Volatility Modeling & Portfolio Optimization

Commercial Banking, Corp.

RFP #: VM - S1.H2

February 18, 2019

Orange Team #1
Morgan Groves, Bill Jenista, Mia Wu, Pierce Secola & Dave Hiltbrand

Executive Summary

This report is a response to Commercial Banking, Corp's (hereafter the Bank) Request for Proposal (RFP) VM – S1.H2 for analytics services to model risk versus return among thirty stocks and develop an optimized portfolio with the top five stocks. Using the information in the RFP, the analytics team modeled the following five stocks IBM, Johnson & Johnson, Nike, Procter & Gamble, and Walmart (referenced by ticker symbol IBM, JNJ, NKE, PG, and WMT within the report) in order to predict five days of volatility using a combination of GARCH family models. Furthermore, the team optimized a portfolio of the five aforementioned stocks using both predictions from their respective models and historical median returns. Upon calculating the optimum portions for each stock within the portfolio, the team created an efficient frontier plot as displayed below. This plot allows the Bank's customers to evaluate the level of risk they are assuming when seeking specific minimum returns.

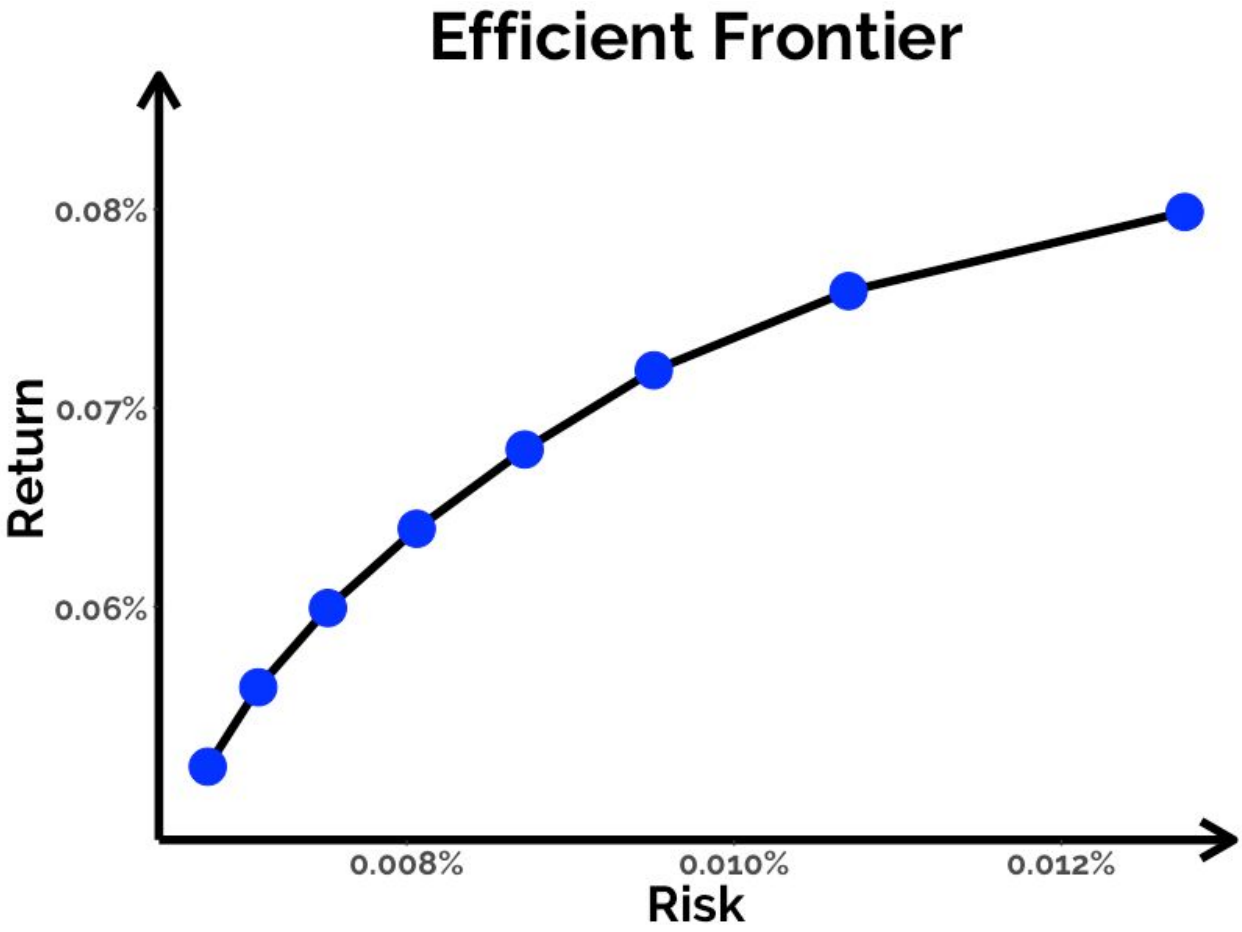

Figure 1 - Efficient Frontier is a bivariate comparison between the level of risk and the desired return for the optimized portfolio.

Methodology

As per the RFP, the Bank is requesting an analysis of the Dow Jones Industrial Average (DJIA) stocks. We collected the daily closing prices for the DJIA stocks from February 1, 2017 to February 8, 2019 from Yahoo Finance. From the closing prices, we calculated daily returns for each stock. In accordance with the Bank's RFP, we tested each stock for suitability of being modeled using a GARCH approach and determined the five most appropriate stocks. For these five stocks, we compared multiple GARCH extensions and determined the best for each stock. The stocks were ranked by sensitivity to market shock and length of shock persistence. The best models were then used to forecast return volatility for February 11 through 15, 2019.

Lastly, the forecasted volatility and median returns were used to optimize the daily stock strategy for the five day period. Our solution minimized risk for a stated 5-day return. Using our solution, we created an efficient frontier for 5-day returns ranging from 0.26% to 0.40%.

Analysis

To select stocks for the financial portfolio, the team analyzed which stock's volatility would be most appropriate to model. Modeling a stock's volatility is significant because this process provides insight into the range of potential values a stock can cover. Specifically, the model analyzed how sensitive a stock's volatility is to changes in the market, as well as how long the effect of a market shock would last for the specific stock. After finding the best model for each stock, the team concluded the volatility of IBM, JNJ, and PG are not very sensitive market events but event effects take a while to die out. NKE and WMT are more sensitive to market changes, but effects dissipate quickly.

The team forecasted five days of volatility for the selected portfolio of stocks. To understand how the stock's volatility moved together within the market covariance values were created using historical data. The team felt that using 2 years of historical data provided a more accurate depiction of how the stocks volatility moved together, rather than their predicted 5 day volatility. Using this information, the team could better estimate the risk of each stock and maximize returns on the portfolio given a target risk.

Utilizing an optimization algorithm, the team created a portfolio that balances risk and return within the portfolio. The ideal ratio between the stocks is given below in Table 1. Keep in mind that all of the client's available funds are invested to fill the five proportions. Finally, the team repeated this portfolio optimization process multiple times for different

rates of return, to understand the relationship between risk and returns of the portfolio (Figure 1).

Optimized Proportion for Each Stock	
Stock Name	**Percentage**
IBM	2.8
Johnson & Johnson	29.9
Nike	9.5
Proctor & Gamble	10.5
Walmart	47.4

Table 1 - Proportion of each stock recommended based on the optimization model.

Conclusion

Using advanced techniques, the analytics team optimized portfolio of top stocks that the Bank can utilize immediately. The company discovered IBM, Johnson & Johnson, Nike, Procter & Gamble, and Walmart were the most suitable stocks for predicting volatility. Using a combination of GARCH family models, the team predicted 5 days of volatility for each of the stocks. The team then used these predicted values along with the historical median of each stock to build an optimization model. The output of the model presented a clear combination of stocks to minimize risk for given returns: IBM - 2.8%, JNJ - 29.9%, NKE - 9.5%, PG - 10.5%, and WMT - 47.4%. The efficient frontier in FIgure 1 will allow the Bank to determine how to present the expected risk for the minimum returns their clients are seeking. Looking ahead the team would like to explore riskless assets using a smaller proportion of available client funds.

Appendix

A.1 - R code for Part 1

```
# load CSV data file
library(stringr)
library(graphics)
library(quantmod)
library(TTR)
library(ks)
library(scales)
library(forecast)
library(aTSA)
library(ccgarch)
library(fGarch)
library(rugarch)
library(dplyr)

DJIA =
c("MMM","AXP","AAPL","BA","CAT","CVX","CSCO","KO","DWDP","XOM","GS","HD","IBM","INTC","JNJ",
      "JPM","MCD","MRK","MSFT","NKE","PFE","PG","TRV","UNH","UTX","VZ","V","WMT","WBA","DIS")


# getSymbols(DJIA)
#
# stocks <-
cbind(MMM[,4],AXP[,4],AAPL[,4],BA[,4],CAT[,4],CVX[,4],CSCO[,4],KO[,4],DWDP[,4],XOM[,4],GS[,4],HD[,4]
,
#
IBM[,4],INTC[,4],JNJ[,4],JPM[,4],MCD[,4],MRK[,4],MSFT[,4],NKE[,4],PFE[,4],PG[,4],TRV[,4],UNH[,4],UTX[,4
],VZ[,4],V[,4],WMT[,4],WBA[,4],DIS[,4])
#
# paste_names <- paste(DJIA,".returns",sep = '')
#
# #Create Rolling returns for each stock
# #Cbind the stock DF with the returns DF
#
# iterations = 3049
# variables = 30
#
# output <- matrix(ncol=variables, nrow=iterations)
#
# for(i in 1:variables){
#   output[,i] <- ROC(stocks[,i])
# }
#
# stocks <- cbind(stocks,output)
#
# stocks <- stocks["2017-02/2019-02-08"]
#
```

```
# paste_names
#
# colnames(stocks)[31:60] <- c(paste_names)

stocks <- readr::read_csv("financial_analytics/hw/Stocks.csv")
stocks <- stocks[,2:61]

# Part 1 #

# Rank stocks by ARCH test significance
ARCH_tests = list()
for (i in 1:30){
 ARCH_tests[DJIA[i]] = arch.test(arima(stocks[,i+30], order = c(0,0,0)), output = TRUE)[,4]
}
warnings()
tests <- data.frame(unlist(ARCH_tests), DJIA)
tests <- arrange(tests,desc(unlist.ARCH_tests.))
## Top 5: WMT, JNJ, PG, IBM, NKE
top5 =
c("IBM.Close","JNJ.Close","NKE.Close","PG.Close","WMT.Close","ibm_r","jnj_r","nke_r","pg_r","wmt_r")

# Create GARCH models and get parameters and AIC for each stock

info = data.frame()

for (i in top5[6:10]){

 temp <- data.frame()

 j <- 1
 # GARCH-Normal
 GARCH.N <- garchFit(formula= ~ garch(1,1), data=stocks[,i],
              cond.dist="norm", include.mean = FALSE)
 temp[i,"model"] <- "GARCH-Normal"
 temp[i,"alpha"] <- GARCH.N@fit$par["alpha1"]
 temp[i,"beta"]  <- GARCH.N@fit$par["beta1"]
 temp[i,"AIC"]   <- GARCH.N@fit$ics["AIC"]

 j <- 2
 # GARCH-t
 GARCH.t <- garchFit(formula= ~ garch(1,1), data=stocks[,i],
              cond.dist="std", include.mean = FALSE)
 temp[j,"model"] <- "GARCH-t"
 temp[j,"alpha"] <- GARCH.t@fit$par["alpha1"]
 temp[j,"beta"]  <- GARCH.t@fit$par["beta1"]
 temp[j,"AIC"]   <- GARCH.t@fit$ics["AIC"]

 j <- 3
 # QGARCH-Normal
 Skew.GARCH.N <- garchFit(formula= ~ garch(1,1), data=stocks[,i],
                  cond.dist="snorm",include.mean = FALSE)
 temp[j,"model"] <- "QGARCH-Normal"
 temp[j,"alpha"] <- Skew.GARCH.N@fit$par["alpha1"]
 temp[j,"beta"]  <- Skew.GARCH.N@fit$par["beta1"]
```

```
 temp[j,"AIC"]   <- Skew.GARCH.N@fit$ics["AIC"]

 j <- 4
 # QGARCH-t
 Skew.GARCH.t <- garchFit(formula= ~ garch(1,1), data=stocks[,i],
                 cond.dist="sstd", include.mean = FALSE)
 temp[j,"model"] <- "QGARCH-t"
 temp[j,"alpha"] <- Skew.GARCH.t@fit$par["alpha1"]
 temp[j,"beta"]  <- Skew.GARCH.t@fit$par["beta1"]
 temp[j,"AIC"]   <- Skew.GARCH.t@fit$ics["AIC"]

 info <- bind_rows(info, temp)
}

# best models
best_GARCH_models <- info[c(2,8,10,14,20),]
top <- c("IBM", "JNJ", "NKE","PG", "WMT")
best_GARCH_models["stock"] <- top

# Next 5 days of volatility for using top model for each stock
# IBM best model is GARCH-t
ibm_mod <- garchFit(formula= ~ garch(1,1), data=stocks[,'ibm_r'],
              cond.dist="std", include.mean = FALSE)
ibm_for <- predict(ibm_mod, n.ahead=5)$standardDeviation

# JNJ best model is QGARCH-t
jnj_mod <- garchFit(formula= ~ garch(1,1), data=stocks[,'jnj_r'],
              cond.dist="sstd", include.mean = FALSE)
jnj_for <- predict(jnj_mod, n.ahead=5)$standardDeviation

# NKE best model is GARCH-t
nke_mod <- garchFit(formula= ~ garch(1,1), data=stocks[,'nke_r'],
              cond.dist="std", include.mean = FALSE)
nke_for <- predict(nke_mod, n.ahead=5)$standardDeviation

# PG best model is GARCH-t
pg_mod <- garchFit(formula= ~ garch(1,1), data=stocks[,'pg_r'],
             cond.dist="std", include.mean = FALSE)
pg_for <- predict(pg_mod, n.ahead=5)$standardDeviation

# WMT best model is QGARCH-t
wmt_mod <- garchFit(formula= ~ garch(1,1), data=stocks[,'wmt_r'],
              cond.dist="sstd", include.mean = FALSE)
wmt_for <- predict(wmt_mod, n.ahead=5)$standardDeviation

# rank by alpha (most susceptible to market shock)
rank_alpha <- arrange(best_GARCH_models, desc(alpha))
## from most to least susceptible: NKE, WMT, JNJ, IBM, PG

# rank by beta (longest lasting shock effects)
rank_beta <- arrange(best_GARCH_models, desc(beta))
## from longest to shortest effect: PG, IBM, JNJ, NKE, WMT

# Part 2 ####
```

```
# historical median return for each of 5 top stocks
hist_med_ret <- purrr::map_dbl(stocks[,top5[6:10]], median) %>%
 cbind(top5[6:10], .)

readr::write_csv(stocks[,c('Index',top5)], path = "financial_analytics/data/stocks_top5.csv")
```

A.2 - SAS code for Optimization

```
/* Load Needed Data */
proc import datafile = 'stocks_top5.csv'
            out = stocks_top5 dbms = csv replace;
run;

/* Reset the Format to Avoid SGPLOT Warnings Later On */
data stocks_top5;
 set stocks_top5;
 *Date = input(Index,anydtdte32.);
 format index date7.;
run;

/* Add 5 Observations for Forecasting at End of Series */
data stocks_top5(drop=i);
            set stocks_top5 end = eof;
            output;
            if eof then do i=1 to 5;
                        if weekday(index) in(2,3,4,5) then index=index+1;
                        else index=index+3;
   ibm_r=.;
            jnj_r=.;
            nke_r=.;
            pg_r=.;
            wmt_r=.;
   output;
 end;
run;

/* Estimate Different GARCH Models */
proc autoreg data=stocks_top5;
  ibm_garch_t:  model ibm_r =  / garch=(p=1, q=1, type=tGARCH) dist=t method=ml maxiter=250;
                output out=ibm_garch_t ht=pred_var_ibm;
  jnj_qgarch:  model jnj_r =  / garch=(p=1, q=1, type=QGARCH) dist=t method=ml maxiter=250;
                output out=jnj_qgarch ht=pred_var_jnj;
  nke_garch_t:  model nke_r = / garch=(p=1, q=1, type=tGARCH) dist=t method=ml maxiter=250;
                output out=nke_garch_t ht=pred_var_nke;
  pg_garch_t:  model pg_r = / garch=(p=1, q=1, type=tGARCH) dist=t method=ml maxiter=250;
                output out=pg_garch_t ht=pred_var_pg;
  wmt_qgarch:  model wmt_r =  / garch=(p=1, q=1, type=QGARCH) dist=t method=ml maxiter=250;
                output out=wmt_qgarch ht=pred_var_wmt;
run;

/* Get New Forecasted Values for Variance */
data combined (keep=date pred_var_ibm pred_var_jnj pred_var_nke pred_var_pg pred_var_wmt);
```

```
	merge ibm_garch_t jnj_qgarch nke_garch_t pg_garch_t wmt_qgarch;
	if index <= '08FEB2019'd then delete;
run;

proc means data=combined median;
	var pred_var_ibm pred_var_jnj pred_var_nke pred_var_pg pred_var_wmt;
	output out=P_GARCH median=p_ibm p_jnj p_nke p_pg p_wmt;
run;

data _null_;
	set P_GARCH;
	call symput("ibm_pred", p_ibm);
	call symput("jnj_pred", p_jnj);
	call symput("nke_pred", p_nke);
	call symput("pg_pred", p_pg);
	call symput("wmt_pred", p_wmt);
run;

proc corr data=stocks_top5 cov  out=Corr;
	var ibm_r jnj_r nke_r pg_r wmt_r;
run;

proc means data=stocks_top5 median noprint;
	var ibm_r jnj_r nke_r pg_r wmt_r;
	output out=Median(drop=_type_ _freq_) median=;
run;

/*data Median;
 set Median;
 rename ibm_r_Median=ibm_r jnj_r_Median=jnj_r nke_r_Median=nke_r pg_r_Median=pg_r
wmt_r_Median=wmt_r;
 _NAME_ = ' ';
run;*/

proc transpose data=median out=_expected_monthly_returns(rename=(Col1=monthly_return));
run;

data Cov;
	set Corr;
	where _TYPE_='COV';
run;

data Cov;
	set Cov;
	if _NAME_="ibm_r" then ibm_r=&ibm_pred;
	if _NAME_="jnj_r" then jnj_r=&jnj_pred;
	if _NAME_="nke_r" then nke_r=&nke_pred;
	if _NAME_="pg_r" then pg_r=&pg_pred;
	if _NAME_="wmt_r" then wmt_r=&wmt_pred;
run;



/*Use OPTMODEL to setup and solve the minimum variance problem*/
```

```
proc optmodel printlevel=0 FDIGITS=8;
set <str> Stock_Symbols;

/*DECLARE OPTIMIZATION PARAMETERS*/
/*Expected return for each stock*/
num expected_return_stock{Stock_Symbols};
/*Covariance matrix of stocks*/
num Covariance{Stock_Symbols,Stock_Symbols};
/*Required portfolio return: PARAMETER THAT WE WILL ANALYZE*/
num Required_Portfolio_Return;
/*Range of parameter values*/
set parameter_values = {0.0026 to 0.004 by 0.0002};

/*OUTPUT*/
/*Array to hold the value of the objective function*/
num Portfolio_Stdev_Results{parameter_values};
/*Array to hold the value of the exp. return*/
num Expected_Return_Results{parameter_values};
/*Array to hold the value of the weights*/
num Weights_Results{parameter_values,Stock_Symbols};

/* DECLARE OPTIMIZATION VARIABLES AND THEIR CONSTRAINTS*/
/*Short positions are not allowed*/
var weights{Stock_Symbols}>=0;

/* Declare implied variables (Optional)*/
impvar exp_portf_return = sum{i in Stock_Symbols} expected_return_stock[i] * weights[i];

/* Declare constraints */
con c1: sum{i in Stock_Symbols} weights[i] = 1;
con c2: (exp_portf_return+1)**5 - 1 = Required_Portfolio_Return;
con c3: exp_portf_return >= 0.0005;

/*READ INPUT DATA*/
/*Read the expected monthly returns. The first column, _name_ holds the    */
/*index of stock symbols we want to use; that's why we include it with [].*/
read data work._expected_monthly_returns into Stock_Symbols=[_name_]
expected_return_stock=monthly_return;
/*Read the covariance matrix*/
read data work.cov into [_name_] {j in Stock_Symbols} <Covariance[_name_,j]=col(j)>;

/* DECLARE OBJECTIVE FUNCTION */
min Portfolio_Stdev =
  sum{i in Stock_Symbols, j in Stock_Symbols}Covariance[i,j] * weights[i] * weights[j];

/*SOLVE THE PROBLEM FOR EACH PARAMETER VALUE*/
for {r in parameter_values} do;
 /*Set the minimum portfolio return value to be used in each case*/
 Required_Portfolio_Return=r;
 solve;
 /*Store the value of the objective function*/
 Portfolio_Stdev_Results[r]=Portfolio_Stdev;
 /*Store the value of the expected returns*/
 Expected_Return_Results[r]=exp_portf_return;
```

```
   /*Store the weights*/
   for {i in Stock_Symbols} do;
     Weights_Results[r,i]=weights[i];
   end;
 end;

  /*Store the portfolio return and std.dev from all runs in a SAS dataset*/
  create data fin.obj_value_stddev_results from
      [parameter_values] Portfolio_Stdev_Results Expected_Return_Results;

  /*Store the weights from all runs in a SAS dataset*/
  create data fin.min_stddev_weight_results from
      [_param_ _stock_]={parameter_values , stock_symbols} Weights_Results;
quit;



/*Efficient Frontier*/
proc sgplot data=Obj_value_stddev_results;
series x=Portfolio_Stdev_Results y=expected_return_results;
quit;

/* calculating average proportion for each stock over the 5 day period*/
proc means data=min_stddev_weight_results mean;
class _stock_;
var weights_results;
output out=fin.proportions mean=;
run;
```

A.3 - R code for Efficient Frontier

```
library(tidyverse)
library(extrafont)
font_import()
results <- haven::read_sas("financial_analytics/data/opt_results.sas7bdat")

sd_risk <- sd(results$Portfolio_Stdev_Results)
ggplot(results, aes(Portfolio_Stdev_Results, Expected_Return_Results)) +
 geom_line(size=2) +
 geom_point(size=8, color='blue') +
 theme_bw() +
 labs(x='Risk', y='Return', title='Efficient Frontier') +
 theme(plot.title = element_text(hjust = 0.5, size = 32, face = 'bold', family = 'Raleway'),
       axis.title = element_text(size = 24, face = 'bold', family = 'Raleway'),
       axis.text = element_text(size = 16, face = 'bold', family = 'Raleway'),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
       axis.line = element_line(arrow = arrow(),
                                size = 2),
```

```
    panel.border = element_blank()) +
scale_y_continuous(limits = c(0.0005, .00085),
                   breaks = c(0.0006, 0.0007, 0.0008),
                   labels = c('0.06%', '0.07%', '0.08%')) +
scale_x_continuous(breaks = c(0.00008, 0.0001, 0.00012),
                   labels = c('0.008%', '0.010%', '0.012%'))
```