# Jumping into Data Science with R & Python

## Aim of this Quick Read:

This quick read aims to help jump start, all those data enthusiasts towards data science, as, a brain teaser. After reading it and practicing for once, this brain teaser assists the user in their day-to-day R / Python programming needs and in future as a ready reference for basics.

## Installation of R:

- **Windows:** An R-**.**.exe file is available at http://cran.r-project.org/bin/windows/base/
  - Download latest stable and double click *.exe file (select 64 bit) and accept default installation.
- **Linux:** In Ubuntu, R is recognized as 'GNU S' - A language and environment for statistical computing and graphics:
  - Please refer to page http://cran.r-project.org/bin/linux/ for other Linux options
  - E.g. first, add the mirror[1] entry in your /etc/apt/sources.list
    deb http://cran.stat.nus.edu.sg/bin/linux/ubuntu lucid/
    - ❖ Then, just run a) sudo apt-get update and b) sudo apt-get install r-base

## Where to Find & How to Start "R":

- **Windows:** Generally, during installation, R installs a shortcut on your desktop, double click it, otherwise at location "xx:\Program Files\R\R-xxxx\bin\x64\" click "Rgui" (better create a desktop shortcut).
- **Linux:** Just "R' to start the program and type "q()" to quit.

## Installation of Python:

- I prefer Miniconda, refer to the following document for the quick guidance on installation https://docs.conda.io/en/latest/miniconda.html
- If you are interested on creating and managing virtual environments using miniconda, refer to the following https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

## Very Important:

- Both Python & R are case-sensitive: "A" is not equal to "a".
- "#" used for comments

[1] Different R mirrors are available at page http://cran.r-project.org/mirrors.html, choose a location close to you.

## Everything in R & Python are objects:

- All entities in R, including functions and data structures, exist as 'objects'.
- Just type, the name of an object at the command prompt, the contents of the object are printed out.

## What else do you need, a common IDE for both R/Python:

My choice would be RStudio (choose version >= 1.4.xxx, you can also choose VSCode.

### Then what hinders us to jump start into DS using R/Python:

- Business/Research Questions (or) Data
- Please remember all data science is towards solving either business or research questions, towards the same data scientist use data to help business organizations in making objective decisions; without data or certain level of data maturity in the organization, applications of data sciences is not complete.
- Now, question is to start with data and business problem or to start with business problem and data; it is always chick and egg problem.

### Let's move with the below data, just to deep dive into data science experience:

| Car.Sales | Sale.Price | Mileage | Advt.Expenses |
|---:|---:|---:|---:|
| 4662 | 298346 | 21 | 27443 |
| 3726 | 300723 | 19 | 19050 |
| 984 | 300805 | 22 | 79935 |
| 1272 | 299389 | 15 | 42939 |
| 1665 | 299574 | 18 | 68879 |
| 1694 | 299271 | 21 | 77331 |
| 3375 | 298004 | 14 | 17513 |
| 3841 | 300659 | 18 | 54784 |
| 1610 | 299319 | 27 | 84655 |
| 2743 | 299663 | 17 | 15627 |
| 4272 | 300179 | 24 | 44814 |
| 4844 | 299403 | 22 | 65791 |
| 3274 | 300230 | 18 | 77945 |
| 3572 | 300003 | 15 | 75743 |
| 2711 | 299328 | 18 | 24162 |
| 1440 | 301089 | 23 | 31911 |
| 1746 | 300319 | 18 | 79503 |
| 3268 | 300740 | 28 | 98139 |
| 2618 | 300127 | 16 | 61292 |
| 1311 | 298794 | 16 | 65546 |

| | | | |
|---|---|---|---|
| 3197 | 300425 | 23 | 11828 |
| 2539 | 299703 | 20 | 68260 |
| 2888 | 300231 | 15 | 31072 |
| 540 | 299634 | 17 | 52659 |
| 1352 | 301315 | 13 | 41480 |
| 1302 | 300385 | 15 | 79744 |
| 4938 | 300568 | 23 | 81840 |
| 1080 | 299675 | 14 | 77598 |
| 4330 | 301235 | 20 | 19780 |
| 592 | 299698 | 22 | 87346 |
| 680 | 299696 | 17 | 46756 |
| 4165 | 299277 | 19 | 50783 |
| 2493 | 300741 | 22 | 30866 |
| 2145 | 299509 | 22 | 30391 |
| 4332 | 299061 | 22 | 23032 |
| 662 | 302285 | 21 | 38373 |
| 4867 | 299775 | 20 | 91250 |
| 4219 | 299798 | 20 | 96594 |
| 2018 | 300013 | 20 | 30302 |
| 2406 | 300990 | 23 | 76111 |
| 4004 | 299270 | 32 | 22283 |
| 4069 | 299895 | 21 | 18726 |
| 2243 | 301346 | 17 | 29922 |
| 1953 | 299757 | 22 | 89962 |
| 2030 | 300261 | 19 | 64727 |
| 2344 | 300439 | 18 | 77544 |
| 1065 | 300844 | 18 | 40422 |
| 794 | 300615 | 16 | 42262 |
| 4341 | 299439 | 21 | 71387 |
| 4531 | 298558 | 22 | 13183 |
| 3345 | 299697 | 16 | 91607 |
| 1410 | 300598 | 20 | 66835 |

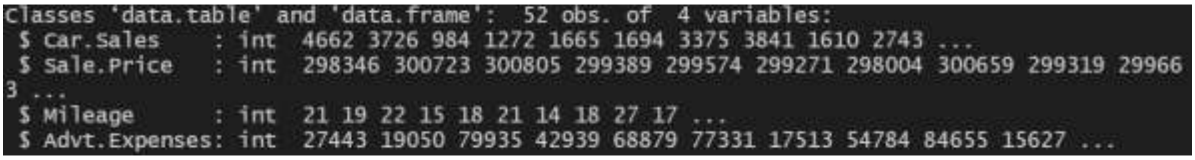
Now, above, we have certain car sales related data, like first column about number of car units sold, its respective average sales price, mileage it provides, and advertisement expenses spent on the car model.

Looking at above data, what business questions can be solved? Can it be what driving sales of the car, does its mileage or advertisement spent?

What else business question we can solve; can we say car sales are seasonal? Please be careful, does we have data to answer this. Enough questions let's gets hands dirty.

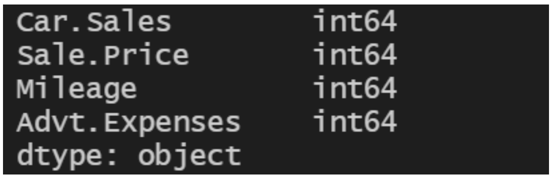
First and foremost, need to read this data to either your R/Python environment. I have stored this file in *.csv format. For all those who are new to R/Python, please remember both language strength for data science lies in the libraries/packages available. Below is the code:

R Code:
r_carsales <- data.table::fread("………………/Car_Sales.csv")
str(r_carsales)



```
Classes 'data.table' and 'data.frame':  52 obs. of  4 variables:
 $ Car.Sales     : int  4662 3726 984 1272 1665 1694 3375 3841 1610 2743 ...
 $ Sale.Price    : int  298346 300723 300805 299389 299574 299271 298004 300659 299319 29966
3 ...
 $ Mileage       : int  21 19 22 15 18 21 14 18 27 17 ...
 $ Advt.Expenses: int  27443 19050 79935 42939 68879 77331 17513 54784 84655 15627 ...
```

Python Code:
import pandas as pd
py_carsales = pd.read_csv("…………………/Car_Sales.csv")
py_carsales.dtypes



```
Car.Sales        int64
Sale.Price       int64
Mileage          int64
Advt.Expenses    int64
dtype: object
```

From above, one can infer that both R/Python read them as integers, now how does here data science helps; let's start with basic statistics, though all four columns have been read by both languages as integers, does all values in the columns can be considered same numbers? No, Sales are in units, price is in currency, mileage in miles, and advertisement expenses in currency thousands. So, can we directly analyse them with whatever analysis we want to explore, let say, does sales depend on price, mileage and advt. expenses directly, if by how much?

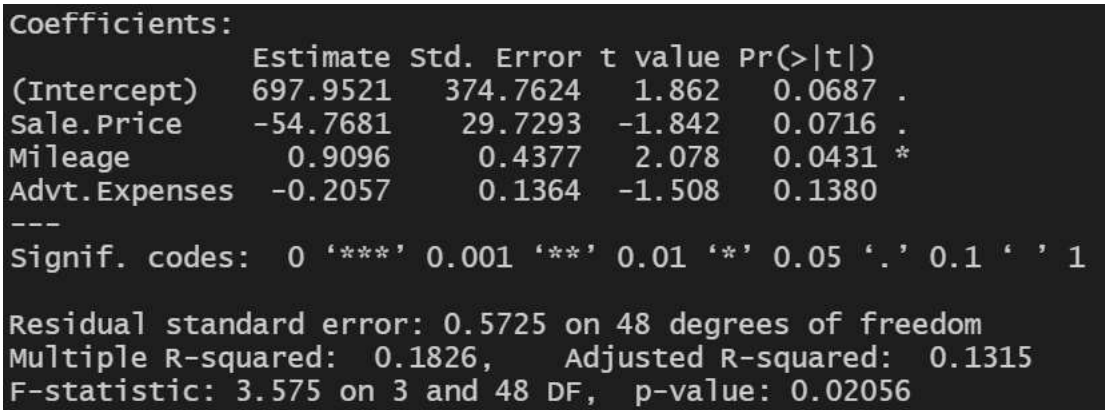
Answer is no, as they are not in same standard units for comparison. So, data science is about:

1) Understanding data properly - here your statistics helps – for instance, how to bring them same standard units,
2) Checking whether data suits business questions raised or not – again statistics help here through summary statistics,
3) Obtained data is enough for modelling or not – any missing or missing of valid information,
4) Which model to choose – ML/Statistics will help - validating your insights once modelling is done, and
5) Finally, explain or visualize them in simple terms – plenty of options exists.

Let's jump into modelling directly as we don't have any missing and all valid values in the data provided (it is 99.99% true with examples, but, in reality, we will have always several data related issues foremost is missing data), and go deeper how data science helps us better with insights from data.

R Code:

summary(lm(Car.Sales ~ ., data=yourdf))



```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)    697.9521   374.7624   1.862   0.0687 .
Sale.Price     -54.7681    29.7293  -1.842   0.0716 .
Mileage          0.9096     0.4377   2.078   0.0431 *
Advt.Expenses   -0.2057     0.1364  -1.508   0.1380  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5725 on 48 degrees of freedom
Multiple R-squared:  0.1826,    Adjusted R-squared:  0.1315 
F-statistic: 3.575 on 3 and 48 DF,  p-value: 0.02056
```

Python Code:

import statsmodels.api as sm

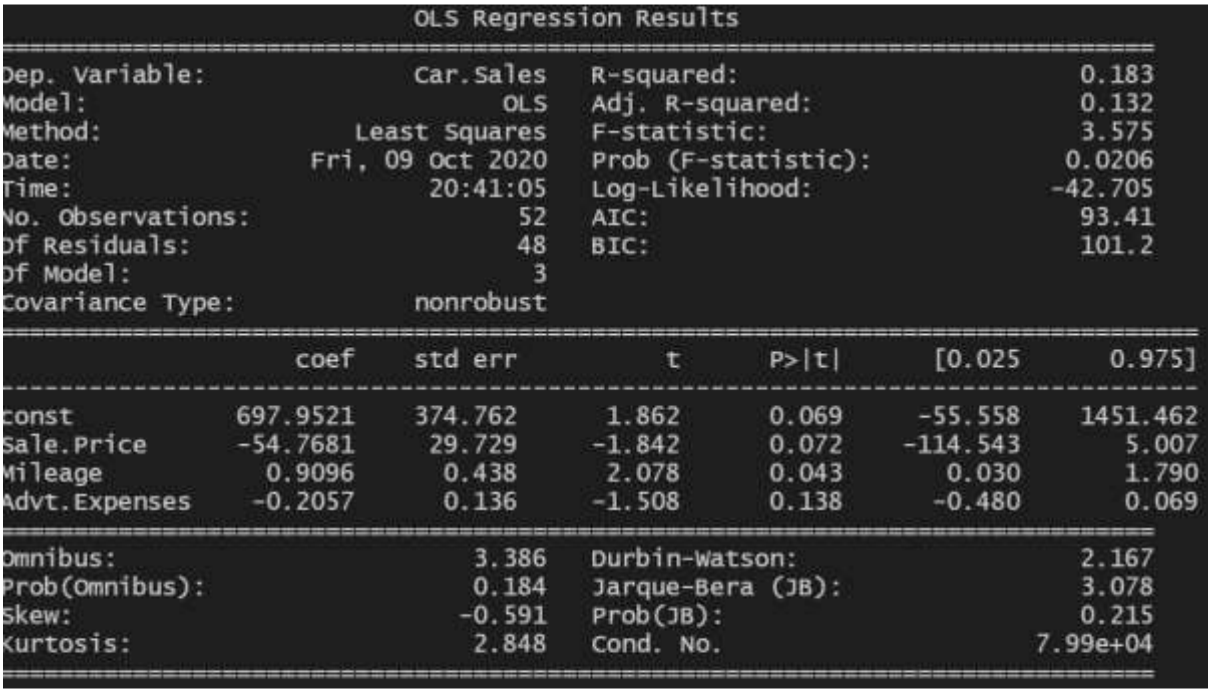
variables = list(yourdf.columns)

y = 'Car.Sales'

x = [var for var in variables if var not in y ]

model = sm.OLS(yourdf [y], sm.add_constant(yourdf [x])).fit()

model.summary()



```
                            OLS Regression Results
==============================================================================
Dep. Variable:              Car.Sales   R-squared:                       0.183
Model:                            OLS   Adj. R-squared:                  0.132
Method:                 Least Squares   F-statistic:                     3.575
Date:                Fri, 09 Oct 2020   Prob (F-statistic):             0.0206
Time:                        20:41:05   Log-Likelihood:                -42.705
No. Observations:                  52   AIC:                             93.41
Df Residuals:                      48   BIC:                             101.2
Df Model:                           3
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
const           697.9521    374.762      1.862      0.069     -55.558    1451.462
Sale.Price      -54.7681     29.729     -1.842      0.072    -114.543       5.007
Mileage           0.9096      0.438      2.078      0.043       0.030       1.790
Advt.Expenses    -0.2057      0.136     -1.508      0.138      -0.480       0.069
==============================================================================
Omnibus:                        3.386   Durbin-Watson:                   2.167
Prob(Omnibus):                  0.184   Jarque-Bera (JB):                3.078
Skew:                          -0.591   Prob(JB):                        0.215
Kurtosis:                       2.848   Cond. No.                     7.99e+04
==============================================================================
```

Now, what to look for in the above output, if you go by theory or routine examples, adjusted-r-square, p-values, etc. However, your adjusted-r-square and p-values directly depend on your employed data, not to go in for them now at first.

Let's go back to where we started, that is, we want to understand which ones are driving sales in the car data set. Is it price, mileage or advertisement expenses? For that let's look at coefficient's values respectively:

What does "Sale.Price" value is telling? – Its sign is negative, and it is telling people are buying more cars of less price.

What does "Mileage" value is telling? – Its sign is positive, and it is telling people are buying more cars of more mileage ones.

What does "Advt.Expenses" value is telling? – Its sign is negative, and its p-value is insignificant. Herein what your business understandings help, currently, though it is insignificant, it is telling if you spend more on advertisement, your sales will not increase. This cannot be true, if that is true, then no body will advertise, then, what might went wrong is the question?

Herein, validation of the models comes into the picture, where lot of data science learning and understanding helps us to get right insights.

As, said earlier, does all the data we have is enough for insights we want to generate is a key question, if does we are processing all the data appropriately to the model, as model doesn't select right data, it is you, who need to process and provide right data to model of let it be ML/Stats.

I have provided data set in my repository for your practices, you can practice and can come up with standardization or data processing things that will help here to provide better insights.