OR538 Final Project - All

# Obtaining the data set

The first step is to create the dataset by pulling values and generating the rate of change of interest (i.e., log-returns)

In this case we are going to use the stock prices for two companies in the information technology industry in particular the Data & Processing and outsourced services (i.e., Paycheck inc. and PayPal), one company in the Consumer Staples industry (i.e., Walmart), one market (i.e., S&P 500) and the risk-free rates for 2019 provided by the US Department of Treasury.

We will be analizing the data set obtained in a time-period of 2 years, we will gather data from November 23, 2017 to November 23, 2019. The function getSymbols provided by the quantmod library will be leverage to pull data from Yahoo Finance and the federal reserve.

Below is a table mapping the symbol to the security and the source of data.

|  |  |  |
| --- | --- | --- |
| Symbol | Description | Source |
| PAYX | PayCheck Inc Stock Prices | Yahoo Finance |
| PYPL | Paypal Stock Prices | Yahoo Finance |
| WMT | Walmart Stock Prices | Yahoo Finance |
| SP | S&P 500 Stock Prices | Yahoo Finance |

pacman::p\_load(quantmod, tseries)  
  
symbols <- c("PAYX", "PYPL", "WMT", "SP")  
  
getSymbols(symbols, from = "2017-11-23", to = "2019-11-23")

## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use  
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")  
## and getOption("getSymbols.auto.assign") will still be checked for  
## alternate defaults.  
##   
## This message is shown once per session and may be disabled by setting   
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "PAYX" "PYPL" "WMT" "SP"

# Creating data frames in an extensible time-series data type

getSymbols will provide the stock prices in an extensible time-series format containing the stocks price for each one of the symbols. However, it will be more beneficial to manage data frames containing the necessary transformations that would enable us to perform the best possible analyis of financial time-series information.

The script below will create 3 data frames containing the adjusted price, return and log return for each one of the stocks. Separating this information in individual data frames will simplify the usage of such information for data visualization, model training and prediction.

df\_adj <- data.frame(PAYX=PAYX$PAYX.Adjusted,  
 PYPL=PYPL$PYPL.Adjusted,  
 WMT=WMT$WMT.Adjusted,  
 SP=SP$SP.Adjusted)  
colnames(df\_adj) <- symbols  
  
df\_ret <- data.frame(PAYX=diff(PAYX$PAYX.Adjusted),  
 PYPL=diff(PYPL$PYPL.Adjusted),  
 WMT=diff(WMT$WMT.Adjusted),  
 SP=diff(SP$SP.Adjusted))  
colnames(df\_ret) <- symbols  
  
df\_log\_ret <- data.frame(PAYX=diff(log(PAYX$PAYX.Adjusted)),  
 PYPL=diff(log(PYPL$PYPL.Adjusted)),  
 WMT=diff(log(WMT$WMT.Adjusted)),  
 SP=diff(log(SP$SP.Adjusted)))  
colnames(df\_log\_ret) <- symbols

# Consolidating the dataset

To properly share the information; in this section we will be consolidating all the data set information obtained and generated in a single csv file.

df\_export <- data.frame(PAYX, diff(PAYX$PAYX.Adjusted), diff(log(PAYX$PAYX.Adjusted)),  
 PYPL, diff(PYPL$PYPL.Adjusted), diff(log(PYPL$PYPL.Adjusted)),  
 WMT, diff(WMT$WMT.Adjusted), diff(log(WMT$WMT.Adjusted)),  
 SP, diff(SP$SP.Adjusted), diff(log(SP$SP.Adjusted)))  
  
colnames(df\_export) <- c("PAYX.Open", "PAYX.High", "PAYX.Low", "PAYX.Close",   
 "PAYX.Volume","PAYX.Adjusted", "PAYX.Adj.Ret", "PAYX.Adj.Log.Ret",  
 "PYPL.Open", "PYPL.High", "PYPL.Low", "PYPL.Close",  
 "PYPL.Volume","PYPL.Adjusted", "PYPL.Adj.Ret", "PYPL.Adj.Log.Ret",  
 "WMT.Open", "WMT.High", "WMT.Low", "WMT.Close",   
 "WMT.Volume", "WMT.Adjusted","WMT.Adj.Ret", "WMT.Adj.Log.Ret",  
 "SP.Open", "SP.High", "SP.Low", "SP.Close",  
 "SP.Volume", "SP.Adjusted", "SP.Adj.Ret", "SP.Adj.Log.Ret")  
  
write.csv(df\_export,   
 "/Users/angelsierra/ML\_Project/OR538-FinalPrj/OR538-portfolio-securities.csv")

# Data Pre-Processing

The intent of this section is to perform the necessary pre-processing for the time series values and leverage data visualization techniques to show the transformations and validate key assumptions around the time-series.

pacman::p\_load(xts)  
folder <- "/Users/angelsierra/ML\_Project/OR538-FinalPrj/Figures/"  
pdf(file = paste(folder,"Figure1-Adjusted Price for Multiple Securities.pdf"),height=7,width=8)  
plot(as.xts(df\_adj), main="Adjusted Price", legend.loc = "topleft")  
graphics.off()  
  
plot(as.xts(df\_adj), main="Adjusted Price", legend.loc = "topleft")

**Figure 1:** Adjusted price plot for PayCheck Inc., PayPal, Walmart, and S&P 500 securities.

Figure **1** plots the adjusted price for all the securities analyzed in this project. We can observe from this plot that the securities present significant differences between prices for each security and they do not present stationarity.

We will calculate the price difference for each adjusted price and observe if this would help with the two main concerns found when reviewing Figure **1**.

pdf(file = paste(folder,"Figure2-Returns for Multiple Securities.pdf"),height=7,width=8)  
plot(as.xts(df\_ret), main="Returns", legend.loc = "topleft")  
graphics.off()  
  
plot(as.xts(df\_ret), main="Returns", legend.loc = "topleft")

**Figure 2:** Adjusted price returns plot for PayCheck Inc., PayPal, Walmart, and S&P 500 securities.

Figure **2** plots the adjusted price returns for all the securities analyzed in this project. We can start observing mean-reversion, a common condition that validates the stationarity assumption. However, we can observe that the color distribution is not even between securities. For instance, we can see a heavier pressence in the time series for PayCheck, Paypal and Walmart but not so much for S&P 500. In an attempt to fix this differences we would employ the log transformation in all series.

pdf(file = paste(folder,"Figure3-Log Returns for Multiple Securities.pdf"),height=7,width=8)  
plot(as.xts(df\_log\_ret), main="Log Returns", legend.loc = "topleft")  
graphics.off()  
  
plot(as.xts(df\_log\_ret), main="Log Returns", legend.loc = "topleft")

**Figure 3:** Adjusted price log returns plot for PayCheck Inc., PayPal, Walmart, and S&P 500 securities.

Figure **3** plots the adjusted price log returns for all the securities analyzed in this project. In this transformation we can obser that the color distribution seems to be evenly distributed between the multiple securities analyzed in this project. For this reason, we will use the Adjusted Price log returns to train our models.

pdf(file = paste(folder,"Figure4-Returns vs.Log Returns.pdf"),height=7,width=8)  
par(mfrow=c(2,2))  
plot(as.xts(df\_ret), main="Returns", legend.loc = "topleft")  
plot(as.xts(df\_log\_ret), main="Log Returns", legend.loc = "topleft")  
graphics.off()

**Note:** The above script was created to consolidate the returns and log-returns plot. This was created to allow side-by-side comparison between the two transformations.

# Finding Outliers

pdf(file = paste(folder, "Figure5-Box Plot for Multiple Securities.pdf"), height=7,width=8)  
par(mfrow=c(2,2))  
boxplot(df\_adj, main="Adjusted Price for Multiple Securities",  
 xlab="Security Symbol", ylab="Log Returns")  
boxplot(df\_ret, main="Returns for Multiple Securities",  
 xlab="Security Symbol", ylab="Log Returns")  
boxplot(df\_log\_ret, main="Log Returns for Multiple Securities",  
 xlab="Security Symbol", ylab="Log Returns")  
graphics.off()  
  
  
par(mfrow=c(2,2))  
boxplot(df\_adj, main="Adjusted Price for Multiple Securities",  
 xlab="Security Symbol", ylab="Log Returns")  
boxplot(df\_ret, main="Returns for Multiple Securities",  
 xlab="Security Symbol", ylab="Log Returns")  
boxplot(df\_log\_ret, main="Log Returns for Multiple Securities",  
 xlab="Security Symbol", ylab="Log Returns")

**Figure 5:** Box Plot for PayCheck Inc., PayPal, Walmart, and S&P 500 securities.

Figure **5** allow us to observe that the full data set to be used are centered around zero. However, we can see a couple of outliers but to properly manage this outliers we may need to work with an investment subject matter expert.

# Evaluate Time-series Autocorrelation

pdf(file=paste(folder,"Figure6-ACF-lag-default.pdf"),height=7,width=8)  
par(mfrow=c(2,2))  
acf(df\_log\_ret[,"PAYX"], main="PayCheck Inc.", na.action = na.pass)  
acf(df\_log\_ret[,"PYPL"], main="PayPal", na.action = na.pass)  
acf(df\_log\_ret[,"WMT"], main="Walmart", na.action = na.pass)  
acf(df\_log\_ret[,"SP"], main="S&P 500", na.action = na.pass)  
graphics.off()  
  
par(mfrow=c(2,2))  
acf(df\_log\_ret[,"PAYX"], main="PayCheck Inc.", na.action = na.pass)  
acf(df\_log\_ret[,"PYPL"], main="PayPal", na.action = na.pass)  
acf(df\_log\_ret[,"WMT"], main="Walmart", na.action = na.pass)  
acf(df\_log\_ret[,"SP"], main="S&P 500", na.action = na.pass)

**Figure 6:** ACF for PayCheck Inc., PayPal, Walmart, and S&P 500 securities.

Figure **6** allow us to validate that the adjusted price log returns for all the securities are stationary. Only lags greater than 12 start showing autocorrelations crossing the test lower bounds.

We can further inspect the autocorrelation for each one of these securities using the Ljung-Box test.

# Paycheck Test  
PayxLbt11 <- Box.test(df\_log\_ret[,"PAYX"], lag = 11, type = "Ljung-Box")  
PayxLbt12 <- Box.test(df\_log\_ret[,"PAYX"], lag = 12, type = "Ljung-Box")  
PayxLbt13 <- Box.test(df\_log\_ret[,"PAYX"], lag = 13, type = "Ljung-Box")  
  
  
# Paypal Test  
PyplLbt11 <- Box.test(df\_log\_ret[,"PYPL"], lag = 11, type = "Ljung-Box")  
PyplLbt12 <- Box.test(df\_log\_ret[,"PYPL"], lag = 12, type = "Ljung-Box")  
PyplLbt13 <- Box.test(df\_log\_ret[,"PYPL"], lag = 13, type = "Ljung-Box")  
  
# Walmart Test  
WmtLbt11 <- Box.test(df\_log\_ret[,"WMT"], lag = 11, type = "Ljung-Box")  
WmtLbt12 <- Box.test(df\_log\_ret[,"WMT"], lag = 12, type = "Ljung-Box")  
WmtLbt13 <- Box.test(df\_log\_ret[,"WMT"], lag = 13, type = "Ljung-Box")  
  
  
# S&P Test  
SpLbt11 <- Box.test(df\_log\_ret[,"SP"], lag = 11, type = "Ljung-Box")  
SpLbt12 <- Box.test(df\_log\_ret[,"SP"], lag = 12, type = "Ljung-Box")  
SpLbt13 <- Box.test(df\_log\_ret[,"SP"], lag = 13, type = "Ljung-Box")  
  
lag11Results <- c(PayxLbt11$p.value, PyplLbt11$p.value, WmtLbt11$p.value, SpLbt11$p.value)  
lag12Results <- c(PayxLbt12$p.value, PyplLbt12$p.value, WmtLbt12$p.value, SpLbt12$p.value)  
lag13Results <- c(PayxLbt13$p.value, PyplLbt13$p.value, WmtLbt13$p.value, SpLbt13$p.value)  
  
dfLjungTest <- as.data.frame(rbind(lag11Results,lag12Results,lag13Results))  
  
colnames(dfLjungTest) <- c("PayCheck Inc.", "Paypal", "Walmart", "S&P 500")  
rownames(dfLjungTest) <- c("lag = 11", "lag = 12", "lag = 13")  
  
dfLjungTest

## PayCheck Inc. Paypal Walmart S&P 500  
## lag = 11 0.9038208 0.6353162 0.8482605 0.6909179  
## lag = 12 0.9341674 0.7144577 0.8896300 0.6207957  
## lag = 13 0.9241058 0.7449712 0.9164899 0.2242331

Inspection of the PayCheck Inc. security allow us to observe that the best p-value is obtained when the lag is set to 12. However, Paypal and Walmart securities seems to show increasing p-values with increased number of lag. In the case of the S&P 500 security case we observe that the p-values decreases with increased lag number.

Box.test(df\_log\_ret[,"PAYX"], lag = 13, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: df\_log\_ret[, "PAYX"]  
## X-squared = 6.5439, df = 13, p-value = 0.9241

Box.test(df\_log\_ret[,"PYPL"], lag = 13, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: df\_log\_ret[, "PYPL"]  
## X-squared = 9.3635, df = 13, p-value = 0.745

Box.test(df\_log\_ret[,"WMT"], lag = 13, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: df\_log\_ret[, "WMT"]  
## X-squared = 6.7096, df = 13, p-value = 0.9165

Box.test(df\_log\_ret[,"SP"], lag = 13, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: df\_log\_ret[, "SP"]  
## X-squared = 16.479, df = 13, p-value = 0.2242

# Modeling Paycheck Inc.

In this section we will be applying AR, MA, ARMA and ARIMA modeling for the PayCheck Inc. security and evaluating the models by using Ljung-Box Test on the residuals of each model.

## AR Modeling

Key points: \* Stationarity implies that the amount of variance is equal to the amount of feedback () \* \* \* For model evalutation using Ljung-Box needs the fitdf parameter to be adjusted and its value depends on the AR model. For instance, for an AR(1) model the \* In the test a p-value of 0 indicates that at least one of the first **df** lags present an autocorrelation that is not zero

For each results plot residuals against time, ACF for the residuals and q-q plot using normal distribution for the errors (i.e. qqnorn).

any significant autocorrelation in the residuals implies that the model did not fit well.

One typical problem of AR(p) models is that they generally need a large number of p to fit the data set.

============

pacman::p\_load("forecast") # Library needed for auto.arima  
  
AR0 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,0))  
AR1 <- arima(df\_log\_ret[,"PAYX"], order = c(1,0,0))  
AR2 <- arima(df\_log\_ret[,"PAYX"], order = c(2,0,0))  
AR3 <- arima(df\_log\_ret[,"PAYX"], order = c(3,0,0))  
  
AR0Res <- c("0", AR0$aic)  
AR1Res <- c("1", AR1$aic)  
AR2Res <- c("2", AR2$aic)  
AR3Res <- c("3", AR3$aic)  
  
df\_AR\_results <- as.data.frame(rbind(AR0Res, AR1Res, AR2Res, AR3Res))  
colnames(df\_AR\_results) <- c("p", "AIC")  
  
df\_AR\_results

## p AIC  
## AR0Res 0 -3049.25413104755  
## AR1Res 1 -3047.81710165528  
## AR2Res 2 -3047.49030380538  
## AR3Res 3 -3046.04965434403

**Table 2:** AR Model Results

Table **2** allow us to observe that the best result is obtained when we use an AR(0) model.

# MA Modeling

MA0 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,0))  
MA1 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,1))  
MA2 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,2))  
MA3 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,3))  
  
MA0Res <- c("0", MA0$aic)  
MA1Res <- c("1", MA1$aic)  
MA2Res <- c("2", MA2$aic)  
MA3Res <- c("3", MA3$aic)  
  
df\_MA\_results <- as.data.frame(rbind(MA0Res, MA1Res, MA2Res, MA3Res))  
colnames(df\_MA\_results) <- c("q", "AIC")  
  
df\_MA\_results

## q AIC  
## MA0Res 0 -3049.25413104755  
## MA1Res 1 -3047.88758150813  
## MA2Res 2 -3047.40677052042  
## MA3Res 3 -3046.08839135109

**Table 3:** MA Model Results

Table **3** allow us to observe that the best result is obtained when we use an MA(0) model.

# ARMA Modeling

ARMA00 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,0))  
ARMA01 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,1))  
ARMA02 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,2))  
ARMA03 <- arima(df\_log\_ret[,"PAYX"], order = c(0,0,3))  
  
ARMA10 <- arima(df\_log\_ret[,"PAYX"], order = c(1,0,0))  
ARMA11 <- arima(df\_log\_ret[,"PAYX"], order = c(1,0,1))  
ARMA12 <- arima(df\_log\_ret[,"PAYX"], order = c(1,0,2))  
ARMA13 <- arima(df\_log\_ret[,"PAYX"], order = c(1,0,3))  
  
ARMA20 <- arima(df\_log\_ret[,"PAYX"], order = c(2,0,0))  
ARMA21 <- arima(df\_log\_ret[,"PAYX"], order = c(2,0,1))  
ARMA22 <- arima(df\_log\_ret[,"PAYX"], order = c(2,0,2))

## Warning in log(s2): NaNs produced  
  
## Warning in log(s2): NaNs produced  
  
## Warning in log(s2): NaNs produced

ARMA23 <- arima(df\_log\_ret[,"PAYX"], order = c(2,0,3))  
  
ARMA30 <- arima(df\_log\_ret[,"PAYX"], order = c(3,0,0))  
ARMA31 <- arima(df\_log\_ret[,"PAYX"], order = c(3,0,1))  
ARMA32 <- arima(df\_log\_ret[,"PAYX"], order = c(3,0,2))  
ARMA33 <- arima(df\_log\_ret[,"PAYX"], order = c(3,0,3))  
  
ARMA00Res <- c("0", "0", ARMA00$aic)  
ARMA01Res <- c("0", "1", ARMA01$aic)  
ARMA02Res <- c("0", "2", ARMA02$aic)  
ARMA03Res <- c("0", "3", ARMA03$aic)  
  
ARMA10Res <- c("1", "0", ARMA10$aic)  
ARMA11Res <- c("1", "1", ARMA11$aic)  
ARMA12Res <- c("1", "2", ARMA12$aic)  
ARMA13Res <- c("1", "3", ARMA13$aic)  
  
ARMA20Res <- c("2", "0", ARMA20$aic)  
ARMA21Res <- c("2", "1", ARMA21$aic)  
ARMA22Res <- c("2", "2", ARMA22$aic)  
ARMA23Res <- c("2", "3", ARMA23$aic)  
  
ARMA30Res <- c("3", "0", ARMA30$aic)  
ARMA31Res <- c("3", "1", ARMA31$aic)  
ARMA32Res <- c("3", "2", ARMA32$aic)  
ARMA33Res <- c("3", "3", ARMA33$aic)  
  
df\_ARMA\_results <- as.data.frame(rbind(ARMA00Res, ARMA01Res, ARMA02Res, ARMA03Res,  
 ARMA10Res, ARMA11Res, ARMA12Res, ARMA13Res,  
 ARMA20Res, ARMA21Res, ARMA22Res, ARMA23Res,  
 ARMA30Res, ARMA31Res, ARMA32Res, ARMA33Res))  
colnames(df\_ARMA\_results) <- c("p", "q", "AIC")  
  
df\_ARMA\_results

## p q AIC  
## ARMA00Res 0 0 -3049.25413104755  
## ARMA01Res 0 1 -3047.88758150813  
## ARMA02Res 0 2 -3047.40677052042  
## ARMA03Res 0 3 -3046.08839135109  
## ARMA10Res 1 0 -3047.81710165528  
## ARMA11Res 1 1 -3046.44785272058  
## ARMA12Res 1 2 -3046.52299276511  
## ARMA13Res 1 3 -3044.53606662146  
## ARMA20Res 2 0 -3047.49030380538  
## ARMA21Res 2 1 -3046.46799310694  
## ARMA22Res 2 2 -3044.53764813347  
## ARMA23Res 2 3 -3042.53200405431  
## ARMA30Res 3 0 -3046.04965434403  
## ARMA31Res 3 1 -3044.56792785024  
## ARMA32Res 3 2 -3042.5087815591  
## ARMA33Res 3 3 -3040.54181375935

**Table 4:** ARMA Model Results

Table **4** allow us to observe that the best result is obtained when we use an ARMA(0,0) model.

ARIMAautoAic <- auto.arima(df\_log\_ret[,"PAYX"], max.p = 20, max.q = 20, max.d = 20, ic = "aic")  
  
ARIMAautoAic

## Series: df\_log\_ret[, "PAYX"]   
## ARIMA(0,0,0) with zero mean   
##   
## sigma^2 estimated as 0.0001341: log likelihood=1525.81  
## AIC=-3049.62 AICc=-3049.61 BIC=-3045.4

# Conclusion

Manually performing ARIMA models for each stock could be a tedious task and require a significant amount of effort to perform. Since the end goal of the project is to get as close as possible to generate an optimal portfolio for investment we will focus the next sections on using auto.arima function with each stock and evaluate the residuals for each stock.

# Modeling Paycheck Inc.

In this section we will be applying AR, MA, ARMA and ARIMA modeling for the PayCheck Inc. security and evaluating the models by using Ljung-Box Test on the residuals of each model.

pacman::p\_load("forecast") # Library needed for auto.arima  
  
fitPayx <- auto.arima(df\_log\_ret[,"PAYX"], max.p = 20, max.q = 20, max.d = 20, ic = "aic")  
fitPypl <- auto.arima(df\_log\_ret[,"PYPL"], max.p = 20, max.q = 20, max.d = 20, ic = "aic")  
fitWmt <- auto.arima(df\_log\_ret[,"WMT"], max.p = 20, max.q = 20, max.d = 20, ic = "aic")  
fitSp <- auto.arima(df\_log\_ret[,"SP"], max.p = 20, max.q = 20, max.d = 20, ic = "aic")  
  
fitPayxRes <- c(round(fitPayx$bic,3), round(fitPayx$aic,3),  
 round(fitPayx$aicc,3), "ARIMA(0,0,0)")  
fitPyplRes <- c(round(fitPypl$bic,3), round(fitPypl$aic,3),  
 round(fitPypl$aicc,3), "ARIMA(1,0,1) = ARMA(1,1)")  
fitWmtRes <- c(round(fitWmt$bic,3), round(fitWmt$aic,3),  
 round(fitWmt$aicc,3), "ARIMA(0,0,0)")  
fitSpRes <- c(round(fitSp$bic,3), round(fitSp$aic,3),  
 round(fitSp$aicc,3), "ARIMA(0,0,0)")  
  
df\_fit\_results <- as.data.frame(rbind(fitPayxRes, fitPyplRes, fitWmtRes, fitSpRes))  
colnames(df\_fit\_results) <- c("BIC", "AIC", "AICc", "Model")  
rownames(df\_fit\_results) <- c("PayCheck Inc.", "Paypal", "Walmart", "S&P 500")  
  
df\_fit\_results

## BIC AIC AICc Model  
## PayCheck Inc. -3045.397 -3049.616 -3049.608 ARIMA(0,0,0)  
## Paypal -2530.806 -2543.462 -2543.414 ARIMA(1,0,1) = ARMA(1,1)  
## Walmart -2969.089 -2973.308 -2973.3 ARIMA(0,0,0)  
## S&P 500 -2756.626 -2760.845 -2760.837 ARIMA(0,0,0)

**Table 2:**

Table **2** allow us to observe that only the Paypal securities needed a greater order to find the best ARIMA model.

# PayCheck Inc. Training Model Observations

pdf(file = paste(folder,"Figure7-PayCheck Model Results.pdf"),height=7,width=8)  
#  
par(mfrow=c(2,2))  
acf(df\_log\_ret[,"PAYX"], main="PayCheck Inc.", na.action = na.pass)  
acf(fitPayx$residual, main = "PayCheck Inc. Residuals", na.action = na.pass)  
qqnorm(fitPayx$resid, datax = T, main="PayCheck Inc. Residuals QQNorm")   
qqline(fitPayx$resid, datax = T)  
plot(ts(fitPayx$resid),ylab="Residual",main="PayCheck Inc. Residuals Time-Series")  
#  
graphics.off()  
  
par(mfrow=c(2,2))  
  
acf(df\_log\_ret[,"PAYX"], main="PayCheck Inc.", na.action = na.pass)  
acf(fitPayx$residual, main = "PayCheck Inc. Residuals", na.action = na.pass)  
qqnorm(fitPayx$resid, datax = T, main="PayCheck Inc. Residuals QQNorm")   
qqline(fitPayx$resid, datax = T)  
plot(ts(fitPayx$resid),ylab="Residual",main="PayCheck Inc. Residuals Time-Series")

**Figure 7:** PayCheck Training Model Validation

PayCheck Inc. ACF plots presented in figure **7** confirms that the residuals follow the same autocorrelation than the original dataset. This similarity makes the ARIMA(0,0,0) a good model to be used for forecasting.

The QQnorm plot of PayCheck Inc. reiduals present heavy tails, a common characteristic of t-distributions. This behavior doesn’t align with the expected normal distribution for residuals.

The time-series plot of PayCheck Inc. residuals shows some volatility; in particular around the 50th observation, and between the 250th and 300th time observation.

pdf(file = paste(folder,"Figure8-Paypal Model Results.pdf"),height=7,width=8)  
#  
par(mfrow=c(2,2))  
acf(df\_log\_ret[,"PYPL"], main="Paypal", na.action = na.pass)  
acf(fitPypl$residual, main = "Paypal Residuals", na.action = na.pass)  
qqnorm(fitPypl$resid, datax = T, main="Paypal Residuals QQNorm")   
qqline(fitPypl$resid, datax = T)  
plot(ts(fitPypl$resid),ylab="Residual",main="Paypal Residuals Time-Series")  
#  
graphics.off()  
  
par(mfrow=c(2,2))  
  
acf(df\_log\_ret[,"PYPL"], main="Paypal", na.action = na.pass)  
acf(fitPypl$residual, main = "Paypal Residuals", na.action = na.pass)  
qqnorm(fitPypl$resid, datax = T, main="Paypal Residuals QQNorm")   
qqline(fitPypl$resid, datax = T)  
plot(ts(fitPypl$resid),ylab="Residual",main="Paypal Residuals Time-Series")

**Figure 8:** Paypal Training Model Validation

Paypal ACF plots presented in figure **8** confirms that the residuals follow the same autocorrelation than the original dataset; in particular at lag 20 both ACF plots present an autocorrelation that crosses the lower test bound. This similarity makes the ARIMA(1,0,1) {i.e., ARMA(1,1)} a good model to be used for forecasting.

The QQnorm plot of Paypal residuals present heavy tails, a common characteristic of t-distributions. This behavior doesn’t align with the expected normal distribution for residuals.

The time-series plot of Paypal residuals shows some volatility; in particular around the 50th observation, between the 200th and 300th time observation, and between 450th and 500th time observation.

pdf(file = paste(folder,"Figure9-Walmart Model Results.pdf"),height=7,width=8)  
#  
par(mfrow=c(2,2))  
acf(df\_log\_ret[,"WMT"], main="Walmart", na.action = na.pass)  
acf(fitWmt$residual, main = "Walmart Residuals", na.action = na.pass)  
qqnorm(fitWmt$resid, datax = T, main="Walmart Residuals QQNorm")   
qqline(fitWmt$resid, datax = T)  
plot(ts(fitWmt$resid),ylab="Residual",main="Walmart Residuals Time-Series")  
#  
graphics.off()  
  
par(mfrow=c(2,2))  
  
acf(df\_log\_ret[,"WMT"], main="Walmart", na.action = na.pass)  
acf(fitWmt$residual, main = "Walmart Residuals", na.action = na.pass)  
qqnorm(fitWmt$resid, datax = T, main="Walmart Residuals QQNorm")   
qqline(fitWmt$resid, datax = T)  
plot(ts(fitWmt$resid),ylab="Residual",main="Walmart Residuals Time-Series")

**Figure 9:** Walmart Training Model Validation

Walmart ACF plots presented in figure **9** confirms that the residuals follow the same autocorrelation than the original dataset. This similarity makes the ARIMA(0,0,0) a good model to be used for forecasting.

The QQnorm plot of Walmart residuals present heavy tails, a common characteristic of t-distributions. This behavior doesn’t align with the expected normal distribution for residuals.

The time-series plot for Walmart residuals shows some volatility; in particular between 50th and 100th time observations, and close to the 200th time observation.

pdf(file = paste(folder,"Figure10-SP500 Model Results.pdf"),height=7,width=8)  
#  
par(mfrow=c(2,2))  
acf(df\_log\_ret[,"SP"], main="S&P 500", na.action = na.pass)  
acf(fitSp$residual, main = "S&P 500 Residuals", na.action = na.pass)  
qqnorm(fitSp$resid, datax = T, main="S&P 500 Residuals QQNorm")   
qqline(fitSp$resid, datax = T)  
plot(ts(fitSp$resid),ylab="Residual",main="S&P 500 Residuals Time-Series")  
#  
graphics.off()  
  
par(mfrow=c(2,2))  
  
acf(df\_log\_ret[,"SP"], main="S&P 500", na.action = na.pass)  
acf(fitSp$residual, main = "S&P 500 Residuals", na.action = na.pass)  
qqnorm(fitSp$resid, datax = T, main="S&P 500 Residuals QQNorm")   
qqline(fitSp$resid, datax = T)  
plot(ts(fitSp$resid),ylab="Residual",main="S&P 500 Residuals Time-Series")

**Figure 10:** S&P 500 Training Model Validation

S&P 500 ACF plots presented in figure **10** confirms that the residuals follow the same autocorrelation than the original dataset; in particular at lag 12 both ACF plots present an autocorrelation that crosses the lower test bound. This similarity makes the ARIMA(0,0,0) a good model to be used for forecasting.

The QQnorm plot of S&P 500 residuals present heavy tails, a common characteristic of t-distributions. This behavior doesn’t align with the expected normal distribution for residuals.

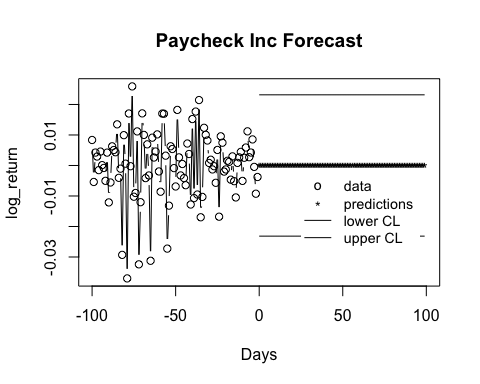
The time-series plot of S&P 500 residuals shows some volatility; in particular around the 250th observation, and 500th time observation.

pdf(file = paste(folder,"Figure11-Securities Residuals QQ Plot.pdf"),height=7,width=8)  
#  
par(mfrow=c(2,2))  
qqnorm(fitPayx$resid, datax = T, main="PayCheck Inc. Residuals")   
qqline(fitPayx$resid, datax = T)  
qqnorm(fitPypl$resid, datax = T, main="Paypal Residuals")   
qqline(fitPypl$resid, datax = T)  
qqnorm(fitWmt$resid, datax = T, main="Walmart Residuals")   
qqline(fitWmt$resid, datax = T)  
qqnorm(fitSp$resid, datax = T, main="S&P 500 Residuals")   
qqline(fitSp$resid, datax = T)  
#  
graphics.off()  
  
par(mfrow=c(2,2))  
qqnorm(fitPayx$resid, datax = T, main="PayCheck Inc. Residuals")   
qqline(fitPayx$resid, datax = T)  
qqnorm(fitPypl$resid, datax = T, main="Paypal Residuals")   
qqline(fitPypl$resid, datax = T)  
qqnorm(fitWmt$resid, datax = T, main="Walmart Residuals")   
qqline(fitWmt$resid, datax = T)  
qqnorm(fitSp$resid, datax = T, main="S&P 500 Residuals")   
qqline(fitSp$resid, datax = T)

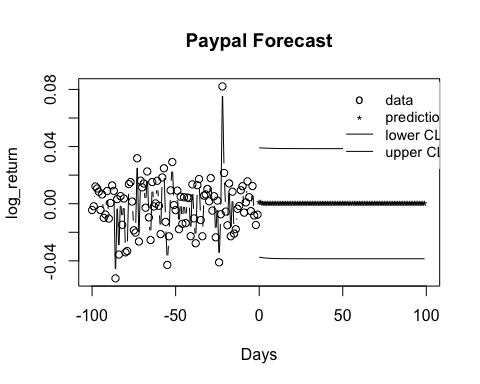
**Figure 11:** QQnorm plots for PayCheck Inc., Paypal, Walmart and S&P 500

Figure **11** allow us to observe heavy tails for all securities. However, S&P 500 seems to show the least amount of tail.

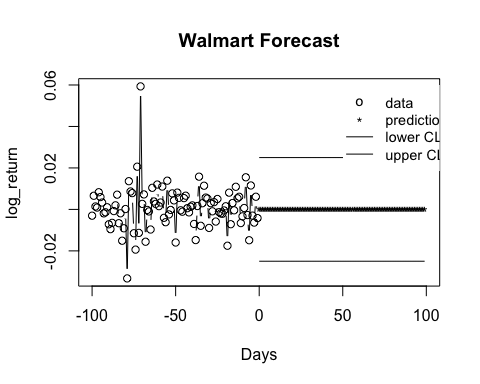
pred.payx <- predict(fitPayx, n.ahead = 100, se.fit = TRUE)  
pred.pypl <- predict(fitPypl, n.ahead = 100, se.fit = TRUE)  
pred.wmt <- predict(fitWmt, n.ahead = 100, se.fit = TRUE)  
pred.sp <- predict(fitSp, n.ahead = 100, se.fit = TRUE)  
  
t1 = (503-99):503  
t2 = 504:(504+49+50)  
day\_range = seq(-100,100,1)  
  
plot(day\_range[1:100],df\_log\_ret[t1,"PAYX"], type = "b",   
 main = "Paycheck Inc Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.payx$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.payx$pred - 2\*pred.payx$se)  
lines(day\_range[101:200], pred.payx$pred + 2\*pred.payx$se)  
legend(25,-0.001,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))



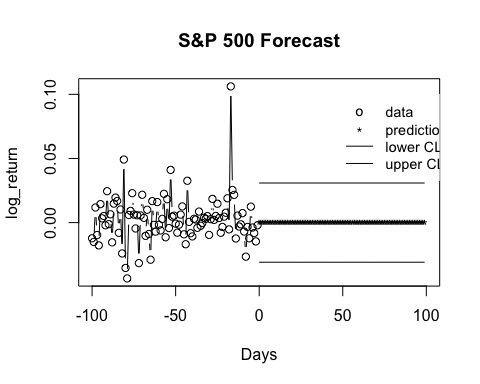
plot(day\_range[1:100],df\_log\_ret[t1,"PYPL"], type = "b",   
 main = "Paypal Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.pypl$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.pypl$pred - 2\*pred.pypl$se)  
lines(day\_range[101:200], pred.pypl$pred + 2\*pred.pypl$se)  
legend(50,0.085,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))



plot(day\_range[1:100],df\_log\_ret[t1,"WMT"], type = "b",   
 main = "Walmart Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.wmt$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.wmt$pred - 2\*pred.wmt$se)  
lines(day\_range[101:200], pred.wmt$pred + 2\*pred.wmt$se)  
legend(50,0.06,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))



plot(day\_range[1:100],df\_log\_ret[t1,"SP"], type = "b",   
 main = "S&P 500 Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.sp$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.sp$pred - 2\*pred.sp$se)  
lines(day\_range[101:200], pred.sp$pred + 2\*pred.sp$se)  
legend(50,0.1,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))



pdf(file = paste(folder,"Figure12-PayCheck Inc., Paypal, Walmart, and S&P 500.pdf"),height=7,width=8)  
par(mfrow=c(2,2))  
#  
plot(day\_range[1:100],df\_log\_ret[t1,"PAYX"], type = "b",   
 main = "Paycheck Inc Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.payx$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.payx$pred - 2\*pred.payx$se)  
lines(day\_range[101:200], pred.payx$pred + 2\*pred.payx$se)  
legend(25,0.001,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))  
#  
plot(day\_range[1:100],df\_log\_ret[t1,"PYPL"], type = "b",   
 main = "Paypal Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.pypl$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.pypl$pred - 2\*pred.pypl$se)  
lines(day\_range[101:200], pred.pypl$pred + 2\*pred.pypl$se)  
legend(25,0.09,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))  
#  
plot(day\_range[1:100],df\_log\_ret[t1,"WMT"], type = "b",   
 main = "Walmart Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.wmt$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.wmt$pred - 2\*pred.wmt$se)  
lines(day\_range[101:200], pred.wmt$pred + 2\*pred.wmt$se)  
legend(25,0.06,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))  
#  
plot(day\_range[1:100],df\_log\_ret[t1,"SP"], type = "b",   
 main = "S&P 500 Forecast", ylab = "log\_return", xlab = "Days",   
 xlim = c(-100,100))  
points(day\_range[101:200], pred.sp$pred,type="p",pch="\*")  
lines(day\_range[101:200], pred.sp$pred - 2\*pred.sp$se)  
lines(day\_range[101:200], pred.sp$pred + 2\*pred.sp$se)  
legend(25,0.1,c("data","predictions","lower CL","upper CL"),cex=0.9,  
 box.lty=0,pch=c("o","\*",NA,NA),lty=c(NA,NA,1,1))  
#  
graphics.off()