STA 4320 CHAP 3.3.2

Prof. He Jiang

Oct 2024

Sec 3.3.2

Advertising dataset

```
fpath = getwd()
Advertising = read.csv(paste0(fpath, "/Advertising.csv"))
```

Credit and Auto dataset

```
require(ISLR2)
```

```
## Loading required package: ISLR2
```

Interaction

Advertising interaction between TV and radio

```
reg = lm(sales ~ TV * radio, data = Advertising)
# equivalent to
# reg = lm(sales ~ TV + radio + TV:radio, data = Advertising)

summary(reg)
```

```
##
## Call:
## lm(formula = sales ~ TV * radio, data = Advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3366 -0.4028  0.1831  0.5948  1.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.750e+00  2.479e-01  27.233   <2e-16 ***
## TV          1.910e-02  1.504e-03  12.699   <2e-16 ***
## radio       2.886e-02  8.905e-03   3.241   0.0014 **
## TV:radio    1.086e-03  5.242e-05  20.727   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9435 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

Compare to model with only TV and radio separately

```
reg_0 = lm(sales ~ TV + radio, data = Advertising)
summary(reg_0)$r.sq
```

```
## [1] 0.8971943
```

```
# note in the increase in R2 for the interaction model
```

Regression of Balance on Income and Student without interaction

Y = beta_0 + beta_1 * income + beta_2 * student + epsilon

```
head(Credit)
```

```
##    Income Limit Rating Cards Age Education Own Student Married Region Balance
## 1  14.891  3606    283     2  34        11  No      No     Yes  South     333
## 2 106.025  6645    483     3  82        15 Yes     Yes     Yes   West     903
## 3 104.593  7075    514     4  71        11  No      No      No   West     580
## 4 148.924  9504    681     3  36        11 Yes      No      No   West     964
## 5  55.882  4897    357     2  68        16  No      No     Yes  South     331
## 6  80.180  8047    569     4  77        10  No      No      No  South    1151
```

```
reg = lm(Balance ~ Income + Student, data = Credit)
summary(reg)
```

```
## 
## Call:
## lm(formula = Balance ~ Income + Student, data = Credit)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -762.37 -331.38  -45.04  323.60  818.28 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 211.1430    32.4572   6.505 2.34e-10 ***
## Income        5.9843     0.5566  10.751  < 2e-16 ***
## StudentYes  382.6705    65.3108   5.859 9.78e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 391.8 on 397 degrees of freedom
## Multiple R-squared:  0.2775, Adjusted R-squared:  0.2738 
## F-statistic: 76.22 on 2 and 397 DF,  p-value: < 2.2e-16
```

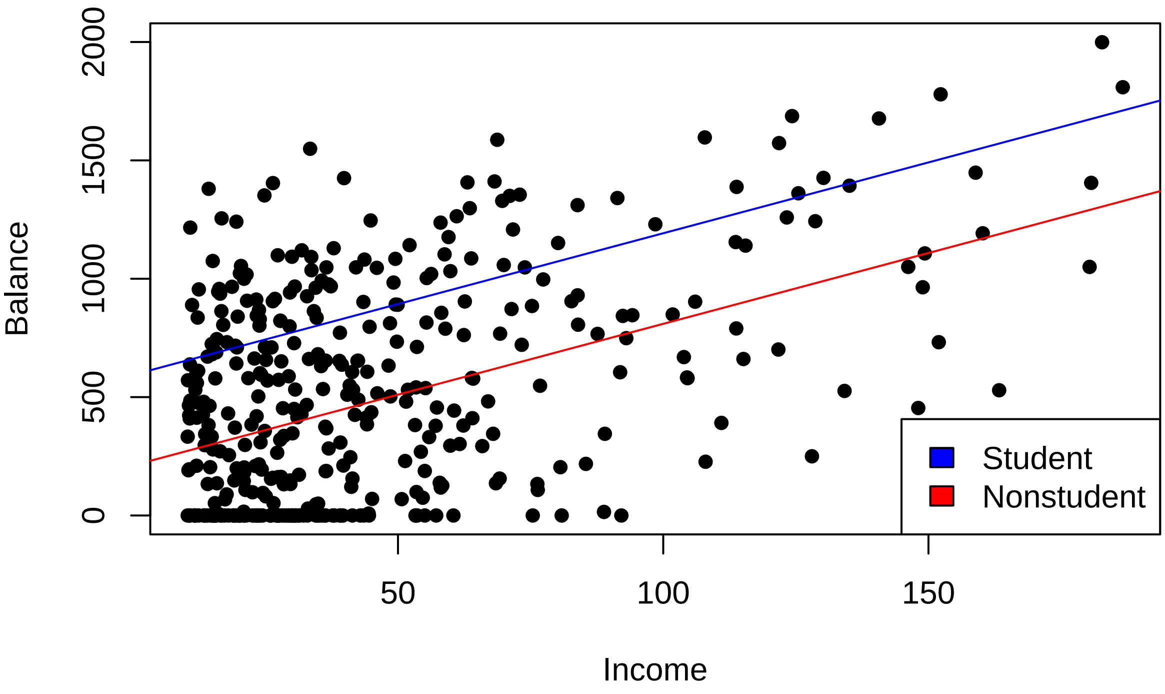
Plot of the two lines with same slope

```
beta_0_hat = as.numeric( reg$coefficients[1] )
beta_1_hat = as.numeric( reg$coefficients[2] )
beta_2_hat = as.numeric( reg$coefficients[3] )

plot(Credit$Income, Credit$Balance,
     pch = 16,
     xlab = "Income",
     ylab = "Balance")
# student line
abline(a = beta_0_hat + beta_2_hat,
```

```r
       b = beta_1_hat,
       col = "blue")
# nonstudent line
abline(a = beta_0_hat,
       b = beta_1_hat,
       col = "red")

legend("bottomright",
       legend = c("Student", "Nonstudent"),
       fill = c("blue", "red"))
```



Regression of Balance on Income and Student with interaction

Y = beta_0 + beta_1 * income + beta_2 * student + beta_3 * income*student + epsilon

```r
reg = lm(Balance ~ Income + Student + Income:Student, data = Credit)
summary(reg)
```

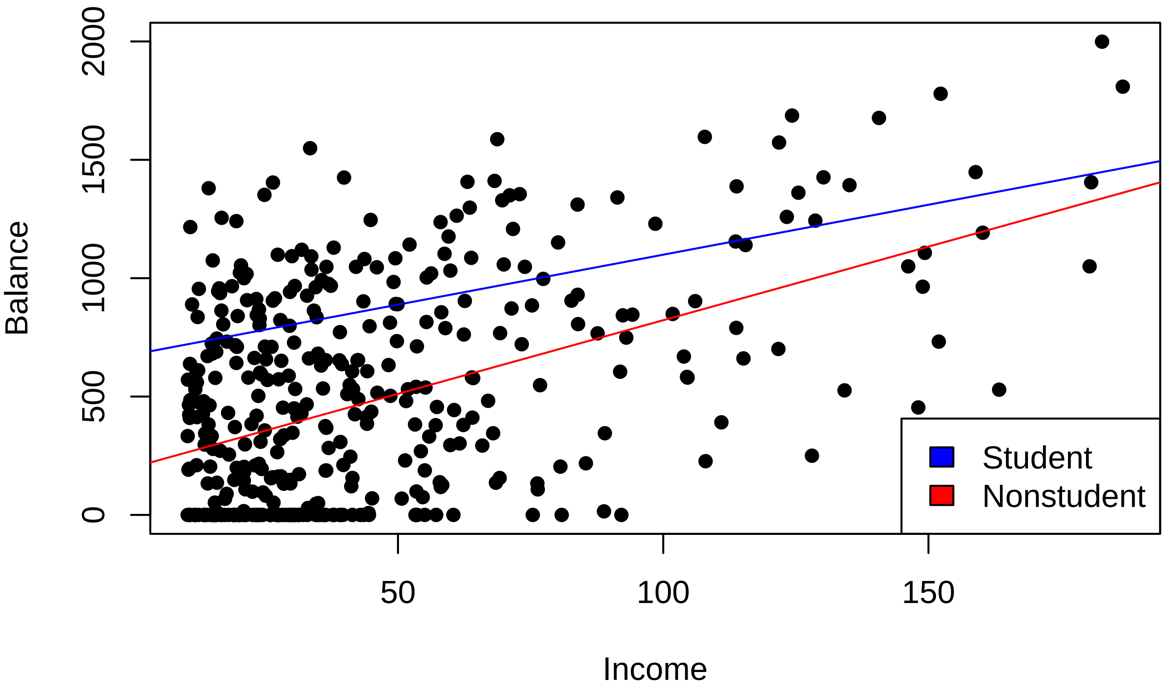
```
##
## Call:
## lm(formula = Balance ~ Income + Student + Income:Student, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -773.39 -325.70  -41.13  321.65  814.04
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       200.6232    33.6984   5.953 5.79e-09 ***
## Income              6.2182     0.5921  10.502  < 2e-16 ***
## StudentYes        476.6758   104.3512   4.568 6.59e-06 ***
## Income:StudentYes  -1.9992     1.7313  -1.155    0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 391.6 on 396 degrees of freedom
## Multiple R-squared:  0.2799, Adjusted R-squared:  0.2744
## F-statistic:  51.3 on 3 and 396 DF,  p-value: < 2.2e-16
```

Plot of the two lines with different slope

```
beta_0_hat = as.numeric( reg$coefficients[1] )
beta_1_hat = as.numeric( reg$coefficients[2] )
beta_2_hat = as.numeric( reg$coefficients[3] )
beta_3_hat = as.numeric( reg$coefficients[4] )

plot(Credit$Income, Credit$Balance,
     pch = 16,
     xlab = "Income",
     ylab = "Balance")
# student line
abline(a = beta_0_hat + beta_2_hat,
       b = beta_1_hat + beta_3_hat,
       col = "blue")
# nonstudent line
abline(a = beta_0_hat,
       b = beta_1_hat,
       col = "red")

legend("bottomright",
       legend = c("Student", "Nonstudent"),
       fill = c("blue", "red"))
```



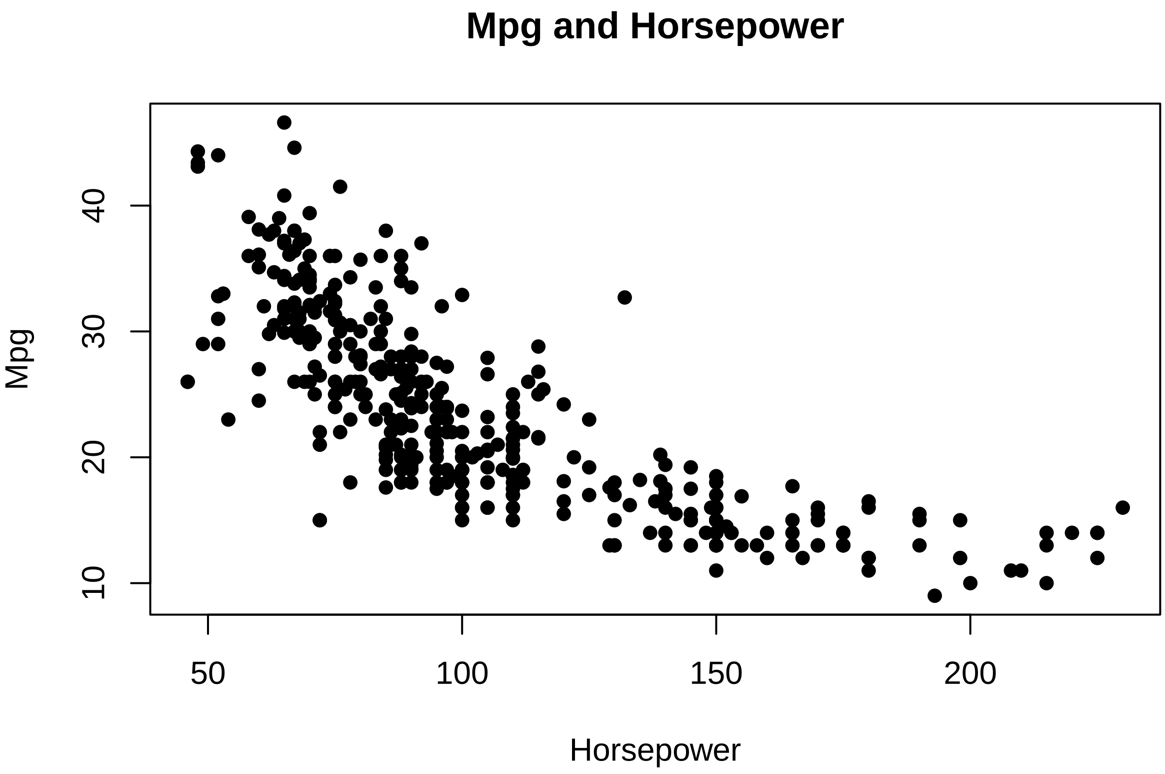
Polynomial

Auto: mpg vs horsepower

Scatterplot

```
y = Auto$mpg
x = Auto$horsepower

plot(x, y,
     main = "Mpg and Horsepower",
     pch = 16,
     xlab = "Horsepower",
     ylab = "Mpg")
```



Polynomial of degree 2

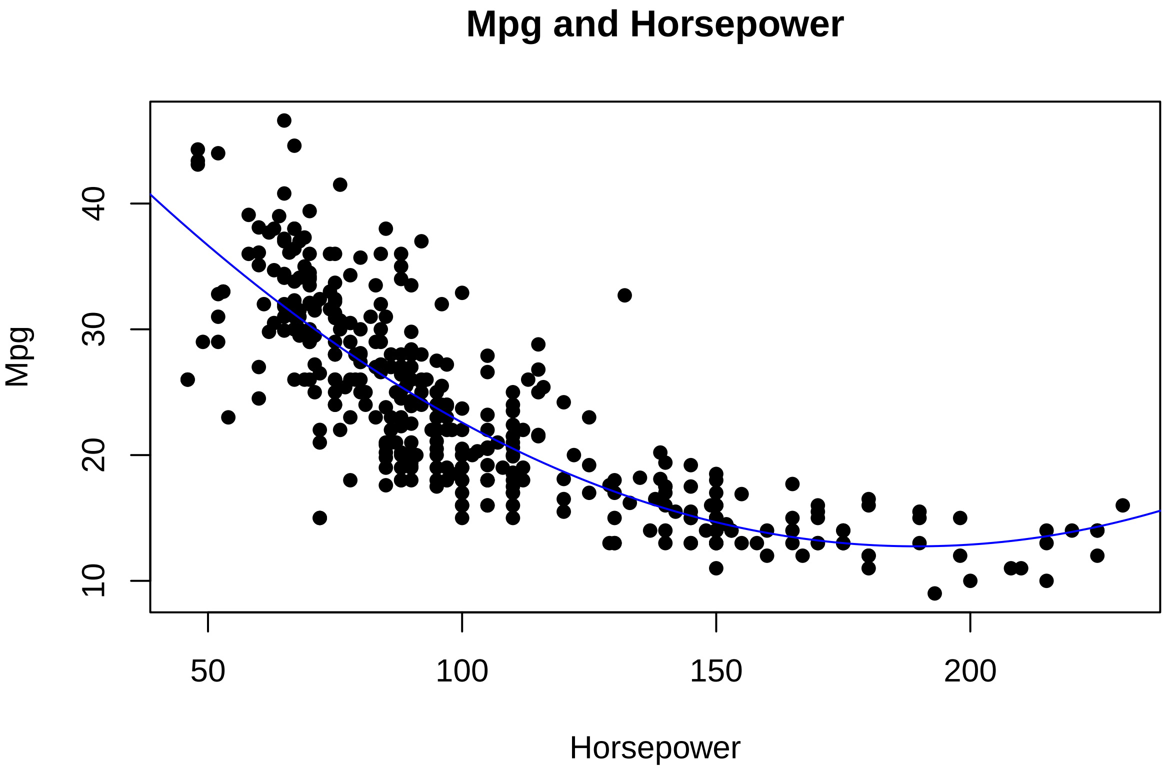
```
reg_2 = lm(mpg ~ horsepower + I(horsepower^2), data = Auto)
# alternatively here reg_2 = lm(y ~ x + I(x^2))
summary(reg_2)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + I(horsepower^2), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7135  -2.5943  -0.0859   2.2868  15.8961
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      56.9000997  1.8004268   31.60   <2e-16 ***
## horsepower       -0.4661896  0.0311246  -14.98   <2e-16 ***
## I(horsepower^2)   0.0012305  0.0001221   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.374 on 389 degrees of freedom
## Multiple R-squared:  0.6876, Adjusted R-squared:  0.686
## F-statistic:   428 on 2 and 389 DF,  p-value: < 2.2e-16
```

Plot

```r
plot(x, y,
     main = "Mpg and Horsepower",
     pch = 16,
     xlab = "Horsepower",
     ylab = "Mpg")

# plotting here requires the
x_poss = seq(min(x) - 10, max(x) + 10, length.out = 1000)
y_poss = predict(reg_2, data.frame(horsepower = x_poss))
lines(x_poss, y_poss, col = "blue")
```



An alternative degree 5 polynomial

```r
reg_5 = lm(mpg ~ poly(horsepower, 5), data = Auto)
summary(reg_5)
```

```
##
## Call:
## lm(formula = mpg ~ poly(horsepower, 5), data = Auto)
##
```

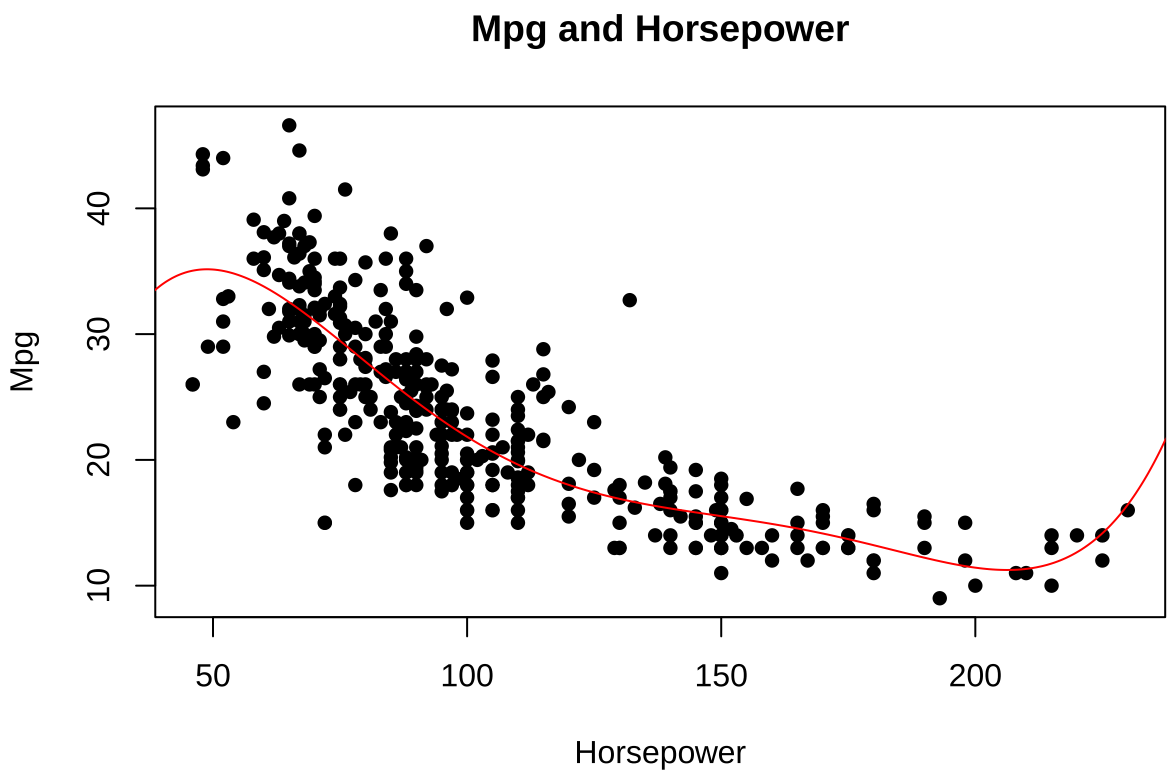
```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -15.4326  -2.5285  -0.2925   2.1750  15.9730 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             23.4459     0.2185 107.308  < 2e-16 ***
## poly(horsepower, 5)1  -120.1377     4.3259 -27.772  < 2e-16 ***
## poly(horsepower, 5)2    44.0895     4.3259  10.192  < 2e-16 ***
## poly(horsepower, 5)3    -3.9488     4.3259  -0.913  0.36190    
## poly(horsepower, 5)4    -5.1878     4.3259  -1.199  0.23117    
## poly(horsepower, 5)5    13.2722     4.3259   3.068  0.00231 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.326 on 386 degrees of freedom
## Multiple R-squared:  0.6967, Adjusted R-squared:  0.6928 
## F-statistic: 177.4 on 5 and 386 DF,  p-value: < 2.2e-16
```

Plot

```r
plot(x, y,
     main = "Mpg and Horsepower",
     pch = 16,
     xlab = "Horsepower",
     ylab = "Mpg")

# plotting here requires the
x_poss = seq(min(x) - 10, max(x) + 10, length.out = 1000)

y_poss = predict(reg_5, data.frame(horsepower = x_poss))
lines(x_poss, y_poss, col = "red")
```



```
# could also add the degree 2 curve in the same picture
# lines(x_poss, predict(reg_2, data.frame(horsepower = x_poss)), col = "blue")
```