

```

"""
    Project   : Mini Project 5
    Course    : Econometrics
    Author    : Tawanda Vera
    Date      : 03/26/2018

"""

import pandas as pd
from scipy.stats import norm
import quandl # first pip install quandl in anaconda command prompt
import arch   # first pip install arch in anaconda command prompt (as admin)
import matplotlib.pyplot as plt

#####
# 1. Quantify the Maximum Expected Loss for the next day
# using a Value-at-Risk (VaR) model.
#####

n_shares=100000          # number of Apple shares
confidence_level=0.95     # input 2
n_days=1                 # time
stock_price = 126        # stock price
volatility = 0.025        # volatility
z=norm.ppf(confidence_level) # z-score at 95% Confidence Level
position = n_shares*stock_price # current market value of stock
VaR = position*z*volatility # When the time period is short, such as 1 day
                                # we ignore the impact of mean for the period.
# Value at Risk for holding Apple Shares
print("Holding=",position, "VaR=", round(VaR,4), "in ", n_days, "Days");

# Comments
print("Q1. Today, the value of the investment fund's holding of Apple shares\
is $12,600,000. The maximum expected loss is $518,128.8925 in the next day\
with a confidence of 95 percent.\n ");

#####

# Having errors, so downloaded csv from quandl
aapl = quandl.get("EOD/AAPL", authtoken="nPxEmstFZUATpB25wB_-")

df = pd.DataFrame(aapl)
returns = 100 * df['Adj_Close'].pct_change().dropna() # returns
returns.plot()
plt.title("APPLE Stock Returns")
plt.show()

# By default forecasts will only be produced for the final observation
# in the sample so that they are out-of-sample.
# Forecasts start with specifying the model and estimating parameters.
#
model= arch.arch_model(returns, vol='Garch', p=1, o=0, q=1, dist='Normal')

results =model.fit(update_freq=5) # model fit

print('Garch Forecast=', results.summary()) # Garch Forecast results

```

```

# Forecasts are contained in an ARCHModelForecast object
# which has 4 attributes:
# mean - The forecast means
# residual_variance is equal to variance
# The variance will differ from the residual variance, whenever the model
# has mean dynamics, e.g., in an AR process.
# simulations - the simulations used to generate forecasts.
# Only used if the forecast method is set to 'simulation' or 'bootstrap'.
# If using 'analytical' (the default), this is None.

forecasts = results.forecast()
sims = results.forecast(method='simulation')

# The three main outputs are all returned in DataFrames with columns
# of the form h.# where # is the number of steps ahead.
# That is, h.1 corresponds to one-step ahead forecasts
# while h.10 corresponds to 10-steps ahead.
# The default forecast only produces 1-step ahead forecasts.
#

print(forecasts.mean.iloc[-3:].dropna())
print(forecasts.residual_variance.iloc[-3:].dropna())
print(forecasts.variance.iloc[-3:].dropna())
print(sims.variance.iloc[-3:].dropna())

# Multiplied by 0.01 to convert it from percentage to decimal
#
fv = 0.01 * (forecasts.variance.iloc[-1].values.item())**0.5

# Closing price of AAPL shares
closing_price = df['Adj_Close'][-1:].values.item()
#
# Current market position
portfolio_value = closing_price * n_shares
#
# Value at Risk Model with Forecasted Volatility
#
Value_at_Risk = portfolio_value * fv * z

# Value at Risk for holding Apple Shares using forecasted volatility
#
print("Holding=", portfolio_value, "VaR=", round(Value_at_Risk,4), "in ",
      n_days, "Days")

# Comments
print("Question 2. VaR using Forecasted Volatility: For the investment fund's\
      holding of Apple shares with a value of USD",portfolio_value,
      "there is a 95 percent confidence that in the next day\
      the maximum expected loss will be USD",round(Value_at_Risk,4))

#####

```