Problem 1 – 2.10

Data processing

```
> # Import data
> filename = "P052.txt"
> mydata = read.table(filename, sep="\t",header = T)
```

Part a.

The covariance between the heights of the husbands and wives is:

```
> cov(mydata$Husband, mydata$Wife)
[1] 69.41294
```

Part b.

The covariance between the heights of the husbands and wives if measured in inches rather than cm is:

```
> # Convert to inches
> mydata_inches = mydata/2.54
> head(mydata_inches)
   Husband     Wife
1 73.22835 68.89764
2 70.86614 66.14173
3 62.99213 60.62992
4 73.22835 65.35433
5 64.17323 63.77953
6 67.71654 59.84252
>
> # Compute covaraince
> cov(mydata_inches$Husband, mydata_inches$Wife)
[1] 10.75903
```

Part c.

The correlation coefficient between the heights of husband and wife is:

```
> cor(mydata$Husband, mydata$Wife)
[1] 0.7633864
```

Part d.

The correlation coefficient between the heights of husband and wife if measured in inches is the same as the one if measured in cm because the correlation coefficient is scale invariant

```
> cor(mydata_inches$Husband, mydata_inches$Wife)
[1] 0.7633864
```

Part e.

The correlation is equal to 1 if every man married a woman exactly 5 centimeters shorter than him is equal to 1 because the relationship will be deterministic and can be written by an explicit equation (if you know the husband's height you know exactly the wife's height).

```
> # Change wife heights to 5 cm less than husband's
> mydata_5short = mydata
> mydata_5short$Wife = mydata$Husband -5
> head(mydata_5short)
  Husband Wife
1     186  181
2     180  175
```

```
3      160  155
4      186  181
5      163  158
6      172  167
>
> # Compute correlation coefficinet
> cor(mydata_5short$Husband, mydata_5short$Wife)
[1] 1
```

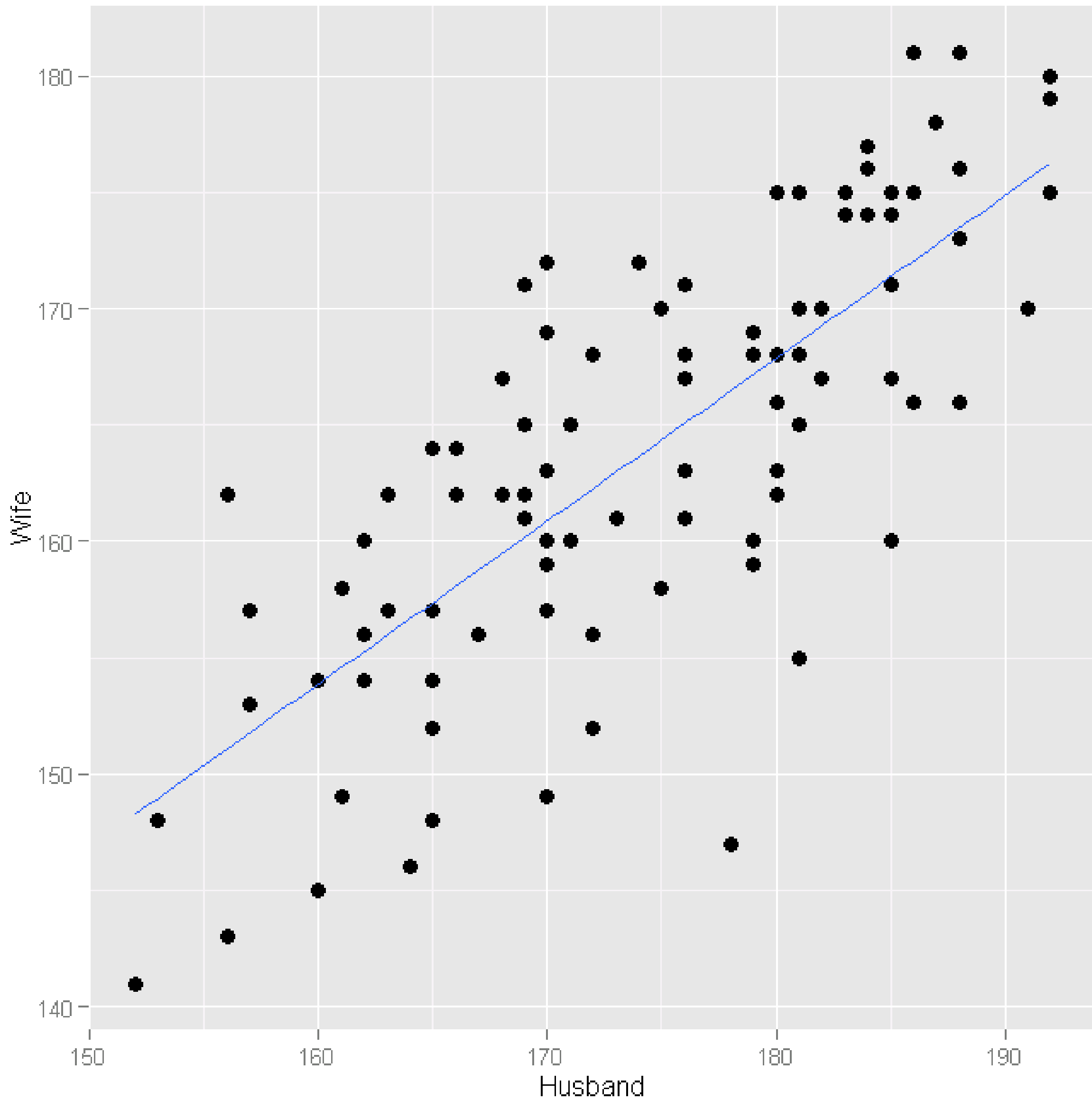
Part f.

Either variable can be used as the response variable in this case because we are trying to fit a model that relates heights and not looking for predicting anything in particular. From a social point of view, it might make sense to choose husband's height as the dependent variable because women usually have preference for men's heights and we might want to predict what the height of the husband is given the height of the women. But for this homework, assume: X = height of husband (predictor), Y = height of wife (response).

Part g.

The p-value < 0.05 for the t-test for the slope, so reject null hypothesis that the slope is zero at 0.05 level and conclude that is significantly different than zero. The conclusion for the test of the slope indicates a strong positive linear relationship between heights of wife and husband. Or in other words, the height of wife is a statistically significant predictor of the height of the husband.

```
> fit1 = lm(Wife ~ Husband, data = mydata)
>
> ggplot(mydata,aes(x=Husband, y = Wife)) + geom_point(size = 3) +
+   stat_smooth(method = 'lm', se= FALSE)
>
> summary(fit1)

Call:
lm(formula = Wife ~ Husband, data = mydata)

Residuals:
     Min       1Q   Median       3Q      Max 
-19.4685  -3.9208   0.8301   3.9538  11.1287 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 41.93015   10.66162   3.933 0.000161 ***
Husband      0.69965    0.06106  11.458  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.928 on 94 degrees of freedom
Multiple R-squared:  0.5828,	Adjusted R-squared:  0.5783 
F-statistic: 131.3 on 1 and 94 DF,  p-value: < 2.2e-16

>
> summary(fit1)$coef["Husband","Pr(>|t|)"]
[1] 1.536359e-19
>
```



Part h.

The p-value < 0.05 for the t-test for the intercept, so reject null hypothesis that the intercept is zero at 0.05 level. The conclusion for the test of the intercept indicates that is significantly different than zero.

```
> summary(fit1)$coef["(Intercept)","Pr(>|t|)"]
[1] 0.0001605824
```

Problem 2 – 2.12

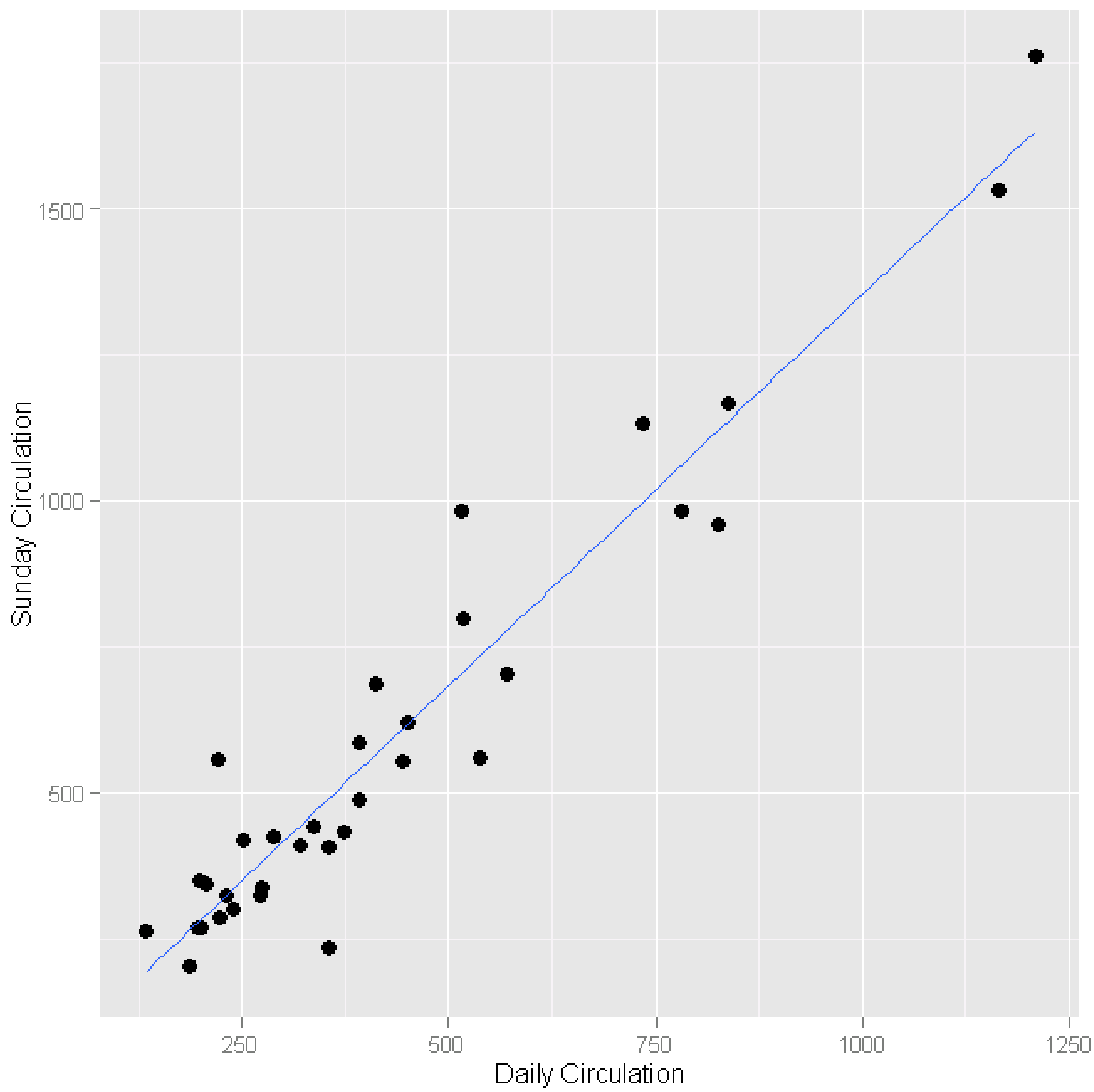
Data processing

```
> # Import data
> filename = "P054.txt"
> mydata = read.table(filename, sep="\t",header = T)
```

Part a.

The scatterplot suggests a strong linear relationship between Daily and Sunday circulation. This makes sense since people that tend to read the daily news would be interested in the news for Sunday

```
> plot1= ggplot(mydata,aes(x=Daily, y = Sunday)) + geom_point(size = 3) +
+         xlab("Daily Circulation") + ylab("Sunday Circulation") +
+         stat_smooth(method = 'lm', se= FALSE, formula=y~x)
> plot1
```



Part b.

Regression line coefficients are highlighted, intercept =13.8356 and slope = 1.3397.

```
> fit1 = lm(Sunday~Daily,data=mydata)
> summary(fit1)

Call:
lm(formula = Sunday ~ Daily, data = mydata)

Residuals:
    Min      1Q  Median      3Q     Max 
-255.19  -55.57  -20.89   62.73  278.17 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.83563   35.80401   0.386    0.702    
Daily        1.33971    0.07075  18.935   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 109.4 on 32 degrees of freedom
Multiple R-squared:  0.9181,	Adjusted R-squared:  0.9155 
F-statistic: 358.5 on 1 and 32 DF,  p-value: < 2.2e-16
```

Part c.

95% Confidence intervals for intercept and slope:

```
> confint(fit1, level=0.95)
                 2.5 %    97.5 %
(Intercept) -59.094743 86.766003
Daily         1.195594  1.483836
```

Part d.

The p-value < 0.05 for the t-test for the slope, so reject null hypothesis that the slope is zero at 0.05 level and conclude that is significantly different than zero. The conclusion for the test of the slope indicates a strong positive linear relationship between daily circulation and Sunday circulation. Or in other words, daily circulation is a statistically significant predictor of the Sunday circulation. Alternatively, the same conclusion is reached since the 95% CI for the slope does not include zero.

```
> summary(fit1)$coef["Daily","Pr(>|t|)"]
[1] 6.016802e-19
```

Note also that the the p-value > 0.05 for the t-test for the intercept, so cannot reject null hypothesis that the intercept is zero at 0.05 level. The conclusion for the test of the intercept indicates that is not significantly different than zero.

Part e.

About 92% of the variability in Sunday circulation is accounted by daily circulation.

```
> summary(fit1)$r.squared
[1] 0.9180597
```

Part f.

An interval estimate (based on 95% level) for the true average Sunday circulation of newspapers with Daily circulation of 500,000:

```
> newdata = data.frame(Daily=500)
> predict(fit1, newdata, interval="confidence",level=0.95 )
```

```
      fit      lwr     upr
1 683.693 644.1951 723.191
```

Part g.

Interval estimate (based on 95% level) for the predicted Sunday circulation of this paper:

```
> p_500
      fit      lwr      upr
1 683.693 457.3367 910.0493
```

The interval in (f) is confidence interval of the mean Sunday circulation for a daily circulation of 500K, while the interval in (g) is a prediction interval of a point-estimate or next observation of a Sunday circulation for a daily circulation of 500K.The interval in (g) is therefore wider because accounts for the mean uncertainty in the mean in addition to the scatter.

Part h.

Interval estimate for the predicted Sunday circulation with daily circulation of 2,000,000

```
> newdata = data.frame(Daily=2000)
> p_2000 = predict(fit1, newdata, interval="prediction",level=0.95 )
> p_2000
       fit      lwr      upr
1 2693.265 2373.463 3013.068
```

This interval is much wider (~41% wider) than (g) since is further away from the center of observations. It is unlikely to be accurate because a daily circulation of 2,000,000 is outside the range of observation (max is 1209).

```
> summary(mydata$Daily)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  133.2   233.0   355.2   431.0   516.6  1209.0
>
> ((p_2000[,"upr"]-p_2000[,"lwr"])/(p_500[,"upr"]-p_500[,"lwr"])-1)*100
[1] 41.28275
```

Problem 3 – 2.1

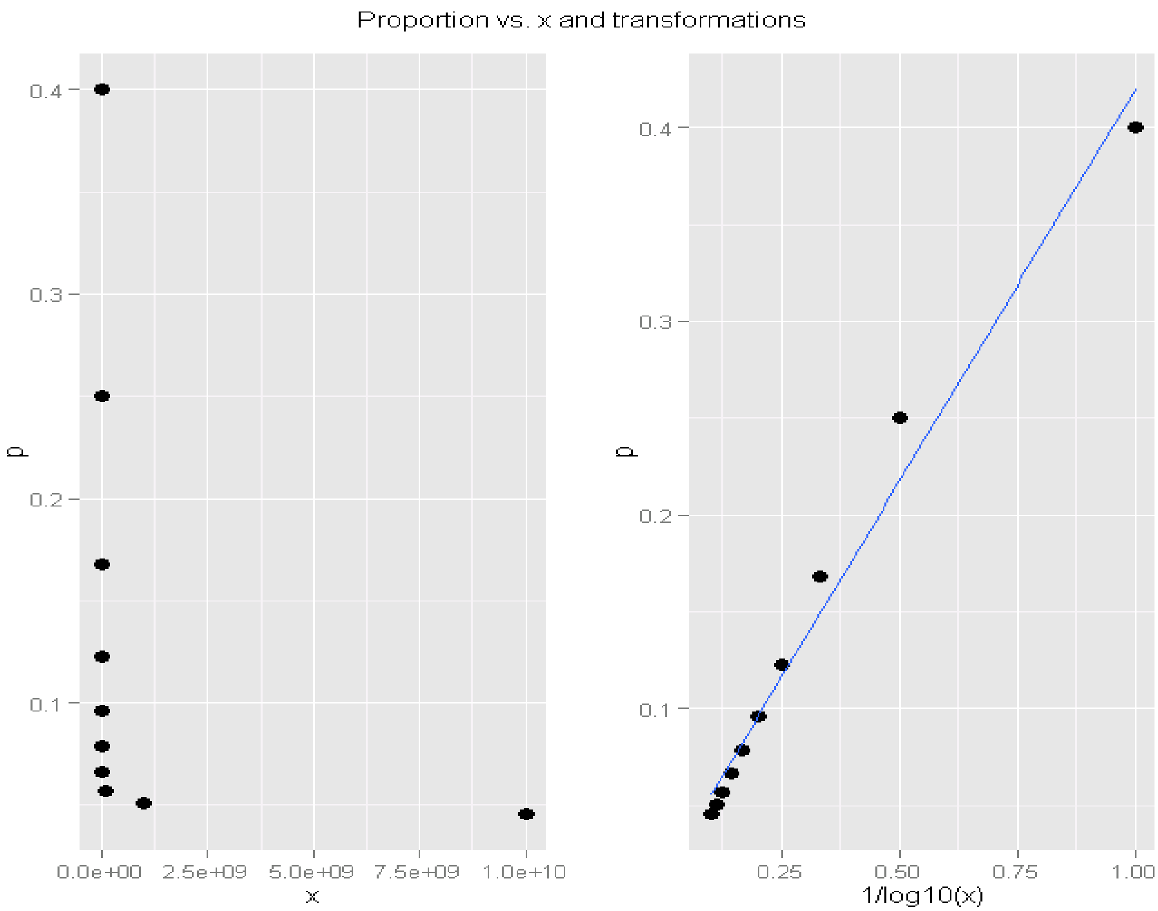
Data processing

```
> prime = data.frame(x=rep(10,10)^(1:10),
+
y=c(4,25,168,1229,9592,78498,664579,5671455,50847534,455052512))
>
> prime$p = prime$y/prime$x
> prime
       x         y          p
1  1e+01         4 0.40000000
2  1e+02        25 0.25000000
3  1e+03       168 0.16800000
4  1e+04      1229 0.12290000
5  1e+05      9592 0.09592000
6  1e+06     78498 0.07849800
7  1e+07    664579 0.06645790
8  1e+08   5671455 0.05671455
9  1e+09  50847534 0.05084753
10 1e+10 455052512 0.04550525
```

Part a.

Based on the theoretical model, the chosen transformation for linearization was chosen to be 1/log(x) as the predictor variable, and the proportion as the response variable. From the scatterplot, it can be seen that this transformation seems to be adequate since a linear relationships between proportion and 1/log(x) can be seen. Note that any base of the log can be used, in this case base 10 was chosen for convenience.

```
> plot1 =
+   ggplot(prime,aes(x=x, y = p)) +
+   geom_point(size = 3)
>
> plot2 =
+   ggplot(prime,aes(x=1/log10(x), y = p)) +
+   geom_point(size = 3)  +
+   stat_smooth(method = 'lm', se= FALSE, formula=y~x)
>
> grid.arrange(plot1,plot2,ncol=2, main = "Proportion vs. x and
transformations")
```



Part b.

The straight line after making the transformation suggests a slope = 0.404 and intercept of 0.015. The p-value < 0.05 for the t-test for the slope, so reject null hypothesis that the slope is zero at 0.05 level and conclude that is significantly different than zero. The conclusion for the test of the slope indicates a strong positive linear relationship between 1/log(x) and the proportion of primes. Note also that the p-value > 0.05 for the t-test for the intercept, so cannot reject null hypothesis that the intercept is zero at 0.05 level. The conclusion for the test of the intercept indicates that is not significantly different than zero.

```
> prime$x_transf = 1/log10(prime$x)
> prime$x_transf2 = 1/log(prime$x)
>
> ## Part b
> fit1 = lm(p~x_transf,data=prime)
>
> summary(fit1)

Call:
lm(formula = p ~ x_transf, data = prime)

Residuals:
      Min        1Q    Median        3Q       Max
-0.019444 -0.009058 -0.005143  0.005074  0.032761
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.015034   0.007748    1.94   0.0883 .  
x_transf    0.404410   0.019682   20.55 3.29e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01637 on 8 degrees of freedom
Multiple R-squared:  0.9814,	Adjusted R-squared:  0.9791 
F-statistic: 422.2 on 1 and 8 DF,  p-value: 3.294e-08
```

In order to check the slope coefficient is close to what is predicated by the prime number theorem p(x) = 1/log_e(x), express it in terms of log_10(x) using identity: log_e(x)= log_10(x)/log_10(e), so p(x) = log_10(e)*1/log_10(x). Therefore, the theory predicts a slope of log_10(e) = 0.434 and zero intercept. Since the 0.434 is included in the 95% confidence interval of the slope [0.36,0.45], we cannot reject the null hypothesis that the slope is equal to 0.434, suggesting the empirical model is in accordance with the theoretical model. According to the empirical model, the true value of the slope lies between the shown confidence interval. Note: alternatively, one can fit same model but use natural log of x, and in that case the slope will be compared to the theoretical slope = 1 (since p(x)=1*1/log(x)), reaching the same conclusion

```
> conf_b = confint(fit1, level=0.95)
> 
> b_theory = log10(exp(1))
> conf_b
                   2.5 %     97.5 %
(Intercept) -0.002833139 0.03290088
x_transf     0.359024525 0.44979582
> b_theory
[1] 0.4342945
```

Problem 3 – 2.2

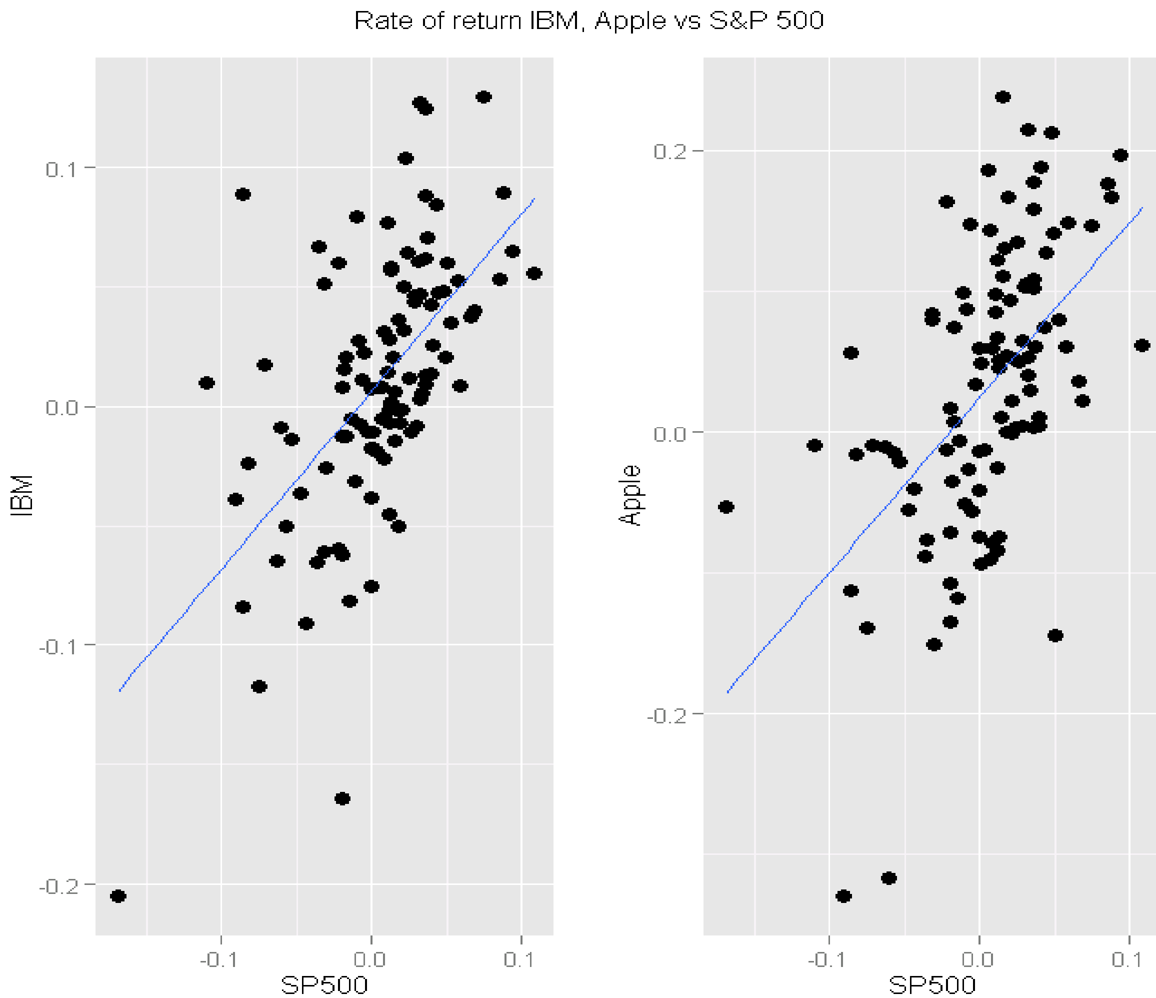
Data processing

```
> filename = "IBM_Apple_SP500.csv"
> mydata = read.csv(filename,header = T, stringsAsFactors = F)

> # Change name for SP500
> colnames(mydata)[2] = "SP500"
>
> # Convert % to numeric
> for (i in 2:4){
+   mydata[i] = as.numeric(sub("%", "", mydata[[i]]))/100
+ }
>
> # Look at data
> names(mydata)
[1] "Date"  "SP500" "IBM"   "Apple"
> head(mydata)
      Date   SP500     IBM   Apple
1 9/3/2013  0.0395  0.0422  0.0039
2 8/1/2013 -0.0313 -0.0608  0.0838
3 7/1/2013  0.0495  0.0206  0.1412
4 6/3/2013 -0.0150 -0.0813 -0.1183
5 5/1/2013  0.0208  0.0319  0.0224
6 4/1/2013  0.0181 -0.0505  0.0003
> nrow(mydata)
[1] 104
```

Part a.

For both IBM and Apple, there seems to be a strong linear relationship between their rate of return and that of the SP500. The scatter plot look very similar, so it is not clear from the plot whether IBM or Apple has a stronger linear relationship. There seems to be a little bit more variability in Apple's data with a larger range of rate of return larger.

```
> plot1=
+   ggplot(mydata,aes(x=SP500, y = IBM)) +
+   geom_point(size = 3) +
+   stat_smooth(method = 'lm', se= FALSE, formula=y~x)
>
> plot2=
+   ggplot(mydata,aes(x=SP500, y = Apple)) +
+   geom_point(size = 3) +
+   stat_smooth(method = 'lm', se= FALSE, formula=y~x)
>
> grid.arrange(plot1,plot2,ncol=2, main = "Rate of return IBM, Apple vs S&P
500")
```



Part b.

The slope = 0.774809 and intercept = 0.006416 for IBM, and slope = 1.244856 and intercept = 0.02483 for Apple. The p-values < 0.05 for the t-tests for both slopes, so reject null hypothesis that the slope are zero at 0.05 level and conclude that are significantly different than zero. The conclusion for the tests of the slopes indicates a strong positive linear relationship between IBM and SP500 rate of return, and Apple and SP500 rate of return.

The magnitude of beta(Apple) is about 67% higher than that of beta(IBM), suggesting Apple had a higher expected return relative to S&P 500 compared to IBM (for the same change in the S&P 500 rate of return, Apple had on average a larger change in its rate of return compared to IBM)

```
> lm.IBM = lm(IBM~SP500,mydata)
> lm.Apple = lm(Apple~SP500,mydata)
> summary(lm.IBM)

Call:
lm(formula = IBM ~ SP500, data = mydata)
```

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.155646 -0.024261 -0.006636  0.022188  0.146414

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.006416   0.004414   1.454    0.149
SP500       0.744809   0.098977   7.525 2.15e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04478 on 102 degrees of freedom
Multiple R-squared:  0.357,	Adjusted R-squared:  0.3507
F-statistic: 56.63 on 1 and 102 DF,  p-value: 2.15e-11

> summary(lm.Apple)

Call:
lm(formula = Apple ~ SP500, data = mydata)

Residuals:
      Min        1Q    Median        3Q       Max
-0.265378 -0.059191  0.004677  0.055363  0.194413

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.024863   0.008606   2.889  0.00472 **
SP500       1.244856   0.193007   6.450  3.8e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08732 on 102 degrees of freedom
Multiple R-squared:  0.2897,	Adjusted R-squared:  0.2827
F-statistic:  41.6 on 1 and 102 DF,  p-value: 3.799e-09


> lm.Apple$coeff["SP500"]/lm.IBM$coeff["SP500"]
   SP500
1.671377
>
```

Part c.

The sample standard deviations (SD's) of rates of return for S & P 500, IBM and Apple and the correlation matrix are shown below. The calculations (shown below) of the betas using the appropriate correlation coefficients and standard deviations agree with the betas from part (b).

```
> cor_matrix = cor(mydata[2:4])
> cor_matrix
          SP500       IBM     Apple
SP500 1.0000000 0.5974779 0.5382317
IBM   0.5974779 1.0000000 0.4147253
Apple 0.5382317 0.4147253 1.0000000
>
> cor_coeff = cor_matrix[1,2:3]
> cor_coeff
      IBM     Apple
0.5974779 0.5382317

>
> sd = apply(mydata[,2:ncol(mydata)], 2, function(x) sd(x))
```

```
> sd
     SP500        IBM      Apple
0.04457853 0.05557105 0.10310404
>
> beta = cor_coeff*sd[2:3]/sd[1]
> beta
      IBM     Apple
0.7448088 1.2448564
>
> beta.lm= c(lm.IBM$coeff['SP500'],lm.Apple$coeff['SP500'])
> names(beta.lm) = names(beta)
> beta.lm
      IBM     Apple
0.7448088 1.2448564
```

Part d.

Beta is the coefficient of the regression where it symbolizes the ratio of the return on the stock (APPLE and IBM) to return on benchmark stock (S&P 500). So a larger slope means a higher expected return. Beta is also proportional to the ratio of the standard deviation of the stock to standard deviation of the benchmark. Thus, a larger ratio, which implies higher volatility of the stock with respect to the benchmark, translates into a higher expected return. So a higher expected return is riskier and accompanied by higher volatility. In this case both Apple and IBM have similar correlations with the S&P 500, but Apple has much more variability (almost twice the sd) than IBM, which is reflected in a larger beta of Apple vs. IBM.

Problem 3 – 2.3

Data processing

```
> # Import data
> filename = "beef.csv"
> mydata = read.csv(filename,header = T, stringsAsFactors = F)
>
> # Look at data
> names(mydata)
[1] "year"          "month"         "chuck_qty"     "chuck_price"   "porter_qty"
"porter_price" "rib_qty"      "rib_price"
> head(mydata)
  year month chuck_qty chuck_price porter_qty porter_price rib_qty rib_price
1 2001     1       120        2.28         53         6.04      74      7.02
2 2001     2        76        2.61         81         5.37      79      7.16
3 2001     3       102        2.12         60         5.74      71      7.33
4 2001     4       106        2.41         65         6.93     112      7.38
5 2001     5        87        2.39         92         5.95     113      6.47
6 2001     6        94        2.11        157         5.24      89      7.14
```

Answer

Use power law for demand-price relationship in economics where y = demand , x = price is given by y = a*x^b, so linearize to ln(y) = ln(a)+b*ln(x). The coefficient beta is the price elasticity and represents the percentage change in demand due to 1% change in price (b<0 means as price increase, demand decreases).

The calculations are shown below an indicate that the price elasticities are:

```
    chuck    porter       rib
-1.368665 -2.656487 -1.446004
```

According to the book, chuck is the least expensive cut and rib eye is the most expensive, price elasticities of the three cuts are not in the expected order, which would be Porter > Rib > Chuck in terms of magnitude of elasticity. It is expected that the higher the price the more elastic since consumers are more price sensitive for expensive items and are willing to give them up more readily when prices rise compared to items that are a necessity. Also for the same percent change in price, the absolute change in price for more expensive products is greater, so it is expected to have higher impact on the demand. As expected, the sign is negative, indicating an increase in price is expected to have a reduction in demand (law of demand). All coefficients have magnitude > 1, suggesting a highly elastic demand, which makes sense since steak is not considered a necessity.

Note: in reality porter is the most expensive, so the order of the price elasticities would make sense.

A 10% increase in price for each cut would result in 13.7%, 26.6% and 14.5% reduction in demand for chuck, porter and rib cuts respectively.

```
> vars = list(chuck=names(mydata)[grepl("chuck", names(mydata))],
+             porter=names(mydata)[grepl("porter", names(mydata))],
+             rib=names(mydata)[grepl("rib", names(mydata))])
>
> models.lm  = lapply(vars, function(x) {
+   lm(substitute(log(j) ~ log(i), list(i = as.name(x[2]),j =
as.name(x[1]))), data = mydata)})
>
> summary.lm = lapply(models.lm, summary)
> summary.lm
$chuck

Call:
lm(formula = substitute(log(j) ~ log(i), list(i = as.name(x[2]),
    j = as.name(x[1]))), data = mydata)

Residuals:
     Min       1Q   Median       3Q      Max
-0.32463 -0.12036 -0.01714  0.09430  0.49725

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        5.8899     0.2871  20.513  < 2e-16 ***
log(chuck_price)  -1.3687     0.3199  -4.278 9.44e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1812 on 46 degrees of freedom
Multiple R-squared:  0.2846,	Adjusted R-squared:  0.2691
F-statistic:  18.3 on 1 and 46 DF,  p-value: 9.441e-05


$porter

Call:
lm(formula = substitute(log(j) ~ log(i), list(i = as.name(x[2]),
    j = as.name(x[1]))), data = mydata)

Residuals:
     Min       1Q   Median       3Q      Max
-0.57655 -0.23544  0.00317  0.23511  0.49991

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         9.1123     0.5136  17.742  < 2e-16 ***
log(porter_price)  -2.6565     0.2752  -9.654 1.23e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.283 on 46 degrees of freedom
Multiple R-squared:  0.6695,	Adjusted R-squared:  0.6624
F-statistic:  93.2 on 1 and 46 DF,  p-value: 1.233e-12


$rib

Call:
lm(formula = substitute(log(j) ~ log(i), list(i = as.name(x[2]),
    j = as.name(x[1]))), data = mydata)

Residuals:
     Min       1Q   Median       3Q      Max
-0.54075 -0.21801  0.03995  0.20328  0.70950
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)       7.6627     0.7537  10.167 2.39e-13 ***
log(rib_price)   -1.4460     0.3731  -3.876 0.000335 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2943 on 46 degrees of freedom
Multiple R-squared:  0.2462,	Adjusted R-squared:  0.2298 
F-statistic: 15.02 on 1 and 46 DF,  p-value: 0.0003352


> 
> coeff = c(summary.lm$chuck$coeff["log(chuck_price)","Estimate"],
+           summary.lm$porter$coeff["log(porter_price)","Estimate"],
+           summary.lm$rib$coeff["log(rib_price)","Estimate"])
> 
> names(coeff)= names(summary.lm)
> 
> # Price elasticities estimates
> coeff
    chuck    porter       rib 
-1.368665 -2.656487 -1.446004 

# % change if increase 10% in price
> coeff*10
    chuck    porter       rib 
-13.68665 -26.56487 -14.46004 
```

Problem 4 – 2.4

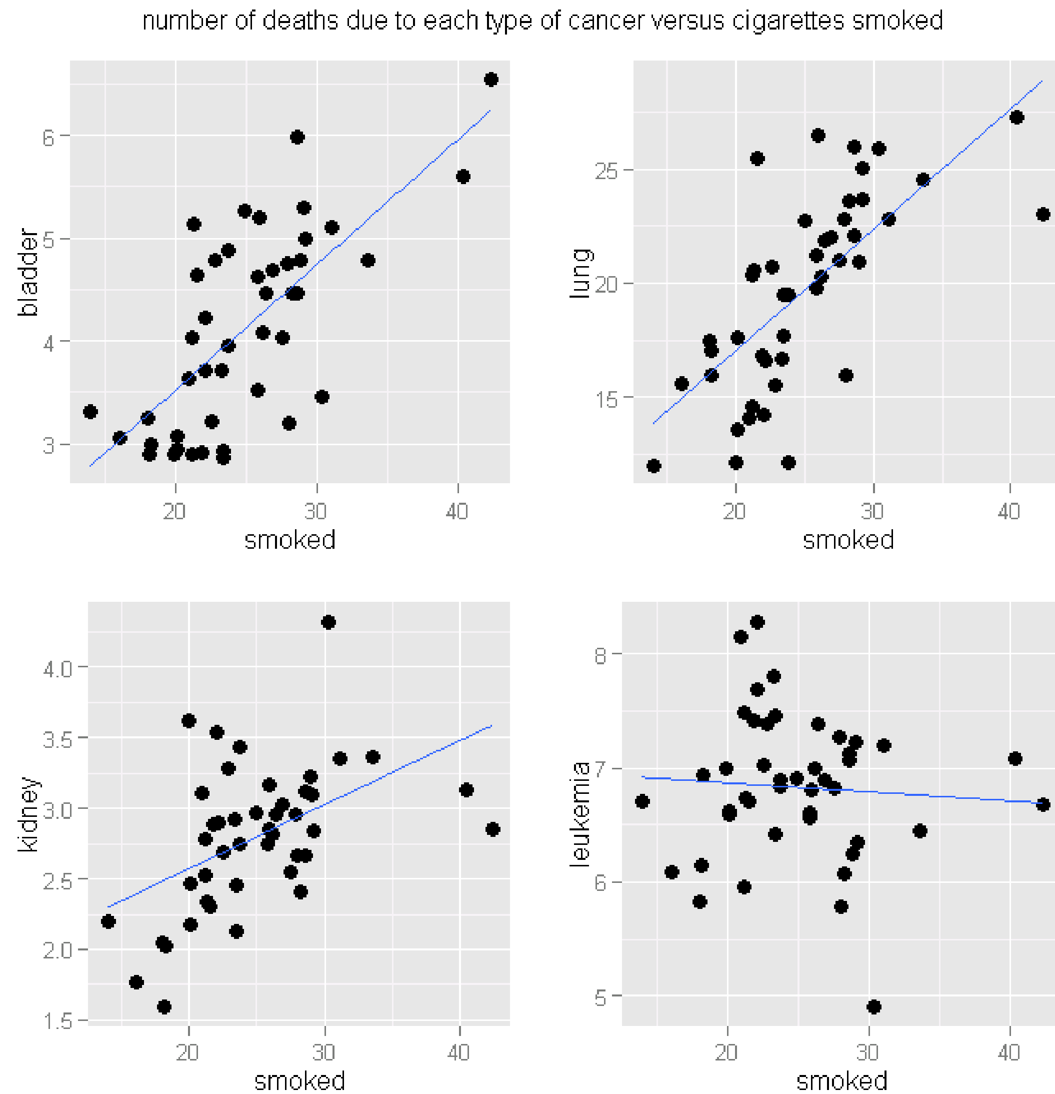
Data processing

```
> # Import data
> filename = "Smoking-Cancer Data.xlsx"
> mydata = readWorksheet(loadWorkbook(filename),sheet=1)
> names(mydata) = c("state","smoked","bladder","lung","kidney","leukemia")
>
> # Look at data
> names(mydata)
[1] "state"    "smoked"   "bladder"  "lung"     "kidney"   "leukemia"
> head(mydata)
  state smoked bladder  lung kidney leukemia
1    AK  30.34    3.46 25.88   4.32     4.90
2    AL  18.20    2.90 17.05   1.59     6.15
3    AZ  25.82    3.52 19.80   2.75     6.61
4    AR  18.24    2.99 15.98   2.02     6.94
5    CA  28.60    4.46 22.07   2.66     7.06
6    CT  31.10    5.11 22.83   3.35     7.20
```

Scatter plot

Scatter plots are shown in the next page and suggest that bladder and lung cancer might have a possible linear relationship with cigarettes smoked. Leukemia doesn't exhibit any linear relationship with cigarettes smokes, while kidney shoes a non-linear relationship general, but might exhibit a linear relationship if a few extreme points are removed. The scatter plot also suggest outliers, especially for the data for kidney cancer. A boxplot analysis below shows the possible outliers.

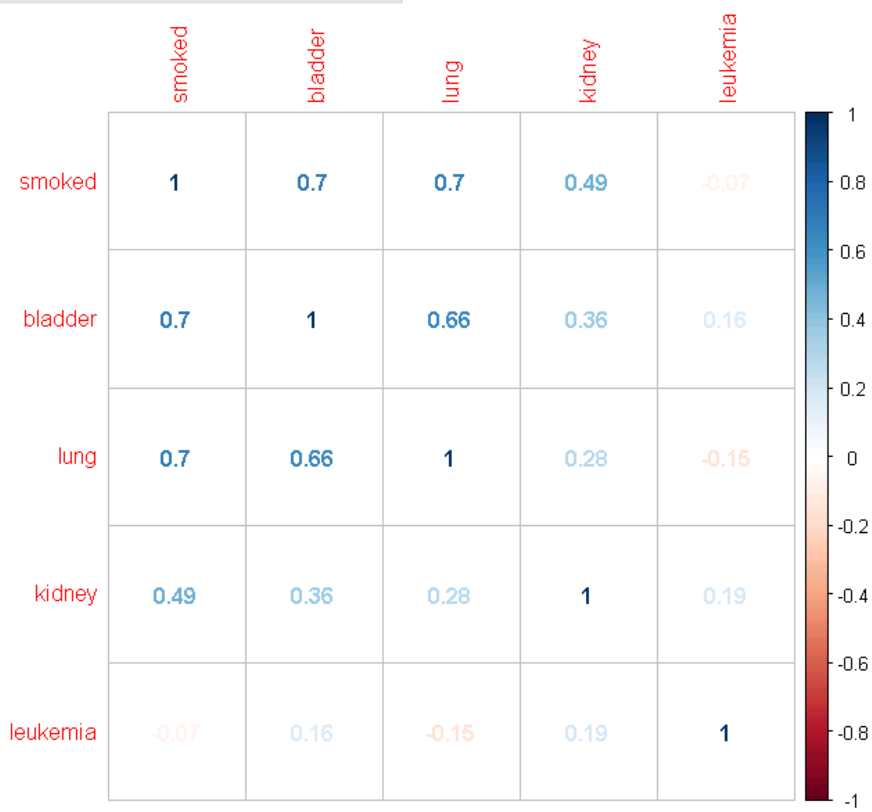
```
> plot1 =
+   ggplot(mydata,aes(x=smoked, y = bladder)) +
+   geom_point(size = 3) +
+   stat_smooth(method = 'lm', se= FALSE)
>
> plot2 =
+   ggplot(mydata,aes(x=smoked, y = lung)) +
+   geom_point(size = 3) +
+   stat_smooth(method = 'lm', se= FALSE)
>
> plot3 =
+   ggplot(mydata,aes(x=smoked, y = kidney)) +
+   geom_point(size = 3)  +
+   stat_smooth(method = 'lm', se= FALSE)
>
> plot4 =
+   ggplot(mydata,aes(x=smoked, y = leukemia)) +
+   geom_point(size = 3)  +
+   stat_smooth(method = 'lm', se= FALSE)
>
> grid.arrange(plot1,plot2,plot3,plot4,ncol=2,
+   main = "number of deaths due to each type of cancer versus cigarettes
smoked")

> boxplot(mydata$leukemia, main="xx")$out
[1] 4.9
> boxplot(mydata$bladder, main="xx")$out
numeric(0)
> boxplot(mydata$kidney, main="xx")$out
[1] 4.32
> boxplot(mydata$lung, main="xx")$out
numeric(0)
```



Correlation

A correlation matrix is shown below. It appears that lung and bladder cancer deaths are most highly positively correlated (linearly) with cigarette smoking. Kidney cancer shows some correlation, and leukemia does not exhibit any correlation with smoking. This is in agreement with the scatter plot. The analysis shows the p-value for the correlations, indicating that lung, bladder and kidney are statistically significant correlations with smoking, while leukemia is not.

```
> cor_p = cor(mydata[-1],method="pearson")
> cor_s = cor(mydata[-1],method="spearman")
> cor_p
               smoked   bladder       lung    kidney   leukemia
smoked     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
> cor_s
               smoked   bladder        lung    kidney    leukemia
smoked     1.00000000 0.6696271  0.75020262 0.5134097 -0.02452691
bladder    0.66962713 1.0000000  0.66058151 0.4400620  0.18246220
lung       0.75020262 0.6605815  1.00000000 0.2688187 -0.07963916
kidney     0.51340969 0.4400620  0.26881872 1.0000000  0.38024248
leukemia  -0.02452691 0.1824622 -0.07963916 0.3802425  1.00000000
>
> corrplot(cor_p,method="number")
> rcorr(as.matrix(mydata[-1]))
          smoked bladder  lung kidney leukemia
smoked      1.00    0.70  0.70   0.49    -0.07
bladder     0.70    1.00  0.66   0.36     0.16
lung        0.70    0.66  1.00   0.28    -0.15
kidney      0.49    0.36  0.28   1.00     0.19
leukemia   -0.07    0.16 -0.15   0.19     1.00
n= 44
P
          smoked bladder lung   kidney leukemia
smoked           0.0000  0.0000 0.0008 0.6587
bladder   0.0000         0.0000 0.0168 0.2930
lung      0.0000 0.0000         0.0629 0.3260
kidney    0.0008 0.0168  0.0629        0.2199
leukemia  0.6587 0.2930  0.3260 0.2199
```



R-code

```
#### Homework 1
#### Text-Book Problems: 2.10, 2.12
#### My Book Problems: 2.1, 2.2, 2.3, 2.4

#### Setup #################################################

# Install packages if needed
# install.packages("ggplot2")
# install.packages("grid")
# install.packages("gridExtra")
# install.packages("XLConnect")
# install.packages("corrplot")
# install.packages("Hmisc")


# Load packages
library(ggplot2)
library(grid)
library(gridExtra)
library(XLConnect)
library(corrplot)
library(Hmisc)

# My PC
# main = "\\\\nas1/labuser169"

# Aginity
main = "\\\\nas1/labuser169"
course = "MSIA_401_Statistical Methods for Data Mining"
assignment = "Homework"
setwd(file.path(main,course, assignment))

#### Problem 1 - 2.10 #################################################

# Import data
filename = "P052.txt"
mydata = read.table(filename, sep="\t",header = T)

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

## Part a
```

```
# Compute covariance
cov(mydata$Husband, mydata$Wife)

## Part b

# Convert to inches
mydata_inches = mydata/2.54
head(mydata_inches)

# Compute covaraince
cov(mydata_inches$Husband, mydata_inches$Wife)

## Part c

# Compute correlation coefficinet
cor(mydata$Husband, mydata$Wife)

## Part d

# Compute correlation coefficient
cor(mydata_inches$Husband, mydata_inches$Wife)

## Part e

# Change wife heights to 5 cm less than husband's
mydata_5short = mydata
mydata_5short$Wife = mydata$Husband -5
head(mydata_5short)

# Compute correlation coefficinet
cor(mydata_5short$Husband, mydata_5short$Wife)

## Part f

# Ans: Either variable can be used as the response variable in this case
#     because we are trying to fit a model that relates heights and not looking for predicting
#     anything in particular
#     Let X = height of husband (predicotr), Y = height of wife (reponse) for this model

## Part g

fit1 = lm(Wife ~ Husband, data = mydata)

ggplot(mydata,aes(x=Husband, y = Wife)) + geom_point(size = 3) +
  stat_smooth(method = 'lm', se= FALSE)

summary(fit1)
```

```
summary(fit1)$coef["Husband","Pr(>|t|)"]

# Ans: p-value < 0.05, reject null hypothesis that the slope is zero at 0.05 level

## Part h
summary(fit1)$coef["(Intercept)","Pr(>|t|)"]

# Ans: p-value < 0.05, reject null hypothesis that the intercept is zero at 0.05 level


#### Problem 2 - 2.12 ##################################################
# Import data
filename = "P054.txt"
mydata = read.table(filename, sep="\t",header = T)

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

## Part a

plot1= ggplot(mydata,aes(x=Daily, y = Sunday)) + geom_point(size = 3) +
    xlab("Daily Circulation") + ylab("Sunday Circulation") +
    stat_smooth(method = 'lm', se= FALSE, formula=y~x)

plot1

# Ans: Yes the scatterplot suggests a strong linear relationship between Daily
#     and Sunday circulation. This makes sense since people that tend to read
#     the daily news would be interested in the news for Sunday

## Part b

fit1 = lm(Sunday~Daily,data=mydata)
summary(fit1)


## Part c

confint(fit1, level=0.95)

## Part d
summary(fit1)$coef["Daily","Pr(>|t|)"]

# Ans: p-value < 0.05, reject null hypothesis that the slope is zero at 0.05 level
#     The conclusion for the test of the slope indicates a strong positive linear
```

```
#     relationship between sunday and daily circulation. Or in other words, daily circulation
#     is a statsically significant predictor of sunday circulation.
#     Alternatively,the same conclusion is reached since the 95% CI for the slope does not include zero.

## Part e
summary(fit1)$r.squared

# Ans: about 92% of the variability in Sunday circulation is accounted by daily circulation

## Part f

newdata = data.frame(Daily=500)
predict(fit1, newdata, interval="confidence",level=0.95 )

## Part g
p_500 = predict(fit1, newdata, interval="prediction",level=0.95 )
p_500


# Ans: The interval in (f) is confidence interval of the mean Sunday circulation for
#     a daily circulation of 500K, while the interval in (g) is a prediction interval
#     of a point-estimate or next observation of a Sunday circulation for a daily circulation of 500K.
#     The interval in (g) is therefore wider because accounts for the mean uncertainty in the mean
#     in addition to the scatter.

## Part h
newdata = data.frame(Daily=2000)
p_2000 = predict(fit1, newdata, interval="prediction",level=0.95 )
p_2000
summary(mydata$Daily)

((p_2000[,"upr"]-p_2000[,"lwr"])/(p_500[,"upr"]-p_500[,"lwr"])-1)*100

# Ans: This interval is much wider (~41% wider) than g scince is further away from
#     the center of observations.
#     It is unlikely to be accurate because a daily circulation of 2,000,000  is outside
#     the range of observation (max is 1209).

#### Problem 3 - 2.1 ################################################

prime = data.frame(x=rep(10,10)^(1:10),
           y=c(4,25,168,1229,9592,78498,664579,5671455,50847534,455052512))

prime$p = prime$y/prime$x
prime

## Part a
```

```
plot1 =
  ggplot(prime,aes(x=x, y = p)) +
  geom_point(size = 3)

plot2 =
  ggplot(prime,aes(x=1/log10(x), y = p)) +
  geom_point(size = 3)  +
  stat_smooth(method = 'lm', se= FALSE, formula=y~x)

grid.arrange(plot1,plot2,ncol=2, main = "Proportion vs. x and transformations")

prime$x_transf = 1/log10(prime$x)
prime$x_transf2 = 1/log(prime$x)

# Ans: Linearizing transformation: x= 1/log10(x)

## Part b
fit1 = lm(p~x_transf,data=prime)

summary(fit1)
conf_b = confint(fit1, level=0.95)


# My model: p(x) = b*1/log_10(x) + a, a~0
# Theory: p(x) = 1/log_e(x)
# Express Theory in terms of log_10(x) -> using identity: log_e(x)= log_10(x)/log_10(e)
# so p(x) = log_10(e)*1/log_10(x)

b_theory = log10(exp(1))

conf_b
b_theory

# Note that both the intercept = 0 and theoretical slope = 0.434 fall inside their respective confidence
intervals,
# so one cannot reject the null that intercept = 0 and slope = 0.43. This implies that the empirical model
# matches the theoretical model
#
# Note: alternatively, one can fit same model but use natural log of x, and in that case the slope will
# be compared to the theoretical slope = 1 (since p(x)=1*1/log(x)), reaching the same conclusion

fit2 = lm(p~x_transf2,data=prime)

summary(fit2)
conf_b = confint(fit2, level=0.95)
conf_b
```

```
#### Problem 4 - 2.2 ############################################

# Import data
filename = "IBM_Apple_SP500.csv"
mydata = read.csv(filename,header = T, stringsAsFactors = F)

# Change name for SP500
colnames(mydata)[2] = "SP500"

# Convert % to numeric
for (i in 2:4){
  mydata[i] = as.numeric(sub("%", "", mydata[[i]]))/100
}

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

## Part a
plot1=
  ggplot(mydata,aes(x=SP500, y = IBM)) +
  geom_point(size = 3) +
  stat_smooth(method = 'lm', se= FALSE, formula=y~x)

plot2=
  ggplot(mydata,aes(x=SP500, y = Apple)) +
  geom_point(size = 3) +
  stat_smooth(method = 'lm', se= FALSE, formula=y~x)

grid.arrange(plot1,plot2,ncol=2, main = "Rate of return IBM, Apple vs S&P 500")

# Ans: For both IBM and Apple, there seems to be a strong linear relationship betwen their rate
#      of return and that of the SP500. The scatter plot look very similar, so it is not clear
#      from the plot whether IBM or Apple has a stronger linear linear relationship. There seems
#      to be a little bit more variability in Apple's data.

## Part b
lm.IBM = lm(IBM~SP500,mydata)
lm.Apple = lm(Apple~SP500,mydata)
summary(lm.IBM)
summary(lm.Apple)

lm.Apple$coeff["SP500"]/lm.IBM$coeff["SP500"]

# ANS: The magintue of beta(Apple) is about 67% higher than that of beta(IBM), suggesting
#      Apple had a higher expected return relative to S&P 500 compared to IBM (for the same change
```

```
#     in the S&P 500 rate of return, Apple had on average a larger change in its rate of return
#     compared to IBM)

## Part c

cor_matrix = cor(mydata[2:4])
cor_matrix

cor_coeff = cor_matrix[1,2:3]
cor_coeff

sd = apply(mydata[,2:ncol(mydata)], 2, function(x) sd(x))
sd

beta = cor_coeff*sd[2:3]/sd[1]
beta

beta.lm= c(lm.IBM$coeff['SP500'],lm.Apple$coeff['SP500'])
names(beta.lm) = names(beta)
beta.lm

# ANS: beta is the coefficient of the regression where ite symbolizes the ratio of the return
#     on the stock (APPLE and IBM) to return on benchmark stock (S&P 500). So a larger slope
#     menas a higher expected return. Beta is also proportional to the ratio of the
#     standard deviation of the stock to standard devation of the benchmark.Thus, a larger
#     ratio, which implies higher volatility of the stock with respect to the benchmark,
#     translates into a higher expected return. So a higher expected return is riskier and
#     accompanied by higher volatility.
#     In this case both Apple and IBM have similar correlations with the S&P 500, but Apple
#     has much more variablity (almost twice the sd) than IBM, which is reflected in a larger
#     beta of Apple vs. IBM.

#### Problem 5 - 2.3 ##############################################

# Import data
filename = "beef.csv"
mydata = read.csv(filename,header = T, stringsAsFactors = F)

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

## Estimate price elasticities


# Power law for demand-price relationship in economics
```

```
# y = demand , x = price
# y = a*x^b -> ln(y) = ln(a)+b*ln(x)
# 100b = perecentage change in demand due to 1% change in price = price elasticity
# b<0 means price increaes, demand decreases.

vars = list(chuck=names(mydata)[grepl("chuck", names(mydata))],
        porter=names(mydata)[grepl("porter", names(mydata))],
        rib=names(mydata)[grepl("rib", names(mydata))])

models.lm  = lapply(vars, function(x) {
  lm(substitute(log(j) ~ log(i), list(i = as.name(x[2]),j = as.name(x[1]))), data = mydata)})

summary.lm = lapply(models.lm, summary)
summary.lm

coeff = c(summary.lm$chuck$coeff["log(chuck_price)","Estimate"],
       summary.lm$porter$coeff["log(porter_price)","Estimate"],
       summary.lm$rib$coeff["log(rib_price)","Estimate"])

names(coeff)= names(summary.lm)

# Price elasticities estimates
coeff

# ANS: Order in terms of price/quality: Porter > Rib > Chuck
#     Order in terms of magnitude of elasticity: Porter > Rib > Chuck
#     The order makes sense since the higher the price the more elastic since
#     consumers are more price sensitive for expensive items and are willing to give them up more
#     readily when prices rise compared to items that are a necessity. Also for the same percent
#     change in price, the absolute change in price for more expensive products is greater, so
#     it is expected to have higher impact on the demand.
#     As expected, the sign is negative, indicating an increase in price
#     is expected to have a reduction in demand (law of demand)
#     All coefficients have magnitude > 1, suggesting a highly elastic demand,
#     which makes sense since steak is not considered a necessity.
#
#     100b = perecentage change in demand due to 1% change in price = price elasticity

#     price elasticity higher for cheaper items?
#     rib eye more expensive so order doesn;t make sense
#     multiply by 100 or 10

# % change if increase 10% in price
coeff*10

#    # A 10% increase in price would result in 13.7%, 25.7% and 14.5% reduction in demand
#    for chuck, porer and rib cuts respectively
```

```
#### Problem 6 - 2.4 ################################################

# Import data
filename = "Smoking-Cancer Data.xlsx"
mydata = readWorksheet(loadWorkbook(filename),sheet=1)
names(mydata) = c("state","smoked","bladder","lung","kidney","leukemia")

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

## Scatter plots

plot1 =
  ggplot(mydata,aes(x=smoked, y = bladder)) +
  geom_point(size = 3) +
  stat_smooth(method = 'lm', se= FALSE)

plot2 =
  ggplot(mydata,aes(x=smoked, y = lung)) +
  geom_point(size = 3) +
  stat_smooth(method = 'lm', se= FALSE)

plot3 =
  ggplot(mydata,aes(x=smoked, y = kidney)) +
  geom_point(size = 3)  +
  stat_smooth(method = 'lm', se= FALSE)

plot4 =
  ggplot(mydata,aes(x=smoked, y = leukemia)) +
  geom_point(size = 3)  +
  stat_smooth(method = 'lm', se= FALSE)

grid.arrange(plot1,plot2,plot3,plot4,ncol=2,
  main = "number of deaths due to each type of cancer versus cigarettes smoked")

# bladder and lung look a little bit linear
# leukemia doesn't look linear
# kidney looks linear except for a few outliers


# outliers

boxplot(mydata$leukemia, main="xx")$out
boxplot(mydata$bladder, main="xx")$out
```

```
boxplot(mydata$kidney, main="xx")$out
boxplot(mydata$lung, main="xx")$out


## Correlation Analysis
cor_p = cor(mydata[-1],method="pearson")
cor_s = cor(mydata[-1],method="spearman")
cor_p
cor_s

corrplot(cor_p,method="number")
rcorr(as.matrix(mydata[-1]))
```