demo

```r
library(tidyverse)
library(forecast)
```

Data Preprocessing

```r
# Load the historical data of your stock
stock <- read_csv("data/RBL.AX.csv")
select <- dplyr::select
```

```r
# We only need date and adj.close price
stock <- select(stock, Date, price = `Adj Close`) %>%
  mutate(price = as.numeric(price)) %>%
  filter(!is.na(price))

# Compute log return (log(today) - log(yesterday))
stock <- mutate(stock, log_r = log(price) - lag(log(price)))
```

```r
# Check whether there are AR structure in the log return
# ARIMA(0,0,0) implies there is no ARIMA structure
auto.arima(stock$log_r) %>%
  summary()
## Series: stock$log_r
## ARIMA(0,0,0) with zero mean
##
## sigma^2 estimated as 0.002121:  log likelihood=1710.39
## AIC=-3418.77   AICc=-3418.77   BIC=-3413.83
##
## Training set error measures:
##                       ME      RMSE        MAE MPE MAPE      MASE        ACF1
## Training set 0.001965211 0.046056 0.02853715 NaN  Inf 0.6717744 0.004592055
```

```r
# We use the empircal mean to estimate the expected log return
# Positive estimated mean of log return means, in expectation, you can make profit
# from this stock
est_mean <- mean(stock$log_r, na.rm = T)
est_mean
## [1] 0.001965211
```

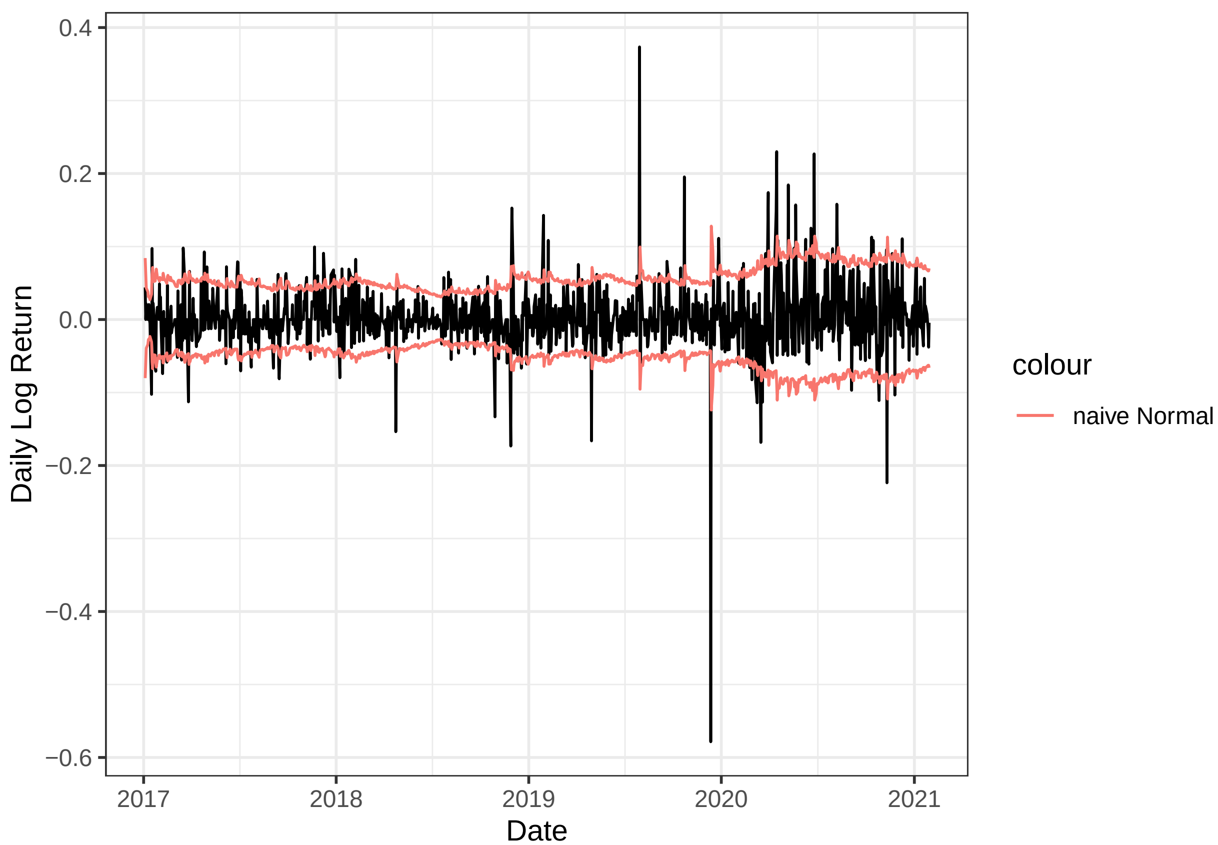
```r
# Compute the daily square root of residuals
stock <- mutate(stock, risk = sqrt((log_r - est_mean)^2))
```

Naive model - Normal

```r
# Using auto.arima() to fit the data
mod1 <- auto.arima(stock$risk)

# Assume the log return follows a normal distribution
```

```r
stock <- mutate(stock,
                fitted = c(NA, mod1$fitted),
                upper = c(NA, mod1$fitted * 1.96 + est_mean),
                lower = c(NA, mod1$fitted * -1.96 + est_mean))
```

```r
# Plot our predicted 95% CI against the real log return
# Most of the time the black line should fall between the 95% CI
ggplot(stock) +
  geom_line(aes(Date, log_r)) +
  geom_line(aes(Date, upper, col = "naive Normal")) +
  geom_line(aes(Date, lower, col = "naive Normal")) +
  theme_bw() +
  ylab("Daily Log Return")
```



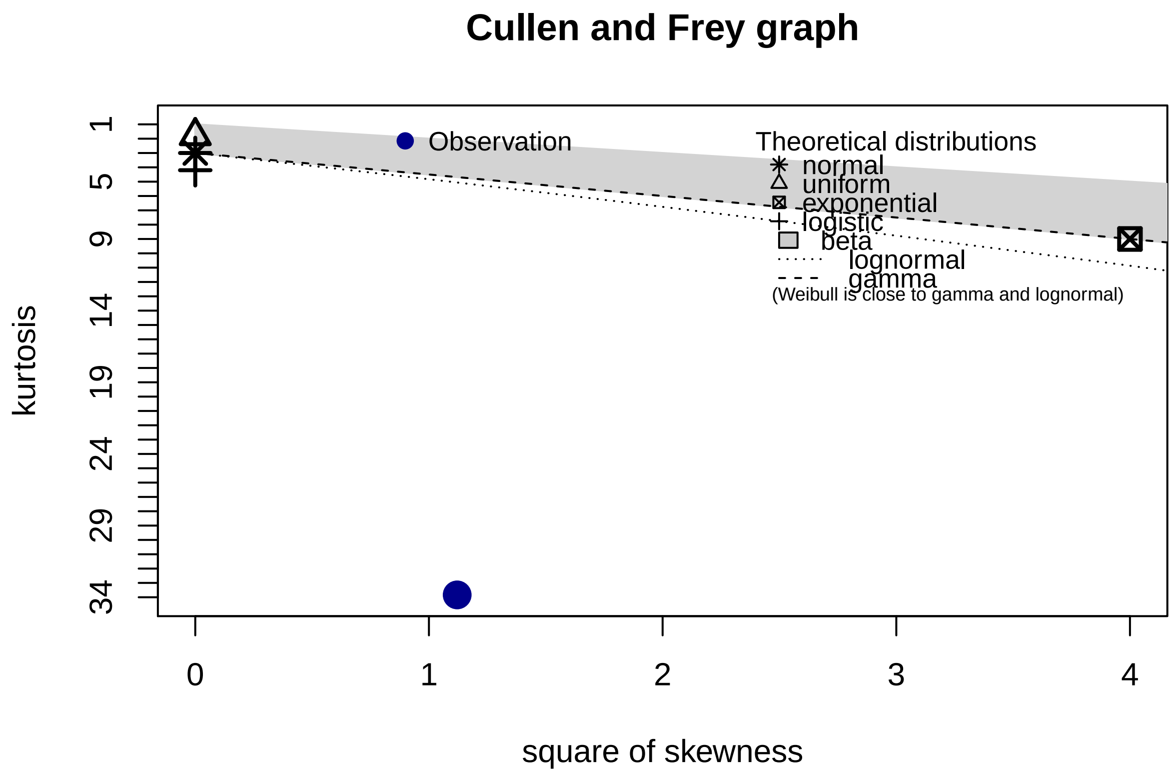

```r
# How many percent of the time the log return will exceed the 95% CI
# If it is greater than 5%, it means the assumption of the distribution is wrong or
# the prediction of the risk is inaccurate
mean(stock$log_r > stock$upper | stock$log_r < stock$lower, na.rm = T)
## [1] 0.1309408
```

```r
# Predict the next day log return
point_forecast <- forecast(mod1, 1) %>%
  .[["mean"]]

# Chance of suffer a loss
paste("Chance of suffer a loss:",
```

```r
      round(pnorm(0, mean = est_mean, sd = point_forecast)*100, 3), "%")
## [1] "Chance of suffer a loss: 47.629 %"

# 95% CI of log return
up95 <- point_forecast * 1.96 + est_mean
lo95 <- point_forecast * -1.96 + est_mean
paste("Log return 95% CI: (", round(lo95, 3), ",", round(up95 , 3), ")")
## [1] "Log return 95% CI: ( -0.063 , 0.067 )"

# 95% CI of real return
up95 <- exp(up95) - 1
lo95 <- exp(lo95) - 1
paste("Real return 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Real return 95% CI: ( -0.061 , 0.069 )"

# 95% CI of next day's price
up95 <- (1 + up95) * stock$price[length(stock$price)]
lo95 <- (1 + lo95) * stock$price[length(stock$price)]
paste("Adjusted Price 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Adjusted Price 95% CI: ( 6.292 , 7.162 )"
```

Naive model - Student-t

```r
# Modify our assumption of the distribution of log return
# We could check which distribution is suitable for our data
library(fitdistrplus)
descdist(stock$log_r[-1], discrete = FALSE)
```



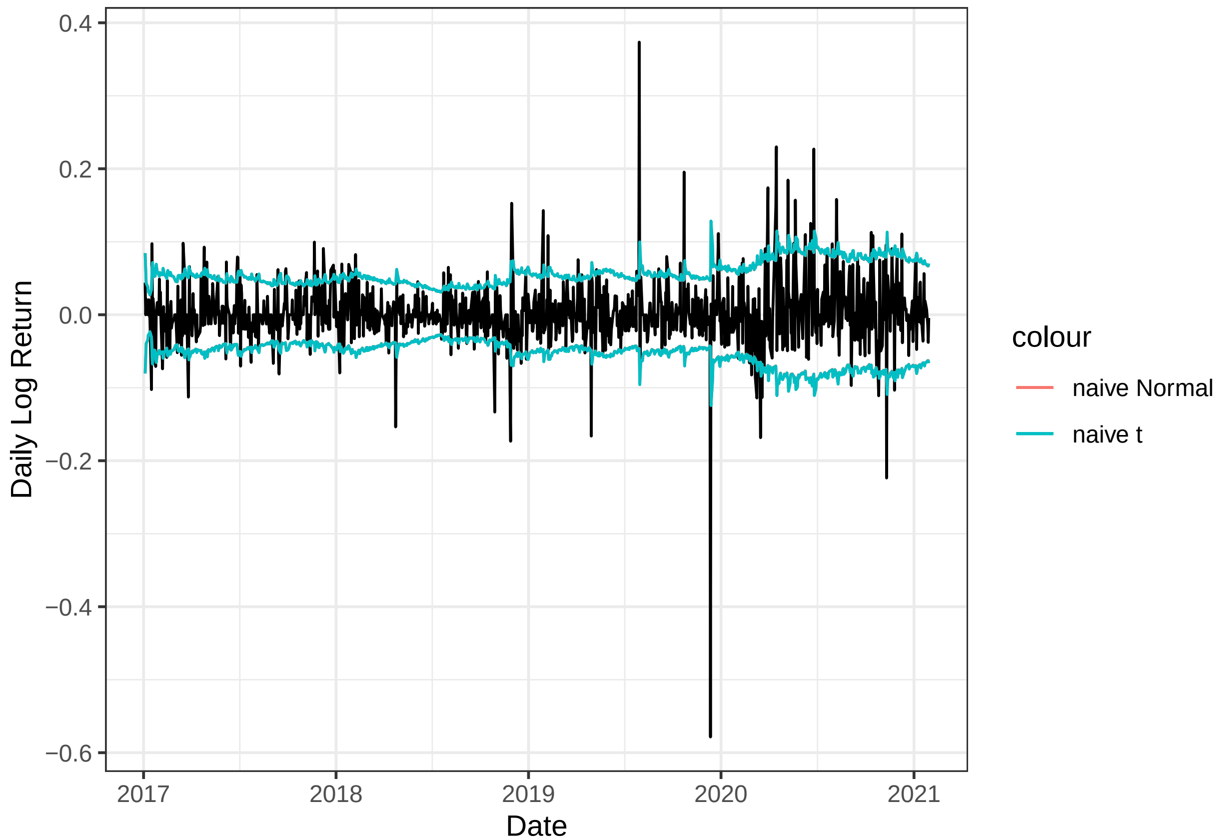

```
## summary statistics
## ------
## min:  -0.5784382   max:  0.3734489
## median:  0
## mean:  0.001965211
## estimated sd:  0.04603639
## estimated skewness:  -1.058687
## estimated kurtosis:  33.83639
```

```
# Now we fit a student-t distribution and compute the critical value
df <- fitdist(stock$log_r[-1], "t", start=list(df=2)) %>%
  .[['estimate']] %>%
  as.numeric()

crit <- qt(p = 0.975, df = df)
crit
## [1] 1.96718
```

```
# Compute the 95% CI
stock <- mutate(stock,
                upper_t = c(NA, mod1$fitted * crit + est_mean),
                lower_t = c(NA, mod1$fitted * -crit + est_mean))
```

```
# Plot our predicted 95% CI against the real log return
# Most of the time the black line should fall between the red lines or the blue lines

ggplot(stock) +
```

```r
  geom_line(aes(Date, log_r)) +
  geom_line(aes(Date, upper, col = "naive Normal")) +
  geom_line(aes(Date, lower, col = "naive Normal")) +
  geom_line(aes(Date, upper_t, col = "naive t")) +
  geom_line(aes(Date, lower_t, col = "naive t")) +
  theme_bw() +
  ylab("Daily Log Return")
```




```r
# How many percent of the time the log return will exceed the 95% CI
mean(stock$log_r > stock$upper_t | stock$log_r < stock$lower_t, na.rm = T)
## [1] 0.1299709
```

```r
# Predict the next day log return
point_forecast <- forecast(mod1, 1) %>%
  .[["mean"]]

# Chance of suffer a loss
paste("Chance of suffer a loss:",
      round(pt((-est_mean)/point_forecast, df = df)*100, 3), "%")
## [1] "Chance of suffer a loss: 47.63 %"

# 95% CI of log return
up95 <- point_forecast * crit + est_mean
lo95 <- point_forecast * -crit + est_mean
paste("Log return 95% CI: (", round(lo95, 3), ",", round(up95 , 3), ")")
## [1] "Log return 95% CI: ( -0.063 , 0.067 )"
```

```r
# 95% CI of real return
up95 <- exp(up95) - 1
lo95 <- exp(lo95) - 1
paste("Real return 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Real return 95% CI: ( -0.061 , 0.069 )"

# 95% CI of next day's price
up95 <- (1 + up95) * stock$price[length(stock$price)]
lo95 <- (1 + lo95) * stock$price[length(stock$price)]
paste("Adjusted Price 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Adjusted Price 95% CI: ( 6.291 , 7.164 )"
```

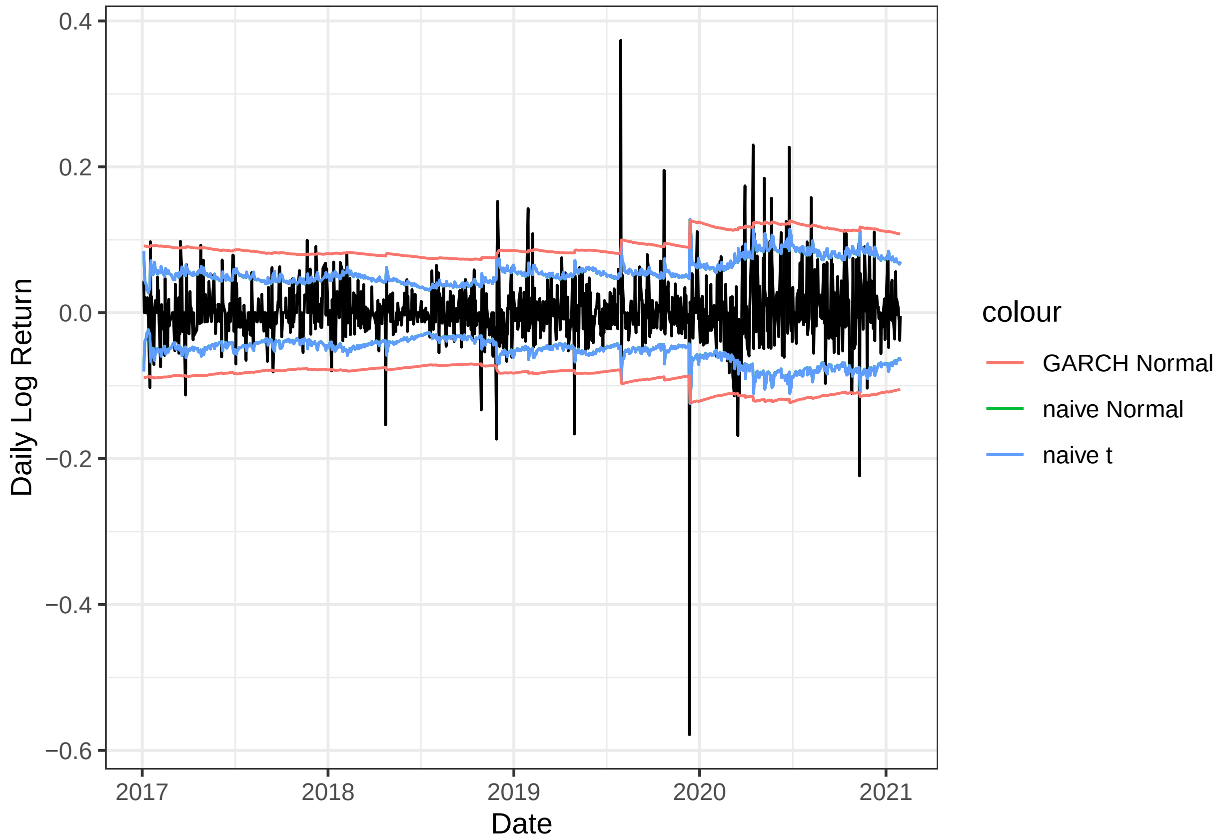
GARCH model - Normal

```r
library(rugarch)
```

```r
# Fit a ARMA(0,0) - GARCH(1,1) Model
garch_spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                               garchOrder = c(1,1)),
           mean.model = list(armaOrder = c(0,0),
                             include.mean = TRUE))

mod2 <- ugarchfit(garch_spec, stock$log_r[-1])
```

```r
# Compute the upper and lower bound manually
stock <- mutate(stock,
                upper_garch = c(NA, sigma(mod2) * 1.96 + fitted(mod2)),
                lower_garch = c(NA, sigma(mod2) * -1.96 + fitted(mod2)))
```

```r
# Plot our predicted 95% CI against the real log return
# Most of the time the black line should fall between the 95% CI

ggplot(stock) +
  geom_line(aes(Date, log_r)) +
  geom_line(aes(Date, upper, col = "naive Normal")) +
  geom_line(aes(Date, lower, col = "naive Normal")) +
  geom_line(aes(Date, upper_t, col = "naive t")) +
  geom_line(aes(Date, lower_t, col = "naive t")) +
  geom_line(aes(Date, upper_garch, col = "GARCH Normal")) +
  geom_line(aes(Date, lower_garch, col = "GARCH Normal")) +
  theme_bw() +
  ylab("Daily Log Return")
```




```
# How many percent of the time the log return will exceed the 95% CI
# If it is greater than 5%, it means the assumption of the distribution is wrong or
# the prediction of the risk is inaccurate
mean(stock$log_r > stock$upper_garch | stock$log_r < stock$lower_garch,
     na.rm = T)
## [1] 0.03394762
```

```
# Predict the next day log return
garch_est_mean <- ugarchforecast(mod2, n.ahead = 1)@forecast$seriesFor
garch_est_risk <- ugarchforecast(mod2, n.ahead = 1)@forecast$sigmaFor

# Chance of suffer a loss
paste("Chance of suffer a loss:",
      round(pnorm(0, mean = garch_est_mean, sd = garch_est_risk)*100, 3), "%")
## [1] "Chance of suffer a loss: 48.966 %"

# 95% CI of log return
up95 <- garch_est_risk * 1.96 + garch_est_mean
lo95 <- garch_est_risk * -1.96 + garch_est_mean
paste("Log return 95% CI: (", round(lo95, 3), ",", round(up95 , 3), ")")
## [1] "Log return 95% CI: ( -0.105 , 0.107 )"

# 95% CI of real return
up95 <- exp(up95) - 1
lo95 <- exp(lo95) - 1
paste("Real return 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
```

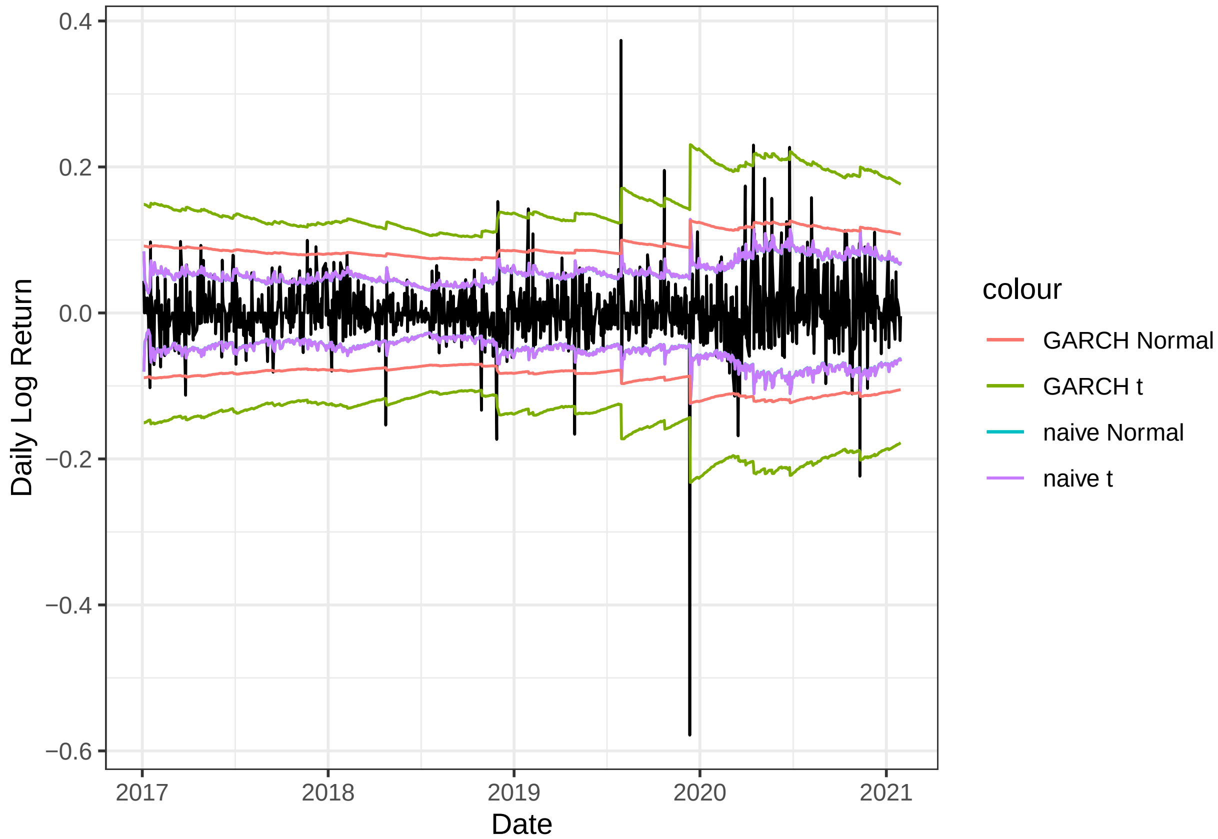
```
## [1] "Real return 95% CI: ( -0.099 , 0.113 )"

# 95% CI of next day's price
up95 <- (1 + up95) * stock$price[length(stock$price)]
lo95 <- (1 + lo95) * stock$price[length(stock$price)]
paste("Adjusted Price 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Adjusted Price 95% CI: ( 6.034 , 7.46 )"
```

GARCH - Student-t

```
# Fit a ARMA(0,0) - GARCH(1,1) - student-t Model
garch_spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                               garchOrder = c(1,1)),
           mean.model = list(armaOrder = c(0,0),
                             include.mean = TRUE),
           distribution = "std")

mod3 <- ugarchfit(garch_spec, stock$log_r[-1])
```

```
# Compute the critical value
crit <- qt(0.975, df = mod3@fit$coef['shape'])
crit
## [1] 3.246707

# Compute the upper and lower bound manually
stock <- mutate(stock,
                upper_garch_t = c(NA, sigma(mod3) * crit + fitted(mod3)),
                lower_garch_t = c(NA, sigma(mod3) * -crit + fitted(mod3)))
```

```
# Plot our predicted 95% CI against the real log return
# Most of the time the black line should fall between the 95% CI

ggplot(stock) +
  geom_line(aes(Date, log_r)) +
  geom_line(aes(Date, upper, col = "naive Normal")) +
  geom_line(aes(Date, lower, col = "naive Normal")) +
  geom_line(aes(Date, upper_t, col = "naive t")) +
  geom_line(aes(Date, lower_t, col = "naive t")) +
  geom_line(aes(Date, upper_garch, col = "GARCH Normal")) +
  geom_line(aes(Date, lower_garch, col = "GARCH Normal")) +
  geom_line(aes(Date, upper_garch_t, col = "GARCH t")) +
  geom_line(aes(Date, lower_garch_t, col = "GARCH t")) +
  theme_bw() +
  ylab("Daily Log Return")
```




```
# How many percent of the time the log return will exceed the 95% CI
# If it is greater than 5%, it means the assumption of the distribution is wrong or
# the prediction of the risk is inaccurate
mean(stock$log_r > stock$upper_garch_t | stock$log_r < stock$lower_garch_t,
     na.rm = T)
## [1] 0.01163919
```

```
# Predict the next day log return
garch_est_mean <- ugarchforecast(mod3, n.ahead = 1)@forecast$seriesFor
garch_est_risk <- ugarchforecast(mod3, n.ahead = 1)@forecast$sigmaFor

# Chance of suffer a loss
paste("Chance of suffer a loss:",
      round(pt((-garch_est_mean)/garch_est_risk,
               df = mod3@fit$coef['shape'])*100,
            3),
      "%")
## [1] "Chance of suffer a loss: 50.587 %"

# 95% CI of log return
up95 <- garch_est_risk * crit + garch_est_mean
lo95 <- garch_est_risk * -crit + garch_est_mean
paste("Log return 95% CI: (", round(lo95, 3), ",", round(up95 , 3), ")")
## [1] "Log return 95% CI: ( -0.177 , 0.175 )"

# 95% CI of real return
```

```r
up95 <- exp(up95) - 1
lo95 <- exp(lo95) - 1
paste("Real return 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Real return 95% CI: ( -0.162 , 0.192 )"

# 95% CI of next day's price
up95 <- (1 + up95) * stock$price[length(stock$price)]
lo95 <- (1 + lo95) * stock$price[length(stock$price)]
paste("Adjusted Price 95% CI: (", round(lo95, 3), ",", round(up95, 3), ")")
## [1] "Adjusted Price 95% CI: ( 5.613 , 7.984 )"
```