

Financial_statement_analysis_with_Morningstar_data

Export key ratios of Johnson & Johnson from Morningstar(<http://financials.morningstar.com/ratios/r.html?t=JNJ>
(<http://financials.morningstar.com/ratios/r.html?t=JNJ>)) Read the morning star data

```
dat <- read.table(file="JNJ Key Ratios.csv", header=F, sep=',',
                  dec = '.', stringsAsFactors=F, quote = "\"" ,
                  fill = TRUE , skip = 2)
```

The structure of the key ratio data.

```
str(dat)

## 'data.frame':    102 obs. of  12 variables:
##  $ V1 : chr   "" "Revenue USD Mil" "Gross Margin %" "Operating Income USD Mil" ...
##  $ V2 : chr   "2005-12" "50,514" "72.4" "13,009" ...
##  $ V3 : chr   "2006-12" "53,324" "71.8" "13,150" ...
##  $ V4 : chr   "2007-12" "61,095" "70.9" "13,661" ...
##  $ V5 : chr   "2008-12" "63,747" "71.0" "15,988" ...
##  $ V6 : chr   "2009-12" "61,897" "70.2" "15,590" ...
##  $ V7 : chr   "2010-12" "61,587" "69.5" "16,527" ...
##  $ V8 : chr   "2011-12" "65,030" "68.7" "16,153" ...
##  $ V9 : chr   "2012-12" "67,224" "67.8" "15,869" ...
##  $ V10: chr   "2013-12" "71,312" "68.7" "18,377" ...
##  $ V11: chr   "2014-12" "74,331" "69.4" "20,959" ...
##  $ V12: chr   "TTM" "73,590" "69.3" "20,677" ...
```

Extract different sections of the financial data. This could apply to financial data of other companies on Morningstar.

```
financials <- dat[1:16, ]
profitability <- dat[c(18:27, 29:36),]
revenue <- dat[c(38, 40:48),]
net_income <- dat[c(38, 50:53),]
eps <- dat[c(38, 55:58),]
cash_flow <- dat[c(60:65),]
financial_health <- dat[c(67:92),]
efficiency_ratios <- dat[c(94:102),]
```

Clean up the profitability section.

```
# replace 1st row
profitability[1,] <- c("Date", 2005:2014, "TTM")
row.names(profitability) <- profitability[,1]
profitability <- profitability[,-1]
colnames(profitability) <- profitability[1,]
profitability <- profitability[-1,]
# remove the last column of TTM
profitability <- profitability[, 1:10]
```

Reorganize the data to make stacked area graph.

```
profitability[1:8,]
```

##	2005	2006	2007	2008	2009	2010	2011
## Revenue	100.00	100.00	100.00	100.00	100.00	100.00	100.00
## COGS	27.62	28.24	29.05	29.04	29.80	30.51	31.31
## Gross Margin	72.38	71.76	70.95	70.96	70.20	69.49	68.69
## SG&A	33.41	32.69	33.47	33.71	31.99	31.54	32.25
## R&D	12.50	13.36	12.57	11.89	11.29	11.11	11.61
## Other	0.72	1.05	2.54	0.28	1.73		
## Operating Margin	25.75	24.66	22.36	25.08	25.19	26.84	24.84
## Net Int Inc & Other	1.28	2.69	-0.62	1.48	0.27	0.68	-5.83
##	2012	2013	2014				
## Revenue	100.00	100.00	100.00				
## COGS	32.22	31.33	30.60				
## Gross Margin	67.78	68.67	69.40				
## SG&A	31.04	30.61	29.54				
## R&D	11.40	11.47	11.67				
## Other	1.73	0.81					
## Operating Margin	23.61	25.77	28.20				
## Net Int Inc & Other	-3.11	-4.08	-0.53				

```
m <- matrix(0, nrow=50, ncol=3)
df <- data.frame(m)
colnames(df) <- c("Year", "OperatingMarginOrExpense", "PercentageOfSales")
# Force type to be character
df <- data.frame("Year"=numeric(50), "OperatingMarginOrExpense"=character(50), "PercentageOfSales"=numeric(50))

# add the first column "Year"
year <- sapply(2005:2014, function(x){rep(x, 5)})
year[1:length(year)]
```

```
## [1] 2005 2005 2005 2005 2005 2006 2006 2006 2006 2006 2006 2007 2007 2007 2007
## [15] 2007 2008 2008 2008 2008 2008 2008 2009 2009 2009 2009 2009 2010 2010 2010
## [29] 2010 2010 2011 2011 2011 2011 2011 2012 2012 2012 2012 2012 2013 2013 2013
## [43] 2013 2013 2013 2014 2014 2014 2014 2014
```

```
df[, "Year"] <- year[1:length(year)]
# fill the 2nd column
df[,2] <- rep(c("COGS", "SG&A", "R&D", "Other", "Operating Margin"), 10)
df[,2]
```

```
## [1] "COGS" "SG&A" "R&D"
## [4] "Other" "Operating Margin" "COGS"
## [7] "SG&A" "R&D" "Other"
## [10] "Operating Margin" "COGS" "SG&A"
## [13] "R&D" "Other" "Operating Margin"
## [16] "COGS" "SG&A" "R&D"
## [19] "Other" "Operating Margin" "COGS"
## [22] "SG&A" "R&D" "Other"
## [25] "Operating Margin" "COGS" "SG&A"
## [28] "R&D" "Other" "Operating Margin"
## [31] "COGS" "SG&A" "R&D"
## [34] "Other" "Operating Margin" "COGS"
## [37] "SG&A" "R&D" "Other"
## [40] "Operating Margin" "COGS" "SG&A"
## [43] "R&D" "Other" "Operating Margin"
## [46] "COGS" "SG&A" "R&D"
## [49] "Other" "Operating Margin"
```

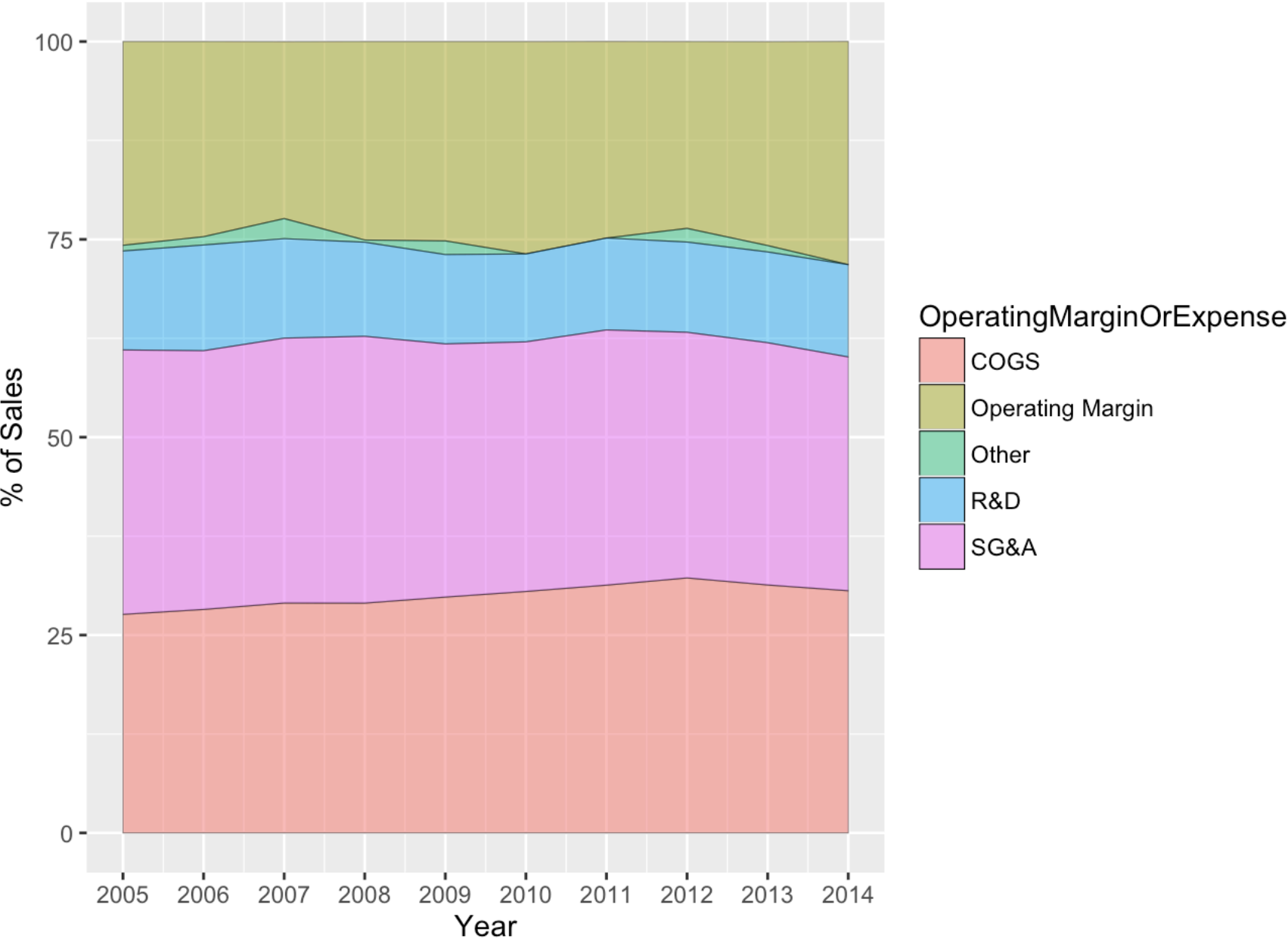
```
# last column
temp <- vector()
PercentageOfSales <- sapply(1:10, function(x) {temp <- c(temp, profitability[c(2,4,5,6,7), x])})
df[,3] <- as.numeric(PercentageOfSales)
# Replace the NA with 0
df[is.na(df)] <- 0
```

Now, we can plot. First, the stacked area figure for different expenses. It shows the relative proportion of those expenses.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

```
ggplot(df, aes(x=Year, y=PercentageOfSales, fill=OperatingMarginOrExpense)) +
  geom_area(colour="black", size=0.2, alpha=0.5) +
  scale_x_continuous(breaks= c(2005:2014), limits=c(2005, 2014)) +
  xlab("Year") + ylab("% of Sales")
```



Sales

```

# Clean data
# replace the first row with c("Date", 2005:2014, "TTM")
financials[1,] <- c("Date", 2005:2014, "TTM")

# row names to column 1
row.names(financials) <- financials[,1]

# remove the first column
financials <- financials[, -1]

# column names to the first row
colnames(financials) <- financials[1,]

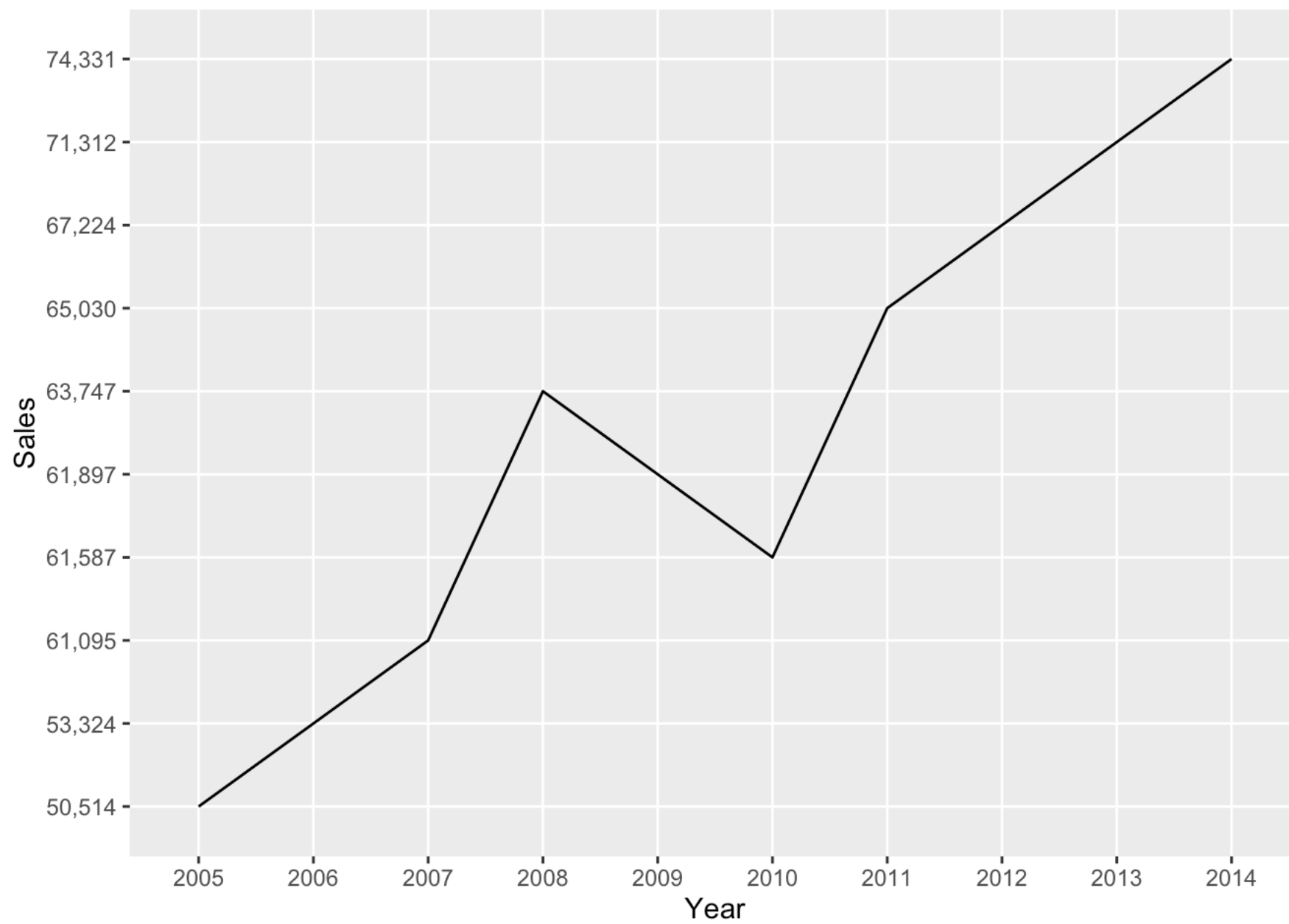
# remove the 1st row
financials <- financials[-1,]
# remove last column of financials
financials <- financials[,1:10]

# create a df2
df2 <- matrix(0, nrow=10, ncol=2)
df2 <- data.frame(df2)
# fill in df2 with sales data
df2[,2] <- as.character(financials[1,])
df2[,1] <- colnames(financials)

# add colnames
colnames(df2) <- c("Year", "Sales")

# add group=1 for factor type
ggplot(df2, aes(x=Year, y=Sales, group=1)) + geom_line()

```



```

# plot stacked bar figure of operating cash flow, including cap spending
# free cash flow
# data already cleaned as above

# extract data for the plot

df3 <- financials[12:13,]

# add a row with year
df3 <- rbind(as.numeric(colnames(df3)), df3)
# rename rows
rownames(df3) <- c("Year", "Capital_Spending", "Free_Cash_Flow")

# Transpose
df3 <- t(df3)

# transform data type to numeric first
(df3 <- gsub(",", "", df3))

```

```

##      Year   Capital_Spending Free_Cash_Flow
## 2005 "2005" "-2632"          "9245"
## 2006 "2006" "-2738"          "11510"
## 2007 "2007" "-3310"          "11939"
## 2008 "2008" "-3066"          "11906"
## 2009 "2009" "-2365"          "14206"
## 2010 "2010" "-2384"          "14001"
## 2011 "2011" "-2893"          "11405"
## 2012 "2012" "-2934"          "12462"
## 2013 "2013" "-3595"          "13819"
## 2014 "2014" "-3714"          "14757"

```

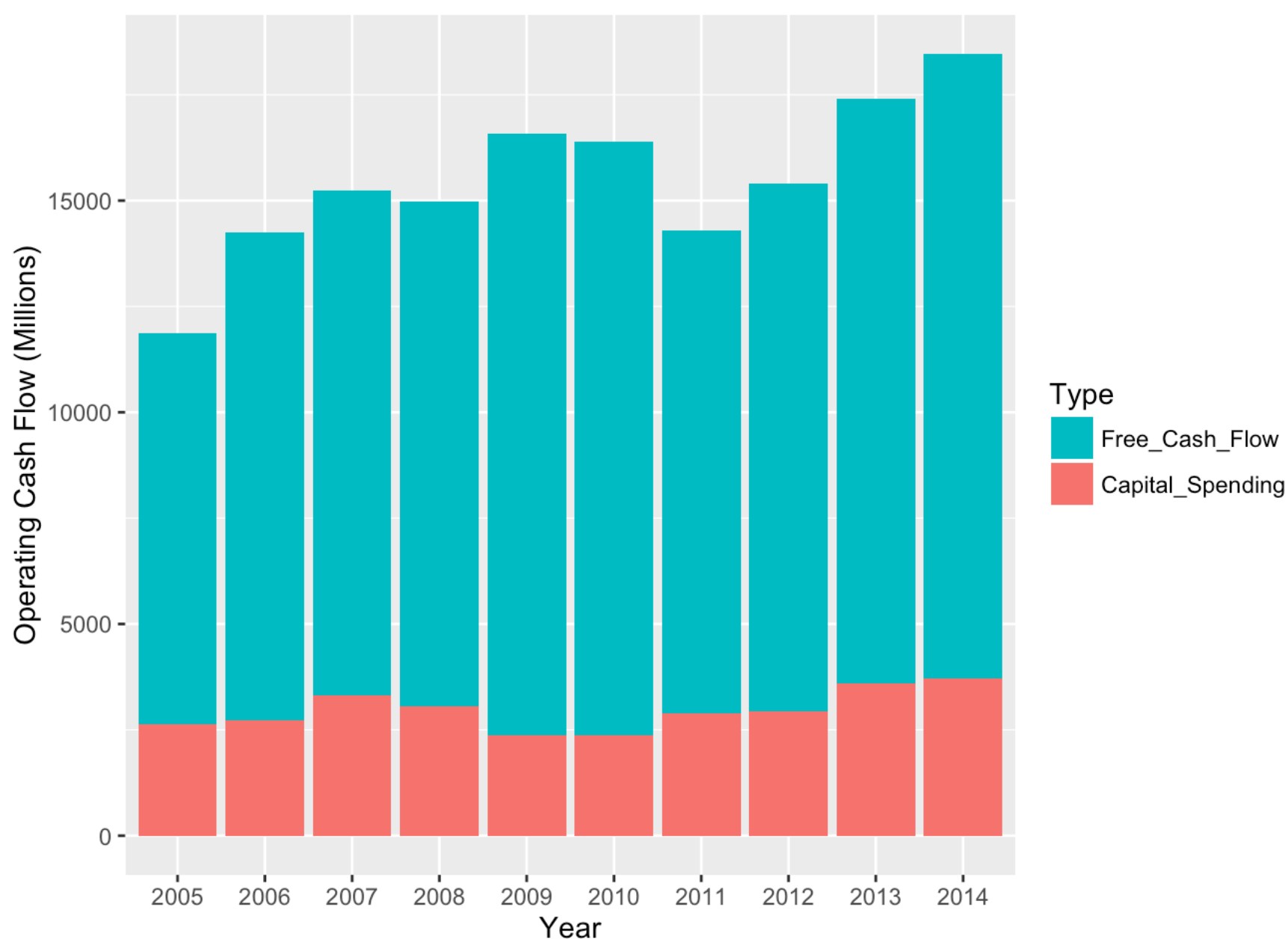
```

# negative to positive
df3 <- gsub("-", "", df3)
# transform into data frame
df3 <- data.frame(df3)
# remove the last row
df3 <- df3[1:10,]
# melt the data frame
library(reshape)
?melt
df4 <- melt(df3, id.vars="Year", measure.vars=c("Capital_Spending", "Free_Cash_Flow"), variable_name="Type")

# rename col3 into
colnames(df4)[3] <- "Operating_Cash_Flow"
# change to numeric, to avoid discrete y ticks
df4$Operating_Cash_Flow <- as.numeric(as.character(df4[,3]))

# let's do the stacked bar plot
ggplot(df4, aes(y=Operating_Cash_Flow, x=Year, fill=Type)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(reverse=TRUE)) +
  ylab("Operating Cash Flow (Millions)")

```



```
#theme(axis.text.y=element_blank()) +
#scale_y_discrete(limits=)
```

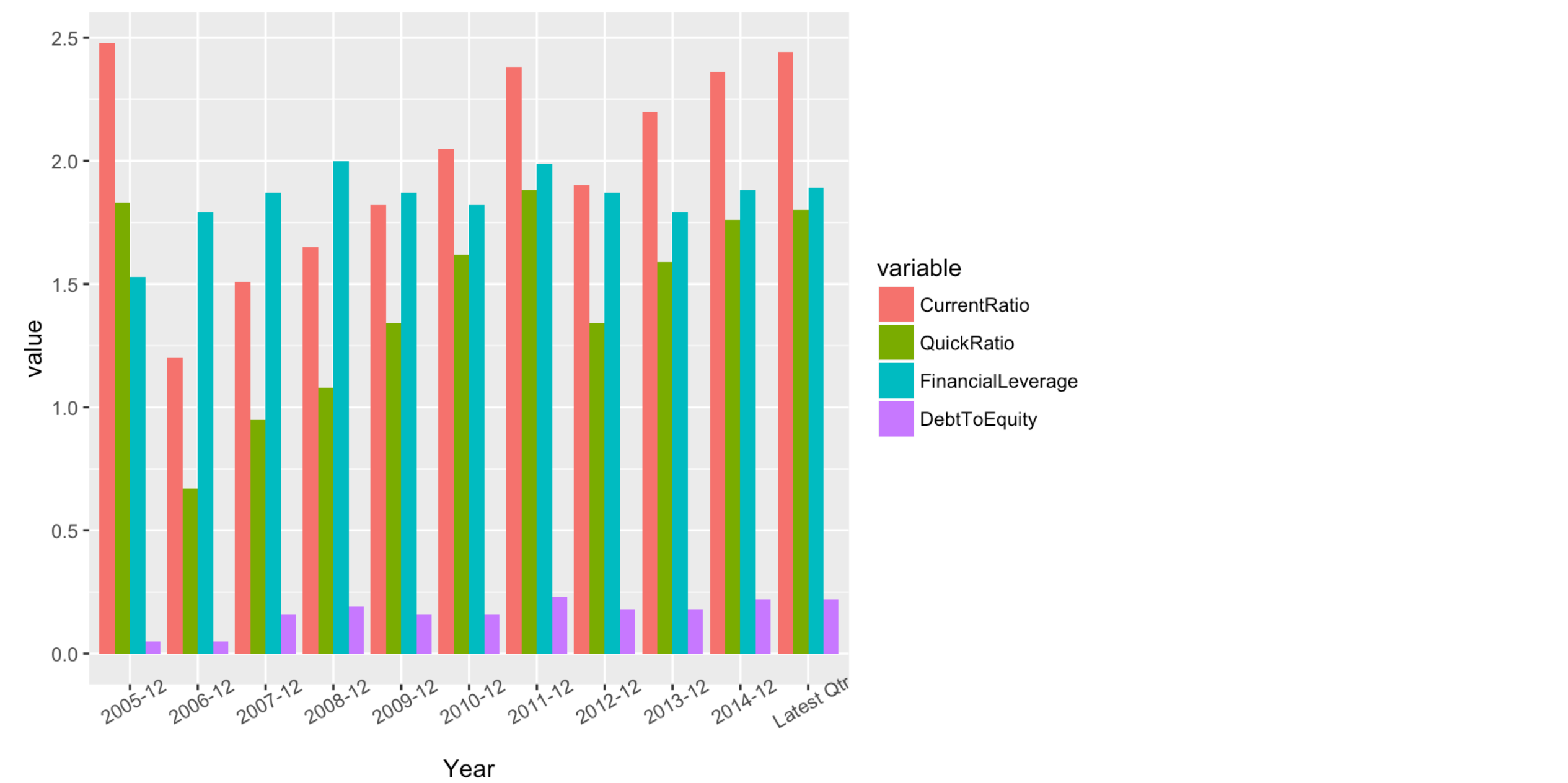
Liquidation ratios

```
# liquidity ratios
# in financial_health

df5 <- financial_health[22:26,]
df5 <- as.data.frame(t(df5))
colnames(df5) <- c("Year", "CurrentRatio", "QuickRatio", "FinancialLeverage", "DebtToEquity")
# remove first row
df5 <- df5[-1,]

# melt df5 for ggplot
df6 <- melt(df5, id.vars="Year", measure.vars=c("CurrentRatio", "QuickRatio", "FinancialLeverage", "DebtToEquity"),
            variable_names="Ratios")
# change to numeric, to avoid discrete y ticks
df6$value <- as.numeric(as.character(df6$value))

# multiple bar plot
ggplot(df6, aes(x=Year, y=value, fill=variable)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle=30))
```



```
#theme(axis.title.x=element_text(angle=45))
```

The whole process could be automated if we export data directly from the website, without downloading the .csv file locally.