Pset 04
18.S096

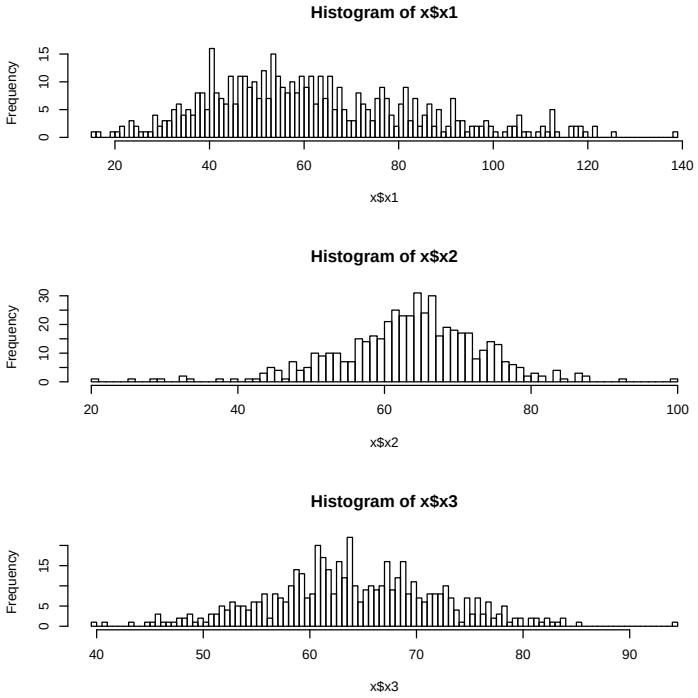
(Thank you L.N.)

Spring 2018

1.

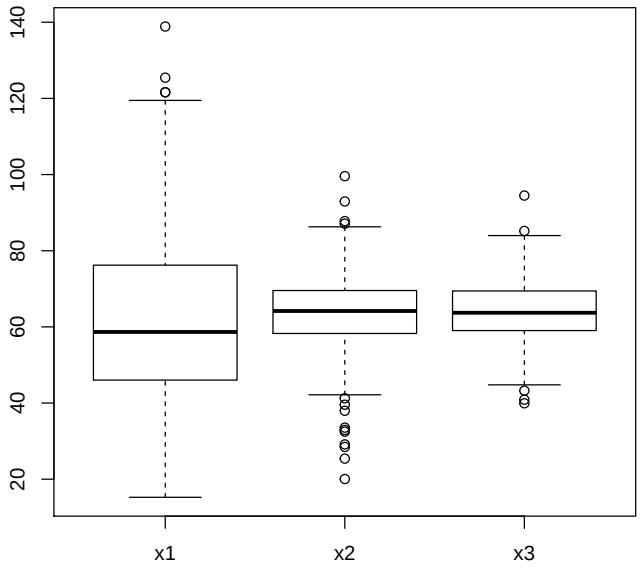
(a)

```
> x=read.csv(file= "pset4_x.csv", row.names=1 )
> dim(x)

[1] 500   3

> head(x)

         x1       x2       x3
1  61.70558 20.05120 58.98837
2  53.67220 50.72680 65.46915
3  27.61730 61.10642 57.31497
4  83.69144 62.63499 76.76225
5 100.00354 72.74861 66.63606
6  33.69408 69.68055 57.43625

> apply(x,2,summary)

               x1       x2       x3
Min.     15.24956 20.05120 39.93561
1st Qu.  46.05140 58.28631 59.03002
Median   58.66442 64.17307 63.70577
Mean     62.01869 63.47535 64.18115
3rd Qu.  76.13621 69.51427 69.42850
Max.    138.87584 99.58072 94.48221

> par(mfcol = c(3,1))
> hist(x$x1, nclass = 100)
> hist(x$x2, nclass = 100)
> hist(x$x3, nclass = 100)
```



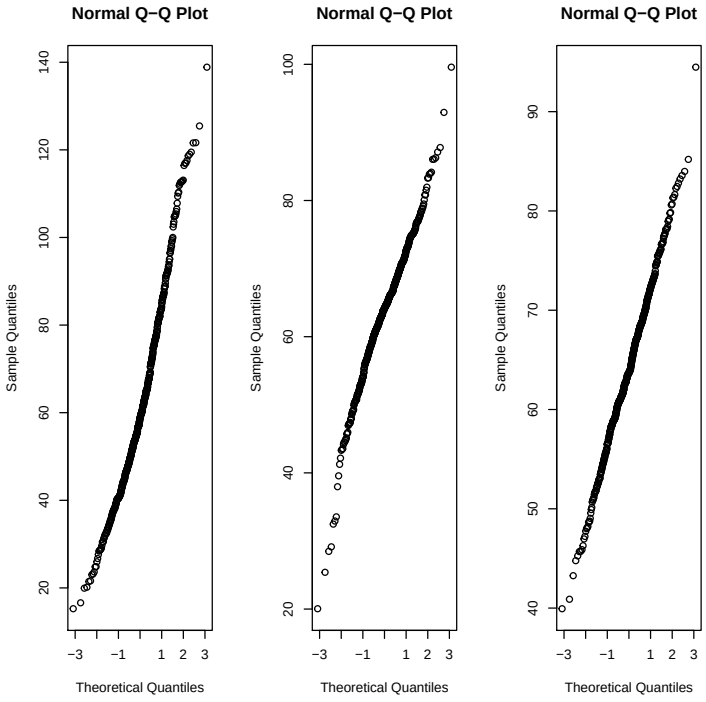
From these it looks like x_2 and x_3 somewhat resemble a gaussian and x_1 does not as it has a longer tail (may be gamma).

```
> boxplot(x)
```



And x_2, x_3 are semetric around the median where x_1 is skewed in one direction. This is evidence that x_1 is not a gaussian and x_2, x_3 might be.

```
> par(mfcol = c(1,3))
> x.qqnorm.bycol <- apply(x,2,qqnorm)
```



x_3 looks very linear which is evidence that it is a gaussian. x_1, x_2 also look close to linear.

(b)

```
> library(MASS)
> x3 <- x$x3
> fitdistr.x3.normal <- fitdistr(x3, densfun = "normal")
> fitdistr.x3.cauchy <- fitdistr(x3, densfun = "cauchy")
> fitdistr.x3.gamma<-fitdistr(x3,densfun="gamma")
> fitdistr.x3.t<-fitdistr(x3,densfun="t")
> fitdistr.x3.loglik <- c(normal = fitdistr.x3.normal$loglik,
+                         cauchy = fitdistr.x3.cauchy$loglik,
+                         gamma = fitdistr.x3.gamma$loglik,
+                         t = fitdistr.x3.t$loglik)
> #get loglikelihood of fitting different distributions to data
> fitdistr.x3.loglik

   normal    cauchy     gamma         t
-1754.618 -1838.592 -1756.902 -1754.338
```

(c)

```
> fnc.fitall <- function(x3){
+   fitdistr.x3.normal <- fitdistr(x3, densfun = "normal")
+   fitdistr.x3.cauchy <- fitdistr(x3, densfun = "cauchy")
+   fitdistr.x3.gamma<-fitdistr(x3,densfun="gamma")
+   fitdistr.x3.t<-fitdistr(x3,densfun="t")
+   fitdistr.x3.loglik <- c(normal = fitdistr.x3.normal$loglik,
+                         cauchy = fitdistr.x3.cauchy$loglik,
+                         gamma = fitdistr.x3.gamma$loglik,
+                         t = fitdistr.x3.t$loglik)
+   result <- list(loglikes = fitdistr.x3.loglik,
+                  fitdistr.normal = fitdistr.x3.normal,
+                  fitdistr.cauchy = fitdistr.x3.cauchy,
+                  fitdistr.gamma = fitdistr.x3.gamma,
+                  fitdistr.t  =fitdistr.x3.t)
+   return(result)
+ }#returns a list of what we did in part b
```

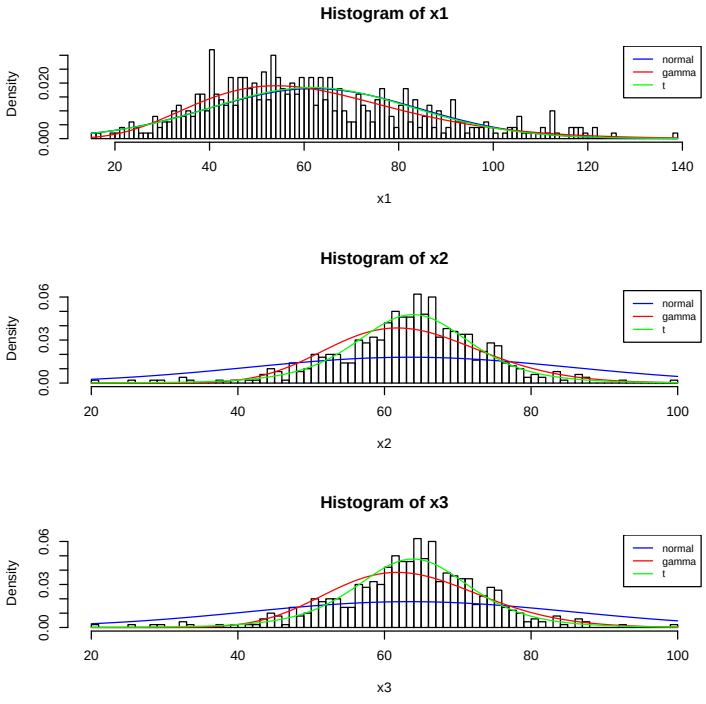
(d)

```
> fitdistr.x1 <- fnc.fitall(x$x1)
> fitdistr.x1$loglikes

   normal    cauchy     gamma         t
-2258.339 -2338.552 -2236.395 -2258.288

> fitdistr.x2 <- fnc.fitall(x$x2)
> fitdistr.x2$loglikes

   normal    cauchy     gamma         t
-1854.649 -1898.829 -1880.582 -1841.067

> fitdistr.x3 <- fnc.fitall(x$x3)
> fitdistr.x3$loglikes

   normal    cauchy     gamma         t
-1754.618 -1838.592 -1756.902 -1754.338
```

For x_1 the gamma distribution has the highter loglikelihood with normal and t being close to it and the cauchy being lowest.
For x_2 the t distribution has the highest loglikelihood with the normal and gamma being a litte higher and the cauchy being farthest away.
For x_3 the t distribution has the highest loglikelihood with the normal nearly having the same value, the gamma being very close and the cauchy being the lowest.
The t distribution has 3 parameters, the normal, gamma and t distribution have 2 parameters and the cauchy only has 1. From our loglikelihood the t distribution (3 parameters) the lowest loglikelihood for two of the samples and is the second hightest on the other. The gamma distribution (2 parameters) is the highest in the sample that the t distribution is not the highest.

(e)

```
> fnc.plotfitdistr <- function(x, main0 = "title", xlab0 = "x_axis"){
+   hist(x, nclass = 100, freq = FALSE, main = main0, xlab = xlab0, ylab = "Density")
+   fitdistr.x <- fnc.fitall(x)
+   curve(dnorm(x, mean = fitdistr.x$fitdistr.normal$estimate["mean"],
+                                  sd = fitdistr.x1$fitdistr.normal$estimate["sd"]),
+                                   add = TRUE, col = "blue")
+   curve(dgamma(x, shape = fitdistr.x$fitdistr.gamma$estimate["shape"],
+                rate = fitdistr.x$fitdistr.gamma$estimate["rate"]), add = TRUE, col = "red'
+   curve(dt((x-fitdistr.x$fitdistr.t$estimate["m"])/fitdistr.x$fitdistr.t$estimate["s"],
+            df = fitdistr.x$fitdistr.t$estimate["df"])/fitdistr.x$fitdistr.t$estimate["s"],
+         add = TRUE, col = "green")
+   legend("topright", legend = c("normal", "gamma", "t"), col =c("blue","red","green"),
+          lty=1, cex = .8)
+ }

> par(mfcol = c(3,1))
> fnc.plotfitdistr(x$x1, main0 = "Histogram of x1", xlab0 = "x1")
> fnc.plotfitdistr(x$x2, main0 = "Histogram of x2", xlab0 = "x2")
> fnc.plotfitdistr(x$x2, main0 = "Histogram of x3", xlab0 = "x3")
```



(f)
For x_1 the highest loglikelihood is of a gamma distribution and it appears fits better than the other distributions. For x_2, x_3 the highest loglikelihood was from a t distribution and it also appears to fit the best.
(g)

```
> names(fitdistr.x3.t)

[1] "estimate" "sd"       "vcov"     "loglik"   "n"

> fitdistr.x3.t$estimate

        m         s        df
64.170860  7.784243 27.023153

> fitdistr.x3.t$sd

         m          s         df
 0.3604169  0.3597979 24.6705619

> #confirm sd values are diagonal roots of vcov
> sqrt(diag(fitdistr.x3.t$vcov))

         m          s         df
 0.3604169  0.3597979 24.6705619
```

We should treat the df as a measure of the spread of the distribution.

2.

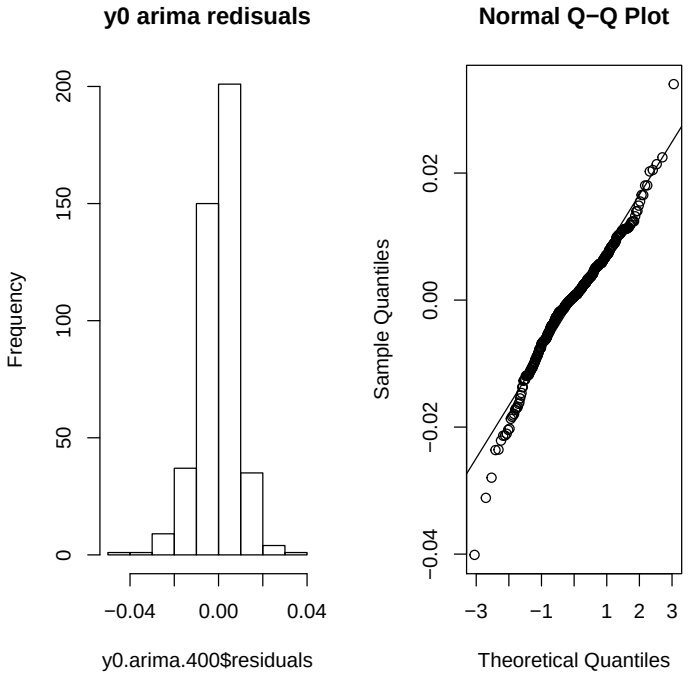
```
> library(zoo)
> sp500 <- read.zoo(file = "SP500.csv")
> y <- diff(log(sp500))
> y0 <- as.numeric( y )
> y0.ar <- ar(x=y0, method = "mle")
> y0.ar

Call:
ar(x = y0, method = "mle")

Coefficients:
      1        2        3        4
 0.0351  -0.0177  -0.0275  -0.1454

Order selected 4  sigma^2 estimated as  6.892e-05

> names(y0.ar)

 [1] "order"        "ar"           "var.pred"     "x.mean"       "aic"
 [6] "n.used"       "order.max"    "partialacf"   "resid"        "method"
[11] "series"       "frequency"    "call"         "asy.var.coef"

> ar.est=y0.ar$ar
> ar.sd=sqrt(diag(y0.ar$asy.var.coef))
> ar.t=ar.est/ar.sd
> ar.pval=2*(1-pnorm(abs(ar.t)))
> coef.table<-cbind(ar.est,ar.sd,ar.t,ar.pval)
> coef.table

          ar.est      ar.sd       ar.t     ar.pval
[1,]  0.03511561 0.04722324  0.7436088 0.457113202
[2,] -0.01767442 0.04723482 -0.3741820 0.708268924
[3,] -0.02745456 0.04723482 -0.5812355 0.561081738
[4,] -0.14544042 0.04722324 -3.0798486 0.002071059

> y0.arima.400<-arima(y0, order=c(4,0,0),method="ML")
> y0.arima.400

Call:
arima(x = y0, order = c(4, 0, 0), method = "ML")

Coefficients:
         ar1      ar2      ar3      ar4  intercept
      0.0352  -0.0176  -0.0274  -0.1455      1e-04
s.e.  0.0472   0.0478   0.0477   0.0476      3e-04

sigma^2 estimated as 6.892e-05:  log likelihood = 1480.42,  aic = -2948.83
```

(a)

```
> sum(dnorm(y0.arima.400$residuals, sd = sqrt(y0.arima.400$sigma2), log = TRUE))

[1] 1480.461

> y0.arima.400$loglik

[1] 1480.416

> #notice how they are .05 apart
```

(b)

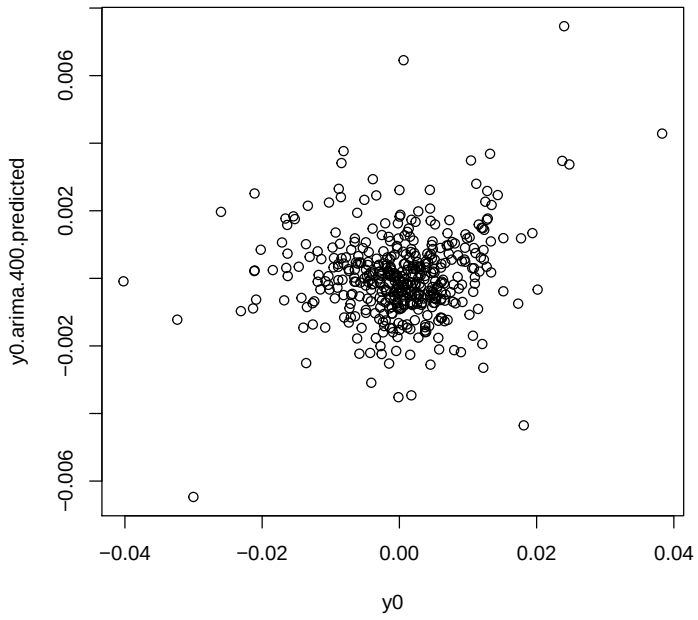
```
> par(mfcol = c(1,2))
> hist(y0.arima.400$residuals, main = "y0 arima redisuals")
> qqnorm(y0.arima.400$residuals)
> abline(a=0, b= sqrt(y0.arima.400$sigma2))
```



The data is not consitent with a gaussian. The histogram is not close symetric.
In addition the normal qq plot is not linear, especially at the tails.

(c)

```
> #predicted = observed - residuals
> y0.arima.400.predicted <- y0 - y0.arima.400$residuals
> cor(y0,y0.arima.400.predicted)

[1] 0.1544787

> plot(y0,y0.arima.400.predicted)
```



It does not predict the time series very well as it has a small correlation. However, it still does have positive correlation meaning that the model has some ability to predict the outcome.

(d)

```
> y0.arima.400.predicted.var <- var(y0.arima.400.predicted)
> y0.arima.400.predicted.var;var(y0)

[1] 1.667174e-06

[1] 7.076521e-05

> y0.arima.400.predicted.var/var(y0)

[1] 0.02355924
```

Because the variance of the observed data is much higher than the variance of the predicted data I expect the predicted model to be a good predictor when the SP returns following a trend [the obsevered data is not varying by a lot]. I expect the preditctor to not be good at predicting when the SP returns change trends[to observed data is varying a lot compared to the previous days].
Statistical significance implies that the value of predictor is better at predicting with the fourth lag-coeffcient than without the fourth lag-coffcient. It does not mean that the predictor is good in general (could have a bad predictor that is better with predicting with the fourth lag-coffcient than without but is still not a good predictor either way).

3.

```
> y0=as.numeric(y)
> T <- length(y0)
> index.lag0<-c(5:T)
> y0.lag0=y0[index.lag0]
> y0.lag1=y0[index.lag0-1]
> y0.lag2=y0[index.lag0-2]
> y0.lag3=y0[index.lag0-3]
> y0.lag4=y0[index.lag0-4]
> ones=0*y0.lag0+1
> xmat=cbind(ones,y0.lag1,y0.lag2,y0.lag3,y0.lag4)
> head(xmat)

     ones       y0.lag1       y0.lag2       y0.lag3       y0.lag4
[1,]    1  0.0060633452 -0.0025149269 -0.0003330203 -0.0089014131
[2,]    1 -0.0002122317  0.0060633452 -0.0025149269 -0.0003330203
[3,]    1  0.0003482487 -0.0002122317  0.0060633452 -0.0025149269
[4,]    1  0.0023040304  0.0003482487 -0.0002122317  0.0060633452
[5,]    1 -0.0126559665  0.0023040304  0.0003482487 -0.0002122317
[6,]    1  0.0107598763 -0.0126559665  0.0023040304  0.0003482487
```

(a)

```
> y0.lmfit <- lm(y0.lag0 ~ . , data = as.data.frame(xmat))
> y0.lmfit

Call:
lm(formula = y0.lag0 ~ ., data = as.data.frame(xmat))

Coefficients:
(Intercept)         ones      y0.lag1      y0.lag2      y0.lag3      y0.lag4
  7.614e-05           NA    3.543e-02   -1.843e-02   -2.588e-02   -1.462e-01
```

These parameters are close to those computed in 2 with lag1,lag4 being very close to problem 2 (around .0009 off) and lag2,lag3 a little farther away (around .002 off).

(b)

```
> sum(dnorm(y0.lmfit$residuals, sd = sqrt(sum((y0.lmfit$residuals)^2)), log = TRUE))

[1] 361.4477

> y0.arima.400$loglik

[1] 1480.416

> #very different values
> #now compute problem 2 with same offset from 3
> y0.arima.400.4later<-arima(y0[index.lag0], order=c(4,0,0),method="ML")
> y0.arima.400.4later$loglik

[1] 1465.676

> sum(dnorm(y0.arima.400.4later$residuals, sd = sqrt(y0.arima.400.4later$sigma2), log = TRUE

[1] 1465.72

> #notice how computing with arima or calculating it stragiht give simlar value like in 2 a
> #Still very different from calculated
```

(c)

```
> dlaplace<-function(x, location=0,scale=1){dx=(0.5/scale)* exp(-abs(x-location)/scale)}
> y0.mle.dlplace <-fitdistr(y0.lag0, densfun = dlaplace,start = list(location = 0, scale = 1
> y0.mle.dlplace

     location        scale
  0.0003854905   0.0060807466
 (0.0003148122) (0.0002695647)

> y0.mle.dlplace$loglik

[1] 1485.924

> y0.arima.400$loglik

[1] 1480.416

> y0.arima.400.4later$loglik

[1] 1465.676
```

The mle laplace is higher than loglik of the arima model from before (both the adjusted and non-adjusted). This means the laplace fits the data better.

4.

```
> library(L1pack) #want to perfrom linear Least absolute deviations estimation
> y0.lad <- lad(y0.lag0 ~ y0.lag1+y0.lag2+y0.lag3+y0.lag4 , data = as.data.frame(xmat))
> y0.lad$logLik

[1] 1488.659

> y0.arima.400$loglik

[1] 1480.416

> y0.mle.dlplace$loglik

[1] 1485.924
```

Notice how the loglik is higher for the error term being laplace is higher than previous logliks.