

R for Beginners - R for Finance Code File-1

This R code book written by [Rohit Dhankar](https://github.com/RohitDhankar) . GitHub - <https://github.com/RohitDhankar>

Code and Data > <https://github.com/RohitDhankar/R-Beginners-Online-Virtual-Learning-Session>

Good practice to keep track of current Working Directory , list all Objects in R ENVIRONMENT - specially so when committing changes to Git or any other version control Remote directory.

R for Finance

```
library("forecast");
library("ggplot2");
library("ggfortify")

##
## Attaching package: 'ggfortify'

## The following object is masked from 'package:forecast':
##
##      gglagplot

library("tseries")

# Set Seed -- ensure reproducible results

set.seed(123)

infy_df <- read.csv("~/Desktop/R_Own/R_Finance/DATA_Files/INFY.csv")
str(infy_df)

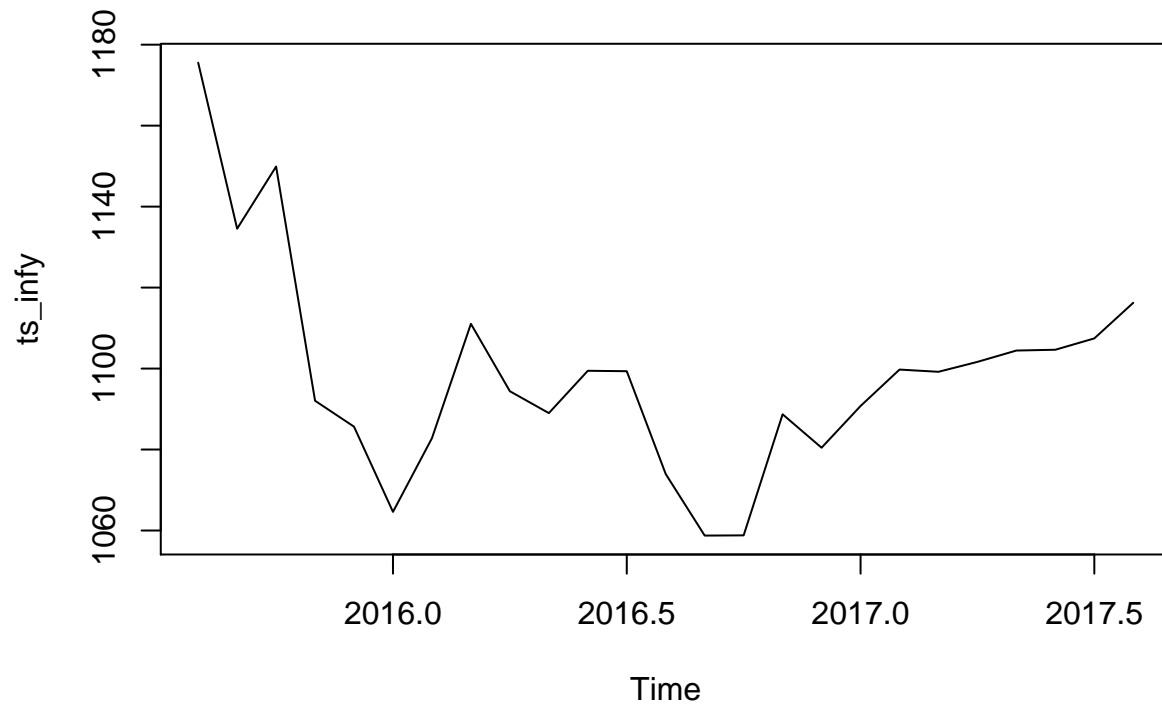
## 'data.frame':   494 obs. of  1 variable:
##  $ Close.Price: num  1176 1135 1150 1092 1086 ...

summary(infy_df)

##      Close.Price
##      Min.       : 911.1
##      1st Qu.: 982.0
##      Median :1049.1
##      Mean    :1060.4
##      3rd Qu.:1134.1
##      Max.     :1267.6

#start_date <- infy_df$Date[1] ## [1] 19-Aug-2015
#len_df<-length(infy_df$Date)
#end_date <-infy_df$Date[len_df] ## [1] 17-Aug-2017

# Convert DF to TS
#ts_infy <-ts(infy_df, start = start_date, end = end_date, frequency = 12) #
ts_infy <-ts(infy_df, start = c(2015,8), end = c(2017,8), frequency = 12) #
plot(ts_infy)
```



```
#
#
#

# CHECK --- Could i have done this faster in Python ??

# Func - seq(as.Date ...)
# REFER -- https://stat.ethz.ch/R-manual/R-devel/library/base/html/seq.Date.html

#
# #
# head(mdf,n=20)
# #
# length(mdf$cty_name)
# #
# summary(mdf) # Summary of DF
# #
# str(mdf) # Structure of DF
# # #
# #Explicitly check the Class and Typeof
# class(mdf$cty_name);class(mdf$country_name);class(mdf$var_dates)
# #
# typeof(mdf$cty_name);typeof(mdf$country_name);typeof(mdf$var_dates)
```

Section -2

Also many other sources from the net.

Rohit Dhankar claims no copyright to any of this code.

```
# Start the clock!
# ptm <- proc.time()
#
# vec_gross_sale <- p_sale_count_rnd*p_sale_cost_rnd
#
# summary(vec_gross_sale)
#
# proc.time() - ptm
#
#
# As seen below in our case
# ELAPSED time - 1st 0.011 , 2nd - 0.012
# Thus the WALL CLOCK or REAL / ELAPSED
# timings are almost same .
#
# The USER TIME and SYSTEM TIME's in our case
# add upto -
# 1st - 0.008
# 2nd - 0.012

# Thus it would seem we are better off
# with Vector Multiplication

# But we also need to consider
# once we have the "vec_gross_sale"
# we will need to add it to out "mdf"

# Kindly also note the Timings will
# differ for each system - also for each run
# of the chunk of code on same sys

# Definition of user Time --- The 'user time' is the CPU time
# charged for execution of user instructions of the calling process.
#
# REFER- https://stat.ethz.ch/R-manual/R-devel/library/base/html/proc.time.html

# Now to multiply TWO Columns of the DF
# Also called COLUMNAR VECTORS

# Again start the clock!
# ptm <- proc.time()
#
# mdf$gross_sale<- mdf$p_sale_count_rnd*mdf$p_sale_cost_rnd
#
# proc.time() - ptm
# #
# str(mdf)
# #
# summary(mdf)
```

```
# #  
# write.csv(mdf,file="Mkt_DATA_Files/mdf.csv")  
# ## Writes to Sub Directory - DATA_Files  
# #
```