Yifan\_Wu\_HW2.Rmd

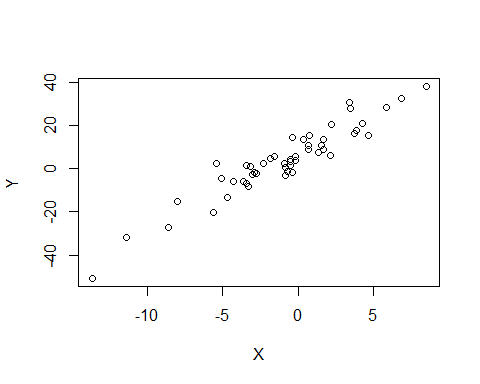
Yifan Wu

February 3, 2018

# 3.7.4

## (a).

true\_beta\_0 = 8  
true\_beta\_1 = 3.5  
  
x<-rnorm(50, 0, 5)  
y <- true\_beta\_1 \* x + true\_beta\_0 + rnorm(50,0,5)  
  
df <- data.frame(X = x,Y = y)  
plot(df$X, df$Y, xlab="X", ylab="Y")



fit <- lm(Y ~ X , df)  
summary(fit)

##   
## Call:  
## lm(formula = Y ~ X, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.390 -3.749 -0.390 3.768 14.328   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.2567 0.7717 9.403 1.82e-12 \*\*\*  
## X 3.5688 0.1777 20.083 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.332 on 48 degrees of freedom  
## Multiple R-squared: 0.8936, Adjusted R-squared: 0.8914   
## F-statistic: 403.3 on 1 and 48 DF, p-value: < 2.2e-16

fit2 <- lm(Y ~ X + I(X^2) + I(X^3) , df)  
summary(fit2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2) + I(X^3), data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.2016 -3.1660 -0.5804 3.4400 12.6907   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.925754 0.932165 7.430 2.07e-09 \*\*\*  
## X 3.108821 0.286392 10.855 2.81e-14 \*\*\*  
## I(X^2) 0.030894 0.039945 0.773 0.4432   
## I(X^3) 0.007428 0.004166 1.783 0.0812 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.22 on 46 degrees of freedom  
## Multiple R-squared: 0.9023, Adjusted R-squared: 0.896   
## F-statistic: 141.7 on 3 and 46 DF, p-value: < 2.2e-16

The residual for the cubic regression model will be smaller than that of the simple linear regression. Since the higher the degree of fitted polynomial, the more likely it is to overfit the data. For a regression line that the highest degree is 3 instead of 1, it will wiggle more to accomodate as many data point as possible by minimizing RSS.

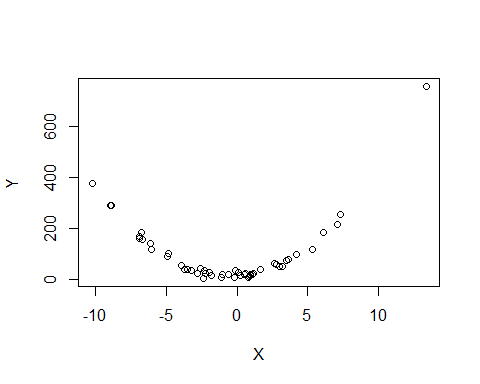
From the above simulation we can see that, the residual for the cubic polynomial regression is smaller than that of the simple linear model.

## (b).

By using test, we assume the null hypothesis that the vectors is a zero vector: the response is independent with all the predictors. Since the true relationship between x and y is linear. The and will unsignificant in terms of predicting y. So the p-value for and will be very big, far exceeds any reasonable significant level. From our simulated data we can see that, the p-value for and is very big. But when only look at the p-value for the model, it is hard to tell which model is a better fit because they both are significant (p-value < 0.05).

## (c).

true\_beta\_0 = 8  
true\_beta\_1 = 4  
true\_beta\_2 = 4  
  
set.seed(1000)  
x<-rnorm(50, 0, 5)  
y <- true\_beta\_2 \* x^2 + true\_beta\_1 \* x + true\_beta\_0 + rnorm(50,0,10)  
  
df <- data.frame(X = x,Y = y)  
plot(df$X, df$Y, xlab="X", ylab="Y")



fit <- lm(Y ~ X , df)  
summary(fit)

##   
## Call:  
## lm(formula = Y ~ X, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -91.94 -75.63 -49.61 37.45 615.65   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 97.004 18.600 5.215 3.86e-06 \*\*\*  
## X 3.498 3.977 0.879 0.384   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 129.6 on 48 degrees of freedom  
## Multiple R-squared: 0.01586, Adjusted R-squared: -0.004644   
## F-statistic: 0.7735 on 1 and 48 DF, p-value: 0.3835

fit2 <- lm(Y ~ X + I(X^2) + I(X^3) , df)  
summary(fit2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2) + I(X^3), data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -22.743 -6.214 1.036 6.750 21.268   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12.219474 1.758813 6.948 1.09e-08 \*\*\*  
## X 3.426274 0.556088 6.161 1.66e-07 \*\*\*  
## I(X^2) 3.876710 0.049925 77.651 < 2e-16 \*\*\*  
## I(X^3) 0.003493 0.006422 0.544 0.589   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.1 on 46 degrees of freedom  
## Multiple R-squared: 0.9943, Adjusted R-squared: 0.9939   
## F-statistic: 2662 on 3 and 46 DF, p-value: < 2.2e-16

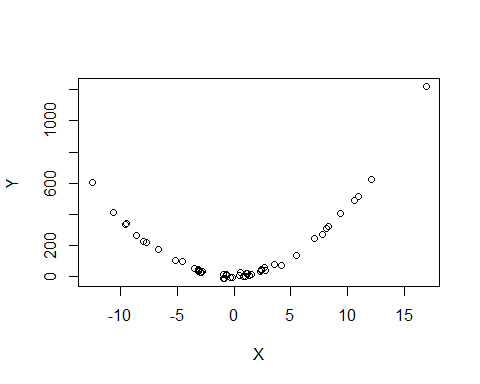
For RSS, the higher the degree of the fitted polynomial is, the lower the RSS will be. The cubic model will have a lower RSS than the simple linear regression model. This is because the higher the degree is, the more wiggly the fitted line is so it can show more detail trend in the data when minimizing the RSS.

## (d).

Between using simple linear regression and cubic polynomial regression, it is not enough information to say which one can do better than the other one because the true relaitionship between response and the predictor is unknown.

Case 1: when the true relationship is quadratic:

true\_beta\_0 = 8  
true\_beta\_1 = 3.5  
true\_beta\_2 = 4  
  
set.seed(1001)  
x<-rnorm(50, 0, 5)  
y <- true\_beta\_2 \* x^2 + true\_beta\_1 \* x + true\_beta\_0 + rnorm(50,0,10)  
  
df <- data.frame(X = x,Y = y)  
plot(df$X, df$Y, xlab="X", ylab="Y")



fit <- lm(Y ~ X , df)  
summary(fit)

##   
## Call:  
## lm(formula = Y ~ X, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -166.07 -152.07 -93.02 93.25 852.37   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 157.423 31.100 5.062 6.53e-06 \*\*\*  
## X 12.458 5.021 2.481 0.0166 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 219.5 on 48 degrees of freedom  
## Multiple R-squared: 0.1137, Adjusted R-squared: 0.09523   
## F-statistic: 6.157 on 1 and 48 DF, p-value: 0.01664

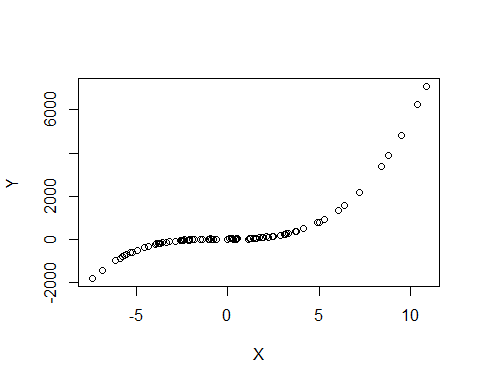
fit2 <- lm(Y ~ X + I(X^2) + I(X^3) , df)  
summary(fit2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2) + I(X^3), data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.6503 -8.0388 -0.2392 8.7657 21.5460   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.628713 1.891425 3.505 0.00103 \*\*   
## X 3.246073 0.451471 7.190 4.73e-09 \*\*\*  
## I(X^2) 4.014341 0.033142 121.127 < 2e-16 \*\*\*  
## I(X^3) 0.001590 0.003542 0.449 0.65561   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.6 on 46 degrees of freedom  
## Multiple R-squared: 0.998, Adjusted R-squared: 0.9979   
## F-statistic: 7726 on 3 and 46 DF, p-value: < 2.2e-16

We can see that if the true relationship between x and y is non-linear(in our simulated case, it is quadratic), the simple linear regression tells us that although the term is significant, there still more to be discovered since the residual is huge. When we are fitting a cubic polynomial, the term and is significant and the term is not and the residual is small at the mean time. So we know that the true relaitionship is duadratic. The p-value for the whole model is smaller in the cubic regression model.

Case 2: when the true relationship is cubic:

true\_beta\_0 = 8  
true\_beta\_1 = 5  
true\_beta\_2 = 5  
true\_beta\_3 = 5  
  
x<-rnorm(100, 0, 5)  
y <- true\_beta\_3 \* x^3 + true\_beta\_2 \* x^2 + true\_beta\_1 \* x + true\_beta\_0 + rnorm(50,0,10)  
  
df <- data.frame(X = x,Y = y)  
plot(df$X, df$Y, xlab="X", ylab="Y")



fit <- lm(Y ~ X , df)  
summary(fit)

##   
## Call:  
## lm(formula = Y ~ X, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -824.0 -575.1 -35.4 402.2 4136.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 247.32 77.97 3.172 0.00202 \*\*   
## X 248.80 19.50 12.761 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 779.7 on 98 degrees of freedom  
## Multiple R-squared: 0.6243, Adjusted R-squared: 0.6204   
## F-statistic: 162.8 on 1 and 98 DF, p-value: < 2.2e-16

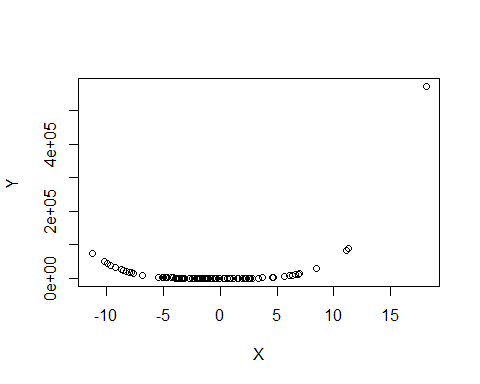
fit2 <- lm(Y ~ X + I(X^2) + I(X^3) , df)  
summary(fit2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2) + I(X^3), data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23.9080 -7.1401 -0.7005 6.7650 24.5352   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.62060 1.31093 6.576 2.52e-09 \*\*\*  
## X 5.95324 0.45256 13.155 < 2e-16 \*\*\*  
## I(X^2) 5.11599 0.06747 75.824 < 2e-16 \*\*\*  
## I(X^3) 4.97791 0.01000 497.557 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.74 on 96 degrees of freedom  
## Multiple R-squared: 0.9999, Adjusted R-squared: 0.9999   
## F-statistic: 5.571e+05 on 3 and 96 DF, p-value: < 2.2e-16

We can see that when the true relationship is cubic, the simple linear regresson is doing badly because although we know tha the term is significant, the residual is very large. When we are fitting a cubic polynomial to the data, all terms are significant and the residual is small, we know that the true relationship is roughly cubic. The p-value for the whole model is the same.

Case 3: When the true relationship is more complex than cubic (i.e. 4th power, 5th power…):

true\_beta\_0 = 8  
true\_beta\_1 = 5  
true\_beta\_2 = 5  
true\_beta\_3 = 5  
true\_beta\_4 = 5  
  
x<-rnorm(100, 0, 5)  
y <- true\_beta\_4 \* x^4 + true\_beta\_3 \* x^3 + true\_beta\_2 \* x^2 + true\_beta\_1 \* x + true\_beta\_0 + rnorm(50,0,10)  
  
df <- data.frame(X = x,Y = y)  
plot(df$X, df$Y, xlab="X", ylab="Y")



fit <- lm(Y ~ X , df)  
summary(fit)

##   
## Call:  
## lm(formula = Y ~ X, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -34229 -23192 -10744 3743 477824   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 16101 5581 2.885 0.004812 \*\*   
## X 4308 1088 3.958 0.000143 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 55410 on 98 degrees of freedom  
## Multiple R-squared: 0.1378, Adjusted R-squared: 0.129   
## F-statistic: 15.67 on 1 and 98 DF, p-value: 0.0001431

fit2 <- lm(Y ~ X + I(X^2) + I(X^3) , df)  
summary(fit2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2) + I(X^3), data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -43009 -9195 2275 9664 51143   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -7660.559 1694.730 -4.520 1.76e-05 \*\*\*  
## X -3121.929 442.053 -7.062 2.58e-10 \*\*\*  
## I(X^2) 672.707 41.281 16.296 < 2e-16 \*\*\*  
## I(X^3) 61.963 3.753 16.512 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 13920 on 96 degrees of freedom  
## Multiple R-squared: 0.9467, Adjusted R-squared: 0.9451   
## F-statistic: 568.6 on 3 and 96 DF, p-value: < 2.2e-16

When the true relationship is more complex than cubic, the linear model and cubic model has the same behavior, their terms are all significant but their residual are large. We know that we should keep adding more power term to the model. One thing to be noticed is that, the residual for the cubic polynomial is smaller than that of the linear model, so we know that the cubic model is a better fit for the data. The p-value for the whole model is smaller in the cubic regression model.

Conclusion:

When using hypothesis test, the cubic regressoin can always gives us more information about the true relationship between the response and the predictors. But when only look at the p-value for the entire model, it is hard to tell which model is a better fit because they both are significant (p-value < 0.05) and sometime at the same valye and sometime one is bigger than the other.

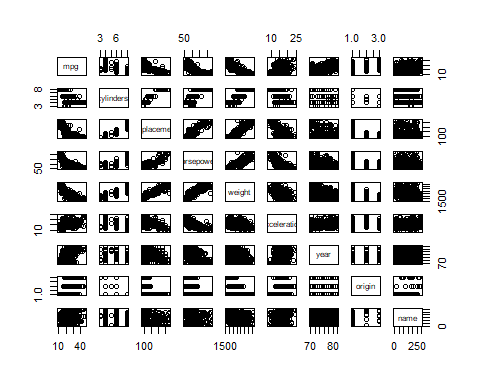
# 3.7.9

## (a). Produce a scatterplot matrix which includes all of the variables in the data set.

library(ISLR)  
dim(Auto)

## [1] 392 9

pairs(Auto)



## (b).

df2 <- subset(Auto, select = -c(name))  
round(cor(df2),3)

## mpg cylinders displacement horsepower weight acceleration  
## mpg 1.000 -0.778 -0.805 -0.778 -0.832 0.423  
## cylinders -0.778 1.000 0.951 0.843 0.898 -0.505  
## displacement -0.805 0.951 1.000 0.897 0.933 -0.544  
## horsepower -0.778 0.843 0.897 1.000 0.865 -0.689  
## weight -0.832 0.898 0.933 0.865 1.000 -0.417  
## acceleration 0.423 -0.505 -0.544 -0.689 -0.417 1.000  
## year 0.581 -0.346 -0.370 -0.416 -0.309 0.290  
## origin 0.565 -0.569 -0.615 -0.455 -0.585 0.213  
## year origin  
## mpg 0.581 0.565  
## cylinders -0.346 -0.569  
## displacement -0.370 -0.615  
## horsepower -0.416 -0.455  
## weight -0.309 -0.585  
## acceleration 0.290 0.213  
## year 1.000 0.182  
## origin 0.182 1.000

## (c).

mul\_lr <- lm(mpg~.,df2)  
summary(mul\_lr)

##   
## Call:  
## lm(formula = mpg ~ ., data = df2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.5903 -2.1565 -0.1169 1.8690 13.0604   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.218435 4.644294 -3.707 0.00024 \*\*\*  
## cylinders -0.493376 0.323282 -1.526 0.12780   
## displacement 0.019896 0.007515 2.647 0.00844 \*\*   
## horsepower -0.016951 0.013787 -1.230 0.21963   
## weight -0.006474 0.000652 -9.929 < 2e-16 \*\*\*  
## acceleration 0.080576 0.098845 0.815 0.41548   
## year 0.750773 0.050973 14.729 < 2e-16 \*\*\*  
## origin 1.426141 0.278136 5.127 4.67e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.328 on 384 degrees of freedom  
## Multiple R-squared: 0.8215, Adjusted R-squared: 0.8182   
## F-statistic: 252.4 on 7 and 384 DF, p-value: < 2.2e-16

### (i).

There is a linear relationship between the response and the predictors because the p-value for the entire model is smaller than 0.05.

### (ii).

The p-value for “displacement”, “weight”, “year”, and “origin” are all statisticall significant. These are all strong evidence of linear dependence. As a result, there is a relationship between the response and “displacement”, “weight”, “year”, and “origin”.

### (iii).

The coefficient for “year” is 0.75, this means that “year” and “mpg” is positively correlated. For every unit increase in “year”, mpg will increase by 0.75 unit.

# 3.7.12

## (a).

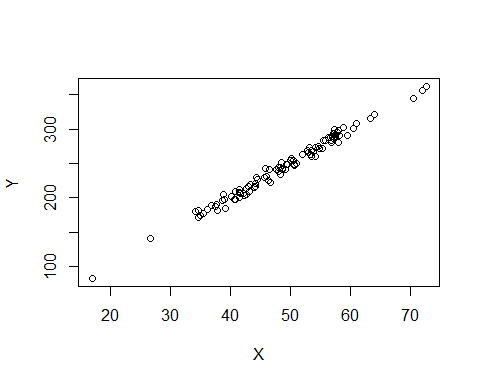
First we let

Because we can cancel out the numerator, that leaves us:

When , the coe???cient estimate for the regression of X onto Y the same as the coe???cient estimate for the regression of Y onto X.

## (b).

true\_beta\_1 = 5  
  
set.seed(0204)  
x<-rnorm(100, 50, 10)  
y <- true\_beta\_1 \* x + rnorm(100,0,5)  
  
df3 <- data.frame(X = x,Y = y)  
plot(df3$X, df3$Y, xlab="X", ylab="Y")



fit3 <- lm(Y ~ X , df3)  
summary(fit3)

##   
## Call:  
## lm(formula = Y ~ X, data = df3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.6678 -4.0051 0.3927 3.5382 12.4882   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.28084 2.89819 0.442 0.66   
## X 4.98083 0.05819 85.601 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.453 on 98 degrees of freedom  
## Multiple R-squared: 0.9868, Adjusted R-squared: 0.9867   
## F-statistic: 7327 on 1 and 98 DF, p-value: < 2.2e-16

fit4 <- lm(X ~ Y , df3)  
summary(fit4)

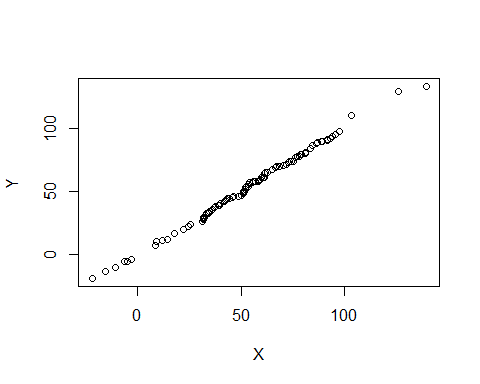
##   
## Call:  
## lm(formula = X ~ Y, data = df3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.51436 -0.69281 -0.07338 0.85081 2.38123   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.391862 0.577237 0.679 0.499   
## Y 0.198120 0.002314 85.601 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.088 on 98 degrees of freedom  
## Multiple R-squared: 0.9868, Adjusted R-squared: 0.9867   
## F-statistic: 7327 on 1 and 98 DF, p-value: < 2.2e-16

From the above simulation, we can see that the coefficient for the regression of X on to Y(0.20) is different from the coefficient for the regression of Y onto X(4.98).

## (c).

Below is the simulated data:

true\_beta\_1 = 1  
  
x<-rnorm(100, 50, 30)  
y <- true\_beta\_1 \* x + rnorm(100,0,5)   
  
df4 <- data.frame(X = sort(x),Y = sort(y))  
plot(df4$X, df4$Y, xlab="X", ylab="Y")



fit5 <- lm(Y ~ X , df4)  
summary(fit5)

##   
## Call:  
## lm(formula = Y ~ X, data = df4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.0255 -0.9896 0.0242 0.9713 6.3409   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.06370 0.39319 -2.705 0.00805 \*\*   
## X 1.01579 0.00657 154.608 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.903 on 98 degrees of freedom  
## Multiple R-squared: 0.9959, Adjusted R-squared: 0.9959   
## F-statistic: 2.39e+04 on 1 and 98 DF, p-value: < 2.2e-16

fit6 <- lm(X ~ Y , df3)  
summary(fit6)

##   
## Call:  
## lm(formula = X ~ Y, data = df3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.51436 -0.69281 -0.07338 0.85081 2.38123   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.391862 0.577237 0.679 0.499   
## Y 0.198120 0.002314 85.601 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.088 on 98 degrees of freedom  
## Multiple R-squared: 0.9868, Adjusted R-squared: 0.9867   
## F-statistic: 7327 on 1 and 98 DF, p-value: < 2.2e-16

# 3.7.13

## (a).

set.seed(1002)  
x <- rnorm(100, 0, 1)

## (b).

set.seed(1003)  
eps <- rnorm(100, 0, 0.25)

## (c).

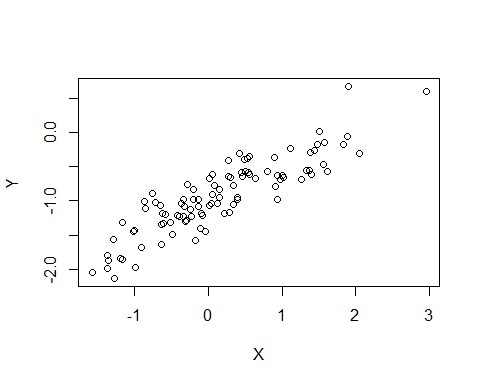
true\_beta\_0 = -1  
true\_beta\_1 = 0.5  
  
y <- true\_beta\_1 \* x + true\_beta\_0 + eps  
length(y)

## [1] 100

The length of vector y is 100. and .

## (d).

df5 <- data.frame(X = x,Y = y)  
plot(df5$X, df5$Y, xlab="X", ylab="Y")



We can see from the scatter plot that there is a clear positive linear relationship between x and y.

## (e).

fit8 <- lm(Y ~ X , df5)  
summary(fit8)

##   
## Call:  
## lm(formula = Y ~ X, data = df5)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.47645 -0.18557 -0.00735 0.16839 0.71031   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.00848 0.02548 -39.58 <2e-16 \*\*\*  
## X 0.51070 0.02763 18.48 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2513 on 98 degrees of freedom  
## Multiple R-squared: 0.7771, Adjusted R-squared: 0.7748   
## F-statistic: 341.6 on 1 and 98 DF, p-value: < 2.2e-16

We can see that the model is a good fit because the p-value for the whole model is smaller than 0.05 and so is the p-value of coefficient of x and the intercept.

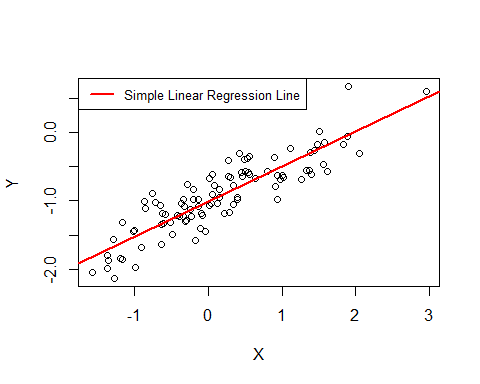
True , fitted ;

True , fitted .

The fitted coefficients are both very close to the truth.

## (f).

plot(df5$X, df5$Y, xlab="X", ylab="Y")  
abline(fit8, col="red",lwd=2) # regression line (y~x)  
legend("topleft", legend = "Simple Linear Regression Line", col="red",lty=1:2, cex=0.8,lwd=2)



## (g).

fit9 <- lm(Y ~ X + I(X^2) , df5)  
summary(fit9)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2), data = df5)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.49210 -0.18430 -0.00648 0.16299 0.75210   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.98888 0.03089 -32.015 <2e-16 \*\*\*  
## X 0.52851 0.03185 16.592 <2e-16 \*\*\*  
## I(X^2) -0.02626 0.02346 -1.119 0.266   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2509 on 97 degrees of freedom  
## Multiple R-squared: 0.7799, Adjusted R-squared: 0.7754   
## F-statistic: 171.9 on 2 and 97 DF, p-value: < 2.2e-16

The p-value for the qudratic term is 0.266, which is bigger than any reasonable significance level. So there is not evidence saying that the quadratic term does not improves the model ???t.

## (h).

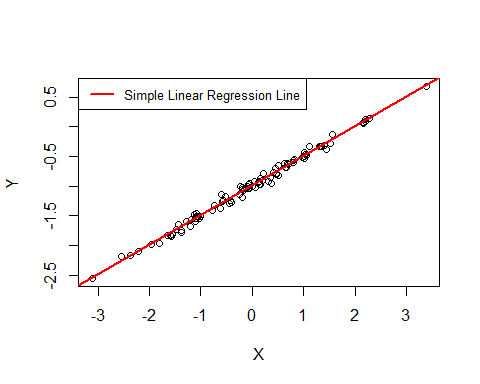
set.seed(1005)  
x <- rnorm(100, 0, 1)  
eps2 <- rnorm(100, 0, 0.05)  
  
true\_beta\_0 = -1  
true\_beta\_1 = 0.5  
  
y <- true\_beta\_1 \* x + true\_beta\_0 + eps2  
length(y)

## [1] 100

df6 <- data.frame(X = x,Y = y)  
plot(df6$X, df6$Y, xlab="X", ylab="Y")  
  
fit10 <- lm(Y ~ X , df6)  
summary(fit10)

##   
## Call:  
## lm(formula = Y ~ X, data = df6)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.152393 -0.036770 -0.002437 0.035297 0.159363   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.991931 0.005417 -183.1 <2e-16 \*\*\*  
## X 0.499709 0.004647 107.5 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.05413 on 98 degrees of freedom  
## Multiple R-squared: 0.9916, Adjusted R-squared: 0.9915   
## F-statistic: 1.156e+04 on 1 and 98 DF, p-value: < 2.2e-16

#plot(df6$X, df6$Y, xlab="X", ylab="Y")  
abline(fit10, col="red",lwd=2) # regression line (y~x)  
legend("topleft", legend = "Simple Linear Regression Line", col="red",lty=1:2, cex=0.8,lwd=2)



fit11 <- lm(Y ~ X + I(X^2) , df6)  
summary(fit11)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2), data = df6)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.153761 -0.036311 -0.000862 0.034157 0.158235   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.990400 0.006612 -149.795 <2e-16 \*\*\*  
## X 0.499702 0.004667 107.064 <2e-16 \*\*\*  
## I(X^2) -0.001128 0.002765 -0.408 0.684   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.05436 on 97 degrees of freedom  
## Multiple R-squared: 0.9916, Adjusted R-squared: 0.9914   
## F-statistic: 5732 on 2 and 97 DF, p-value: < 2.2e-16

We can see that if we decrease the noise level, the data point will be less scattered and the linear trend becomes more obvious. The estimated and they are both very close to the truth and even closer compare to the original more noisy simulation. When we try to fit the data using a quadratic model, the term is not significant because the p-value is too big.

## (i).

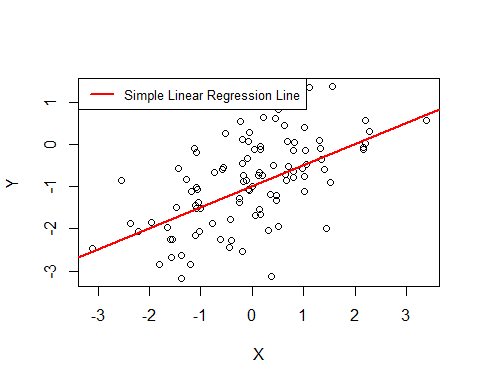
set.seed(1005)  
x <- rnorm(100, 0, 1)  
eps3 <- rnorm(100, 0, 0.8)  
  
true\_beta\_0 = -1  
true\_beta\_1 = 0.5  
  
y <- true\_beta\_1 \* x + true\_beta\_0 + eps3  
length(y)

## [1] 100

df7 <- data.frame(X = x,Y = y)  
plot(df7$X, df7$Y, xlab="X", ylab="Y")  
  
fit12 <- lm(Y ~ X , df7)  
summary(fit12)

##   
## Call:  
## lm(formula = Y ~ X, data = df7)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.4383 -0.5883 -0.0390 0.5648 2.5498   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.87090 0.08668 -10.048 < 2e-16 \*\*\*  
## X 0.49535 0.07436 6.662 1.59e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.866 on 98 degrees of freedom  
## Multiple R-squared: 0.3117, Adjusted R-squared: 0.3047   
## F-statistic: 44.38 on 1 and 98 DF, p-value: 1.587e-09

#plot(df7$X, df7$Y, xlab="X", ylab="Y")  
abline(fit10, col="red",lwd=2) # regression line (y~x)  
legend("topleft", legend = "Simple Linear Regression Line", col="red",lty=1:2, cex=0.8,lwd=2)



fit13 <- lm(Y ~ X + I(X^2) , df7)  
summary(fit13)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2), data = df7)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.4602 -0.5810 -0.0138 0.5465 2.5318   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.84639 0.10579 -8.001 2.67e-12 \*\*\*  
## X 0.49523 0.07468 6.632 1.89e-09 \*\*\*  
## I(X^2) -0.01804 0.04425 -0.408 0.684   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8697 on 97 degrees of freedom  
## Multiple R-squared: 0.3129, Adjusted R-squared: 0.2987   
## F-statistic: 22.08 on 2 and 97 DF, p-value: 1.249e-08

We can see that if we increase the noise level, the data point will be more scattered and the linear trend becomes less obvious. The estimated and they are both very close to the truth but deviate more from the truth compare to the original less noisy simulation. When we try to fit the data using a quadratic model, the term is not significant because the p-value is too big. When increase the noisy level, although the p-value is still smaller than 0.05, the p-value is bigger than the previous less noisy one.

## (j).

round(confint(fit8, '(Intercept)', level=0.95),2)

## 2.5 % 97.5 %  
## (Intercept) -1.06 -0.96

round(confint(fit8, 'X', level=0.95),2)

## 2.5 % 97.5 %  
## X 0.46 0.57

CI for original data set:

round(confint(fit10, '(Intercept)', level=0.95),2)

## 2.5 % 97.5 %  
## (Intercept) -1 -0.98

round(confint(fit10, 'X', level=0.95),2)

## 2.5 % 97.5 %  
## X 0.49 0.51

CI for less noisy data set:

round(confint(fit12, '(Intercept)', level=0.95),2)

## 2.5 % 97.5 %  
## (Intercept) -1.04 -0.7

round(confint(fit12, 'X', level=0.95),2)

## 2.5 % 97.5 %  
## X 0.35 0.64

CI for more noisy data set:

From the CI for and we can conclude that the more noisy the data is, the wider the CIs for their coefficients are.

# 3.7.15

## (a).

We find out the p-value for the regression model for all predictors with “Crime” and find out the ones that exceeds 0.05:

library(MASS)  
library(reshape2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(broom)  
library(reshape)

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':  
##   
## rename

## The following objects are masked from 'package:reshape2':  
##   
## colsplit, melt, recast

df\_b = Boston  
melt\_boston <- melt((df\_b), id.vars = "crim")  
  
lr\_bostn=melt\_boston %>% group\_by(variable) %>% do(tidy(lm(crim ~ value, data=.)))  
lr\_bostn

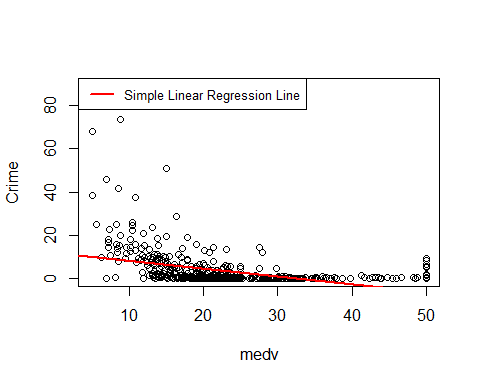
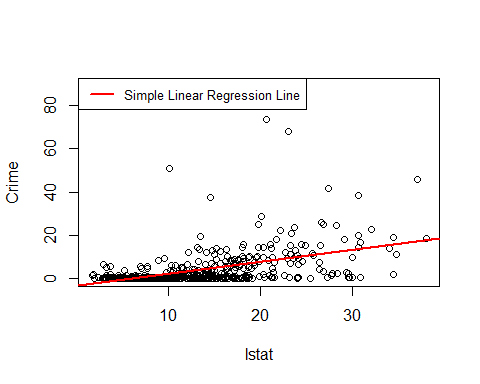
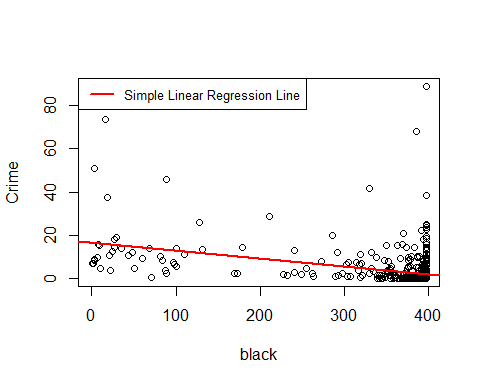
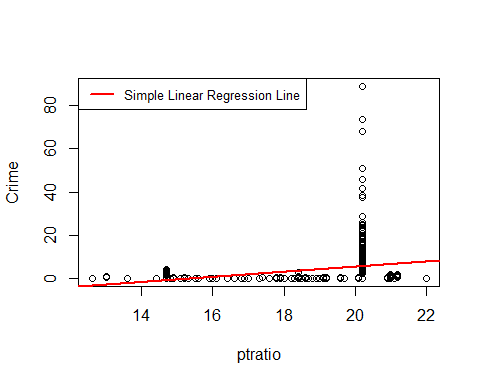
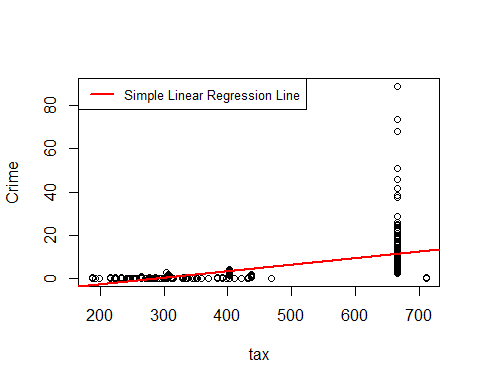
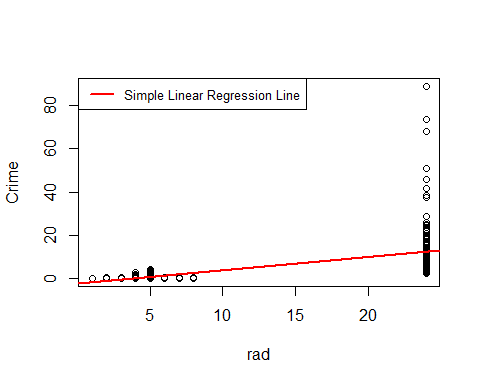
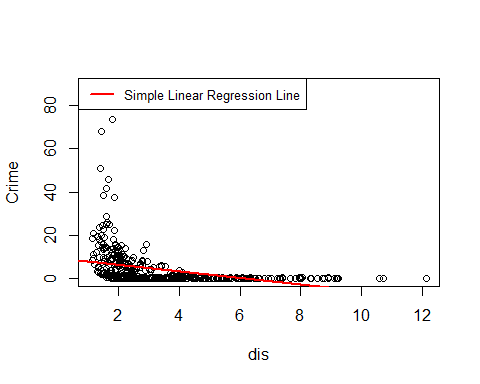
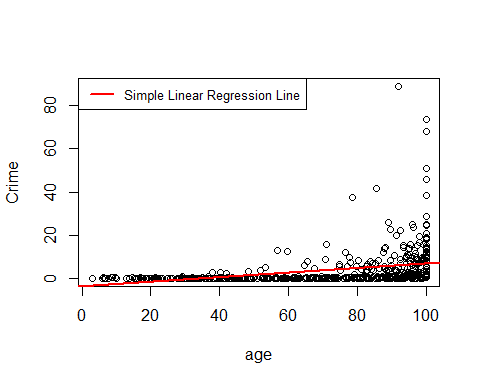
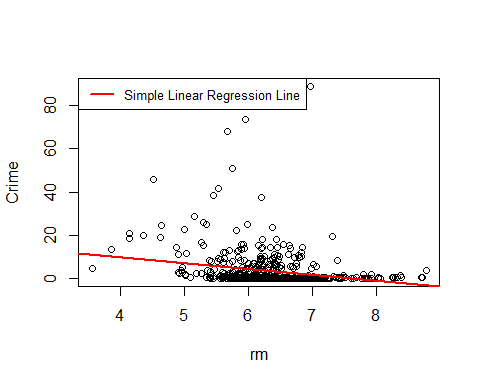
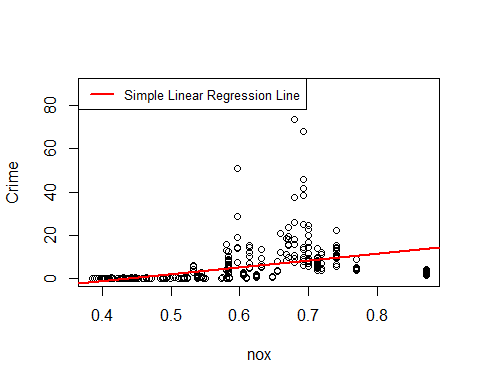
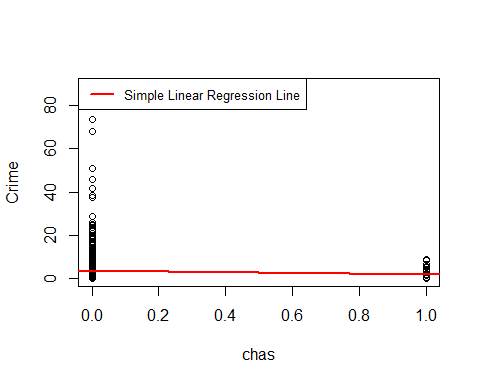
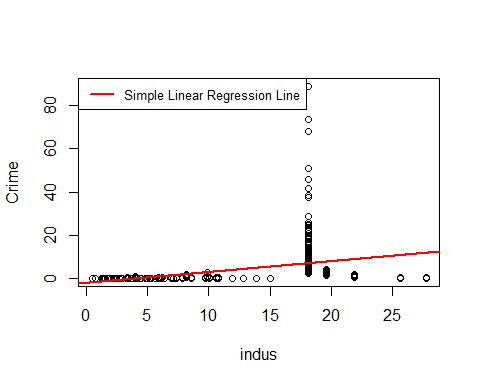
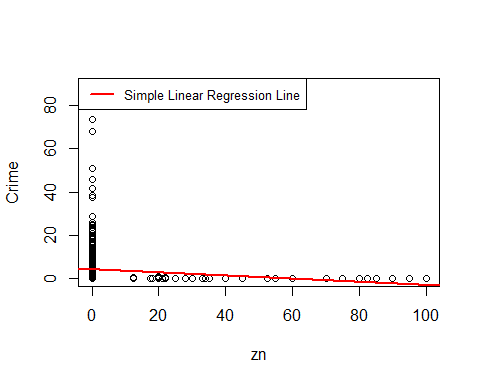
## # A tibble: 26 x 6  
## # Groups: variable [13]  
## variable term estimate std.error statistic p.value  
## <fct> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 zn (Intercept) 4.45 0.417 10.7 4.04e-24  
## 2 zn value - 0.0739 0.0161 - 4.59 5.51e- 6  
## 3 indus (Intercept) - 2.06 0.667 - 3.09 2.09e- 3  
## 4 indus value 0.510 0.0510 9.99 1.45e-21  
## 5 chas (Intercept) 3.74 0.396 9.45 1.24e-19  
## 6 chas value - 1.89 1.51 - 1.26 2.09e- 1  
## 7 nox (Intercept) -13.7 1.70 - 8.07 5.08e-15  
## 8 nox value 31.2 3.00 10.4 3.75e-23  
## 9 rm (Intercept) 20.5 3.36 6.09 2.27e- 9  
## 10 rm value - 2.68 0.532 - 5.04 6.35e- 7  
## # ... with 16 more rows

lr\_boston\_not\_sig = subset(lr\_bostn, lr\_bostn$p.value > 0.05)   
lr\_boston\_not\_sig

## # A tibble: 1 x 6  
## # Groups: variable [1]  
## variable term estimate std.error statistic p.value  
## <fct> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 chas value -1.89 1.51 -1.26 0.209

We can see from the above result that, except for the predictor “chas”, every other variables in the data frame “Boston” has statistically signi???cant association the response.

for(i in names(Boston[-1])){  
 plot( Boston[[i]], Boston$crim, xlab=i, ylab="Crime")  
 abline(lm(Boston$crim ~ Boston[[i]], data = Boston), col="red",lwd=2) # regression line (y~x)  
 legend("topleft", legend = "Simple Linear Regression Line", col="red",lty=1:2, cex=0.8,lwd=2)  
 }



The same result could be interpreted from the plot too. Due the fact that the variable “chas” is a binary variable, it makes sense that a linear model will do badly on fitting it with a continuous variable such as “Crime”. For the fitting for every other variable with “Crime”, we can see that the red line is a good fit for the data.

## (b).

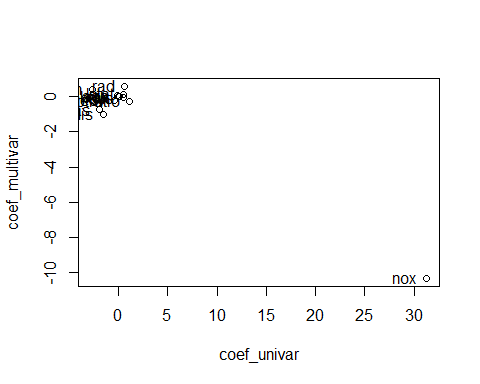
mul\_lr\_boston <- lm(crim~.,Boston)  
summary(mul\_lr\_boston)

##   
## Call:  
## lm(formula = crim ~ ., data = Boston)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.924 -2.120 -0.353 1.019 75.051   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.033228 7.234903 2.354 0.018949 \*   
## zn 0.044855 0.018734 2.394 0.017025 \*   
## indus -0.063855 0.083407 -0.766 0.444294   
## chas -0.749134 1.180147 -0.635 0.525867   
## nox -10.313535 5.275536 -1.955 0.051152 .   
## rm 0.430131 0.612830 0.702 0.483089   
## age 0.001452 0.017925 0.081 0.935488   
## dis -0.987176 0.281817 -3.503 0.000502 \*\*\*  
## rad 0.588209 0.088049 6.680 6.46e-11 \*\*\*  
## tax -0.003780 0.005156 -0.733 0.463793   
## ptratio -0.271081 0.186450 -1.454 0.146611   
## black -0.007538 0.003673 -2.052 0.040702 \*   
## lstat 0.126211 0.075725 1.667 0.096208 .   
## medv -0.198887 0.060516 -3.287 0.001087 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.439 on 492 degrees of freedom  
## Multiple R-squared: 0.454, Adjusted R-squared: 0.4396   
## F-statistic: 31.47 on 13 and 492 DF, p-value: < 2.2e-16

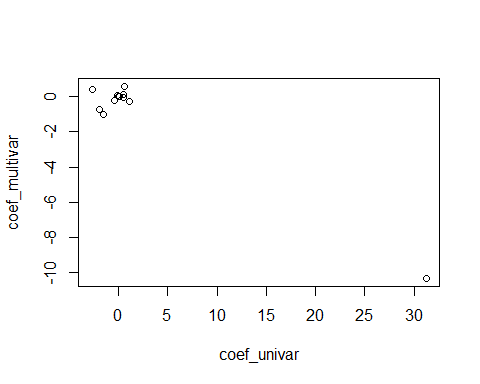
Unlike the result from the seperate simple linear regressoin, only a few variables is significant when we are fitting the data using multiple linear regression. The significant variables are: “zn”, “dis”, “rad”, “black”, and “medv”. So for the above predictors, we can reject the null hypothesis .

## (c).

coef\_univar = lr\_bostn$estimate[lr\_bostn$term != "(Intercept)"]  
coef\_multivar = mul\_lr\_boston$coefficients[-1]  
coef = data.frame(coef\_univar, coef\_multivar)  
  
plot(coef\_univar, coef\_multivar)  
with(coef, text(coef\_multivar~coef\_univar, labels = names(Boston[-1]), pos = 2))



plot(coef\_univar, coef\_multivar)

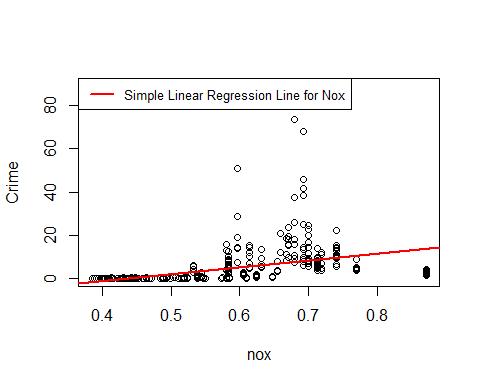


From the above result we can see that most of the coefficients lies around 0 with small difference within each predictor except for “nox”, which apprears to be an outlier whose coefficients varies greatly. Next we investigate “Nox” on its own to find out what’s happening:

#Univeriate linear regression for nox:  
summary(lm(Boston$crim ~ Boston$nox, Boston))

##   
## Call:  
## lm(formula = Boston$crim ~ Boston$nox, data = Boston)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.371 -2.738 -0.974 0.559 81.728   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -13.720 1.699 -8.073 5.08e-15 \*\*\*  
## Boston$nox 31.249 2.999 10.419 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.81 on 504 degrees of freedom  
## Multiple R-squared: 0.1772, Adjusted R-squared: 0.1756   
## F-statistic: 108.6 on 1 and 504 DF, p-value: < 2.2e-16

plot( Boston$nox, Boston$crim, xlab="nox", ylab="Crime")  
abline(lm(Boston$crim ~ Boston$nox, Boston), col="red",lwd=2) # regression line (y~x)  
legend("topleft", legend = "Simple Linear Regression Line for Nox", col="red",lty=1:2, cex=0.8,lwd=2)



#Multivariate linear regressoin:  
summary(mul\_lr\_boston)

##   
## Call:  
## lm(formula = crim ~ ., data = Boston)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.924 -2.120 -0.353 1.019 75.051   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.033228 7.234903 2.354 0.018949 \*   
## zn 0.044855 0.018734 2.394 0.017025 \*   
## indus -0.063855 0.083407 -0.766 0.444294   
## chas -0.749134 1.180147 -0.635 0.525867   
## nox -10.313535 5.275536 -1.955 0.051152 .   
## rm 0.430131 0.612830 0.702 0.483089   
## age 0.001452 0.017925 0.081 0.935488   
## dis -0.987176 0.281817 -3.503 0.000502 \*\*\*  
## rad 0.588209 0.088049 6.680 6.46e-11 \*\*\*  
## tax -0.003780 0.005156 -0.733 0.463793   
## ptratio -0.271081 0.186450 -1.454 0.146611   
## black -0.007538 0.003673 -2.052 0.040702 \*   
## lstat 0.126211 0.075725 1.667 0.096208 .   
## medv -0.198887 0.060516 -3.287 0.001087 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.439 on 492 degrees of freedom  
## Multiple R-squared: 0.454, Adjusted R-squared: 0.4396   
## F-statistic: 31.47 on 13 and 492 DF, p-value: < 2.2e-16

For multivariable linear regressoin: the coefficient for nox(-10.31) is not significant(0.53).

For univeriate linear regresson: the coefficient for nox(31.25) is very significant(2e-16).

This suggestes that when we are taking all variables into account: while holding all other variables constant, “crim” and “nox” is negatively correlated with each other. Holding all other variables constant, increase in one unit in “nox” will cause decrease in “crim” for 10 units. The inconsistant result between univariate regression and multivariate regresion means that “nox” is interacting with other variables in the data.