Lydia Yu
912328745
15.780 HW2

Problem 1

```
ufc = read.csv('UFC_data.csv')
```

1)

```
> # 1) check collinearity
> cor(ufc[, 3:ncol(ufc)-1])
                      Tweets   Hashtags       URLs   Mentions Unique_Users Average_Sentiment
Tweets             1.0000000  0.9928522  0.9920856  0.9987944    0.9988430        -0.2260260
Hashtags           0.9928522  1.0000000  0.9767988  0.9932973    0.9875267        -0.1635636
URLs               0.9920856  0.9767988  1.0000000  0.9940740    0.9908729        -0.3299873
Mentions           0.9987944  0.9932973  0.9940740  1.0000000    0.9964259        -0.2381790
Unique_Users       0.9988430  0.9875267  0.9908729  0.9964259    1.0000000        -0.2286064
Average_Sentiment -0.2260260 -0.1635636 -0.3299873 -0.2381790   -0.2286064         1.0000000
```

The correlation coefficients for Average_Sentiment and every other feature are all below the threshold of 0.5 -- there are 5 feature pairs below the threshold.

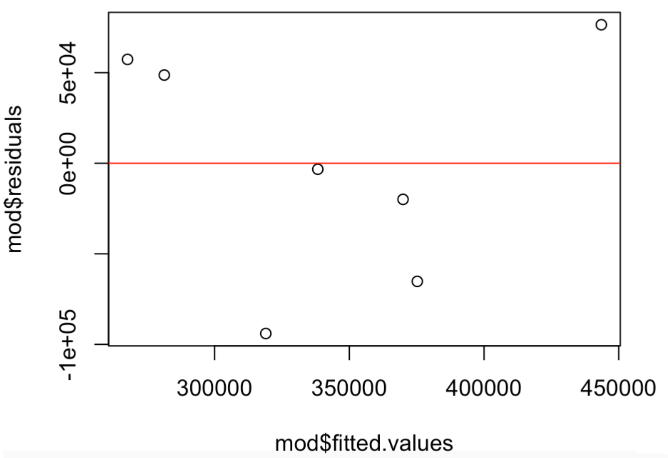
2)

```
#2) lin reg to predict buyrate using tweets
train = ufc[1:7, ]
test = ufc[8:nrow(ufc), ]
mod = lm(Buyrate ~ Tweets, data = train)
```

a. The slope estimate for Tweets is 0.08347, the p-value is 0.09, and the in-sample R^2 is 0.4674 (adjusted to 0.3609)

```
> # a) model slope estimate, p value, and in-sample R^2
> summary(mod)

Call:
lm(formula = Buyrate ~ Tweets, data = train)

Residuals:
     1      2      3      4      5      6      7 
 57317  48684 -19941 -65223  -3301  76464 -93999 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 2.461e+05  5.307e+04   4.638  0.00564 **
Tweets      8.347e-01  3.984e-01   2.095  0.09034 . 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 70710 on 5 degrees of freedom
Multiple R-squared:  0.4674,	Adjusted R-squared:  0.3609 
F-statistic: 4.388 on 1 and 5 DF,  p-value: 0.09034
```

b.




```
# b. plot of fitted values vs residuals in training set
plot(mod$fitted.values, mod$residuals)
abline(h = 0, col = "red")
```

It's difficult to tell whether or not a linear model is a good fit because there are so few data points. On one hand, there are approximately the same number of residuals above 0 as there are below 0, and they don't seem to follow any pattern. However, it's also possible to say that based on this plot, lower Buyrates have positive residuals, while higher Buyrates have negative residuals, and then very high Buyrates have positive residuals again. But because there are only 7 training data points, it's difficult to make a conclusion.

3) **Training** MSE: 3,571,502,518; MAE: 52,132.79; MAPE: 0.167

```
> # training errors:
> MAE(train$Buyrate, train_pred)
[1] 52132.79
> MSE(train$Buyrate, train_pred)
[1] 3571502518
> MAPE(train$Buyrate, train_pred)
[1] 0.1665619
```

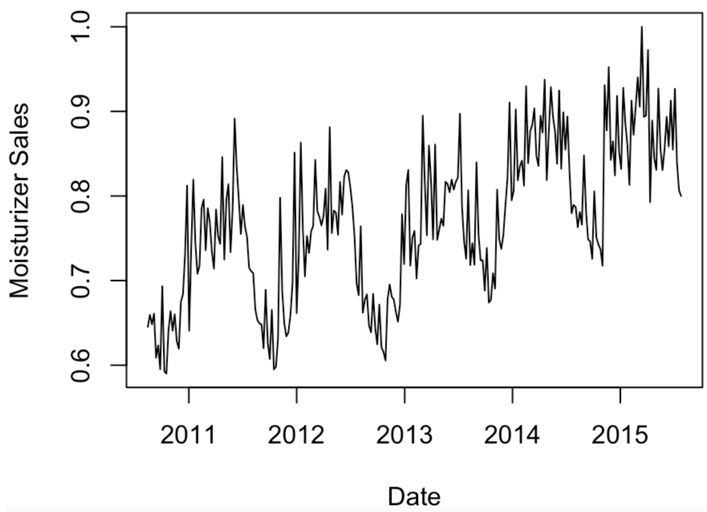
Test MSE: 59,670,355,968; MAE: 154,254.6; MAPE: 0.235

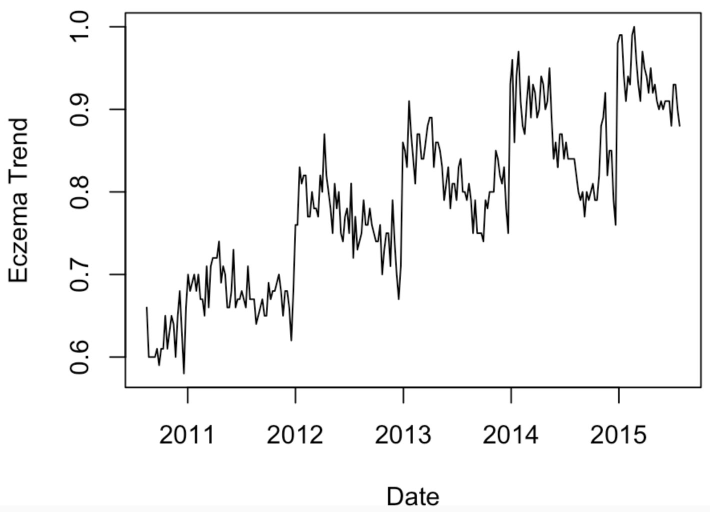
```
> # test errors:
> MAE(test$Buyrate, test_pred)
[1] 154254.6
> MSE(test$Buyrate, test_pred)
[1] 59670355968
> MAPE(test$Buyrate, test_pred)
[1] 0.2349466
```

```
# 3) MSE, MAE, and MAPE of the model on training and test
# Helper functions:
MAE = function(actual, pred) {
  abs_errors = abs(actual - pred)
  return(mean(abs_errors))
}

MSE = function(actual, pred) {
  sq_errors = (actual - pred)^2
  return(mean(sq_errors))
}

MAPE = function(actual, pred) {
  percent_errors = abs(actual - pred) / abs(actual)
  return(mean(percent_errors))
}
# get training and test predictions:
train_pred = predict(mod, newdata = train)
test_pred = predict(mod, newdata = test)
# training errors:
MAE(train$Buyrate, train_pred)
MSE(train$Buyrate, train_pred)
MAPE(train$Buyrate, train_pred)
# test errors:
MAE(test$Buyrate, test_pred)
MSE(test$Buyrate, test_pred)
MAPE(test$Buyrate, test_pred)
```

Problem 2

1)





Both plots have seasonal trends that repeat about every 12 months. The trends are also both generally increasing over time.

```
# PROBLEM 2
sales = read.csv('MoisturizerSalesGoogleTrend.csv')
sales$Date = anytime(sales$Date)

# 1) plot sales vs time and trend vs time
plot(sales$Date, sales$MoisturizerSales, type = "l",
     xlab = "Date", ylab = "Moisturizer Sales")
plot(sales$Date, sales$GoogleTrendVolumeEczema, type = "l",
     xlab = "Date", ylab = "Eczema Trend")
```

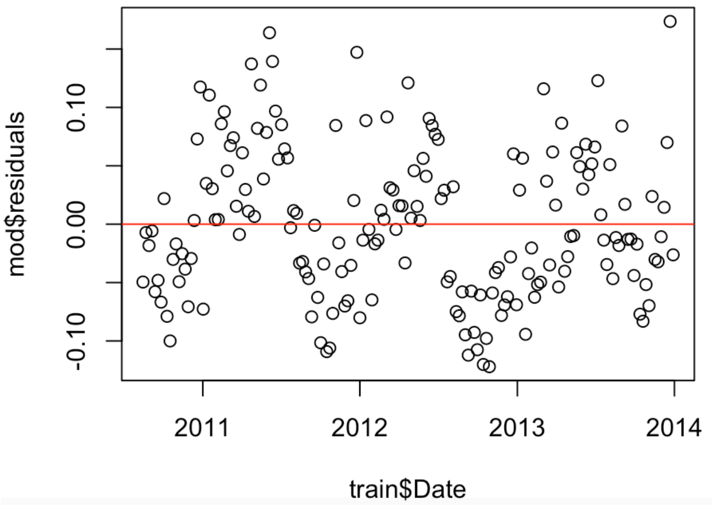
2) slope: 0.468; p-value: 2.7e-12; training MAPE: 0.048; test MAPE: 0.096

```
> # slope, p-value
> summary(mod)

Call:
lm(formula = MoisturizerSales ~ GoogleTrendVolumeEczema, data = train)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.122053 -0.049292 -0.008806  0.045776  0.173635 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)              0.38589    0.04639   8.319 2.43e-14 ***
GoogleTrendVolumeEczema  0.46804    0.06222   7.522 2.70e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06421 on 175 degrees of freedom
Multiple R-squared:  0.2443,	Adjusted R-squared:   0.24 
F-statistic: 56.59 on 1 and 175 DF,  p-value: 2.703e-12
```

```
> MAPE(train$GoogleTrendVolumeEczema, train_pred)
[1] 0.04845156
> MAPE(test$GoogleTrendVolumeEczema, test_pred)
[1] 0.09584963
```

```r
# 2) lin reg to pred moisturizer sales w/ trends data
train = sales[1:177, ]
test = sales[178:nrow(sales), ]
mod = lm(MoisturizerSales ~ GoogleTrendVolumeEczema, data = train)
# slope, p-value
summary(mod)
# MAPE for train/test
train_pred = predict(mod, newdata = train)
test_pred = predict(mod, newdata = test)
MAPE(train$GoogleTrendVolumeEczema, train_pred)
MAPE(test$GoogleTrendVolumeEczema, test_pred)
```

3)



There seems to be a pattern that repeats about every year.

4)

```r
# 4) build time series
# convert google eczema trend to ts (weekly data => freq=52)
eczema = ts(train$GoogleTrendVolumeEczema, frequency = 52)
mod2 = auto.arima(eczema)
```

a. p=3, d=1, q=0

```r
> # a) get params p,d,q of arima model
> arimaorder(mod2)
        p         d         q         P         D         Q Frequency
        2         1         0         1         1         0        52
```

b. MAPE=0.084

```r
> MAPE(train$MoisturizerSales, train_pred)
[1] 0.08384948
```

```r
# b) training MAPE
train_pred = as.vector(mod2$fitted)
MAPE(train$MoisturizerSales, train_pred)
```

5)

```
> summary(mod3)

Call:
lm(formula = residuals ~ GoogleTrendVolumeEczema, data = resid)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.094228 -0.012924  0.003369  0.009234  0.080946 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)  
(Intercept)             -0.04767    0.01880  -2.536   0.0121 *
GoogleTrendVolumeEczema  0.06519    0.02521   2.585   0.0105 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02602 on 175 degrees of freedom
Multiple R-squared:  0.03679,	Adjusted R-squared:  0.03129 
F-statistic: 6.684 on 1 and 175 DF,  p-value: 0.01054
```

```
# 5) reg model to predict resids from eczema trend
# create resid df
resid = data.frame(train$Date, train$GoogleTrendVolumeEczema, as.vector(mod2$residuals))
colnames(resid) = c("Date", "GoogleTrendVolumeEczema", "residuals")
# build reg model
mod3 = lm(residuals ~ GoogleTrendVolumeEczema, data = resid)
summary(mod3)
```

6) MAPE = 0.087

```
> MAPE(train$MoisturizerSales, arima_lm)
[1] 0.08697353
```

```
# 6) add together arima and lin reg preds
# get mod3 resid predictions for the train data
mod3_train_pred = predict(mod3, newdata = resid)
# get arima predictions for the train data
arima_train_pred = predict(mod2, new_data = train)
arima_lm = mod2$fitted + mod3_train_pred
MAPE(train$MoisturizerSales, arima_lm)
```