

Hyman-Alex_Project3

Alex Hyman, Matt LaFlair, Sasha Singh

8/27/2018

Introduction

Trade growth is influenced by the world business cycle and is very volatile and unpredictable. Keeping an adequate supply of ships at all times is essential for the free flow of world trade. Given the complexity of the cargo flows to be transported, this difficult task is tightly controlled by market forces.

Although freight rates are highly volatile, with extremes in both directions, on average transport costs fell by 80% in real terms during the second half of the 20th century. This demonstrates the long-term cost effectiveness of the tramp shipping business.

Tramp shipping revenues are determined competitively in the international market place, generally through the well developed network of shipbrokers and agents to transact the business. However, the precise nature of the process differs for the bulk, and specialized segments

Question 1: Important Data

The key business questions we are going to address to help influence the freight-forwarder's decision are:

1. How would the performance of these commodities affect the size and timing of shipping arrangements?

The performance of these commodities would certainly affect the size and timing of shipping arrangements. Since tramp ships trade on the spot market with no fixed schedule or itinerary/ports-of-call, the shipping arrangement is tremendously affected, especially since all three metals are being shipped from various locations in different timezones. The size on the other hand, can be negatively or positively affected by the demand of the metals. If there isn't a demand for these metals in the market, there certainly wouldn't be a need to purchase these metals in bulk. While the term buy low, sell high is highly recommended by investors, it would not make sense in this case.

2. How would the value of new shipping arrangements affect the value of our business with our current Customers?

The value of this new shipping arrangement will affect the value of our business with our current customers (ship owners, manufacturers, traders) negatively because the ship owners would have to ship more of the product, manufacturers would have to make more of the product, and the traders would not be able to purchase the product when they want. Prior to the use of the tramp trade, these customers received their metals in a timely scheduled manner, since with the new arrangement, these metals are received at random times, this causes our loyal customers to find our company as unreliable.

3. How would we manage the allocation of existing resources given we have just landed in this new market?

To manage the allocation of existing resources given we have just landed in this new market, we would have to consider the capital the ship owner pays and the operating costs of the vessel. All in all, the volatility is rarely constant and often has a structure (mean reversion) and is dependent on the past; where extreme events are likely to happen with other extreme events.

Particular factors we can consider are the various exchange rates and customer stock prices by comparing volatility and correlation.

```
#Removes all objects in this environment to blank out for the dashboard  
rm(list = ls())  
#Reading in the data
```

```

data <- na.omit(read.csv(url("https://turing.manhattan.edu/~wfoote01/finalytics/data/metaldata.csv"), h
#Applies the log difference to the numeric columns in the data matrix and multiplies by 100 to get perc
data.r <- apply(log(data[,-1]), 2, diff) * 100
#Finding the magnitude of the percent change with absolute value
size <- na.omit(abs(data.r))

#adds .size to the end of the column
colnames(size) <- paste(colnames(size), ".size", sep = "")
#If the returns is positive, make a new matrix have a 1 in that place
#If the returns is negative, put a -1 in that place, otherwise a 0
direction <- ifelse(data.r > 0, 1, ifelse(data.r < 0, -1, 0))
#Adding .dir to the end of the column names in the direction data frame
colnames(direction) <- paste(colnames(direction), ".dir", sep = "")
#Making a vector of the dates
dates <- as.Date(data[-1,1], "%m/%d/%Y")
#Making dates characters
dates.chr <- as.character(dates)
#Combining all columns to make a values data frame
values <- cbind(data.r, size, direction)
#Creating tidy data frames
data.df <- data.frame(dates = dates, returns = data.r, size = size, direction = direction)
data.df.nd <- data.frame(dates = dates.chr, returns = data.r, size = size, direction = direction, stringr

#Creating a time series object
data.xts <- na.omit(as.xts(values, dates))
#Making a zoo object
data.zr <- as.zooreg(data.xts)
returns <- data.xts

```

Negative returns are more likely to occur than positive returns (left skew). Additionally, past shocks persist and may or may not dampen (rock in a pool).

Question 2: Stylized Facts

We first loaded the time series price data into the variable labeled “data “. This “data” will subsequently be manipulated later on for further analysis. We will equate this “data” variable to “prices”. To properly understand the data set, its volatilities and stylized facts of the market we will look at the percent change using the log function to normalize the data set to perform accurate analysis. Next we view the volatility of the data by viewing its % change absolute value, while also creating an array of the direction of the change. Finally we will set up tables and variables of the data set to be used below in further exploratory analysis. All of this data will allow us to understand the relationship between markets and help in decision making when it comes to business strategy.

Below we view the correlations between the nickel, copper and aluminum markets. We will view the correlations between these markets over a specified 90 day period using the roll_apply function. This will calculate the correlations of the returns as well as its volatility.

```

#Function to use in roll apply
corr_rolling <- function(x){
  #Number of columns in x
  dim <- ncol(x)
  #calculate the correlation of r, nut only include the the
  #lower triangle of the correlation matrix because the resulting
  #matrix will be c x c

```

```

corr_r <- cor(x)[lower.tri(dim), diag = FALSE]]
#Return all teh correlations
return(corr_r)
}
#Function to use in the rollapply function
vol_rolling <- function(x){
  #Importing the library for matrix statistics
  library(matrixStats)
  #calculate the standard deviation in the column
  vol_r <- colSds(x)
  #return the SDs
  return(vol_r)
}
#making a matrix of only the returns (no dates)
ALL.r <- data.xts[, 1:3]
#Creating a window of 90 days
window <- 90
#apply the rolling correlation function on the returns data
corr_r <- rollapply(ALL.r, width = window, corr_rolling, align = "right", by.column = FALSE)
#giving column names to the correlation matrix that has the two metals being measured
colnames(corr_r) <- c("nickel.copper", "nickel.aluminium", "copper.aluminium")
#apply the rolling correlation function on the returns data
vol_r <- rollapply(ALL.r, width = window, vol_rolling, align = "right", by.column = FALSE)
#giving the column names to specify they are volatilities being measured
colnames(vol_r) <- c("nickel.vol", "copper.vol", "aluminium.vol")
#Creating a vector that has the dates for all rows
year <- format(index(corr_r), "%Y")
#Creates a dataframe that has all the rolling calculations and all the raw data
#and the year of that piece of data
r_corr_vol <- merge(ALL.r, corr_r, vol_r, year)
#importing library for quantile regression
library(quantreg)

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##      backsolve

#Creating a vector of quantiles from 0.05 to 0.95 in increments of 0.05
taus <- seq(0.05, 0.95, 0.05)
#Fitting a quantile regression of nickel on copper
fit.rq.nickel.copper <- rq(log(nickel.copper) ~
  log(copper.vol), tau = taus, data = r_corr_vol)

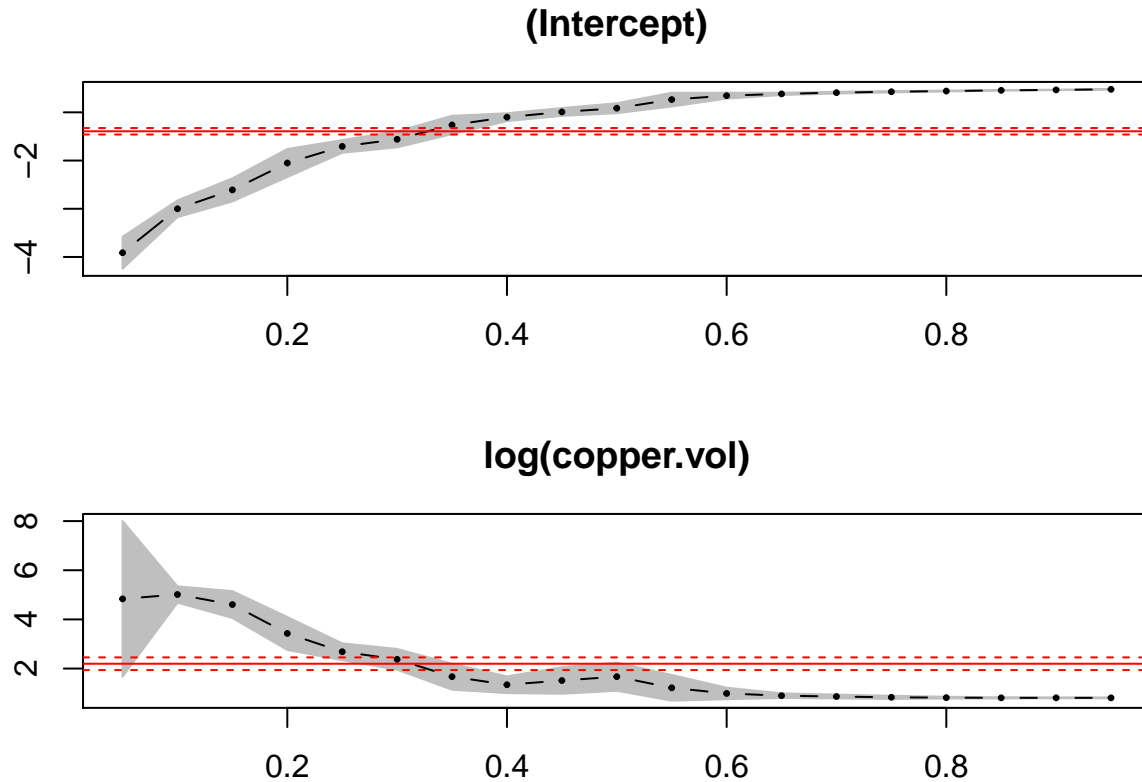
## Warning in log(nickel.copper): NaNs produced
#Fitting linear regression on copper and nickel
fit.lm.nickel.copper <- lm(log(nickel.copper) ~
  log(copper.vol), data = r_corr_vol)

## Warning in log(nickel.copper): NaNs produced

```

```
#Creating a summary of the quantile regression fit
ni.cu.summary <- summary(fit.rq.nickel.copper,
  se = "boot")
#plotting the summary/quantile regression coefficients
plot(ni.cu.summary)
```

```
## Warning in log(nickel.copper): NaNs produced
```



In the above chunk we perform the quantile regression with the ‘rq’ command of the nickel/copper market correlations and the copper market volatility, setting it as the variable “fit.rq.nickel.copper”. The “lm” command performs a simple regression of the same data set, with confidence intervals, as the quantile regression, displaying in red. ‘taus’ is saved to specify the quantiles in which we will run the regressions which run for every quantile from .05 to .95, in .05 increments. The quantile regression is run with the market correlations defining dependent variable and copper market volatility defined as the independent variable.

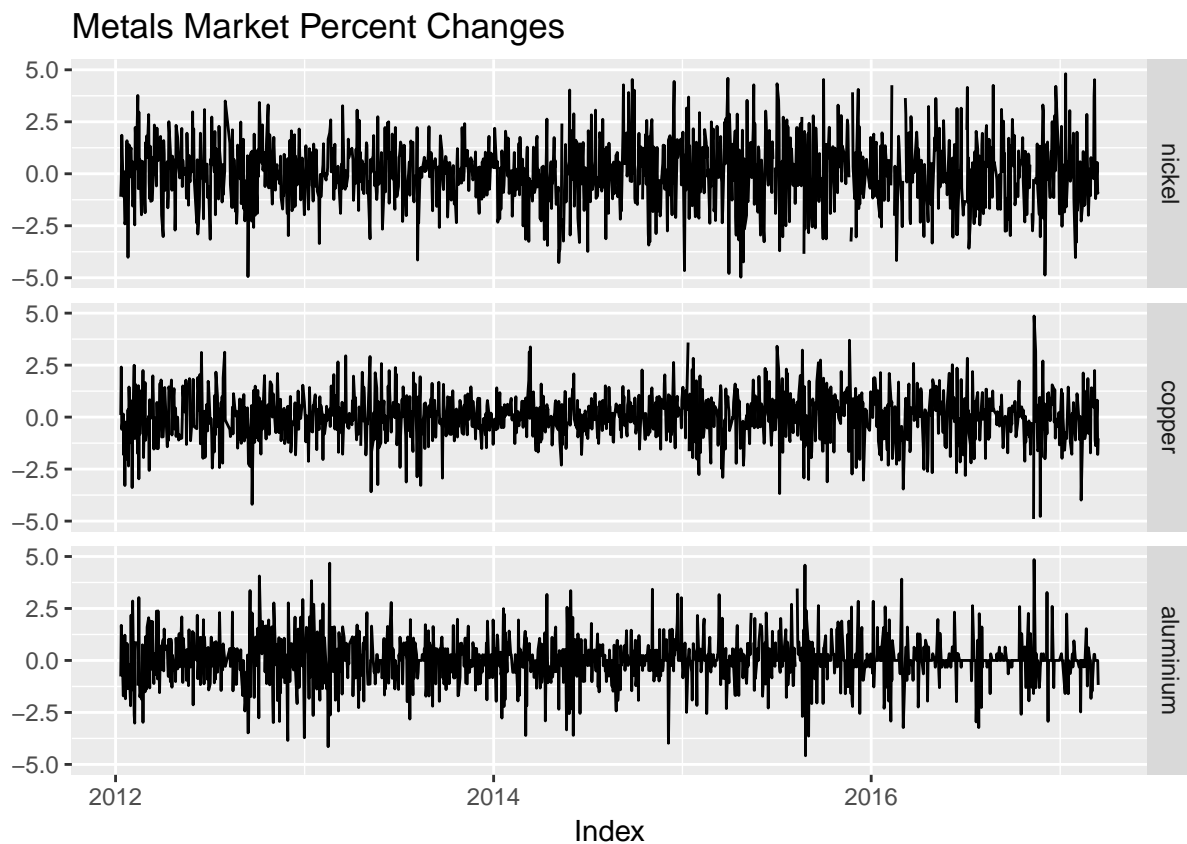
The first plot displays the intercept (y axis) of the accompanying quantile, presented on the x axis. The bottom plot details the similar quantile on the x axis, but the y displays the associated coefficients. The quantile coefficients (slopes) tell us that for every quantile increase, .05 to 0.1 and so on, what the change in the nickel/copper market correlation is due to copper market volatility. From a quick look you can easily tell that a linear regression is not sufficient in providing confidence in predicting copper/nickel market movement correlations given the volatility of copper.

However, one of the major points that jumps out is the leveling off of the coefficients as we near the higher quantiles, meaning that as we look at more and more of the data the the relationship between the copper/nickel correlation and the copper volatility becomes less and less nearing zero.

The below chunks plot the visualization of the log differ returns and the volatility size of the market for copper, nickel and aluminum.

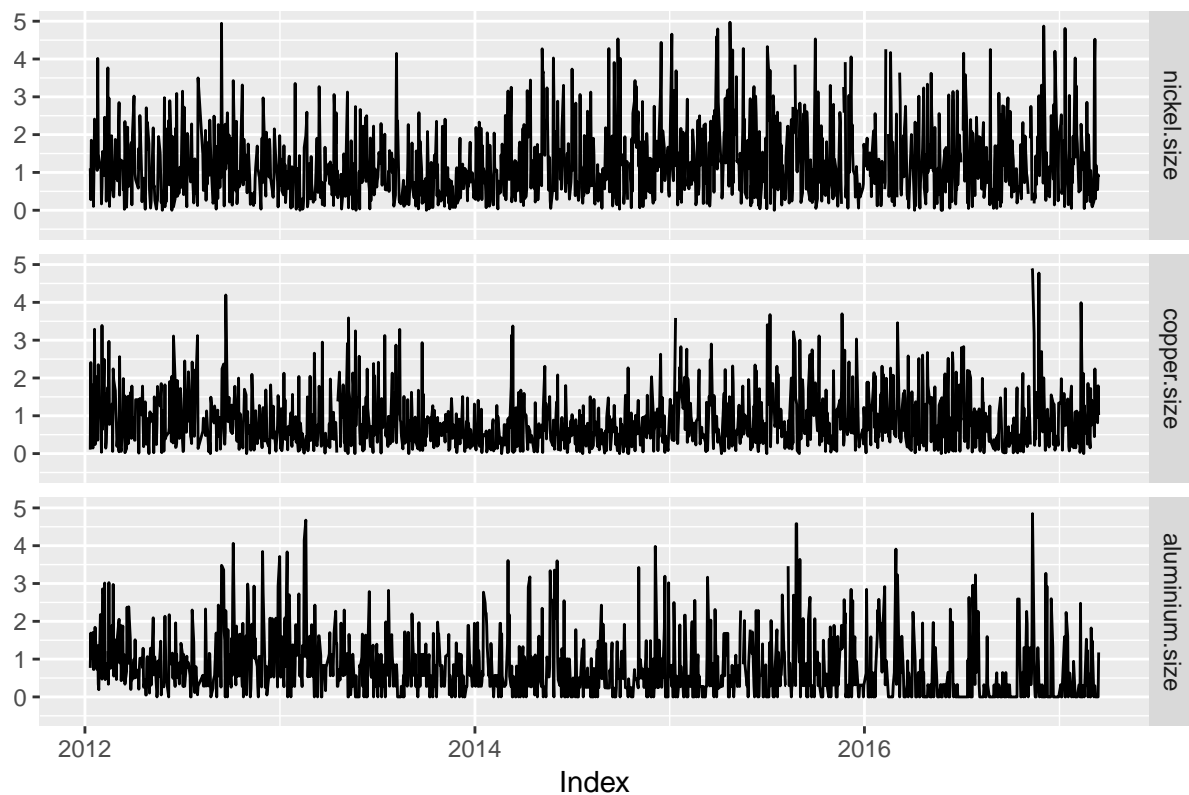
```
#Title for the plot
title.chg <- "Metals Market Percent Changes"
```

```
#Plotting the returns for copper, nickel, and aluminum from -5 to 5
autoplot.zoo(data.xts[, 1:3]) + ggtitle(title.chg) +
  ylim(-5, 5)
```



```
#plotting the magnitude of change for the metals
title2 <- "Metals Market Magnitude of Change"
autoplot.zoo(data.xts[, 4:6]) + ggtitle(title2) +
  ylim(-.5, 5)
```

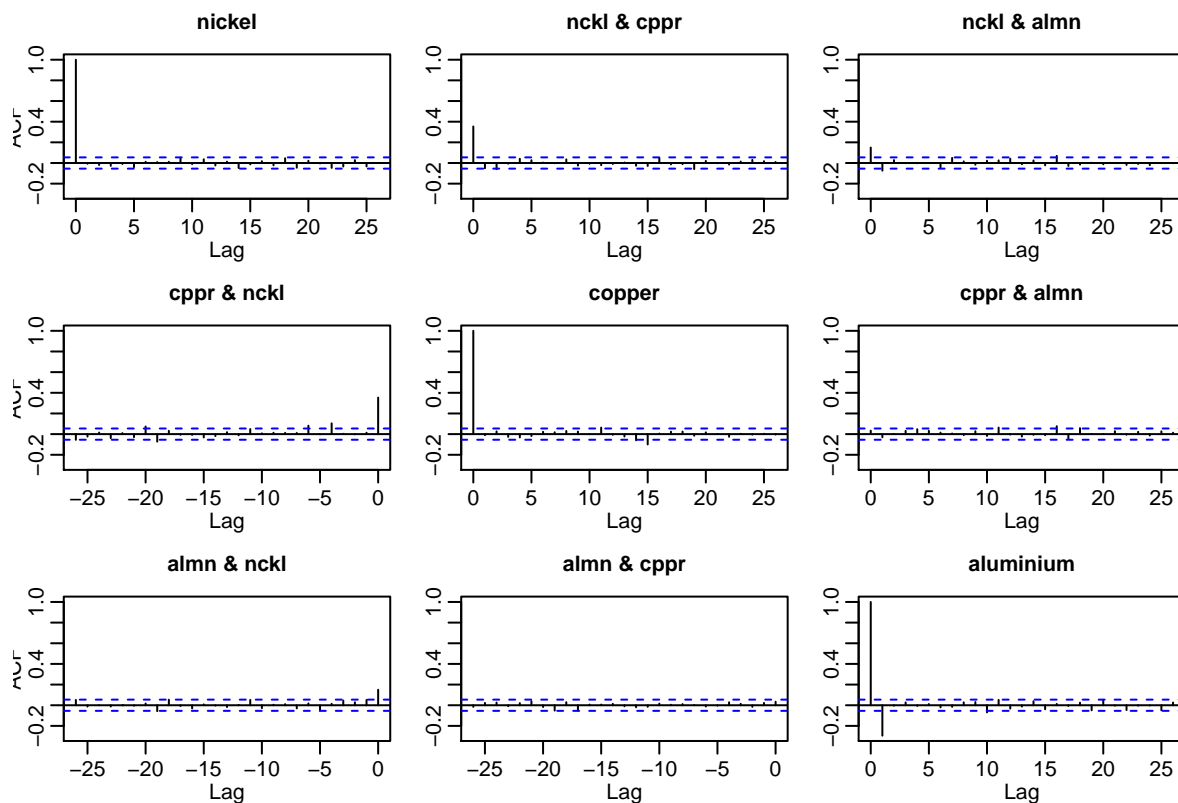
Metals Market Magnitude of Change



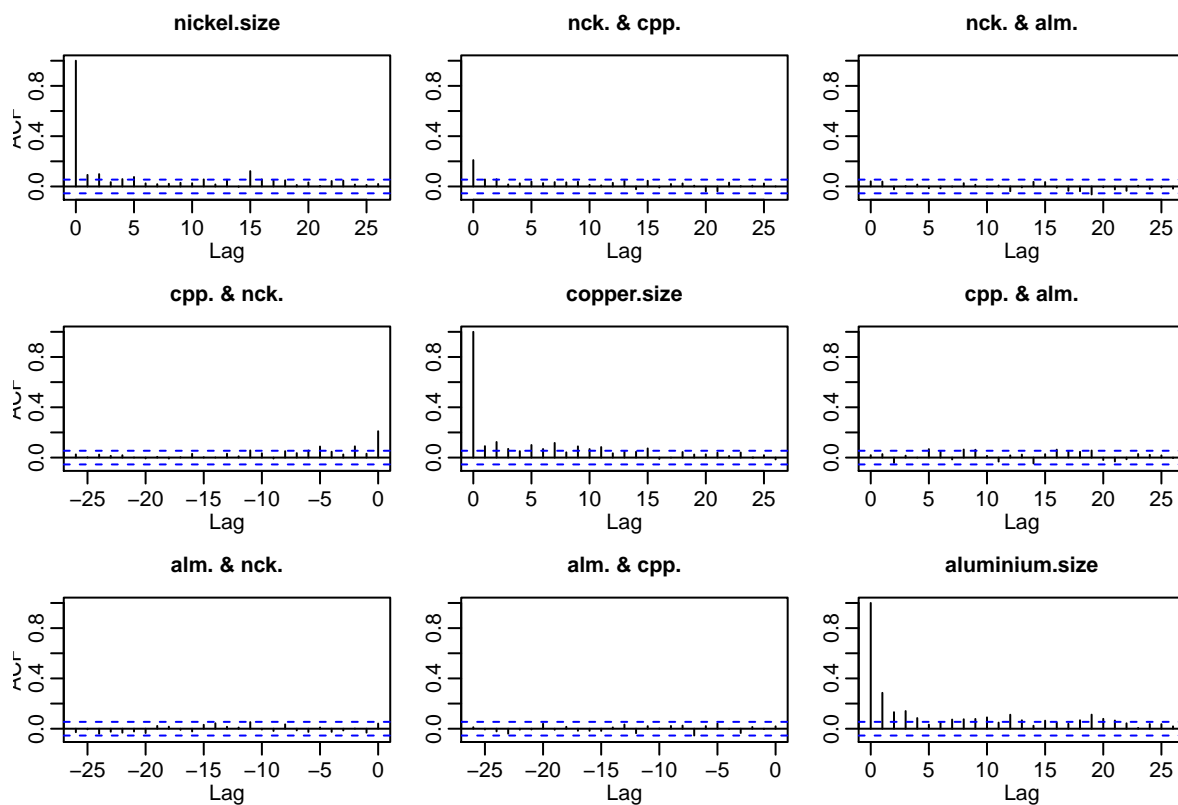
It is very apparent that there is significant movement throughout the markets in the study period. Visual inspection of the potential relationships between the markets show that nickel and copper tend to move in a similar pattern, with nickel showing significant more in its magnitude of returns. This is supported when viewing its volatility sizes, compared to copper returns.

The autocorrelation function below allows us to view a time series against its past to determine if persistence exists within the market, or rather if the markets historical prices affect the current price or the prices of other time series.

```
#Autocorrelation in returns for the metals  
acf(coredata(data.xts[, 1:3])) # returns
```



```
#Autocorrelation for magnitude of returns
acf(coredata(data.xts[, 4:6])) # sizes
```



In this case we view the returns of copper, nickel, and aluminum against one another to see if a relationship exists between today's price and that of the previous market price, either in its own market or the other two. Using the confidence interval as a judge of the whether that particular data value is significant enough we see that today's nickel has almost no significant correlation to past values of nickel. However nickel does look to have slight correlation to both the last two days prices of copper. Nickel does also show to have correlation to the aluminum market price of the 2 and 16 day historical price. But most importantly it should be noted that today's nickel price has a significant correlation coefficient of ~ 0.36 to today's price of copper and 0.17 to aluminum.

Copper autocorrelation shows some relationship with the 4,5, and 20th lag days in the nickel market and a slight correlation to the 15,16,17th day in the aluminum market. Copper does show a slight correlation and persistence in its own market for the 14th and 15th previous trading day. Finally, aluminum shows no real relationship to the previous days in the copper and nickel market while shown a strong negative relationship to its previous market return.

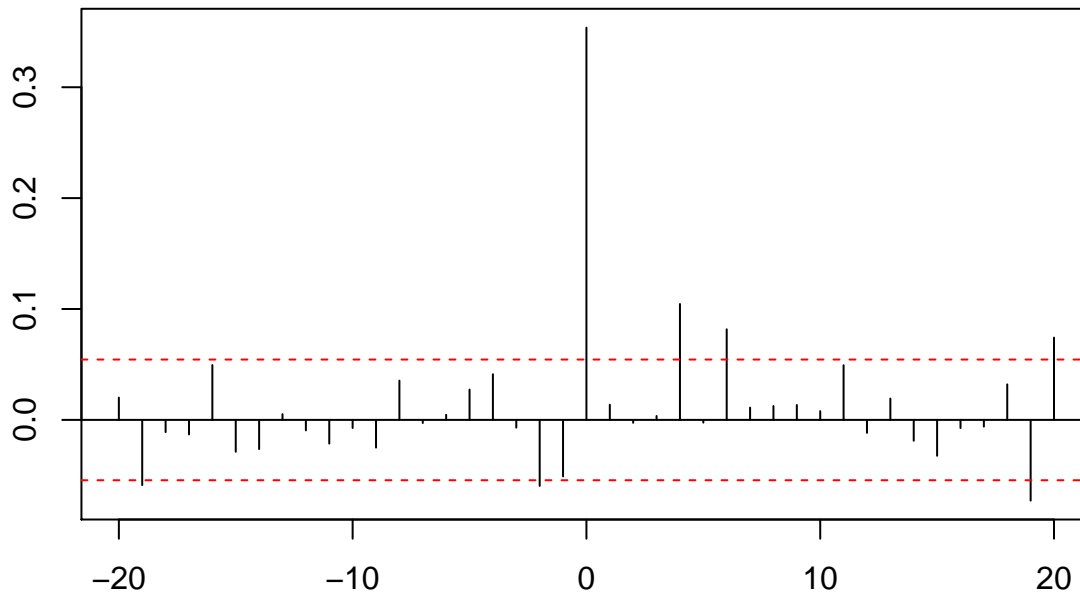
Looking at the volatility of returns, which is depicted using the acf command once again, but using the magnitude of the volatilities instead yield interesting results. The nickel volatility sizes shows a notable relationship with 1,2,4,5 day lags and a significant relationship with its 15 day lag. This shows some persistence within the nickel volatility and some localized grouping of swings in the market price. Nickel does have a relationship with the 19 day aluminum market lag, but otherwise shows little affect to it or the copper market volatility. Copper and aluminum show strong volatility persistence within their respective markets, aluminum more so than copper. Aluminum show no volatility relationship to other markets while copper shows a significant relationship to the nickel 5 day lag volatility.

It is apparent that the Nickel and Copper markets, via visualization of the time series and the auto correlations, that a relationship exist while no really exist with the aluminum market.

The below chunk performs the cross correlation between the returns of nickel (one) and copper (two). It is helped to determine the whether one time series causes changes in another.

```
# making time series of nickel returns
one <- ts(data.df$returns.nickel)
# making time series of copper returns
two <- ts(data.df$returns.copper)
#Creating title for ccf plot
title.chg <- "Nickel vs. Copper"
#ccf for nickel and copper
ccf(one, two, main = title.chg, lag.max = 20,
     xlab = "", ylab = "", ci.col = "red")
```


Nickel vs. Copper

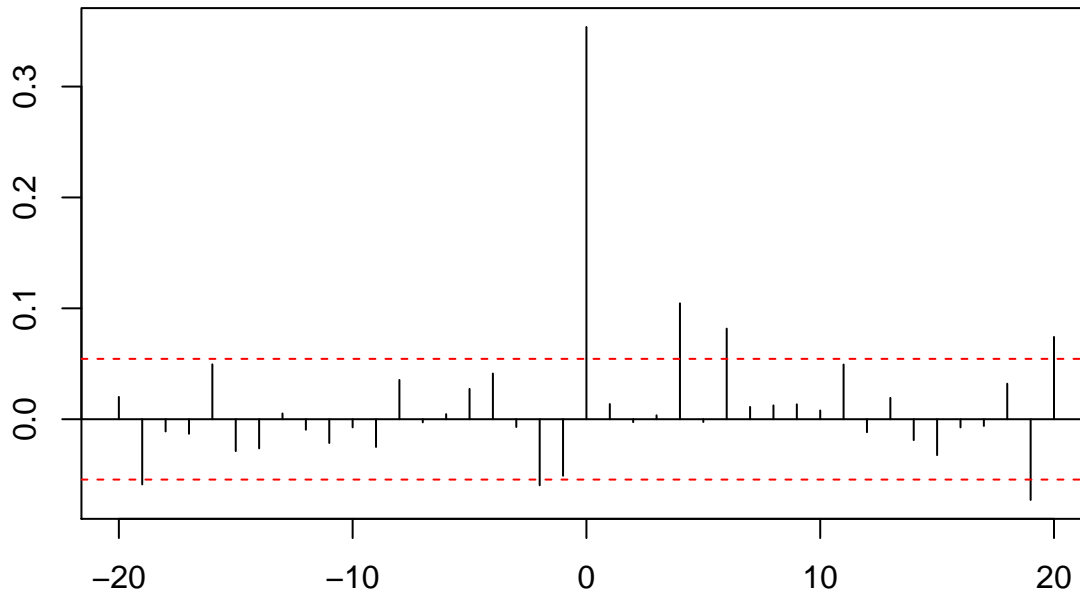


We have discussed this above by viewing the correlation of today's price versus the time series lag of itself or the other two commodities. Focusing in on the Nickel and copper markets, we can see that the same-day of market relationship exists of around ~0.36 coefficient. The lag of 4, 6, and 20 show significant as we originally discussed above.

The below chunk will creates a repeatable function that will allow us to create cross correlation of any univariate time series data with some inputs (one) for x and (two) for y.

```
#Creating function for repetition
#provide the two vectors, title, lag, and color of confidence interval
run_ccf <- function(one, two, main = title.chg,
  lag = 20, color = "red") {
  #Stop the function if the vectors are not equal length
  stopifnot(length(one) == length(two))
  #create one time series
  one <- ts(one)
  #creating other time series
  two <- ts(two)
  #Running the cross correlation function
  ccf(one, two, main = main, lag.max = lag,
    xlab = "", ylab = "", ci.col = color)
  # end run_ccf
}
title <- "nickel-copper"
run_ccf(one, two, main = title, lag = 20,
  color = "red")
```

nickel-copper

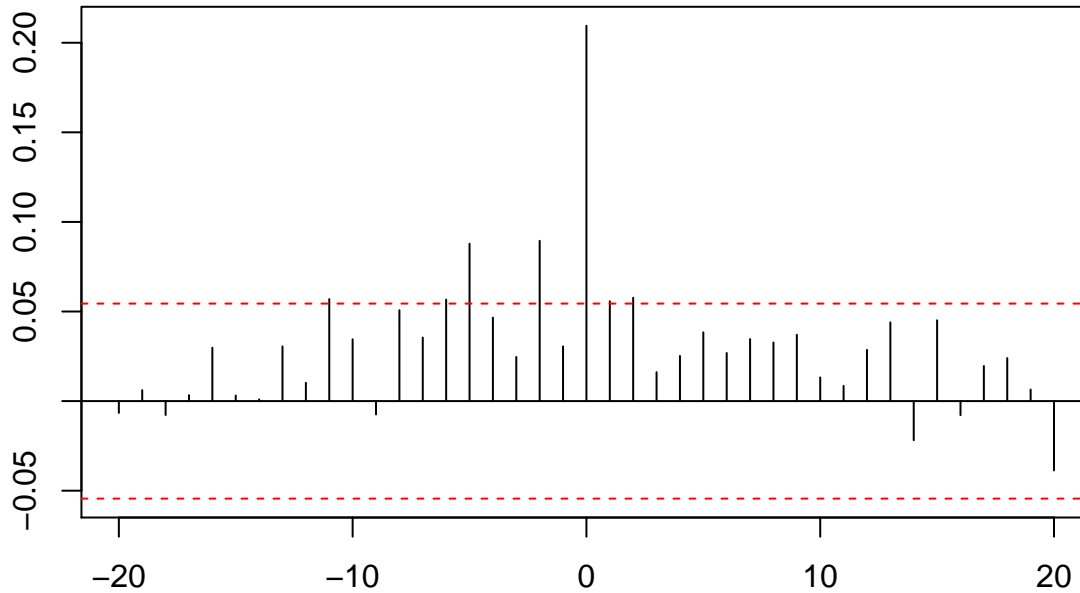


Similar to the previous cross correlations we are now looking at the volatility, or size of the returns between nickel (one) and copper (two). Again, looking at the volatilities we can see that the nickel day of, 2, 5, 11 day lags shows significant correlation while the intermediary days show some clumping of variability in returns, but below our confidence threshold. We can also see that the nickel market seems to lead the copper market volatility magnitude more with a positive correlation, one goes up the other follows suit.

The below chunk will create a function to display some statistics of our time series data, returns and its magnitudes that may help us create a better picture for our business decisions.

```
#Running the ccf function for volatilities  
# now for volatility (sizes)  
#  
one <- abs(data.zr[, 4])  
two <- abs(data.zr[, 5])  
title <- "Nickel-Copper: volatility"  
run_ccf(one, two, main = title, lag = 20,  
        color = "red")
```

Nickel–Copper: volatility



```
#Data Moments for nickel and
data_moments <- function(data) {
  library(moments)
  library(matrixStats)
  mean.r <- colMeans(data)
  median.r <- colMedians(data)
  sd.r <- colSds(data)
  IQR.r <- colIQRs(data)
  skewness.r <- skewness(data)
  kurtosis.r <- kurtosis(data)
  result <- data.frame(mean = mean.r,
    median = median.r, std_dev = sd.r,
    IQR = IQR.r, skewness = skewness.r,
    kurtosis = kurtosis.r)
  return(result)
}
# Run data moments for the returns and sizes of all metals
answer <- data_moments(data.xts[, 1:6])
```

```
##
## Attaching package: 'moments'
## The following objects are masked from 'package:timeDate':
##
##      kurtosis, skewness
```

```
# Build pretty table with 4 digits
answer <- round(answer, 4)
#Knitting the table
knitr::kable(answer)
```

	mean	median	std_dev	IQR	skewness	kurtosis
nickel	0.0489	0.0928	1.7072	2.0479	0.1932	5.2688

	mean	median	std_dev	IQR	skewness	kurtosis
copper	0.0200	0.0595	1.1815	1.3508	-0.2010	4.8894
aluminium	0.0149	0.0000	1.2080	1.0672	-0.1557	6.3417
nickel.size	1.2830	1.0257	1.1268	1.3496	1.9210	9.6851
copper.size	0.8830	0.6823	0.7849	0.9217	1.7713	7.8654
aluminium.size	0.8072	0.5510	0.8986	1.0136	1.8899	8.4006

```
#Average size in change of nickel
mean(data.xts[, 4])
```

```
## [1] 1.282969
```

The above compiled table shows us statistics related to the aluminum, copper and nickel markets. All three market returns and volatility magnitudes have a moderately high kurtosis, relative to the normal distribution of 3. This indicates that the tails are heavy, more so in the volatility magnitudes. The directional skewness and kurtosis reveal little information about our data and thus are not shown. However, it is worth noting that the mean copper direction yields that it tends to move in the positive direction ~5% of the time. The directional movement of the nickel market tends to further support our prior hypothesis of correlation existing between itself and the copper market as it too has a 4.5% favoring in the positive direction.

The skewness of the returns shows that they are relatively asymmetric, with copper and aluminum having a very slight skew in the negative direction. Which brings us to the volatility skewness; copper, nickel and aluminum volatility tend to have a significant movement in their respective markets but directionality is not known due to it being absolute values. The interquartile range, middle 50% of the returns data, shows higher returns for nickel than copper and aluminum. This is further supported by the average nickel returns which tend to be higher than that of copper and aluminum. Our original visual inspection of the time series data indicated higher volatility magnitudes for nickel returns is supported by higher (~40%) mean volatility magnitudes for nickel than the other commodities.

The provided stylized facts paint an interesting picture of the copper, aluminum and nickel markets. What is clear is that if one is looking to diversify investments in regards to metal commodities to eliminate systematic risks solely investing in the nickel and copper market is not wise. It is best to choose either nickel or copper, depending on your acceptable risk in regard to volatility magnitude and pair it with aluminum investments as there is little correlation between the nickel/copper and the aluminum markets.

Question 3: Required Capital

Because we are shipping metals to a spot market where prices can vary widely, we need to know how much capital we need to have covered in case of a terrible day at the market. In this instance, we want to know what is the maximum amount of cash we could lose if the returns are in the 95% of losses, meaning our losses only have a 5% chance of being worse than the historical data. The value that we find at the 95th percentile for losses will be considered our value at risk. To cover our potential losses, but not necessarily over-hedge, we need to find out what the average losses are for losses greater than our value at risk. This will be considered our expected shortfall, as well as our required capital. We would most likely want to be insured for the expected shortfall amount, just so we can be sure that we do not have any losses greater than we can stand.

First, we will examine nickel returns. Considering the historical returns for nickel in the London Stock Exchange, we will find at what returns resulted in the 95% of losses. To do this we will want to multiply our returns by -1 in order to reflect that we are plotting losses. We will then use the quantile function to find where in the distribution of historical losses are we only allowing a 5% chance of having a greater loss. This value will be stored as our value at risk. Then, we will find what is the mean loss in our historical distribution for losses greater than our value at risk. This will be our expected shortfall or the required capital. Finally, we will show the distribution of losses along with the value at risk and expected shortfall on a density function.

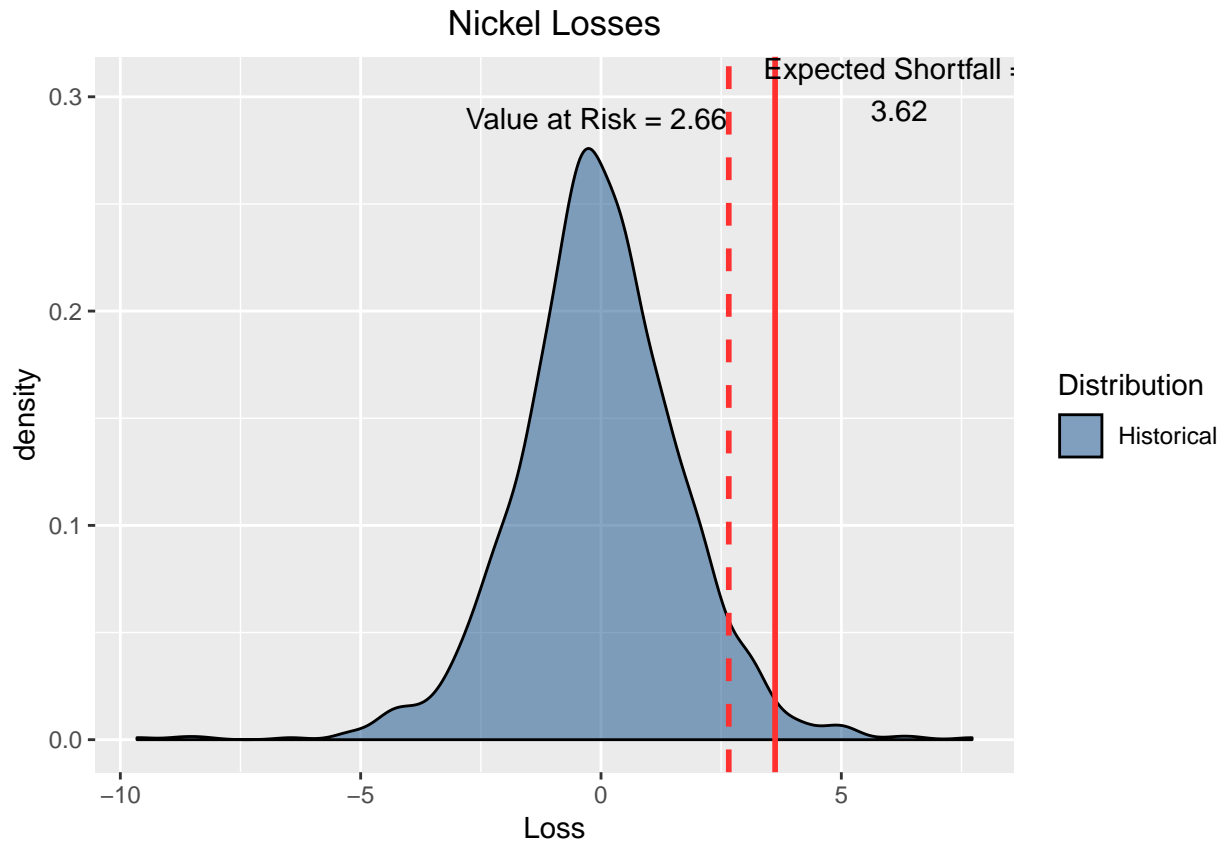
```

#Multiplying returns of nickel by negative 1 to get the losses
loss1 <- returns[, 1] * -1
#Naming the column
colnames(loss1) <- "Loss"
#Creating data frame with a column "Distribution" that states
#That this is the historical returns
loss1.df <- data.frame(Losses = loss1[,
  1], Distribution = rep("Historical",
  each = length(loss1)))
#95% confidence interval
alpha <- 0.95
# finding the value that is at the accepted risk in the returns
VaR.hist <- quantile(loss1, alpha)
#Creating text for the plot that will say what the Value at risk is
VaR.text <- paste("Value at Risk =",
  round(VaR.hist, 2))

# Determine the max y value of the
# density plot. This will be used to
# place the text above the plot
VaR.y <- max(density(loss1.df$Loss)$y)

# Expected Shortfall is mean of values greater than VaR
ES.hist <- mean(loss1[loss1 >
  VaR.hist])
#Creating text for the expected shortfall
ES.text <- paste("Expected Shortfall =\n",
  round(ES.hist, 2))
#Creating a ggplot object with the returns being x and fill being the Distribution
p <- ggplot(loss1.df, aes(x = Loss,
  fill = Distribution)) + geom_density(alpha = 0.5) + #Making it a density plot
  geom_vline(aes(xintercept = VaR.hist),#Adding vertical line at the VaR
  linetype = "dashed", size = 1,
  #Adding a vertical line at the expected shortfall
  color = "firebrick1") + geom_vline(aes(xintercept = ES.hist),
  size = 1, color = "firebrick1") +
  #adding the text to the plot
  annotate("text", x = VaR.hist - 2.75,
    y = VaR.y * 1.05, label = VaR.text) +
  annotate("text", x = 2.5 + ES.hist,
    y = VaR.y * 1.1, label = ES.text) +
  scale_fill_manual(values = "dodgerblue4") + ggtitle("Nickel Losses") +
  theme(plot.title = element_text(hjust = 0.5))
#Showing the plot
p

```



The above plot shows that 95% of the time, our percent loss for nickel is 2.66%, which will be our value at risk. The average of the losses greater than the value at risk was determined to be 3.62%. This value will be our expected shortfall, or the average of losses greater than our value at risk.

Next we will calculate the historical value at risk and expected shortfall for our current metals: copper and aluminum. Presumably these values had been calculated previously, and we are already hedging against these metals. The process that had been used to calculate the value at risk and expected shortfall for Nickel will be used for calculating the historical values for expected shortfall and value at risk for aluminum and copper.

```
#Multiplying returns of copper by negative 1 to get the losses
loss1 <- returns[, 2] * -1
#Naming the column
colnames(loss1) <- "Loss"
#Creating data frame with a column "Distribution" that states
#That this is the historical returns
loss1.df <- data.frame(Losses = loss1[,
  1], Distribution = rep("Historical",
    each = length(loss1)))
#95% confidence interval
alpha <- 0.95
# finding the value that is at the accepted risk in the returns
VaR.hist <- quantile(loss1, alpha)
#Creating text for the plot that will say what the Value at risk is
VaR.text <- paste("Value at Risk =",
  round(VaR.hist, 2))

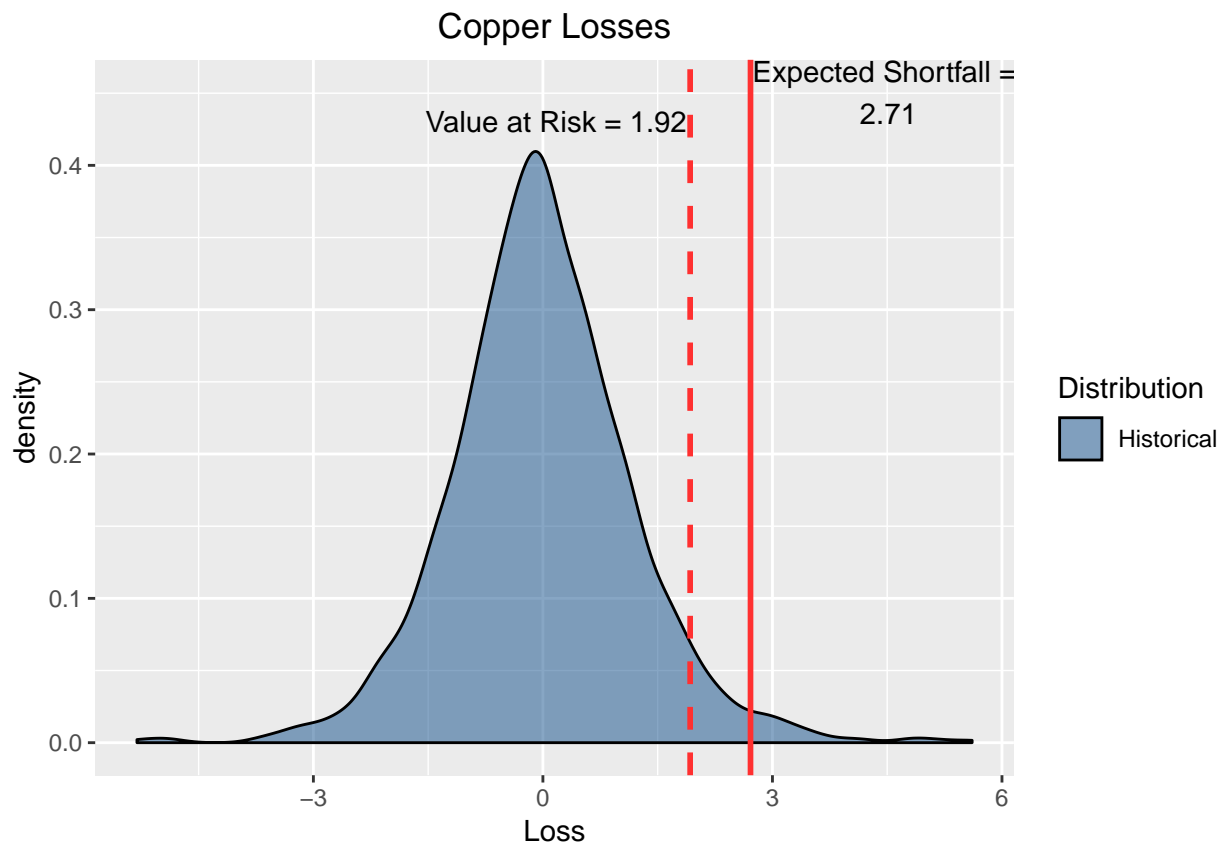
# Determine the max y value of the
# desity plot. This will be used to
```

```

# place the text above the plot
VaR.y <- max(density(loss1.df$Loss)$y)

# Expected Shortfall is mean of values greater than VaR
ES.hist <- mean(loss1[loss1 >
  VaR.hist])
#Creating text for the expected shortfall
ES.text <- paste("Expected Shortfall =\n",
  round(ES.hist, 2))
#Creating a ggplot object with the returns being x and fill being the Distribution
p <- ggplot(loss1.df, aes(x = Loss,
  fill = Distribution)) + geom_density(alpha = 0.5) + #Making it a density plot
  geom_vline(aes(xintercept = VaR.hist),#Adding vertical line at the VaR
    linetype = "dashed", size = 1,
    #Adding a vertical line at the expected shortfall
    color = "firebrick1") + geom_vline(aes(xintercept = ES.hist),
    size = 1, color = "firebrick1") +
  #adding the text to the plot
  annotate("text", x = VaR.hist - 1.75,
    y = VaR.y * 1.05, label = VaR.text) +
  annotate("text", x = ES.hist + 1.75,
    y = VaR.y * 1.1, label = ES.text) +
  scale_fill_manual(values = "dodgerblue4") + ggtitle("Copper Losses") +
  theme(plot.title = element_text(hjust = 0.5))
#Showing the plot
p

```

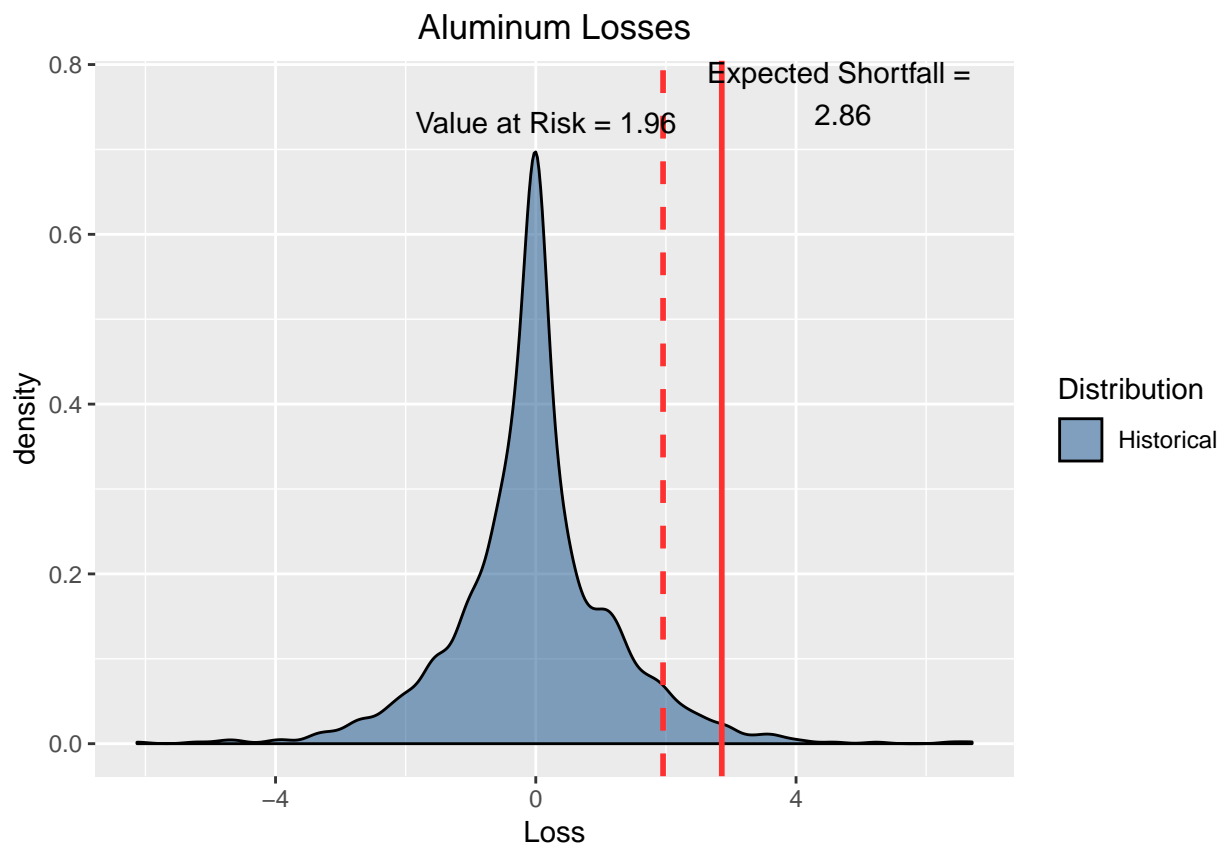


The losses for copper show that 95% of the time, losses will be less than 1.92%, and that the average of losses to copper greater than the value at risk is 2.71%. This distribution shows what was reflected in the data moment table: copper is less volatile than nickel, but the returns are more skewed towards losses.

```
#Multiplying returns of copper by negative 1 to get the losses
loss1 <- returns[, 3] * -1
#Naming the column
colnames(loss1) <- "Loss"
#Creating data frame with a column "Distribution" that states
#That this is the historical returns
loss1.df <- data.frame(Losses = loss1[,
  1], Distribution = rep("Historical",
  each = length(loss1)))
#95% confidence interval
alpha <- 0.95
# finding the value that is at the accepted risk in the returns
VaR.hist <- quantile(loss1, alpha)
#Creating text for the plot that will say what the Value at risk is
VaR.text <- paste("Value at Risk =",
  round(VaR.hist, 2))

# Determine the max y value of the
# desity plot. This will be used to
# place the text above the plot
VaR.y <- max(density(loss1.df$Loss)$y)

# Expected Shortfall is mean of values greater than VaR
ES.hist <- mean(loss1[loss1 >
  VaR.hist])
#Creating text for the expected shortfall
ES.text <- paste("Expected Shortfall =\n",
  round(ES.hist, 2))
#Creating a ggplot object with the returns being x and fill being the Distribution
p <- ggplot(loss1.df, aes(x = Loss,
  fill = Distribution)) + geom_density(alpha = 0.5) + #Making it a density plot
  geom_vline(aes(xintercept = VaR.hist),#Adding vertical line at the VaR
    linetype = "dashed", size = 1,
    #Adding a vertical line at the expected shortfall
    color = "firebrick1") + geom_vline(aes(xintercept = ES.hist),
    size = 1, color = "firebrick1") +
  #adding the text to the plot
  annotate("text", x = VaR.hist - 1.8,
    y = VaR.y * 1.05, label = VaR.text) +
  annotate("text", x = ES.hist + 1.8,
    y = VaR.y * 1.1, label = ES.text) +
  scale_fill_manual(values = "dodgerblue4") + ggtitle("Aluminum Losses") +
  theme(plot.title = element_text(hjust = 0.5))
#Showing the plot
p
```

The distribution of losses for aluminum show that there is a 95% chance that losses will be below 1.96%, which will be our value at risk. The average of losses greater than the value at risk is 2.86%, which is the value we will hedge against. Once again, aluminum's losses are likely to be less than those of nickel, and there appears to be a more narrow spread. However, due to the sharp peak of the curve, the tails could be extreme.

Next, we will use historical data to see, if we had 1/3 ton of each metal at the time we began collecting data, what would our daily losses look like if we always had an equal amount of each metal. Using this data, we could determine what our total portfolio's value at risk is, and how much capital is required for a ton of each metal together. We will determine the value of metal we started with by multiplying the mass of the metal by the prices on the first day. We will then multiply the metals by the returns historical daily returns, and sum the three metals losses for the day to see how much money was lost. These daily amounts of money lost will be plotted and used to give a dollar value to value at risk and expected shortfall. We will plan to hedge against the value of the expected shortfall for every ton of equally distributed metals on our ship.

```
#Getting the first days prices for the metals in our data
price.last <- as.numeric(tail(data[,
-1], n = 1))
# we have 1/3 ton of each metal
position.rf <- c(1/3, 1/3, 1/3)
#The price of the metal we started with is our position
#multiplied by the prices on the first day we have data
w <- position.rf * price.last
# Creating a matrix of our starting prices to use in
#the modeling of risk
weights.rf <- matrix(w, nrow = nrow(data.r),
ncol = ncol(data.r), byrow = TRUE)
#Creating a data frame that shows how much was lost, determined
#by the weights we started with and percent change in price we calculated
```

```

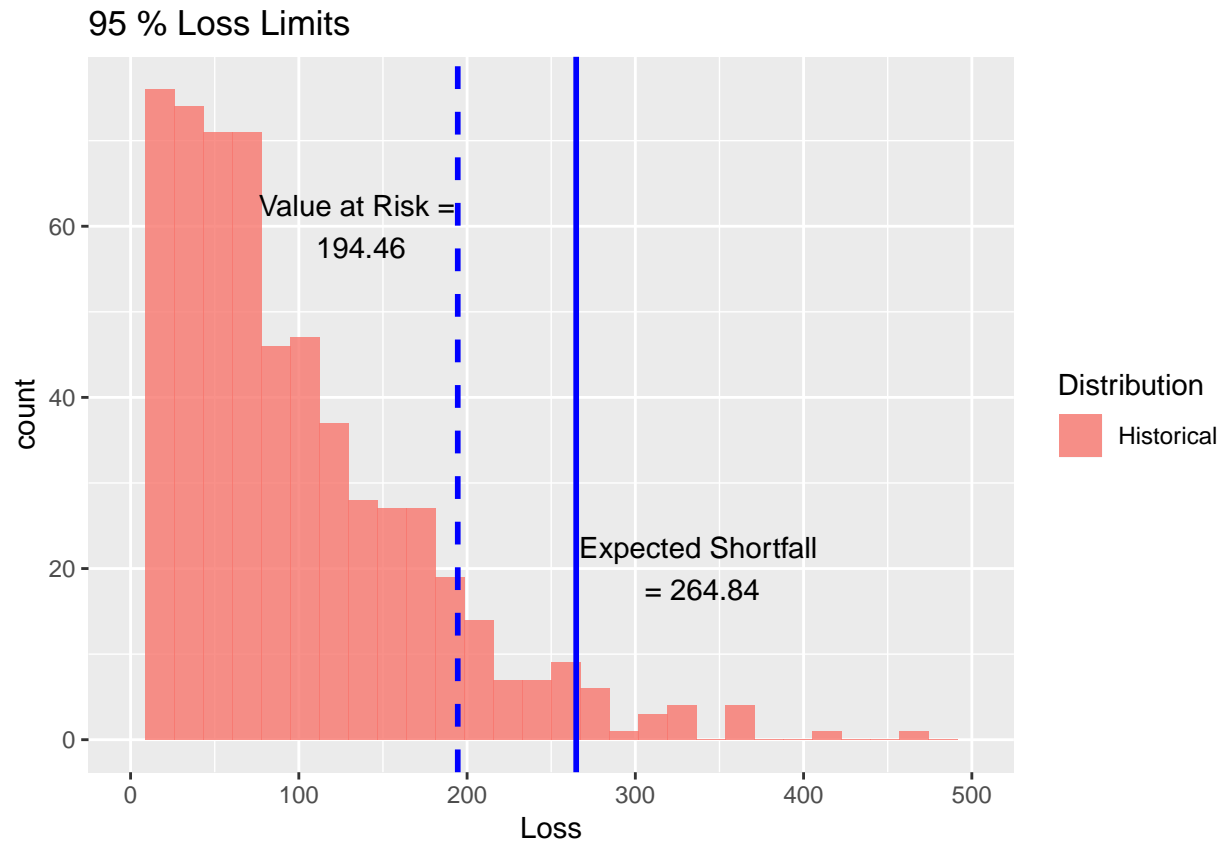
#And summing the row to see how much we lost that day. Rowsums is negative
#to reflect losses
loss.rf <- -rowSums(expm1(data.r/100) *
  weights.rf)
#Putting the daily losses series into a data frame and putting in
#"Historical" to show that this is based on historical data
loss.rf.df <- data.frame(Loss = loss.rf,
  Distribution = rep("Historical",
    each = length(loss.rf)))
## How much is our value at risk for having nickel, aluminum, and copper
#at 95%
alpha.tolerance <- 0.95
VaR.hist <- quantile(loss.rf, probs = alpha.tolerance,
  names = FALSE)
## Calculating expected shortfall
ES.hist <- mean(loss.rf[loss.rf > VaR.hist])
VaR.text <- paste("Value at Risk =\n",
  round(VaR.hist, 2)) # ='VaR'@c12
ES.text <- paste("Expected Shortfall \n=",
  round(ES.hist, 2))
title.text <- paste(round(alpha.tolerance *
  100, 0), "% Loss Limits")
# using histogram bars instead of the
# smooth density only showing potential losses
p <- ggplot(loss.rf.df, aes(x = Loss,
  fill = Distribution)) + geom_histogram(alpha = 0.8, bins = 30) +
  geom_vline(aes(xintercept = VaR.hist),
    linetype = "dashed", size = 1,
    color = "blue") + geom_vline(aes(xintercept = ES.hist),
    size = 1, color = "blue") + annotate("text",
    x = VaR.hist - 60, y = 60, label = VaR.text) +
  annotate("text", x = ES.hist + 75, y = 20,
    label = ES.text) + xlim(0, 500) +
  ggtitle(title.text)
p

```

```

## Warning: Removed 681 rows containing non-finite values (stat_bin).

```



The histogram shows that on a give day, there is only a 5% chance that our losses will be greater than \$194.46 for a ton of uniformly distributed metal on a given day. All losses greater than our value at risk averaged out to be \$264.84, and we will then try to hedge \$264.84 per ton of equally distributed metal, making sure we do not have losses that compromise our entire shipment.

Using the historical loss data, given our threshold for loss, it would be important to know how much more could the losses be than our threshold. In other words, if we are hedging against the threshold amount, how much more money could we be on the hook for? This is important for management to know, so they have an idea of what to expect to pay for in disaster scenarios. However, in using this figure, it would be important to know how much we are paying to hedge against our threshold for loss.

In the following R block, we will use the daily losses we had calculated previously to create a sequence of 100 different thresholds for loss separated equally between the minimum amount we had lost, and the maximum amount of money we had lost. We will the go through each of these thresholds and create a subset of all the losses greater than our threshold. Using this subset and our threshold for loss, we will subtract our threshold from the subset of data to calculate how much more than our threshold was the loss, and finally calculate average of how much more were the losses than our threshold.

We also would like to incorporate a confidence interval for this plot, therefore we would need the standard deviation of how much more than our threshold was the losses greater than our threshold and the number of daily loss instances greater than our threshold. using the z-score for 95% confidence, we will multiply by the standard deviation and divide by square root of the number of instances. This will be added and subtracted to our mean to provide a 95% confidence band around how much more we should expect to pay for losses. Finally, we will plot the data to show our thresholds and expected losses.

```
#Making sure daily loss data is stored in a vector
data <- as.vector(loss.rf)
#What is the least amount of money we had lost (for xlim)
umin <- min(data)
```

```

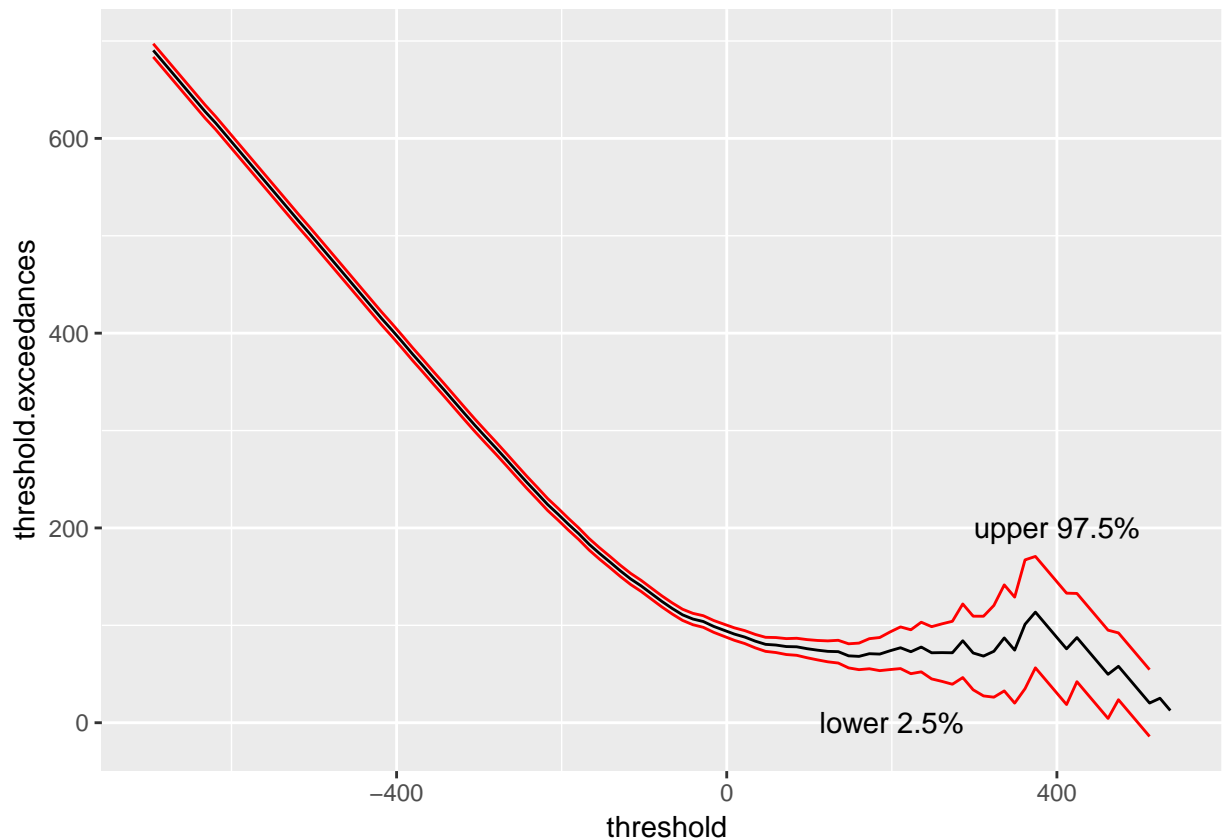
#What was our largest loss (for xlim)
umax <- max(data)
# number of exceedances we want to plot
nint <- 100
# creating vector to contain the number of exceedances we want to plot
grid.0 <- numeric(nint)
#Storing the average exceedance in the vector e
e <- grid.0
#Storing the upper confidence interval in the upper vector
upper <- grid.0
# storing the lower confidence interval in the lower variable
lower <- grid.0
#creating a threshold from the minimum loss to the maximum with 100 different breaks
u <- seq(umin, umax, length = nint) # threshold u grid
#95% confidence level
alpha <- 0.95
#for each of the loss thresholds in the sequence u
for (i in 1:nint) {
  #subset data that is greater than the threshold
  data <- data[data > u[i]]
  #subtract the threshold from the data, and store the mean of that value in the
  #average exceedances vector
  e[i] <- mean(data - u[i])
  #Calculate the standard deviation for the data greater than the threshold
  sdev <- sqrt(var(data))
  #How many samples are in the data
  n <- length(data)
  #The confidence limits for this threshold is calculated around the mean with a
  #confidence interval of 5, meaning upper limit is at 97.5% and lower 2.5%
  #So adding the confidence interval of 97.5% to the mean
  upper[i] <- e[i] + (qnorm((1 + alpha)/2) *
    sdev)/sqrt(n)
  #Subtracting confidence interval from mean
  lower[i] <- e[i] - (qnorm((1 + alpha)/2) *
    sdev)/sqrt(n)
}
#Creating a data frame with the threshold for loss, average of how
#much greater the losses are than our threshold for loss, and the
#upper and lower limits for the average exceedance for our threshold
mep.df <- data.frame(threshold = u, threshold.exceedances = e,
  lower = lower, upper = upper)
#with the thresholds and limits df, use the threshold for loss as the x
#and the average of how much greater the losses are than the threshold as y
p <- ggplot(mep.df, aes(x = threshold,
  y = threshold.exceedances)) + geom_line() + #and make it a line plot
  geom_line(aes(x = threshold, y = lower), #draw the upper limit as a red line
    colour = "red") + geom_line(aes(x = threshold, #draw the lower limit as a red line
  y = upper), colour = "red") + annotate("text", #Add text showing the upper and lower limits
  x = 400, y = 200, label = "upper 97.5%") +
  annotate("text", x = 200, y = 0,
    label = "lower 2.5%")
p

```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```

```
## Warning: Removed 3 rows containing missing values (geom_path).
```

```
## Warning: Removed 3 rows containing missing values (geom_path).
```



The above plot shows if the actual losses are greater than our threshold for losses, there is a fairly narrow confidence band up until approximately \$150 in losses. After the losses are greater than \$150, because the samples are smaller, it is somewhat more difficult to know how much more we would likely have to pay. This plot also tells us, that is we were to insure for our expected shortfall amount of \$264.84, on average we would also have to pay approximately \$80 on a per ton of equally distributed metals by weight. While this is the measured sample mean of additional funds required, there is a 95% chance that the actual average is between \$42 and \$100.

Question 4: Modeling

What we care about most here are the extremes of our data set, the losses. These extremes, or tails, depict potential losses and the subsequent needed hedging for risk mitigation as discussed above. With low density distributions, such as with distribution tails, it is usual difficult to model. The right skewed generalized pareto distribution helps model these low density data sets to predict future losses for risk mitigation and development business strategy. The understating of these skewed “rare” events is extremely important as these events can lead to enormous losses.

After defining our threshold or quantile, the below chunk will fit the distribution using the command `fit.GPD`, to our historical loss data set and estimate our predicted value at risk and expected shortfall. We can use these distribution estimations and compare them to the determined VaR and ESF directly from the data set above to see if the distribution is a good fit.

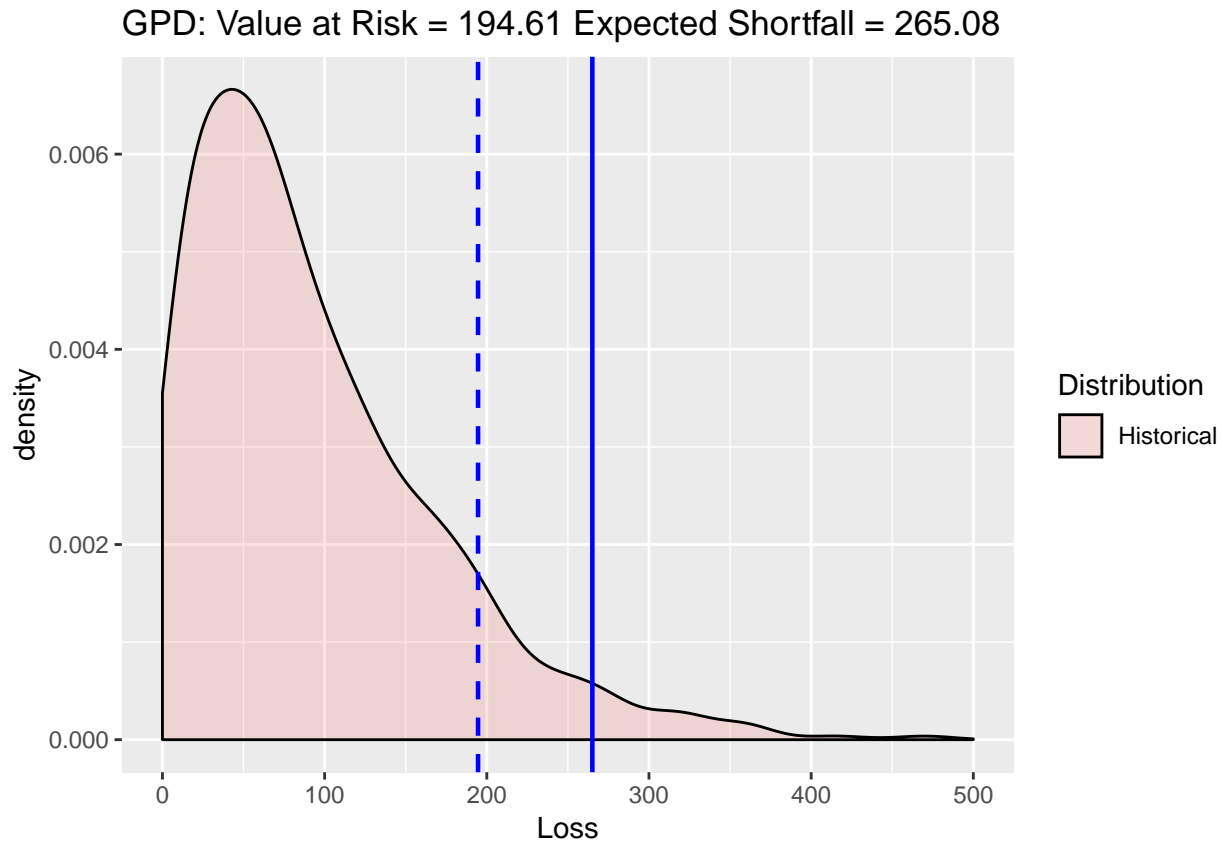
```
## Toleranvce for risk  
alpha.tolerance <- 0.95
```

```

#getting the losses at that tolerance for risk
u <- quantile(loss.rf, alpha.tolerance,
  names = FALSE)
#Losses exceed out tolerance by how much?
loss.excess <- loss.rf[loss.rf > u] - u
#Using the generalized pareto distribution to fit losses with
#thresholds determined by u
fit <- fit.GPD(loss.rf, threshold = u)
#getting the fitted shape
xi.hat <- fit$par.ests[["xi"]]
#getting the fitted scale
beta.hat <- fit$par.ests[["beta"]]
#saving the losses in the data variable
data <- loss.rf
#how often is loss greater than our tolerance
n.relative.excess <- length(loss.excess)/length(loss.rf)
#VaR for pareto
VaR.gpd <- u + (beta.hat/xi.hat) * (((1 -
  alpha.tolerance)/n.relative.excess)^(-xi.hat) -
  1)
#estimated shortfall for pareto
ES.gpd <- (VaR.gpd + beta.hat - xi.hat *
  u)/(1 - xi.hat)
# Plot away
VaRgpd.text <- paste("GPD: Value at Risk =",
  round(VaR.gpd, 2))
ESgpd.text <- paste("Expected Shortfall =",
  round(ES.gpd, 2))
title.text <- paste(VaRgpd.text, ESgpd.text,
  sep = " ")
loss.plot <- ggplot(loss.rf.df, aes(x = Loss,
  fill = Distribution)) + geom_density(alpha = 0.2)
loss.plot <- loss.plot + geom_vline(aes(xintercept = VaR.gpd),
  colour = "blue", linetype = "dashed",
  size = 0.8)
loss.plot <- loss.plot + geom_vline(aes(xintercept = ES.gpd),
  colour = "blue", size = 0.8)
## annotate('text', x = 300, y = 0.0075, label = VaRgpd.text, colour = 'blue') + annotate('text', x = 300, y = 0.0075, label = ESgpd.text, colour = 'blue')
loss.plot <- loss.plot + xlim(0, 500) +
  ggtitle(title.text)
loss.plot

```

```
## Warning: Removed 681 rows containing non-finite values (stat_density).
```

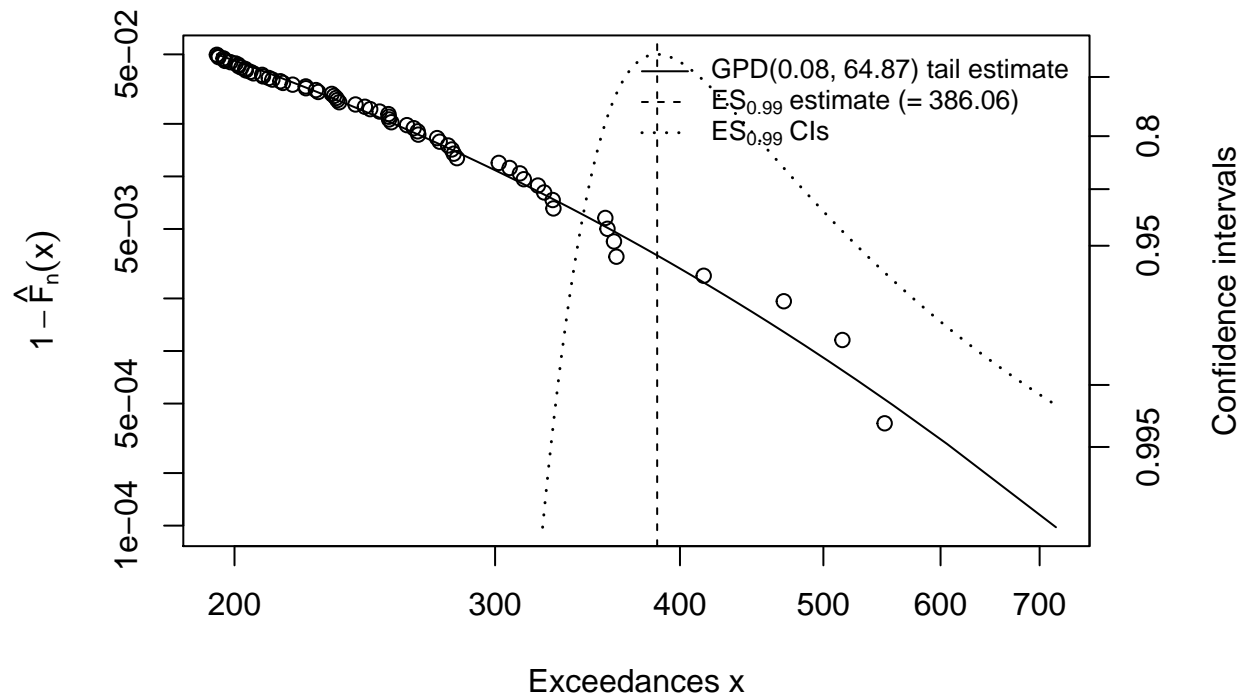


The above plot shows the fit of the generalized pareto distribution with the subsequent expected losses for a 95% threshold and the average expected loss (expected shortfall) of the 5% extreme. The GPD value at risk and estimated shortfall values align particularly well with the 95% loss limit graph that used the historical data previously.

Using the `showRM` function from the `QRM` package we will look at the confidence intervals in relation to the expected shortfalls as well as the quantile estimates. This also plots the fitted pareto distribution of the tail.

```
# Confidence in GPD
showRM(fit, alpha = 0.99, RM = "ES",
       method = "BFGS")
```

Estimated tail probabilities

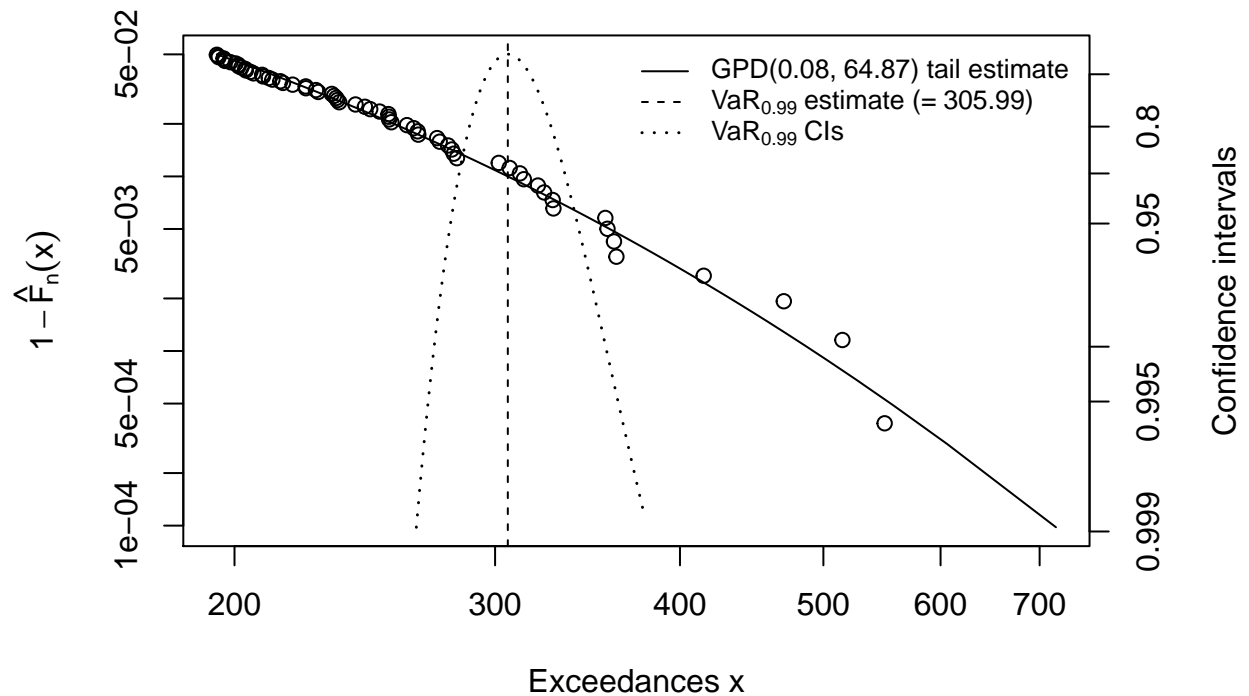


The left vertical axis depicts the probabilities of being within the tail (in excess of the determined threshold), while the horizontal displays the losses over the established threshold. The right vertical access depicts the confidence intervals. Visual inspections show that at choosing to absorb 99% of the losses the expected shortfall upper bound is large indicating a potentially high loss, heavily tailed that could exceed the expected shortfall. This does not support significant confidence at this threshold value.

Using the `showRM` function from the `QRM` package we will look at the confidence intervals in relation to the value at risk threshold.

```
showRM(fit, alpha = 0.99, RM = "VaR",
       method = "BFGS")
```


Estimated tail probabilities



The value at risk for a threshold of 99%, shows an estimate maximum loss of 305. Visualizing a horizontal dash line from the specified confidence interval reveals the boundaries of the confidence interval. The more narrow the bound the more accurate the Var estimation is determined to be.

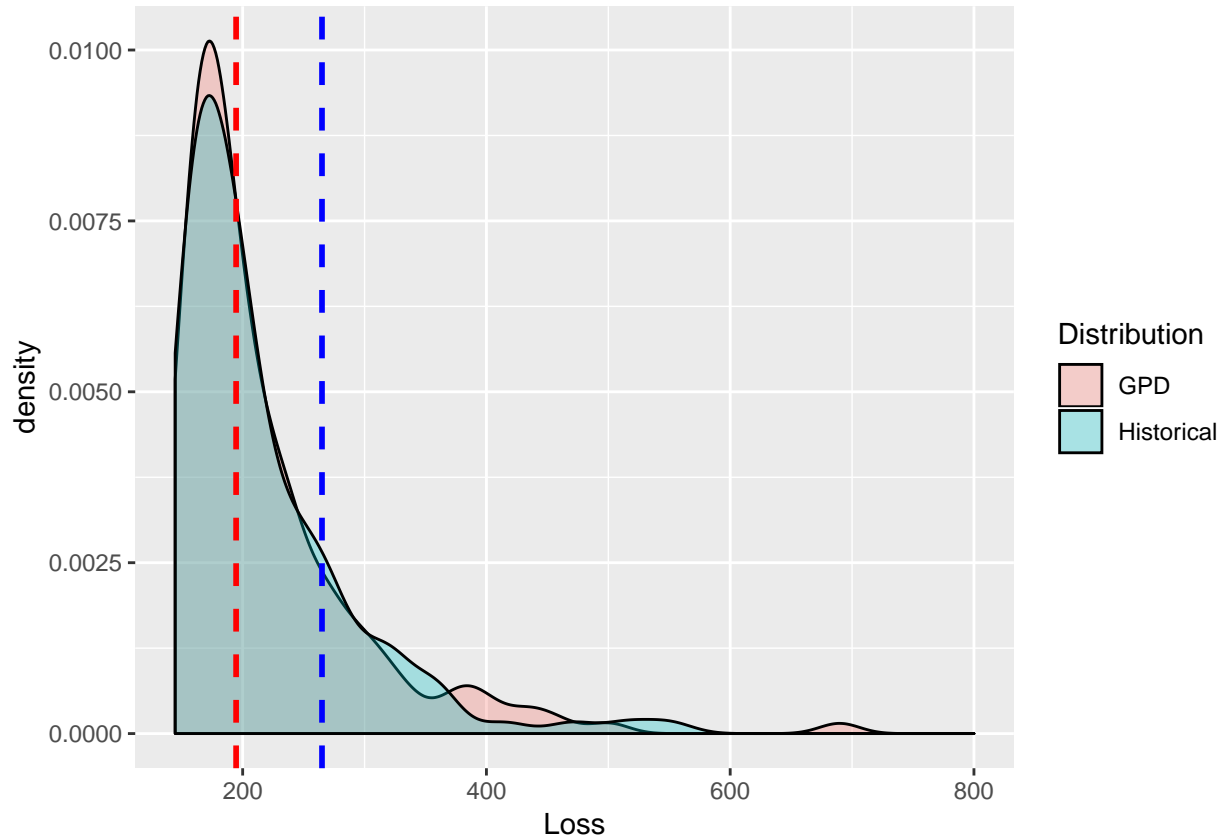
It is important to understand the distribution is a good fit for our data set. If it is not a good fit we will not be able to properly perform risk mitigation and the company may suffer significant losses in the investment of these commodities.

```
set.seed(1016)
n.sim <- 1297
#randomly modeling the losses for pareto
gpd.loss<- rGPD(n.sim, xi.hat, beta.hat)
p.df <- data.frame(Loss = c(loss.rf, gpd.loss), Distribution = rep(c("Historical", "GPD"), each = n.sim))

p <- ggplot(p.df, aes(x = Loss, fill = Distribution)) + geom_density(alpha = 0.3) + geom_vline(aes(xintercept = ES.gpd),
size = 1) + geom_vline(aes(xintercept = ES.gpd),
color = "blue", linetype = "dashed",
size = 1) + xlim(VaR.gpd - 50, 800)

p
```

```
## Warning: Removed 2298 rows containing non-finite values (stat_density).
```



The generalized pareto distribution shows more peakedness within the tail indicating higher expected losses than the historical data supports. In particular the GPD predicted peak around 700 shows significant density, while historical data shows no significant data points after loss of 600. Depending on our quantile threshold of acceptable risk, management may hedge risk that historically is not supported. With that said it should be noted that this is a relatively good fit for our historical data set. Knowing this we can now use this as a decision tool to determine portfolio optimization for other, no uniform arrangement of our commodities. We would look for other commodity weights that would reduce the peakedness of our tails, allowing us to have more predictive returns.