Slides 0 : **Empirical Study: Application of Fama-French Three-Factor Model in S&P500**

[OUTLINE]

• Data Preparation

* To get S&P 500 data
* Data cleaning and Data transforming

• Application of Fama-French three-factor model in S&P 500

* Fama-French in 2010-2017
* Trend analysis in 1980-2015
* Varying Samples
* Correlation of The factors and Top 20 and Bottom 20 returns

Slides : To get S&P 500 data

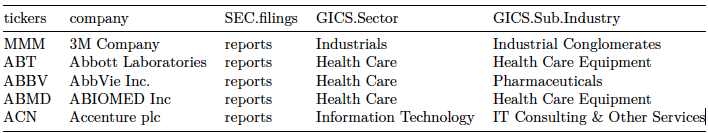
• Download the SP500 stocks from Yahoo finance

code: (Github/Code/Get\_SP500\_Data )

companies<-GetSP500Stocks() # get S&P 500 stocks with sectors information

head(Companies, n=5)

table



Slides : To get S&P 500 data

code: (Github/Code/Get\_SP500\_Data )

stocks<-BatchGetSymbols(tickers = companies$tickers,

first.date = "2010-01-01",

ast.date = "2017-12-31")

stock is a list that contains 2 dataframes: df.control, df.tickers

* df.control contains descriptive information like whether the download for the ticker is successful.
* df.tickers contains the downloaded price data. Each row is the price data for one ticker at one date, hence we need to process the data into a format easier to work with.

Slides : To get S&P 500 data

• loop over the good data: read one ticker at a time and merge into SP500.data

code: (Github/Code/Get\_SP500\_Data)

# Find out the good tickers / 465 stocks are "KEEP" for 2010 - 2017

good.tickers <- stocks$df.control$ticker [stocks$df.control$threshold.decision == "KEEP"]

for(i in 1:length(good.tickers)) {

# X is a temp dataframe that has 2 columns, 1st is date (for matching), 2nd is the actual data, e.g. closing price

# Choose relevant data by matching tickers

X <- data.frame(date = stocks$df.tickers$ref.date[stocks$df.tickers$ticker == good.tickers[i]],

stocks$df.tickers$price.adjusted[stocks$df.tickers$ticker == good.tickers[i]])

colnames(X)[2] <- stocks$df.tickers$ticker[stocks$df.tickers$ticker == good.tickers[i]]

# missing dates will have NA by default

SP500.data <- merge.data.frame(SP500.data, X, by = "date", all.x = TRUE)

}

write.csv(SP500.data, "SP500\_price.adjusted\_2010-2017.csv") # write the processed data to CSVs

Slides: Data cleaning and Data transforming

• Convert downloaded daily data to monthly price data series into XTS series

code: (Github/Code/Get\_SP500\_Data )

library(quantmond)

temp <- xts(SP500.data$AMZN, order.by = as.POSIXct(SP500.data$date))

temp.monthly <- monthlyReturn(temp)

Slides: Data cleaning and Data transforming

• Remove stocks with NAs in the series.

We need to remove NAs for using the monthlyReturn() function. Most NAs are due to data not available on the starting date of the series, e.g. the company has not IPO yet.

Here we face choices:

1. Remove all columns with NAs, then all remaining stocks could have the regression in the same period, i.e. with the same number of observations. (2010-2017)
2. Dynamically frame the data based on the available non-NA data points, but then some stocks in the regression analysis will have fewer observations. (1980-2015 every 5 years case)

Application of Fama-French three-factor model in S&P 500

Slides: Fama-French in 2010-2017

code: Github/Code/SP500-FF-Regression.R

the following code are Latex already (SPL\_Fama\_French.Rmd)

```{r message=F, warning=F}

library(quantmod)

# Read SP500 daily data and convert date column to date format

SP500.data <- read.csv("Data/SP500\_price.adjusted\_2010-2017.csv")

SP500.data$date <- as.Date(SP500.data$date)

# Select 2010 - 2017 range

Stock.Prices.Daily <- SP500.data[SP500.data$date>="2010-01-01" &

SP500.data$date<="2017-12-31",-1]

# Current FF3 till 201803, monthly

FF3 <- read.csv("Data/original/FF3.csv")

FF <- FF3[FF3$X >= 201001 & FF3$X <= 201712,]

# Convert series to XTS for using quantmod's monthlyReturn function

Stock.Prices.Daily <- xts(Stock.Prices.Daily[,-1],

order.by = as.POSIXct(Stock.Prices.Daily$date))

# Number of stocks to start with

ncol(Stock.Prices.Daily)

# Remove stocks with NAs in the series, otherwise monthly Return will not work properly

Stock.Prices.Daily <- Stock.Prices.Daily[, colSums(is.na(Stock.Prices.Daily)) == 0]

# Apply monthlyReturn function to each column (it seems it converts only one column at a time)

Stock.Prices.Monthly <- do.call(cbind, lapply(Stock.Prices.Daily, monthlyReturn))

# Stock.Prices.Monthly <- na.omit(Stock.Prices.Monthly)

colnames(Stock.Prices.Monthly) <- colnames(Stock.Prices.Daily)

# Number of stocks left

ncol(Stock.Prices.Monthly)

```

As in this example, we start with 465 stocks and remove 23 stocks with incomplete data.

```{r}

Results <- list()

for(i in 1:ncol(Stock.Prices.Monthly)) {

RiRF <- Stock.Prices.Monthly[,i] - FF$RF

Regression <- lm(RiRF ~ FF$Mkt.RF + FF$SMB + FF$HML)

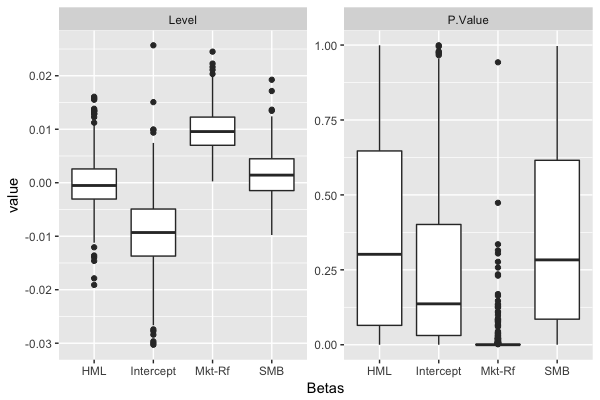
Results[[i]] <- summary(Regression)

}

```

Slides: Fama-French in 2010-2017

plot / FF3 Betas



Slides: Fama-French in 2010-2017

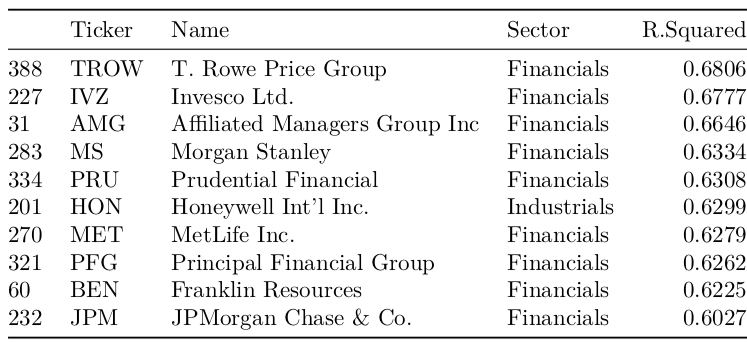
Goodness of fit

```{r}

# sort with R2 from largest to smallest, get top 10

kable(head(R2[order(R2$R.Squared, decreasing = T),], n=10), digits = 4)

```



Slides: Trend analysis in 1980-2015

• loop over above codes to download data from 1980 - 2015, group every 5 yrs.

code: (Github/Code/Get\_SP500\_Data )

library(lubridate)

List.of.start.date <- seq(as.Date("1980/1/1"), as.Date("2016/1/1"), "years")

List.of.start.date <- List.of.start.date[year(List.of.start.date)%%5==0]

Download.Stat <- data.frame(Data = List.of.start.date)

temp <- vector()

for(i in 1:(length(List.of.start.date)-1))

{

start.date <- as.Date(List.of.start.date[i])

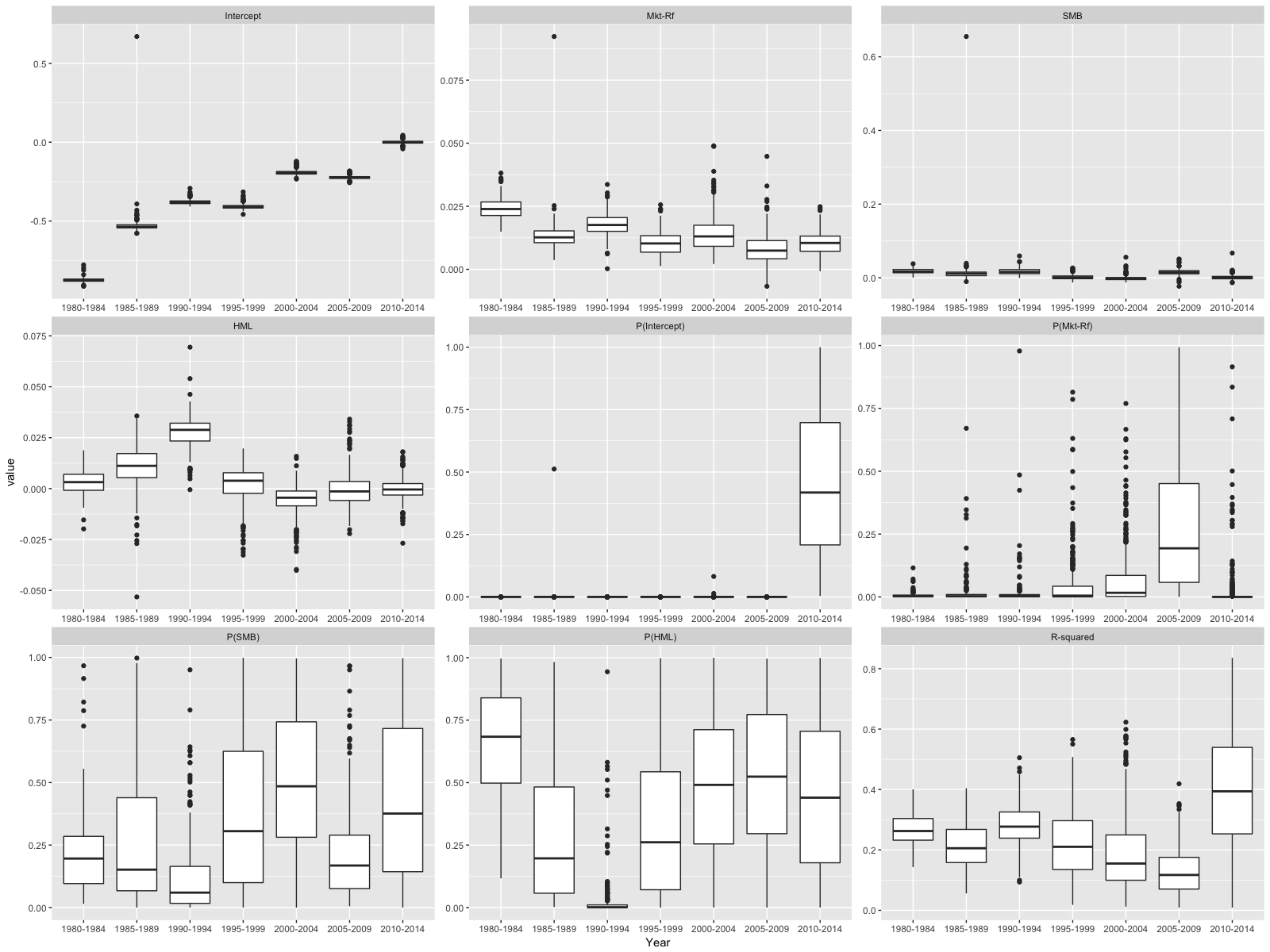
end.date <- as.Date(List.of.start.date[i+1])-1

…

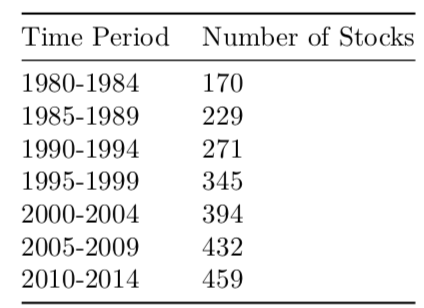
}

Slides: Trend analysis in 1980-2015

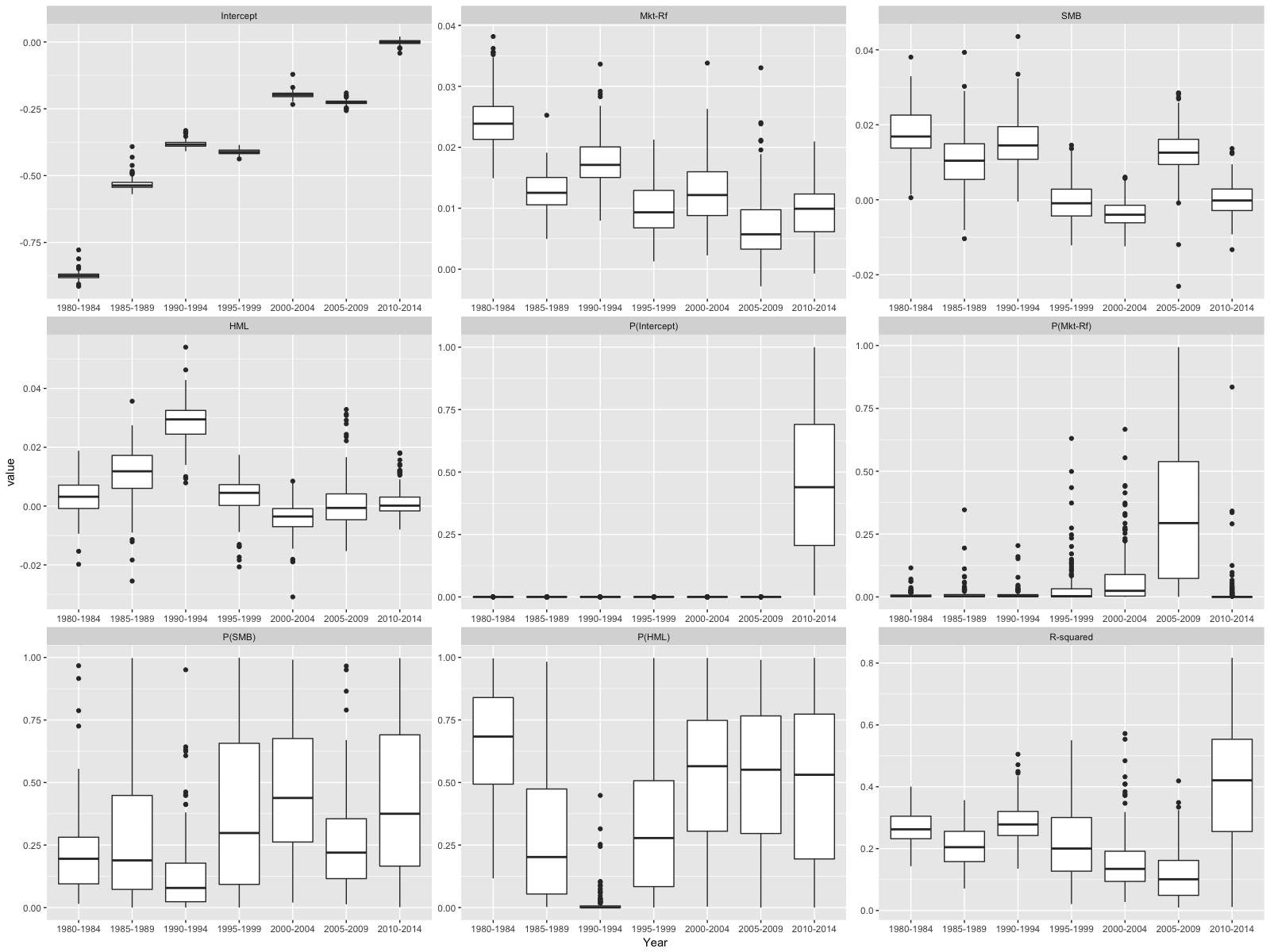
Plots/SP500-1980-2015.PNG

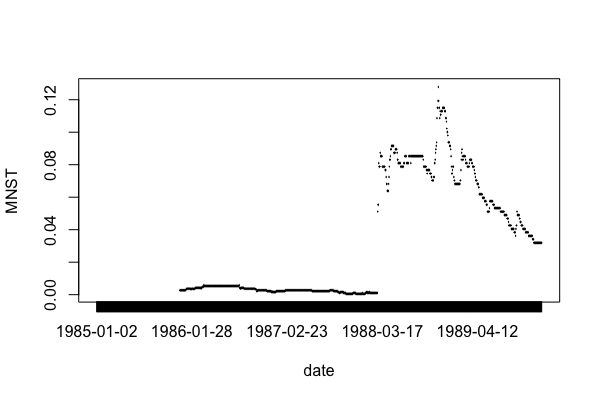


Slides: Varying Samples



Plots/SP500-1980-2015-168-stock.PNG





Slides: Correlation of The factors and Top 20 and Bottom 20 returns

Plots/SP500-FF3-2010-2017.png

