5
Sweet <- 0.5
df <- data.frame(Grease, Sweet)


df$Grease[1]
```

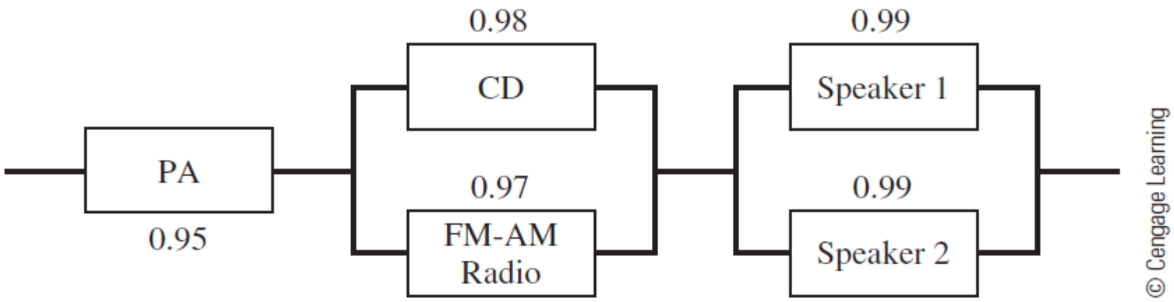
```
## [1] 0.5
```

```
for (i in 1:10)
{
  a <- (0.25 * df$Grease[i] + 0.07 * df$Sweet[i])
  b <- (0.75 * df$Grease[i] + 0.93 * df$Sweet[i])
  df <- rbind(df, c(a, b))
}
df
```

```
##        Grease     Sweet
## 1  0.50000000 0.5000000
## 2  0.16000000 0.8400000
## 3  0.09880000 0.9012000
## 4  0.08778400 0.9122160
## 5  0.08580112 0.9141989
## 6  0.08544420 0.9145558
## 7  0.08537996 0.9146200
## 8  0.08536839 0.9146316
## 9  0.08536631 0.9146337
## 10 0.08536594 0.9146341
## 11 0.08536587 0.9146341
```

The results show that after few iterations, the number of students in each dining hall has reach steady state with around 91.46% of students eat at Sweet Dining Hall and 8.54% of students eat at Grease Dining Hall.

Page 232 #1. Consider a stereo with CD player, FM-AM radio tuner, speakers (dual), and power amplifier (PA) components, as displayed with the reliabilities shown in Figure 6.11. Determine the system's reliability. What assumptions are required in your model?



Alt text

Power Amplifier (PA): series system

$R_{PA} = 0.95$

CD and FM-AM radio tuner: parallel system

$R_{CD-FM-AM} = R_{CD} + R_{FM-AM} - R_{CD} * R_{FM-AM} = 0.98 + 0.97 - 0.98 * 0.97 = 0.9994$

Speaker: parallel system

$R_{speaker} = R_{speaker1} + R_{speaker2} - R_{speaker1} * R_{speaker2} = 0.99 + 0.99 - 0.99 * 0.99 = 0.9999$

System Reliability: series system

$R_{System} = R_{PA} * R_{CD-FM-AM} * R_{Speaker} = 0.95 * 0.9994 * 0.9999 = 0.949$

I am assuming the power amplifier forms a series system with other parts of stereo. CD player and FM-AM radio tuner are parallel each other, but its integrity is also part of series sysmte. Same idea apply for two speakers.

Page 240 Use the basic linear model y = ax + b to fit the following data sets. Provide the model, provide the values of SSE, SSR, SST, and R2, and provide a residual plot.

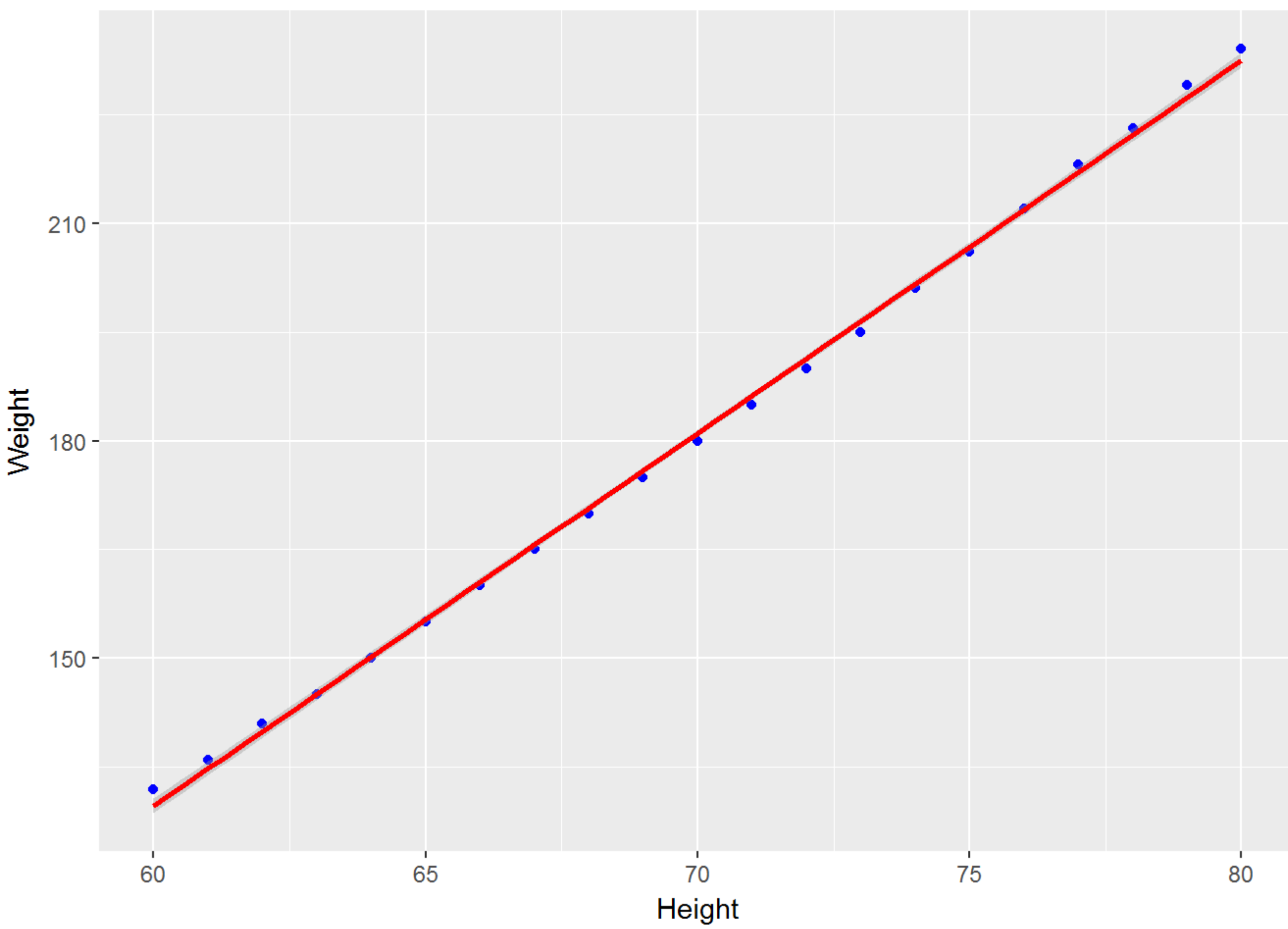
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
df <- data.frame(Height = 60:80, Weight = c(132, 136, 141, 145, 150, 155, 160, 165, 170, 175, 18
0, 185, 190, 195, 201, 206, 212, 218, 223, 229, 234))
df
```

```
##    Height Weight
## 1      60    132
## 2      61    136
## 3      62    141
## 4      63    145
## 5      64    150
## 6      65    155
## 7      66    160
## 8      67    165
## 9      68    170
## 10     69    175
## 11     70    180
## 12     71    185
## 13     72    190
## 14     73    195
## 15     74    201
## 16     75    206
## 17     76    212
## 18     77    218
## 19     78    223
## 20     79    229
## 21     80    234
```

```
ggplot(data = df, aes(x = Height, y = Weight)) + geom_point(color='blue')+   geom_smooth(method
= "lm", formula = y ~ x, color = "red")
```



1. For Table 2.7, predict weight as a function of height.

The linear model for this equation is Weight = 5.1364 * Height -178.49

Slope: $a = \frac{m\sum x_i y_i - \sum x_i \sum y_i}{m\sum x_i^2 - (\sum x_i)^2}$

Intercept: $b = \frac{\sum x_i^2 \sum y_i - \sum x_i y_i \sum x_i}{m\sum x_i^2 - (\sum x_i)^2}$

```
m <- nrow(df)
numerator1 <- m * sum(df$Height * df$Weight) - sum(df$Height) * sum(df$Weight)
denomenator1 <- (m * sum(df$Height ^ 2) - (sum(df$Height) ^ 2))
a <- numerator1 / denomenator1
a
```

```
## [1] 5.136364
```

```
numerator2 <- sum(df$Height ^ 2) * sum(df$Weight) - sum(df$Height * df$Weight) * sum(df$Height)
denomenator2 <- m * sum(df$Height ^ 2) - sum(df$Height) ^ 2
b <- numerator2 / denomenator2
b
```

```
## [1] -178.4978
```

$$SSE \quad = \quad \sum_{i=1}^{m} \left[y_i - (ax_i + b) \right]^2$$

```
SSE <- sum((df$Weight - (a * df$Height + b)) ^ 2)
SSE
```

```
## [1] 24.6342
```

$$SST \quad = \quad \sum_{i=1}^{m} (y_i - \bar{y})^2$$

```
SST <- sum((df$Weight - mean(df$Weight)) ^ 2)
SST
```

```
## [1] 20338.95
```

$SSR = SST - SSE$

```
SSR <- SST - SSE
SSR
```

```
## [1] 20314.32
```

$R^2 = 1 - \frac{SSE}{SST}$

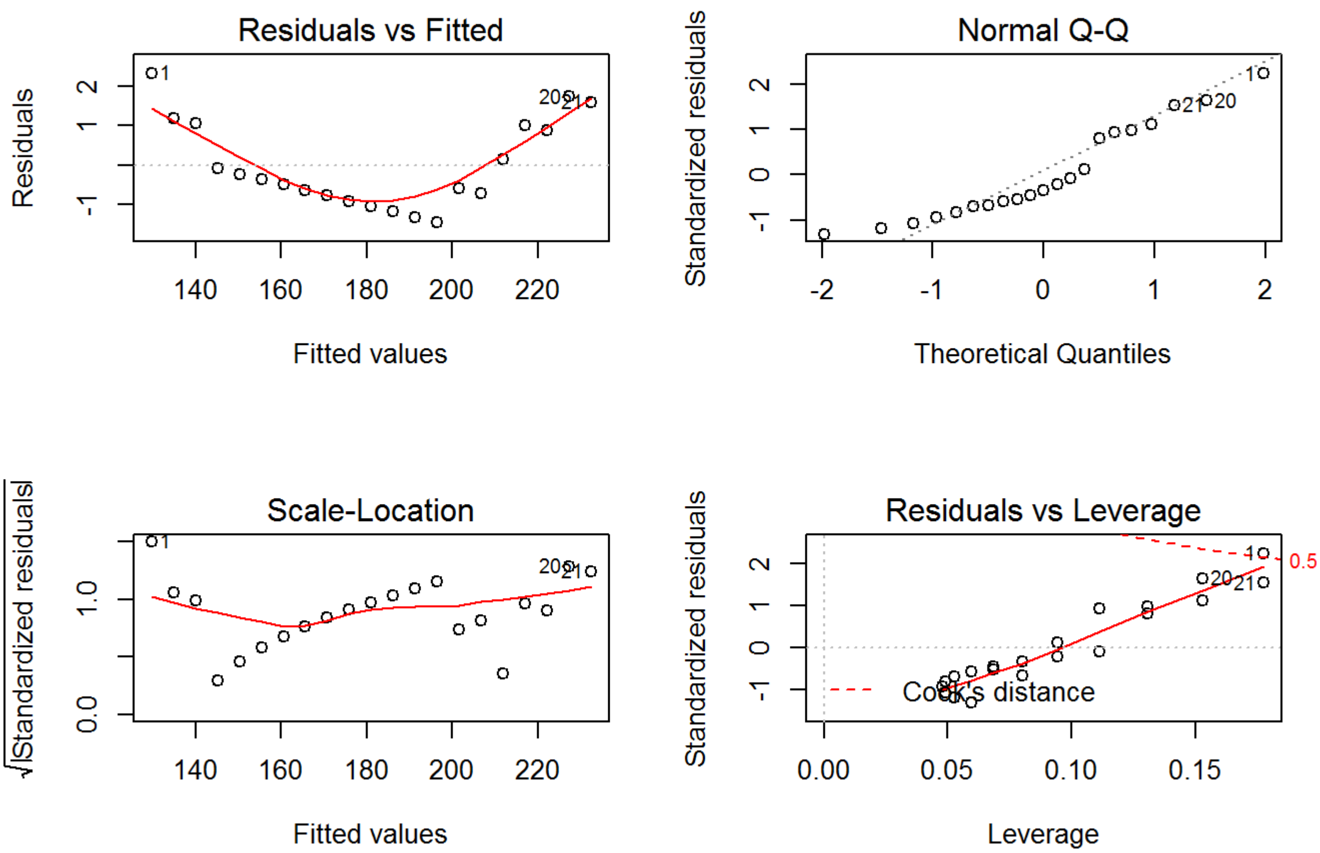
```
r_square <- 1 - (SSE / SST)
r_square
```

```
## [1] 0.9987888
```

```
m1 <- lm(df$Weight ~ df$Height)
summary(m1)
```

```
##
## Call:
## lm(formula = df$Weight ~ df$Height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4567 -0.7749 -0.3658  0.9978  2.3160
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -178.49784    2.88313  -61.91   <2e-16 ***
## df$Height      5.13636    0.04103  125.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.139 on 19 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9987
## F-statistic: 1.567e+04 on 1 and 19 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(m1)
```



2. For Table 2.7, predict weight as a function of the cube of the height.

The linear model for this equation is Weight = 0.000347 * Height^3 + 59.46

Slope: $a = \frac{m \sum x_i y_i - \sum x_i \sum y_i}{m \sum x_i^2 - (\sum x_i)^2}$

Intercept: $b = \frac{\sum x_i^2 \sum y_i - \sum x_i y_i \sum x_i}{m \sum x_i^2 - (\sum x_i)^2}$

```
df$cube_height <- df$Height ^ 3


m <- nrow(df)
numerator1 <- m * sum(df$cube_height * df$Weight) - sum(df$cube_height) * sum(df$Weight)
denomenator1 <- (m * sum(df$cube_height ^ 2) - (sum(df$cube_height) ^ 2))
a <- numerator1 / denomenator1
a
```

```
## [1] 0.0003467044
```

```
numerator2 <- sum(df$cube_height ^ 2) * sum(df$Weight) - sum(df$cube_height * df$Weight) * sum(d
f$cube_height)
denominator2 <- m * sum(df$cube_height ^ 2) - sum(df$cube_height) ^ 2
b <- numerator2 / denominator2
b
```

```
## [1] 59.4584
```

$$SSE \quad = \quad \sum_{i=1}^{m} \left[y_i - (ax_i + b) \right]^2$$

```
SSE <- sum((df$Weight - (a * df$cube_height + b)) ^ 2)
SSE
```

```
## [1] 39.86196
```

$$SST \quad = \quad \sum_{i=1}^{m} (y_i - \bar{y})^2$$

```
SST <- sum((df$Weight - mean(df$Weight)) ^ 2)
SST
```

```
## [1] 20338.95
```

$$SSR = SST - SSE$$

```
SSR <- SST - SSE
SSR
```

```
## [1] 20299.09
```

$$R^2 = 1 - \frac{SSE}{SST}$$

```
r_square <- 1 - (SSE / SST)
r_square
```

```
## [1] 0.9980401
```

```
m2 <- lm(df$Weight ~ df$cube_height)
summary(m2)
```

```
##
## Call:
## lm(formula = df$Weight ~ df$cube_height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9710 -1.0878  0.3279  1.1349  1.6461
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.946e+01  1.276e+00   46.60   <2e-16 ***
## df$cube_height 3.467e-04  3.525e-06   98.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.448 on 19 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.9979
## F-statistic:  9675 on 1 and 19 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(m2)
```