Data, Everywhere!

Stat 133, Fall 2017

George Khalilieh

11/27/2017

Introduction

Data. It's everywhere. But if you don't know where to look and how to use it, it's effectively nowhere in practice. And if you know where to look , you can find plenty of it. Some people would even say there is too much data available.

Take a look below at a few data sources.

Finance.

Natural Language.

Machine Learning.

These topics are hot nowadays. If you peer through the above links, you can find more than enough to explore these fields.

But you have to look no further than the R language itself to get some cool data.

The datasets we are using come from here.

Motivation

I have worked with data wrangling and searching for data before, and I really appreciate how R has interesting datasets packed in. No imports required. I found this astounding. The package we are using is called datasets, and is structured so that any useR can simply type in the name of a dataset and access its contents instantly. No downloads, formatting, or wrangling required. Everything just works straight out of the box. In Python you have to use sklearn's datasets or read in CSV files in the working directory. For R, it takes less than 5 seconds to access cool built in datasets.

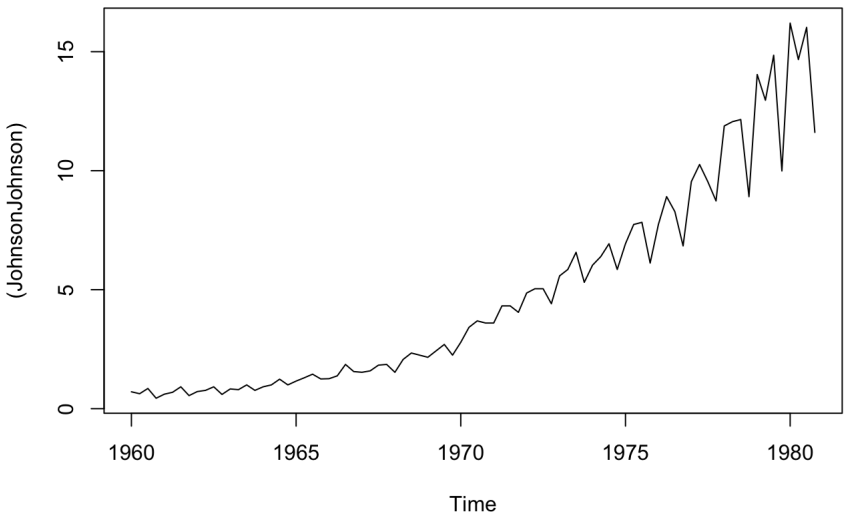
Analysis and Examples

```
# an example
JohnsonJohnson
```

```
##       Qtr1  Qtr2  Qtr3  Qtr4
## 1960  0.71  0.63  0.85  0.44
## 1961  0.61  0.69  0.92  0.55
## 1962  0.72  0.77  0.92  0.60
## 1963  0.83  0.80  1.00  0.77
## 1964  0.92  1.00  1.24  1.00
## 1965  1.16  1.30  1.45  1.25
## 1966  1.26  1.38  1.86  1.56
## 1967  1.53  1.59  1.83  1.86
## 1968  1.53  2.07  2.34  2.25
## 1969  2.16  2.43  2.70  2.25
## 1970  2.79  3.42  3.69  3.60
## 1971  3.60  4.32  4.32  4.05
## 1972  4.86  5.04  5.04  4.41
## 1973  5.58  5.85  6.57  5.31
## 1974  6.03  6.39  6.93  5.85
## 1975  6.93  7.74  7.83  6.12
## 1976  7.74  8.91  8.28  6.84
## 1977  9.54 10.26  9.54  8.73
## 1978 11.88 12.06 12.15  8.91
## 1979 14.04 12.96 14.85  9.99
## 1980 16.20 14.67 16.02 11.61
```

Above, I pulled up a dataset of Johnson and Johnson Quarterly earnings per share from 1960 to 1980. I have imported nothing, and you can type `JohnsonJohnson`. And, voila! You have quarterly earnings (dollars) per Johnson & Johnson share from 1960–1980.

```
plot((JohnsonJohnson))
```

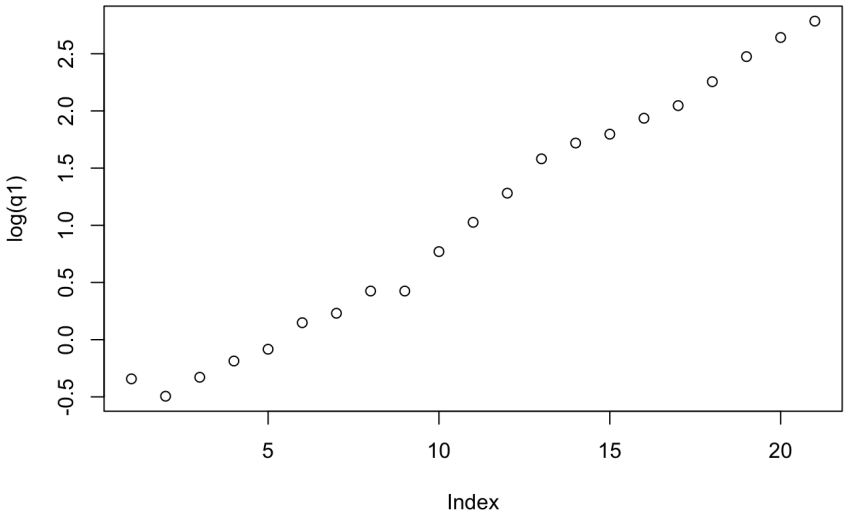



```
# plotting Q1 EPS
length(JohnsonJohnson)
```

```
## [1] 84
```

```
q1 <- JohnsonJohnson[seq(from = 1, to = 84, by = 4)]

plot(log(q1))
```




```
q2 <- JohnsonJohnson[seq(from = 2, to = 83, by = 4)]

paste("The average of q1 EPS is", mean(q1))
```

```
## [1] "The average of q1 EPS is 4.79142857142857"
```

Now, Johnson and Johnson from 1960 - 1980 is not a typical case study, but the ease of access to this data is fantastic.

Now, I want you to try taking the average of the Q2 EPS. Try it below.

```
# your code here
```

A possible solution below:

```
(mean(q2))
```

```
## [1] 4.965714
```

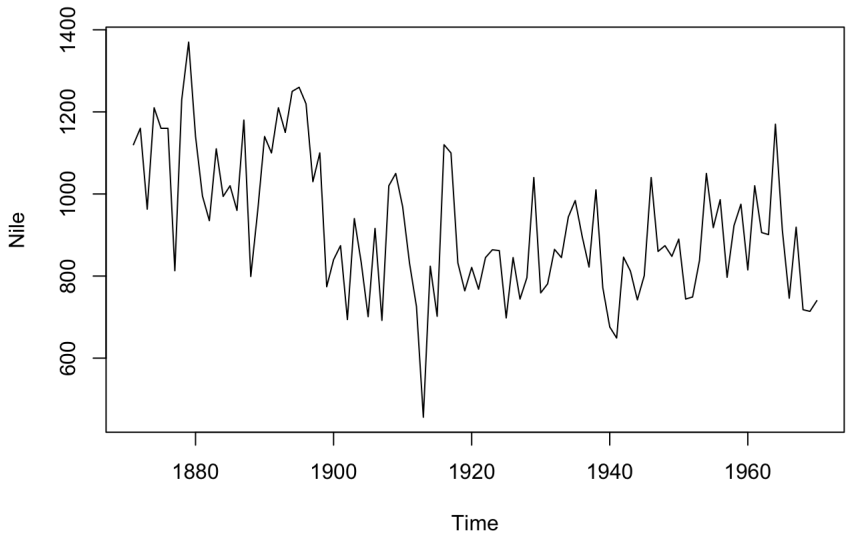
Now, let's look at a different dataset.The flow of the river Nile from 1871- 1970. Source.

```
Nile
```

```
## Time Series:
## Start = 1871
## End = 1970
## Frequency = 1
##   [1] 1120 1160  963 1210 1160 1160  813 1230 1370 1140  995  935 1110  994
##  [15] 1020  960 1180  799  958 1140 1100 1210 1150 1250 1260 1220 1030 1100
##  [29]  774  840  874  694  940  833  701  916  692 1020 1050  969  831  726
##  [43]  456  824  702 1120 1100  832  764  821  768  845  864  862  698  845
##  [57]  744  796 1040  759  781  865  845  944  984  897  822 1010  771  676
##  [71]  649  846  812  742  801 1040  860  874  848  890  744  749  838 1050
##  [85]  918  986  797  923  975  815 1020  906  901 1170  912  746  919  718
##  [99]  714  740
```

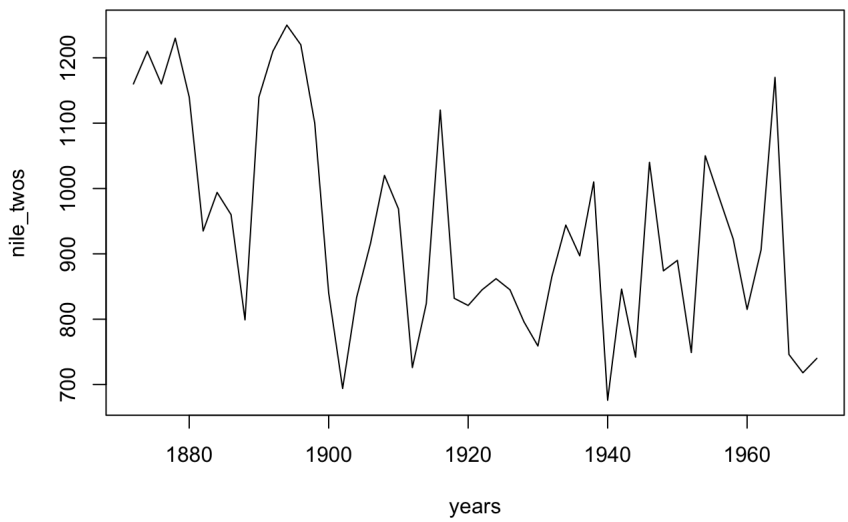
We can plot the flow as below.

```
plot(Nile)
```



The dataset contains 99 years of flow data. Let's look at the flows for every two years.

```
nile_twos <- Nile[seq(2, length(Nile), 2)]
years <- seq(2, length(Nile), 2) + 1870
plot(years, nile_twos,type = '1')
```

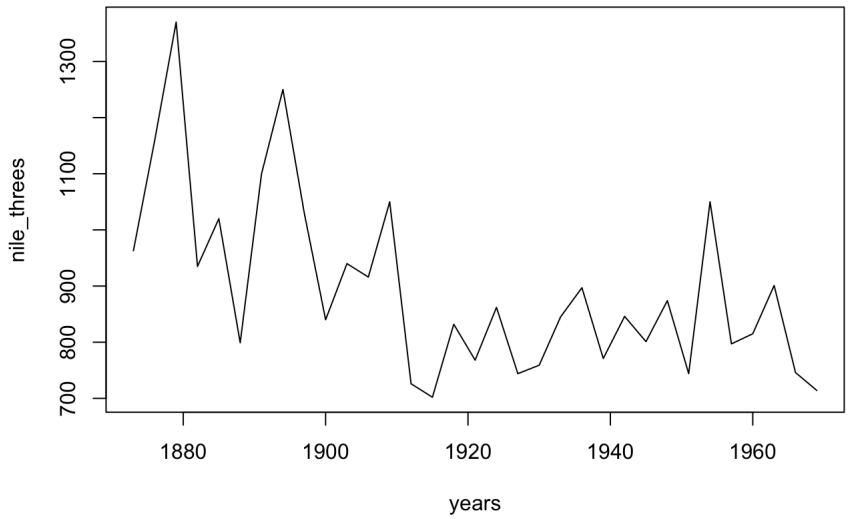


How would you get every 3 year in the dataset? Try it below:

```
# your code here
```

A possible solution:

```
nile_threes <- Nile[seq(3, length(Nile), 3)]
years <- seq(3, length(Nile), 3) + 1870
plot(years, nile_threes,type = 'l')
```



How would you calculate the average flow over the most recent 10 years?

```
# your code below
```

A solution below:

```
mean(Nile[(length(Nile) - 10): length(Nile)])
```

```
## [1] 869.1818
```

How would you calculate the average flow over the most recent 20 years?

```
# your code below
```

A solution below:

```
mean(Nile[(length(Nile) - 20): length(Nile)])
```

```
## [1] 877.6667
```

How would you calculate the average flow over the most recent 50 years?

```
# your code below
```

A solution below:

```
mean(Nile[(length(Nile) - 50): length(Nile)])
```

```
## [1] 853.7255
```

Below, play around with the data however you like.

```
# playground space
```

Let's look at socieconomic Data for Switzerland. Source.

```
swiss
```

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
## Neuveville        76.9        43.5          17        15     5.16
## Porrentruy        76.1        35.3           9         7    90.57
```

```
## Broye             83.8        70.2          16         7    92.85
## Glane             92.4        67.8          14         8    97.16
## Gruyere           82.4        53.3          12         7    97.67
## Sarine            82.9        45.2          16        13    91.38
## Veveyse           87.1        64.5          14         6    98.61
## Aigle             64.1        62.0          21        12     8.52
## Aubonne           66.9        67.5          14         7     2.27
## Avenches          68.9        60.7          19        12     4.43
## Cossonay          61.7        69.3          22         5     2.82
## Echallens         68.3        72.6          18         2    24.20
## Grandson          71.7        34.0          17         8     3.30
## Lausanne          55.7        19.4          26        28    12.11
## La Vallee         54.3        15.2          31        20     2.15
## Lavaux            65.1        73.0          19         9     2.84
## Morges            65.5        59.8          22        10     5.23
## Moudon            65.0        55.1          14         3     4.52
## Nyone             56.6        50.9          22        12    15.14
## Orbe              57.4        54.1          20         6     4.20
## Oron              72.5        71.2          12         1     2.40
## Payerne           74.2        58.1          14         8     5.23
## Paysd'enhaut      72.0        63.5           6         3     2.56
## Rolle             60.5        60.8          16        10     7.72
## Vevey             58.3        26.8          25        19    18.46
## Yverdon           65.4        49.5          15         8     6.10
## Conthey           75.5        85.9           3         2    99.71
## Entremont         69.3        84.9           7         6    99.68
## Herens            77.3        89.7           5         2   100.00
## Martigwy          70.5        78.2          12         6    98.96
## Monthey           79.4        64.9           7         3    98.22
## St Maurice        65.0        75.9           9         9    99.06
## Sierre            92.2        84.6           3         3    99.46
## Sion              79.3        63.1          13        13    96.83
## Boudry            70.4        38.4          26        12     5.62
## La Chauxdfnd      65.7         7.7          29        11    13.79
## Le Locle          72.7        16.7          22        13    11.22
## Neuchatel         64.4        17.6          35        32    16.92
## Val de Ruz        77.6        37.6          15         7     4.97
## ValdeTravers      67.6        18.7          25         7     8.65
## V. De Geneve      35.0         1.2          37        53    42.34
## Rive Droite       44.7        46.6          16        29    50.43
## Rive Gauche       42.8        27.7          22        29    58.33
##              Infant.Mortality
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
## Neuveville               20.6
## Porrentruy               26.6
## Broye                    23.6
## Glane                    24.9
## Gruyere                  21.0
## Sarine                   24.4
## Veveyse                  24.5
## Aigle                    16.5
## Aubonne                  19.1
## Avenches                 22.7
## Cossonay                 18.7
## Echallens                21.2
## Grandson                 20.0
## Lausanne                 20.2
## La Vallee                10.8
## Lavaux                   20.0
## Morges                   18.0
## Moudon                   22.4
## Nyone                    16.7
## Orbe                     15.3
## Oron                     21.0
## Payerne                  23.8
## Paysd'enhaut             18.0
## Rolle                    16.3
## Vevey                    20.9
## Yverdon                  22.5
## Conthey                  15.1
## Entremont                19.8
## Herens                   18.3
## Martigwy                 19.4
## Monthey                  20.2
## St Maurice               17.8
## Sierre                   16.3
## Sion                     18.1
## Boudry                   20.3
## La Chauxdfnd             20.5
## Le Locle                 18.9
## Neuchatel                23.0
## Val de Ruz               20.0
```

```
## ValdeTravers             19.5
## V. De Geneve             18.0
## Rive Droite              18.2
## Rive Gauche              19.3
```

Let's extract the Fertility column.

```
fertility <- swiss$Fertility
summary(fertility)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   35.00   64.70   70.40   70.14   78.45   92.50
```

How would we get the summary of the `Agriculture` column? Try it below:

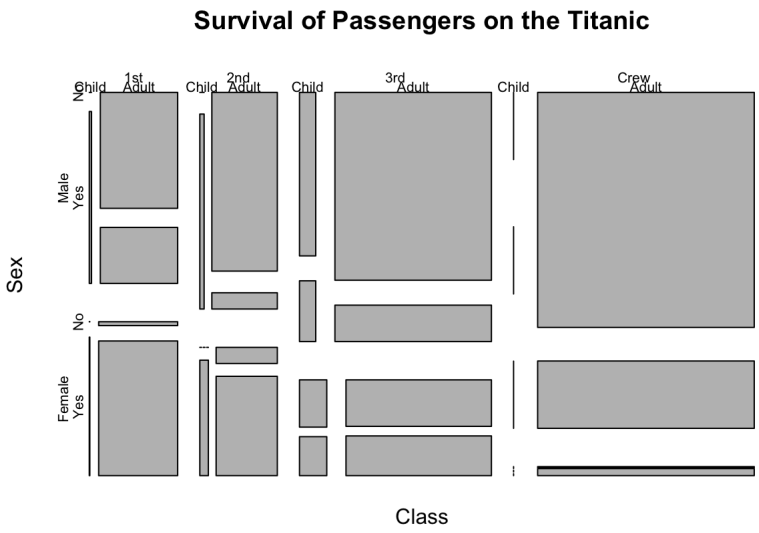
```
# your code here
```

```
# solution.
agriculture <- swiss$Fertility
summary(fertility)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   35.00   64.70   70.40   70.14   78.45   92.50
```

A final interesting dataset to look at is the dataset of the Titanic passengers, showing age, sex, and survival status. Source.

```
plot(Titanic, main = "Survival of Passengers on the Titanic")
```




```
# this sums together passenger class by gender
apply(Titanic, c(1, 2), sum)
```

```
##       Sex
## Class  Male Female
##   1st   180    145
##   2nd   179    106
##   3rd   510    196
##   Crew  862     23
```

You can use the apply function to sum together columns on the dataset. Try modifying the above statement to use `c(3, 4)` instead of `c(1, 3)`.

```
apply(Titanic, c(3, 4), sum)
```

```
##        Survived
## Age         No Yes
##   Child     52  57
##   Adult   1438 654
```

Final Thoughts and Take Home Message

I hope these analyses have inspired you to play around with R's great datasets package. I was really happy to find these datasets to play around with, and I hope I've encouraged you to look into the datasets and just play around more with R. You can uncover some really cool insights from

a few keystrokes.A big hurdle for me getting started in Data Science was finding data that interested me and in a form that I knew how to use. This very much helped me get more comfortable with new forms of data and more varieties of information.

References

- https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html
- https://github.com/caesar0301/awesome-public-datasets
- https://github.com/caesar0301/awesome-public-datasets#finance
- https://github.com/caesar0301/awesome-public-datasets#natural-language -https://github.com/caesar0301/awesome-public-datasets#machine-learning
- https://www.amstat.org/publications/jse/v3n3/datasets.dawson.html
- http://www.ssfpack.com/DKbook.html
- https://opr.princeton.edu/archive/pefp/switz.aspx