# More on Time Series

jsn-jin @ UCLA

Feb 14, 2020

## Contents



# Monte Carlo Simulation: bivariate normal distribution

1. Let $X$ and $Y$ be distributed bivariate normal with

$$\mu_X = 0.01, \quad \mu_Y = 0.05, \quad \sigma_X = 0.5, \quad \sigma_Y = 0.3.$$

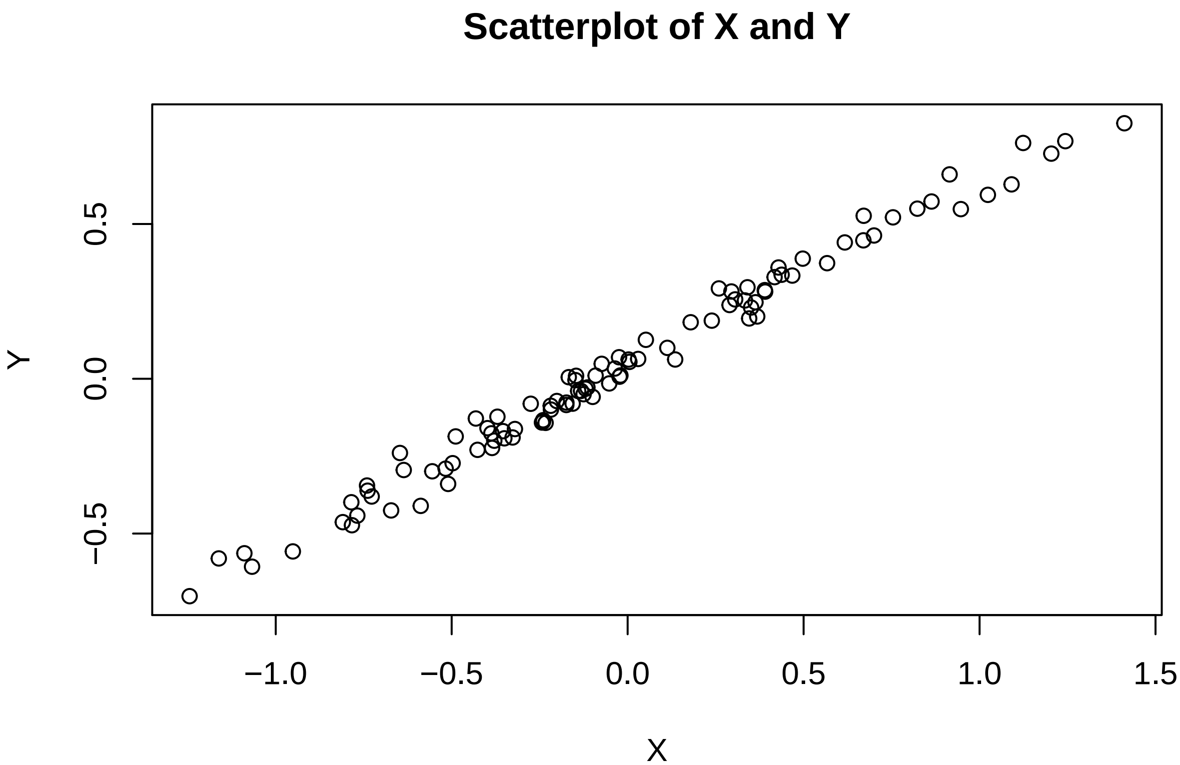
(a) Using R package function `rmvnorm()`, simulate 100 observations from the bivariate distribution with $\rho_{XY} = 0.99$.

```
library(mvtnorm)

# Create the covariance matrix:
sigma <- matrix(c(0.5^2, 0.99*0.5*0.3, 0.99*0.5*0.3, 0.3^2), ncol = 2)

mu <- c(0.01, 0.05)

data <- rmvnorm(n = 100, mean = mu, sigma = sigma)
head(data)
```

```
##              [,1]         [,2]
## [1,] -1.06729986 -0.607099775
## [2,]  0.35088294  0.230144205
## [3,]  0.02990322  0.064002663
## [4,]  0.42878244  0.359482614
## [5,]  0.28952791  0.238273734
## [6,] -0.02306669  0.006832172
```

(b) Using the `plot()` function create a scatterplot of the observations.

```
plot(data[,1], data[,2], main = "Scatterplot of X and Y", xlab = "X", ylab = "Y")
```




(c) Using the function `pmvnorm()`, compute the joint probability $P(X \leq 0, Y \leq 0)$.

```
P <- pmvnorm(mean=mu, sigma=sigma, lower=-Inf, upper=c(0,0))
sprintf("The probability is %#.3f.", P)
```

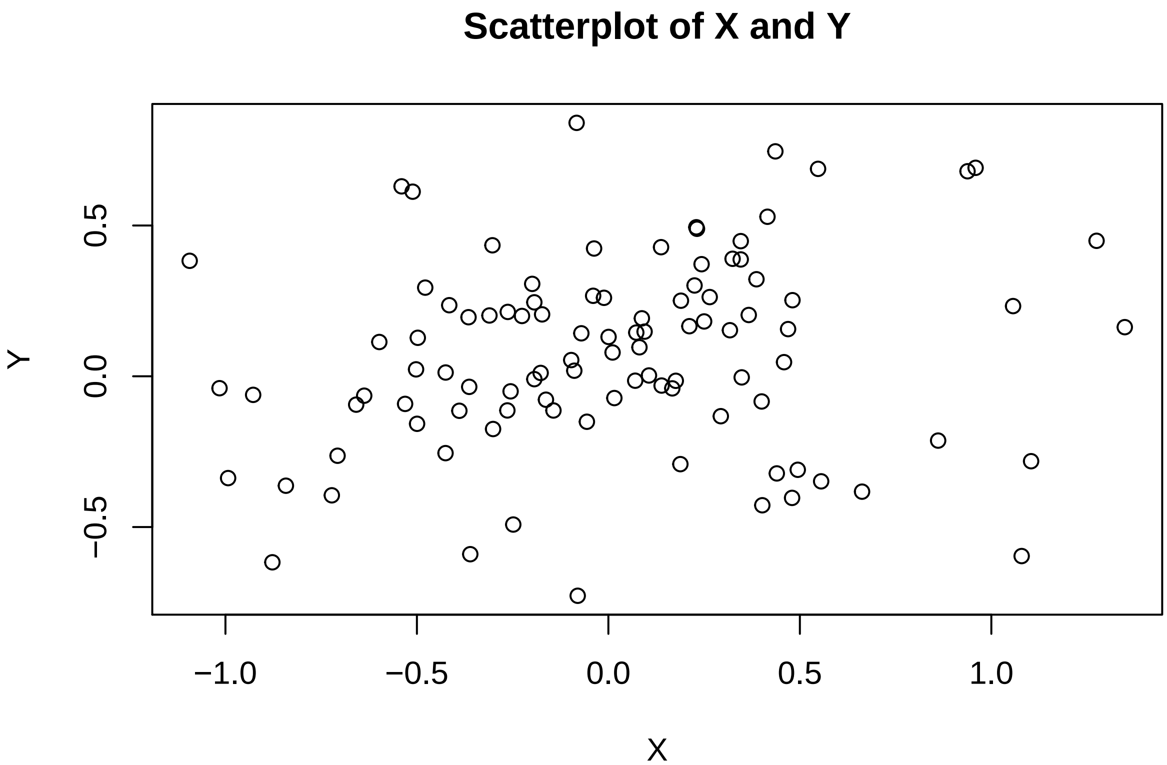
```
## [1] "The probability is 0.429."
```

(d) Do the same exercise with $\rho_{XY} = 0$ and $\mu_X = 0, \mu_Y = 0$, i.e., $X$ and $Y$ are independent.

```
sigma <- matrix(c(0.5^2, 0, 0, 0.3^2), ncol = 2)

mu <- c(0, 0)

data <- rmvnorm(n = 100, mean = mu, sigma = sigma)
head(data)
```

```
##             [,1]       [,2]
## [1,] -0.03736123  0.4236814
## [2,]  0.41545232  0.5290128
## [3,]  1.34851066  0.1628718
## [4,] -0.49771980  0.1276064
## [5,]  0.47968159 -0.4033169
## [6,]  0.43601195  0.7457851
```

```r
plot(data[,1], data[,2], main="Scatterplot of X and Y", xlab = "X", ylab = "Y")
```




```r
P <- pmvnorm(mean = mu, sigma = sigma, lower = -Inf, upper = c(0,0))
sprintf("The probability is %#.3f.", P)
```

```
## [1] "The probability is 0.250."
```

## ARMA Processes

The ARMA model can be understood as a combination of both the AR model and the MA model. Consider the ARMA (1,1) model:

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t + \theta\epsilon_{t-1} \quad \text{with } |\phi| < 1,$$

where

$$\epsilon_t \sim MDS(0, \sigma_\epsilon^2).$$

If $\theta = 0$, then the ARMA (1,1) model is just the AR(1) model.

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t \quad \text{with } |\phi| < 1,$$

If $\phi = 0$, then the ARMA (1,1) model is just the MA(1) model.

$$Y_t - \mu = \epsilon_t + \theta\epsilon_{t-1} \quad \text{with } |\phi| < 1,$$

Let's consider a specific ARMA model.

$$Y_t = 0.5Y_{t-1} + \epsilon_t - 0.5\epsilon_{t-1},$$

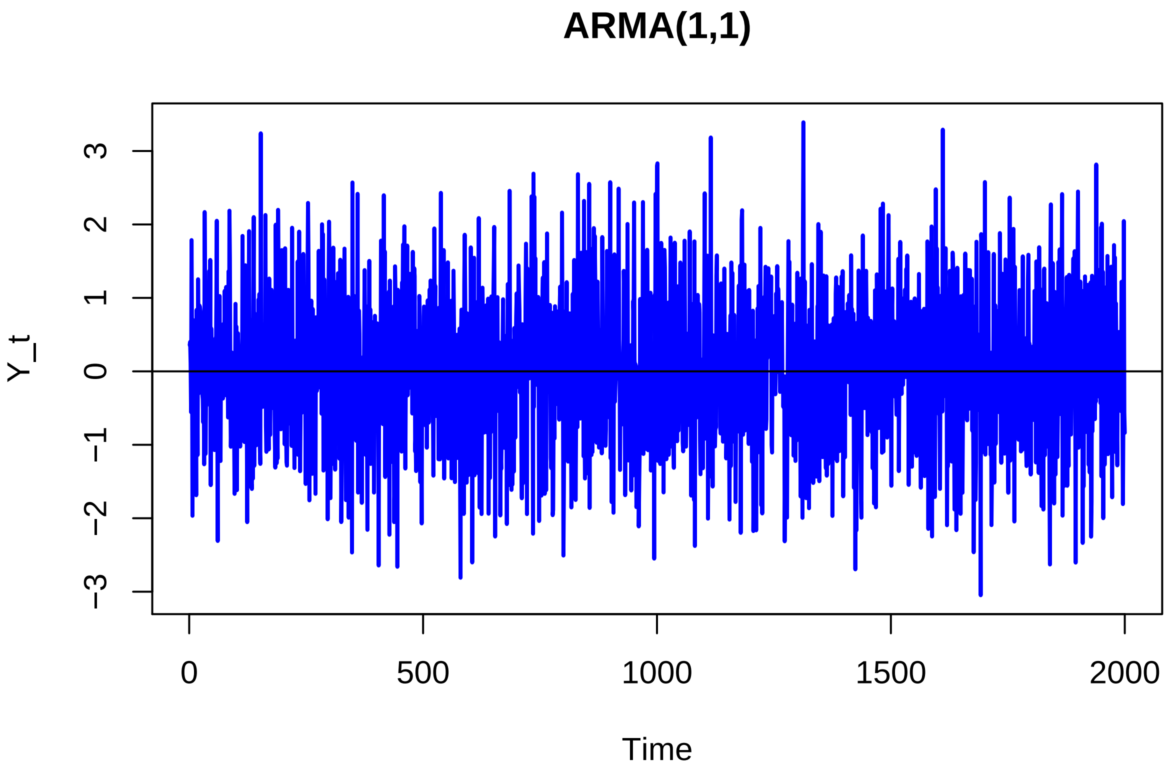
and

$$\epsilon_t \sim IID(0, 1).$$

### Simulation

Let's first run some simulations.

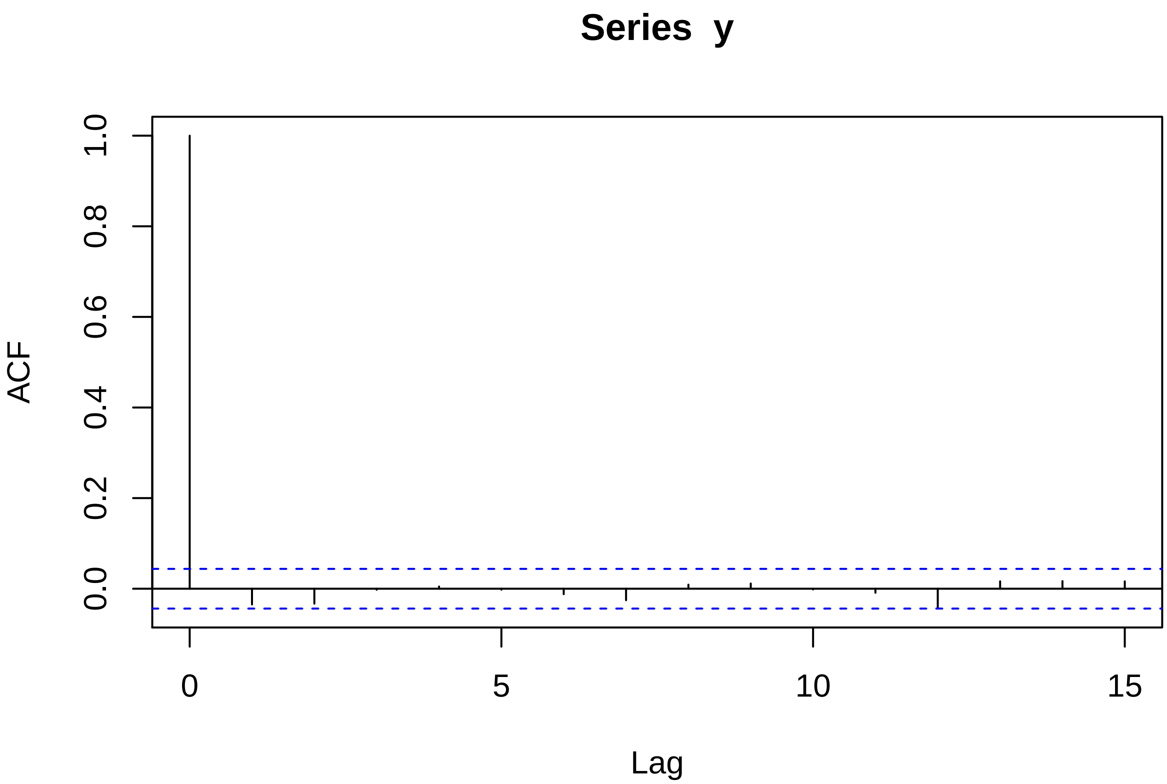
```
set.seed(123)

mu <- 0
phi <- 0.5
theta <- -0.5
sigma.e <- 1
error.model <- function(n){rnorm(n, sd = sigma.e)}
n.obs <- 2000

# ARMA is a special case of ARIMA model such that the middle order = 0
y <- arima.sim(model = list(order = c(1, 0, 1), ar = phi, ma = theta), n = n.obs, rand.gen = error.mode

ts.plot(y,
        main = "ARMA(1,1)",
        xlab = "Time",
        ylab = "Y_t",
        col = "blue",
        lwd = 2)

abline(h = mu)
```




### ACF

Next, we plot the sample ACF.

```
acf(y, lag.max = 15)
```




```r
cat("The sample ACFs are:")
```

```
## The sample ACFs are:
```

```r
acf(y, plot = FALSE)[1:10,]
```

```
##
## Autocorrelations of series 'y', by lag
##
##      1      2      3      4      5      6      7      8      9     10
## -0.035 -0.033 -0.002  0.005 -0.002 -0.012 -0.025  0.009  0.012 -0.002
```

### Estimation

We can fit an ARMA(1,1) model to the simulated data to get estimates $(\hat{\phi}, \hat{\theta}, \hat{\mu}, \hat{\sigma}^2)$.

```r
library(forecast)
```

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

```r
AR <- Arima(y, order = c(1, 0, 1), include.mean = TRUE)

AR
```

```
## Series: y
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1    mean
##       0.5769  -0.6152  0.0274
```

```
## s.e.   0.3861   0.3732  0.0203
##
## sigma^2 estimated as 1.001:  log likelihood=-2837.23
## AIC=5682.46   AICc=5682.48   BIC=5704.87
```

# Analyzing Time Series Data: An Introduction

Let's use the stock price data of the Goldman Sachs Group, Inc. to illustrate some useful methods for data analysis. Today, we will discuss how to use histogram, kernel density, ACF plot, box plot and Q-Q plot to analyze financial data.

## Load the Data Set

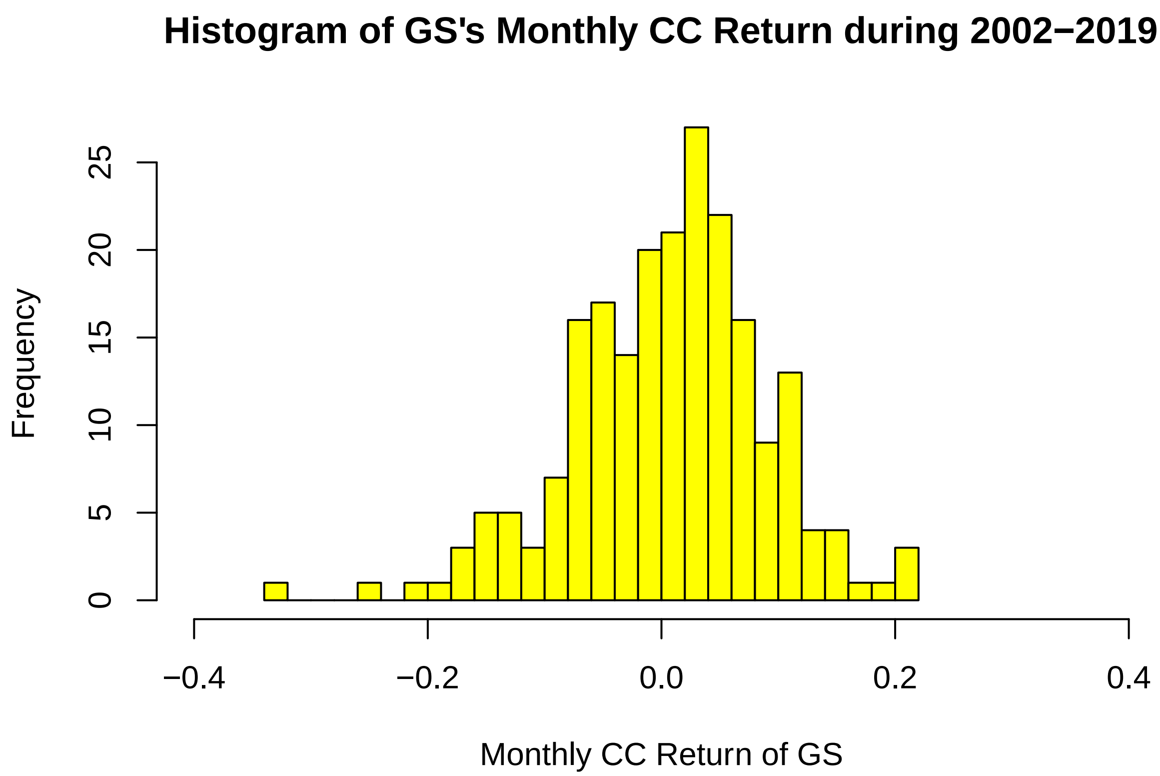
```r
rm(list = ls())
df <- read.csv(file = "GS.csv", head = TRUE, sep = ",", stringsAsFactors = FALSE)
```

## Calculate Monthly CC Return

```r
df$Date <- as.Date(df$Date, "%Y-%m-%d")
N <- length(df$Adj.Close)
cc.ret <- log(df$Adj.Close[2:N]) - log(df$Adj.Close[1:(N-1)])
df.ccret <- data.frame(df$Date[2:N], cc.ret)
colnames(df.ccret)[1] <- "Date"
```

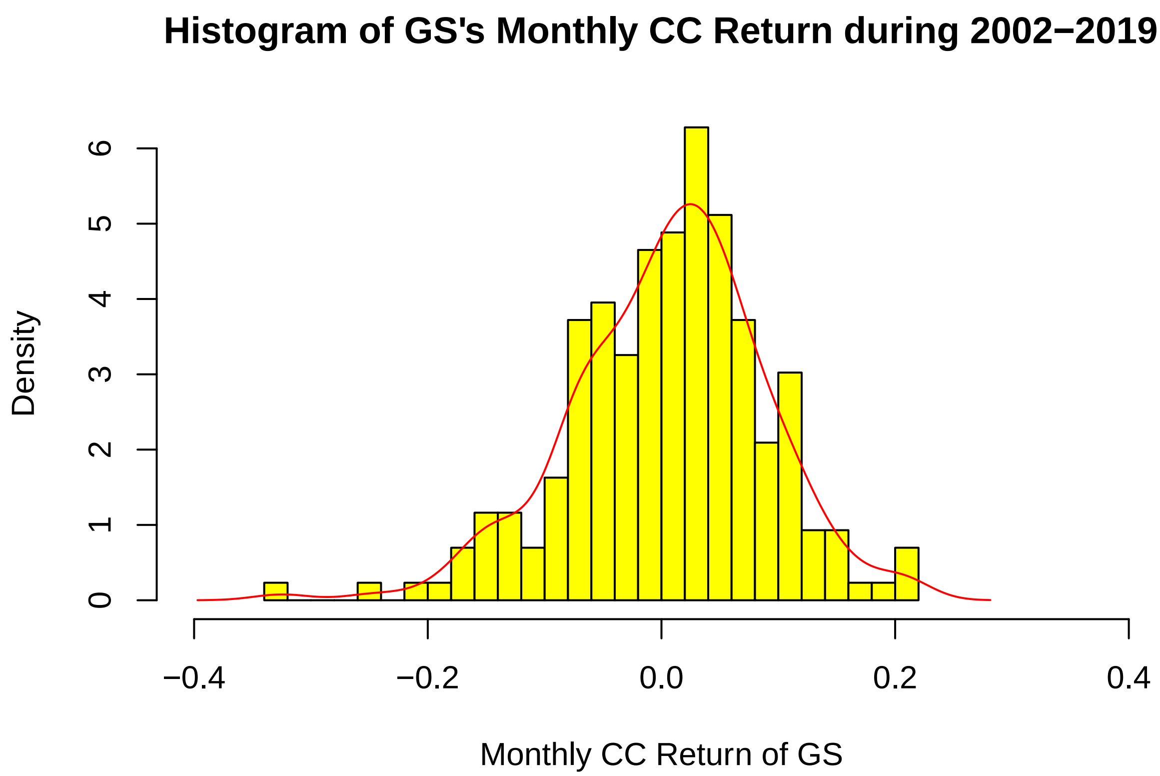
## Histogram

```r
hist(df.ccret$cc.ret,
     breaks = 20,
     xlim = c(-0.4, 0.4),
     xlab = "Monthly CC Return of GS",
     main = "Histogram of GS's Monthly CC Return during 2002-2019",
     col = "yellow")
```



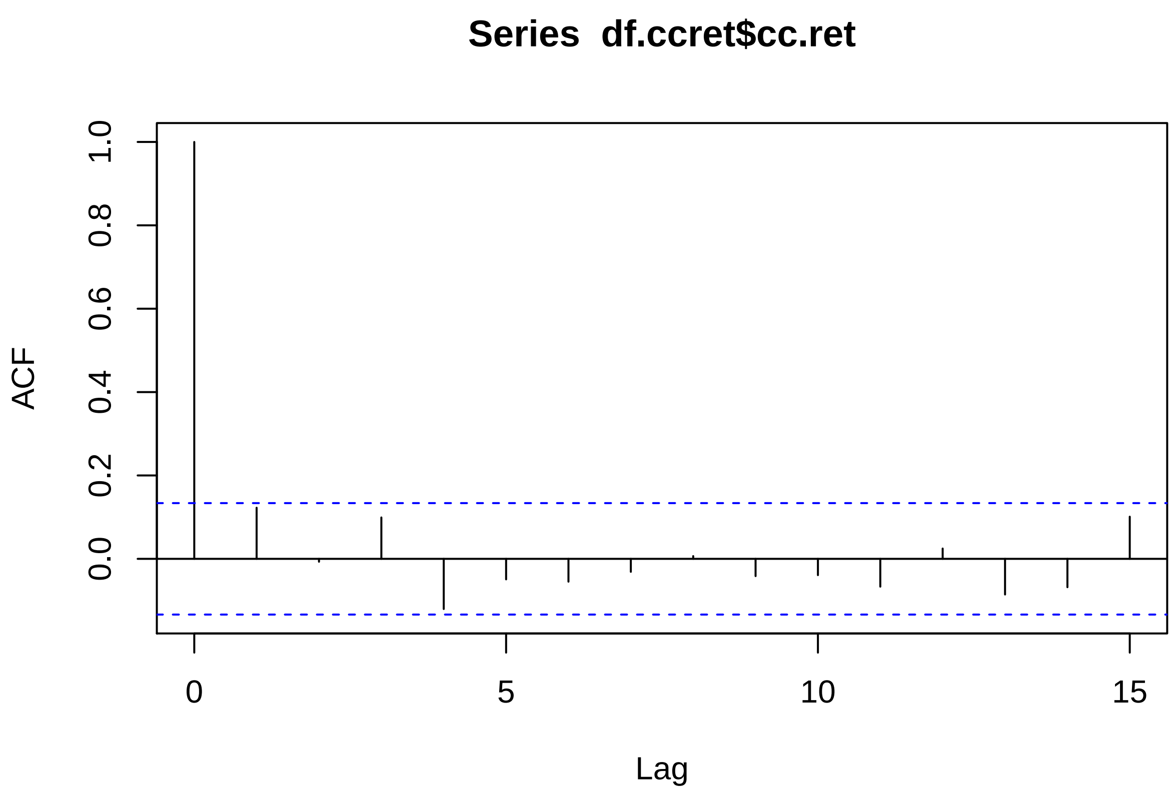

### Histogram with Kernel Density

```
hist(df.ccret$cc.ret,
     probability = TRUE,
     breaks = 20,
     xlim = c(-0.4, 0.4),
     xlab = "Monthly CC Return of GS",
     main = "Histogram of GS's Monthly CC Return during 2002-2019",
     col = "yellow")

lines(density(df.ccret$cc.ret), col = "red")
```




### ACF

```r
acf(df.ccret$cc.ret,lag.max = 15)
```



```r
cat("The sample ACFs are:")
```

```
## The sample ACFs are:
```

```r
acf(df.ccret$cc.ret, plot = FALSE)[1:10,]
```

```
##
## Autocorrelations of series 'df.ccret$cc.ret', by lag
##
##      1      2      3      4      5      6      7      8      9     10
##  0.123 -0.007  0.099 -0.120 -0.049 -0.055 -0.031  0.007 -0.041 -0.039
```

### Box Plot

A boxplot is a standardized way of displaying the dataset based on a five-number summary: the minimum, the maximum, the sample median, and the first and third quartiles.

Minimum: the lowest data point *excluding any outliers*.

Maximum: the largest data point *excluding any outliers*.

Median (Q2 / 50th Percentile) : the middle value of the dataset.

First quartile (Q1 / 25th Percentile) : is also known as the lower quartile $q_n(0.25)$ and is the middle value between the smallest number (not the minimum) and the median of the dataset.

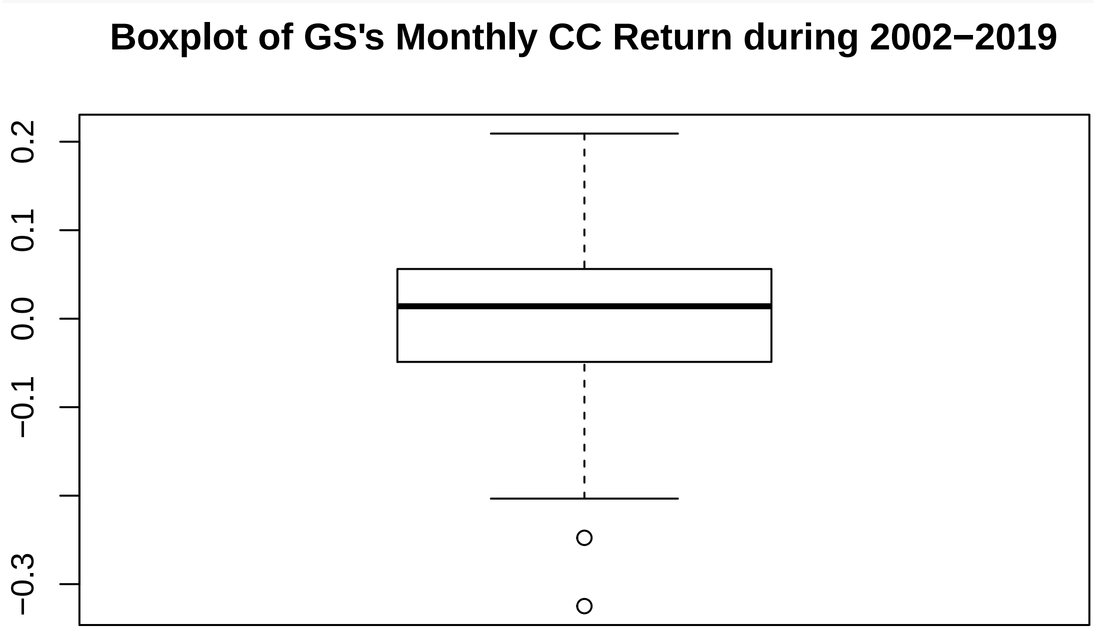
Third quartile (Q3 / 75th Percentile) : is also known as the upper quartile $q_n(0.75)$ and is the middle value between the largest number (not the maximum) and the median of the dataset.

An important element used to construct the box plot by determining the minimum and maximum data values feasible, but is not part of the aforementioned five-number summary, is the interquartile range or IQR denoted below:

Interquartile Range (IQR) : is the distance between the upper and lower quartile.

$$IQR = Q_3 - Q_1 = q_n(0.75) - q_n(0.25).$$

```r
boxplot(df.ccret$cc.ret, main = "Boxplot of GS's Monthly CC Return during 2002-2019")
```



### Normal Q-Q Plot

The points plotted in a Q–Q plot are always non-decreasing when viewed from left to right. If the two distributions being compared are identical, the Q–Q plot follows the 45-degree line $y = x$. If the two distributions agree after linearly transforming the values in one of the distributions, then the Q–Q plot follows some line, but not necessarily the line $y = x$. If the general trend of the Q–Q plot is flatter than the line $y = x$, the distribution plotted on the horizontal axis is more dispersed than the distribution plotted on the vertical axis. Conversely, if the general trend of the Q–Q plot is steeper than the line $y = x$, the distribution plotted on the vertical axis is more dispersed than the distribution plotted on the horizontal axis. Q–Q plots are often arced, or "S" shaped, indicating that one of the distributions is more skewed than the other, or that one of the distributions has heavier tails than the other.

```r
qqnorm(df.ccret$cc.ret, pch = 1, frame = FALSE)
qqline(df.ccret$cc.ret, col = "steelblue", lwd = 2)
```