

## Assignment 2

Mgmt 237E: Empirical Methods

Ian Laker, PrasanthKumar, Nitish Ramkumar

### Problem 1

Lets load the data into R

```
options(warn = -1)
suppressMessages(library(lubridate))
suppressMessages(library(xts))
options(warn = 0)

# Retrieve Data
portfolio.data <- read.csv("49_Industry_Portfolios.csv",header=TRUE,sep = ",",stringsAsFactors = FALSE,
portfolio.data$X <- as.Date(paste(floor(as.numeric(portfolio.data$X)/100),as.numeric(portfolio.data$X) %
portfolio.data <- xts(portfolio.data[,-1],order.by = portfolio.data$X)

datafactors <- read.csv("F-F_Research_Data_Factors.csv",header=TRUE,sep = ",",stringsAsFactors = FALSE,
datafactors$X <- as.Date(paste(floor(as.numeric(datafactors$X)/100),as.numeric(datafactors$X) %% 100,"0
datafactors <- xts(datafactors[,-1],order.by = datafactors$X)

#Constraints
#Date
portfolio.data <- portfolio.data[index(portfolio.data)>="1960-01-01" & index(portfolio.data)<="2015-12-31",]
datafactors <- datafactors[index(datafactors)>="1960-01-01" & index(datafactors)<="2015-12-31",]

#Remove columns with -99.99 value
contains99 <- logical(dim(portfolio.data)[2])
for(col in 1:dim(portfolio.data)[2]){
  contains99[col] <- sum(portfolio.data[,col] %in% -99.99) > 0
}
portfolio.data <- portfolio.data[,!contains99]

#Excess Returns
excessReturns <- function(industryReturns){
  industryReturns - datafactors$RF
}

excess.returns <- matrix(nrow=dim(portfolio.data)[1],ncol=dim(portfolio.data)[2])
for(col in 1:dim(portfolio.data)[2]){
  excess.returns[,col] <- portfolio.data[,col] - datafactors$RF
}
colnames(excess.returns) <- colnames(portfolio.data)
```

For each industry, regress the excess return on market excess return

```
regressionData <- function(excess.industryReturns){
  reg <- lm(excess.industryReturns~datafactors$Mkt.RF[index(excess.industryReturns)])
  lmSumm <- summary(reg)

  result <- c(lmSumm$coefficients[2,c(1,2)],lmSumm$coefficients[1,c(1,2)],lmSumm$r.squared)
```

```

}

excess.returns <- xts(excess.returns,order.by = as.Date(index(portfolio.data)))
regressionInfo <- apply(excess.returns, 2, regressionData)

```

**a**

Lets plot the betas of various industry

```

betaTable <- regressionInfo[c(1,2),]
betaBars <- barplot(height = betaTable[1,],
                    names.arg = colnames(betaTable),
                    beside = true, las = 2,
                    ylim = c(0, 1.75),
                    cex.names = 0.75, xaxt = "n",
                    main = "Plot of beta for various industries",
                    ylab = "Beta",
                    border = "black", axes = TRUE,width=35)

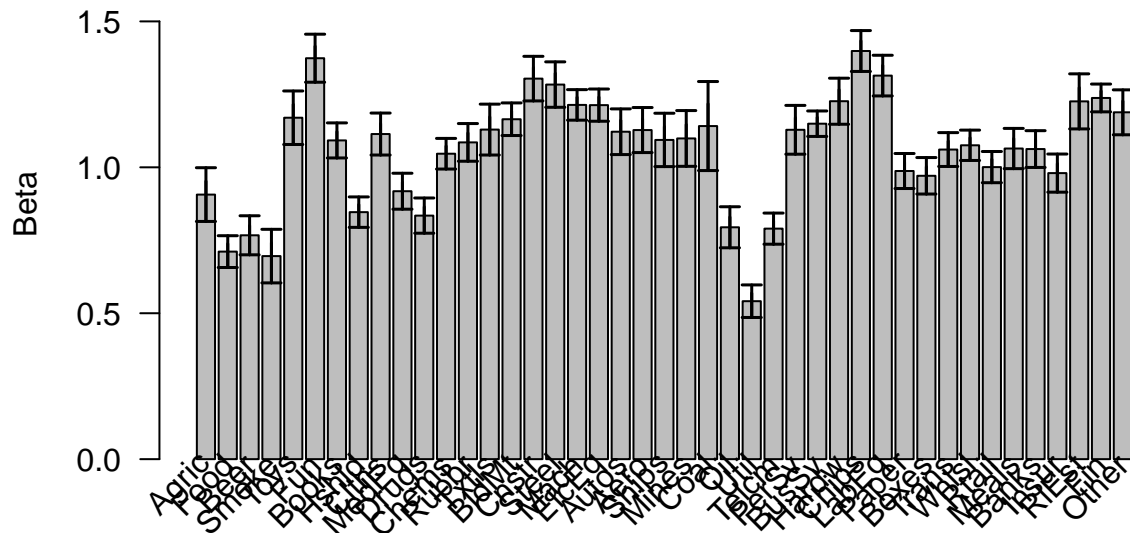
text(x = betaBars, y = 0, srt = 45,
     adj = 1, labels = colnames(betaTable), xpd = TRUE)

segments(betaBars, betaTable[1,]- betaTable[2,] * 2, betaBars,
          betaTable[1,] + betaTable[2,], lwd = 1.5)

arrows(betaBars, betaTable[1,]- betaTable[2,] * 2, betaBars,
        betaTable[1,] + betaTable[2,]*2, lwd = 1.5, angle = 90,
        code = 3, length = 0.05)

```

**Plot of beta for various industries**



**b**

The range of the betas can be found by using the *range* function

```
range(betaTable[1,])
```

```
## [1] 0.5412599 1.3985571
```

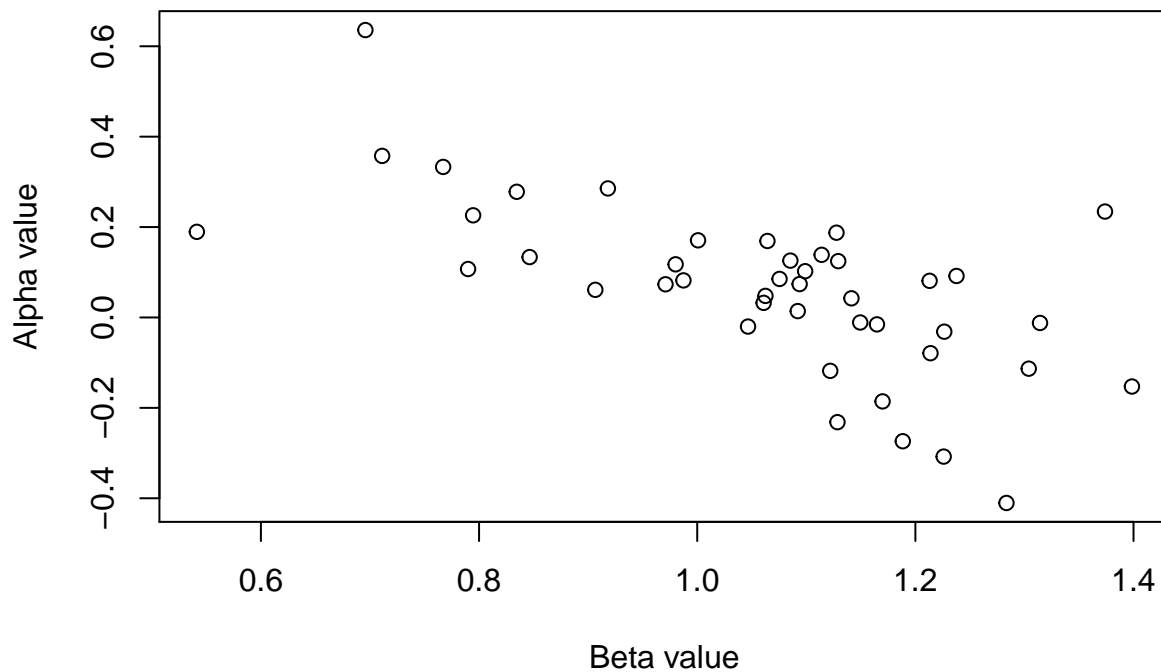
The max, min and mean of the  $R^2$  across industries is as below

```
results <- c()
results[1] <- min(regressionInfo[5,])
results[2] <- max(regressionInfo[5,])
results[3] <- mean(regressionInfo[5,])
names(results) <- c("Min", "Max", "Mean")
results
```

```
##      Min      Max      Mean
## 0.2505057 0.8055172 0.5776919
```

**c**

```
plot(regressionInfo[1,],regressionInfo[3,],xlab="Beta value",ylab="Alpha value")
```



We can see that for lower beta, we mostly get higher alpha.

So based on this if we don't take much risk (by having beta less than market beta), we get more return. This goes against CAPM.

## Problem 2

```
fiveYearBeta <- matrix(nrow=((2010-1960)/5)+1,ncol=dim(excess.returns)[2])
for(fiveYearStart in seq(1960,2010,by=5)){
  excess.5year.returns <- excess.returns[paste0(fiveYearStart,"/",fiveYearStart+4)]
  fiveYearBeta[((fiveYearStart-1960)/5)+1,] <- apply(excess.5year.returns, 2, regressionData)[1,]
}
```