Problem Set 4
Mgmt 237Q: Econometrics
Professor Rossi

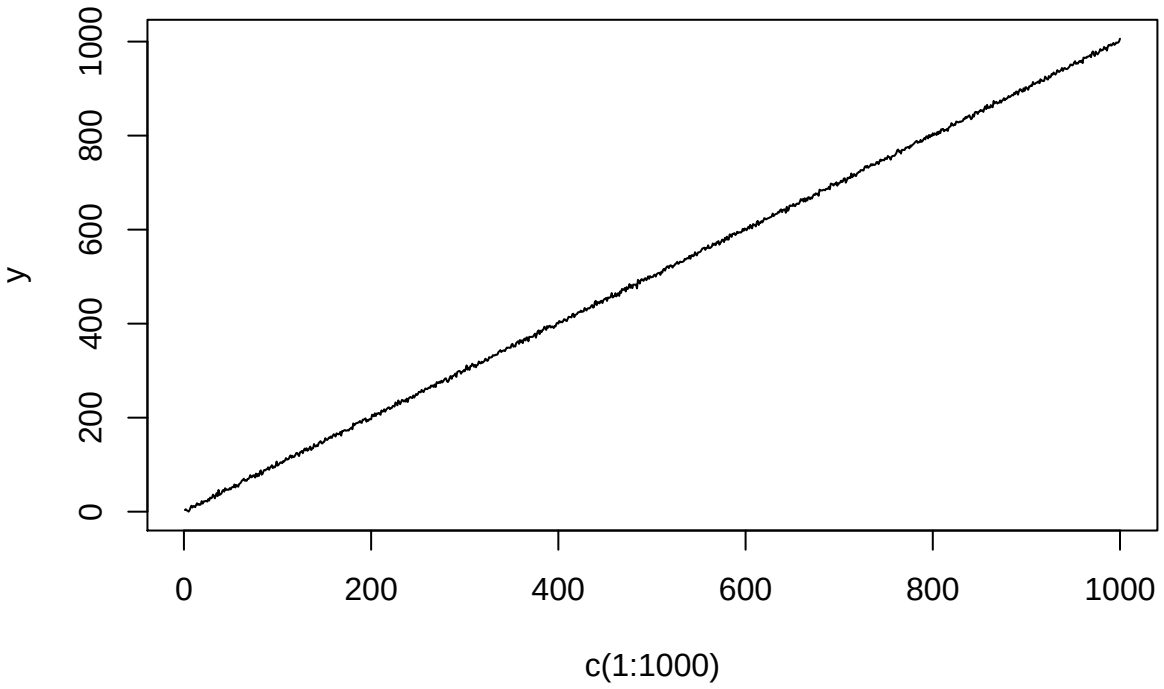
This problem set is designed to review material on time series and advanced regression topics. Include both your R code and output in your answers.

Question 1

Simulate data for the following models and provide a plot of each:

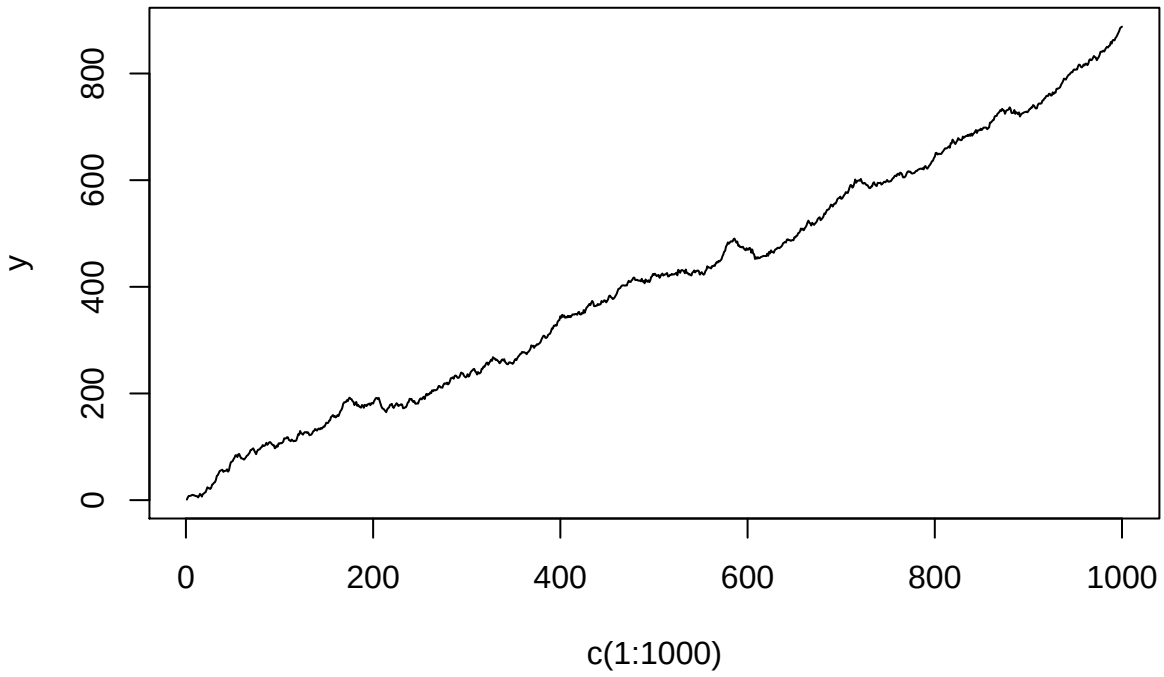
a. A linear time trend: $y_t = \alpha + \beta t + \varepsilon_t$

```
alpha = 1
beta = 1
y = alpha + beta * c(1:1000) + rnorm(1000,0,3)
plot(c(1:1000),y,cex = 0.1,type = "l")
```



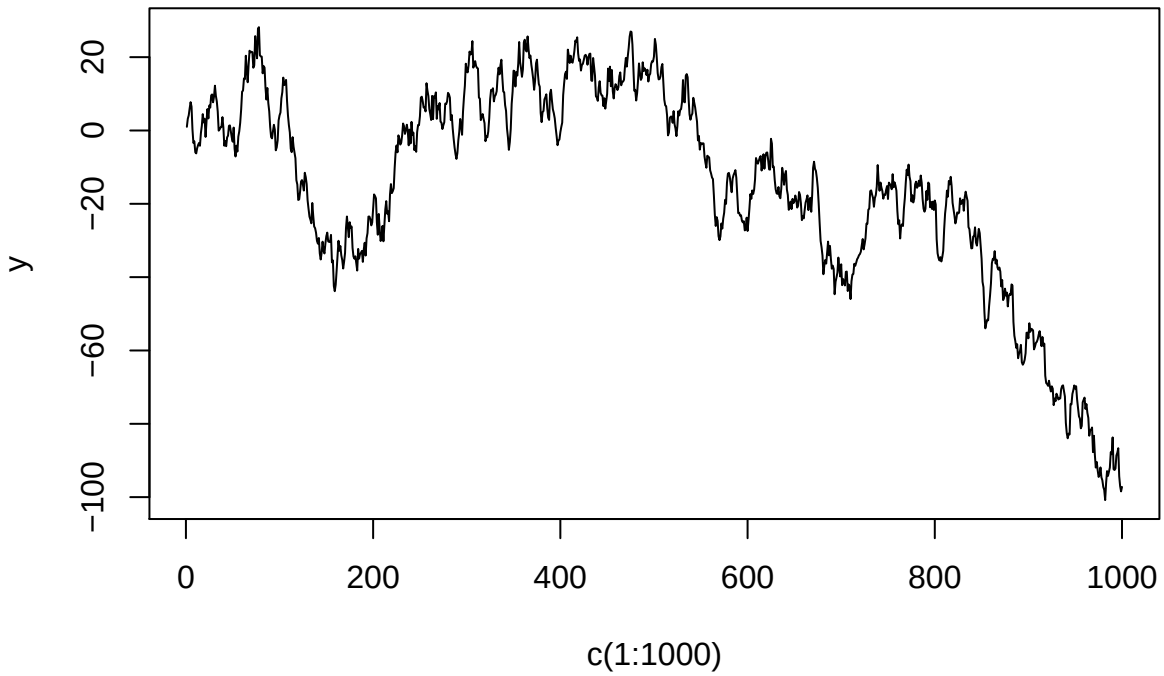
b. An AR(1): $y_t = \alpha + \beta y_{t-1} + \varepsilon_t$

```
y = vector()
y[1] = 1
for (i in c(2:1000)){
  y[i] = alpha + beta * y[i-1] + rnorm(1,0,3)
}
plot(c(1:1000),y,cex = 0.1,type = "l")
```



c. A random walk: $y_t = y_{t-1} + \varepsilon_t$

```
y = vector()
y[1] = 1
for (i in c(2:1000)){
  y[i] = y[i-1] + rnorm(1,0,3)
}
plot(c(1:1000),y,type = "l",cex= 0.1)
```



Question 2

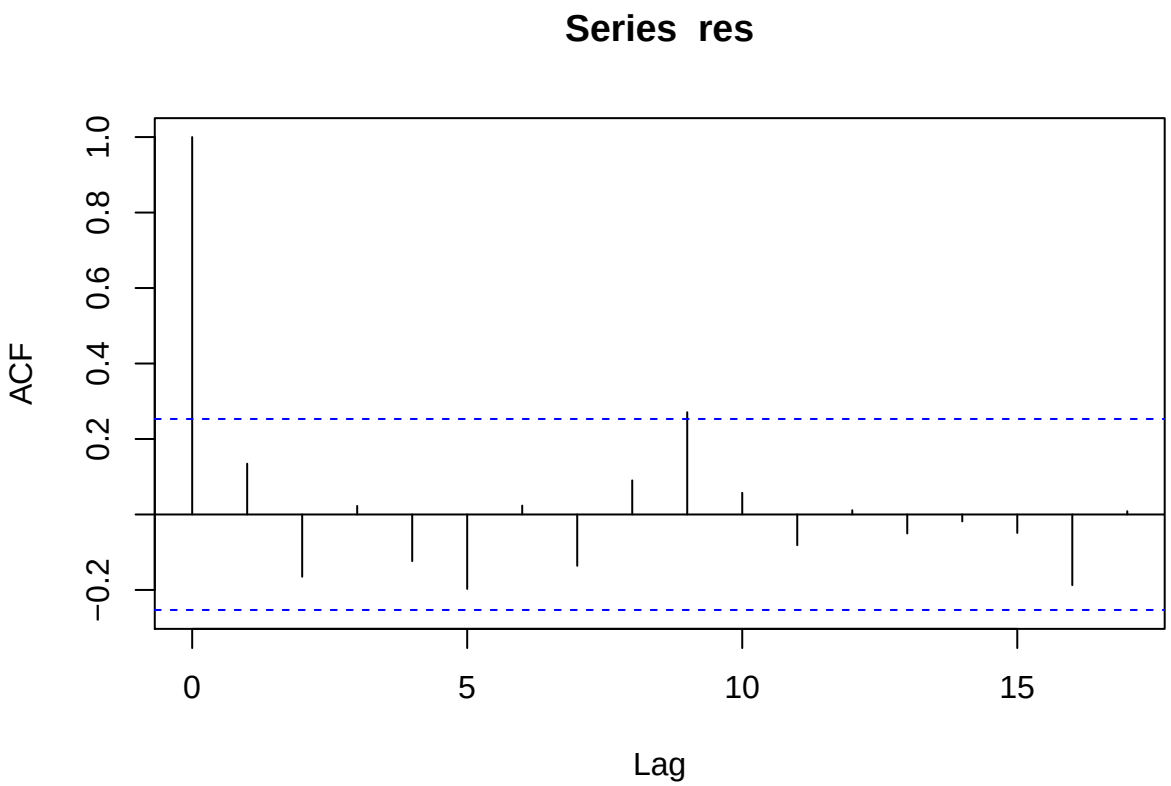
a. Using the `beerprod` data from the `DataAnalytics` package, regress beer production on its 1-period, 6-period, and 12-period lags. This should be one regression, not three separate regressions.

```
library("DataAnalytics")
data("beerprod")
y = beerprod$b_prod
regression = lm(b_prod ~ back(b_prod) + back(b_prod,6) + back(b_prod,12),data = beerprod)
regression
```

```
##
## Call:
## lm(formula = b_prod ~ back(b_prod) + back(b_prod, 6) + back(b_prod,
##     12), data = beerprod)
##
## Coefficients:
##      (Intercept)      back(b_prod)   back(b_prod, 6)  back(b_prod, 12)
##          7.83088           0.04601          -0.21904           0.68823
```

b. Test to see if there is any autocorrelation left in the residuals. Comment on what you find.

```
res = regression$residuals
acf(res)
```

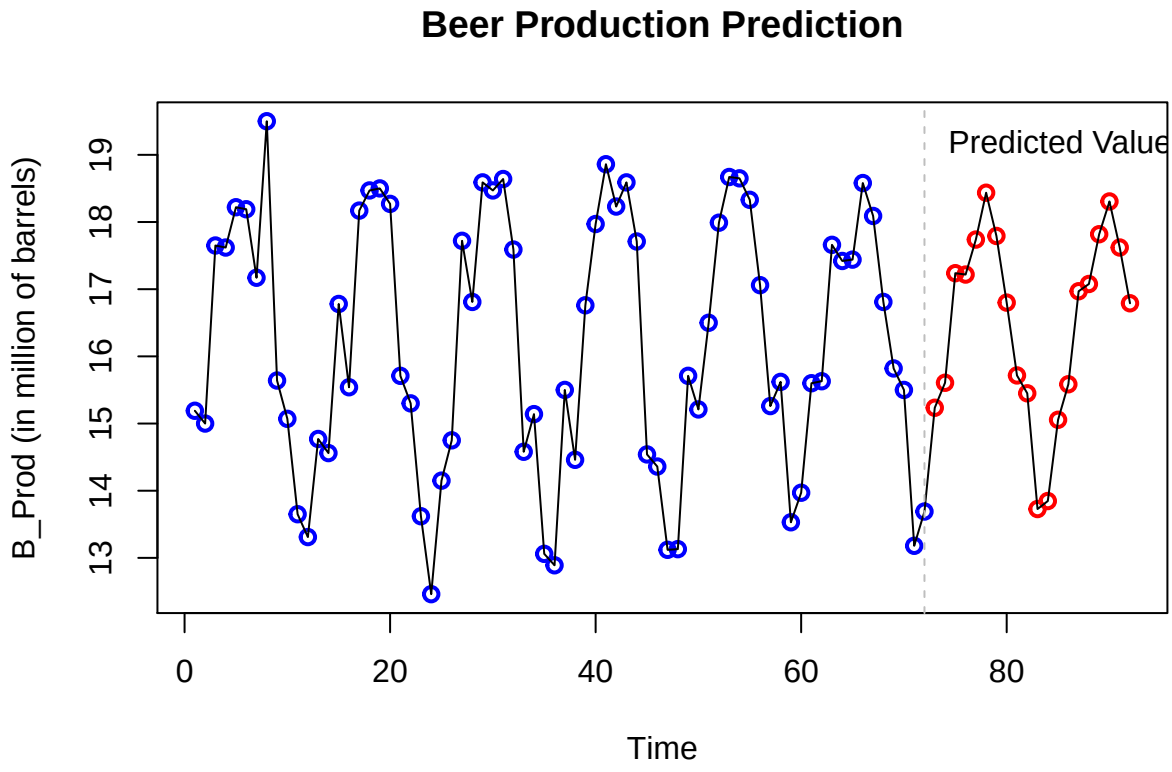



```
cat("As observed through the ACF test, there appears to be no significant
      autocorrelation present in the residuals.")
```

```
## As observed through the ACF test, there appears to be no significant
##       autocorrelation present in the residuals.
```

c. Predict beer production for the next 20 months. Plot your prediction.

```
nstep = 72 + 20 - 1
pred = beerprod[1:72,]
for(i in 72:nstep){
  pred[i+1]=regression$coefficients[1] + regression$coefficients[2] * pred[i] +
    regression$coefficients[3] * pred[i-5] +
    regression$coefficients[4] * pred[i-11]
}
x = c(1:92)
plot(x,pred,main="Beer Production Prediction",xlab="Time",
     ylab="B_Prod (in million of barrels)",col=ifelse(x>72, "red", "blue"),lwd=2)
lines(x,pred, pch=16)
abline(v=72,col="grey",lty=2)
text(72,19,"Predicted Values",adj=c(-0.1,-0.1) )
```



Question 3

a. Assuming the AR(1) model is stationary, prove that the coefficient on the lagged dependent variable (β) is equal to the correlation between the dependent variable and its lag (ρ).

$$
\begin{aligned}
&\because \text{AR(1) model is stationary} \\
&\therefore \rho = \frac{cov(Y_t, Y_{t-s})}{Var(Y_t)} \\
&\therefore Var(Y_t) = Var(Y_{t-s}) \\
\text{Suppose } Y_t &= \alpha + \beta Y_{t-s} + \epsilon \\
\rho &= \frac{cov(\alpha + \beta Y_{t-s} + \epsilon, Y_{t-s})}{Var(Y_t)} \\
&= \frac{cov(\alpha, Y_{t-s}) + cov(\beta Y_{t-s}, Y_{t-s}) + cov(\epsilon, Y_{t-s})}{Var(Y_t)} \\
&= \frac{0 + \beta cov(Y_{t-s}, Y_{t-s}) + 0}{Var(Y_t)} \\
&= \frac{\beta Var(Y_{t-s})}{Var(Y_t)} \\
&= \beta
\end{aligned}
$$

b. In the lecture slides for Chapter 4, slide 15 states, "if all the true autocorrelations are 0, then the standard deviation of the sample autocorrelations is about $1/\sqrt{T}$". Prove this for an AR(1) model. (Hint: recall the formula for s_{b_1} from the Chapter 1 slides.)

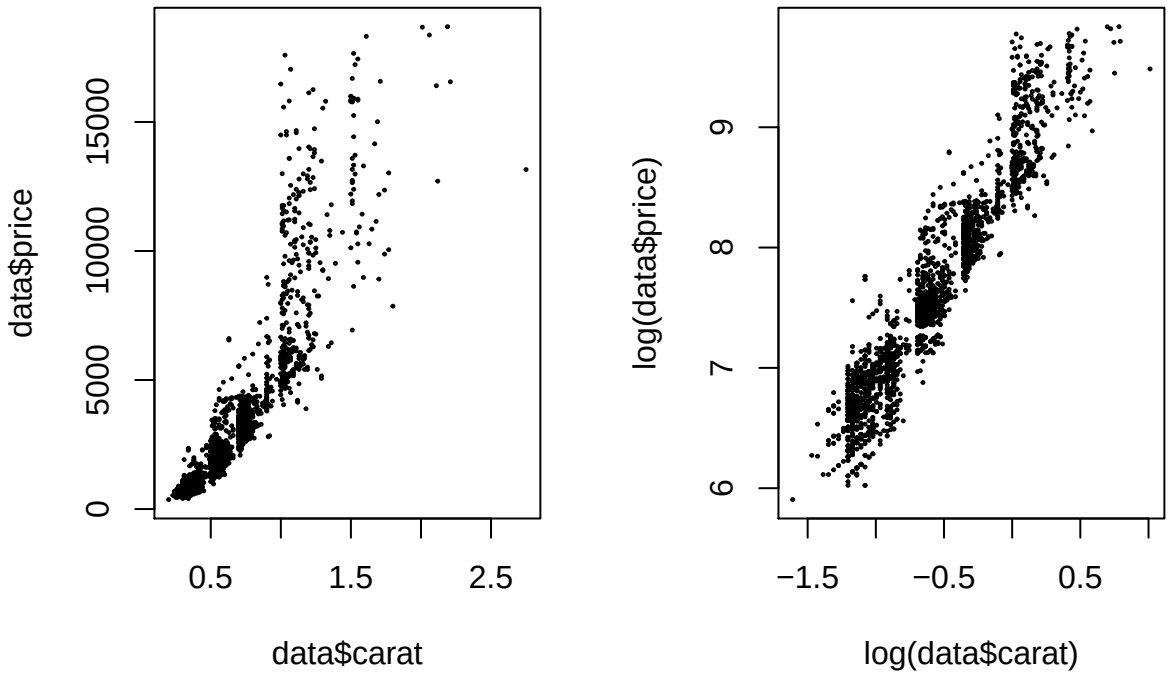
$$
\begin{aligned}
s_{b_1} &= \sqrt{\frac{s^2}{(N-1)s_x^2}} \\
&= \sqrt{\frac{s_{Y_t}}{(T-2)s_{Y_{t-1}}}} \\
&\because \rho = \beta = 0 \\
&\therefore s_{Y_t} = s_{Y_{t-1}} \\
&\therefore s_{b_1} = \sqrt{\frac{1}{(T-2)}} \approx \sqrt{\frac{1}{T}}
\end{aligned}
$$

Question 4

Let's explore the log transformation to address nonlinearity and heterogeneity using the `diamonds` dataset in the `ggplot2` package. Because this is a large dataset, we will focus only on the subset of the data where the cut is "ideal" and the color is "D". Thus, for this question, you should be working with 2,834 data points.

a) Plot (1) carat vs price, and (2) log(carat) vs log(price). Use `par(mfrow=c(1,2))` to put two plots side by side.

```
library("ggplot2")
data("diamonds")
data = diamonds[which(diamonds$cut == "Ideal" & diamonds$color == "D"),]
par(mfrow=c(1,2))
plot(x=data$carat, y = data$price,cex = 0.2)
plot(x=log(data$carat),y=log(data$price),cex = 0.2)
```



b) Regress log(price) on log(carat) and dummy variables for the levels of clarity. What price premium does a diamond with clarity "IF" command relative to a diamond with clarity "SI2"?

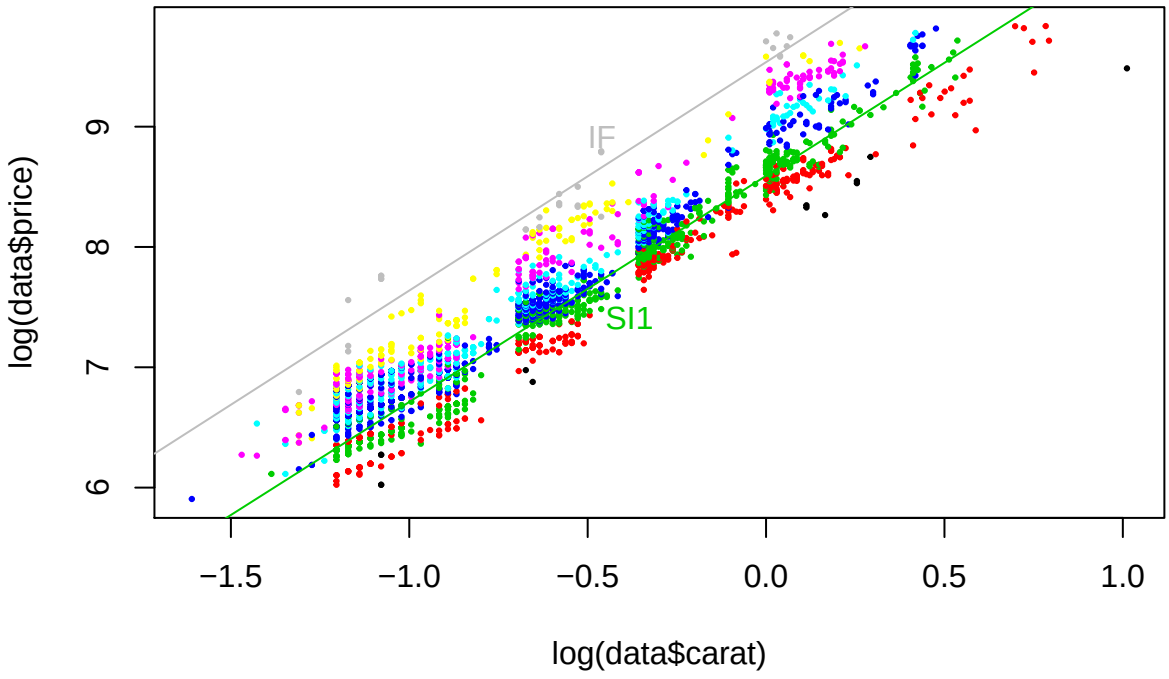
```
a = lm(formula = log(price) ~ log(carat) + factor(clarity,ordered = FALSE),
       data = data)
a
```

```
##
## Call:
## lm(formula = log(price) ~ log(carat) + factor(clarity, ordered = FALSE),
##     data = data)
##
## Coefficients:
##                             (Intercept)
##                                  8.0924
##                              log(carat)
##                                  1.8921
##  factor(clarity, ordered = FALSE)SI2
##                                  0.3220
##  factor(clarity, ordered = FALSE)SI1
##                                  0.5067
##  factor(clarity, ordered = FALSE)VS2
##                                  0.7037
##  factor(clarity, ordered = FALSE)VS1
##                                  0.7396
## factor(clarity, ordered = FALSE)VVS2
##                                  0.9239
## factor(clarity, ordered = FALSE)VVS1
##                                  1.0634
##   factor(clarity, ordered = FALSE)IF
##                                  1.4401
```

$$log(Price_{IF}) - log(Price_{SI2}) = 1.4401 - 0.322 = 1.1181$$
$$log(\frac{Price_{IF}}{Price_{SI2}}) = 1.1181$$
$$\frac{Price_{IF}}{Price_{SI2}} = e^{1.1181} = 3.059$$
$$\text{Price Premium } = \frac{Price_{IF} - Price_{SI2}}{Price_{SI2}} = 205.9\%$$

c) Repeat the second plot in part (a) above (i.e., log(carat) vs log(price)) but make 2 additions. First, color each point by its level of clarity. Second, add the fitted regression lines for the following two levels clarity: "IF" and "SI1". Be sure to match the color of each line to the color of the corresponding points.

```
plot(log(data$carat),log(data$price),cex =0.3,col=data$clarity,pch = 19)
IF = data[which(data$clarity == "IF"),]
SI1 = data[which(data$clarity == "SI1"),]
abline(lm(log(price)~log(carat),data = IF),col="gray")
text(-0.5,8.8, labels = "IF", col = "gray", adj = c(0, -.1))
abline(lm(log(price)~log(carat),data = SI1),col="green3")
text(-0.45,7.3,labels = "SI1",col = "green3",adj = c(0,-0.1))
```



Question 5

a. Using the R dataset mtcars, calculate the correlation between vehicle fuel efficiency (as measured by mpg) and engine displacement (disp).

```
data("mtcars")
correlation = cor(mtcars$mpg,mtcars$disp)
correlation
```

```
## [1] -0.8475514
```

b. Write R code to construct a bootstrapped 95% confidence interval for the correlation. Provide the confidence interval in your answer.

```r
B = 10000

reg_data = data.frame(mtcars$mpg,mtcars$disp)
N = length(reg_data$mtcars.mpg)
BS_cor = vector()
#BS_sample = reg_data[sample(1:N,size=N,replace=TRUE),]
for (b in 1:B){
  BS_sample = reg_data[sample(1:N,size=N,replace=TRUE),]

  BS_cor[b] = cor(BS_sample$mtcars.mpg,BS_sample$mtcars.disp)

}


sd = sd(BS_cor)
interval = quantile(BS_cor,probs = c(0.025,0.975),na.rm = TRUE)

cat("The 95% confident interval is (",2*correlation - interval[2],
    ", ",2 * correlation - interval[1],")" )
```

```
## The 95% confident interval is ( -0.9325999 ,  -0.7816305 )
```

c. Plot the distribution of your bootstrapped correlations and label (on the plot) the sample correlation calculated in part (a).

```r
hist(BS_cor,breaks = 50,col = "pink")
abline(v=correlation,col="blue")
text((correlation - 0.01),510, labels = round(correlation,5),
     col = "red", adj = c(0, -.1))
```