data_Gathering_and_Cleaning

June 14, 2021

1 Data Gathering and Cleaning

1.1 five main steps for data science processing.

[1]:
```
# 1. Data acquisition is where you read data from various sources of␣
 ↪unstructured data, semistructured data, or full-structured data that might␣
 ↪be stored in a spreadsheet, comma-separated file, web page, database, etc.
# 2. Data cleaning is where you remove noisy data and make operations needed to␣
 ↪keep only the relevant data.
# 3. Exploratory analysis is where you look at your cleaned data and make␣
 ↪statistical processing fits for specific analysis purposes.
# 4.An analysis model needs to be created. Advanced tools such as machine␣
 ↪learning algorithms can be used in this step.
# 5.Data visualization is where the results are plotted using various systems␣
 ↪provided by Python to help in the decision-making process.
```

[2]:
```
# libraries for data gathering, cleaning, integration, processing, and␣
 ↪visualizing.
```

[3]:
```
# • Pandas is an open source Python library used to load, organize, manipulate,␣
 ↪model, and analyze data by offering powerful data structures.
# • Numpy is a Python package that stands for "numerical Python. It is a␣
 ↪library consisting of multidimensional array objects and a collection of␣
 ↪routines for manipulating arrays. It can be used to perform mathematical,␣
 ↪logical, and linear algebra operations on arrays.
# • SciPy is another built-in Python library for numerical integration and␣
 ↪optimization.
# • Matplotlib is a Python library used to create 2D graphs and plots. It␣
 ↪supports a wide variety of graphs and plots such as histograms, bar charts,␣
 ↪power spectra, error charts, and so on, with additional formatting such as␣
 ↪control line styles, font properties, formatting axes, and more.
```

1.2 Cleaning Data

[14]:
```
# Creating a Data Frame Including NaN
import pandas as pd
import numpy as np
```

```
dataset = pd.DataFrame(np.random.randn(5, 3),  index=['a', 'c', 'e', 'f',␣
 ↪'h'],columns=['stock1','stock2', 'stock3'])
dataset.rename(columns={"one":'stock1',"two":'stock2',"three":'stock3'},␣
 ↪inplace=True)
dataset = dataset.reindex(['a', 'b', 'c', 'd', 'e','f', 'g', 'h'])
print (dataset)
```

```
     stock1    stock2    stock3
a -0.149687  0.489734 -0.773401
b       NaN       NaN       NaN
c -0.306800 -0.065527 -0.124510
d       NaN       NaN       NaN
e  0.917727  0.391552  0.090338
f -0.709667  1.066056  0.007785
g       NaN       NaN       NaN
h -0.176614 -0.759529 -1.056215
```

[15]:
```
#Checking Null Cases
print (dataset['stock1'].isnull())
```

```
a    False
b     True
c    False
d     True
e    False
f    False
g     True
h    False
Name: stock1, dtype: bool
```

[16]:
```
#Replacing NaN with a Scalar Value
import pandas as pd
import numpy as np
dataset = pd.DataFrame(np.random.randn(5, 3),  index=['a', 'c', 'e', 'f',␣
 ↪'h'],columns=['stock1','stock2', 'stock3'])
dataset.rename(columns={"one":'stock1',"two":'stock2',"three":'stock3'},␣
 ↪inplace=True)
dataset = dataset.reindex(['a', 'b', 'c', 'd', 'e','f', 'g', 'h'])
print (dataset)
dataset.fillna(0)
```

```
     stock1    stock2    stock3
a -0.199989 -0.259611  1.399868
b       NaN       NaN       NaN
c  0.364557 -0.995261 -0.136877
d       NaN       NaN       NaN
e -0.889671  0.088498  0.345939
f -0.585002  0.985415  0.787552
```

```
g       NaN       NaN       NaN
h -0.676979 -0.548171  1.825334
```

[16]:
```
      stock1    stock2    stock3
a -0.199989 -0.259611  1.399868
b  0.000000  0.000000  0.000000
c  0.364557 -0.995261 -0.136877
d  0.000000  0.000000  0.000000
e -0.889671  0.088498  0.345939
f -0.585002  0.985415  0.787552
g  0.000000  0.000000  0.000000
h -0.676979 -0.548171  1.825334
```

[ ]:
```
```

[17]:
```
# Fill missing values forward
import pandas as pd
import numpy as np
dataset = pd.DataFrame(np.random.randn(5, 3),  index=['a', 'c', 'e', 'f',␣
↪'h'],columns=['stock1','stock2', 'stock3'])
dataset.rename(columns={"one":'stock1',"two":'stock2',"three":'stock3'},␣
↪inplace=True)
dataset = dataset.reindex(['a', 'b', 'c', 'd', 'e','f', 'g', 'h'])
print (dataset)
dataset.fillna(method='pad')
```

```
     stock1    stock2    stock3
a -0.162560  0.270799 -0.161781
b       NaN       NaN       NaN
c -0.947626  0.782127  1.106351
d       NaN       NaN       NaN
e -0.968298 -1.057833 -0.417992
f -1.126328  0.406150  2.193152
g       NaN       NaN       NaN
h -0.936814  0.306717  0.439464
```

[17]:
```
      stock1    stock2    stock3
a -0.162560  0.270799 -0.161781
b -0.162560  0.270799 -0.161781
c -0.947626  0.782127  1.106351
d -0.947626  0.782127  1.106351
e -0.968298 -1.057833 -0.417992
f -1.126328  0.406150  2.193152
g -1.126328  0.406150  2.193152
h -0.936814  0.306717  0.439464
```

[18]:
```
# Dropping All NaN Rows
import pandas as pd
```

```
import numpy as np
dataset = pd.DataFrame(np.random.randn(5, 3),  index=['a', 'c', 'e', 'f',␣
 ↪'h'],columns=['stock1','stock2', 'stock3'])
dataset.rename(columns={"one":'stock1',"two":'stock2',"three":'stock3'},␣
 ↪inplace=True)
dataset = dataset.reindex(['a', 'b', 'c', 'd', 'e','f', 'g', 'h'])
print (dataset)
dataset.dropna()
```

```
     stock1    stock2    stock3
a  0.579861 -0.544316 -0.101218
b       NaN       NaN       NaN
c  0.104602 -2.137943  2.169787
d       NaN       NaN       NaN
e -0.051485  0.841976 -1.376751
f -0.134908  0.508223  1.048612
g       NaN       NaN       NaN
h -2.361897 -0.845237  1.206937
```

[18]:
```
     stock1    stock2    stock3
a  0.579861 -0.544316 -0.101218
c  0.104602 -2.137943  2.169787
e -0.051485  0.841976 -1.376751
f -0.134908  0.508223  1.048612
h -2.361897 -0.845237  1.206937
```

[19]:
```
# Using the replace() Function
import pandas as pd
import numpy as np
dataset = pd.DataFrame(np.random.randn(5, 3),  index=['a', 'c', 'e', 'f',␣
 ↪'h'],columns=['stock1','stock2', 'stock3'])
dataset.rename(columns={"one":'stock1',"two":'stock2',"three":'stock3'},␣
 ↪inplace=True)
dataset = dataset.reindex(['a', 'b', 'c', 'd', 'e','f', 'g', 'h'])
print (dataset)
dataset.replace(np.nan, 0 )
```

```
     stock1    stock2    stock3
a  0.910986 -2.795030 -0.225208
b       NaN       NaN       NaN
c -0.295030 -0.095524  0.415947
d       NaN       NaN       NaN
e -0.987055  0.186662  1.995822
f  0.176123  0.845438 -0.366631
g       NaN       NaN       NaN
h  0.525205  0.669292 -0.792342
```

[19]:
```
      stock1    stock2    stock3
a  0.910986 -2.795030 -0.225208
b  0.000000  0.000000  0.000000
c -0.295030 -0.095524  0.415947
d  0.000000  0.000000  0.000000
e -0.987055  0.186662  1.995822
f  0.176123  0.845438 -0.366631
g  0.000000  0.000000  0.000000
h  0.525205  0.669292 -0.792342
```

[8]:
```
# Creating a Matrix of Random Values
import numpy as np
np.random.randn(5, 3)
```

[8]:
```
array([[-0.08296097,  1.55943988,  0.42772814],
       [ 1.14855305,  1.53045705, -1.02749891],
       [-0.60288851,  1.67940253, -0.48861584],
       [ 0.65666547,  1.56301256,  1.26979982],
       [ 1.19686267,  2.90297703,  0.02560203]])
```

1.3 Reading and Cleaning CSV Data

[21]:
```
# Reading a CSV File and Displaying the First Five Records
import pandas as pd
sales = pd.read_csv("Sales.csv")
print ("\n\n<<<<<<< First 5 records <<<<<<<\n\n" )
print (sales.head())
```

```
<<<<<<< First 5 records <<<<<<<


   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

          ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  …  \
0   2/24/2003 0:00  Shipped       1         2     2003  …
1    5/7/2003 0:00  Shipped       2         5     2003  …
2    7/1/2003 0:00  Shipped       3         7     2003  …
3   8/25/2003 0:00  Shipped       3         8     2003  …
4  10/10/2003 0:00  Shipped       4        10     2003  …

                    ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
```

```
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN              Yu             Kwai    Small
1      51100  France      EMEA         Henriot             Paul    Small
2      75508  France      EMEA        Da Cunha            Daniel   Medium
3      90003     USA       NaN           Young            Julie   Medium
4        NaN     USA       NaN           Brown            Julie   Medium

[5 rows x 25 columns]
```

[22]:
```
# Read only a few records
import pandas as pd
salesNrows = pd.read_csv("Sales.csv", nrows=4)
salesNrows
```

[22]:
```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70

         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  …  \
0  2/24/2003 0:00  Shipped       1         2     2003  …
1   5/7/2003 0:00  Shipped       2         5     2003  …
2   7/1/2003 0:00  Shipped       3         7     2003  …
3  8/25/2003 0:00  Shipped       3         8     2003  …

                   ADDRESSLINE1  ADDRESSLINE2      CITY STATE POSTALCODE  \
0        897 Long Airport Avenue           NaN       NYC    NY      10022
1             59 rue de l'Abbaye           NaN     Reims   NaN      51100
2  27 rue du Colonel Pierre Avia           NaN     Paris   NaN      75508
3             78934 Hillside Dr.           NaN  Pasadena    CA      90003

  COUNTRY  TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME  DEALSIZE
0     USA        NaN              Yu             Kwai     Small
1  France       EMEA         Henriot             Paul     Small
2  France       EMEA        Da Cunha           Daniel    Medium
3     USA        NaN           Young            Julie    Medium

[4 rows x 25 columns]
```

[23]:
```
# Renaming Column Labels
salesNrows.rename(columns={"ORDERNUMBER":'Order#',"QUANTITYORDERED":'Qty␣
 ↪Ordered'}, inplace=True)
salesNrows
```

[23]:
```
   Order#  Qty Ordered  PRICEEACH  ORDERLINENUMBER    SALES       ORDERDATE  \
0   10107           30      95.70                2  2871.00  2/24/2003 0:00
1   10121           34      81.35                5  2765.90   5/7/2003 0:00
2   10134           41      94.74                2  3884.34   7/1/2003 0:00
3   10145           45      83.26                6  3746.70  8/25/2003 0:00

    STATUS  QTR_ID  MONTH_ID  YEAR_ID  …                  ADDRESSLINE1  \
0  Shipped       1         2     2003  …        897 Long Airport Avenue
1  Shipped       2         5     2003  …            59 rue de l'Abbaye
2  Shipped       3         7     2003  …  27 rue du Colonel Pierre Avia
3  Shipped       3         8     2003  …             78934 Hillside Dr.

   ADDRESSLINE2      CITY STATE POSTALCODE COUNTRY  TERRITORY CONTACTLASTNAME  \
0           NaN       NYC    NY      10022     USA        NaN              Yu
1           NaN     Reims   NaN      51100  France       EMEA         Henriot
2           NaN     Paris   NaN      75508  France       EMEA        Da Cunha
3           NaN  Pasadena    CA      90003     USA        NaN           Young

  CONTACTFIRSTNAME  DEALSIZE
0             Kwai     Small
1             Paul     Small
2           Daniel    Medium
3            Julie    Medium

[4 rows x 25 columns]
```

[24]:
```
# Finding Unique Values in Columns
print (len(salesNrows['PRICEEACH'].unique()))
```

```
4
```

[25]:
```
# Replace all values that are anomalies with NaN
import pandas as pd
sales = pd.read_csv("Sales.csv", nrows=7, na_values=["n.a.", "not avilable"])
mydata = sales.head(7)
mydata
```

[25]:
```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27
```

```
5        10168                36       96.66                 1  3479.76
6        10180                29       86.13                 9  2497.77

          ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  …  \
0   2/24/2003 0:00  Shipped       1         2     2003  …
1    5/7/2003 0:00  Shipped       2         5     2003  …
2    7/1/2003 0:00  Shipped       3         7     2003  …
3   8/25/2003 0:00  Shipped       3         8     2003  …
4  10/10/2003 0:00  Shipped       4        10     2003  …
5  10/28/2003 0:00  Shipped       4        10     2003  …
6  11/11/2003 0:00  Shipped       4        11     2003  …

                    ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0         897 Long Airport Avenue           NaN            NYC    NY
1              59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3              78934 Hillside Dr.           NaN       Pasadena    CA
4                 7734 Strong St.           NaN  San Francisco    CA
5               9408 Furth Circle           NaN     Burlingame    CA
6         184, chausse de Tournai           NaN          Lille   NaN

  POSTALCODE COUNTRY  TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME  DEALSIZE
0    10022.0     USA        NaN              Yu             Kwai     Small
1    51100.0  France       EMEA         Henriot             Paul     Small
2    75508.0  France       EMEA        Da Cunha            Daniel    Medium
3    90003.0     USA        NaN           Young             Julie    Medium
4        NaN     USA        NaN           Brown             Julie    Medium
5    94217.0     USA        NaN          Hirano              Juri    Medium
6    59000.0  France       EMEA           Rance           Martine     Small

[7 rows x 25 columns]
```

1.4 Merging and Integrating Data

[31]:
```
ls -l
```

```
total 5944
-rw-r--r--@ 1 robertkigobe  staff    9361 Jul  5  2020 April2019.csv
-rw-r--r--@ 1 robertkigobe  staff   18914 Jul  5  2020 August2019.csv
-rw-r--r--@ 1 robertkigobe  staff   23672 Jul  5  2020 December 2019.csv
-rw-r--r--@ 1 robertkigobe  staff    6723 Jul  5  2020 February 2019.csv
-rw-r--r--@ 1 robertkigobe  staff    5892 Jul  5  2020 January 2019.csv
-rw-r--r--@ 1 robertkigobe  staff   19632 Jul  5  2020 July 2019.csv
-rw-r--r--@ 1 robertkigobe  staff   15013 Jul  5  2020 June 2019.csv
-rw-r--r--@ 1 robertkigobe  staff    9994 Jul  5  2020 March 2019.csv
-rw-r--r--@ 1 robertkigobe  staff    9581 Jul  5  2020 May 2019.csv
-rw-r--r--@ 1 robertkigobe  staff   13864 Jul  5  2020 November 2019.csv
-rw-r--r--@ 1 robertkigobe  staff   18038 Aug  9  2020 October 2019.csv
```

```
-rw-r--r--  1 robertkigobe  staff    5755 Jun 12 07:34 Python_Lambdas.ipynb
-rw-r--r--@ 1 robertkigobe  staff    1956 Jun 12 10:38 Python_Lambdas.py
-rw-r--r--@ 1 robertkigobe  staff   18034 Jul  5  2020 September 2019.csv
-rw-r--r--  1 robertkigobe  staff    1034 Jun 14 07:43 Untitled.ipynb
-rw-r--r--  1 robertkigobe  staff   54208 Jun 14 11:38 Untitled1.ipynb
-rw-r--r--  1 robertkigobe  staff    3716 Jun 12 04:33 basic_operations.ipynb

-rw-r--r--@ 1 robertkigobe  staff     963 Jun 12 06:11 basic_operations.py
-rw-r--r--  1 robertkigobe  staff  642544 Jun 12 17:48
data_VIsualIzatIon_Using_Plots.ipynb
-rw-r--r--@ 1 robertkigobe  staff  793989 Jun 12 17:47
data_VIsualIzatIon_Using_Plots.pdf
-rw-r--r--  1 robertkigobe  staff    7329 Jun 14 07:31 file_IO_processing.ipynb
-rw-r--r--@ 1 robertkigobe  staff   38065 Jun 14 07:38 file_IO_processing.pdf
-rw-r--r--  1 robertkigobe  staff   10996 Jun 14 11:22 functions.ipynb
-rw-r--r--@ 1 robertkigobe  staff   56961 Jun 14 08:03 functions.pdf
-rw-r--r--@ 1 robertkigobe  staff    3238 Jun 14 08:03 functions.py
-rw-r--r--  1 robertkigobe  staff     906 Jun 13 13:48 list_demo.txt
-rw-r--r--  1 robertkigobe  staff   17056 Jun 13 14:40 lists.ipynb
-rw-r--r--@ 1 robertkigobe  staff   65676 Jun 13 14:43 lists.pdf
-rw-r--r--@ 1 robertkigobe  staff    3536 Jun 13 14:42 lists.py
-rw-r--r--  1 robertkigobe  staff  119407 Jun 12 12:16
python_Basic_Infrential_Analysys.ipynb
-rw-r--r--@ 1 robertkigobe  staff  150887 Jun 12 12:17
python_Basic_Infrential_Analysys.pdf
-rw-r--r--@ 1 robertkigobe  staff    2670 Jun 12 12:16
python_Basic_Infrential_Analysys.py
-rw-r--r--  1 robertkigobe  staff   10588 Jun 12 11:31
python_Numpy_Package.ipynb
-rw-r--r--@ 1 robertkigobe  staff   64595 Jun 12 11:43 python_Numpy_Package.pdf
-rw-r--r--@ 1 robertkigobe  staff    3278 Jun 12 11:43 python_Numpy_Package.py
-rw-r--r--  1 robertkigobe  staff    4483 Jun 12 05:18 python_fundamentals.ipynb
-rw-r--r--@ 1 robertkigobe  staff   43255 Jun 12 05:19 python_fundamentals.pdf
-rw-r--r--@ 1 robertkigobe  staff     971 Jun 12 06:10 python_fundamentals.py
-rw-r--r--  1 robertkigobe  staff  528050 Jun 14 11:09 sales.csv
-rw-r--r--  1 robertkigobe  staff     459 Jun 14 07:23 south_africa.txt
-rw-r--r--  1 robertkigobe  staff    5100 Jun 12 06:07 string_manipulation.ipynb
-rw-r--r--@ 1 robertkigobe  staff   43278 Jun 12 06:07 string_manipulation.pdf
-rw-r--r--@ 1 robertkigobe  staff    1746 Jun 12 06:07 string_manipulation.py
-rw-r--r--  1 robertkigobe  staff    7783 Jun 12 07:02 tabular_data.ipynb
-rw-r--r--@ 1 robertkigobe  staff   45974 Jun 12 07:04 tabular_data.pdf
-rw-r--r--@ 1 robertkigobe  staff    1806 Jun 12 07:02 tabular_data.py
```

[32]:
```
pip install pip-autoremove
pip-autoremove pandas -y
pip install pandas
```

```
  File "<ipython-input-32-7f0321042515>", line 1
    pip install pip-autoremove
        ^
SyntaxError: invalid syntax
```

[33]:
```
import pandas as pd

april = pd.read("April2019.csv")
august = pd.read("August2019.csv")
april.head()
august.head()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-33-9c985f7eb27d> in <module>
      1 import pandas as pd
      2
----> 3 april = pd.read("April2019.csv")
      4 august = pd.read("August2019.csv")
      5 april.head()

~/opt/anaconda3/lib/python3.8/site-packages/pandas/__init__.py in␣
 ↪__getattr__(name)
    242         return _SparseArray
    243
--> 244     raise AttributeError(f"module 'pandas' has no attribute '{name}'")
    245
    246

AttributeError: module 'pandas' has no attribute 'read'
```

[ ]:
```
#Dropping Columns 2009, 2012, 2013, and 2014
b.drop('2014', axis=1, inplace=True)
columns = ['2013', '2012']
b.drop(columns, inplace=True, axis=1)
b.head()
```

[ ]:
```
# Merging Two Data Sets
mergedDataSet = a.merge(b, on="Country Name")
mergedDataSet.head()
```

1.5 Reading Data from the JSON Format

[34]:
```
#Creating and Manipulating JSON Data
import json data = '''{
        "name" : "Ossama",
        "phone" : { "type" : "intl", "number" : "+971 50 244
        5467"},
        "email" : {"hide" : "No" }
        }'''
info = json.loads(data)
print ('Name:',info["name"])
print ('Hide:',info["email"]["hide"])
```

```
  File "<ipython-input-34-e6eff11d077b>", line 2
    import json data = '''{
                   ^
SyntaxError: invalid syntax
```

1.6 Reading and Parsing an HTML File

[35]:
```
import urllib from bs4
import BeautifulSoup
response = urllib.request.urlopen('http://python-data.dr-chuck.net/
 ↪known_by_Rona.html'
html_doc = response.read()
soup = BeautifulSoup(html_doc, 'html.parser')
print(html_doc[:700])
print("\n")
print (soup.title)
print(soup.title.string)
print(soup.a.string)
```

```
  File "<ipython-input-35-a8884ae9d2eb>", line 1
    import urllib from bs4
                  ^
SyntaxError: invalid syntax
```

[ ]:
```
import urllib.request
with urllib.request.urlopen("http://python-data.dr-
          chuck.net/known_by_Rona.html") as url:
                  strhtml = url.read()
          #I'm guessing this would output the html source code?
          print(strhtml[:700])
```

[ ]: