S&P 500 Data

Arnav

2024-01-24

Data Cleaning S&P 500 Data in R

Import Section

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: ’zoo’
##
## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric
##
##
## ######################### Warning from ’xts’ package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R’s lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won’t work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you’re not using dplyr::lag(), or you can add #
## # conflictRules(’dplyr’, exclude = ’lag’) to your .Rprofile to stop           #
## # dplyr from breaking base R’s lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It’s protected by R’s namespace mechanism #
## # Set ‘options(xts.warn_dplyr_breaks_lag = FALSE)‘ to suppress this warning.  #
## #                                                                             #
## ###############################################################################
##
## Attaching package: ’xts’
##
## The following objects are masked from ’package:dplyr’:
##
##     first, last
##
## Loading required package: TTR
## Registered S3 method overwritten by ’quantmod’:
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(readr)
SPY <- read_csv("SPX.csv")
```

```
## Rows: 23323 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (6): Open, High, Low, Close, Adj Close, Volume
## date (1): Date
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
spec(SPY)
```

```
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   ‘Adj Close‘ = col_double(),
##   Volume = col_double()
## )
```

Checking if data valid/structure & attributes of data

How do we know this data *really* is complete? By checking these four things we can approximate or understand if this data set is usable to some degree.

```r
head(SPY)
```

```
## # A tibble: 6 x 7
##   Date        Open  High   Low Close ‘Adj Close‘ Volume
##   <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>  <dbl>
## 1 1927-12-30  17.7  17.7  17.7  17.7        17.7      0
## 2 1928-01-03  17.8  17.8  17.8  17.8        17.8      0
## 3 1928-01-04  17.7  17.7  17.7  17.7        17.7      0
## 4 1928-01-05  17.5  17.5  17.5  17.5        17.5      0
## 5 1928-01-06  17.7  17.7  17.7  17.7        17.7      0
## 6 1928-01-09  17.5  17.5  17.5  17.5        17.5      0
```

```
tail(SPY)
```

```
## # A tibble: 6 x 7
##   Date        Open  High   Low Close ‘Adj Close‘     Volume
##   <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>      <dbl>
## 1 2020-10-28 3342. 3342. 3269. 3271.       3271. 5129860000
## 2 2020-10-29 3277. 3341. 3260. 3310.       3310. 4903070000
## 3 2020-10-30 3294. 3305. 3234. 3270.       3270. 4840450000
## 4 2020-11-02 3296. 3330. 3280. 3310.       3310. 4310590000
## 5 2020-11-03 3336. 3389. 3336. 3369.       3369. 4220070000
## 6 2020-11-04 3406. 3486. 3405. 3443.       3443. 4783040000
```

```
str(SPY)
```

```
## spc_tbl_ [23,323 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date     : Date[1:23323], format: "1927-12-30" "1928-01-03" ...
##  $ Open     : num [1:23323] 17.7 17.8 17.7 17.5 17.7 ...
##  $ High     : num [1:23323] 17.7 17.8 17.7 17.5 17.7 ...
##  $ Low      : num [1:23323] 17.7 17.8 17.7 17.5 17.7 ...
##  $ Close    : num [1:23323] 17.7 17.8 17.7 17.5 17.7 ...
##  $ Adj Close: num [1:23323] 17.7 17.8 17.7 17.5 17.7 ...
##  $ Volume   : num [1:23323] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_date(format = ""),
##   ..   Open = col_double(),
##   ..   High = col_double(),
##   ..   Low = col_double(),
##   ..   Close = col_double(),
##   ..   ‘Adj Close‘ = col_double(),
##   ..   Volume = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
attributes(SPY)
```

Preparing Data to Clean

Let’s rename a column to change one of the names of the variables.

```
colnames(SPY)
```

```
## [1] "Date"      "Open"      "High"      "Low"       "Close"     "Adj Close"
## [7] "Volume"
```

```
SPY <- rename(SPY, "Adjusted Close" = `Adj Close`)
SPY <- rename(SPY, "Volume of Shares Traded" = `Volume`)
head(SPY)
```

```
## # A tibble: 6 x 7
##   Date        Open  High   Low Close `Adjusted Close` `Volume of Shares Traded`
##   <date>     <dbl> <dbl> <dbl> <dbl>            <dbl>                     <dbl>
## 1 1927-12-30  17.7  17.7  17.7  17.7             17.7                         0
## 2 1928-01-03  17.8  17.8  17.8  17.8             17.8                         0
## 3 1928-01-04  17.7  17.7  17.7  17.7             17.7                         0
## 4 1928-01-05  17.5  17.5  17.5  17.5             17.5                         0
## 5 1928-01-06  17.7  17.7  17.7  17.7             17.7                         0
## 6 1928-01-09  17.5  17.5  17.5  17.5             17.5                         0
```

This dataset has a column called volume, but this dataset has errors and some years show a volume of 0 (the total volume of shares traded was 0), which has to be incorrect.

```
SPYVol <- filter(SPY , `Volume of Shares Traded` != 0)
```

Now that we have all the data with actual volume, we need to fix the pricing. On some of these columns, the price of open, high, low, close, and adjusted close are all the same. Therefore, we need to get rid of these.

```
SPYVol <- SPYVol %>% mutate(daysReturn = (`Adjusted Close` - Open) / Open)
```

Now, let's update the data so it can be accessed per year for easier understanding.

```
SPYVol <- SPYVol %>% mutate(year = year(Date))

#yearly summary of returns
tapply(SPYVol$daysReturn , SPYVol$year, summary)
```

```
## $`1950`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0 
## 
## $`1951`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0 
## 
## $`1952`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0 
## 
## $`1953`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0 
## 
## $`1954`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0 
## 
```

```
## $‘1955‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1956‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1957‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1958‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1959‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1960‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1961‘
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
##
## $‘1962‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0617075 -0.0044764 -0.0002839 -0.0005555  0.0050299  0.0464865
##
## $‘1963‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0280648 -0.0018986  0.0007237  0.0006518  0.0037452  0.0164301
##
## $‘1964‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0125301 -0.0013666  0.0004969  0.0004869  0.0028915  0.0089353
##
## $‘1965‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0175777 -0.0016198  0.0003932  0.0003262  0.0029063  0.0138605
##
## $‘1966‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0224292 -0.0044806 -0.0003335 -0.0005160  0.0034752  0.0284341
##
## $‘1967‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0157189 -0.0022468  0.0008460  0.0008587  0.0040502  0.0197785
##
## $‘1968‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0159331 -0.0032169  0.0003036  0.0002732  0.0034922  0.0177939
##
## $‘1969‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.850e-02 -4.983e-03 -9.611e-05 -4.533e-04  4.081e-03  2.079e-02
##
## $‘1970‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2.647e-02 -5.310e-03  6.178e-05  6.811e-05  4.869e-03  4.901e-02
##
## $‘1971‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0152239 -0.0031586  0.0000000  0.0001055  0.0032535  0.0178235
##
## $‘1972‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0124458 -0.0027502  0.0006643  0.0005919  0.0040110  0.0142434
##
## $‘1973‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.028365 -0.007227 -0.001470 -0.000661  0.005406  0.029499
##
## $‘1974‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.026700 -0.010265 -0.001867 -0.001213  0.006215  0.045959
##
## $‘1975‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.019238 -0.005844  0.001106  0.001099  0.007506  0.024678
##
## $‘1976‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0161274 -0.0048063  0.0005028  0.0007414  0.0054993  0.0184818
##
## $‘1977‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0162561 -0.0042868 -0.0002124 -0.0004602  0.0031757  0.0186061
##
## $‘1978‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2.009e-02 -5.098e-03  1.080e-04  2.778e-05  4.679e-03  2.890e-02
##
## $‘1979‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0255872 -0.0035129  0.0006386  0.0005423  0.0043075  0.0205948
##
## $‘1980‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.028381 -0.005263  0.001982  0.000955  0.007551  0.025990
##
## $‘1981‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0240298 -0.0056623 -0.0002369 -0.0002531  0.0052421  0.0249327
##
```

```
## $‘1982‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0395504 -0.0060055 -0.0008833  0.0005650  0.0058049  0.0388279
##
## $‘1983‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0269049 -0.0044248  0.0007839  0.0006962  0.0057273  0.0266844
##
## $‘1984‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.821e-02 -5.453e-03 -7.612e-04  2.554e-05  4.021e-03  2.750e-02
##
## $‘1985‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0145530 -0.0034166  0.0009673  0.0009876  0.0043520  0.0228227
##
## $‘1986‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0480855 -0.0031507  0.0012284  0.0006978  0.0055057  0.0225527
##
## $‘1987‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.2046693 -0.0046926  0.0016055  0.0002862  0.0084701  0.0909936
##
## $‘1988‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0676116 -0.0047129  0.0005529  0.0004810  0.0054131  0.0357750
##
## $‘1989‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.061172 -0.003134  0.001356  0.000978  0.005511  0.027574
##
## $‘1990‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0302441 -0.0066072  0.0010624 -0.0002382  0.0050896  0.0317446
##
## $‘1991‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -3.659e-02 -3.696e-03 -8.927e-05  9.804e-04  6.206e-03  3.706e-02
##
## $‘1992‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.857e-02 -3.232e-03  3.627e-05  1.898e-04  3.600e-03  1.556e-02
##
## $‘1993‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2.389e-02 -2.796e-03  8.673e-05  2.911e-04  3.266e-03  1.925e-02
##
## $‘1994‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2.261e-02 -3.350e-03  1.059e-04 -5.543e-05  3.661e-03  2.084e-02
##
## $‘1995‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0154623 -0.0014958  0.0008994  0.0011780  0.0041730  0.0185221
##
## $‘1996‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0308269 -0.0031420  0.0006807  0.0007553  0.0057736  0.0194385
##
## $‘1997‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.068657 -0.005231  0.001855  0.001133  0.007938  0.051152
##
## $‘1998‘
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.068014 -0.005295  0.001403  0.001019  0.008281  0.050899
##
## $‘1999‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0280578 -0.0065622  0.0003326  0.0007725  0.0077149  0.0352662
##
## $‘2000‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0582779 -0.0084668 -0.0003791 -0.0003328  0.0080189  0.0476385
##
## $‘2001‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0492156 -0.0083404 -0.0006112 -0.0004793  0.0069410  0.0500986
##
## $‘2002‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0415361 -0.0120372 -0.0017818 -0.0008684  0.0079940  0.0573140
##
## $‘2003‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0352315 -0.0061217  0.0012766  0.0009865  0.0074034  0.0354266
##
## $‘2004‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0163204 -0.0040559  0.0006360  0.0003636  0.0046698  0.0163653
##
## $‘2005‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0167205 -0.0043794  0.0005588  0.0001367  0.0042985  0.0197362
##
## $‘2006‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0183263 -0.0034058  0.0009833  0.0005442  0.0036856  0.0216142
##
## $‘2007‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0346455 -0.0032138  0.0007000  0.0002047  0.0058135  0.0292220
##
## $‘2008‘
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -8.723e-02 -1.184e-02 -4.746e-05 -1.419e-03  8.633e-03  1.079e-01
##
```

```
## $`2009`
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## -0.052281 -0.006546  0.001868  0.001186  0.008947  0.065531 
## 
## $`2010`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0322846 -0.0037529  0.0007431  0.0005544  0.0055632  0.0333787 
## 
## $`2011`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0659335 -0.0062704  0.0009630  0.0001161  0.0076249  0.0466869 
## 
## $`2012`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0243001 -0.0039200  0.0003108  0.0005334  0.0045728  0.0240881 
## 
## $`2013`
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## -0.022960 -0.002965  0.001562  0.000949  0.004826  0.025403 
## 
## $`2014`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0228813 -0.0024617  0.0007584  0.0003381  0.0040879  0.0209264 
## 
## $`2015`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -3.661e-02 -4.850e-03 -3.618e-04 -5.974e-05  5.042e-03  3.618e-02 
## 
## $`2016`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0315618 -0.0030643  0.0004694  0.0002855  0.0036726  0.0244139 
## 
## $`2017`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0148362 -0.0011945  0.0003616  0.0003381  0.0022626  0.0080969 
## 
## $`2018`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -3.874e-02 -3.546e-03  5.094e-05 -5.893e-04  4.353e-03  4.425e-02 
## 
## $`2019`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -0.0219713 -0.0018904  0.0008368  0.0006442  0.0037878  0.0232830 
## 
## $`2020`
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
## -5.710e-02 -6.745e-03  1.512e-03 -1.534e-05  6.755e-03  5.488e-02
```

1950-1961 had no returns. We will remove that from the data and start from 1962.

```r
SPYVol <- filter(SPYVol, year > 1961)
```

We will now convert the daysReturn column into percentages, (and round for visual aid).

```
SPYVol <- SPYVol %>%
  select(Date, Open, High, Low, Close, `Adjusted Close`, daysReturn, `Volume of Shares Traded`, year) %
  mutate(
           daysReturn = `daysReturn` * 100, #percentage
           daysReturn = round(daysReturn , digits = 4)) #round

SPYVol
```

```
## # A tibble: 14,814 x 9
##    Date        Open  High   Low Close ‘Adjusted Close‘ daysReturn
##    <date>     <dbl> <dbl> <dbl> <dbl>            <dbl>      <dbl>
##  1 1962-01-02  71.6  72.0  70.7  71.0             71.0    -0.825
##  2 1962-01-03  71.0  71.5  70.4  71.1             71.1     0.240
##  3 1962-01-04  71.1  71.6  70.4  70.6             70.6    -0.689
##  4 1962-01-05  70.6  70.8  69.3  69.7             69.7    -1.39
##  5 1962-01-08  69.7  69.8  68.2  69.1             69.1    -0.775
##  6 1962-01-09  69.1  69.9  68.8  69.2             69.2     0.0434
##  7 1962-01-10  69.2  69.6  68.6  69.0             69.0    -0.275
##  8 1962-01-11  69.0  69.5  68.6  69.4             69.4     0.595
##  9 1962-01-12  69.4  70.2  69.2  69.6             69.6     0.346
## 10 1962-01-15  69.6  70.0  69.1  69.5             69.5    -0.201
## # i 14,804 more rows
## # i 2 more variables: ‘Volume of Shares Traded‘ <dbl>, year <dbl>
```

Analysis

Trends for returns on a daily, monthly, and yearly basis.

```
summary(SPYVol$daysReturn)
```

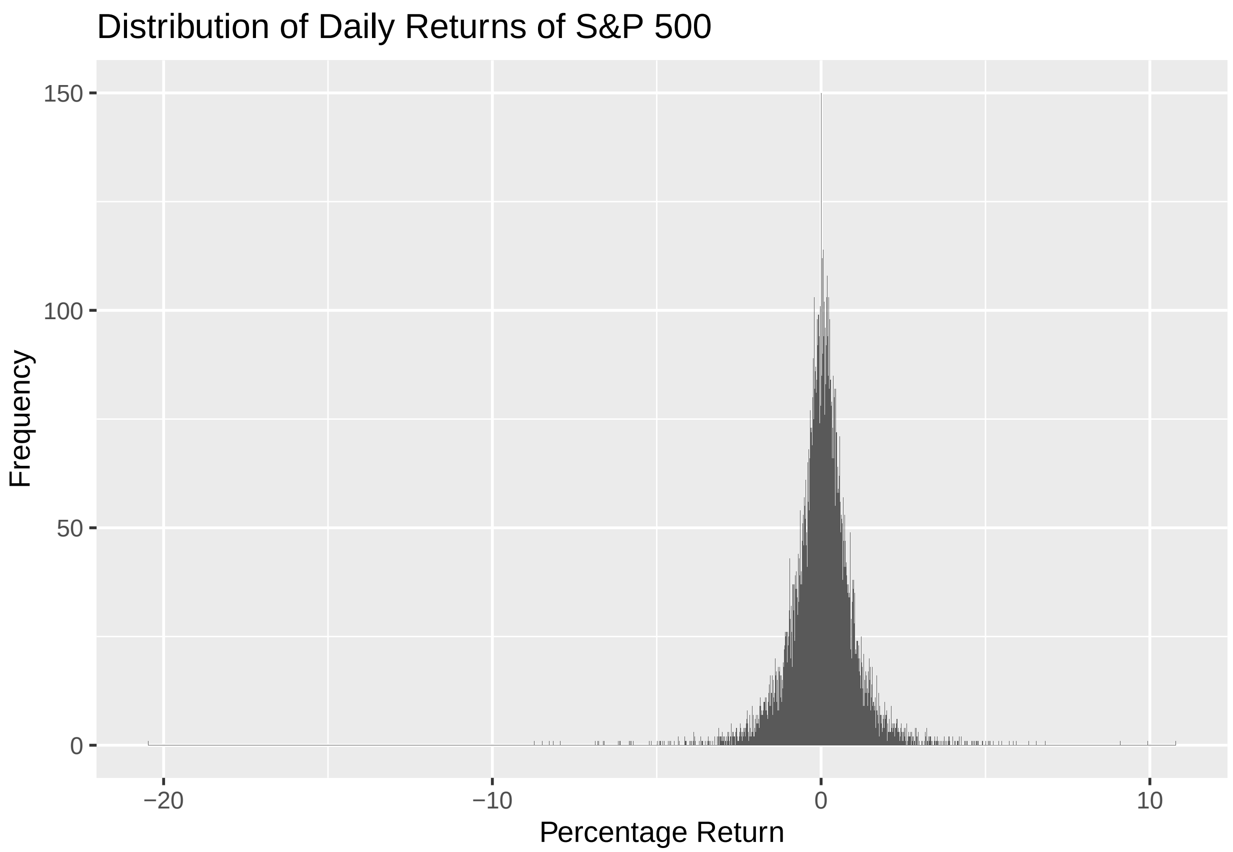
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -20.46690  -0.41350   0.04160   0.02854   0.49378  10.78900
```

```
sd(SPYVol$daysReturn)
```

```
## [1] 0.9810564
```

Distribution of daysReturn

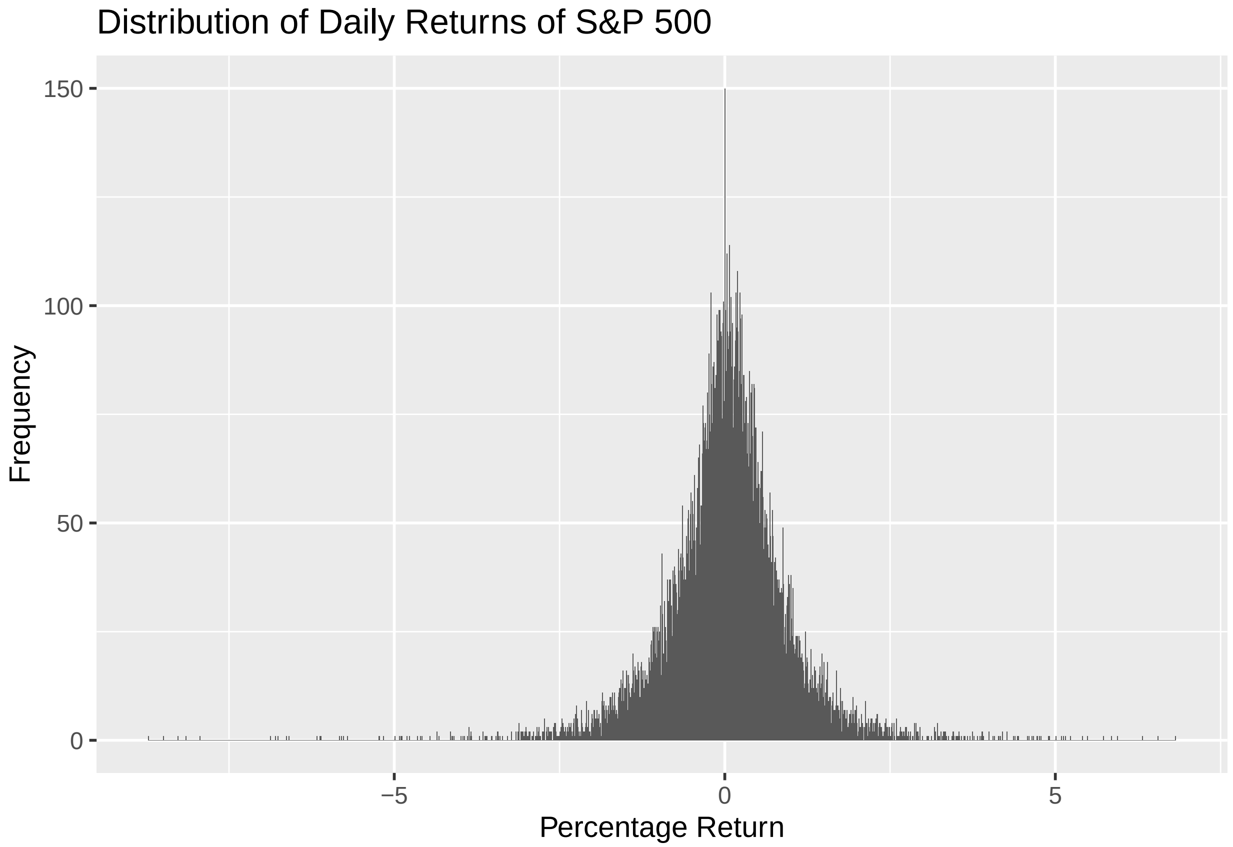
```
daysReturnDistribution <- ggplot(SPYVol, aes(x = daysReturn)) +
  geom_histogram(binwidth = 0.01) +
  labs(y = "Frequency", x = "Percentage Return", title = "Distribution of Daily Returns of S&P 500")
daysReturnDistribution
```



Removing Outliers

```
SPYVolNoOutliers <- filter(SPYVol, daysReturn > -10 & daysReturn < 7.5)

daysReturnDistributionNoOutliers <- ggplot(SPYVolNoOutliers, aes(x = daysReturn)) +
  geom_histogram(binwidth = 0.01) +
  labs(y = "Frequency", x = "Percentage Return", title = "Distribution of Daily Returns of S&P 500")
daysReturnDistributionNoOutliers
```



Best & Worst Days

Although most days are positive, we can see a couple of terrible days. Hence, lets calculate the worst 30 days (not consecutive) in S&P history.

```
worst30 <- SPYVol %>%
  top_n(-30, daysReturn) %>%
  arrange(daysReturn)
worst30
```

```
## # A tibble: 30 x 9
##    Date        Open  High   Low Close ‘Adjusted Close‘ daysReturn
##    <date>     <dbl> <dbl> <dbl> <dbl>            <dbl>      <dbl>
##  1 1987-10-19  283.  283.  225.  225.             225.     -20.5 
##  2 2008-10-15  995.  995.  904.  908.             908.      -8.72
##  3 2008-09-29 1209. 1209. 1106. 1106.            1106.      -8.49
##  4 1987-10-26  248.  248.  227.  228.             228.      -8.27
##  5 2008-12-01  889.  889.  816.  816.             816.      -8.15
##  6 2008-10-09  988. 1005.  909.  910.             910.      -7.94
##  7 1997-10-27  942.  942.  877.  877.             877.      -6.87
##  8 1998-08-31 1027. 1033.  957.  957.             957.      -6.80
##  9 1988-01-08  261.  261.  243.  243.             243.      -6.76
## 10 2008-11-20  806.  821.  748.  752.             752.      -6.63
## # i 20 more rows
## # i 2 more variables: ‘Volume of Shares Traded‘ <dbl>, year <dbl>
```

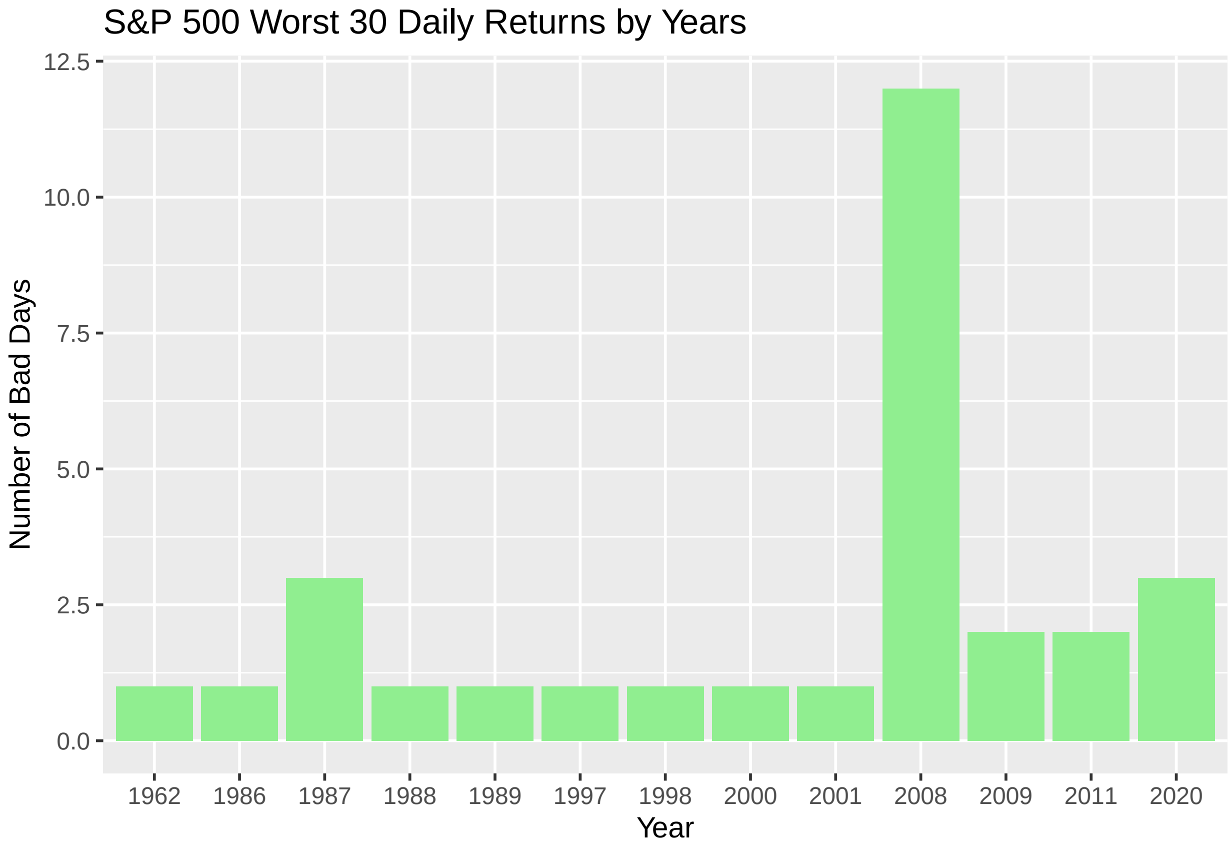
Year by Year:

```
worstDayByYear <- worst30 %>%
  count(year(Date)) %>%
  arrange(desc(n)) %>%
  rename("Year" = "year(Date)", "Count" = "n") %>%
  mutate(Year = as.character(Year), Type = "Worst")
worstDayByYear
```

```
## # A tibble: 13 x 3
##    Year  Count Type
##    <chr> <int> <chr>
##  1 2008     12 Worst
##  2 1987      3 Worst
##  3 2020      3 Worst
##  4 2009      2 Worst
##  5 2011      2 Worst
##  6 1962      1 Worst
##  7 1986      1 Worst
##  8 1988      1 Worst
##  9 1989      1 Worst
## 10 1997      1 Worst
## 11 1998      1 Worst
## 12 2000      1 Worst
## 13 2001      1 Worst
```

Graphing the worst 30 days looks like this:

```
worst30 <- ggplot(worstDayByYear, aes(x = Year, y = Count)) +
  geom_col(fill = "lightgreen") +
  labs(y = "Number of Bad Days", title = "S&P 500 Worst 30 Daily Returns by Years")
worst30
```



Lets be positive! Now we can look at the best 30 in S&P history.

```
best30 <- SPYVol %>%
  top_n(30, daysReturn) %>%
  arrange(daysReturn)
best30
```

```
## # A tibble: 30 x 9
##    Date         Open   High    Low  Close `Adjusted Close` daysReturn
##    <date>      <dbl>  <dbl>  <dbl>  <dbl>            <dbl>      <dbl>
##  1 2001-04-05 1103.  1151.  1103.  1151.            1151.        4.37
##  2 2020-03-24 2344.  2450.  2344.  2447.            2447.        4.39
##  3 2018-12-26 2363.  2468.  2347.  2468.            2468.        4.43
##  4 2008-10-20  944.   985.   944.   985.             985.        4.44
##  5 2011-08-11 1121.  1186.  1121.  1173.            1173.        4.58
##  6 1974-10-09   64.8   68.2   63.7   67.8             67.8       4.60
##  7 1962-05-29   55.5   58.3   53.1   58.1             58.1       4.65
##  8 2011-08-09 1120.  1173.  1102.  1173.            1173.        4.67
##  9 2008-09-30 1114.  1168.  1114.  1166.            1166.        4.72
## 10 2002-10-15  841.   881.   841.   881.             881.        4.73
## # i 20 more rows
## # i 2 more variables: `Volume of Shares Traded` <dbl>, year <dbl>
```

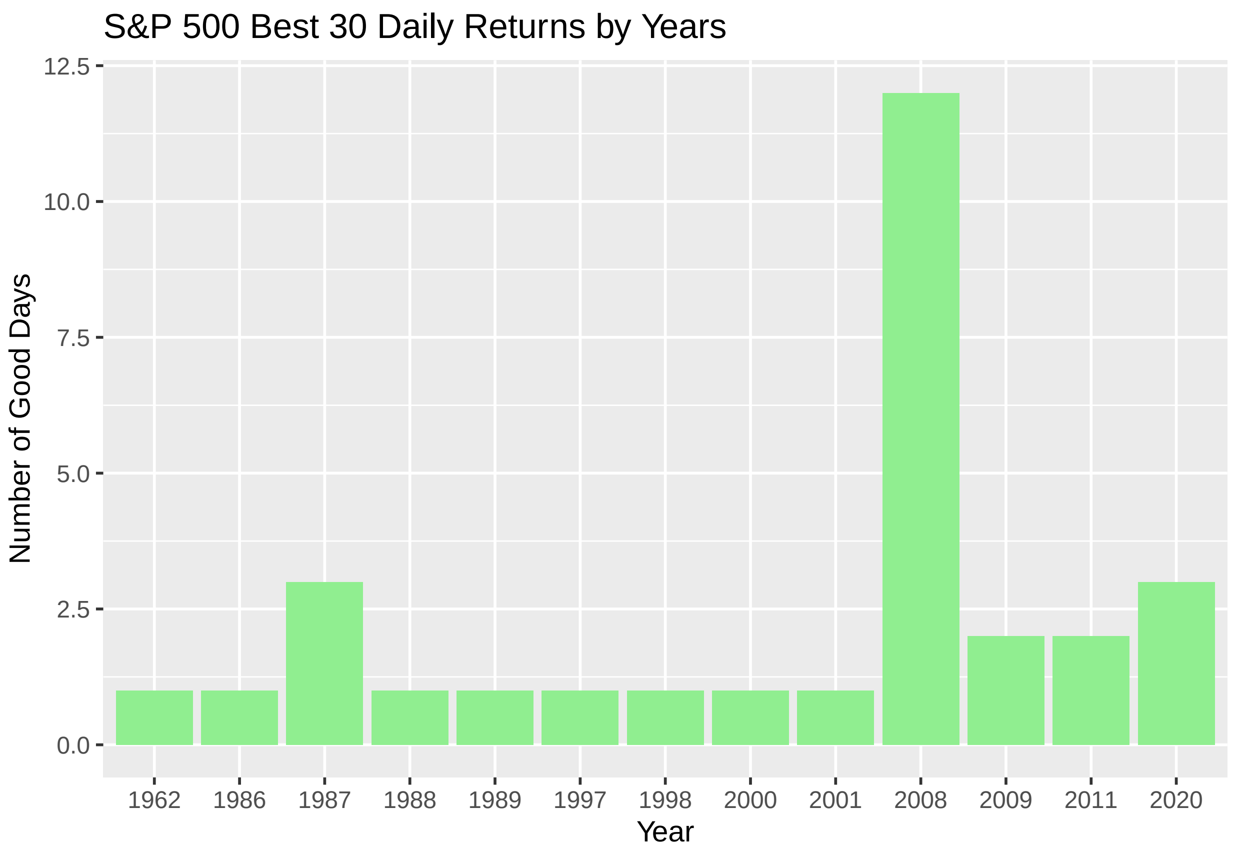
Year By Year

```
bestDayByYear <- best30 %>%
  count(year(Date)) %>%
  arrange(desc(n)) %>%
  rename("Year" = "year(Date)", "Count" = "n") %>%
  mutate(Year = as.character(Year), Type = "Best")
bestDayByYear
```

```
## # A tibble: 14 x 3
##    Year  Count Type
##    <chr> <int> <chr>
##  1 2008      8 Best
##  2 1987      3 Best
##  3 2002      3 Best
##  4 2020      3 Best
##  5 2001      2 Best
##  6 2009      2 Best
##  7 2011      2 Best
##  8 1962      1 Best
##  9 1970      1 Best
## 10 1974      1 Best
## 11 1997      1 Best
## 12 1998      1 Best
## 13 2000      1 Best
## 14 2018      1 Best
```

Graphing the best 30 days looks like this:

```
best30 <- ggplot(worstDayByYear, aes(x = Year, y = Count)) +
  geom_col(fill = "lightgreen") +
  labs(y = "Number of Good Days", title = "S&P 500 Best 30 Daily Returns by Years")
best30
```

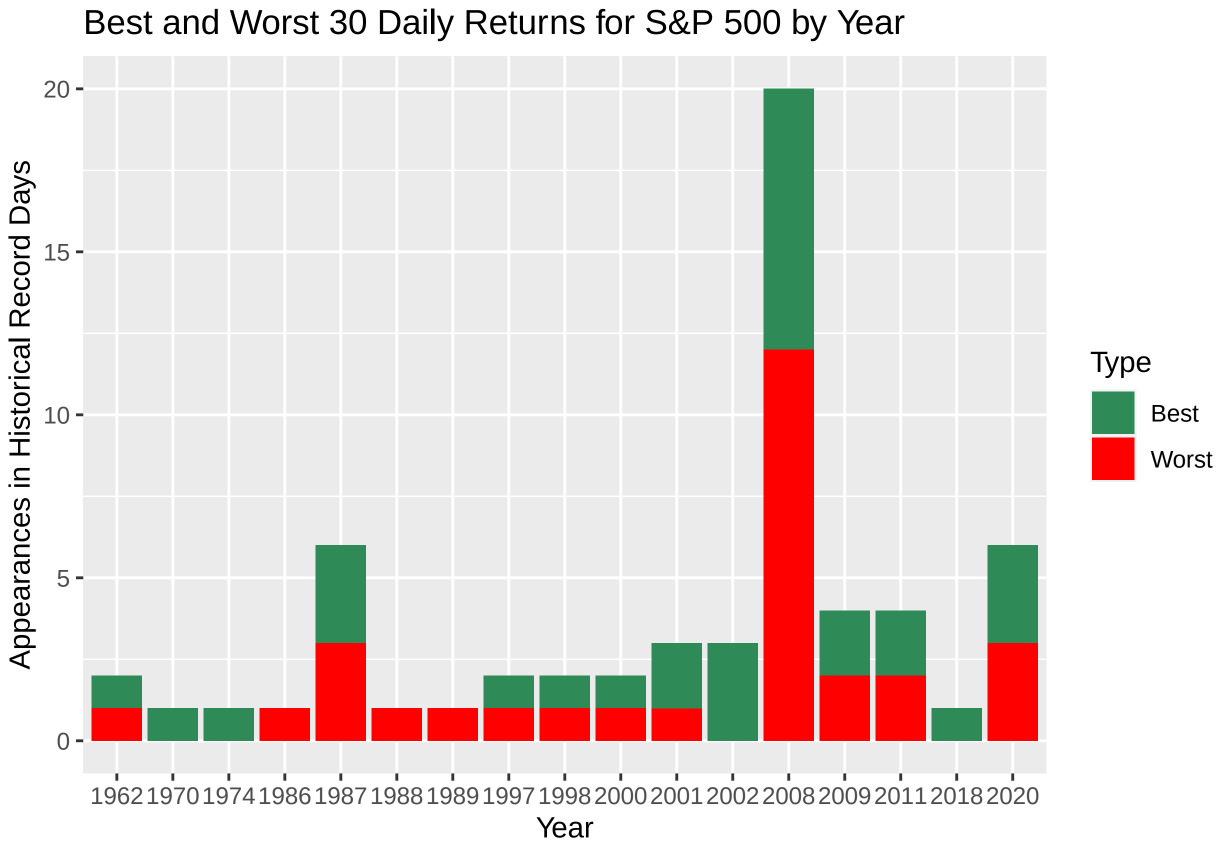


Combining the tables can give us an idea of how many good and bad days are in each year.

```
combinedDays <- merge(x = bestDayByYear, y = worstDayByYear,
                      by = c("Year", "Count", "Type"), all = TRUE)
combinedDays
```

```
##    Year Count  Type
## 1  1962     1  Best
## 2  1962     1 Worst
## 3  1970     1  Best
## 4  1974     1  Best
## 5  1986     1 Worst
## 6  1987     3  Best
## 7  1987     3 Worst
## 8  1988     1 Worst
## 9  1989     1 Worst
## 10 1997     1  Best
## 11 1997     1 Worst
## 12 1998     1  Best
## 13 1998     1 Worst
## 14 2000     1  Best
## 15 2000     1 Worst
## 16 2001     1 Worst
## 17 2001     2  Best
## 18 2002     3  Best
## 19 2008     8  Best
## 20 2008    12 Worst
## 21 2009     2  Best
## 22 2009     2 Worst
## 23 2011     2  Best
## 24 2011     2 Worst
## 25 2018     1  Best
## 26 2020     3  Best
## 27 2020     3 Worst
```

```
combined20 <- ggplot(combinedDays, aes(x = Year, y = Count, fill = Type)) +
  geom_col() +
  scale_fill_manual(values = c("seagreen" , "red")) +
  labs(y = "Appearances in Historical Record Days", title = "Best and Worst 30 Daily Returns for S&P 50
combined20
```



```
#2008 has 12 record worst days and 8 record best days
```