ECON-UB 251 Econometrics I Assignment 4

Kevin Song

2022-12-08

1.1

```
library(tidyquant)
library(dplyr)
library(ggplot2)
stocks <- tq_get(c("AMZN"), get = "stock.prices", from = "1997-05-15")
head(stocks, 3)
```

```
# A tibble: 3 x 8
  symbol date         open   high    low  close     volume adjusted
  <chr>  <date>      <dbl>  <dbl>  <dbl>  <dbl>      <dbl>    <dbl>
1 AMZN   1997-05-15 0.122  0.125  0.0964 0.0979 1443120000   0.0979
2 AMZN   1997-05-16 0.0984 0.0990 0.0854 0.0865  294000000   0.0865
3 AMZN   1997-05-19 0.0880 0.0885 0.0812 0.0854  122136000   0.0854
```

```
tail(stocks, 3)
```

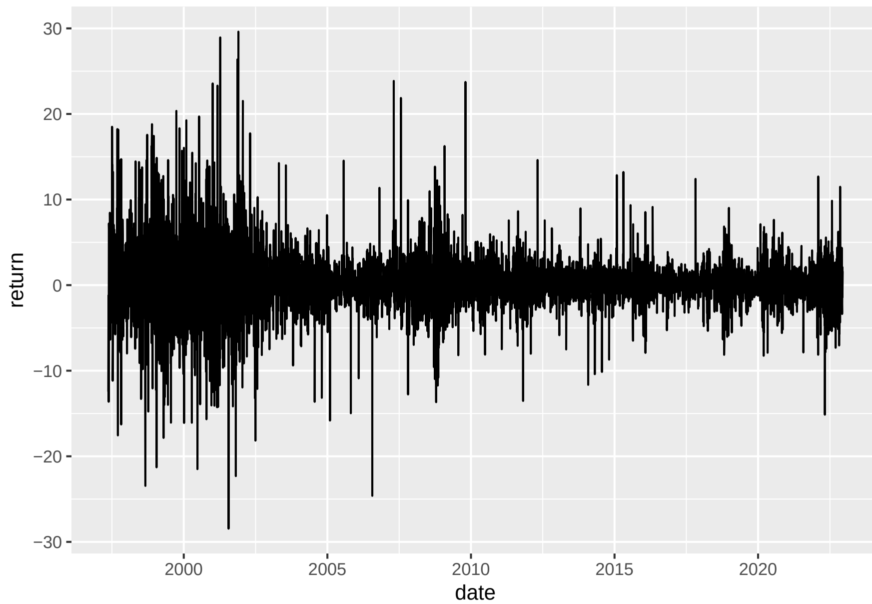
```
# A tibble: 3 x 8
  symbol date        open  high   low close   volume adjusted
  <chr>  <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 AMZN   2022-12-08  89.2  90.9  87.9  90.3 73305900     90.3
2 AMZN   2022-12-09  88.9  90.3  88.6  89.1 67316900     89.1
3 AMZN   2022-12-12  89.2  90.6  87.9  90.6 61852400     90.6
```

I am using stock returns of Amazon (https://finance.yahoo.com/quote/AMZN) from 1997-05-15 to 2022-12-09. I will shorten the starting data after Question 2 in order to align the frequencies in later steps where additional variables will be added (VIX) that do not have data that far back. ## 1.2

```
stocks <- stocks %>%
  group_by(symbol) %>%
  mutate(return = 100 * (log(adjusted) - lag(log(adjusted))),
         range = 100 * (log(high)-log(low)))
tail(stocks)
```

```
# A tibble: 6 x 10
# Groups:   symbol [1]
  symbol date        open  high   low close   volume adjusted return range
  <chr>  <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>  <dbl> <dbl>
1 AMZN   2022-12-05  93.1  94.1  90.8  91.0 71535500     91.0 -3.37   3.51
2 AMZN   2022-12-06  90.5  91.0  87.9  88.2 75503600     88.2 -3.08   3.51
3 AMZN   2022-12-07  88.3  89.9  87.5  88.5 68086900     88.5  0.238  2.72
4 AMZN   2022-12-08  89.2  90.9  87.9  90.3 73305900     90.3  2.11   3.33
5 AMZN   2022-12-09  88.9  90.3  88.6  89.1 67316900     89.1 -1.40   1.87
6 AMZN   2022-12-12  89.2  90.6  87.9  90.6 61852400     90.6  1.63   3.04
```

```r
plot1 <- ggplot(stocks, aes(date, return)) + geom_line()
plot1
```

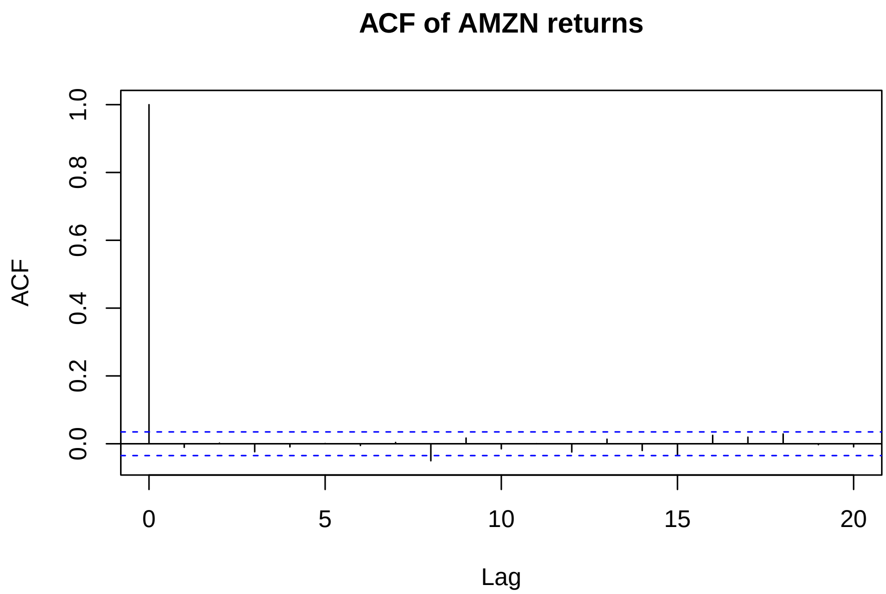


The variation of return ranges from -28 to 30%. The data does not appear to be persistent, as there are no distinct trends that can be observed. There is not a single year where the returns have a definitive trend upward or downward. However, the data is stationary, as despite the high variance the data does stay at around 0%.

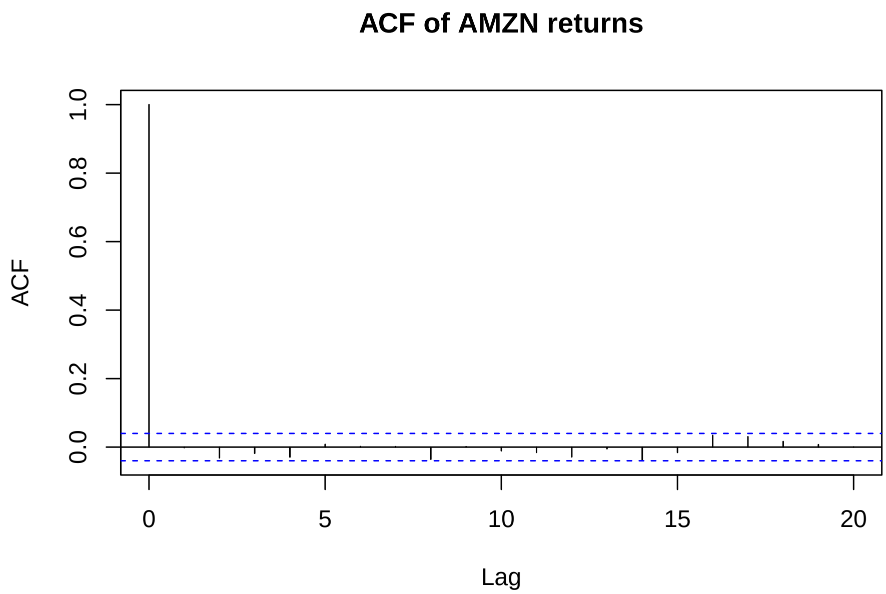
During recessions, there is are extremely low points of returns such as during the dot com 2001 recession and 2008 Lehman collapse, but they are always followed by equally high rebound. Covid had less variation but followed this trend too.

1.3

```r
stocks <- filter(stocks, date > "2010-05-31")
stocks <- stocks %>%
  group_by(symbol) %>%
  mutate(return = 100 * (log(adjusted) - lag(log(adjusted))))
return2 <- 100 * (log(stocks[,8]) - lag(log(stocks[,8])))
acff2<-  acf( return2, main = "ACF of AMZN returns",  na.action = na.omit, lag.max = 20)
```

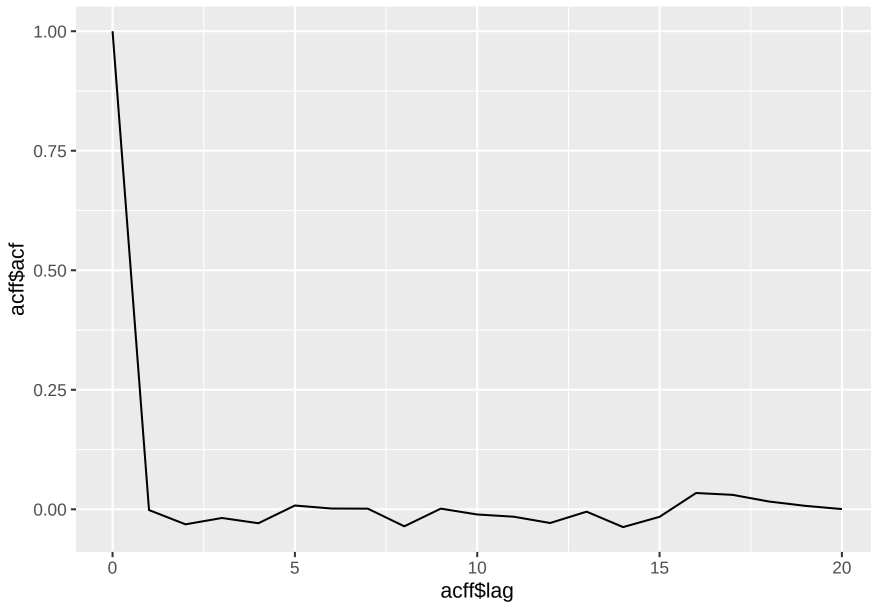



```
stocks <- filter(stocks, date < "2019-12-31")
return3 <- 100 * (log(stocks[,8]) - lag(log(stocks[,8])))
acff3<-  acf(return3, main = "ACF of AMZN returns",  na.action = na.omit, lag.max = 20)
```




```
acff <- acf(filter(stocks, symbol == "AMZN", date < as.Date("2019-12-31"))$return,
lag.max = 20, plot = FALSE, na.action = na.omit)

qplot(acff$lag, acff$acf, geom = "line")
```



Comparing the ACF graphs with and without data during the pandemic results in near identical graphs. Overall, there appear to be no significant lags.

1.4

```
library(aTSA)
adf.test(log(filter(stocks, date < "2019-12-31")$return), nlag = 2)
```

```
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag   ADF p.value
[1,]   0 -35.2    0.01
[2,]   1 -23.6    0.01
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -36.3    0.01
[2,]   1 -24.7    0.01
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -36.9    0.01
[2,]   1 -25.3    0.01
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

Since the p-values are all less than 0.05, we reject the null hypothesis that the data is non-stationary. A drift with no trend should be used, and the series can modeled as log(lag) as it is.

1.5

```
library(dplyr)
ar1 <- lm(stocks$return ~ lag(stocks$return), data = stocks)
```

```r
summary(ar1)
```

```
Call:
lm(formula = stocks$return ~ lag(stocks$return), data = stocks)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.6512  -0.9241  -0.0055   1.0074  14.5124 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)   
(Intercept)         0.111442   0.039534   2.819  0.00486 **
lag(stocks$return) -0.001595   0.020370  -0.078  0.93760   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.938 on 2409 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  2.545e-06, Adjusted R-squared:  -0.0004126 
F-statistic: 0.00613 on 1 and 2409 DF,  p-value: 0.9376
```

```r
ar9 <- lm(return ~ lag(return) + lag(return, 2) + lag(return, 3) +lag(return, 4) + lag(return, 5) + lag
summary(ar9)
```

```
Call:
lm(formula = return ~ lag(return) + lag(return, 2) + lag(return, 
    3) + lag(return, 4) + lag(return, 5) + lag(return, 6) + lag(return, 
    7) + lag(return, 8) + lag(return, 9), data = stocks)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.4049  -0.9122  -0.0090   1.0161  14.6211 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)   
(Intercept)     0.1247976  0.0401353   3.109   0.0019 **
lag(return)    -0.0022225  0.0204433  -0.109   0.9134   
lag(return, 2) -0.0315456  0.0204299  -1.544   0.1227   
lag(return, 3) -0.0173967  0.0204415  -0.851   0.3948   
lag(return, 4) -0.0280848  0.0204437  -1.374   0.1696   
lag(return, 5)  0.0054820  0.0204435   0.268   0.7886   
lag(return, 6) -0.0021780  0.0204334  -0.107   0.9151   
lag(return, 7)  0.0007284  0.0204046   0.036   0.9715   
lag(return, 8) -0.0363823  0.0203913  -1.784   0.0745 . 
lag(return, 9)  0.0016897  0.0204009   0.083   0.9340   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.935 on 2393 degrees of freedom
  (10 observations deleted due to missingness)
Multiple R-squared:  0.003357,  Adjusted R-squared:  -0.000391 
F-statistic: 0.8957 on 9 and 2393 DF,  p-value: 0.5282
```

```r
bic <- data.frame(c(BIC(ar1), BIC(ar9)))
rownames(bic) = c("AR1","AR9")
colnames(bic) = c("BIC")
bic
```

```
         BIC
AR1 10053.78
AR9 10068.12
```

From the BIC table, AR1 actually has a lower penalty and is a better method, if only by the slightest margin. From the summary, none of the lags are significant though, so we'll just use AR1 for future questions.

Adjusted r-squared for AR1 is actually negative: -0.0004126 , which is valid for adjusted but also means almost none of the variance in residuals is explained by lag. The fit is useless.

1.6

I believe EBITDA could be a relevant predictor for stock price. Increasing EBITDA means a company is in a better financial state and means the stock may have lower risk, thus increasing its appeal to investors and possibly increasing the price.

I obtained Amazon's daily close vix (https://fred.stlouisfed.org/series/VXAZNCLS).

```r
library(dplyr)
library(aTSA)
library(alfred)
vix <- get_fred_series("VXAZNCLS", observation_start = "2010-06-01", observation_end = "2019-12-31")
colnames(vix) = c("date", "Volatility")

merged <- inner_join(stocks, vix, by="date")
merged
```

```
# A tibble: 2,413 x 11
# Groups:   symbol [1]
   symbol date        open  high   low close volume adjus~1 return range Volat~2
   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>  <dbl> <dbl>   <dbl>
 1 AMZN   2010-06-01  6.25  6.33  6.15  6.16 7.32e7    6.16 NA      2.84    51.2
 2 AMZN   2010-06-02  6.20  6.32  6.08  6.32 9.53e7    6.32  2.46   3.85    46.7
 3 AMZN   2010-06-03  6.31  6.46  6.24  6.44 1.06e8    6.44  1.92   3.39    45.6
 4 AMZN   2010-06-04  6.32  6.41  6.11  6.14 1.10e8    6.14 -4.76   4.81    51.1
 5 AMZN   2010-06-07  6.29  6.33  6.08  6.10 1.31e8    6.10 -0.621  3.98    53.1
 6 AMZN   2010-06-08  6.1   6.1   5.79  5.94 2.30e8    5.94 -2.63   5.22    53.8
 7 AMZN   2010-06-09  6.02  6.07  5.87  5.90 1.47e8    5.90 -0.786  3.44    53.4
 8 AMZN   2010-06-10  6     6.18  5.96  6.16 1.21e8    6.16  4.40   3.54    49.7
 9 AMZN   2010-06-11  6.07  6.18  6.01  6.15 8.41e7    6.15 -0.146  2.66    48.0
10 AMZN   2010-06-14  6.21  6.28  6.18  6.19 7.85e7    6.19  0.648  1.77    47.6
# ... with 2,403 more rows, and abbreviated variable names 1: adjusted,
#   2: Volatility
```

```r
ar1 <- lm(merged$Volatility ~ lag(merged$Volatility), data = merged)
summary(ar1)
```

```

Call:
lm(formula = merged$Volatility ~ lag(merged$Volatility), data = merged)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-20.0230  -0.8403  -0.0419   0.8957  13.9263 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)            1.259169   0.188224    6.69 2.77e-11 ***
lag(merged$Volatility) 0.961135   0.005565  172.72  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.395 on 2410 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9253,    Adjusted R-squared:  0.9252 
F-statistic: 2.983e+04 on 1 and 2410 DF,  p-value: < 2.2e-16
```

```
ar4 <- lm(Volatility ~ lag(Volatility) + lag(Volatility, 2) + lag(Volatility, 3) +lag(Volatility, 4), da
summary(ar4)
```

```
Call:
lm(formula = Volatility ~ lag(Volatility) + lag(Volatility, 2) + 
    lag(Volatility, 3) + lag(Volatility, 4), data = merged)

Residuals:
     Min       1Q   Median       3Q      Max 
-20.1094  -0.8265  -0.0318   0.8788  14.1115 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)         1.218629   0.192868   6.318 3.14e-10 ***
lag(Volatility)     0.915528   0.020369  44.947  < 2e-16 ***
lag(Volatility, 2)  0.061219   0.027611   2.217   0.0267 *  
lag(Volatility, 3) -0.011695   0.027619  -0.423   0.6720    
lag(Volatility, 4) -0.002702   0.020368  -0.133   0.8945    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.39 on 2404 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.9254,    Adjusted R-squared:  0.9253 
F-statistic:  7459 on 4 and 2404 DF,  p-value: < 2.2e-16
```

```
bic <- data.frame(c(BIC(ar1), BIC(ar4)))
rownames(bic) = c("AR1","AR4")
colnames(bic) = c("BIC")
bic
```

```
         BIC
AR1 11078.79
AR4 11076.49
```

Testing VIX shows that the data is stationary with drift and no trend.This makes sense as VIX is a calculation usually within a certain range. BIC states that AR4 is a better model than AR1, but from the autocorrelation summary, only the 1st lag is significant, so we will also use AR1 for VIX for the ADL(p,q) model.

```r
adl.11 <- lm(return ~ dplyr::lag(return) + dplyr::lag(Volatility), data = merged)
summary(adl.11)
```

```
Call:
lm(formula = return ~ dplyr::lag(return) + dplyr::lag(Volatility),
    data = merged)

Residuals:
     Min       1Q   Median       3Q      Max
-13.7931  -0.9182   0.0140   1.0183  14.4278

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -0.0791657  0.1530600  -0.517    0.605
dplyr::lag(return)      0.0005176  0.0204331   0.025    0.980
dplyr::lag(Volatility)  0.0058284  0.0045215   1.289    0.198

Residual standard error: 1.938 on 2408 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.0006921, Adjusted R-squared:  -0.0001379
F-statistic: 0.8339 on 2 and 2408 DF,  p-value: 0.4345
```

```r
bic.2 <- data.frame(c(BIC(adl.11)))
rownames(bic.2) = c("ADL(1,1)")
colnames(bic.2) = c("BIC")
bic.2
```

```
              BIC
ADL(1,1) 10059.91
```

In ADL(1,1), both estimated coefficients are insignificant at 0.05 level. The BIC is also slightly higher than AR(1), of which both are above 10000. In conclusion, this model is also useless.

1.7

```r
tail(merged)
```

```
# A tibble: 6 x 11
# Groups:   symbol [1]
  symbol date        open  high   low close volume adjus~1  return range Volat~2
  <chr>  <date>     <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>   <dbl> <dbl>   <dbl>
1 AMZN   2019-12-20  90.0  90.1  89.1  89.3 1.03e8    89.3 -0.323  1.14     19.4
2 AMZN   2019-12-23  89.4  89.7  89.2  89.7 4.27e7    89.7  0.363  0.475    19.9
3 AMZN   2019-12-24  89.7  89.8  89.4  89.5 1.76e7    89.5 -0.212  0.446    19.9
4 AMZN   2019-12-26  90.1  93.5  90.0  93.4 1.20e8    93.4  4.35   3.87     25.6
5 AMZN   2019-12-27  94.1  95.1  93.3  93.5 1.24e8    93.5  0.0551 1.88     25.8
6 AMZN   2019-12-30  93.7  94.2  92.0  92.3 7.35e7    92.3 -1.23   2.33     25.7
# ... with abbreviated variable names 1: adjusted, 2: Volatility
```

```r
filter(merged, date > "2019-12-26")
```

```
# A tibble: 2 x 11
# Groups:   symbol [1]
  symbol date        open  high   low close volume adjus~1  return range Volat~2
  <chr>  <date>     <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>   <dbl> <dbl>   <dbl>
```

```
1 AMZN   2019-12-27  94.1  95.1  93.3  93.5 1.24e8    93.5  0.0551  1.88    25.8
2 AMZN   2019-12-30  93.7  94.2  92.0  92.3 7.35e7    92.3 -1.23    2.33    25.7
# ... with abbreviated variable names 1: adjusted, 2: Volatility
```

```r
forecast.ar <- as.numeric(coef(ar4)[1] + coef(ar4)[2]*-1.23283301)

forecast.adl <- as.numeric(coef(adl.11)[1] + coef(adl.11)[2]*-1.23283301 +
coef(adl.11)[3]*25.73)
forecast <- data.frame(c(forecast.ar, forecast.adl))
colnames(forecast) = c("forecast of Amazon return")
rownames(forecast) = c("AR(1)", "ADL(1,1)")
forecast
```

```
         forecast of Amazon return
AR(1)                   0.08993659
ADL(1,1)                0.07016014
```

AR(1) forecasts that compared to 2019-12-30, Amazon's return will change from -1.2328 to 0.0899. ADL(1,1) predicts it will change to 0.07016.

1.8

The RSE of AR(1) is 2.395, so the 90% forecast interval is (0.0899 +- 1.645 * 2.395) = (-3.85, 4.03) The RSE of ADL(1,1) is 1.938, so the 90% forecast interval is (0.07016 +- 1.645 * 1.938) = (-3.12, 3.26)