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

Retrieve the Apple stock price series using the `quantmod` package (as done in the notes). Plot the autocorrelations of the difference in log prices.

```
require(quantmod)
```

```
## Loading required package: quantmod
```

```
## Warning: package 'quantmod' was built under R version 4.0.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.0.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.0.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Warning: package 'TTR' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
require(DataAnalytics)
```

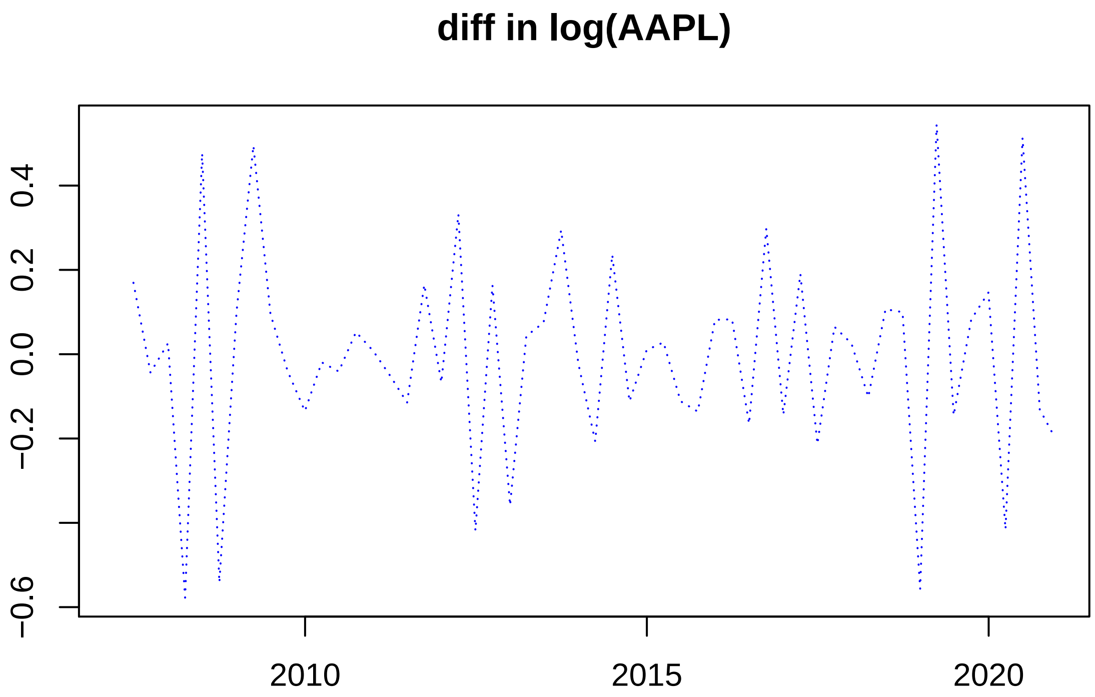
```
## Loading required package: DataAnalytics
```

```r
# gets yahoo finance data. I like this a lot.
getSymbols("AAPL", src="yahoo")
```

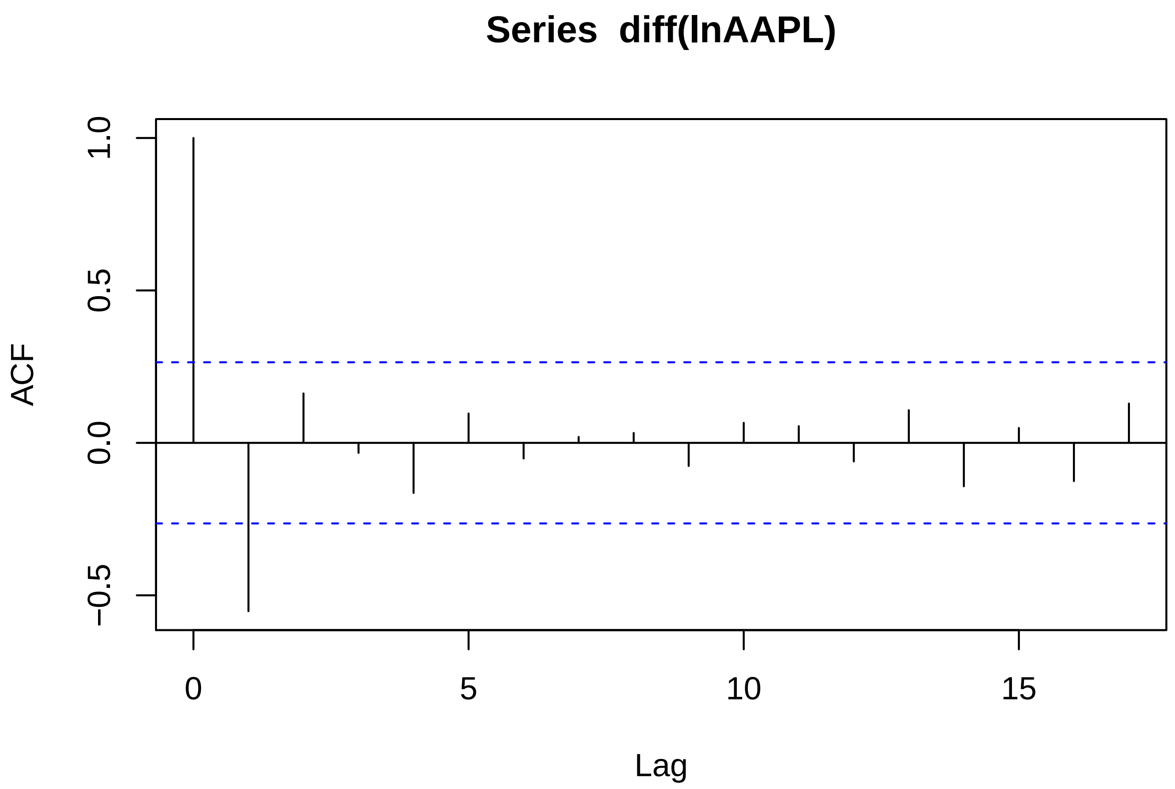
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## [1] "AAPL"
```

```r
# Adj. daily price data from AAPL. Need to convert this to a return
AAPL.Adjusted = AAPL$AAPL.Adjusted
# Delt. -> Provided by quantmod. Calculates % change. Daily though, use quarterlyReturn instead.
lnAAPL = quarterlyReturn(AAPL.Adjusted, type = "log")

plot(x = index(lnAAPL), y = diff(lnAAPL), type = 'l', lty = 3,
     col= "blue", main = "diff in log(AAPL)", xlab = "",
     ylab = "")
```



```
acf(diff(lnAAPL), na.action = na.omit)
```

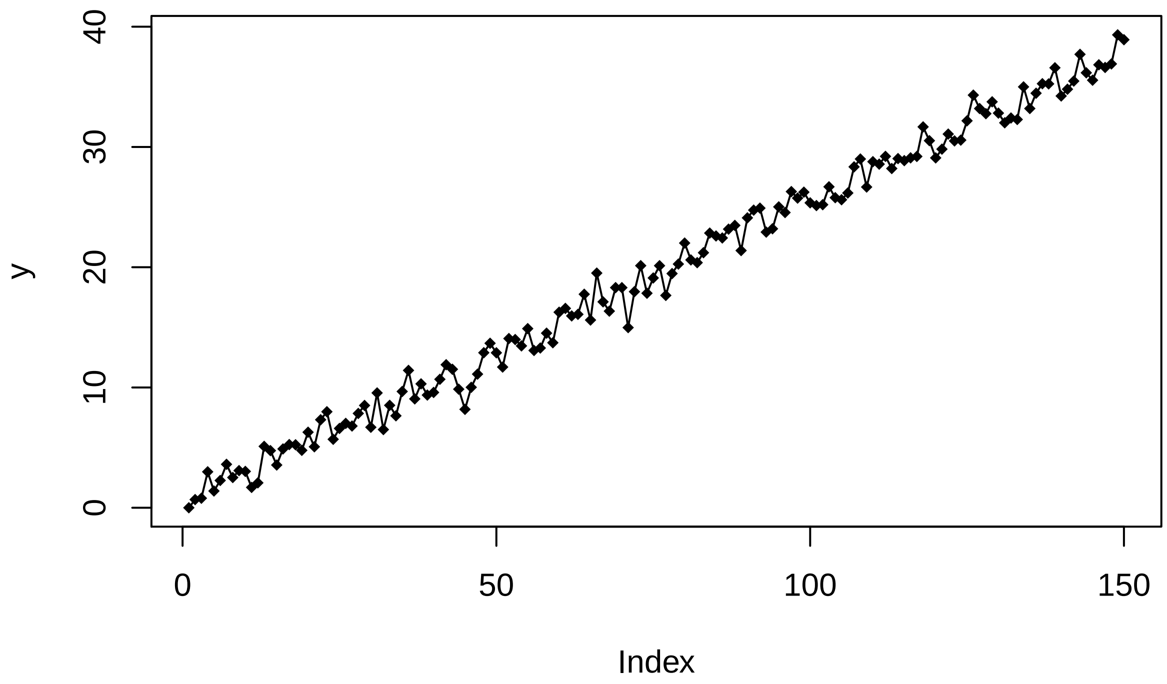


Question 2

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

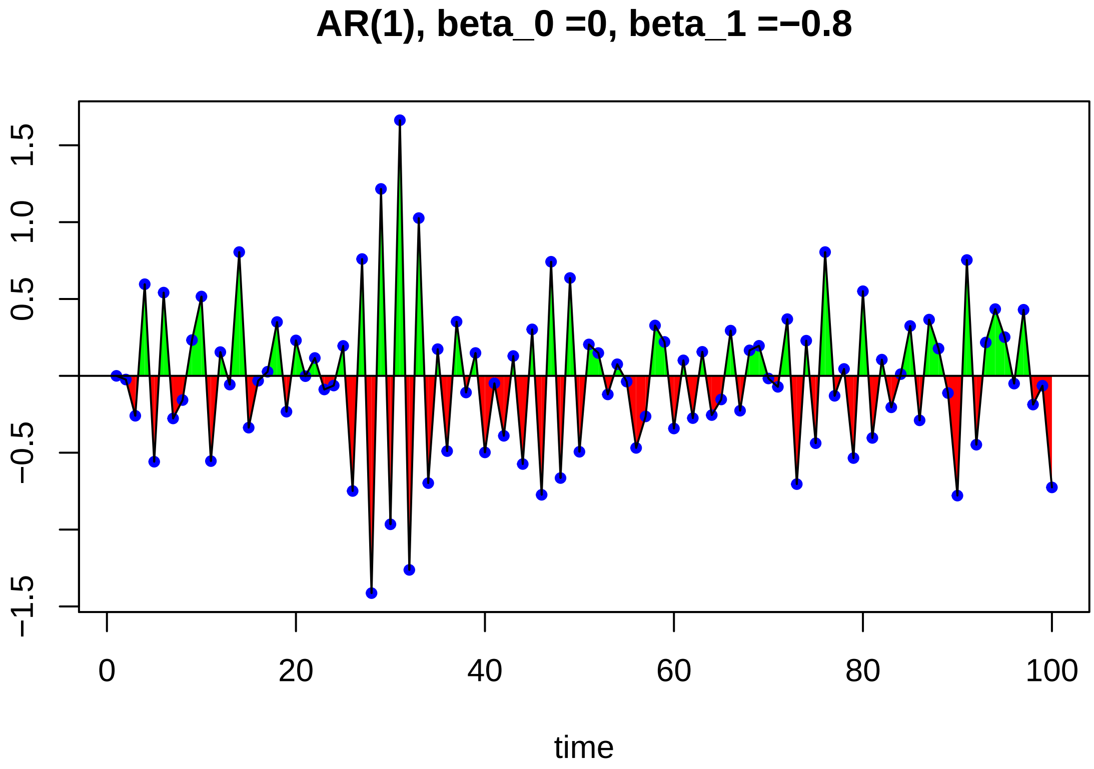
a. A linear time trend: $y_t = \alpha + \beta t + \varepsilon_t$

```
# Chapter 4 code snippets.
len = 150
y = vector(length=len)
y[1] <- 0
for(t in 2:len) {
  y[t] <- 0.5 + 0.25*t + rnorm(1,0,1)
}
plot(y, type="l"); points(y, pch=18, cex = 0.8)
```



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

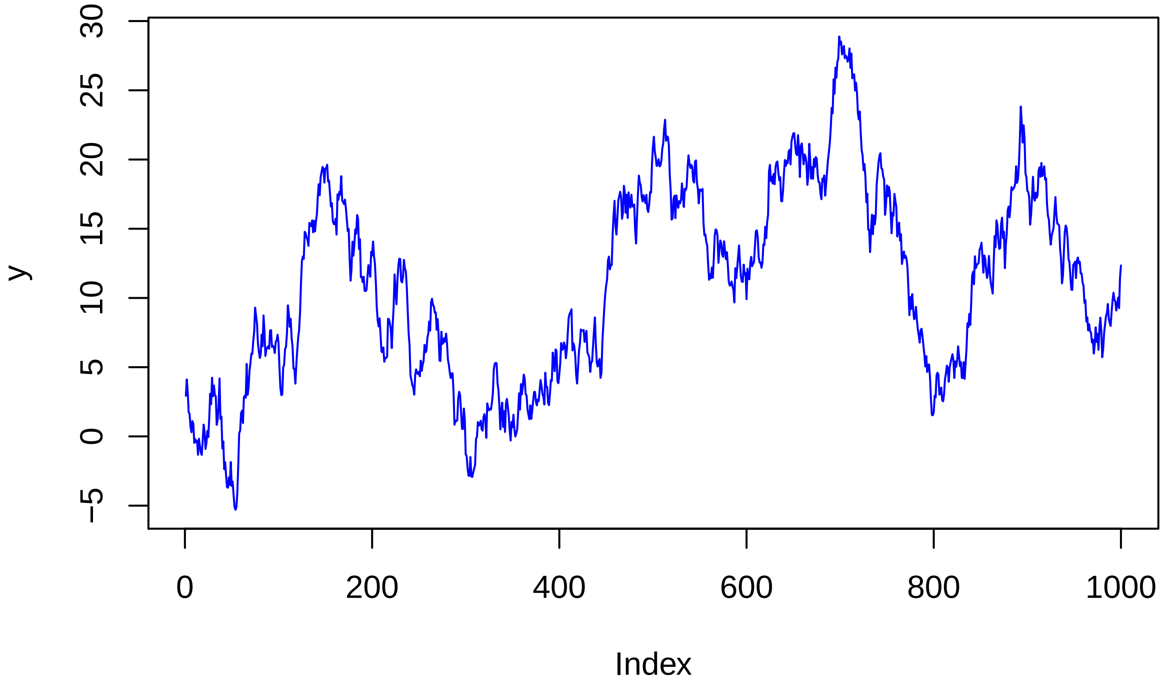
```
# Ch. 4 code snippets

beta0=0
beta1=-.8
T=100
sigma=.3


simar1=function(beta0,beta1,sigma,T){
  mu=beta0/(1-beta1)
  y=double(T)
  y[1]=mu
  for(t in 2:T){y[t]=beta0+beta1*y[t-1]+rnorm(1,sd=sigma)}
  plot(y,type="n",ylab="",xlab="time")
  #
  # color in positive and negative parts
  #
  xint=function(x,y){b=(y[2]-y[1])/(x[2]-x[1]);a=y[1]-b*x[1];xint=(mu-a)/b}
  yoldamu=y[1]>mu
  for(t in 2:T){
    yamu=y[t]>mu
    if(yoldamu)         # here y_t01 is above mean
    {if(yamu)          # here y_t above mean
    {polygon(x=c(t-1,t-1,t,t),y=c(mu,y[t-1],y[t],mu),col="green",lty=0,border="white")}
      else              # here y_t below mean
      {xintercept=xint(c(t-1,t),c(y[t-1],y[t]))
      polygon(x=c(t-1,t-1,xintercept),y=c(mu,y[t-1],mu),col="green",lty=0,border="white")
      polygon(x=c(xintercept,t,t),y=c(mu,mu,y[t]),col="red",lty=0,border="white")}}
    else                 # here y_t-1 is below the mean
    {if(yamu)           # here y_t is above the mean
    {xintercept=xint(c(t-1,t),c(y[t-1],y[t]))
    polygon(x=c(t-1,xintercept,t-1),c(mu,mu,y[t-1]),col="red",lty=0,border="white")
    polygon(x=c(xintercept,t,t),y=c(mu,y[t],mu),col="green",lty=0,border="white")}
      else               # here y_t is below the mean
      {polygon(x=c(t-1,t,t,t-1),y=c(mu,mu,y[t],y[t-1]),col="red",lty=0,border="white")}
    }
    yoldamu=yamu
  }
  points(y,pch=20,col="blue")
  lines(y)
  abline(h=mu)
  title(paste("AR(1), beta_0 =",beta0,", beta_1 =",beta1,sep=""))
  invisible(y)
}

simar1(beta0,beta1,sigma,T)
```



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

```
# From week 8 lecture video
x = rnorm(1000)
```

```
y = cumsum(x)
plot(y, type = 'l', col = "blue")
```



Question 3

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.

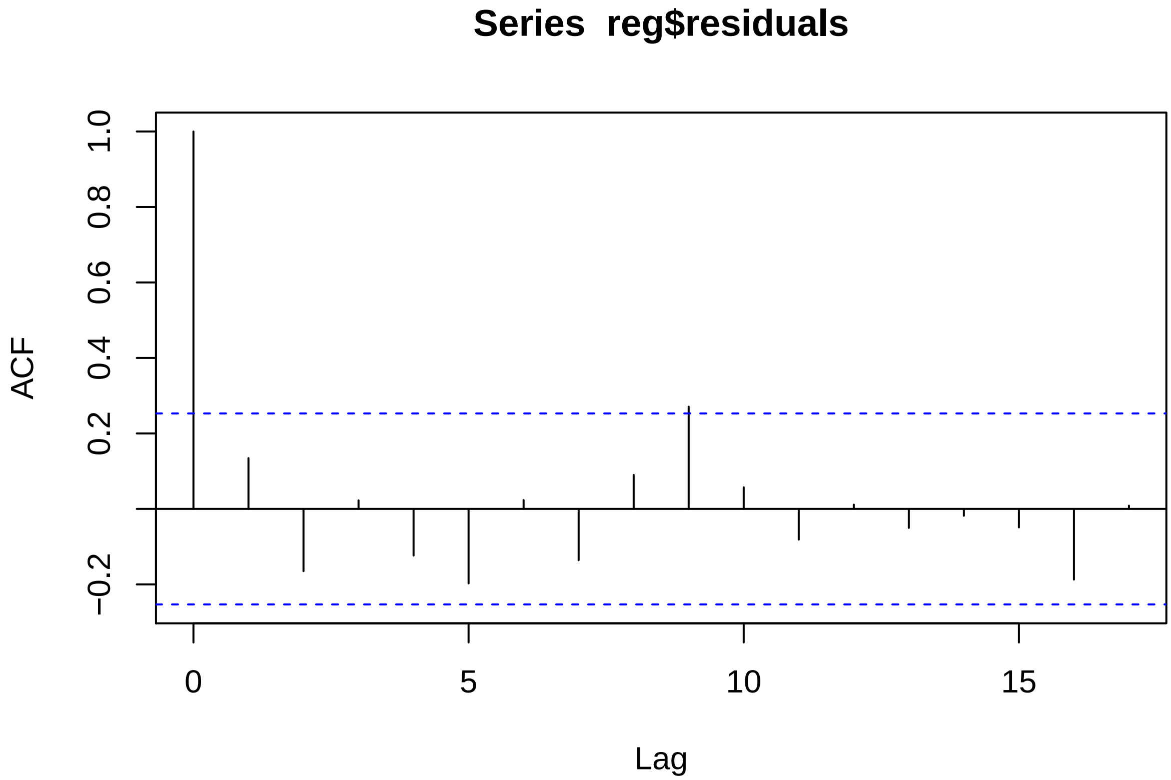
```
data(beerprod)
# implementation as formatted in ch. 4 snippets
beerprod$lag1 = back(beerprod$b_prod)
beerprod$lag6 = back(beerprod$b_prod,6)
beerprod$lag12 = back(beerprod$b_prod,12)

reg = lm(b_prod~lag1+lag6+lag12,data=beerprod)
lmSumm(reg)
```

```
## Multiple Regression Analysis:
##     4 regressors(including intercept) and 60 observations
##
## lm(formula = b_prod ~ lag1 + lag6 + lag12, data = beerprod)
##
## Coefficients:
```

```
##              Estimate Std Error t value p value
## (Intercept)  7.83100   3.08000    2.54   0.014
## lag1         0.04601   0.06570    0.70   0.487
## lag6        -0.21900   0.09554   -2.29   0.026
## lag12        0.68820   0.09445    7.29   0.000
## ---
## Standard Error of the Regression:  0.6491
## Multiple R-squared:  0.891  Adjusted R-squared:  0.885
## Overall F stat: 152.16 on 3 and 56 DF, pvalue= 0
```

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

```
acf(reg$residuals)
```

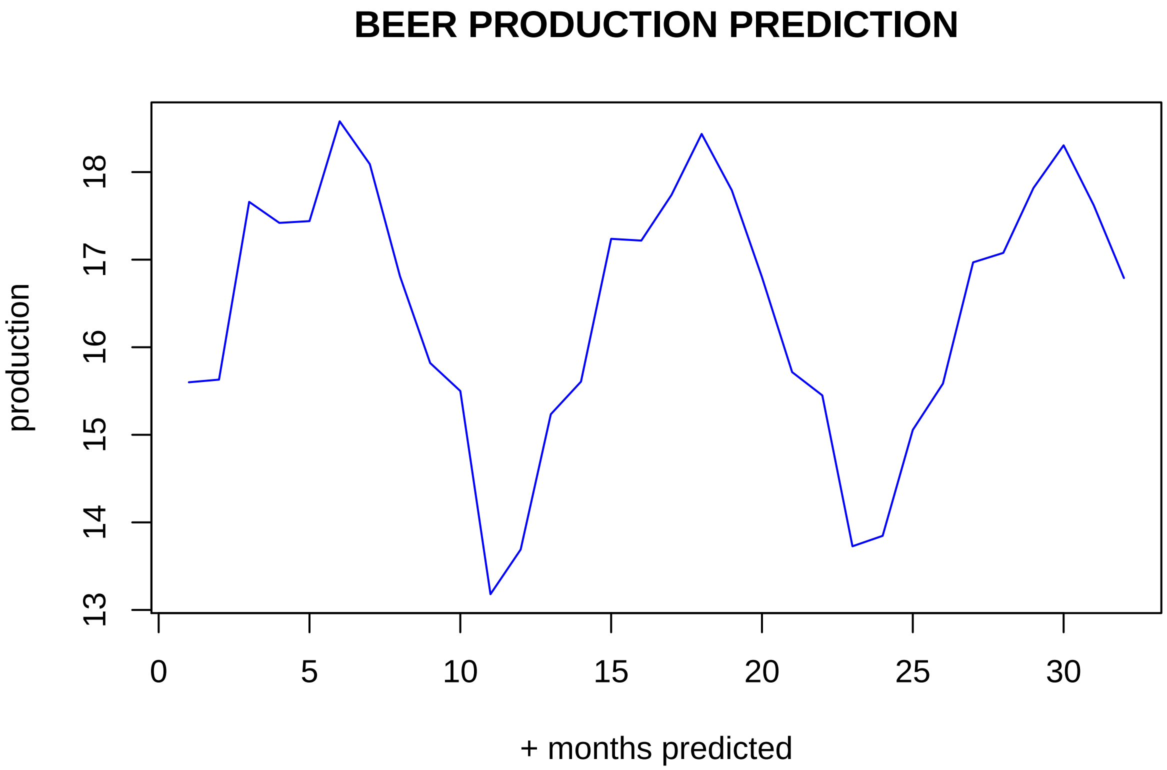


Based on the acf function above, we can see that our regression has incorporated the majority of the autocorrelation in our model. Maybe there's a little bit attributable to sampling error (9 month lag goes beyond our critical value, so maybe if anything we could add that in too).

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

```
# data is monthly; we want to predict 20 months.
nstep = 20
pred.ar = double(nstep+1)
pred.ar[1] = beerprod$b_prod[length(beerprod$b_prod)]
```

```
# addition to my proposed solution vs my original on GitHub : Match lag up with proper time index. That
for (i in 1:12){
  pred.ar[i] = beerprod$b_prod[length(beerprod$b_prod)+i-12]
}
for(i in 1:nstep){
  pred.ar[i+12] = reg$coef[1]+reg$coef[2]*pred.ar[i+11] +
    reg$coef[3]*pred.ar[i+6] + reg$coef[4]*pred.ar[i]
}

plot(pred.ar, xlab = "+ months predicted", ylab = "production",
     main = "BEER PRODUCTION PREDICTION",
     col = "blue", type='l')
```



Question 4

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).

$\rho_1 = \frac{cov(Y_t,Y_{t-1})}{Var(Y_t)}$
Correlation:$\beta_1 = \frac{\sigma_{XY}}{\sigma_X^2}$
Substitute in terms from rho into beta-1:
$Y = Y_{t-1}, X = Y_t$
Then we have: $\beta_1 = \rho_1 = \frac{\sigma_{XY}}{\sigma_X^2} = \frac{cov(Y_t,Y_{t-1})}{Var(Y_t)}$.

b. In the lecture slides for Chapter 4, slide 19 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.)

ar(1) correlation as defined above:
$\rho_1 = \frac{(Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{(Y_t - \bar{Y})^2}$
$\text{Var}(b_1) = \frac{\sigma^2}{(N-1)s_x^2}$
Std Error (with 1 degree of freedom):
$s_{b_1} = \sqrt{\frac{s^2}{(N-1)s_x^2}}$, where in this case $s^2 = s_x^2$.
$s_{b_1} = \frac{1}{\sqrt{(N-1)}}$
Since we have proven the relationship between the correlation between beta and rho already (and indirectly, the variance and stdev of rho and beta) we can substitute the following:
$s_{\rho_1} = \frac{1}{\sqrt{(T-1)}}$
With zero degrees of freedom, the std error of ρ is equal to:
$s_{\rho_1} = \frac{1}{\sqrt{T}}$

Question 5

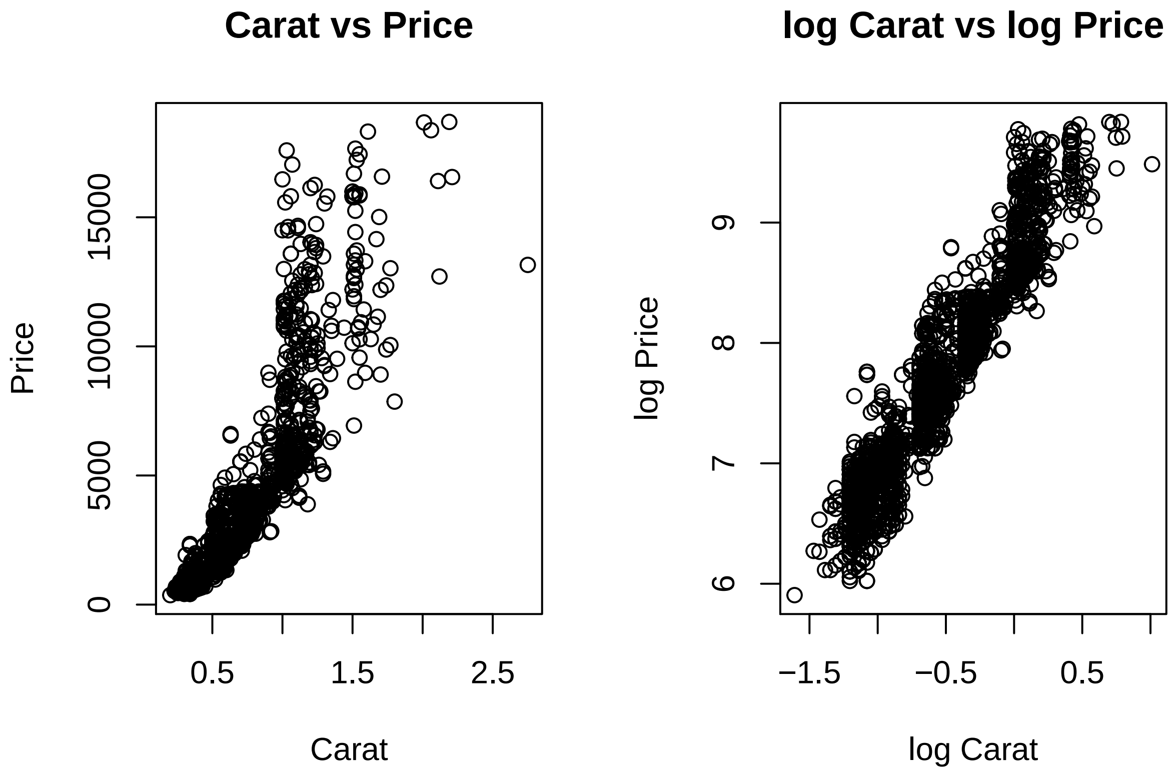
Let's explore the log transformation to address nonlinearity and heterogeneity using the `diamonds` dataset in the `ggplot2` package. Because this is a large dataset, we will focus only on the subset of the data where the cut is "ideal" and the color is "D". Thus, for this question, you should be working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

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
data(diamonds)
# select Ideal and D color diamonds
dsub = subset(diamonds,cut=="Ideal")
dsub = subset(dsub,color=="D")
dsub$logCarat = log(dsub$carat)
dsub$logPrice = log(dsub$price)

par(mfrow=c(1,2))
plot1=plot(x=dsub$carat,y=dsub$price,
             xlab = "Carat", ylab="Price", main="Carat vs Price")
plot2=plot(x=dsub$logCarat,y=dsub$logPrice,
           xlab="log Carat", ylab="log Price", main="log Carat vs log Price")
```



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”?

```
clarityIF = ifelse(dsub$clarity=="IF",1,0)
claritySI2 = ifelse(dsub$clarity=="SI2",1,0)
IFreg = lmSumm(lm(dsub$logPrice~dsub$logCarat + clarityIF))
```

```
## Multiple Regression Analysis:
##     3 regressors(including intercept) and 2834 observations
##
## lm(formula = dsub$logPrice ~ dsub$logCarat + clarityIF)
##
## Coefficients:
##                  Estimate Std Error t value p value
## (Intercept)      8.6670  0.008084 1072.22       0
## dsub$logCarat    1.7890  0.009860  181.42       0
## clarityIF        0.8053  0.045100   17.86       0
## ---
## Standard Error of the Regression:  0.2374
## Multiple R-squared:  0.922  Adjusted R-squared:  0.922
## Overall F stat: 16698.34 on 2 and 2831 DF, pvalue= 0
```

```
SI2reg = lmSumm(lm(dsub$logPrice~dsub$logCarat + claritySI2))
```

```
## Multiple Regression Analysis:
```

```
##     3 regressors(including intercept) and 2834 observations
##
## lm(formula = dsub$logPrice ~ dsub$logCarat + claritySI2)
##
## Coefficients:
##                Estimate Std Error t value p value
## (Intercept)      8.7660  0.008093 1083.11       0
## dsub$logCarat    1.8540  0.009404  197.14       0
## claritySI2      -0.3671  0.012840  -28.60       0
## ---
## Standard Error of the Regression:  0.2206
## Multiple R-squared:  0.933  Adjusted R-squared:  0.932
## Overall F stat: 19568.55 on 2 and 2831 DF, pvalue= 0
```

```
IFcoef = IFreg$coef[3]
SI2coef = SI2reg$coef[3]
```

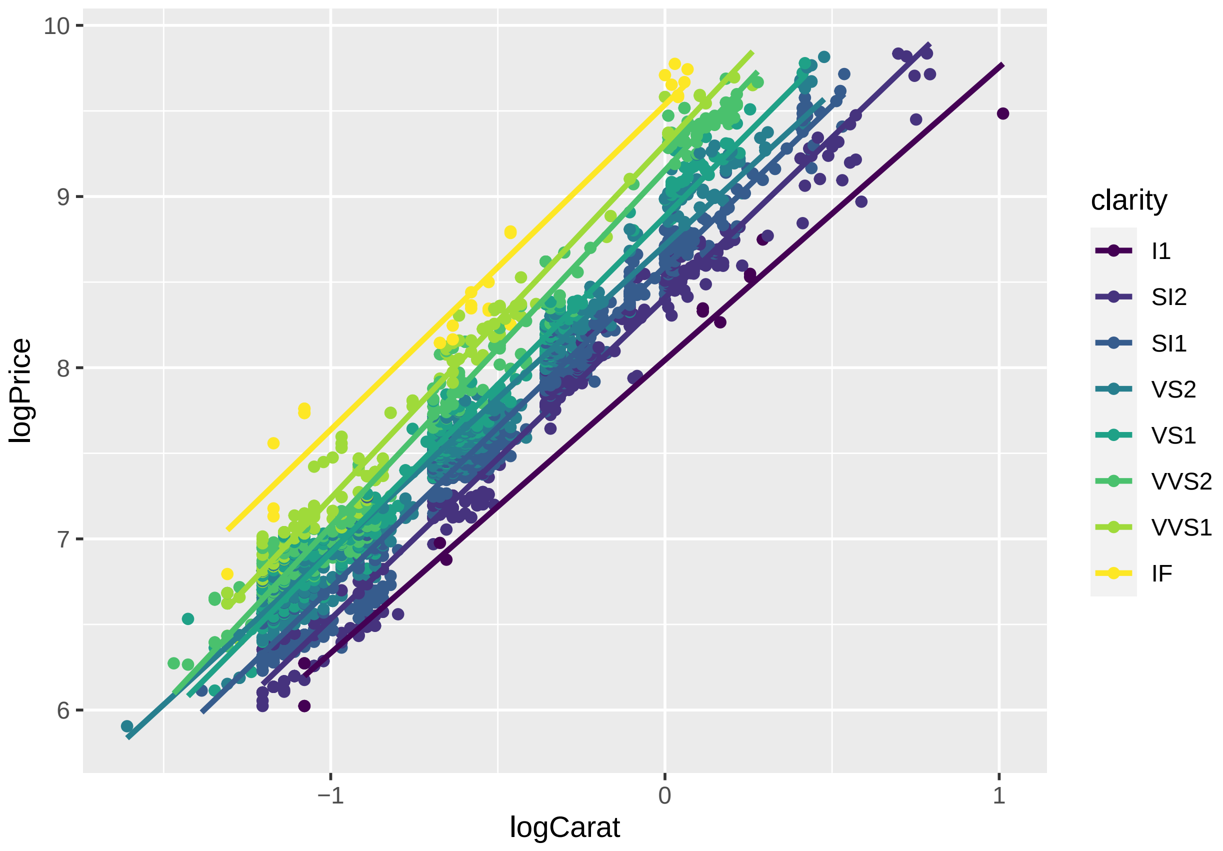
IF clarity diamonds have a coefficient in our regression of 0.8053, and SI2 clarity diamonds have a coefficient in our regression of -0.3671. We can see that an SI2 clarity diamond tends to be less valuable than a typical diamond, having an interesting negative coefficient in our regression.
Meanwhile an IF clarity diamond has a coefficient of 0.8 in our regression, which shows that SI2 is a desirable clarity for a diamond and typically is sought-after, or increases the price of the diamond.
The difference in the two coefficients is 1.1724, which means you could expect the price difference in an SI2 and an IF diamond to be around 1.1724-times.

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.

```
qplot(logCarat, logPrice, data = dsub, colour=clarity,
      geom = c("point","smooth"), method="lm",se=FALSE)
```

```
## Warning: Ignoring unknown parameters: method, se
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```


10
9
8
7
6
logPrice
−1
0
1
logCarat
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF