Data607-MajorAssignment-Tidyverse

Vinayak Kamath

2/8/2020

An Example using TidyVerse packages - ggplot2 and dplyr and using tesla-stock-data-from-2010-to-2020 Data set from Kaggle [*https://www.kaggle.com/timoboz/tesla-stock-data-from-2010-to-2020*]:

Pls load ggplot2 and dplyr package using install.packages("ggplot2") and install.packages("dplyr")

```
# Loading the tidyverse readr/ggplot2/dplyr package:
library(readr)
library(ggplot2)
library(dplyr)
```

Load the library and Read the data using readr:

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Loading the tesla-stock-data-from-2010-to-2020:
theUrl <- "https://raw.githubusercontent.com/kamathvk1982/Data607-MajorAssignment-Tidyverse/master/TSLA
tesla.hist.df <- read_csv(theUrl)
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )
```

```r
# Adding two new columns to get the current day gain/loss and the gain/loss percentage:
tesla.hist.df$GainLoss <-  tesla.hist.df$Close - tesla.hist.df$Open
tesla.hist.df$GainLossPercent <- (tesla.hist.df$GainLoss/tesla.hist.df$Open)*100

# Sample rows from the dataset:
tail(tesla.hist.df)
```

```
## # A tibble: 6 x 9
##   Date        Open  High   Low Close `Adj Close` Volume GainLoss GainLossPercent
##   <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>  <dbl>    <dbl>           <dbl>
## 1 2020-01-27  542.  564.  539.  558.        558. 1.36e7    16.0            2.96
## 2 2020-01-28  568.  577.  558.  567.        567. 1.18e7    -1.59          -0.280
## 3 2020-01-29  576.  590.  567.  581.        581. 1.78e7     5.30           0.921
## 4 2020-01-30  632.  651.  618   641.        641. 2.90e7     8.39           1.33
## 5 2020-01-31  640   653   633.  651.        651. 1.57e7    10.6            1.65
## 6 2020-02-03  674.  786.  674.  780         780  4.71e7   106.            15.8
```

Package Selected : ggplot2 and dplyr

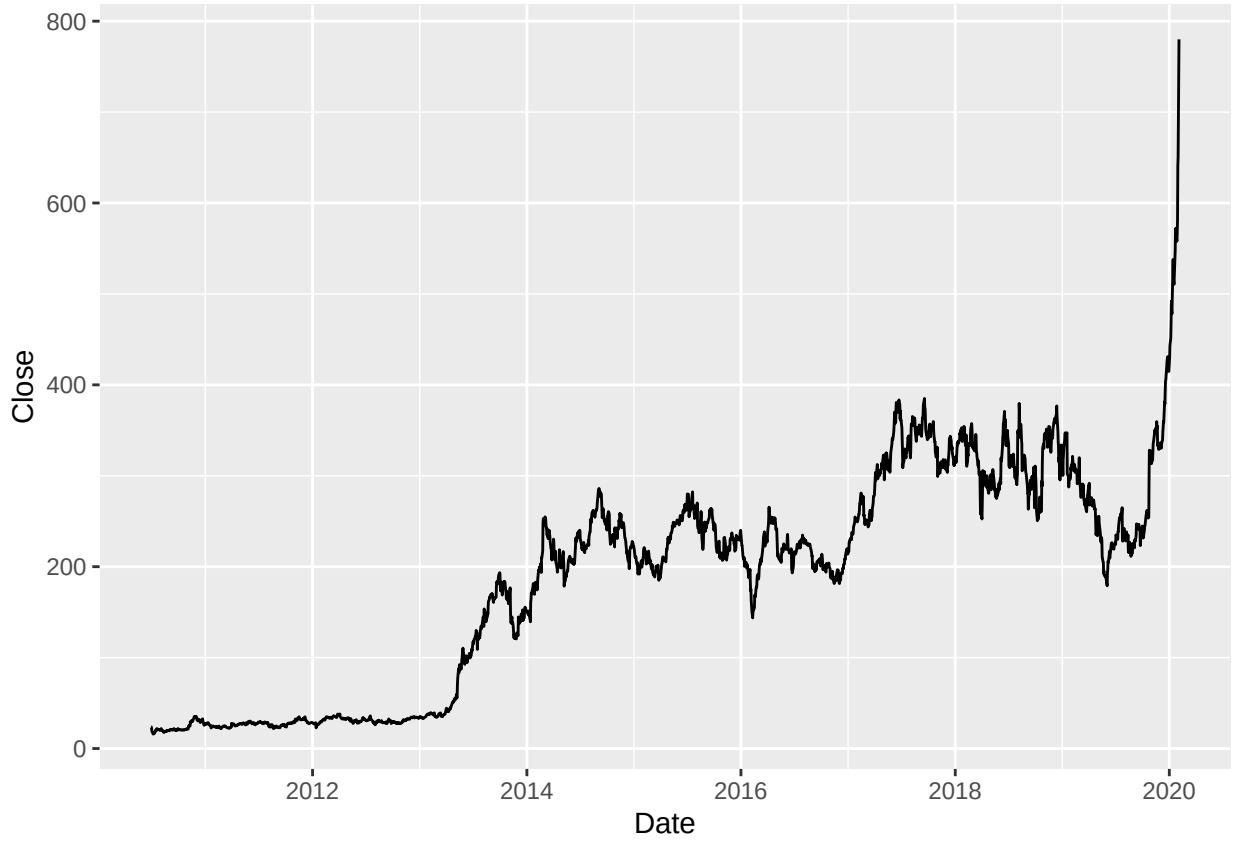
Capability 1: ggplot2 geom_line Usage:

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Demo:

ggplot2 geom_line to show the stock price movement over the years for Tesla shares.

```r
ggplot(tesla.hist.df, aes(Date, Close)) +
  geom_line()
```



Capability 2: dplyr filter Usage:

dplyr filter helps in filtering of data based on one or more conditions.

Demo:

dplyr filter to show the days when the stock price for Tesla moved by over 15% (profit or loss) in one day.

```
tesla.hist.df %>%
  filter(GainLossPercent >= 15 | GainLossPercent <= -15 ) %>%
  arrange(desc(GainLossPercent))
```

```
## # A tibble: 6 x 9
##   Date        Open  High   Low Close `Adj Close` Volume GainLoss GainLossPercent
##   <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>  <dbl>    <dbl>           <dbl>
## 1 2010-06-29  19    25    17.5  23.9        23.9 1.88e7     4.89            25.7
## 2 2010-11-10  24.5  30.0  24.0  29.4        29.4 3.06e6     4.88            19.9
## 3 2020-02-03 674.  786.  674.  780         780   4.71e7   106.              15.8
## 4 2010-07-02  23    23.1  18.7  19.2        19.2 5.14e6    -3.80           -16.5
## 5 2010-07-06  20    20    15.8  16.1        16.1 6.87e6    -3.89           -19.4
## 6 2012-01-13  28.4  28.5  22.6  22.8        22.8 5.50e6    -5.61           -19.8
```

Capability 3: dplyr group by and summarise Usage:

dplyr group by and summarise helps in getting aggregated data from the given data set for one or more columns.

Demo:

dplyr group by and summarise to show the yearly minimum and maximum stock price close and arranging it in descending order of movement in a year.

```
tesla.hist.df %>%
  group_by(format(as.Date(tesla.hist.df$Date), "%Y")) %>%
  summarise(min_close = min(Close) , max_close = max(Close)) %>%
  arrange(desc( (max_close-min_close)/min_close)*100 )
```

```
## # A tibble: 11 x 3
##    `format(as.Date(tesla.hist.df$Date), "%Y")` min_close max_close
##    <chr>                                         <dbl>     <dbl>
##  1 2013                                           32.9     193.
##  2 2019                                          179.      431.
##  3 2010                                           15.8      35.5
##  4 2014                                          139.      286.
##  5 2016                                          144.      265.
##  6 2020                                          430.      780
##  7 2017                                          217.      385
##  8 2012                                           22.8      38.0
##  9 2011                                           21.8      34.9
## 10 2015                                          185       282.
## 11 2018                                          251.      380.
```
