Assignment 6

Allison Tessman

2024-02-22

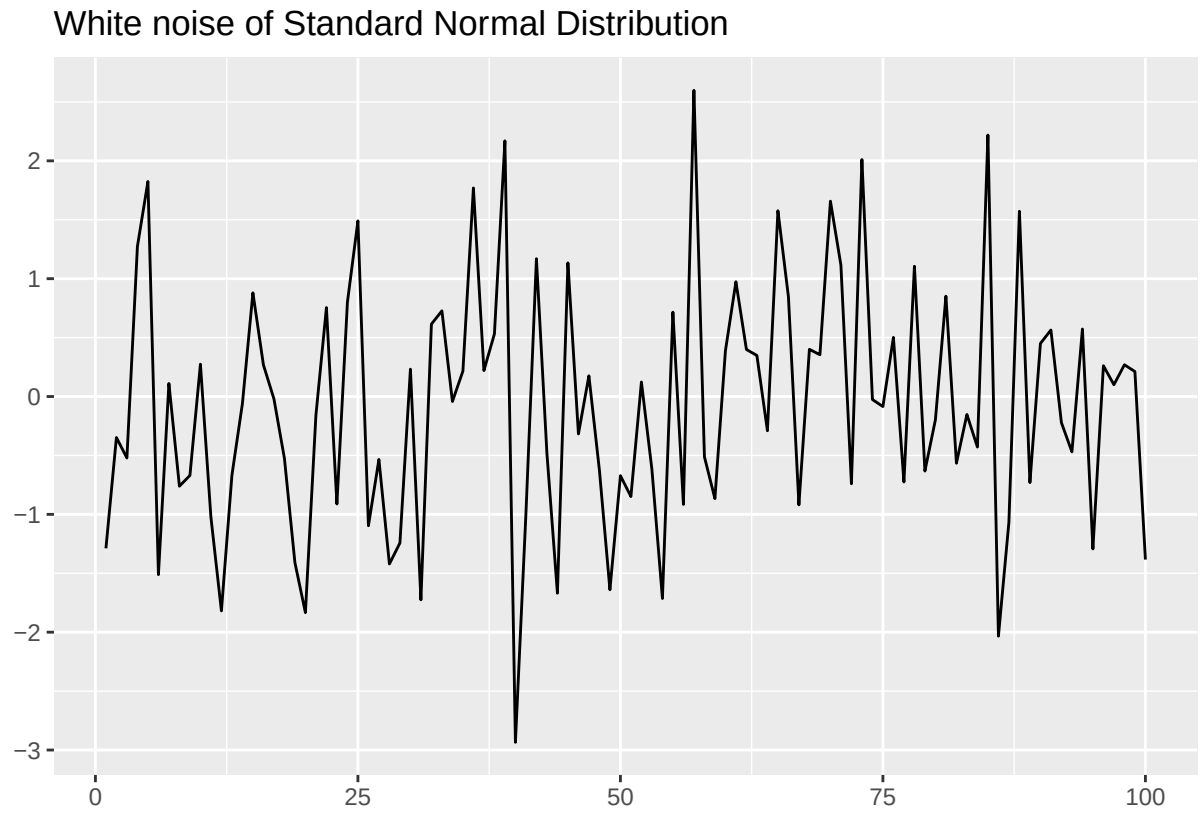
1. Use the codes in the slides to plot the ACF of:

- a white-noise that follows the standard normal distribution (using rnorm function)

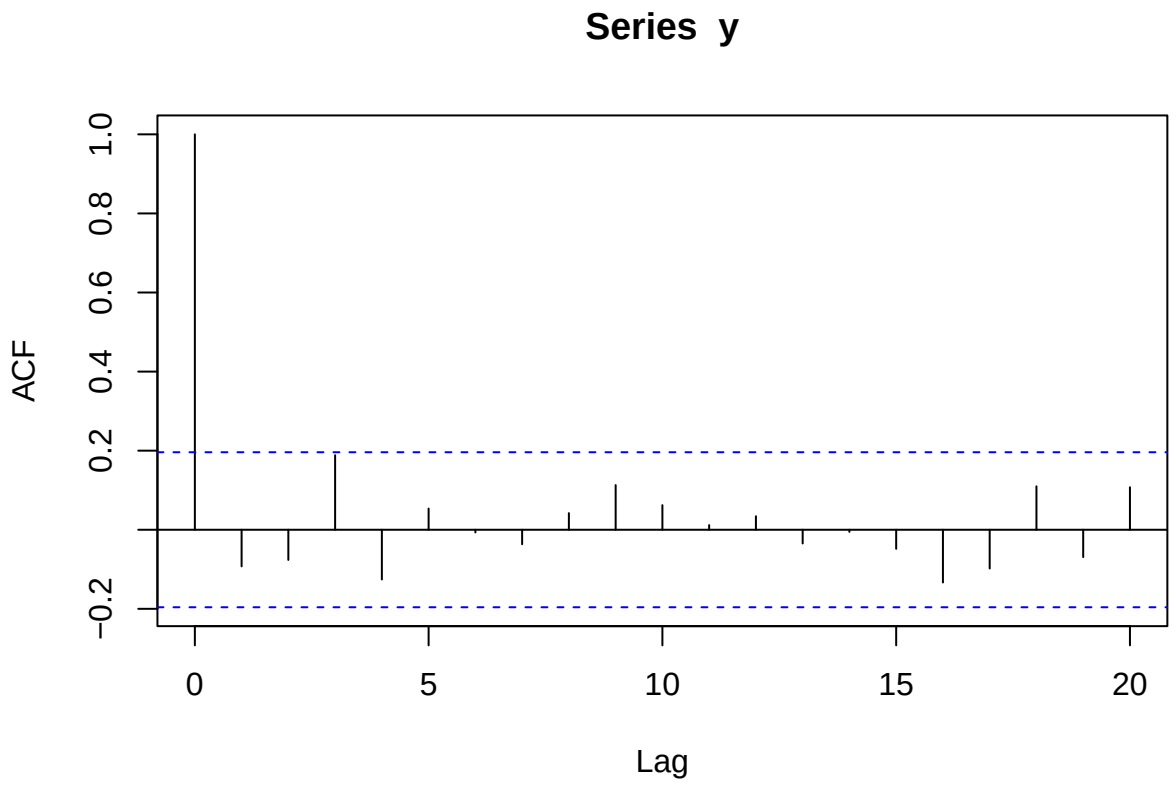
```
set.seed(30)

#Create a white-noise time series
y <- ts(rnorm(100))

library(ggfortify)
```

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

```
autoplot(y) + ggtitle("White noise of Standard Normal Distribution")
```

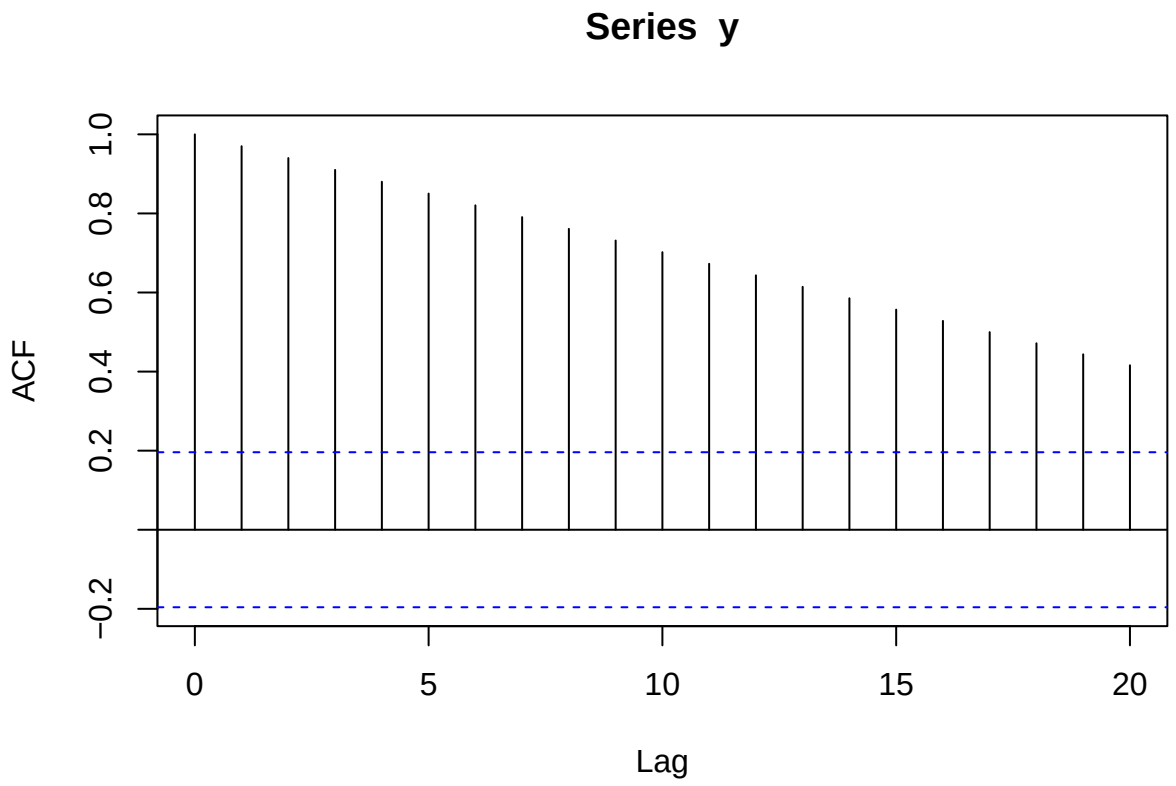



```
#plot its ACF
acf(y)
```



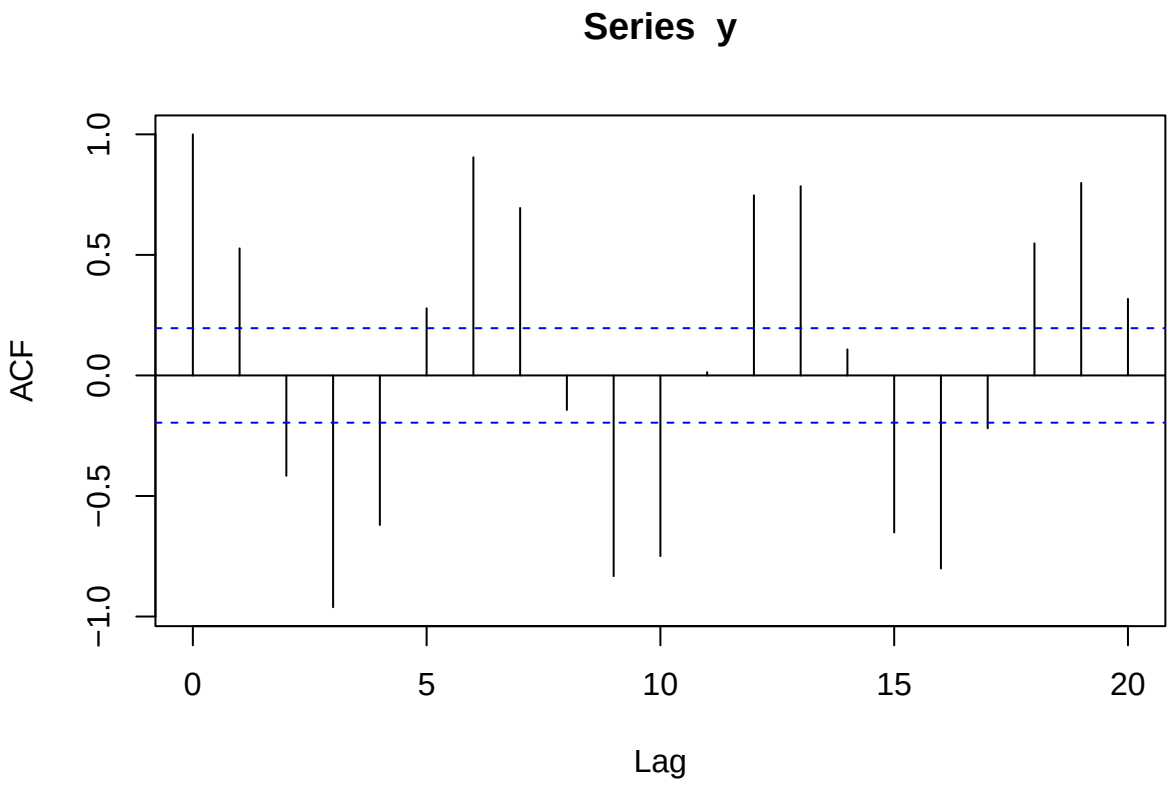
- a time series with trend

```
y = ts(c(1:100))
acf(y)
```



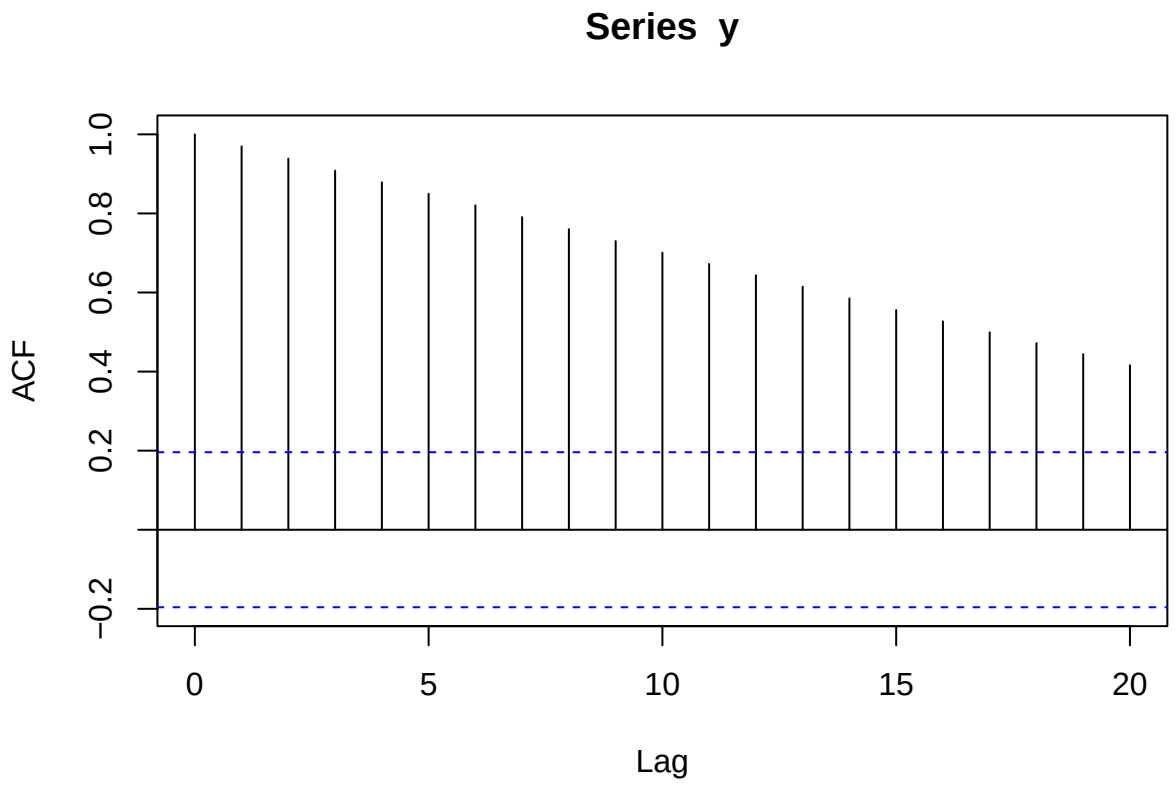
- a time series with seasonality

```
set.seed(30)
y = cos(1:100)
y <- ts(y)

acf(y)
```



- a time series with trend and seasonality

```
y = ts(cos(c(1:100))+c(1:100))
acf(y)
```



Comment on the difference between the ACF of the white-noise and other time series.

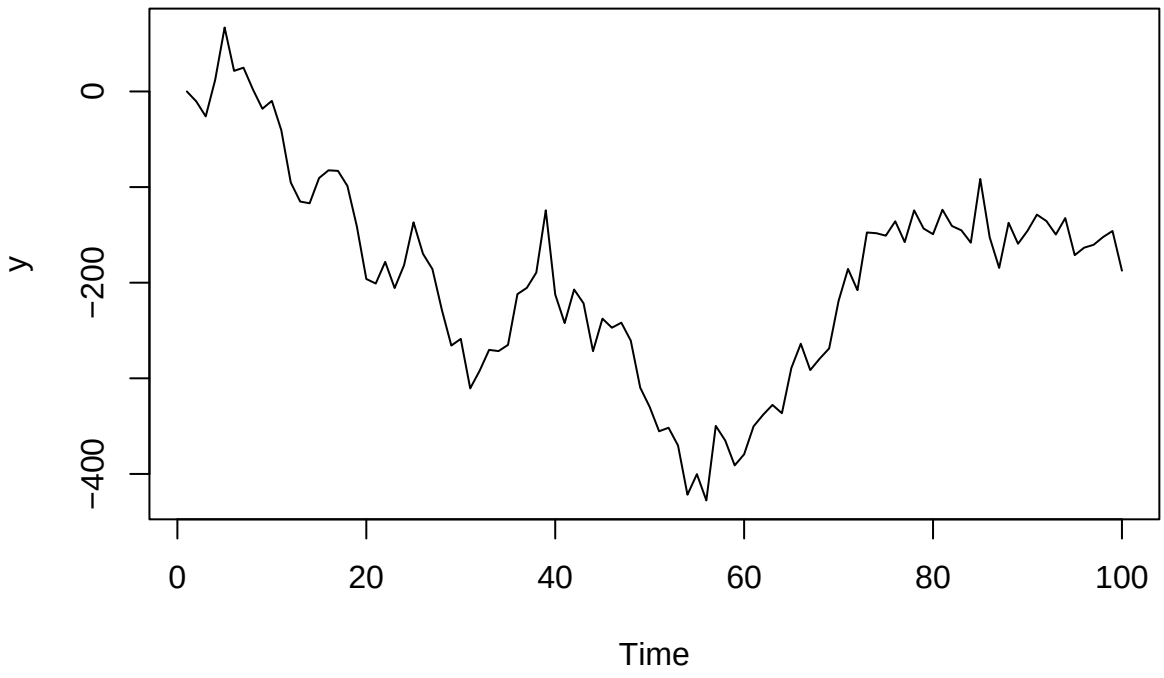
- The ACF of the white noise goes both positive and negative between -0.2 and 0.2, except for the first part that jumps to 1.0
- The ACF of the time series with trend shows a steady decline of the ACF from 1.0 to about 0.5, and there are no negative values
- The ACF of the time series with seasonality goes both positive and negative, but there is a pattern to it with positive values 1.0 and 0.5, then -0.5, -1.0, and -0.75, and then the plot returns to positive values of similar height.

2. Use the codes in the slides to simulate a different random walk Yt and:

- plot the random walk Yt

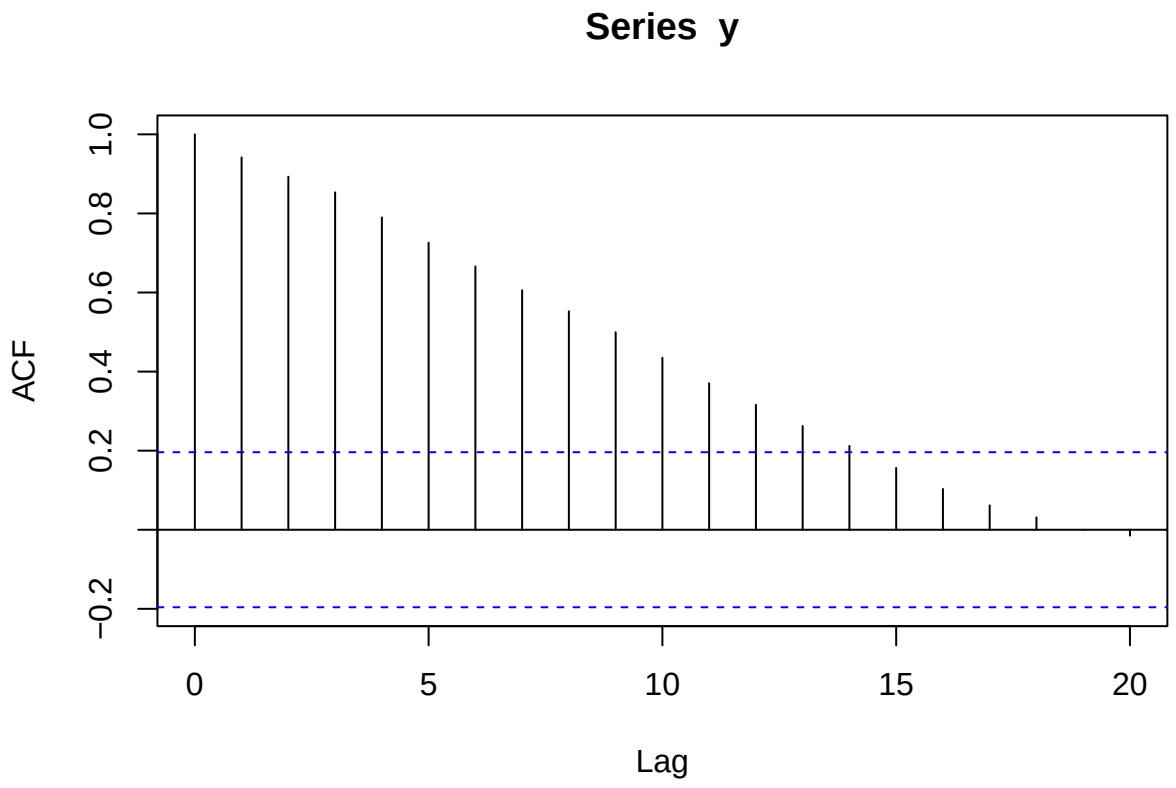
```
n = 100
error_mean = 0
c <- rnorm(n, mean = error_mean, sd = 30)
y_0 = 0
y = c(y_0, 2:n)
```

```r
for (i in 2:n)
{
  y[i] = y[i-1]+c[i]
}
y = ts(y)
plot(y)
```



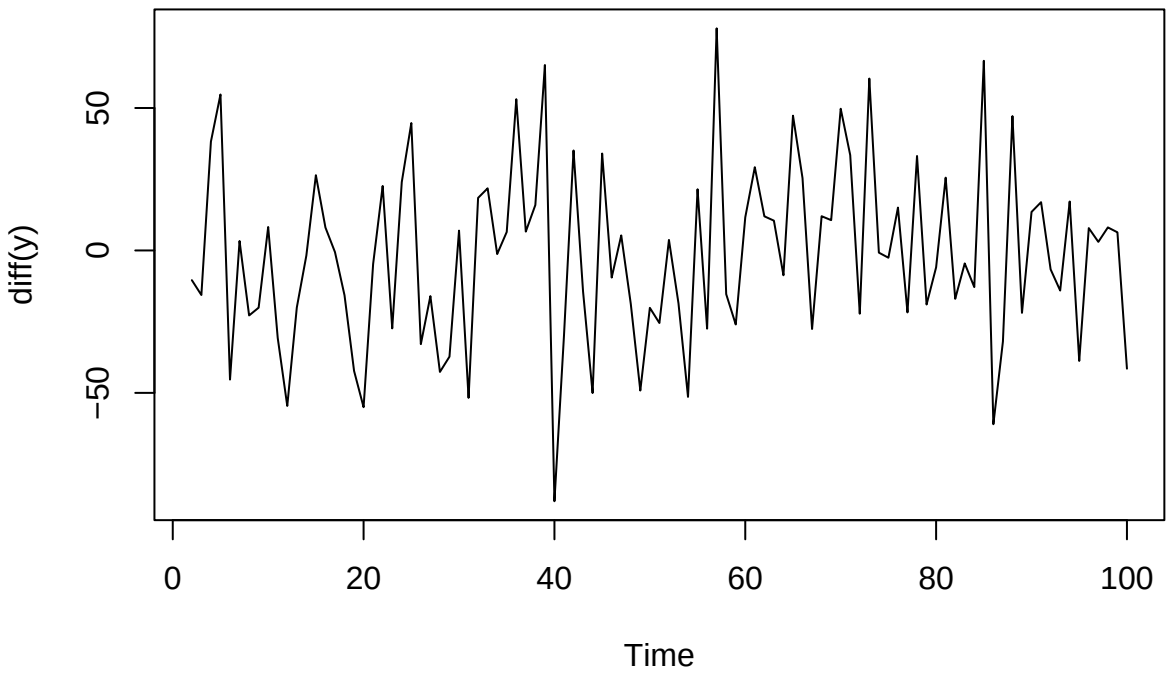
- plot the ACF of the random walk

```r
acf(y)
```



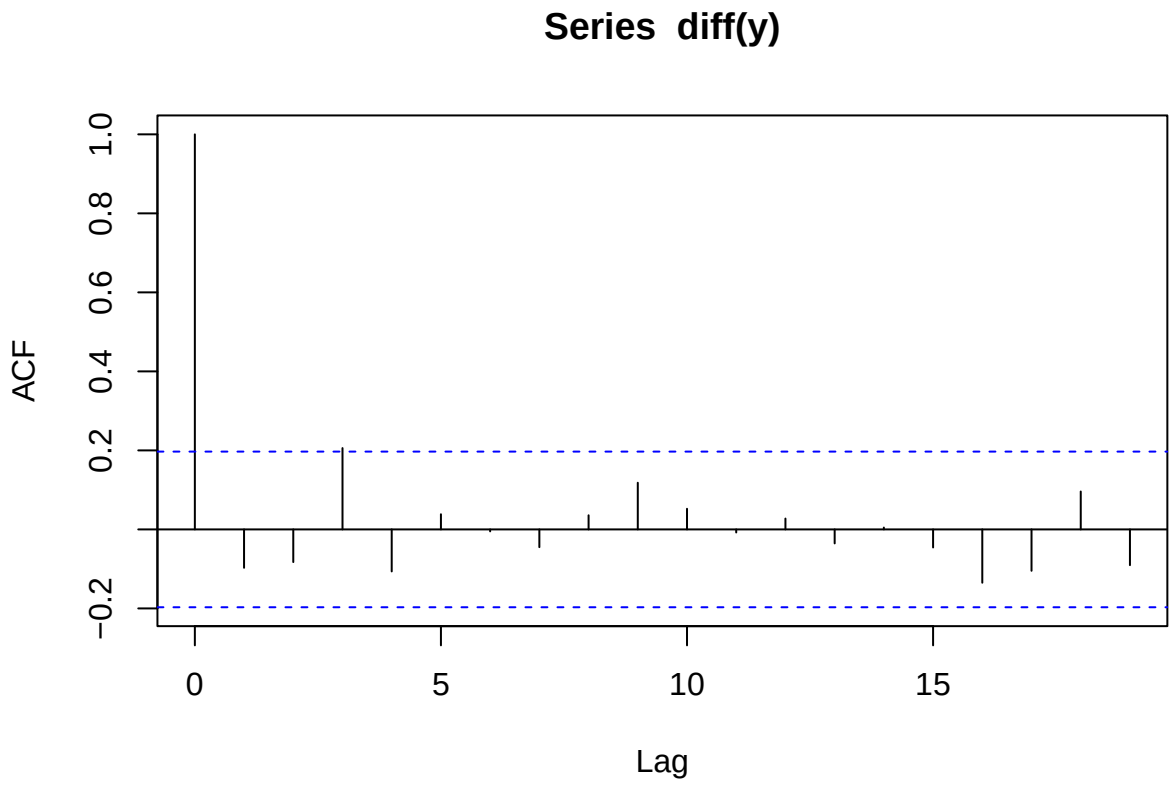
- plot the differencing of Yt

```
plot(diff(y))
```



- plot the ACF of the differencing of Yt

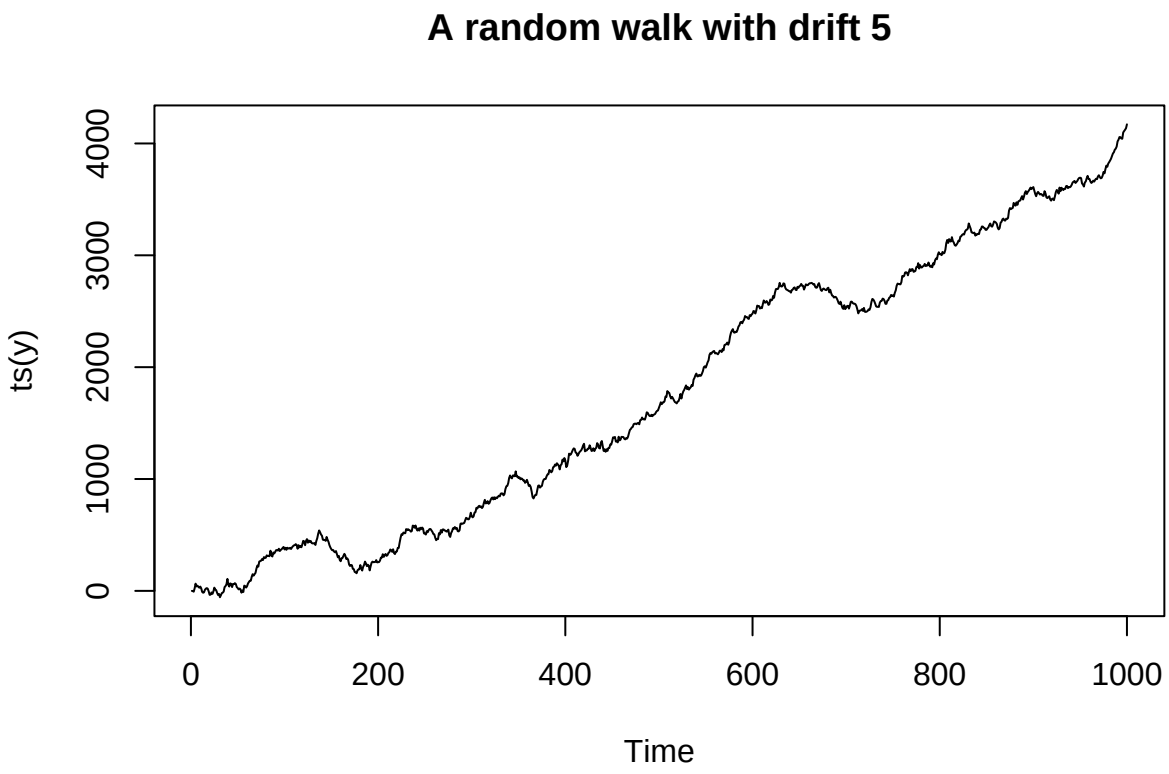
```
acf(diff(y))
```



3. Use the codes in the slides to simulate a different random walk Yt with drift 2024 and:

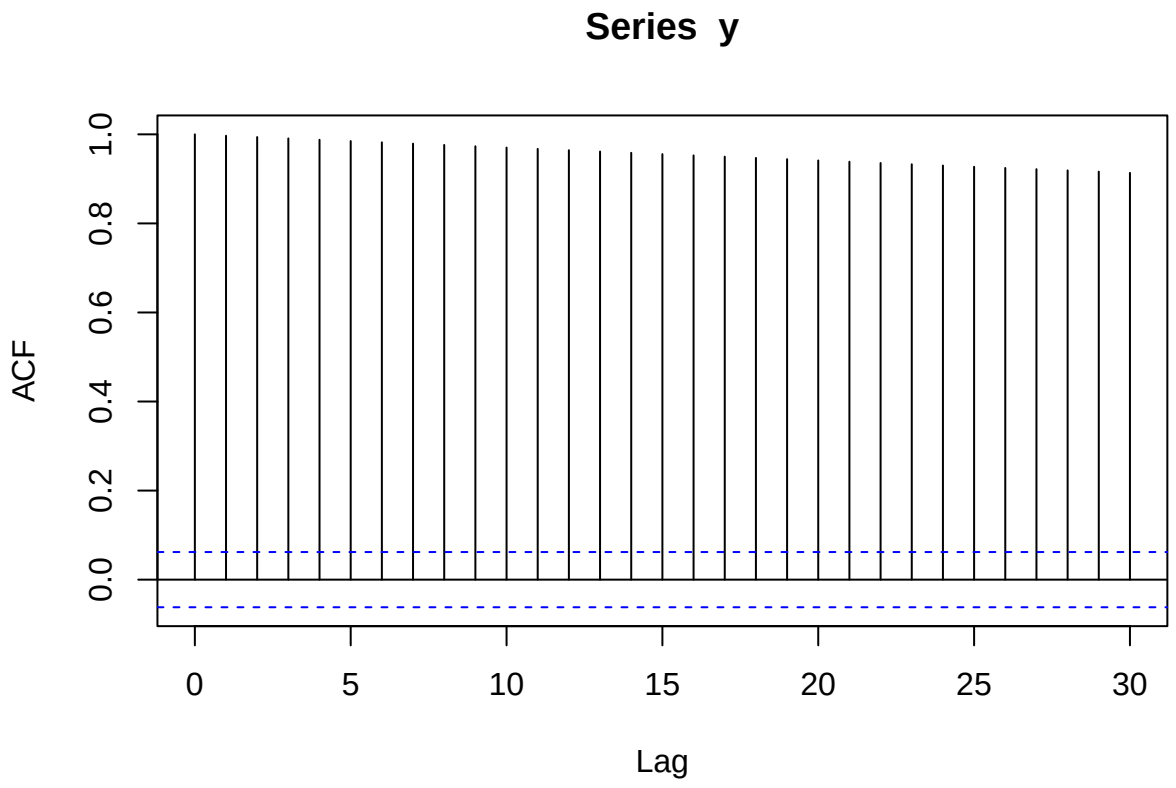
- plot the random walk Yt

```
set.seed(30)
n = 1000
c <- rnorm(n, sd = 20)
y_0 = 0
drift = 5
y = c(y_0, 2:n)
for (i in 2:n)
{
  y[i] = drift  + y[i-1]+c[i]
}

library(ggfortify)
library(latex2exp)

plot(ts(y))

title(paste0("A random walk with drift ", drift))
```



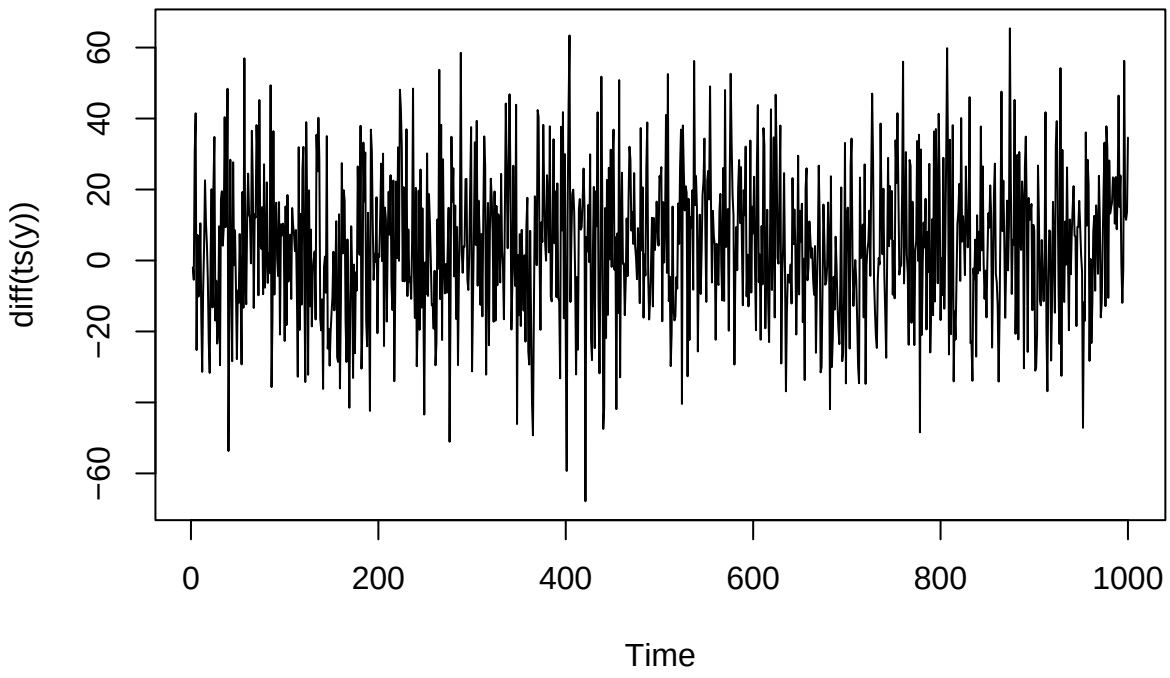
- plot the ACF of the random walk

```
acf(y)
```



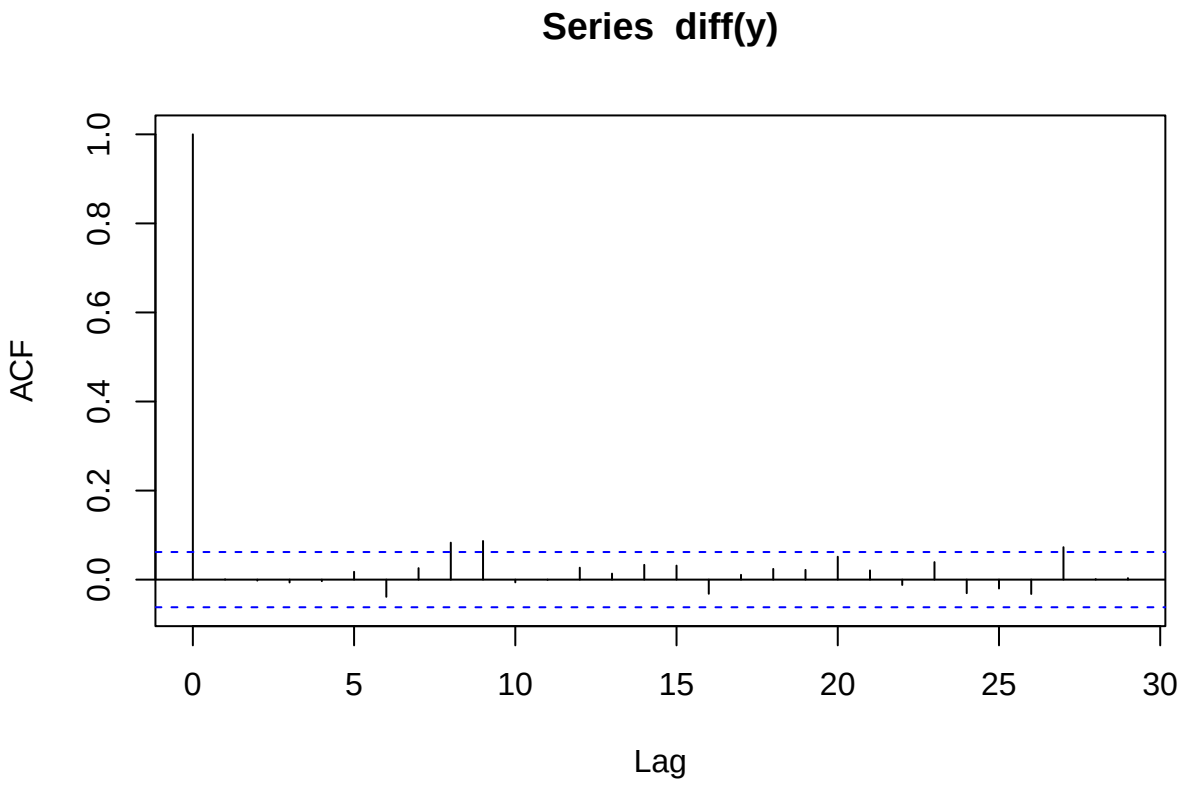
- plot the differencing of Yt

```
plot(diff(ts(y)))
```



- plot th ACF of the differencing of Yt

```
acf(diff(y))
```



4. Use the codes in the slides to plot the stock of Apple. Also:

- Plot the ACF of the stock

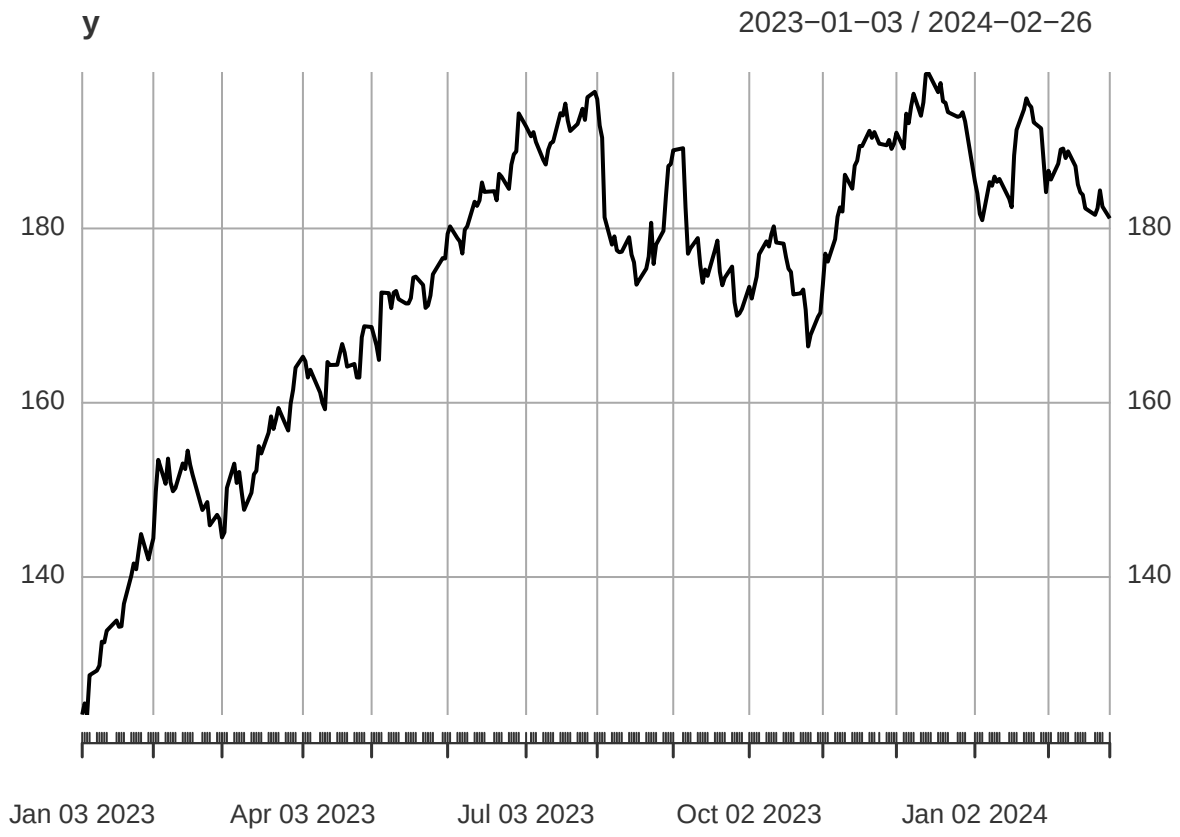
```
library(quantmod)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: TTR

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

```
getSymbols('AAPL', src='yahoo')
```

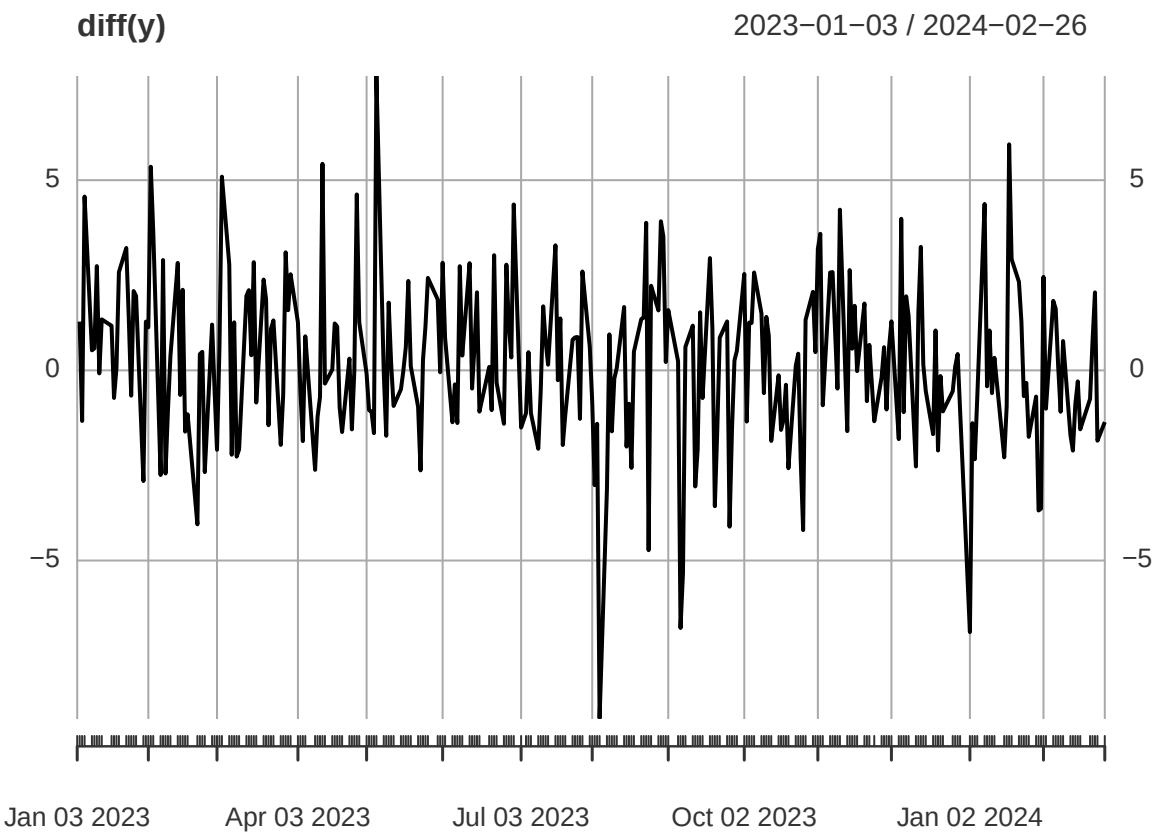
```
## [1] "AAPL"
```

```
y = Ad(AAPL[index(AAPL)>"2023-01-01",])
plot(y)
```



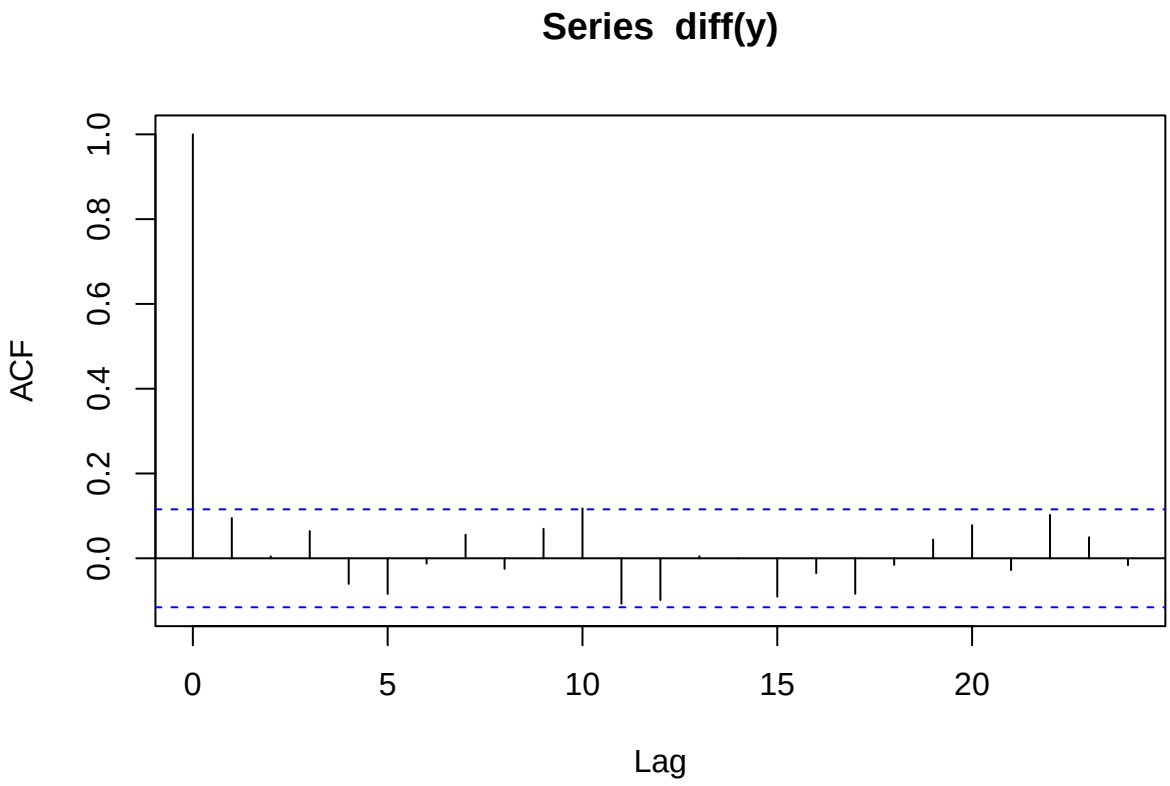
- Plot the differencing of the stock

```
plot(diff(y))
```



- Plot the ACF of the differencing of the stock

```
acf(diff(y), na.action = na.pass)
```



Is it reasonable to assume that Apple stock is following the random walk model?

- Yes, it is reasonable to assume that the stock follows the random walk model.
- It follows the same characteristics as the slides for the Microsoft stock example

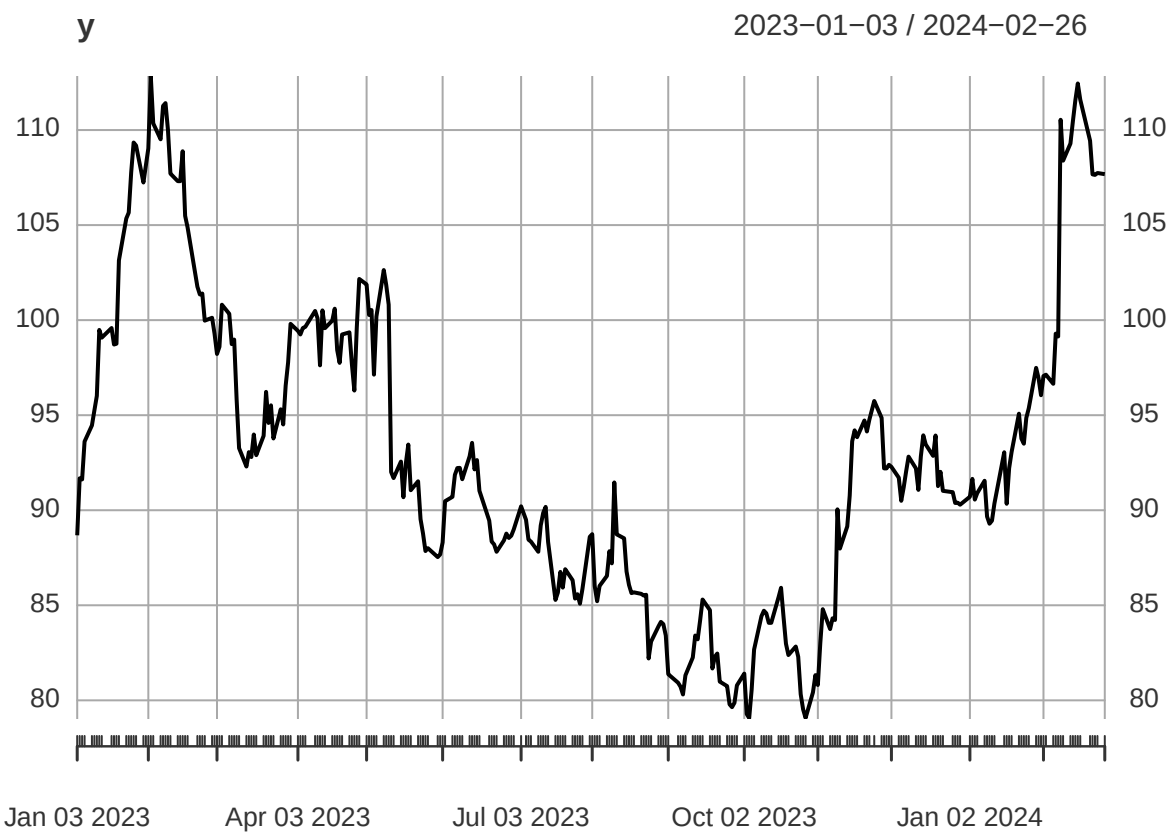
5. Find other stocks that may be following the random walk model.

- Disney (DIS):

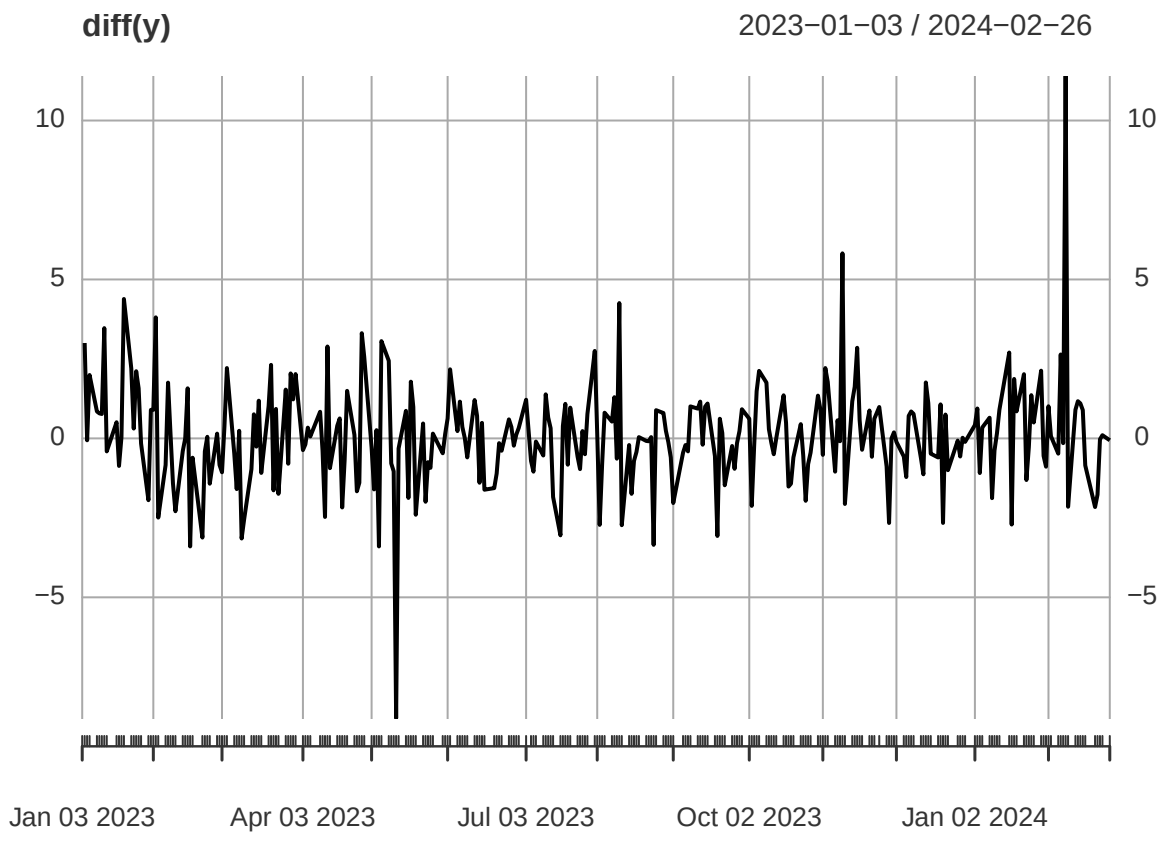
```
library(quantmod)

#Disney stock
getSymbols('DIS', src='yahoo')
```

```
## [1] "DIS"
```

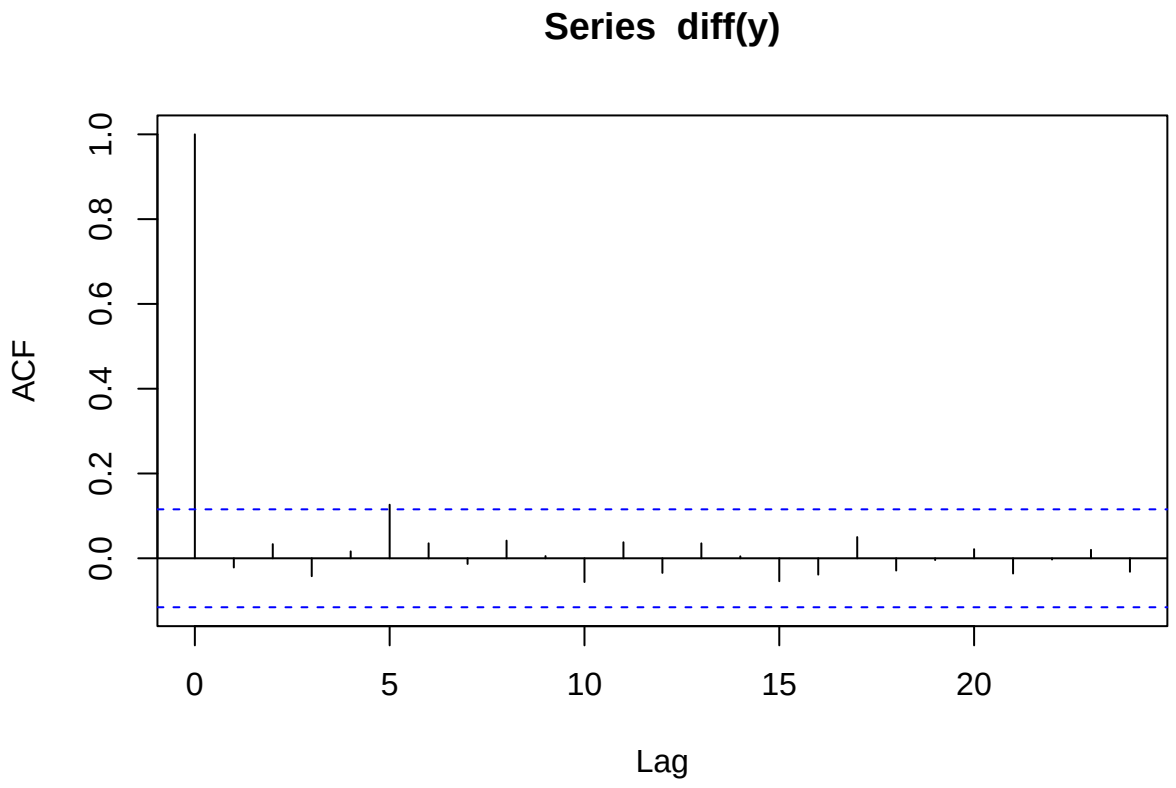
```
y = Ad(DIS[index(DIS)>"2023-01-01",])
plot(y)
```




```
#plot differencing
plot(diff(y))
```




```
#acf of differencing
acf(diff(y), na.action = na.pass)
```

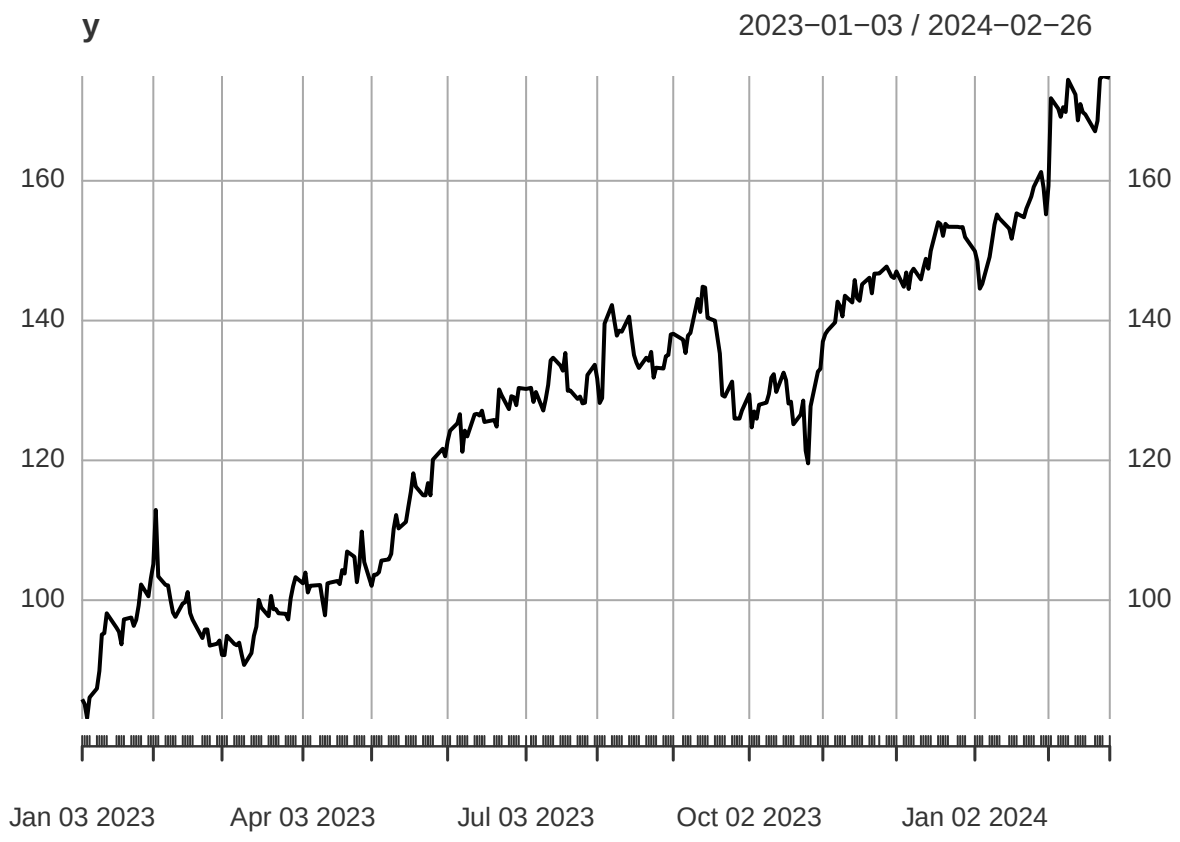


- Amazon (AMZN):

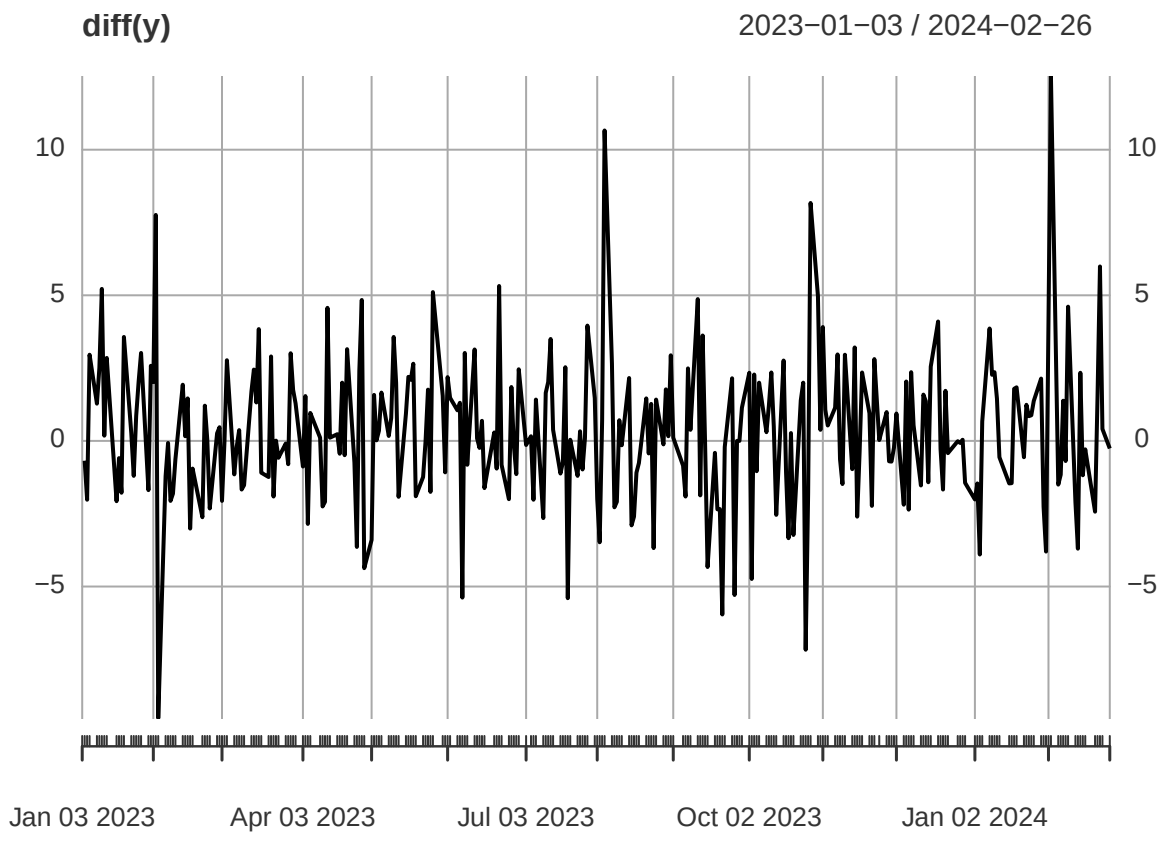
```
library(quantmod)

#Amazon stock
getSymbols('AMZN', src='yahoo')
```

```
## [1] "AMZN"
```

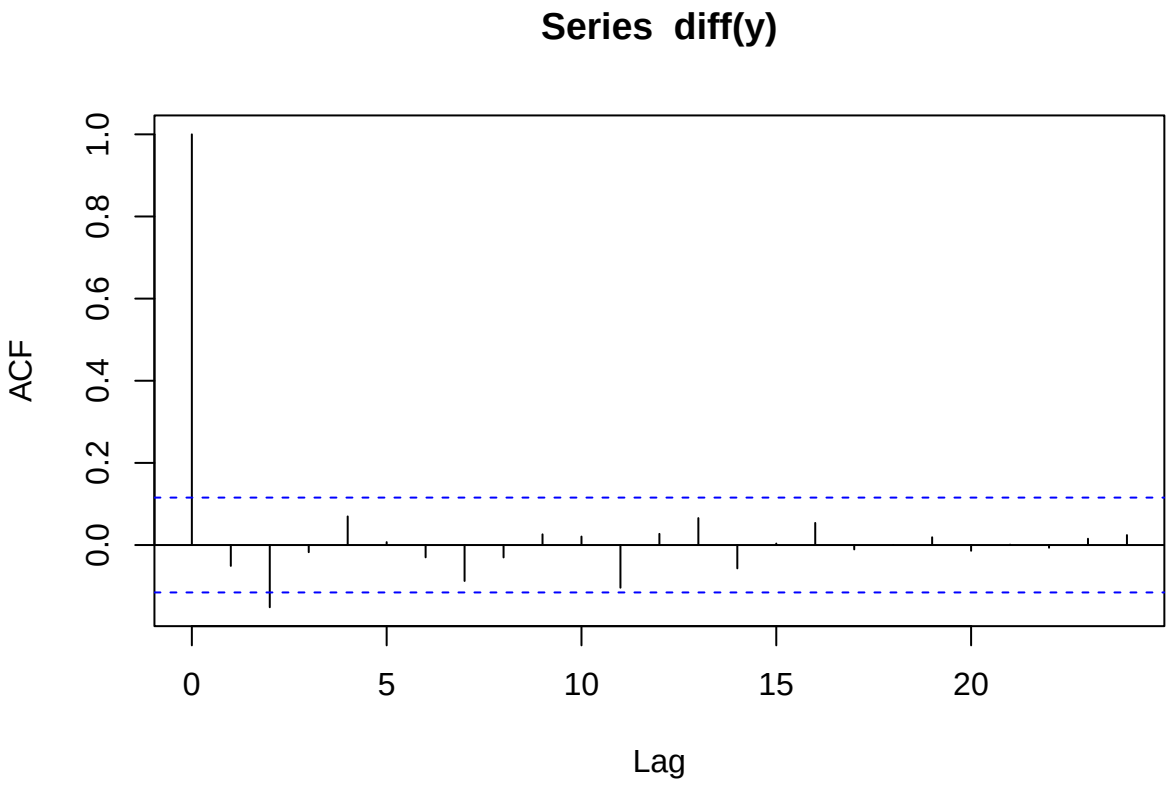
```
y = Ad(AMZN[index(AMZN)>"2023-01-01",])
plot(y)
```




```
#plot differencing
plot(diff(y))
```




```
#acf of differencing
acf(diff(y), na.action = na.pass)
```


Series diff(y)
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
Lag