BTS | Barcelona Technology School

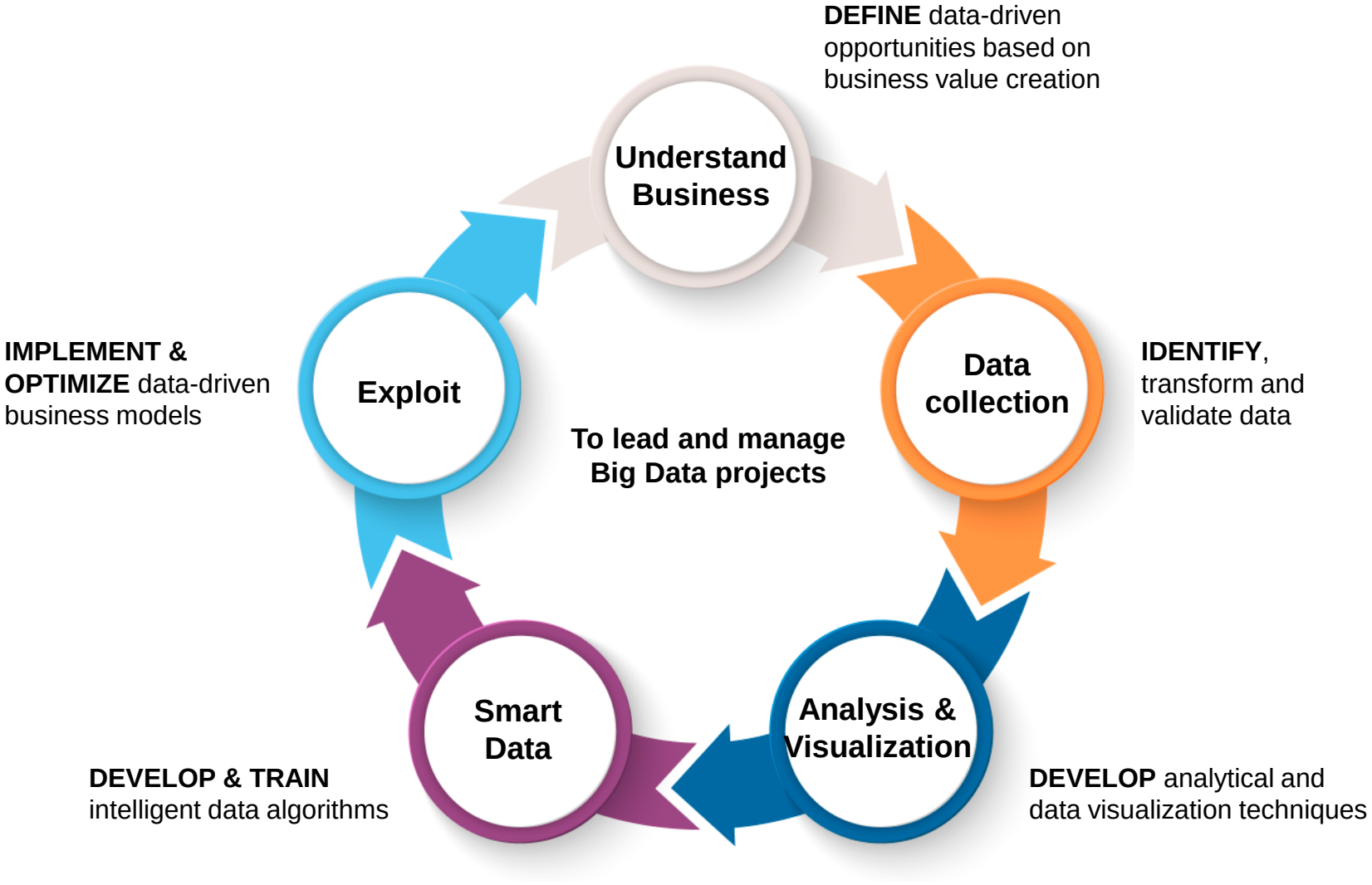
Data Science Foundations

Master in Big Data Solutions 2019-2020

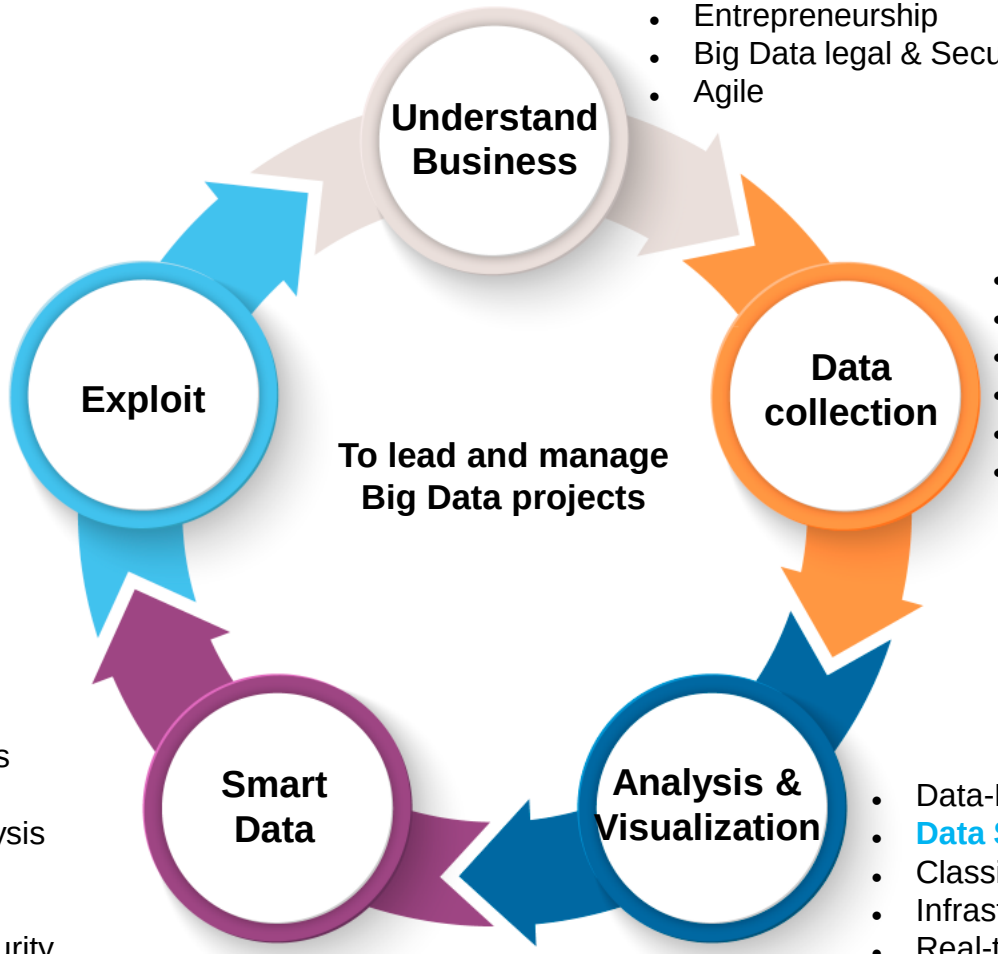
Víctor Pajuelo

victor.pajuelo@bts.tech

Project Lifecycle



Project Lifecycle



- Data-Driven Business
- Creative Technology
- Entrepreneurship
- Big Data legal & Security
- Agile

- Data-Driven Business
- **Data Science Foundations**
- Infrastructure
- Real-time Data Analysis
- Big Data legal & Security
- Agile

- Data-Driven Business
- **Data Science Foundations**
- Classical Data Analysis
- Infrastructure
- Real-time Data Analysis
- Big Data legal & Security
- Agile

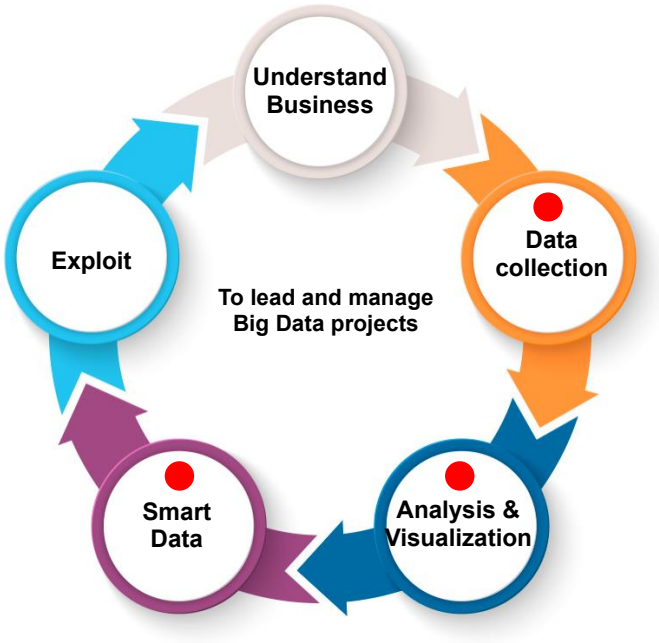
- Data-Driven Business
- Data Visualization
- Advanced Data Analysis
- Artificial Intelligence
- Infrastructure
- Big Data legal & Security
- Real-time Data Analysis
- Agile

- Data-Driven Business
- Advanced Data Analysis
- Infrastructure
- Real-time Data Analysis
- Big Data legal & Security
- Agile

Data Science Foundations

What we will learn

We'll learn how to collect, clean, process, manipulate, understand and visualize our data to accomplish our business objectives.

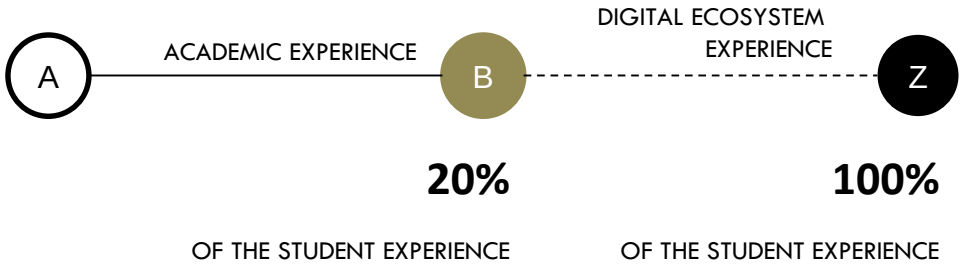


We will learn:

- Practical Data Manipulation in Python
- Basic data operations on relational data
- Loading and filtering text
- Loading and preprocessing images
- Data Manipulation
- Basic Data Analysis
- The process of data analysis
- Structured data types
- Overview of main analytic techniques
- The basics of putting a model into production

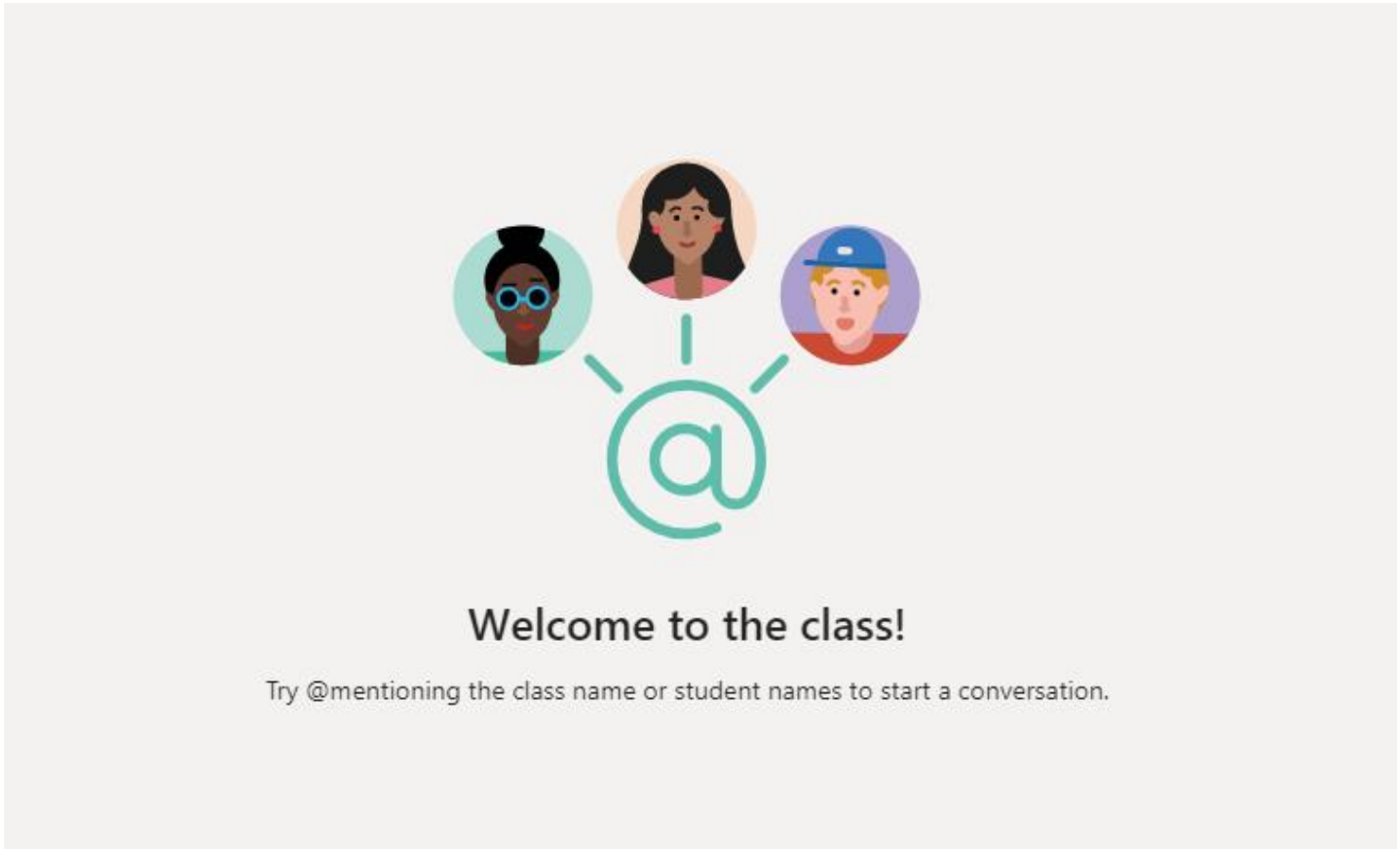
Student Experience

TRANSFORMATION SERIES	Industry congresses	Meetups	Keynotes	Visits to companies	BTS DEMO DAY



WORK ON YOUR PURPOSE - RESEARCH – DISCOVER – BUILD NETWORKING – JOIN COMMUNITIES – PARTICIPATE – GENERATE PUBLIC CONTENT

Curriculum review



A little bit about me

Introducing the course

Part I. Practical Data Manipulation in Python

1. Basic data operations on relational data (Pandas Introduction)
   - Recap of input/output and on-disk formats
   - Cleaning noisy data, normalizing
   - Filtering rows and columns
   - Joining data from multiple sources
   - Split-apply-combine workflows: groupby
2. Loading and processing images and text
   - Image loading, pre-processing and filtering
   - Image pre-processing for object detection and segmentation
   - Text pre-processing, normalization, stemming, stopword removal
   - Converting text to vectors and computing text similarity
3. Date manipulation
   - Basic data types, timezones
   - Resampling, shifting and windowing

Part II. Basic Data Analysis

4. The process of data analysis
   - Exploratory data analysis and insights finding
   - Feature engineering
5. Data types
   - Image processing
   - Time series analysis
   - Text processing
6. Overview of main analytic techniques
   - Regression on time-series, text and images
   - Classification on text and images

Part III. Data Science in production

7. The process of ETL (Extract Transform Load)
   - Data gathering and scraping
8. Packaging your model
   - Model evaluation and deployment
   - Containerize the model
9. The last mile to user: visualization and delivery
   - Visualization and reporting
   - Common delivery platforms

About the teaching methodology

Teaching methodology

- Three sessions of one and a half hours per day, divided in:
  - Theory (usually supported by notebooks)
  - Theory (usually supported by notebooks)
  - In-class exercises (supported by notebooks)

- One assignment per week presented on Thurday's class and due next Thursday..
- Using the first 30 minutes of every Thursday class to comment the solutions.

- Using this git repository for notebooks

Today’s class

Today's Objective

- Understand the basic abstractions of pandas
  - Series, DataFrames and Indexes
- Explore techniques to read standard text files
- Learn how to do basic data exploration
- Do basic analytics with pandas

- Why is this useful for a digital project?
  - Understanding data manipulation is crucial for any data science project

Contents

1. Basic data operations on relational data (Pandas Introduction)
   - Recap of input/output and on-disk formats
   - Cleaning noisy data, normalizing
   - Filtering rows and columns
   - Joining data from multiple sources
   - Split-apply-combine workflows: groupby

Pandas introduction

Pandas

Python Data Analysis library

Pandas

Python Data Analysis library

Pandas is a **Python package** providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.

It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python.

Additionally, it has the broader goal of becoming **the most powerful and flexible open source data analysis / manipulation tool available in any language**.

 Built on top of NumPy! 

Source

Pandas

Data Suitability

- **Tabular data** with **heterogeneously-typed columns**, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) **time series data**.
- **Arbitrary matrix data** (homogeneously typed or heterogeneous) with row and column labels
- Any other form of **observational / statistical data sets**. The data actually doesn't need to be labeled at all to be placed into a pandas data structure

Source

Pandas

Why pandas is very adequate for data analysis?

- **Easy handling of missing data** (represented as NaN) in floating point as well as non-floating point data
- **Size mutability**: columns can be inserted and deleted from DataFrame and higher dimensional objects
- **Automatic and explicit data alignment**: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform **split-apply-combine operations** on data sets, for both aggregating and transforming data
- **Easy conversion of different formats:** differently-indexed data in Python and NumPy data structures into DataFrame objects

Pandas

Why pandas is very adequate for data analysis?

- **Intelligent label-based slicing**, fancy indexing, and subsetting of large data sets
- **Intuitive merging** and **joining** data sets
- **Flexible reshaping** and **pivoting** of data sets
- **Hierarchical labeling of axes** (possible to have multiple labels per tick)
- **Robust IO tools** for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
- **Time series-specific functionality**: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging, etc.

Pandas

Main data structures

Series

0.25	0.5	0.75	1.0

Value mutable / Size immutable

Index

A	B	C	D

Value mutable / Size immutable

DataFrame

	columns
A	0.25
B	0.5
C	0.75
D	1.0

Value mutable / Size mutable

Pandas

Main data structures can be understood as:

Series	DataFrames	Index
As a 1-D NumPy array	As a 2-D NumPy array	As an immutable array
As a specialized Python dictionary	As a specialized Python dictionary	As an ordered set
	As a single Series on a DataFrame	
	As a structured list of dicts	
	As a dictionary of Series objects	

Pandas

Data Structures in code!

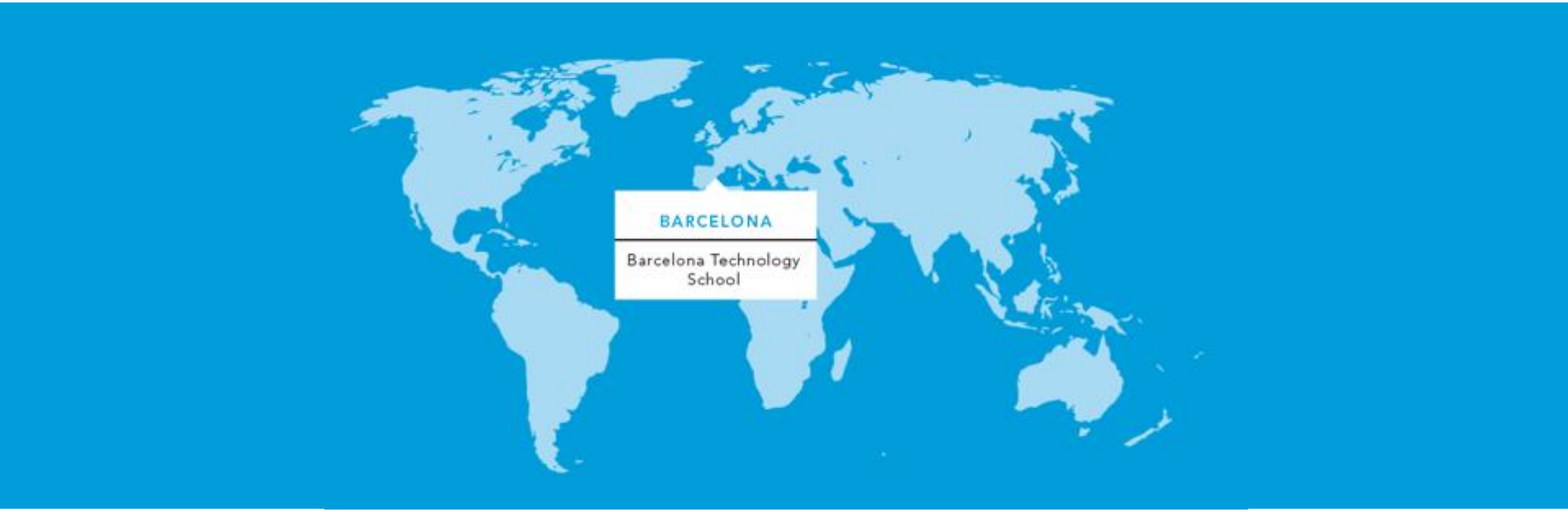
Go to the notebook here

Pandas functionality introduction and Class exercise

Pandas

Intro in code!

Go to the notebook here


BARCELONA
Barcelona Technology School