Introduction

To make it easier to talk about EDA, it can be good to separate ***the what***, ***the why***, and ***the how.***

At a high level, ***why*** we want to perform EDA are:

- To understand the data better and map it to what we know (or don't know) about the domain. As a consequence, learn about the domain too!
- **To make and test assumptions about the data.**
    - High-level: What is planned to be used to solve the problem?
    - Assumptions of algorithms, metrics
    - Assumptions data distribution
    - We have enough of feature X (where X is something we considered important)
- To clean and transform data as necessary for further work down the line.
- To inform later modeling decisions.
    - Algorithm Choice
    - Metric Choice
    - Handling Missing Data
    - Data Balancing
    - Transformation?
- …

Now, onto the ***what***. The usual things we are interested in are in the table below.
(Note: Please also think about the ***why*** of each individual topic listed here. E.g: "Why do we want to treat missing values?")

	Topic	**General**	**Specific**
1	Variable Identification	What do each column represent? What *type* of data?	Do we *understand* each column (maybe it's something domain specific)? If not, ask!
2	Univariate Analysis	How is each column in isolation?	Do the values/distribution make sense?
3	Bi-variate Analysis	How do pairs of columns relate to each other?	Does what we see make sense? Which pairings specifically we might want to focus on?
4	Missing values Identification	How much data is missing in each column? Related: why?	Is it missing at random or there's a pattern? What could be the business reason for it?
5	Missing Values Treatment	How should we handle missing values?	
6	Outlier identification	What kinds of outliers are present in each column?	Are the outliers "true" outliers or errors of measurement? What is the business reason for the

			outliers?
7	Outlier treatment	How should we handle the outliers?	
8	Variable transformation	What transformations should we perform on the variables?	
9	Variable Creation	Any new features we could create?	Are there “intuitive” (for the domain) features we could engineer?

General Tips

- Pandas
    - You usually shouldn’t have to loop through the rows. There should be a vectorized method. *Usually.*
    - Be comfortable with `fillna`, grouping, aggregating, apply, map, pivot, melt, etc.

- Make use of global configurations. E.g: Setting `rcParams` for matplotlib.

- Don’t assume. Test your assumptions. Either through:
    - Visualizations
    - “Statistical” Tests
    - Asking the Domain Expert
- If there’s no way to verify the assumption currently, still explicitly spell it out. E.g: “France seems to have a lot of data missing. We assume all of the analysis also holds for France i.e it is not a special case”.

- Make sure you understand ***what*** analysis you are doing (at each point) and ***why***. A good practice is to write out what you are looking for/at, why you are doing so and then what you found.

- Plots:
    - Choose the right type of plot. Every type shows a slightly different information. Does it match what you are trying to show/see?
    - Be aware of the figsize. Is it looking crammed or maybe unnecessarily wide?
    - Label your plots properly. Especially if it is gonna be used to communicate with others.
        - Are the axes labeled?
        - Is there a legend (if necessary)?
        - Is there a title?
        - Is the font legible? E.g: Is the size okay?
    - Take a moment to make sure the plot makes sense. Can you explain *exactly* what it is showing?
    - Also, address the findings. What does the plot say? Don’t just plot and move on.

- Tools

    - Matplotlib for fine-grained control
    - Pandas plotting for quick stuff. Note: you can (and probably should) pass in the Axes object i.e `ax`.
    - Seaborn for quick and (usually) beautiful stuff.
    - Pandas-profiling for a pretty comprehensive first look. In fact, it covers a lot of sections 1 to 4 above.
    - Plotly if you want something interactive
    - …

- Notebook Hygiene
    - Format it well. E.g: use of section (markdown) headers, **bold,** *italics,* etc.
    - Refactor as you perform the analysis. Throw away unnecessary cells. Create functions for common things. (In due time, move them out into modules).
    - Re-run periodically (unless it's slow), especially before you commit (to make sure it actually runs sequentially end-to-end). If the outputs are many and large, it can be good to clear them before commit so as to not have an over-bloated git repo. Or use a git hook running this.
    - Put in adequate commentary. Your future self will thank you.

- Process
    - Can be good to create a "Template Notebook" with some base imports and configurations that you can duplicate and work on.
    - Don't (I repeat, don't) work on the same notebook in two places. Merging will be hell.
    - Name the notebooks well. The cookiecutter naming convention can be good.
    - Store data well. Perhaps in DVC or just in a well organized archive somewhere so that you can reproduce analysis later if necessary.

Tools

- Pandas visualization
- seaborn
- pandas-profiling
- plotly

Resources

- Data Exploration Guide
- Data Science Handbook

Choice of Plot

- Choosing the chart type
- Pie Charts are bad

Examples

- Exploration starter example
- NFL Punts Analysis (Mostly for the good incorporation of Domain knowledge)