2022

1 © 2022 Institute of Data



Data Science and AI

Module 2

Part 1:

Exploratory Data Analysis (EDA)

2 © 2022 Institute of Data

Agenda: Module 2 Part 1

• Introduction to EDA

• Data **cleaning** & **profiling**

• Assessing data **quality**

• Data rejection & imputation

• Exploring & visualising **continuous** data • Exploring & visualising **categorical** data • **Temporal** data

• **Geographic** data

3 © 2022 Institute of Data

Python EDA Fundamentals

• Where does data come from?

• What does data look like?

• What is **Exploratory Data Analysis**?

• Where does EDA fit in the **Data Science pipeline**?

4 © 2022 Institute of Data

Where does data come from?

• **databases**

• data marts

• data warehouses

• transaction systems • cloud

• mainframes

• distributed file systems • Hadoop

• **APIs**

• scanned documents

• websites

• downloads of datasets, posts, conversations, etc.

• web scrapers

• subscribed feeds

• news

• IoT devices

• multimedia hosts

• images

• video

• audio

• ?

5 © 2022 Institute of Data

What does data look like?

• database **tables**

• reports & extracts

• spreadsheets & workbooks

• **structured & semi-structured files** • **streams**

• encoded files

• bitmaps

• ?

6 © 2022 Institute of Data

What is Exploratory Data Analysis?

*everything we do with a candidate dataset ...*

• after it has been rendered essentially **usable**

• before we start **developing analytics and models** that address our original problem

• to determine whether it will make a useful **proxy** for understanding the phenomenon we are interested in

*where does it fit?*

• (within the data science pipeline)

7 © 2022 Institute of Data

How do we make a dataset “usable”?

• ***wrangling***

• sourcing, loading, and precleaning the data so we can see what it really looks like • fixing critical issues

• ***profiling and cleaning***

• understanding the essential characteristics of the data

• applying preliminary transformations to confer context and meaning • implementing strategies for missing and invalid data

• ***munging***

• reshaping the data to prepare it for analysis

8 © 2022 Institute of Data

Where does EDA fit? • (within the data science pipeline)

Define

• **objectives**

• **data**

sources

• **strategy**

Note that this process is never linear!

You will have to **iterate** over each step and over a number of the steps

Prepare

• wrangle

• clean

• profile

• munge

Analyse

• model

• predict

• Revise

• Evaluate

**Exploratory Data Analysis**

Deliver

• Visualise

• Summarise

• Advise /

Build

• Deploy

9 © 2022 Institute of Data

Data Cleaning & Profiling

• Preliminary data **cleaning**

• Basic data **profiling**

• Assessing data **quality**

• Data rejection and imputation

10 © 2022 Institute of Data

Data Cleaning & Profiling

*def:* **Data profiling**

• examining the characteristics of the dataset • data types

• data ranges (continuous) & categories

• identifying issues with the data

*def:* **Data cleaning**

• making the data usable (preparing it for analysis) • reformatting

• data type conversion

• dealing with dirty data

11 © 2022 Institute of Data

Data Cleaning & Profiling

Fix loading

errors

Parse and

convert

Summarise

data

*... is iterative*

Load raw

data

Summarise

data

Detect & fix

missing data

Detect & fix

invalid data

12 © 2022 Institute of Data

Data Cleaning & Profiling − Details

Load raw data

• from source system 

• database

• HFS

• flat file

• spreadsheet / workbook

• semi-structured file (JSON, XML, HTML)

• API 

• stream (feed, IoT)

• web scraper

• scanned text

13 © 2022 Institute of Data

Data Cleaning & Profiling − Details

Fix loading errors

• missing delimiters

• e.g. badly written mainframe extracts that suppress trailing commas for empty fields • unexpected delimiters

• e.g. ‘|’ or tab character used in “CSV” file

• illegal characters

• e.g. ‘\u’ is normally interpreted as indicating Unicode

may need to suppress default behaviour of function used to load the data • missing control characters

• EOL

• EOF

• other?

14 © 2022 Institute of Data

Data Cleaning & Profiling − Details

Parse and convert

• formatted date strings to dates

• d/m/y, m/d/y, dd/mm/yyyy, dd-mmm-yyyy, day names, month names, ... • formatted time strings to times

• AM/PM vs 24-hr

• time zone conversions

• formatted date+time strings to datetimes

• string to int, string to float

• proprietary formats

• binary, octal, hexadecimal

15 © 2022 Institute of Data

Data Cleaning & Profiling − Details What to do when data conversions fail? • implement a *try* block

• to catch format conversion failures

• use transformations that can handle missing values

• or deal with missing values first

• document conversion failures

• these are *limitations*that should be addressed when interpreting the results of analysis

def try\_parse\_int(s, base=10, val=None): try:

return int(s, base)

except ValueError:

return val

16 © 2022 Institute of Data

Data Cleaning & Profiling − Details

Detect & fix missing values

• drop rows

• replace with NA

• impute values

• mean, median, mode

• of entire column

• of similar data (grouped by other fields)

• nearest neighbour

• assign value from closest point (according to a suitable distance metric)

17 © 2022 Institute of Data

Data Cleaning & Profiling − Details

Dealing with missing or bad data

• replace with NA

• impute values

• out of range

• too small: set to minimum possible value?

• too large: set to maximum possible value?

• drop rows

• impossible values (e.g. out of domain)

• length = green: drop?

• salary = -1: drop?

• drop columns

• too many missing or invalid samples

18 © 2022 Institute of Data

Data Cleaning & Profiling − Details

Summarise data

• counts of missing values

• counts of invalid values

• statistical parameters of distribution

• continuous variables

• bin frequencies

• mean, median, maximum, minimum

• categorical variables

• category frequencies

• most frequent (mode), least frequent

19 © 2022 Institute of Data

Assessing Data Quality

• accuracy, reliability (veracity)

• currency, relevance (value)

• missing and invalid values

• overall

• by column

• by row

*issues:*

• can we afford to throw out rows with missing data? • how will imputation of missing/invalid data affect the outcome?

20 © 2022 Institute of Data

Assessing Data Quality with Python *let* df *be a Pandas DataFrame object*

• view the first few rows: • check for missing values: • pairwise correlations: • (continuous) value ranges: • (discrete) value counts:

• summary:

df.head(), df.head(nrows) df.isnull(), df.isnull().sum() df.corr()

df.min(), df.max()

df.value\_counts()

df.describe()

pandas\_profiling.ProfileReport pydqc

21 © 2022 Institute of Data

Lab 2.1.1: Data Wrangling and Munging

• Purpose:

• To explore Python methods for wrangling, munging, and profiling datasets

• Materials:

• ‘Lab 2.1.1.ipynb’

22 © 2022 Institute of Data

Outliers

*def:* an observation that is distant from other observations in the sample

• measurement inaccuracy 

• measurement errors

• incl. recording errors

• unusual system behaviour

• external phenomena

23 © 2022 Institute of Data

Outlier Detection in 1 Dimension

***extreme value analysis***

• outliers are defined by statistical tests based on mean & variance of sample • *Z*-test

• mark points with low score as outliers

***probabilistic & statistical models***

• based on assumed distribution of data

• calculate probability that each point belongs to the distribution

• mark points with low probability as outliers

24 © 2022 Institute of Data

Outlier Detection in Multiple Dimensions

***linear models***

• reduce data to lower-dimensional spaces

• calculate distance from each point to a reference hyperplane • mark points with largest distance as outliers

• similar concept to *principal component analysis* (PCA)

***proximity-based models***

• define a distance metric and apply to each pair of points

• mark points that are more isolated as outliers

• examples: *cluster analysis*, *density-based analysis*, *nearest-neighbour analysis*

25 © 2022 Institute of Data

Outlier Detection − cont’d

• outliers vs. anomalies

• if unsure, analyse data with *and* without the outliers

26 © 2022 Institute of Data

Outlier Detection − cont’d

• outliers may not be obvious in one dimension 

• some points may only get separated from the mainstream when looking at several dimensions at once

• may indicate subsets of behaviour (“classes”)

27 © 2022 Institute of Data

Continuous Data

• Statistics of sample distributions

• deeper dive: mean, variance, skewness, kurtosis

• Exploring and visualising sample variables • histograms

• box & whisker plots

• violin plots

• Outlier detection

28 © 2022 Institute of Data

Mean & Variance

29 © 2022 Institute of Data

Skewness and Kurtosis

30 © 2022 Institute of Data

Scatterplot

• shows a 2D relationship within the dataset by plotting one column against another



df.plot(kind='scatter', x='sepal\_width’, y='sepal\_length')

plt.scatter(df['sepal\_width’], df['sepal\_length’], s = 10, linewidths = 1, alpha = 0.5)

https://matplotlib.org/api/\_as\_gen/matplotlib.pyplot.scatter.html

31 © 2022 Institute of Data

Histogram

shows the properties of the data sample distribution with no loss of information 

plt.hist(y)

plt.title("Gaussian Histogram") plt.xlabel("Value")

plt.ylabel("Frequency")

df['temp'].hist()

32 © 2022 Institute of Data

Box & Whisker Plots

• shows multiple features of sample distribution 

• median

• interquartile range

• 10th, 90th percentiles

33 © 2022 Institute of Data

Box & Whisker Plots

# get 50 random numbers normally distributed about -1: 

y = np.random.randn(50) – 1

# create an outlier:

y [49] = 3

# plot box & whiskers:

plt.boxplot(y)

34 © 2022 Institute of Data

Violin Plots

• shows the sample distribution itself

https://matplotlib.org/gallery/statistics/customized\_violin.html

35 © 2022 Institute of Data

Quantiles

• quantiles are popular in reporting because they help to create a sense of what is “normal”

• *90% of calls last less than 3 minutes, 22 seconds*

• *80% of revenue was derived from 22% of the product range*

• quantiles are cumulative 

e.g. 80th percentile is a subset of

90th percentile

Q: what would a plot of all

possible quantiles represent?

***>*** the cumulative probability

function

36 © 2022 Institute of Data

Discretisation

• suppose want to look at intervals (“bins”) instead?

pandas.cut(x, bins, right=True, labels=None, retbins=False, precision=3, include\_lowest=False, duplicates='raise')

pandas.cut(df['temp'], bins = 4).head() • continuous data can be sorted into  specified bins 

• ***bins*** can be a vector of ‘cut’

boundaries (for asymmetric bins)

• bin counts can be plotted as a bar

chart (discrete version of histogram)

37 © 2022 Institute of Data

Continuous *n*-Dimensional Data

**marginal distribution**

• the distribution of the entire sample of a given variable from a multivariate sample

• ignores presence of other (n−1) covariates

**conditional distribution**

• the distribution of a given variable *contingent on* values of other (n−1) covariates

• for a pair of covariates X, Y

joint distribution: Pr(X = x, Y = y)

conditional distribution: Pr(X = x | Y = y) *Y has been “marginalised out”*

38 © 2022 Institute of Data

Pairwise Correlations in *n*-Dimensional Data

computes correlation between every pair of columns in a matrix or DataFrame:

• only the figures below 

(or above) the main

diagonal are needed

• uses Pearson’s

correlation by default

39 © 2022 Institute of Data

Pairwise Correlations in *n*-Dimensional Data − cont’d *can visualise correlations as a* **pair plot** 

import seaborn as sns

sns.pairplot(iris)

40 © 2022 Institute of Data

Visualising 2-Dimensional Data

• scatterplot 

• line chart

• bar chart (binned horizontal axis)

• stacked area chart

• *many variations of these*

41 © 2022 Institute of Data

Visualising 3-Dimensional Data 3D Scatterplot Wireframe Plot Surface Plot

https://matplotlib.org/mpl\_toolkits/mplot3d

42 © 2022 Institute of Data

Visualising 3 Dimensions − cont’d

• adding colour allows 

stratification by a

categorical variable

(usually called a “class”)

43 © 2022 Institute of Data

Visualising 3 Dimensions − cont’d using colour in a scatterplot using colour and hue in a contour plot

https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57

44 © 2022 Institute of Data

Visualising 3 Dimensions − cont’d

Slicing

• reduce dimensionality by viewing a plane

• does not have to be parallel to a dimensional axis http://zulko.github.io/blog/2014/11/29/data-animations-with-python-and-moviepy/

45 © 2022 Institute of Data

Visualising 3 Dimensions − cont’d

• heat map

46 © 2022 Institute of Data

Visualising > 3 Dimensions

• dimensional reduction 

• e.g. to animated trajectories

https://hypertools.readthedocs.io/en/latest/

47 © 2022 Institute of Data

Visualising > 3 Dimensions − cont’d

• parallel coordinates 

• can show multiple variables of

same scale

• especially useful for repeated

measures

• each variable is a time

point in a longitudinal study

from pandas.tools.plotting import

parallel\_coordinates

parallel\_coordinates (iris, 'species')

48 © 2022 Institute of Data

Visualising > 3 Dimensions − cont’d *scatterplot with glyphs* 

options for

encoding glyphs:

• size

• colour

• intensity

• transparency

• shape

• texture

49 © 2022 Institute of Data

Sankey Diagram

*state changes, class transitions, redistributions*

**50 © 2022 Institute of Data

Categorical Data

• Statistics of discrete distributions • class frequencies

• Exploring and visualising sample variables • bar plots

• pie / donut charts

• Outlier detection

51 © 2022 Institute of Data

Marginal Distributions of Discrete Variables # donut chart recipe ===

# The slices will be ordered and plotted counter-clockwise. data = [0.27, 0.67, 0.06]

labels = 'Low', 'Medium', 'High’

colors = ['yellowgreen', 'gold', 'lightskyblue’] plt.pie

| Income Bracket | |
| --- | --- |
| Low | 0.27 |
| Medium | 0.67 |
| High | 0.06 |

(data, explode=(0,0), labels=labels, colors=colors, autopct='%1.1f%%', shadow=False)

#draw a circle at the center of pie to make it look like a donut: 

centre\_circle = plt.Circle((0,0), 0.5, fc='white', linewidth=1.25)

fig = plt.gcf()

fig.gca().add\_artist(centre\_circle)

# Set aspect ratio to be equal so that pie is drawn as a circle:

plt.axis('equal')

plt.show()

52 © 2022 Institute of Data

Bar Plots

• **styles:**

• horizontal, vertical

• grouped, stacked

53 © 2022 Institute of Data

Conditional Distributions of Discrete Variables

• **contingency tables**

• 2D:

• var1 = rows, var 2 = columns

• 3D:

• var3 = planes (1 table for each value of var3)

• > 3D:

• multi-dimensional arrays

• can be represented in code even if we can’t visualise them

54 © 2022 Institute of Data

Lab 2.1.2: Data Profiling

• Purpose:

• To explore Python methods for exploring and summarising datasets

• Materials:

• ‘Lab 2.1.2.ipynb’

55 © 2022 Institute of Data

Exploring Large Datasets

• randomised sampling



• repeated sampling

• collect a number of random subsets from the sample population

• analyse each subset

• aggregate the results

56 © 2022 Institute of Data

The Central Limit Theorem •

57 © 2022 Institute of Data

Lab 2.1.3: The Central Limit Theorem

• Purpose:

• To test the central limit theorem by experiment

• Materials:

• ‘Lab 2.1.3.ipynb’

58 © 2022 Institute of Data

Time Series

• What is a time series?

• How are time series represented in Python?

59 © 2022 Institute of Data

Time Series

*def:* a sequence of data points representing the state of a system over time

classes of time series:

• temporally deterministic

• periodic

• pattern repeats at equal intervals

• aperiodic

• state at time *tk*is influenced by state at time *tk-1*

but there is no repeating pattern

• stochastic

• state at time *tk*is unrelated to state at time *tk-1*

60 © 2022 Institute of Data

Programming with Time Series

• timebase is usually regular

• seconds, days, or years (typically)

• may need to cope with leap years

• no gaps

• may need to impute or assign NA for missing time points

*example (Pandas):* 

index = pd.DatetimeIndex(['2014-07-04', '2014-08- 04', 

'2015-07-04', '2015-08-04’])

data = pd.Series([0, 1, 2, 3], index=index)

data

61 © 2022 Institute of Data

Visualising Time Series

Static time series 

• convert DataFrame to

Pandas time series

• timebase is an index of

the DataFrame

• default axis labelling is

aware of timebase

ts.plot()

62 © 2022 Institute of Data

Geospatial Data

• How are geospatial data organised? • Tools for exploring geospatial data • Visualising geospatial data in Python

63 © 2022 Institute of Data

Geospatial Data Formats

• GIS

• range of open (standard) and proprietary formats • raster, vector, grid

• metadata

• typically

• a list with nested structure

• arrays / lists

• coordinates

• attributes

• built-in (e.g. elevation)

• user-defined (e.g. derived statistics)

64 © 2022 Institute of Data

Geospatial Data Formats − cont’d

***Keyhole Markup Language***

• primarily used for Google Earth

• .KMZ/.KML

***Open Streetmap***

• largest crowdsourcing GIS data project of the planet Earth

• .OSM

***GeoJSON***

• open standard format designed for

representing simple geographical features

• .geojson

65 © 2022 Institute of Data

Tools for Exploring Geospatial Data

• interactive maps/ APIs 

• base map may be featureless 

• add ***tiles*** to display features 

• street map

• topography

• satellite view

• data organised, rendered in ***layers*** 

• ability to overlay image data from other sources • weather

• satellite view

• simulations

66 © 2022 Institute of Data

Geospatial Libraries for Python

Folium

• Plot maps

Shapely

• manipulation of geometric objects

Fiona

• read/write vector file formats (e.g. shapefiles or geojson) • projection conversions

Geopandas

• all of the above

67 © 2022 Institute of Data

Visualising Geospatial Data 

Geoplot

• works with GeoPandas

DataMaps

• interactive SVG maps using D3.js

68 © 2022 Institute of Data