Topics_2

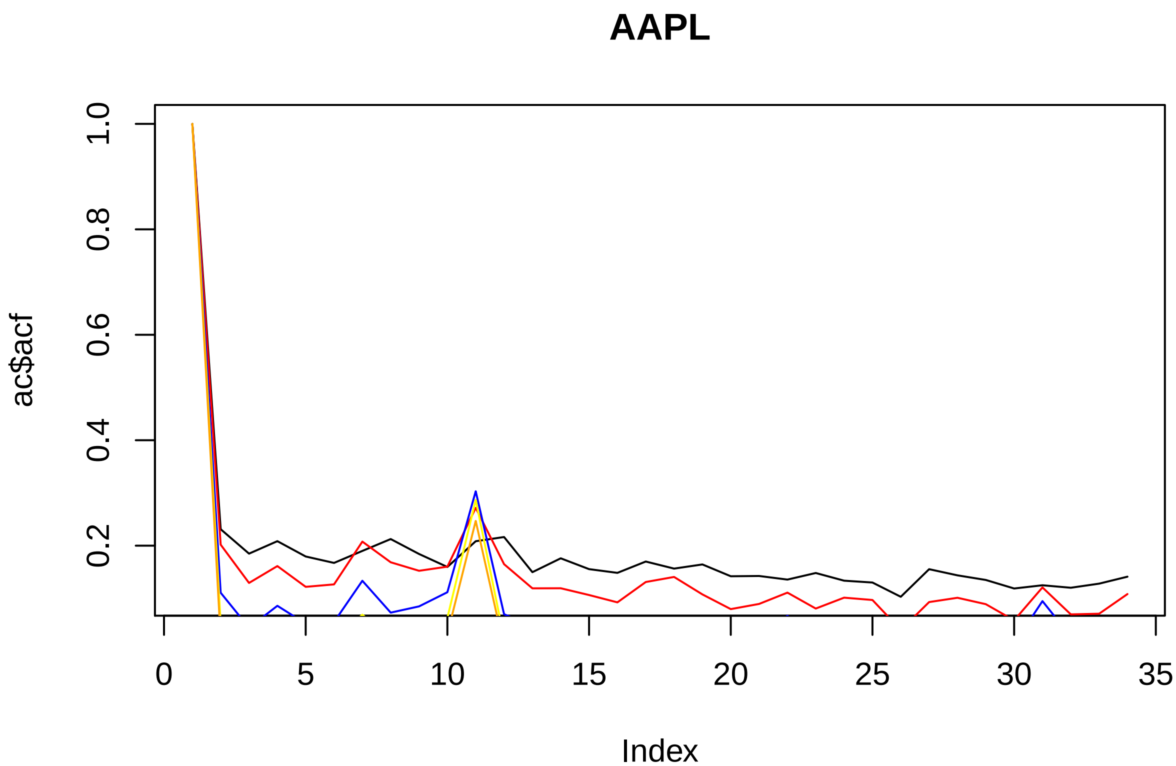
Domagoj Fizulic

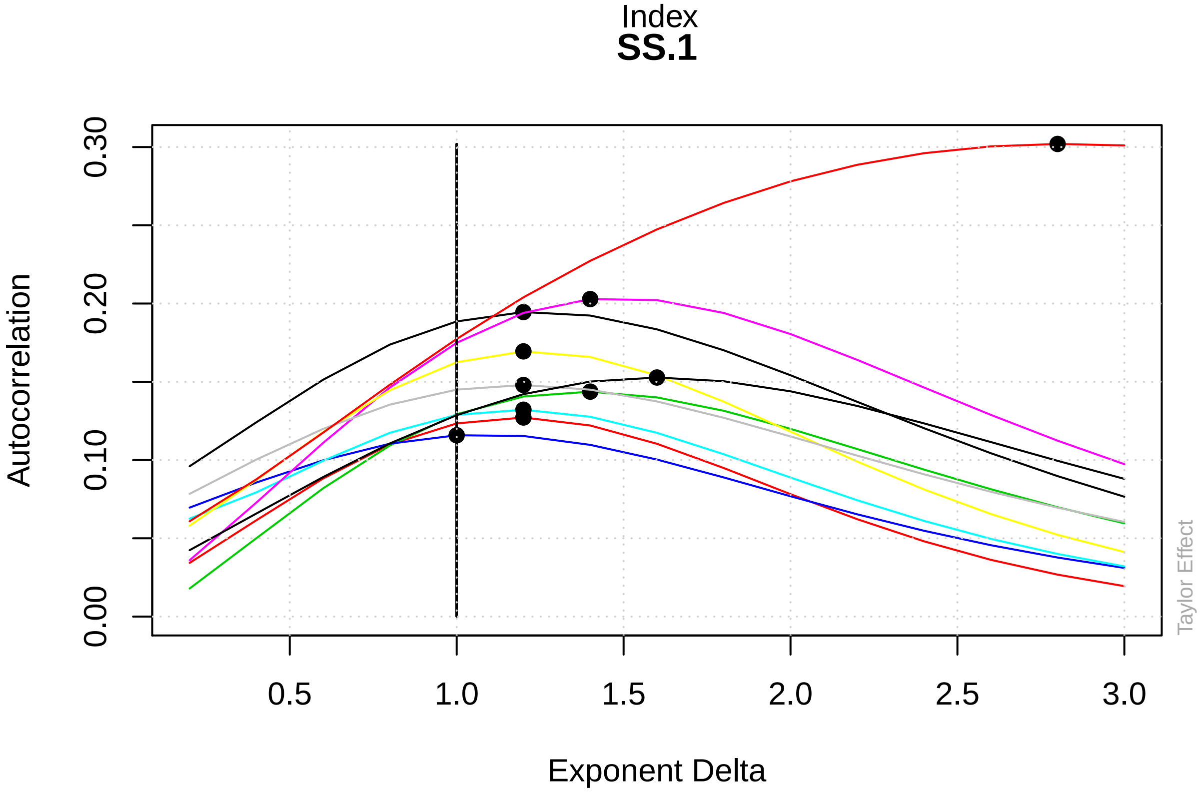
May 20, 2016

Question 1

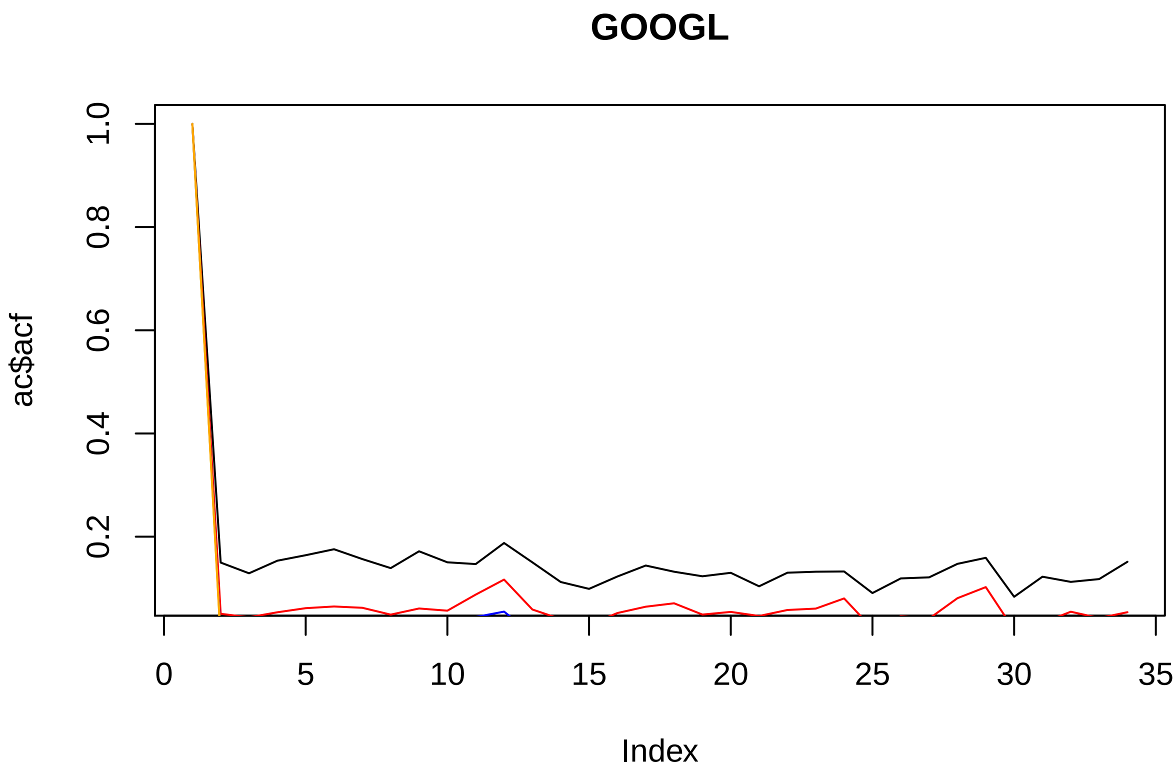
Power 1 is the black line. Other colors are powers 2-5.

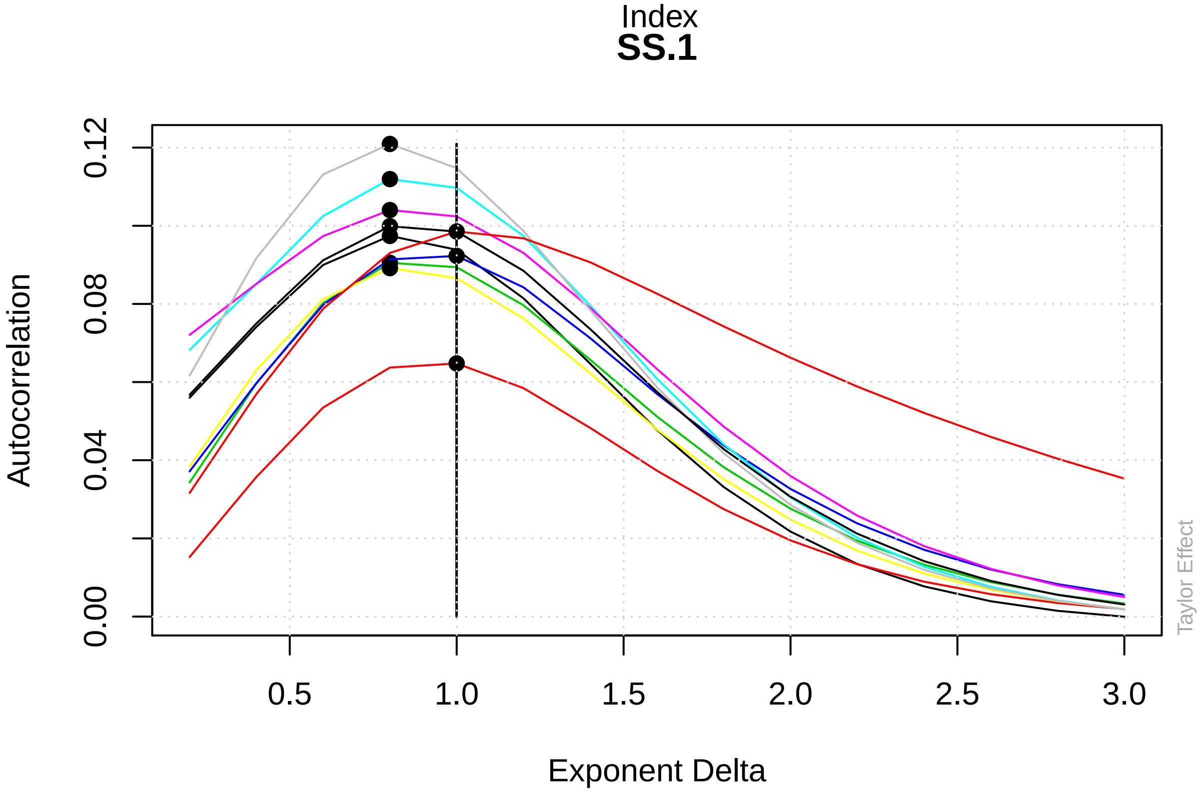
Apple


AAPL
ac$acf
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
Index

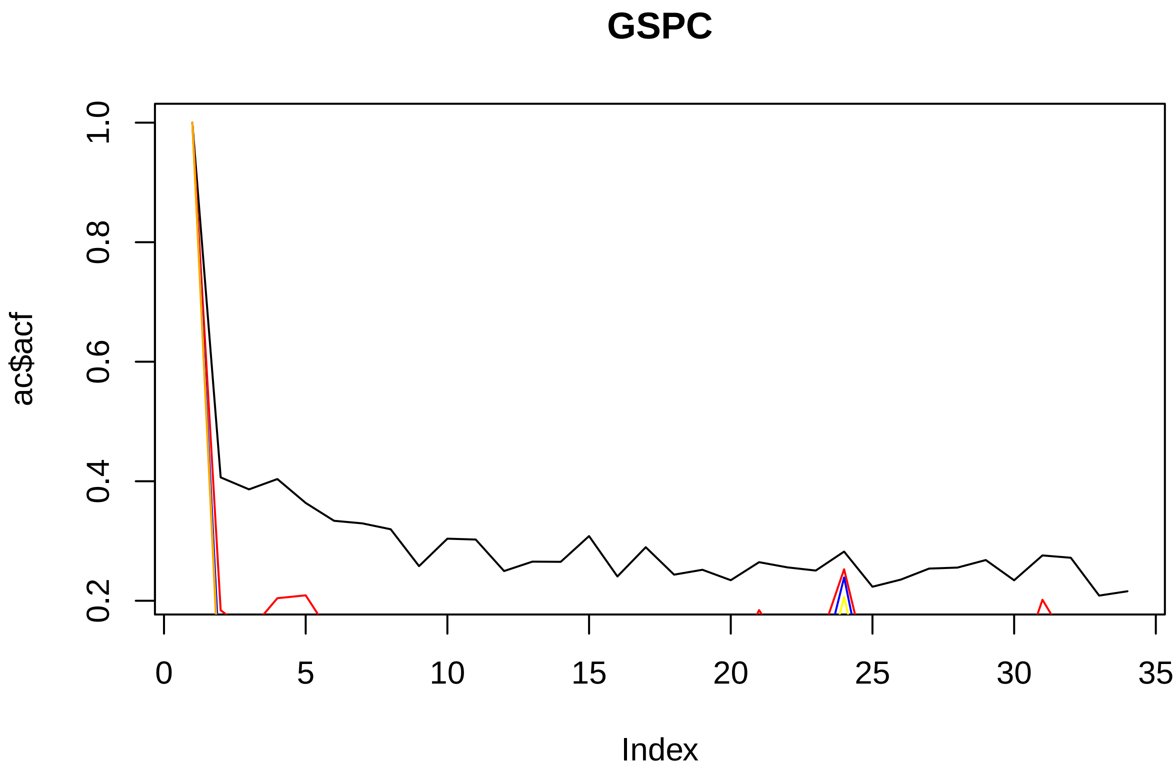

SS.1
Autocorrelation
0.30
0.20
0.10
0.00
0.5
1.0
1.5
2.0
2.5
3.0
Exponent Delta
Taylor Effect

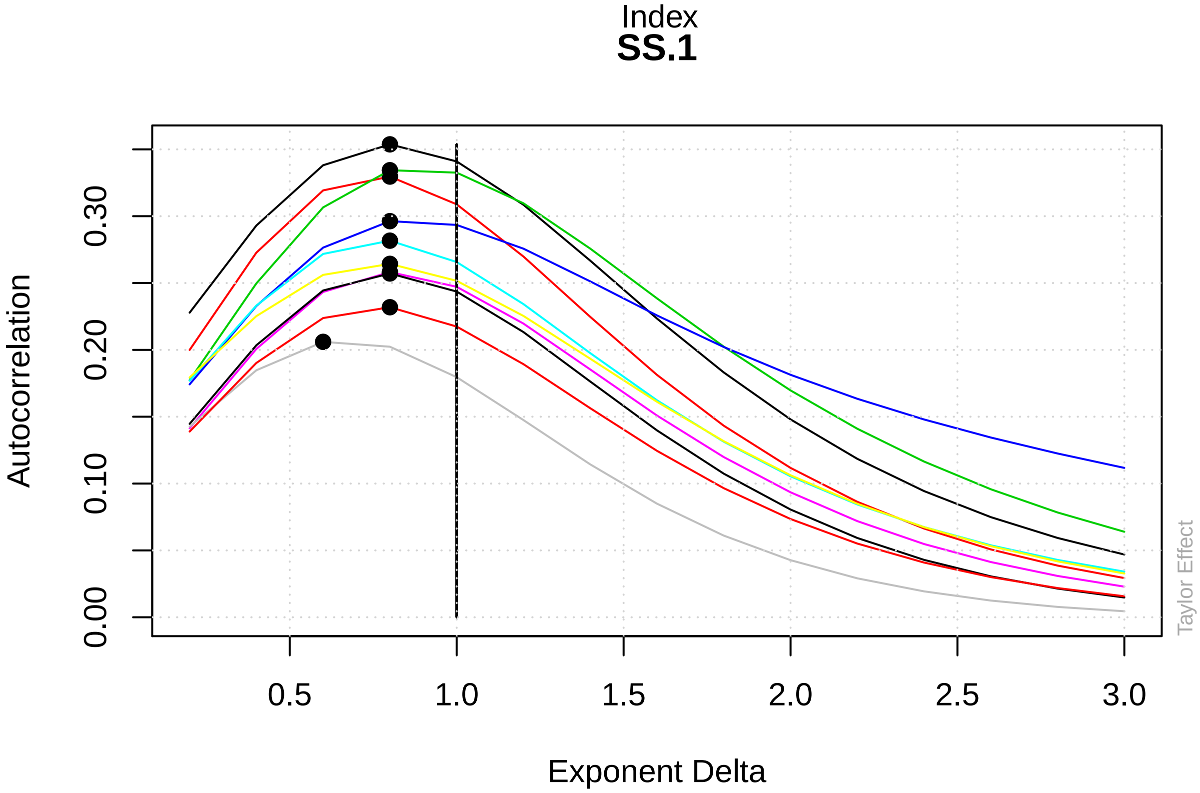
Google


GOOGL
ac$acf
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
Index


SS.1
Autocorrelation
0.12
0.08
0.04
0.00
0.5
1.0
1.5
2.0
2.5
3.0
Exponent Delta
Taylor Effect

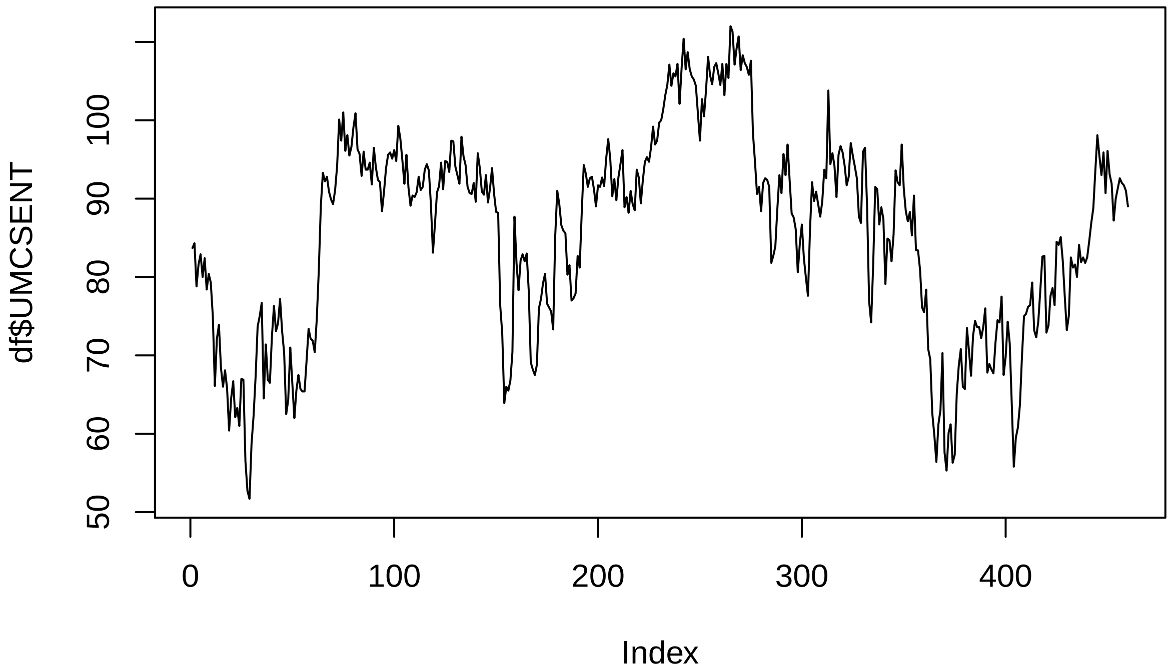
S&P 500

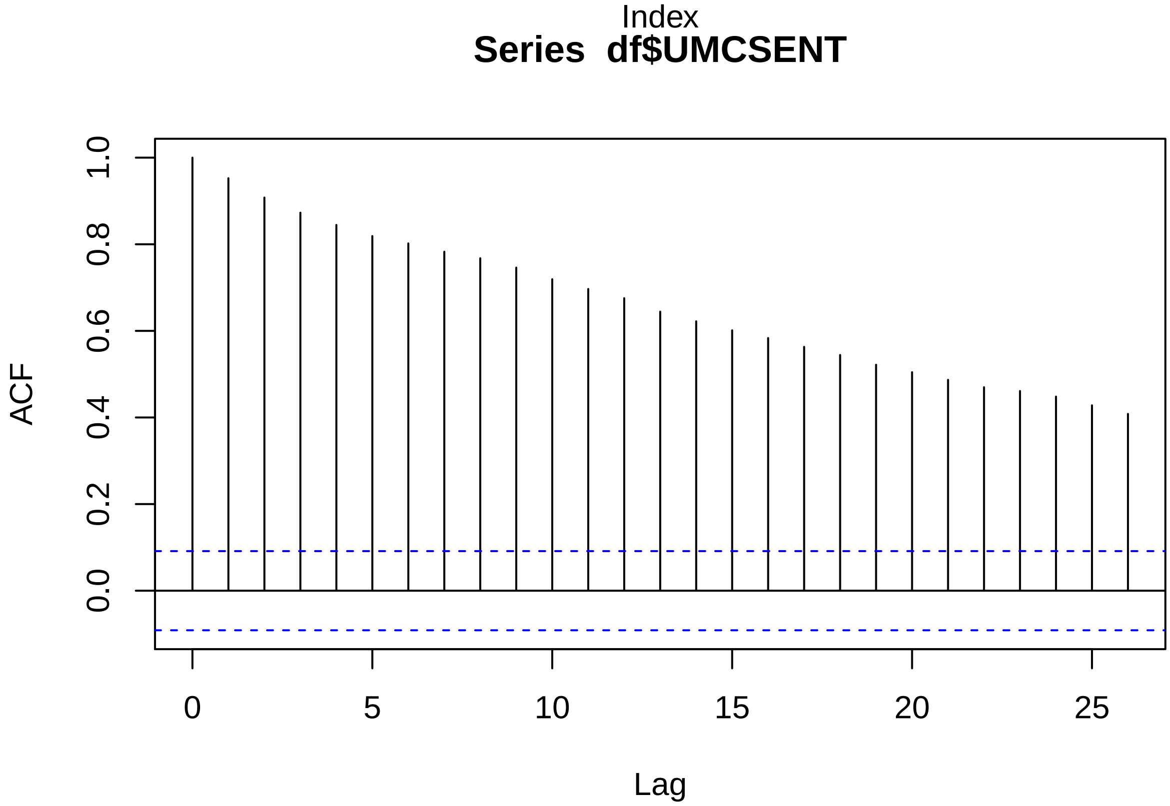

GSPC
ac$acf
1.0
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
Index

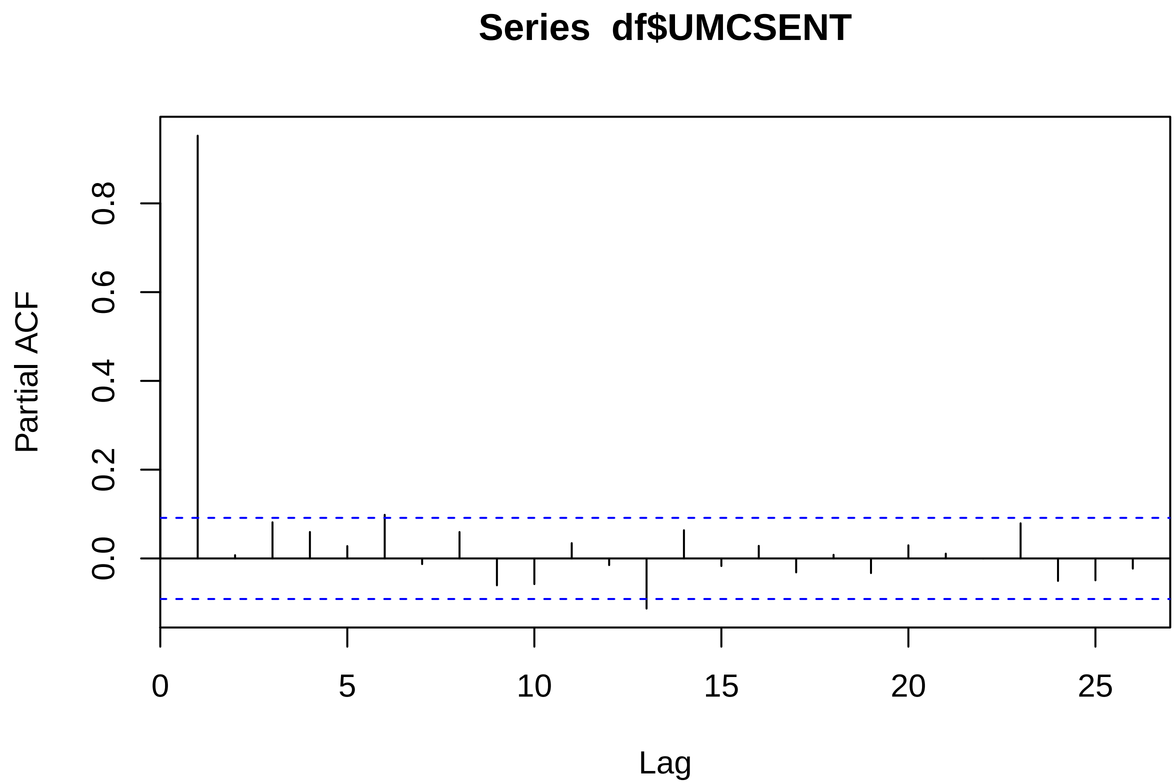

SS.1
Autocorrelation
0.30
0.20
0.10
0.00
0.5
1.0
1.5
2.0
2.5
3.0
Exponent Delta
Taylor Effect

Question 2

```
## [1] "UMCSENT"
```



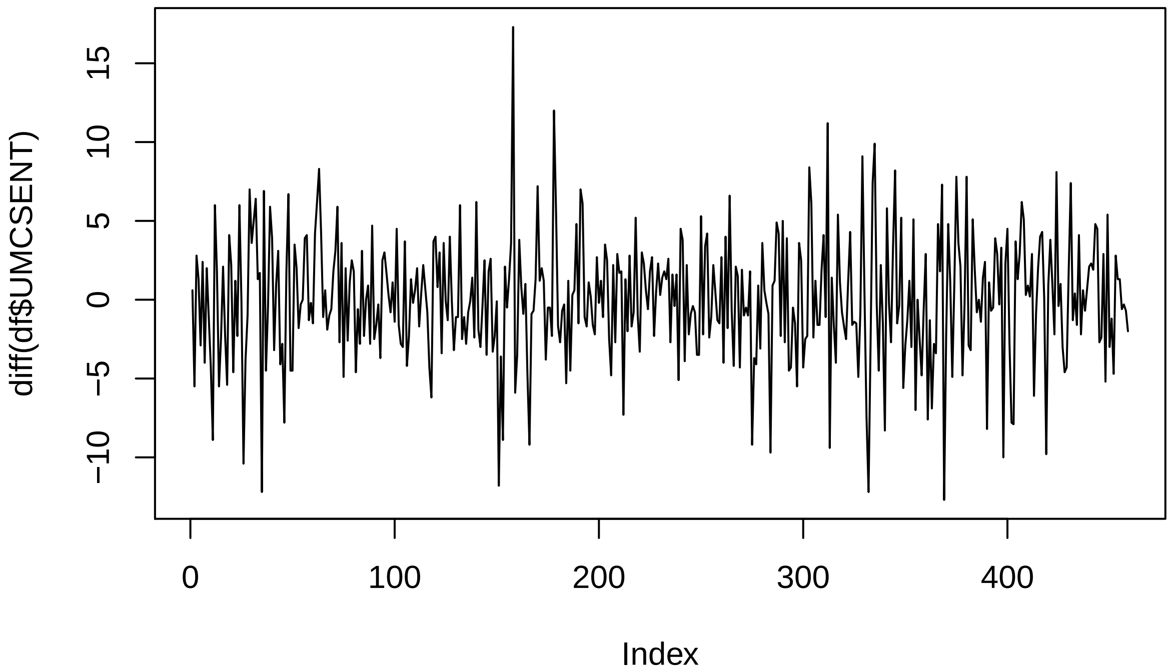


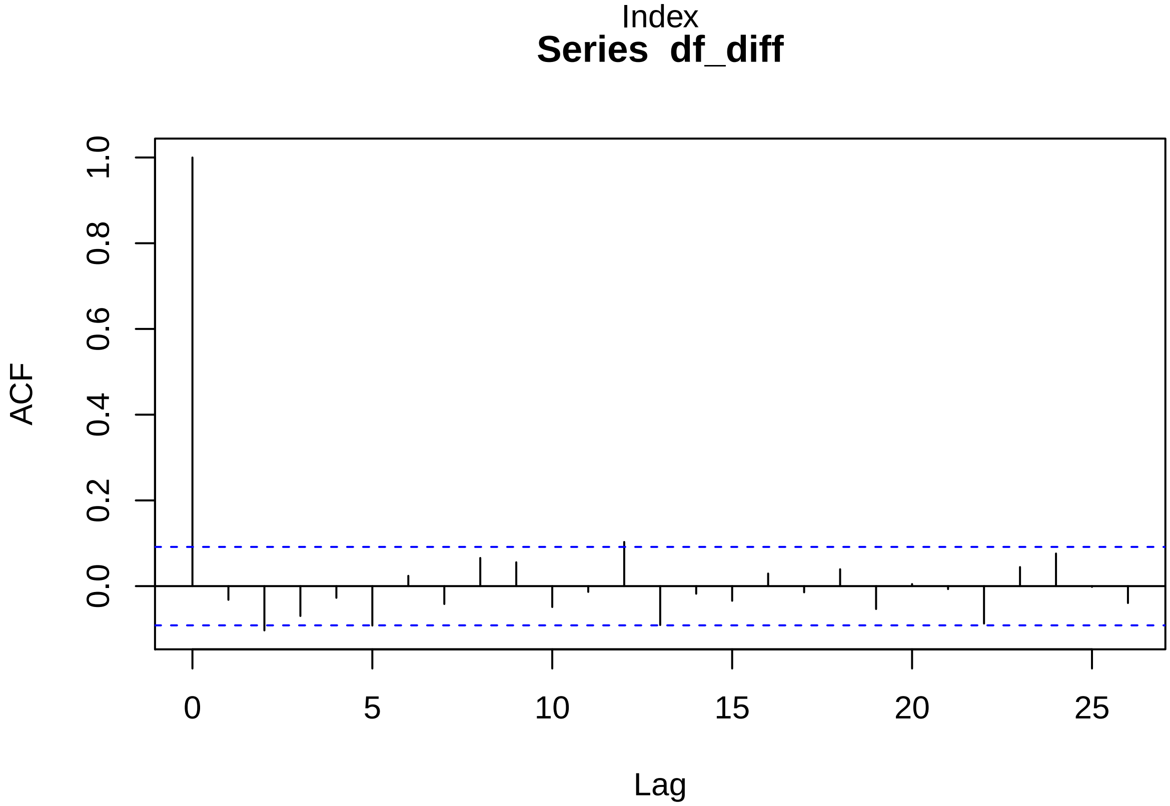


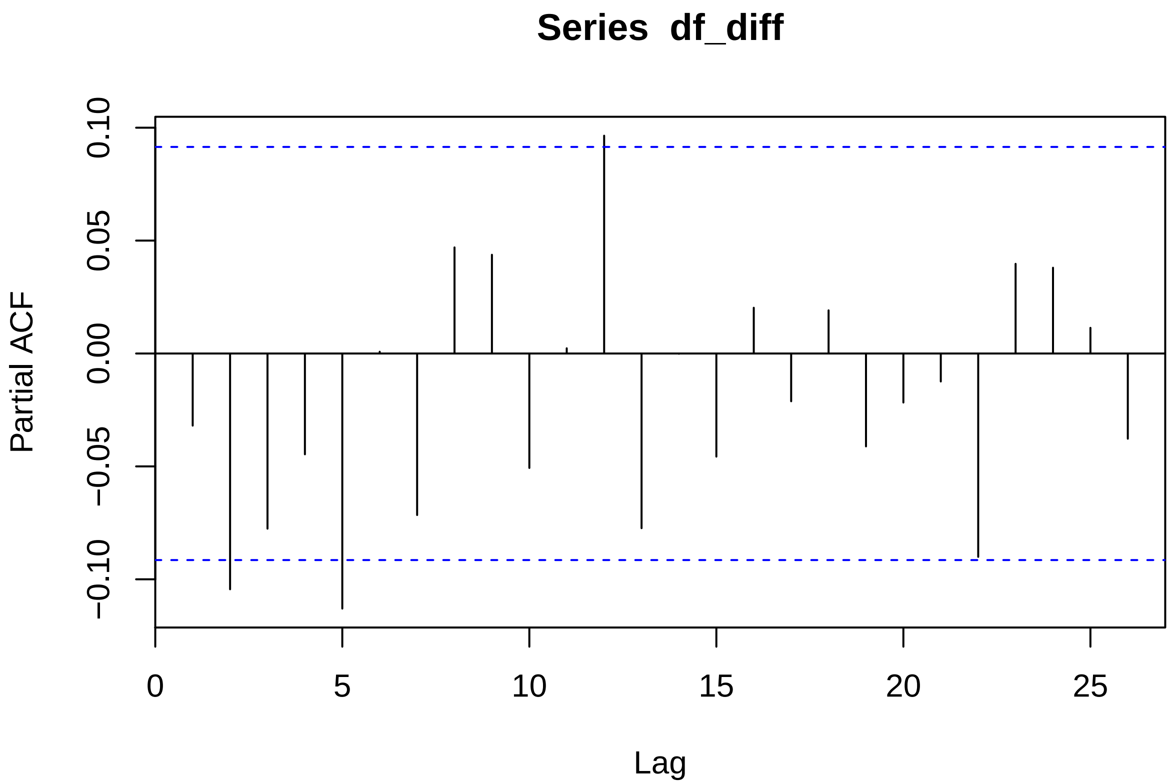
From above plots we ca see the data is not stationary. Confirm with Dickey-Fuller.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df$UMCSENT
## Dickey-Fuller = -2.2297, Lag order = 7, p-value = 0.4808
## alternative hypothesis: stationary
```

Difference the data







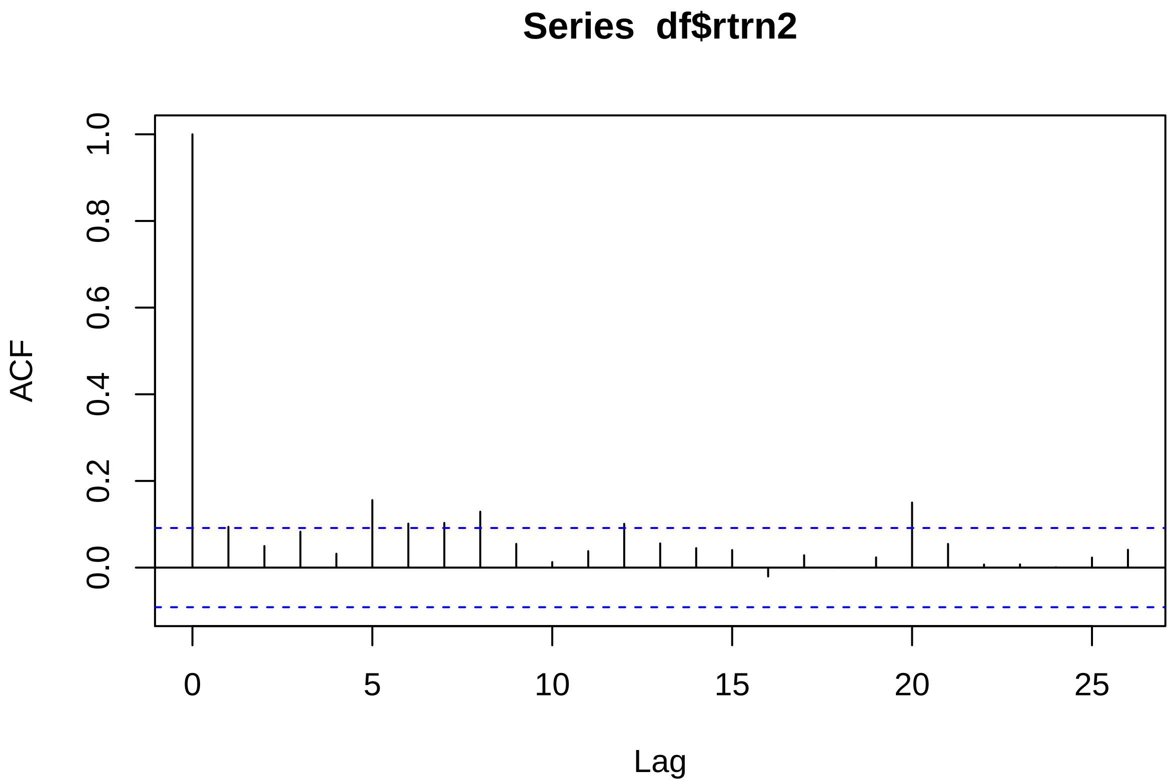

```
## Warning in adf.test(df_diff): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  df_diff
## Dickey-Fuller = -8.5761, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

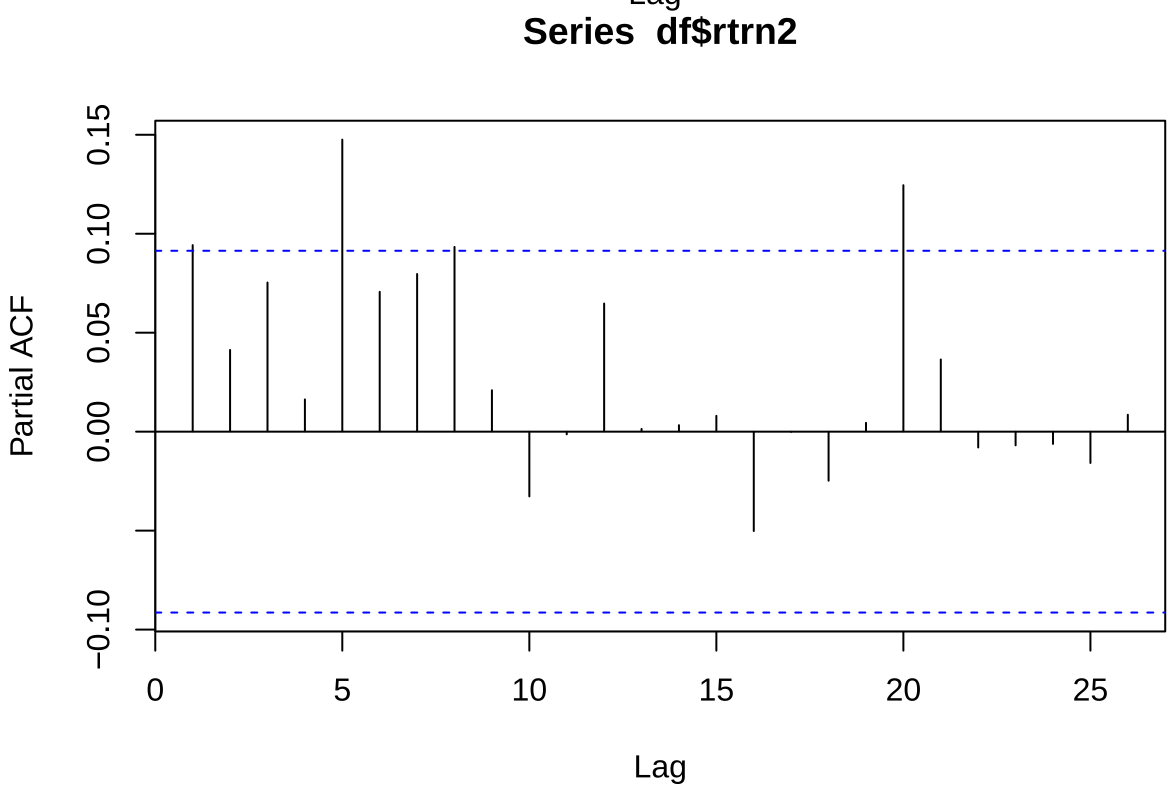
Auto ARIMA wrt AIC

```
## Series: df_diff
## ARIMA(1,0,2) with zero mean
##
## Coefficients:
##          ar1      ma1      ma2
##       0.5755  -0.6311  -0.0934
## s.e.  0.1735   0.1753   0.0567
##
## sigma^2 estimated as 15.22:  log likelihood=-1274.73
## AIC=2557.45   AICc=2557.54   BIC=2573.97
```

Square of returns of UMCSENT

ACF & PACF





Ljung-Box

```
##
##  Box-Ljung test
##
## data:  df$rtrn2
## X-squared = 4.1121, df = 1, p-value = 0.04258
```

ARMA-GARCH

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 1), data = df$rtrn)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x61e95e8>
##  [data = df$rtrn]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1
##  5.9427e-04   7.1608e-01  -8.3085e-01   5.9467e-05   9.6843e-02
##       beta1
##  8.8138e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      5.943e-04   3.937e-04    1.510   0.1312
## ar1     7.161e-01   1.236e-01    5.791 6.98e-09 ***
## ma1    -8.309e-01   9.848e-02   -8.437  < 2e-16 ***
## omega   5.947e-05   2.914e-05    2.041   0.0413 *
## alpha1  9.684e-02   2.418e-02    4.004 6.22e-05 ***
## beta1   8.814e-01   2.551e-02   34.545  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  756.3255    normalized:  1.644186
##
## Description:
##  Fri May 20 21:59:38 2016 by user:
##
##
## Standardised Residuals Tests:
```

```
##                                  Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  38.84378  3.674355e-09
##  Shapiro-Wilk Test  R    W      0.9859023 0.0001941893
##  Ljung-Box Test     R    Q(10)  6.85387   0.7391647
##  Ljung-Box Test     R    Q(15)  15.44574  0.4198112
##  Ljung-Box Test     R    Q(20)  17.9869   0.5882716
##  Ljung-Box Test     R^2  Q(10)  10.41021  0.4052708
##  Ljung-Box Test     R^2  Q(15)  11.73229  0.6991689
##  Ljung-Box Test     R^2  Q(20)  17.89932  0.5940406
##  LM Arch Test       R    TR^2   10.56445  0.566567
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.262285 -3.208399 -3.262619 -3.241066
```

Question 3

NN dividend/price (5 lags for all cases)

```
## $folds
## [1] 0.001811215 0.001365793 0.001792039 0.001544860 0.001277477
##
## $average
## [1] 0.001558277
```

NN PE10

```
## $folds
## [1] 0.001550870 0.002073239 0.001185960 0.001461064 0.001698527
##
## $average
## [1] 0.001593932
```

NN dividend/price + PE10

```
## $folds
## [1] 0.001509842 0.002206664 0.001198036 0.001593195 0.001453325
##
## $average
## [1] 0.001592212
```

SVM dividend/price

```
## $folds
## [1] 0.001822501 0.001404041 0.001825733 0.001516574 0.001294497
##
## $average
## [1] 0.001572669
```

SVM PE10

```
## $folds
## [1] 0.001481373 0.002079052 0.001300448 0.001525307 0.001741522
##
## $average
## [1] 0.00162554
```

SVM dividend/price + PE10

```
## $folds
## [1] 0.001456985 0.002229145 0.001223836 0.001643663 0.001507885
##
## $average
## [1] 0.001612303
```

Tuning Code

```
## h2o deep learning
library(h2o)
localH2O = h2o.init(nthreads=-1)

data_train_h <- as.h2o(data_train,destination_frame = "h2o_data_train")
data_test_h <- as.h2o(data_test,destination_frame = "h2o_data_test")


y <- "target"
x <- setdiff(names(data_train_h), y)


#grid search
hidden_opt <- list(c(200,200), c(100,300,100), c(500,500,500))
l1_opt <- c(1e-5,1e-7)
hyper_params <- list(hidden = hidden_opt, l1 = l1_opt)

model_grid <- h2o.grid("deeplearning",
                        hyper_params = hyper_params,
                        x = (2:ncol(data_train_h)),
                        y = 1,
                        #distribution = "multinomial",
                        training_frame = data_train_h,
                        validation_frame = data_test_h)

# print out the Test MSE for all of the models
for (model_id in model_grid@model_ids) {
  model <- h2o.getModel(model_id)
  mse <- h2o.mse(model, valid = TRUE)
  #mse <- h2o.mse(model, valid = FALSE)
  print(sprintf("Test set MSE: %f", mse))
}
```

```r
h2o.shutdown()

## SVM

# set up the cross-validated hyper-parameter search
svm_grid_1 = expand.grid(
  cost = 10^c(-1,0.5,1.5,2,3,4),
  gamma = 10^c(-3,-2,-1,0,1,2,3)
)

svm_grid_2 = expand.grid(
  cost = 0.11,
  gamma = 0.01
)

# pack the training control parameters
svm_trcontrol_1 = trainControl(
  method = "cv",
  number = 5,
  verboseIter = TRUE,
  returnData = FALSE,
  returnResamp = "all",                # save losses across all models
  #classProbs = TRUE,                   # set to TRUE for AUC to be computed
  #summaryFunction = twoClassSummary,
  summaryFunction = defaultSummary,
  allowParallel = TRUE
)

svm_train_0 = train(
  x = trainset[,-1],
  y = trainset[,1],
  trControl = svm_trcontrol_1,
  tuneGrid = svm_grid_2,
  method = "svmLinear2"
  #kernel = "radial", #radial is default
  #type="eps-regression"
)
```

Question 4

Question 5

Variance of Portfolio Return

Key Concepts:

1) $\rho_{ii} = 1 \forall i \in \{1, 2....n\}$
2) $\rho_{ij} = \rho_{ji} = \frac{\mathrm{Cov(X_i,X_j)}}{\sigma_i \sigma_j} \implies \sigma_i \sigma_j \rho_{ij} = Cov(X_i, X_j)$

We begin by looking at the variance when the value of n is 2.

$$
\begin{aligned}
Var(a_1X_1 + a_2X_2) &= Var(a_1X_1) \ + \ Var(a_2X_2) \ + \ 2Cov(a_1X_1, a_2X_2) \\
&= a_1^2Var(X_1) \ + \ a_2^2Var(X_2) \ + \ 2a_1a_2Cov(X_1, X_2) \\
&= a_1^2\sigma_1^2 \ + \ a_2^2\sigma_2^2 \ + \ 2a_1a_2\sigma_1\sigma_1\rho_{12} \\
&= a_1^2\sigma_1^2 \ + \ a_1a_2\sigma_1\sigma_1\rho_{12} \ + \ a_2^2\sigma_2^2 \ + \ a_2a_1\sigma_2\sigma_1\rho_{21} \\
&= a_1a_1\sigma_1\sigma_1\rho_{11} \ + \ a_1a_2\sigma_1\sigma_1\rho_{12} \ + \ a_2a_2\sigma_2\sigma_2\rho_{22} \ + \ a_2a_1\sigma_2\sigma_1\rho_{21} \\
&= \textstyle\sum_{j=1}^{2} a_1a_j\sigma_1\sigma_j\rho_{1j} \ + \ \sum_{j=1}^{2} a_2a_j\sigma_2\sigma_j\rho_{2j} \\
&= \textstyle\sum_{i=1}^{2}\sum_{j=1}^{2} a_ia_j\sigma_i\sigma_j\rho_{ij}
\end{aligned}
$$

Now, Assume the formula holds for some unspecified value of n = k. It must then be shown that the formula holds for n = k+1, that is:

$$
Var\left(\sum_{i=1}^{k+1} a_iX_i\right) \ = \ \sum_{i=1}^{k+1}\sum_{j=1}^{k+1} a_ia_j\sigma_i\sigma_j\rho_{ij}
$$

Using the induction hypothesis that the formula holds for n = k, the left-hand side can be rewritten to:

$$
\begin{aligned}
Var\left(\textstyle\sum_{i=1}^{k+1} a_iX_i\right) &= Var\left(\textstyle\sum_{i=1}^{k} a_iX_i + a_{k+1}X_{k+1}\right) \\
&= \textstyle\sum_{i=1}^{k}\sum_{j=1}^{k} a_ia_j\sigma_i\sigma_j\rho_{ij} + Var(a_{k+1}X_{k+1}) \ + \ 2Cov(\sum_{i=1}^{k} a_iX_i, a_{k+1}X_{k+1}) \\
&= \textstyle\sum_{i=1}^{k}\sum_{j=1}^{k} a_ia_j\sigma_i\sigma_j\rho_{ij} + a_{k+1}a_{k+1}\sigma_{k+1}\sigma_{k+1}\rho_{k+1k+1} \ + \ 2\sum_{i=1}^{k} Cov(a_iX_i, a_{k+1}X_{k+1}) \\
&= \textstyle\sum_{i=1}^{k}\sum_{j=1}^{k} a_ia_j\sigma_i\sigma_j\rho_{ij} + a_{k+1}a_{k+1}\sigma_{k+1}\sigma_{k+1}\rho_{k+1k+1} \ + \ 2\sum_{i=1}^{k} a_ia_{k+1}\sigma_i\sigma_{k+1}\rho_{ik+1} \\
&= \textstyle\sum_{i=1}^{k+1}\sum_{j=1}^{k+1} a_ia_j\sigma_i\sigma_j\rho_{ij}
\end{aligned}
$$

Hence,

$$
Var\left(\sum_{i=1}^{2} a_iX_i\right) \ = \ \sum_{i=1}^{2}\sum_{j=1}^{2} a_ia_j\sigma_i\sigma_j\rho_{ij}
$$

$$
Var\left(\sum_{i=1}^{k+1} a_iX_i\right) \ = \ \sum_{i=1}^{k+1}\sum_{j=1}^{k+1} a_ia_j\sigma_i\sigma_j\rho_{ij}
$$

We can generalize the above result to n terms and conclude:

$$
Var\left(\sum_{i=1}^{n} a_iX_i\right) \ = \ \sum_{i=1}^{n}\sum_{j=1}^{n} a_ia_j\sigma_i\sigma_j\rho_{ij}
$$

Question 6

Linear Regression on Information Set

Our objective is to prove that the best estimator for X_{t+h} with information set $Z \ = \ (X_t, X_{t-1}, ...X_{t-p})$ when all variables follow $X \ \sim \ N(0, 1)$ is a linear regression on Z.

We use Σ to denote the variance-covariance matrix of X_{t+h} and Z.

$\Sigma \ = \ \mathbb{E}\left[\left(X_{t+h} - \mathbb{E}\left[X_{t+H}\right]\right)^T\left(Z - \mathbb{E}\left[Z\right]\right)\right]$

$$= \mathbb{E}\left[(X_{t+h} - 0)^T (Z - 0)\right]$$
$$= \mathbb{E}\left[(X_{t+h}^T Z\right]$$

Our estimate for $X_{t+h}|Z$ is $\mathbb{E}\left[X_{t+h}|Z\right]$

$$\begin{aligned}
\mathbb{E}\left[X_{t+h}|Z\right] \;&=\; \mathbb{E}X_{t+h} \;+\; \Sigma(\sigma_Z^2)^{-1}\left(Z - \mathbb{E}Z\right) \\
&=\; 0 \;+\; \mathbb{E}\left[X_{t+h}^T Z\right]\mathbb{E}\left[Z - \mathbb{E}Z\right]^{-2}\left(Z - 0\right) \\
&=\; \mathbb{E}\left[X_{t+h}^T Z\right]\mathbb{E}\left[Z^T Z\right]^{-1} Z
\end{aligned}$$

The above is a linear regression on Z, where Z has the form $Z \;=\; \beta Z \;+\; \epsilon$ where $\beta \;=\; \mathbb{E}\left[X_{t+h}^T Z\right]\mathbb{E}\left[Z^T Z\right]^{-1}$