---

title: "The Answer is 42"

output:

flexdashboard::flex\_dashboard:

orientation: rows

social: menu

source\_code: embed

runtime: shiny

---

---

title: "QuantReg"

output:

flexdashboard::flex\_dashboard:

orientation: columns

social: menu

source\_code: embed

runtime: shiny

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)

require(ggplot2)

require(flexdashboard)

require(qrmdata)

require(xts)

require(zoo)

require(xts)

require(ggplot2)

# PAGE: Exploratory Analysis

# Read and review a csv file from FRED

data <- na.omit(read.csv("data/fuelprices.csv", header = TRUE))

# Compute log differences percent using as.matrix to force numeric type

data.r <- diff(log(as.matrix(data[, -1]))) \* 100

# Create size and direction

size <- na.omit(abs(data.r)) # size is indicator of volatility

#head(size)

colnames(size) <- paste(colnames(size),".size", sep = "") # Teetor

direction <- ifelse(data.r > 0, 1, ifelse(data.r < 0, -1, 0)) # another indicator of volatility

colnames(direction) <- paste(colnames(direction),".dir", sep = "")

# Convert into a time series object:

# 1. Split into date and rates

dates <- as.Date(data$DATE[-1], "%m/%d/%Y")

dates.chr <- as.character(data$DATE[-1])

str(dates.chr)

values <- cbind(data.r, size, direction)

# for dplyr pivoting and ggplot2 need a data frame also known as "tidy data"

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, stringsAsFactors = FALSE)

#non-coerced dates for subsetting on non-date columns

# 2. Make an xts object with row names equal to the dates

data.xts <- na.omit(as.xts(values, dates)) #order.by=as.Date(dates, "%d/%m/%Y")))

#str(data.xts)

data.zr <- as.zooreg(data.xts)

returns <- data.xts

# PAGE: Market risk

corr.rolling <- function(x) {

dim <- ncol(x)

corr.r <- cor(x)[lower.tri(diag(dim), diag = FALSE)]

return(corr.r)

}

require(zoo)

ALL.r <- data.xts[, 1:3]

window <- 90 #reactive({input$window})

corr.returns <- rollapply(ALL.r, width = window, corr.rolling, align = "right", by.column = FALSE)

#colnames(corr.returns) <- c("Diesel & WTI", "Diesel & Gas", "WTI & Gas")

corr.returns.df <- data.frame(Date = index(corr.returns), Diesel.WTI = corr.returns[,1], Diesel.Gas = corr.returns[,2], WTI.Gas = corr.returns[,3])

# Market dependencies

require(matrixStats)

R.corr <- apply.monthly(as.xts(ALL.r), FUN = cor)

R.vols <- apply.monthly(ALL.r, FUN = colSds) # from MatrixStats

# Form correlation matrix for one month

R.corr.1 <- matrix(R.corr[20,], nrow = 3, ncol = 3, byrow = FALSE)

rownames(R.corr.1) <- colnames(ALL.r[,1:3])

colnames(R.corr.1) <- rownames(R.corr.1)

R.corr <- R.corr[, c(2, 3, 6)]

colnames(R.corr) <- colnames(corr.returns)

colnames(R.vols) <- c("Diesel.vols", "WTI.vols", "Gas.vols")

R.corr.vols <- merge(R.corr, R.vols)

Diesel.vols <- as.numeric(R.corr.vols[,"Diesel.vols"])

WTI.vols <- as.numeric(R.vols[,"WTI.vols"])

Gas.vols <- as.numeric(R.vols[,"Gas.vols"])

require(quantreg)

# hist(rho.fisher[, 1])

Diesel.corrs <- R.corr.vols[,1]

#hist(Diesel.corrs)

taus <- seq(.05,.95,.05) # Roger Koenker UI Bob Hogg and Allen Craig

fit.rq.Diesel.WTI <- rq(Diesel.corrs ~ WTI.vols, tau = taus)

fit.lm.Diesel.WTI <- lm(Diesel.corrs ~ WTI.vols)

#'

#summary(fit.rq.Diesel.WTI, se = "boot")

#'

#summary(fit.lm.Diesel.WTI, se = "boot")

plot(summary(fit.rq.Diesel.WTI), parm = "WTI.vols", main = "Diesel-WTI correlation sensitivity to WTI volatility")

## Optimization

# helper functions

VaR.hist <- function(alpha, data){

# Value at Risk using historical simulation

data <- na.omit(data)

data <- na.

}

VaR.norm <- function(alpha,mu,sigma,S=1){

# The VaR under the assumption of a normal-distribution

- S \* ( mu + qnorm(alpha) \* sigma )

}

ES.norm <- function(alpha,mu,sigma,S=1){

# The ES under the assumption of a normal-distribution

S \* ( - mu + sigma \* ( dnorm(qnorm(alpha))/alpha ) )

}

VaR.t <- function(alpha,mu,lambda,nu,S=1){

# The VaR under the assumption of a t-distribution

- S \* ( mu + qt(alpha,nu) \* lambda )

}

ES.t <- function(alpha,mu,lambda,nu,S=1){

# The ES under the assumption of a t-distribution

S \* ( - mu + lambda \* ( dt(qt(alpha,nu),nu) / alpha ) \* ( ( nu + qt(alpha,nu)^2 )/( nu - 1 ) ) )

}

VaR.gpd <- function(alpha,mu,lambda,nu,S=1){

# The VaR under the assumption of a t-distribution

- S \* ( mu + qt(alpha,nu) \* lambda )

}

ES.gpd <- function(alpha,mu,lambda,nu,S=1){

# The ES under the assumption of a t-distribution

S \* ( - mu + lambda \* ( dt(qt(alpha,nu),nu) / alpha ) \* ( ( nu + qt(alpha,nu)^2 )/( nu - 1 ) ) )

}

nsim <- 1000

alpha <- 0.01

data <- returns

risk.boot.t <- function(data, ind, alpha = 0.01, nsim = 1000){

# Fit a t-distribution to this data and estimate VaR and ES:

#

data <- unique(data[ind]) # duplicate values seem to give us trouble

params <- fitdistr(data, densfun="t", start=list(m=-0.001, s=0.02, df=4.00))

params <- as.vector(params$estimate)

mu <- params[1]

sigma <- params[2] # estimate of sigma = sqrt( (nu-2)/nu ) standard\_deviation

nu <- params[3]

VaR <- VaR.t(alpha, mu, sigma, nu, nsim)

ES <- ES.t(alpha, mu, sigma, nu, nsim)

return(c( mu, sigma, nu, VaR, ES ))

}

risk.boot.gpd <- function(data, ind, alpha = 0.01, qu = .9, nsim = 1000){

# Fit a t-distribution to this data and estimate VaR and ES:

#

require(QRM)

data <- unique(data[ind]) # duplicate values seem to give us trouble

u <- quantile(data, probs = qu, names = FALSE)

params <- fit.GPD(data, threshold = u, start=list(m=-0.001, s=0.02, df=4.00))

params <- as.vector(params$estimate)

mu <- params[1]

sigma <- params[2] # estimate of sigma = sqrt( (nu-2)/nu ) standard\_deviation

nu <- params[3]

VaR <- VaR.t(alpha, mu, sigma, nu, nsim)

ES <- ES.t(alpha, mu, sigma, nu, nsim)

return(c( mu, sigma, nu, VaR, ES ))

}

#risk.boot.result <- boot(returns[,1], risk.boot.t, R=10)

#VaR.boot <- risk.boot.result$t[,4]

#plot( density(VaR\_boots), lwd=2, main="VaR (kde)" )

#

#returns.params <- fitdistr(returns[,1], densfun="t", start=list(m=-0.001, s=0.02, df=4.00))

# shapiro.test(VaR.boot) <- risk.boot.result$t[,5]

#boot.ci(risk.boot.result, conf=0.95, type="norm", index=4)

#boot.ci(risk.boot.result, conf=0.95, type="norm", index=5)

```

Decision

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

Background

-----------------------------------------------------------------------

### Introduction

A recent West Texas Intermediate crude oil contract worth more than 20 percent of our company's current revenue hit our books. This is a new market for us.

1. Retrieve and begin to analyze data about the new market

2. Compare and contrast this new market with existing markets in conventional gas and ultra-low sulfur diesel.

3. Begin to generate economic scenarios based on events that may, or may not, materialize in the new market.

### A financial analytics workflow

1. What decision(s) are we making?

2. What are the key business questions we need to support this decision?

3. What data do we need?

4. What tools do we need to analyze the data?

5. How do we communicate answers to inform the decision?

Decision

-----------------------------------------------------------------------

### Decision is \*\*supply a new market segment\*\*

- Product: Crude Oil shipments

- Geography: NY, Connecticut, NJ

- Customers: major refiners at the Port of Newark.

### Three business questions:

- How would the performance of these markets affect the size and timing of orders?

- How would the value of the new market affect the value of our business with our current customers?

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

Approach

-----------------------------------------------------------------------

### Stylized facts of the market

Learned the hard Way: not independent, volatile volatility, extreme

- Financial stock, bond, commodity...you name it...have highly interdependent relationships.

- Volatility is rarely constant and often has a structure (mean reversion) and is dependent on the past.

- Past shocks persist and may or may not dampen (rock in a pool).

- Extreme events are likely to happen with other extreme events.

- Negative returns are more likely than positive returns (left skew).

### History speaks

- We will develop the \*value at risk\* and \*expected shortfall\* metrics from the historical simulated distributions of risk factors.

- Given these factors we combine them into a portfolio and calculate their losses.

- Finally with the loss distribution in hand we can compute the risk measures. - This approach is nonparametric.

### Data and analysis to inform the decision

- Customer stock prices: volatility and correlation

- WTI prices: volatility

- Diesel and Conventional gas: volatility

- All together: correlations among these indicators

Data

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

Row {data-width=900}

-----------------------------------------------------------------------

### Fuel Price Percent Changes

```{r}

renderPlot({

require(ggplot2)

title.chg1 <- "Fuel Price Percent Changes"

title.chg2 <- "Size of Fuel Price Percent Changes"

autoplot.zoo(data.xts[,1:3]) + ggtitle(title.chg1) #+ ylim(-5, 5)

})

```

### Size of Fuel Price Percent Changes

```{r}

renderPlot({

require(ggplot2)

title.chg1 <- "Fuel Price Percent Changes"

title.chg2 <- "Size of Fuel Price Percent Changes"

autoplot.zoo(abs(data.xts[,1:3])) + ggtitle(title.chg2) #+ ylim(-5, 5)

})

```

Row

-----------------------------------------------------------------------

### Data Definitions

- \*DDFUELNYH\*: daily Ultra-low Sulphur Diesel Oil New York Harbor prices (\$/gallon)

- \*DCOILWTICO\*: daily West Texas Intermediate Crude oil prices (\$/barrel)

- \*DGASNYH\*: daily Conventional Gasoline prices (\$/gallon)

### Commentary

- Volatility clustering

Exploratory Analysis

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

Row {.sidebar}

-----------------------------------------------------------------------

A quantile divides the returns distribution into two groups. For example 75\% of all returns may fall below a return value of 10\%. The distribution is thus divided into returns above 10\% and below 10\% at the 75\% quantile.

Pull slide to the right to measure the risk of returns at desired quantile levels. The minimum risk quantile is 75\%. The maximum risk quantile is 99\%.

```{r}

sliderInput("alpha.q", label = "Risk Measure quantiles (%):",

min = 0.75, max = 0.99, value = 0.75, step = 0.01)

```

If an error appears, move a slider further to the right.

Row

-----------------------------------------------------------------------

### Diesel Fuel Value at Risk

```{r}

#threshold <- reactive({input$threshold.q}) #BE SURE that {} included

renderValueBox({

alpha <- reactive({ifelse(input$alpha.q>1,0.99,ifelse(input$alpha.q<0,0.001,input$alpha.q))})

returns1 <- returns[,1]

colnames(returns1) <- "Returns" #kluge to coerce column name for df below

q <- quantile(returns1,alpha())

VaR.hist <- q

valueBox(round(VaR.hist, 2),

icon = "glyphicon-signal")

})

```

### WTI Value at Risk

```{r}

#threshold <- reactive({input$threshold.q}) #BE SURE that {} included

renderValueBox({

alpha <- reactive({ifelse(input$alpha.q>1,0.99,ifelse(input$alpha.q<0,0.001,input$alpha.q))})

returns1 <- returns[,2]

colnames(returns1) <- "Returns" #kluge to coerce column name for df below

q <- quantile(returns1,alpha())

VaR.hist <- q

valueBox(round(VaR.hist, 2),

icon = "glyphicon-signal")

})

```

### Gasolene Value at Risk

```{r}

#threshold <- reactive({input$threshold.q}) #BE SURE that {} included

renderValueBox({

alpha <- reactive({ifelse(input$alpha.q>1,0.99,ifelse(input$alpha.q<0,0.001,input$alpha.q))})

returns1 <- returns[,3]

colnames(returns1) <- "Returns" #kluge to coerce column name for df below

q <- quantile(returns1,alpha())

VaR.hist <- q

valueBox(round(VaR.hist, 2),

icon = "glyphicon-signal")

})

```

Row {.tabset .tabset-fade}

-----------------------------------------------------------------------

### Diesel Fuel Returns Distribution

```{r}

# VaR.text <- paste("Value at Risk =", round(VaR.hist, 2))

# ES.text <- paste("Expected Shortfall =", round(ES.hist, 2))

# VaR.x <- quantile(returns1, q)

# ES.x <- quantile(returns1, q)

# title.text <- paste(VaR.text," ",ES.text)

#title.text

#})

renderPlot({

returns1 <- returns[,1]

colnames(returns1) <- "Returns" #kluge to coerce column name for df

returns1.df <- data.frame(Returns = returns1[,1], Distribution = rep("Historical", each = length(returns1)))

ggplot(returns1.df, aes(x = Returns, fill = Distribution)) + geom\_density(alpha = 0.8)

})

#+ 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.x, y #= 0.6, label = VaR.text) + annotate("text", x = ES.x, y = 0.4, label = ES.text) + #ggtitle(title.text)

```

### West Texas Intermediate Crude Oil Returns Distribution

```{r}

renderPlot({

returns1 <- returns[,2]

colnames(returns1) <- "Returns" #kluge to coerce column name for df

returns1.df <- data.frame(Returns = returns1[,1], Distribution = rep("Historical", each = length(returns1)))

ggplot(returns1.df, aes(x = Returns, fill = Distribution)) + geom\_density(alpha = 0.8)

})

```

### Conventional Gasoline Returns Distribution

```{r}

renderPlot({

returns1 <- returns[,3]

colnames(returns1) <- "Returns" #kluge to coerce column name for df

returns1.df <- data.frame(Returns = returns1[,1], Distribution = rep("Historical", each = length(returns1)))

ggplot(returns1.df, aes(x = Returns, fill = Distribution)) + geom\_density(alpha = 0.8)

})

```

PAGE

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

### Measurements

Value at Risk (VaR) is measured as

$$

VaR = q\_{\alpha}(R\_t)

$$

where $\alpha$ is the quantile level of tolerance, and $R\_t$ are returns.

Expected Shortfall (ES) is measured as

$$

ES = E[R\_t \mid R\_t > VaR]

$$

where $E[]$ is the expectation, a measure of central location.

### Observations

- Appear to be symmetrical

- Very high quantile tails

Market Risk

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

column {data = 1000}

-----------------------------------------------------------------------

### Diesel and WTI (90 day rolling correlation)

```{r }

renderPlot({

ggplot(corr.returns.df, aes(x = Date, y = Diesel.WTI)) + geom\_line()

})

```

### Diesel and Gas (90 day rolling correlation)

```{r }

renderPlot({

ggplot(corr.returns.df, aes(x = Date, y = Diesel.Gas)) + geom\_line()

})

```

### WTI and Gas (90 day rolling correlation)

```{r }

renderPlot({

ggplot(corr.returns.df, aes(x = Date, y = WTI.Gas)) + geom\_line()

})

```

column

-----------------------------------------------------------------

### 30 day within-sample correlations and volatilities

```{r}

plot.zoo(R.corr.vols, main= "Monthly Correlations and Volatilities")

```

### Diesel - WTI Dependency

```{r}

plot(summary(fit.rq.Diesel.WTI), parm = "WTI.vols", main = "Diesel-WTI correlation sensitivity to WTI volatility")

```

```{r}

```

Optimization

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

Row

-----------------------------------------------------------------------

### Efficient Portfolio Frontier

```{r}

R <- returns[,1:3]/100

names.R <- colnames(R)

mean.R <- apply(R,2,mean)

cov.R <- cov(R)

sd.R <- sqrt(diag(cov.R)) ## remember these are in daily percentages

require(quadprog)

Amat <- cbind(rep(1,3),mean.R) ## set the equality constraints matrix

mu.P <- seq(-.00075, .0003, length=300) ## set of 300 possible target portfolio returns

sigma.P <- mu.P ## set up storage for std dev's of portfolio returns

weights <- matrix(0, nrow=300, ncol = ncol(R)) ## storage for portfolio weights

colnames(weights) <- names.R

for (i in 1:length(mu.P))

{

bvec = c(1,mu.P[i]) ## constraint vector

result =

solve.QP(Dmat=2\*cov.R,dvec=rep(0,3),Amat=Amat,bvec=bvec,meq=2)

sigma.P[i] = sqrt(result$value)

weights[i,] = result$solution

}

mu.free = 1.3/253/100 ## input value of risk-free interest rate

sharpe <- ( mu.P-mu.free)/sigma.P ## compute Sharpe's ratios

ind <- (sharpe == max(sharpe)) ## Find maximum Sharpe's ratio

ind2 <- (sigma.P == min(sigma.P)) ## find the minimum variance portfolio

renderPlot({

par(mfrow = c(1,1))

plot(sigma.P,mu.P,type="l", lty = 3, lwd = 3) #, xlim=c(0,max(sd.R)\*1.1), ylim=c(min(mean.R)\*1.05, max(mean.R)\*1.1), lty=3, lwd = 3) ## plot the efficient frontier (and inefficient portfolios

## below the min var portfolio)

mu.free = 1.3/253/100 ## input value of risk-free interest rate

points(0,mu.free,cex=3,pch="+") ## show risk-free asset

sharpe <- ( mu.P-mu.free)/sigma.P ## compute Sharpe's ratios

ind <- (sharpe == max(sharpe)) ## Find maximum Sharpe's ratio

options(digits=3)

lines(c(0,2),mu.free+c(0,2)\*(mu.P[ind]-mu.free)/sigma.P[ind],lwd=4,lty=1, col = "blue")

## show line of optimal portfolios

points(sigma.P[ind],mu.P[ind],cex=4,pch="\*") ## show tangency portfolio

ind2 <- (sigma.P == min(sigma.P)) ## find the minimum variance portfolio

points(sigma.P[ind2],mu.P[ind2],cex=2,pch="+") ## show min var portfolio

ind3 <- (mu.P > mu.P[ind2]) ## finally the efficient frontier

lines(sigma.P[ind3],mu.P[ind3],type="l", lwd = 3, col = "red") # xlim=c(0,max(sd.R)\*1.1), ylim=c(min(mean.R)\*1.05, max(mean.R)\*1.1) , lwd=3, col = "red") ## plot the efficient frontier

text(2\*sd.R[1], 2\*mean.R[1], names.R[1] ,cex=1.15)

text(3\*sd.R[2], mean.R[2], names.R[2] ,cex=1.15)

text(4\*sd.R[3], .5\*mean.R[3], names.R[3] ,cex=1.15)

})

```

### Portfolio Weights

The weights for the minimum variance portfolio ("+") are

```{r echo = FALSE}

weights[ind2,]

```

The weights for the tangency portfolio ("\*") are in

```{r echo = FALSE}

weights[ind,]

```

To maximize the return for risk ratio, and for a given notional amount in your portfolio, go long (buy) 930\% of that position in New York Harbor diesel, go short (sell) 106\% of your aggregate position in WTI crude, and go long 13\% in New York Harbor gasolene.

This means in the working capital accounts:

1. \$250 million should be denominated in euros

2. Net of a short (payables?) position of \$180 million

3. With another \$30 million priced in Brent crude.

If our working capital is \$100 million in euros, -\$200 in sterling, and \$200 exposed to Brent, we might think of ways to bring this more into line with the optimal positions we just derived, by changing contract terms and using swaps and other derivative instruments.

- Tangency

- Minimum Variance

Documentation

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

BLAH WONK BLAH ETC