Intro to Data Science - Lab 8

Week 8 - Linear Models

```
# Enter your name here: Hongdi Li
```

Please include nice comments.

Instructions:

Run the necessary code on your own instance of R-Studio.

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this lab assignment by myself, with help from the book and the
professor.
```

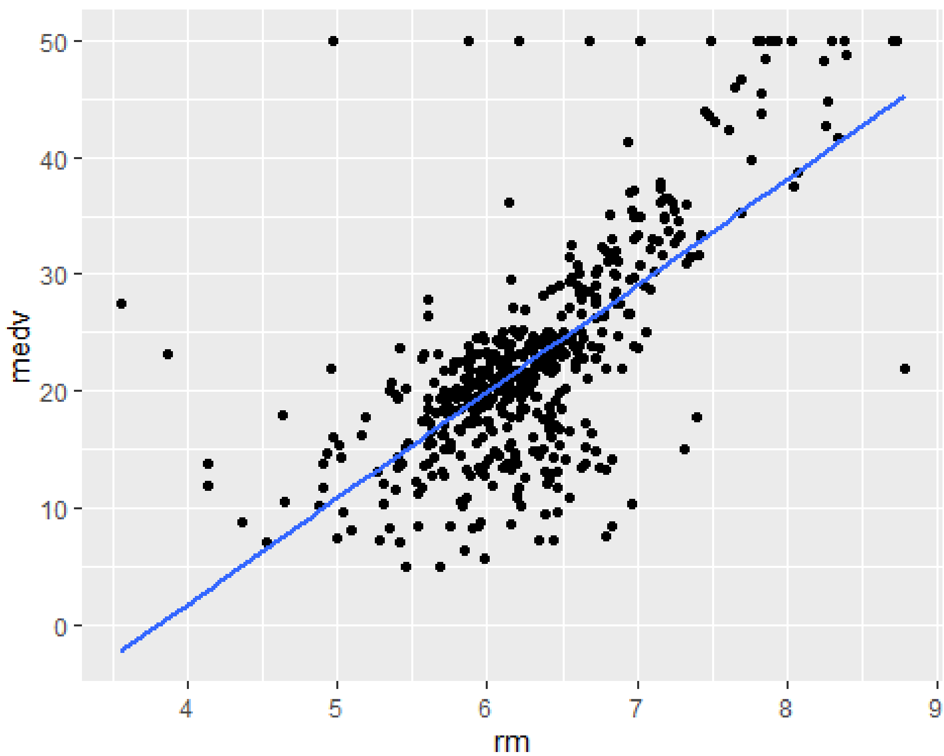
Linear modeling, also referred to as **regression analysis** or multiple regression **bold text**, is a technique for fitting a line, plane, or higher order linear object to data. In their simplest form, linear models have one metric **outcome variable** and one or more **predictor variables** (any combination of metric values, ordered scales such as ratings, or dummy codes).

Make sure to library the **MASS** and **ggplot2** packages before running the following:

```
ggplot(data=Boston) + aes(x=rm, y=medv) + geom_point() +
 geom_smooth(method="lm", se=FALSE)
```

```
library(MASS)
library(ggplot2)
ggplot(data=Boston) + aes(x=rm, y=medv) + geom_point() +
geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



1. Explore this dataset descrption by typing `?Boston` in a code cell.

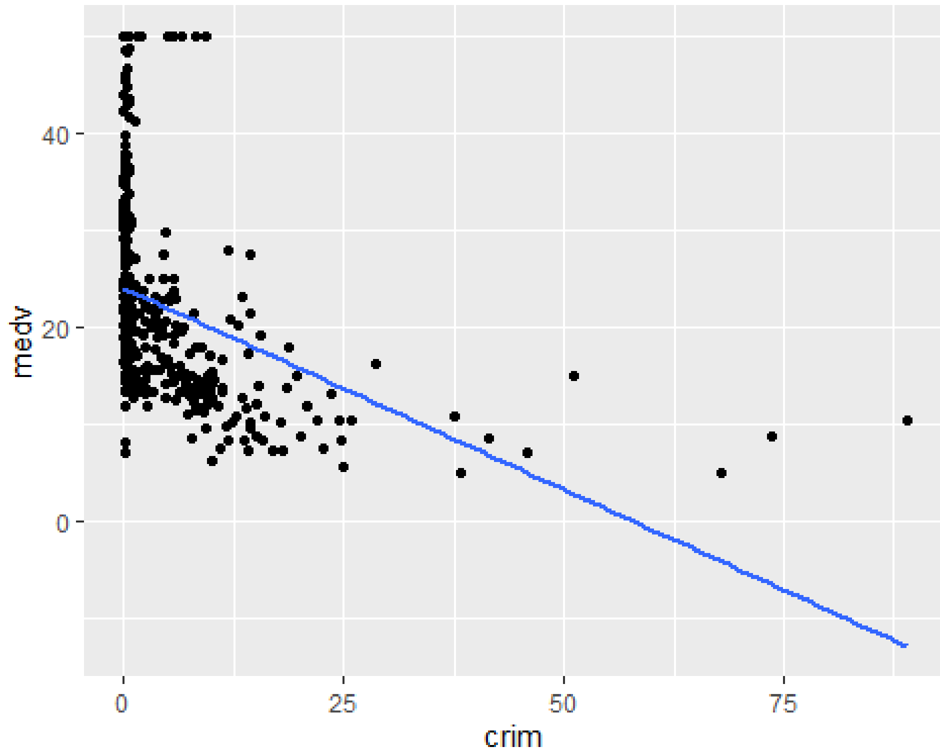
```
?Boston

## starting httpd help server ... done
```

2. The graphic you just created fits a best line to a cloud of points. Copy and modify the code to produce a plot where ** crim ** is the x variable instead of ** rm**.

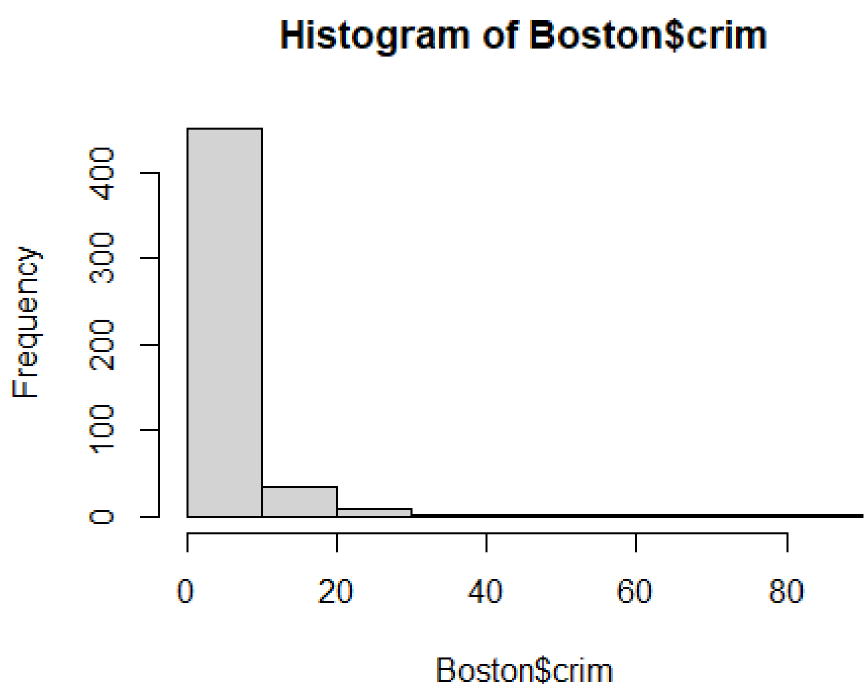
```
ggplot(data=Boston) + aes(x=crim, y=medv) + geom_point() +
geom_smooth(method="lm", se=FALSE)

## `geom_smooth()` using formula 'y ~ x'
```



3. Produce a histogram and descriptive statistics for **Boston$crim**. Write a comment describing any anomalies or oddities.

```
hist(Boston$crim)
```



```r
#The freq is really high at 0 and reduce fast
```

4. Produce a linear model, using the **lm( )** function where **crim** predicts **medv**. Remember that in R s formula language, the **outcome variable** comes first and is separated from the predictors by a **tilde**, like this: `medv ~ crim` Try to get in the habit of storing the output object that is produced by lm and other analysis procedures. For example, I often use **lmOut <- lm(. . .)**

```r
ans<-lm(Boston$medv~Boston$crim)
```

5. Run a **multiple regression** where you use **rm**, **crim**, and **dis** (distance to Boston employment centers). You will use all three predictors in one model with this formula: `medv ~ crim + rm + dis` Now run three separate models for each independent variable separate.

```r
a<-lm(medv ~ crim +dis + rm,data=Boston)
summary(a)

##
## Call:
## lm(formula = medv ~ crim + dis + rm, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.247  -2.930  -0.572   2.390  39.072
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -29.45838    2.60010 -11.330  < 2e-16 ***
## crim         -0.25405    0.03532  -7.193 2.32e-12 ***
## dis           0.12627    0.14382   0.878     0.38
## rm            8.34257    0.40870  20.413  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.238 on 502 degrees of freedom
## Multiple R-squared:  0.5427, Adjusted R-squared:  0.5399
## F-statistic: 198.6 on 3 and 502 DF,  p-value: < 2.2e-16

b<-lm(Boston$crim ~ Boston$medv + Boston$rm + Boston$dis)
summary(b)

##
## Call:
## lm(formula = Boston$crim ~ Boston$medv + Boston$rm + Boston$dis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.441  -3.460  -1.271   1.384  76.955
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  7.64182    3.48912   2.190   0.0290 *
## Boston$medv -0.36780    0.05113  -7.193 2.32e-12 ***
## Boston$rm    1.43098    0.66217   2.161   0.0312 *
## Boston$dis  -1.24741    0.16399  -7.607 1.40e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.506 on 502 degrees of freedom
## Multiple R-squared:  0.243,  Adjusted R-squared:  0.2385
## F-statistic: 53.72 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
c<-lm(Boston$rm ~ Boston$medv + Boston$crim + Boston$dis)
summary(c)
```

```
##
## Call:
## lm(formula = Boston$rm ~ Boston$medv + Boston$crim + Boston$dis)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.95802 -0.23718 -0.00184  0.24503  2.56682
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.963728   0.079004  62.829   <2e-16 ***
## Boston$medv 0.054367   0.002663  20.413   <2e-16 ***
## Boston$crim 0.006441   0.002981   2.161   0.0312 *
## Boston$dis  0.019127   0.011588   1.651   0.0994 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5036 on 502 degrees of freedom
## Multiple R-squared:  0.4893, Adjusted R-squared:  0.4863
## F-statistic: 160.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
d<-lm(Boston$dis ~ Boston$medv + Boston$rm + Boston$crim)
summary(d)
```

```
##
## Call:
## lm(formula = Boston$dis ~ Boston$medv + Boston$rm + Boston$crim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2591 -1.4191 -0.5647  1.1204  7.8165
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.04717    0.89886   2.278   0.0232 *
## Boston$medv  0.01214    0.01383   0.878   0.3804
## Boston$rm    0.28222    0.17098   1.651   0.0994 .
```

```
## Boston$crim -0.08285    0.01089  -7.607  1.4e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.934 on 502 degrees of freedom
## Multiple R-squared:  0.1611, Adjusted R-squared:  0.1561
## F-statistic: 32.13 on 3 and 502 DF,  p-value: < 2.2e-16
```

6. Interpret the results of your analysis in a comment. Make sure to mention the **p-value**, the **adjusted R-squared**, the list of **significant predictors** and the **coefficient** for each significant predictor.

```
#p vlaue for Boston$medv is <<0.05, but for others are larger than 0.05 thus,
#When controlling for other predictors unchanged, the linear relationship
between rm and crim is not significant
#Multiple R-squared:  0.1611 and Adjusted R-squared:  0.1561 , So the
predictors explain around 15-16% of the variance of dis
#It shows that when the above predictors are used to estimate dis, the
average estimation error is 1.934
```

7. Create a one-row **data frame** that contains some plausible values for the predictors. For example, this data frame contains the median values for each predictor: `predDF <- data.frame(crim = 0.26, dis=3.2, rm=6.2)` The numbers used here were selected randomly by looking at min and max data of the variables.

```
predDF <- data.frame(crim = 0.26, dis=3.2, rm=6.2)
```

8. Use the **predict( )** command to predict a new value of **medv** from the one-row data frame. If you stored the output of your lm model in **lmOut**, the command would look like this: `predict(lmOut, predDF)`

```
predict(a, predDF)

##        1
## 22.60355
```