Kristiyan Dimitrov
Predictive Analytics I - Homework I

Exercise 2.3 - Weighted Least Squares

We want to find β s.t. the LS criterion is minimized.

The LS criterion $Q = \sum_{i=1}^{n} w_i (y_i - \beta x_i)^2$. For a minimum we need

This means derivative

$$\frac{\partial Q}{\partial \beta} = 0 = \sum_{i=1}^{n} w_i (y_i^2 + \beta^2 x_i^2 - 2 y_i \beta x_i)' = \sum_{i=1}^{n} (2 w_i \beta x_i^2 - 2 w_i y_i x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{n} 2 w_i \beta x_i^2 = \sum_{i=1}^{n} 2 w_i y_i x_i \Rightarrow \boxed{\beta = \frac{\sum_{i=1}^{n} w_i y_i x_i}{\sum_{i=1}^{n} w_i x_i^2}}$$

Exercise 2.8 - Regression To the Mean

We know that the normalized linear relationship b/w x & y is:

$\frac{\hat{y} - \bar{y}}{s_y} = r \cdot \frac{x - \bar{x}}{s_x}$. From the Galton Data we know $s_y \approx s_x = 2.7$

$\Rightarrow \hat{y} = \bar{y} + r(x - \bar{x})$ where:
- $\bar{y} = 69''$ is avg. son height
- $\bar{x} = 68''$ is avg father height
- x is height of a specific father
- $\hat{y}$ is the expected value of that father's son
- r is correlation coefficient b/w x & y

If $r = .25 \Rightarrow$
- If $x_1 = 64'' \Rightarrow \hat{y}_1 = 69 + .25(64 - 68) = \boxed{68''}$
- If $x_2 = 72'' \Rightarrow \hat{y}_2 = 69 + .25(72 - 68) = \boxed{70''}$

If $r = .75 \Rightarrow$
- If $x_1 = 64'' \Rightarrow \hat{y}_1 = 69 + .75(64 - 68) = \boxed{66''}$
- If $x_2 = 72'' \Rightarrow \hat{y}_2 = 69 + .75(72 - 68) = \boxed{72''}$

We can conclude that with a higher correlation coefficient the "regression to the mean" effect of the sons' heights will be weaker i.e. the height of new sons we observe won't be as close to the mean when r is small.

PA1 - Homework 1

Kristiyan Dimitrov

9/28/2019

Exercise 2.9

```
#First we need to read in our data
data=read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/IBM-Apple-SP500 RR Data.csv
data=data[1:4]
str(data) # It seems the names of the variables are actually included as the first row of data.
```

```
## 'data.frame':    105 obs. of  4 variables:
##  $ X                    : chr  "Date" "9/3/2013" "8/1/2013" "7/1/2013" ...
##  $ Monthly.Return.Rate.: chr  "S&P 500" "3.95%" "-3.13%" "4.95%" ...
##  $ X.1                  : chr  "IBM" "4.22%" "-6.08%" "2.06%" ...
##  $ X.2                  : chr  "Apple" "0.39%" "8.38%" "14.12%" ...
```

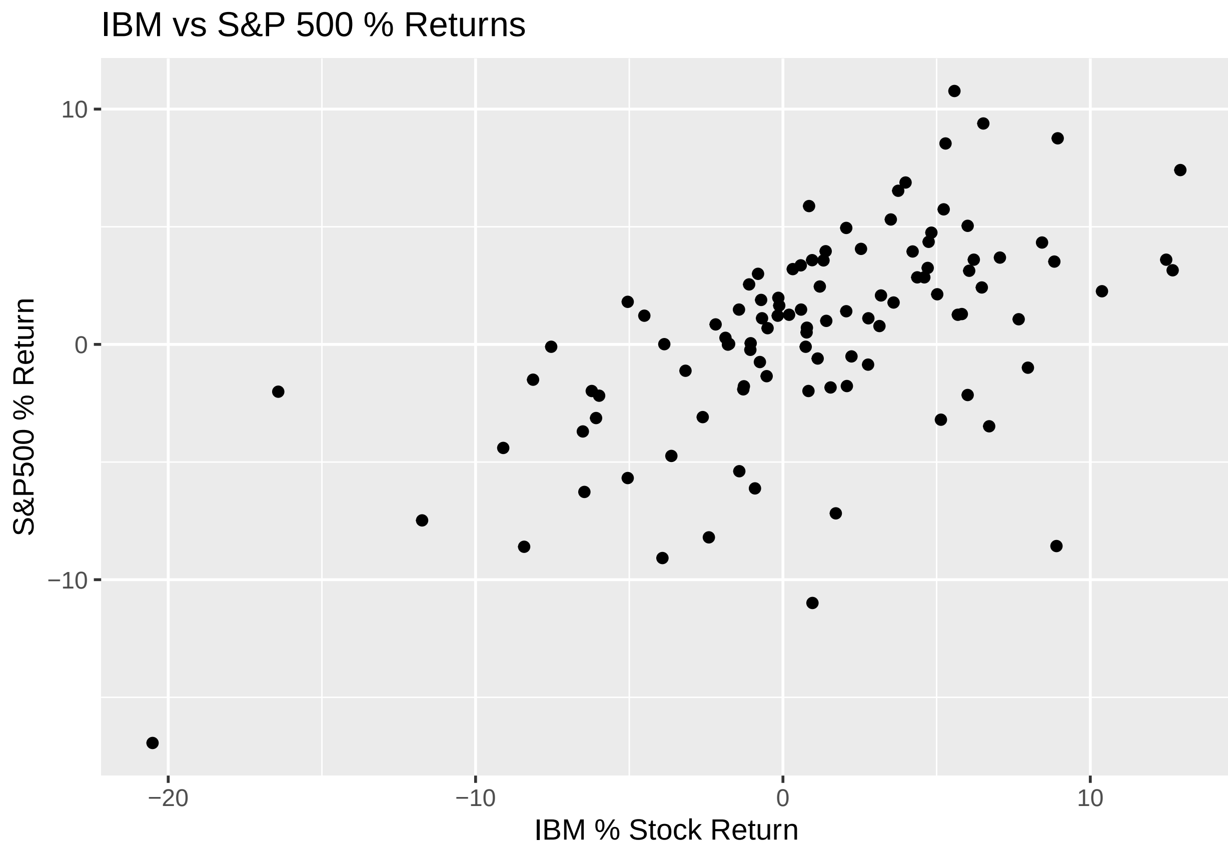
```
colnames(data)<- c("date","sp500","ibm","apple") # Replace the names of the dataframe variables
data <- data %>% #Removing the first row of data
    filter(ibm!="IBM")
str(data)
```

```
## 'data.frame':    104 obs. of  4 variables:
##  $ date : chr  "9/3/2013" "8/1/2013" "7/1/2013" "6/3/2013" ...
##  $ sp500: chr  "3.95%" "-3.13%" "4.95%" "-1.50%" ...
##  $ ibm  : chr  "4.22%" "-6.08%" "2.06%" "-8.13%" ...
##  $ apple: chr  "0.39%" "8.38%" "14.12%" "-11.83%" ...
```

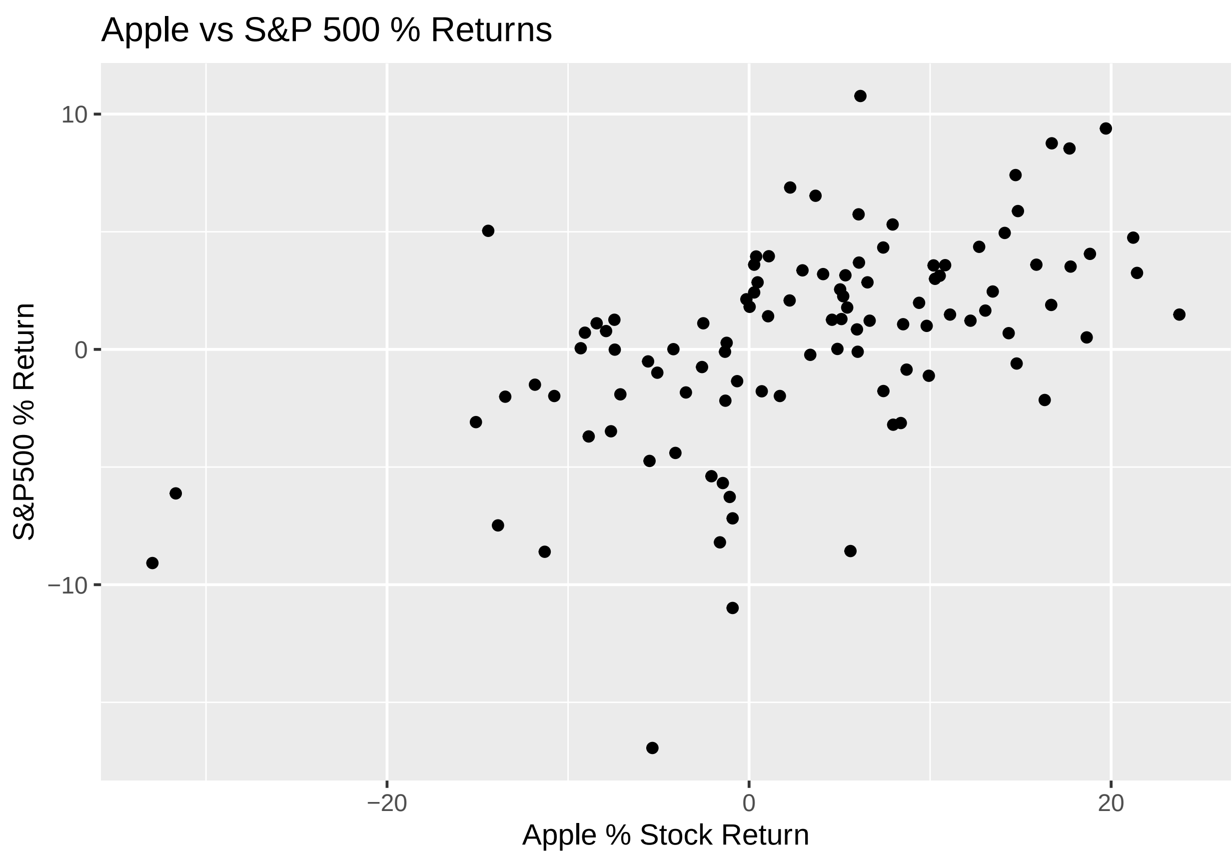
```
#We still have the problem of our data being in chr format.
#Need to cast it as date or integer
data$date<-mdy(data$date) #For the date, we convert formatting from mm-dd-yyyy to yyyy-mm-dd
#In the below three lines we convert sp500, ibm, and apple from chr to numeric
data$sp500 <- (as.numeric(substr(data$sp500,start = 1, stop = nchar(data$sp500)-1)))
data$ibm <- (as.numeric(substr(data$ibm,start = 1, stop = nchar(data$ibm)-1)))
data$apple <- (as.numeric(substr(data$apple,start = 1, stop = nchar(data$apple)-1)))

str(data)
```

```
## 'data.frame':    104 obs. of  4 variables:
##  $ date : Date, format: "2013-09-03" "2013-08-01" ...
##  $ sp500: num  3.95 -3.13 4.95 -1.5 2.08 1.81 3.6 1.11 5.04 0.71 ...
##  $ ibm  : num  4.22 -6.08 2.06 -8.13 3.19 -5.05 6.21 -0.68 6.01 0.78 ...
##  $ apple: num  0.39 8.38 14.12 -11.83 2.24 ...
```

```
#Now that our data is properly formatted, we can create our two scatter plots
ggplot(data = data) +
    geom_point(mapping = aes(x = ibm, y=sp500)) +
    labs(title="IBM vs S&P 500 % Returns", x="IBM % Stock Return", y="S&P500 % Return")
```




```r
ggplot(data = data) +
    geom_point(mapping = aes(x = apple, y=sp500)) +
    labs(title="Apple vs S&P 500 % Returns", x="Apple % Stock Return", y="S&P500 % Return")
```




```r
#Now we do a LS linear fit:
ibm_sp500 <- lm(data$ibm~data$sp500)
apple_sp500 <- lm(data$apple~data$sp500)

#We see that the IBM~S&P 500 beta coefficient is
beta_ibm_sp500<-round(unname(ibm_sp500$coefficients[2]),4)
ibm_sp500$coefficients[2]
```

```
## data$sp500
##  0.7448088
```

```r
#and the Apple~S&P 500 beta coefficient is:
beta_apple_sp500<-round(unname(apple_sp500$coefficients[2]),4)
apple_sp500$coefficients[2]
```

```
## data$sp500
##   1.244856
```

This tells us that Apple has a higher expected %-age return relative to the S&P 500.

```r
# Now we calculate the Standard Deviations of the rates of return for IBM, APPL, and S&P
ibm_sd<-round(sd(data$ibm)/100,4)
apple_sd<-round(sd(data$apple)/100,4)
sp500_sd<-round(sd(data$sp500)/100,4)
print(paste0("Standard Deviation of IBM's Rate or Return: ",ibm_sd))
```

```
## [1] "Standard Deviation of IBM's Rate or Return: 0.0556"
```

```r
print(paste0("Standard Deviation of Apple's Rate or Return: ",apple_sd))
```

```
## [1] "Standard Deviation of Apple's Rate or Return: 0.1031"
```

```
print(paste0("Standard Deviation of S&P's Rate or Return: ", sp500_sd))
```

```
## [1] "Standard Deviation of S&P's Rate or Return: 0.0446"
```

```
#Here we calculate the correlation matrix for the three rates of return
cor_mat<-cor(data[,2:4],method="pearson")
cor_mat
```

```
##           sp500       ibm     apple
## sp500 1.0000000 0.5974779 0.5382317
## ibm   0.5974779 1.0000000 0.4147253
## apple 0.5382317 0.4147253 1.0000000
```

```
#Checking now that the beta coefficient for each stock is beta_hat = r * s_y / s_x
round(beta_apple_sp500,.0001)==round(cor_mat[1,3]*apple_sd/sp500_sd,.0001)
```

```
## [1] TRUE
```

```
round(beta_ibm_sp500,.0001)==round(cor_mat[1,3]*ibm_sd/sp500_sd,.0001)
```

```
## [1] TRUE
```

```
## The above comparisons (and the fact that it's true)
## shows that with a higher volatility (s_y) comes a higher beta_hat
## and therefore a higher expected return for the given stock
##
## Furthermore, in general we know that beta = s_xy / s_xx
## where s_xy= sum of the squared differences(y-y_bar)*(x-x_bar)
## Therefore, the further away the y values are from y_bar i.e.
## the higher the deviation / volatility, the higher the beta
```

Exercise 2.10

```
data=read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/steakprices.csv",stringsAsF
names(data)<-tolower(names(data)) #Converting everything to lowercase
names(data)<-gsub(names(data),pattern="\\.",replacement = "_") # replacing . with _ in names.
str(data)
```

```
## 'data.frame':    48 obs. of  8 variables:
##  $ year         : int  2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
##  $ month        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ chuck_qty    : int  120 76 102 106 87 94 97 79 138 129 ...
##  $ chuck_price  : chr  "$2.28 " "$2.61 " "$2.12 " "$2.41 " ...
##  $ porthse_qty  : int  53 81 60 65 92 157 149 133 97 113 ...
##  $ porthse_price: chr  "$6.04 " "$5.37 " "$5.74 " "$6.93 " ...
##  $ ribeye_qty   : int  74 79 71 112 113 89 146 120 120 106 ...
##  $ ribeye_price : chr  "$7.02 " "$7.16 " "$7.33 " "$7.38 " ...
```

```
#Notice that we have the prices of the steaks as chr with a $ in the front. We need to fix that
data$chuck_price <- (as.numeric(substr(data$chuck_price,start = 2, stop = nchar(data$chuck_price))))
data$porthse_price <- (as.numeric(substr(data$porthse_price,start = 2, stop = nchar(data$porthse_price)
data$ribeye_price <- (as.numeric(substr(data$ribeye_price,start = 2, stop = nchar(data$ribeye_price))))
str(data)
```

```
## 'data.frame':    48 obs. of  8 variables:
##  $ year         : int  2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
```

```
##  $ month        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ chuck_qty    : int  120 76 102 106 87 94 97 79 138 129 ...
##  $ chuck_price  : num  2.28 2.61 2.12 2.41 2.39 2.11 2.66 2.5 2.39 2.3 ...
##  $ porthse_qty  : int  53 81 60 65 92 157 149 133 97 113 ...
##  $ porthse_price: num  6.04 5.37 5.74 6.93 5.95 5.24 5.39 5.54 6.28 5.43 ...
##  $ ribeye_qty   : int  74 79 71 112 113 89 146 120 120 106 ...
##  $ ribeye_price : num  7.02 7.16 7.33 7.38 6.47 7.14 7.02 6.28 7.57 6.72 ...
```

```r
#To calculate the price elasticity, we need to first calculate the log values for all prices & qtys
data <- data %>%
    # Calculating log of chuck price & qty
    mutate(log_chuck_qty=log(chuck_qty),log_chuck_price=log(chuck_price)) %>%
    # Calculating log of porterhouse price & qty
    mutate(log_porthse_qty=log(porthse_qty),log_porthse_price=log(porthse_price)) %>%
    # Calculating log of ribeye price & qty
    mutate(log_ribeye_qty=log(ribeye_qty),log_ribeye_price=log(ribeye_price))

str(data)
```

```
## 'data.frame':    48 obs. of  14 variables:
##  $ year             : int  2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
##  $ month            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ chuck_qty        : int  120 76 102 106 87 94 97 79 138 129 ...
##  $ chuck_price      : num  2.28 2.61 2.12 2.41 2.39 2.11 2.66 2.5 2.39 2.3 ...
##  $ porthse_qty      : int  53 81 60 65 92 157 149 133 97 113 ...
##  $ porthse_price    : num  6.04 5.37 5.74 6.93 5.95 5.24 5.39 5.54 6.28 5.43 ...
##  $ ribeye_qty       : int  74 79 71 112 113 89 146 120 120 106 ...
##  $ ribeye_price     : num  7.02 7.16 7.33 7.38 6.47 7.14 7.02 6.28 7.57 6.72 ...
##  $ log_chuck_qty    : num  4.79 4.33 4.62 4.66 4.47 ...
##  $ log_chuck_price  : num  0.824 0.959 0.751 0.88 0.871 ...
##  $ log_porthse_qty  : num  3.97 4.39 4.09 4.17 4.52 ...
##  $ log_porthse_price: num  1.8 1.68 1.75 1.94 1.78 ...
##  $ log_ribeye_qty   : num  4.3 4.37 4.26 4.72 4.73 ...
##  $ log_ribeye_price : num  1.95 1.97 1.99 2 1.87 ...
```

```r
#The price elasticity for each type of steak is the beta coefficient
# in the LS model b/w the log of the price and log of the qty
chuck_elas <- round(unname(lm(data$log_chuck_qty ~ data$log_chuck_price)$coefficients[2]),4)
porthse_elas <- round(unname(lm(data$log_porthse_qty ~ data$log_porthse_price)$coefficients[2]),4)
ribeye_elas <- round(unname(lm(data$log_ribeye_qty ~ data$log_ribeye_price)$coefficients[2]),4)
```

```
## [1] "The price elasticity of Chuck steaks is: -1.3687"

## [1] "The price elasticity of Porterhouse steaks is: -2.6565"

## [1] "The price elasticity of Ribeye steaks is: -1.446"
```

```r
#Let's calculate the mean prices for each steak
mean_chuck_price <- mean(data$chuck_price)
print(paste0("The mean price for Chuck is: ", mean_chuck_price))
```

```
## [1] "The mean price for Chuck is: 2.4525"
```

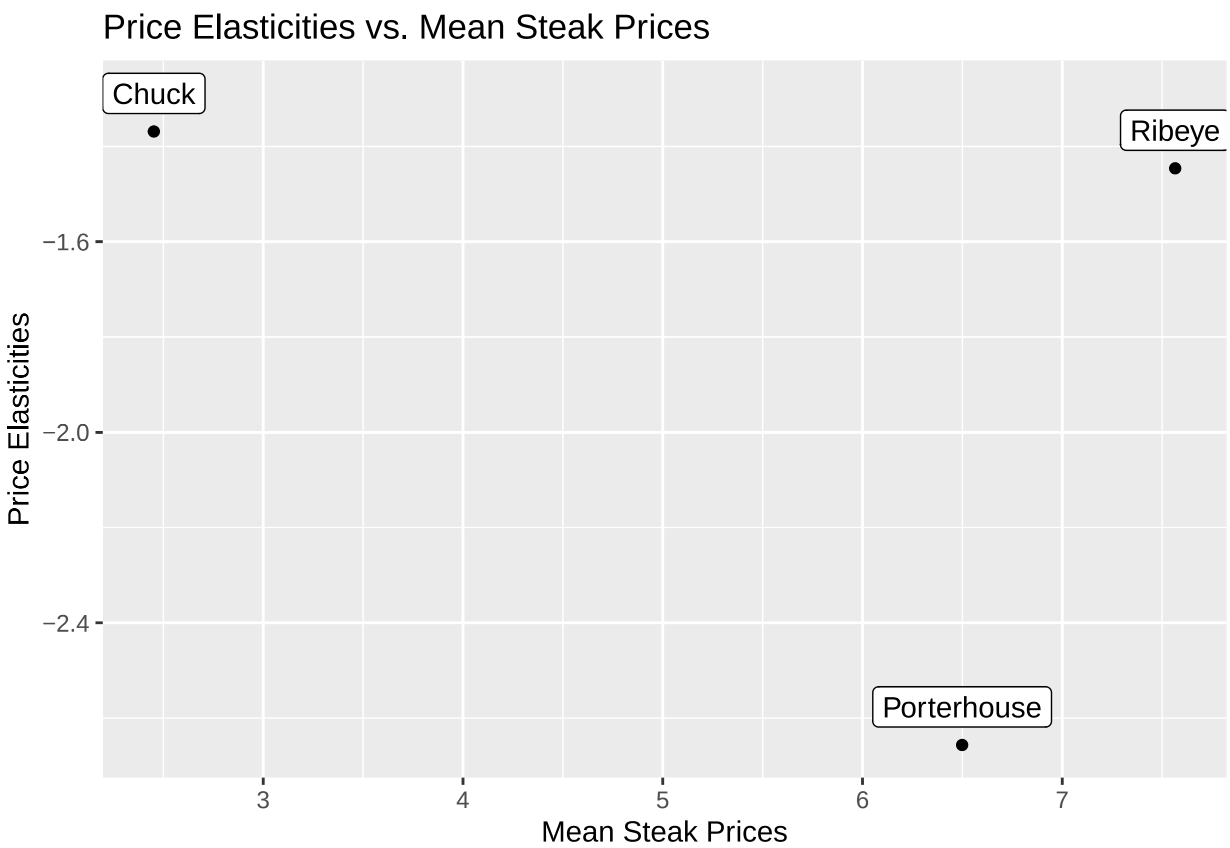
```r
mean_porthse_price <- mean(data$porthse_price)
print(paste0("The mean price for Porthouse is: ",mean_porthse_price))
```

```
## [1] "The mean price for Porthouse is: 6.49875"
```

```r
mean_ribeye_price <- mean(data$ribeye_price)
print(paste0("The mean price for Ribeye is: ", mean_ribeye_price))
```

```
## [1] "The mean price for Ribeye is: 7.56520833333333"
```

```r
#Next we put the mean prices and the price elasticities in a tibble.
prices_and_elas<- tibble(c(mean_chuck_price,mean_porthse_price,mean_ribeye_price),c(chuck_elas,porthse_
#We give the columns some names
names(prices_and_elas)<- c("mean_prices","price_elasticities")
#And finally we plot the elasticities & mean prices
ggplot(data = prices_and_elas) +
    geom_point(mapping = aes(x = prices_and_elas$mean_prices, y=prices_and_elas$price_elasticities)) +
    geom_label(mapping = aes(x = prices_and_elas$mean_prices, y=prices_and_elas$price_elasticities, labe
    labs(title="Price Elasticities vs. Mean Steak Prices", x="Mean Steak Prices", y="Price Elasticities
```




```
## [1] "The above graph shows us that the price elasticity for Porterhouse is much lower. \nThis means
```

```r
#Based on the definition of price elasticity, we can assume that
#the demand will change by an amount equal to elasticity * % price change

#Therefore:
demand_change_chuck=percent(chuck_elas*.1)
demand_change_porterhse=percent(porthse_elas*.1)
demand_change_ribeye=percent(ribeye_elas*.1)
```

```
## [1] "We expect the demand for Chuck to go down by -13.7% if the price goes up by 10%"

## [1] "We expect the demand for Porterhouse to go down by -26.6% if the price goes up by 10%"
```

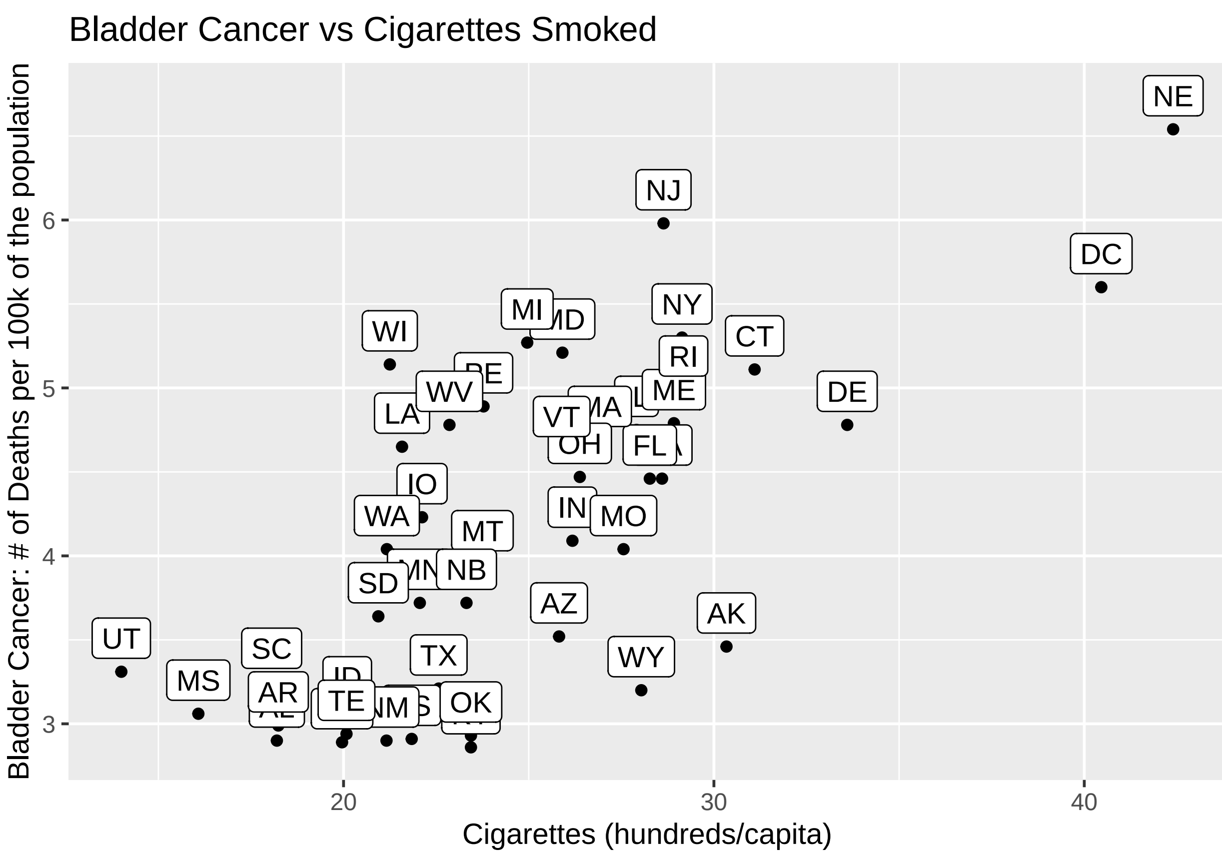
```
## [1] "We expect the demand for Ribeye to go down by -14.5% if the price goes up by 10%"
```

Exercise 2.11

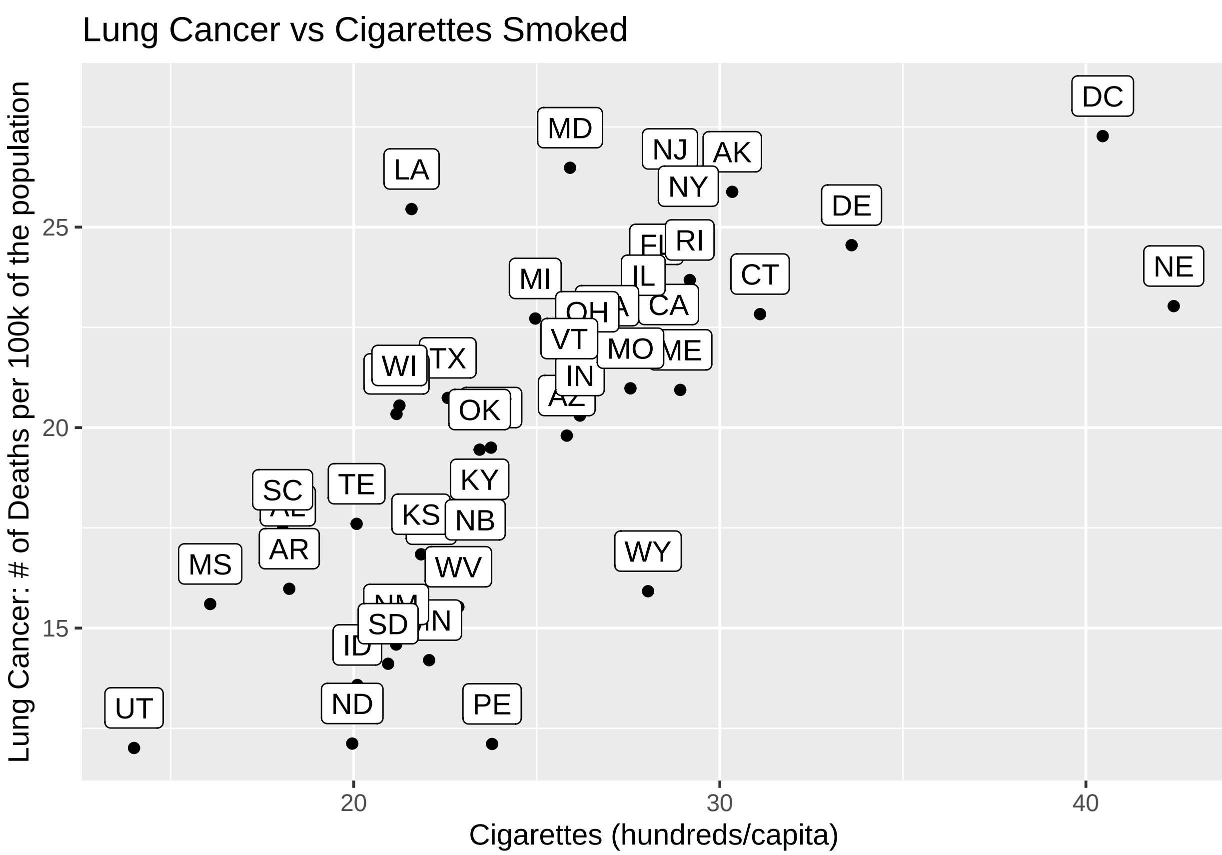
```r
#As always, we begin by importing our data
data=read.csv("/Users/kristiyan/Documents/MSiA 401 - Predictive 1/Homeworks/smoking-cancer.csv",strings
str(data)
```

```
## 'data.frame':    44 obs. of  6 variables:
##  $ STATE   : chr  "AK" "AL" "AZ" "AR" ...
##  $ Smoke   : num  30.3 18.2 25.8 18.2 28.6 ...
##  $ Bladder : num  3.46 2.9 3.52 2.99 4.46 5.11 4.78 5.6 4.46 3.08 ...
##  $ Lung    : num  25.9 17.1 19.8 16 22.1 ...
##  $ Kidney  : num  4.32 1.59 2.75 2.02 2.66 3.35 3.36 3.13 2.41 2.46 ...
##  $ Leukemia: num  4.9 6.15 6.61 6.94 7.06 7.2 6.45 7.08 6.07 6.62 ...
```

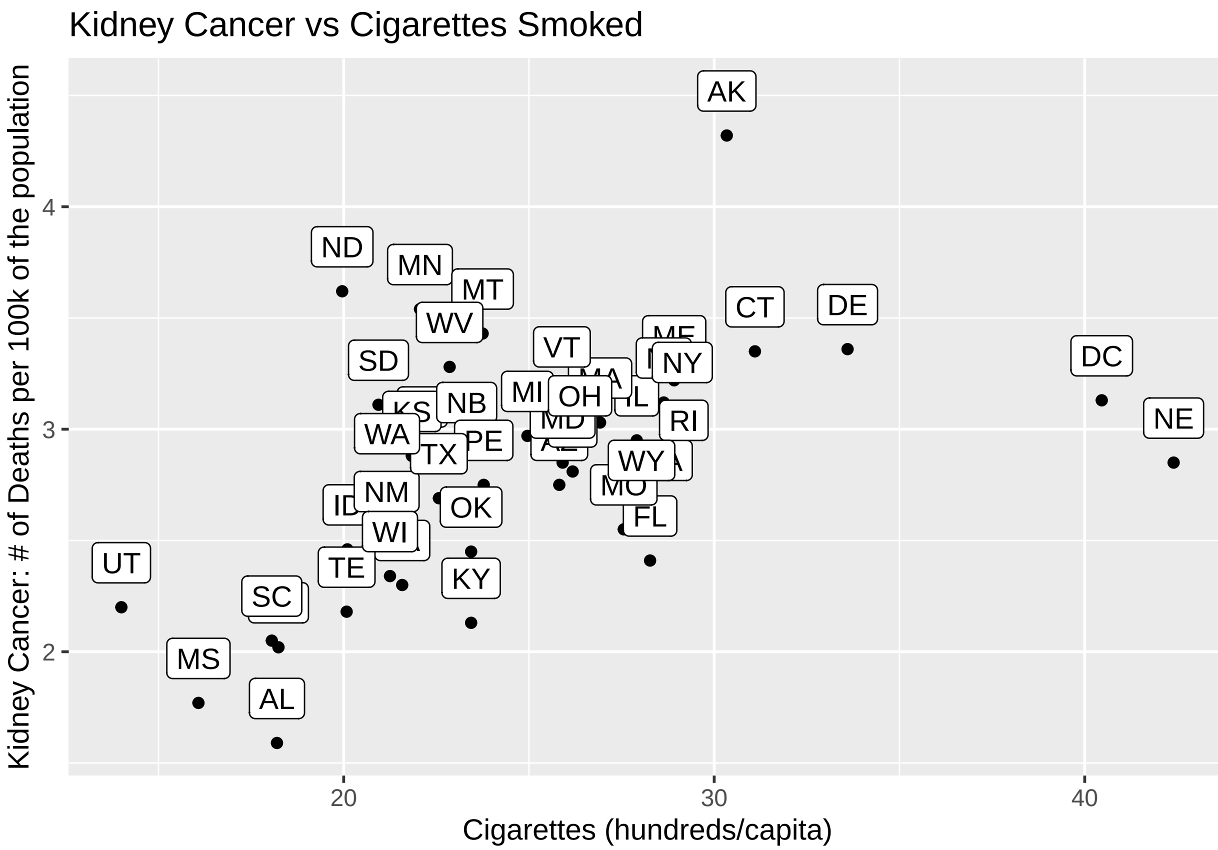
```r
#We make 4 scatterplots of Smoke with each of the types of cancer
ggplot(data = data) +
    geom_point(mapping = aes(x = Smoke, y=Bladder)) +
    labs(title="Bladder Cancer vs Cigarettes Smoked", x="Cigarettes (hundreds/capita)", y="Bladder Canc
    geom_label(mapping = aes(x = Smoke, y=Bladder, label=STATE),nudge_y=.2)
```



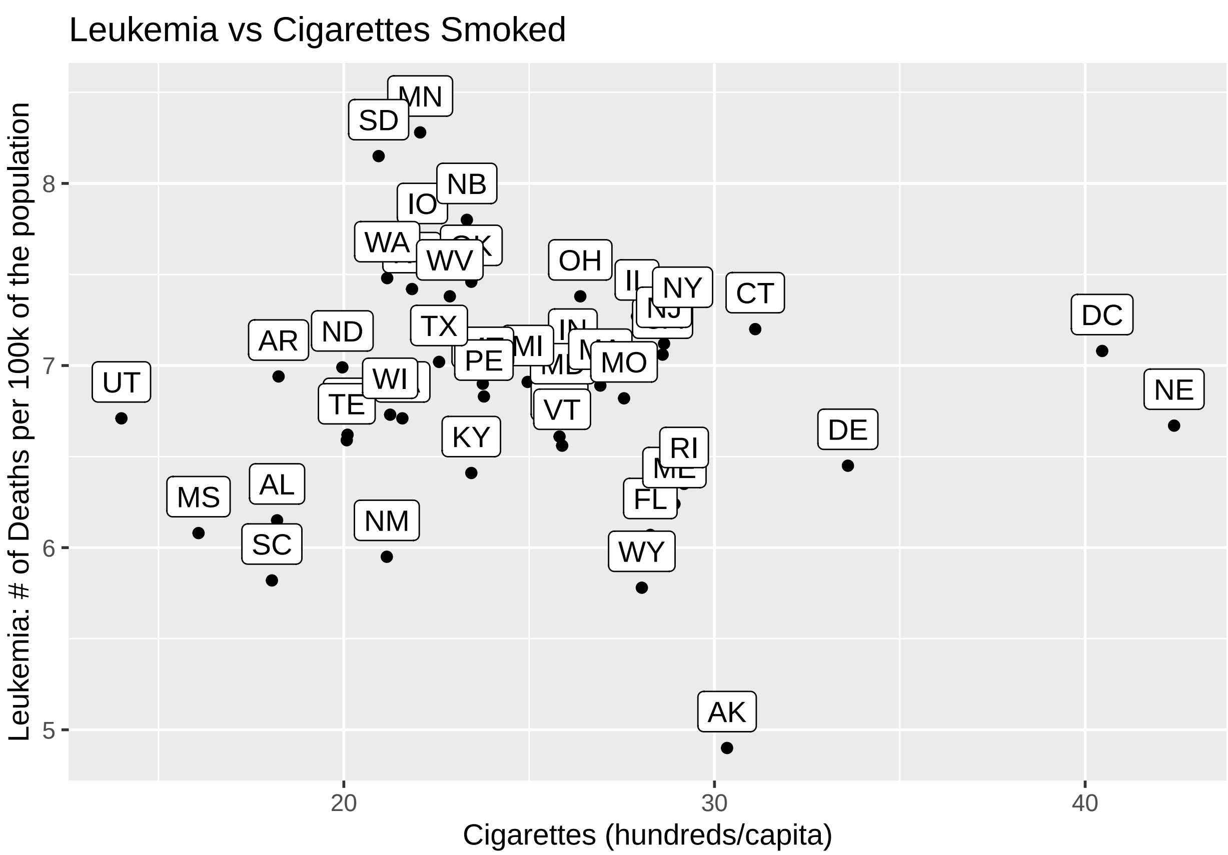

```r
ggplot(data = data) +
    geom_point(mapping = aes(x = Smoke, y=Lung)) +
    labs(title="Lung Cancer vs Cigarettes Smoked", x="Cigarettes (hundreds/capita)", y="Lung Cancer: # o
    geom_label(mapping = aes(x = Smoke, y=Lung, label=STATE),nudge_y=1)
```




```
ggplot(data = data) +
    geom_point(mapping = aes(x = Smoke, y=Kidney)) +
    labs(title="Kidney Cancer vs Cigarettes Smoked", x="Cigarettes (hundreds/capita)", y="Kidney Cancer
    geom_label(mapping = aes(x = Smoke, y=Kidney, label=STATE),nudge_y=.2)
```




```r
ggplot(data = data) +
    geom_point(mapping = aes(x = Smoke, y=Leukemia)) +
    labs(title="Leukemia vs Cigarettes Smoked", x="Cigarettes (hundreds/capita)", y="Leukemia: # of Dea
    geom_label(mapping = aes(x = Smoke, y=Leukemia, label=STATE),nudge_y=.2)
```




```
## [1] "A few observations: We see that there appear to be patterns of linear relationships between Cig
```

```r
#Next, we calculate the correlation matrix
cor_mat<-cor(data[,2:6],method="pearson")
cor_mat
```

```
##                Smoke   Bladder       Lung    Kidney    Leukemia
## Smoke     1.00000000 0.7036219  0.6974025 0.4873896 -0.06848123
## Bladder   0.70362186 1.0000000  0.6585011 0.3588140  0.16215663
## Lung      0.69740250 0.6585011  1.0000000 0.2827431 -0.15158448
## Kidney    0.48738962 0.3588140  0.2827431 1.0000000  0.18871294
## Leukemia -0.06848123 0.1621566 -0.1515845 0.1887129  1.00000000
```

```r
#And we run a correlation test for eachtype of cancer
cor_test_smoke_bladder<-cor.test(data$Smoke,data$Bladder,method="pearson")
cor_test_smoke_lung<-cor.test(data$Smoke,data$Lung,method="pearson")
cor_test_smoke_kidney<-cor.test(data$Smoke,data$Kidney,method="pearson")
cor_test_smoke_leukemia<-cor.test(data$Smoke,data$Leukemia,method="pearson")
```

```
## [1] "The t statistic for Smoke & Bladder is: 6.417"

## [1] "The t statistic for Smoke & Lung is: 6.306"

## [1] "The t statistic for Smoke & Kidney is: 3.617"

## [1] "The t statistic for Smoke & Leukemia is: -0.445"
```