****

# 

# Data Science

# Assignment # 1

# Group Members

Noman Anjum 17K-3755 Section D

Mustafa Manga 17K-3795 Section D

# **Q1 page # 2**

# **Q2 page # 3 - 11**

**Q1)**

**Human Centric Cleaning (An Overview)**

This paper discusses frameworks, design sand methods, for data cleaning operations. Data cleaning majorly involves detecting duplicates, missing values, integrity constraints etc. Data collected from various sources including businesses inherit errors so often. Using this data for inference or prediction can lead to incorrect decision in business and other fields. A number of researches have focused detection and repair of incorrect data, however these all system involves human intervention for effective cleaning. Considering the irony that humans can’t correct enormous data, and rule based algorithms are not very good at cleaning and correcting domain centric data, we can’t rely totally on computers, neither humans can be relied for correcting huge number of errors.

For an effective data cleaning pipeline human involvement is proposed to work along with the system. An efficient cleaning system supports human, automatic and semi-automatic cleaning agents. It isolates humans from logics of cleaning algorithm and sets accountability metrics so as to identify bottlenecks. It must account for cost and expertise when involving human in a task to optimize it. Basic architecture proposed in paper suggest four primary components Detector, Repairers, Cleaning Resources and Validators. Detectors and Repairers are treated as pluggable black boxes. Detector are programs/humans which detect errors upon certain constraints and algorithms, repairers update data with correct values, cleaning resources are utilized by repairers, validators cross check updated values against some ground truth or feedbacks.

Humans have different degrees of expertise on different parts of data, so the role of human should be very clearly identified while engaging human in process. A data user or detector should not necessarily be a technical person, but a repairer must have required technical knowledge to update data without bringing up new errors. A validator should have knowledge about data they are validating but not necessarily need technical knowledge. A person who writes specifications needs to be domain expert to write accurate specifications. A person who has domain specific knowledge about data is the best candidate to engage in data cleaning process and his efficiency and expertise can be measured with different formulas. On the other hand human budgeting is also necessary to effectively calculate cost of each human involved and find optimal number of humans to involve in process. A good interaction model between humans is needed to minimize communication overhead between human roles and for accountability for all human-to-human interaction. Another important factor to minimize cost is to avoid black box repairing algorithms so human is aware of working of agent and can examine operational logic of algorithm on data and can evaluate repairing and perform further tasks.

Quantitative cost optimization for data cleaning tasks needs proper examination of overlapping of operations between human and automatic agent, considering the fact that quality of human repair is better than agent whilst an agent can clean huge data, we can assign only sensitive features to human for cleaning and leave rest to agent. Cutting human cost ultimately can result in less reliable results. In cases where human and agent activity overlaps, considering the fact that cleaning performed by human is more reliable and accurate, agents must not dominate humans. Humans should be prioritized in the cleaning process and for better data repair quality, system should invoke automatic agent first and then ask human to correct overlapping cells, human updates should be ordered last and must not be undone by automatic agent.

# **Q2)**

# **Data Cleaning (Tools vs. Code)**

# Data cleaning is considered as one of the crucial part of data science, researches states that around 70% - 80% time of data scientist is spent in cleaning the data and making it consistent. Simpler and cleaner the data, more the chances of good inferences and predictions.

# We have performed cleaning operations on data of lab 2 and lab 3 along with the HCV data from UCI website. The data set contains laboratory values of blood donors and Hepatitis C patients and demographic values like age[[1]](#footnote-1). Cleaning was performed with the help of two tools and raw python to compare the effectiveness of tools and code. Following are some extracted key points.

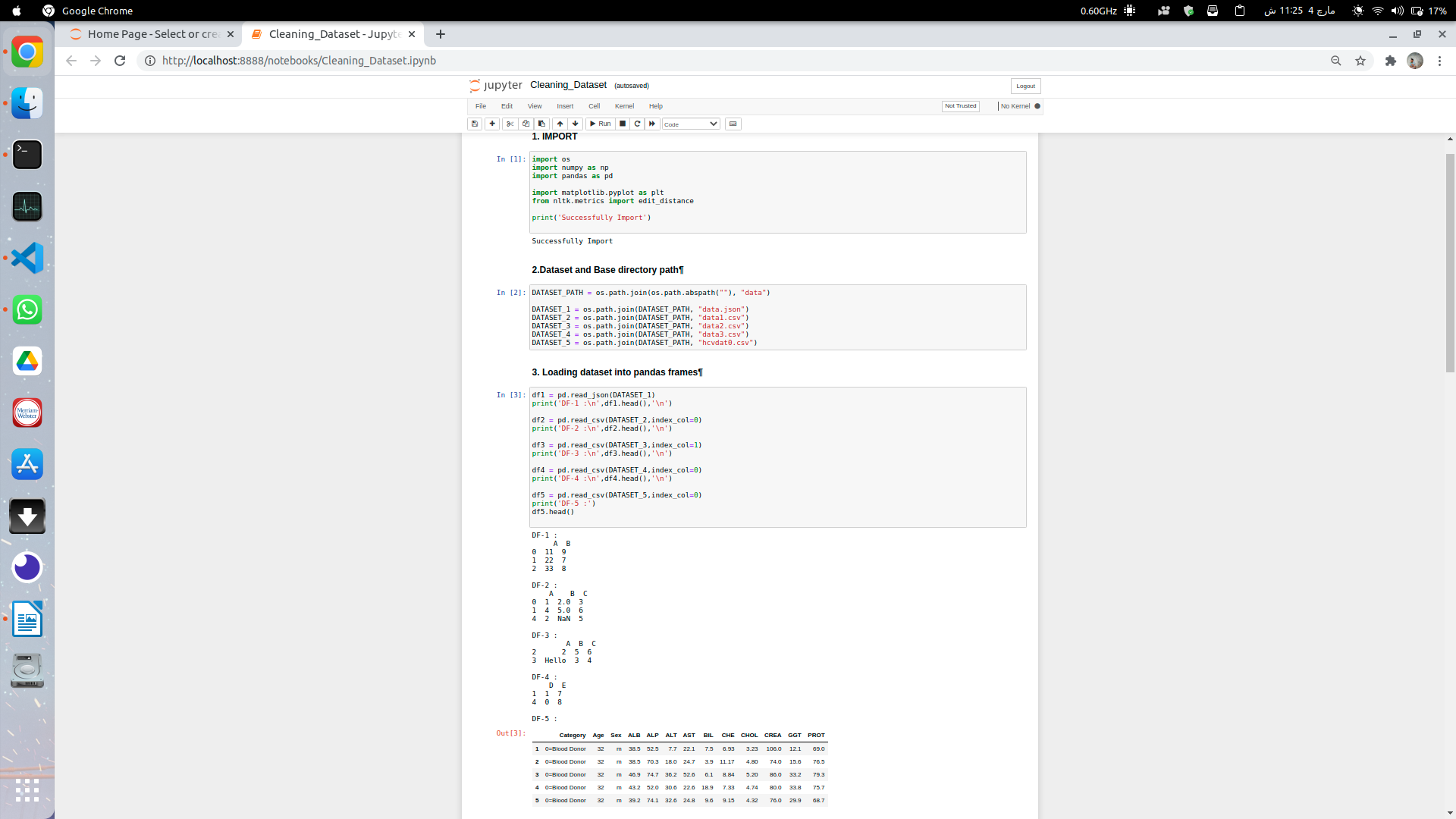
# **Tools Used:**

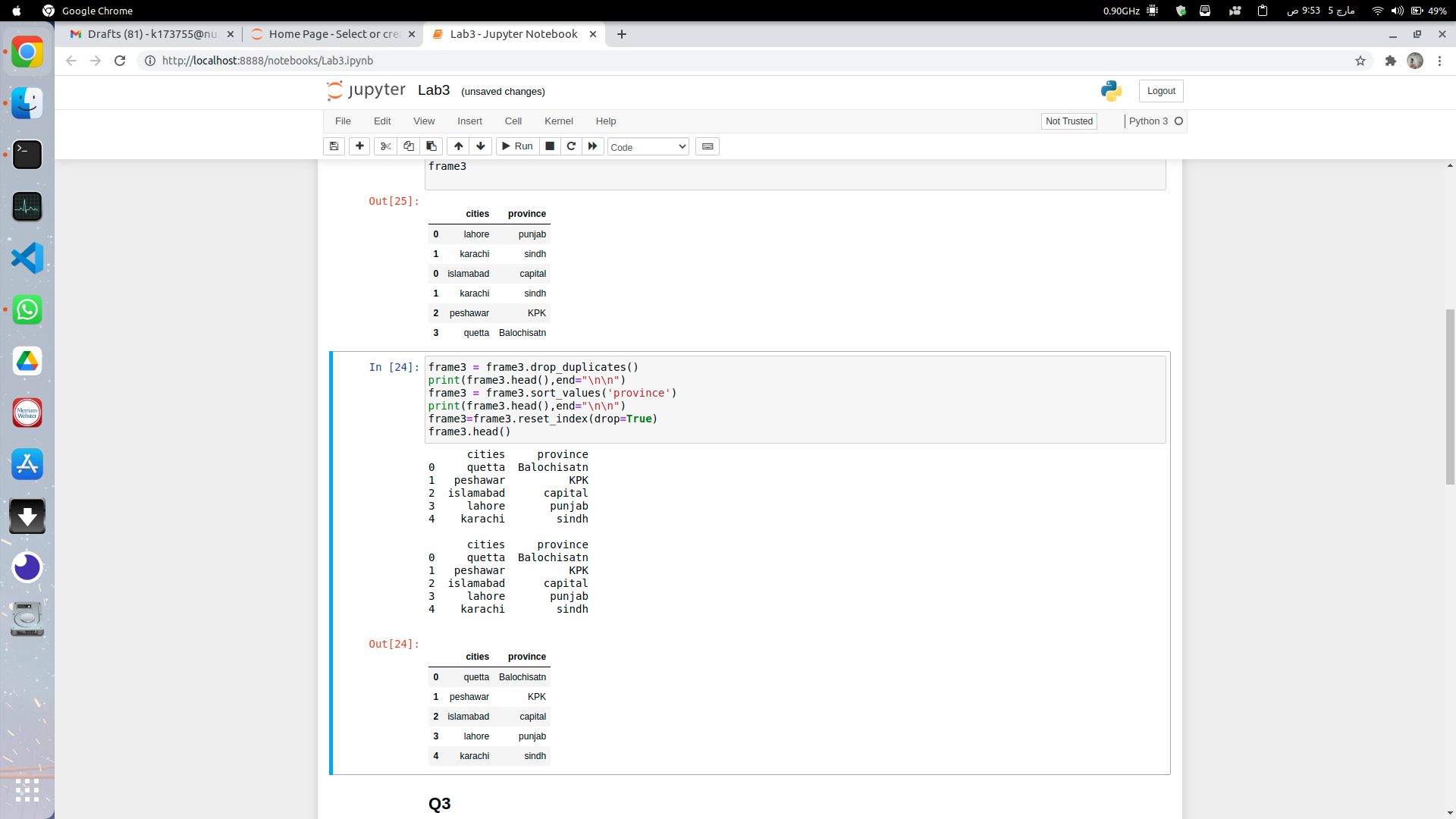
# **Open Refine**

# **Trifacta**

# 

**Python script**



****

**Comparison**

**Simplicity**:

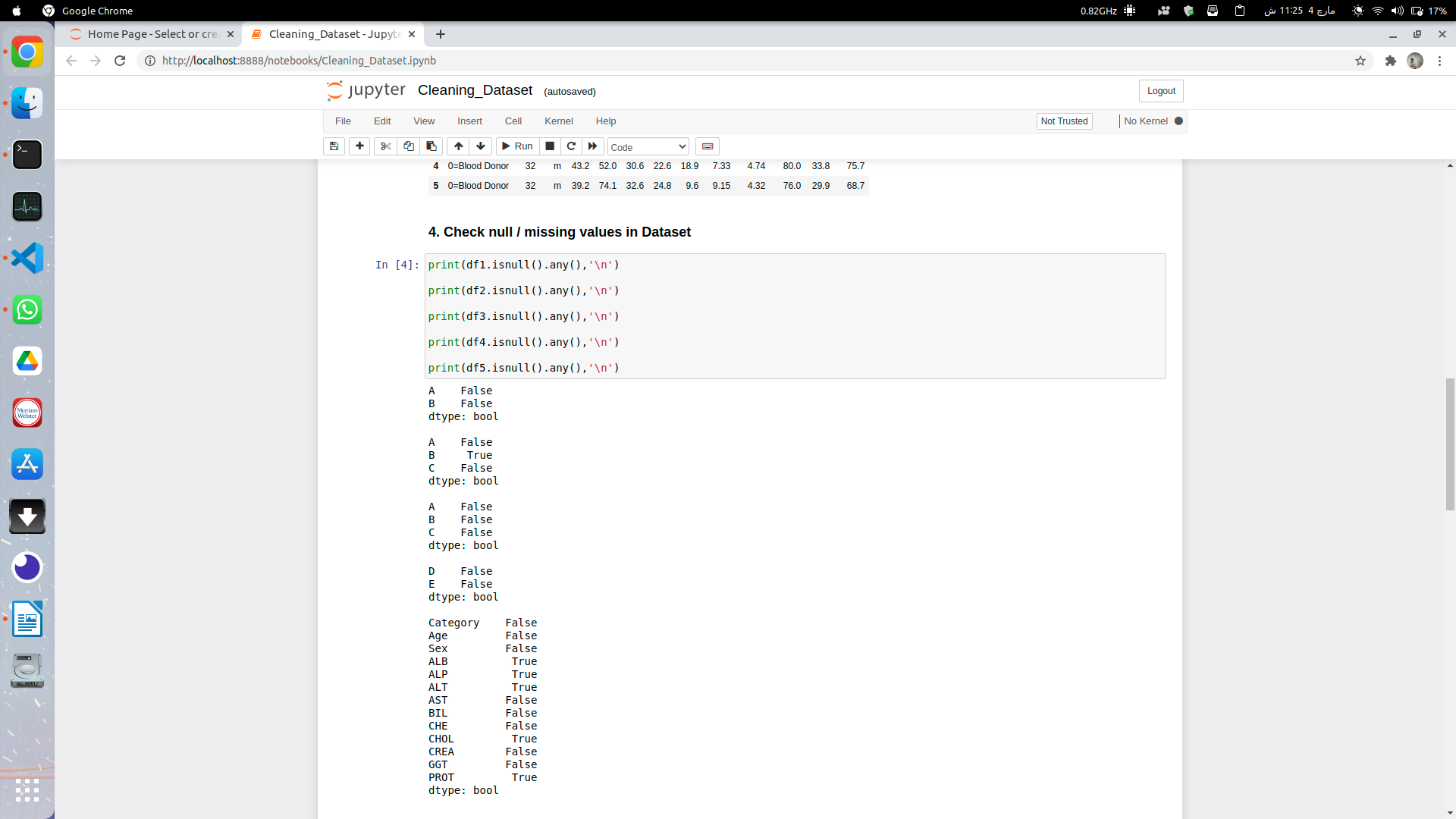
Tools are no doubt the best choice when it comes to simplicity. Cleaning with python is bit difficult as it needs technical expertise along with python language knowledge. Tools on the other hand provide a very simple interface to perform cleaning tasks which can be understood by a person having little knowledge of data and its problems. Tools have automated visualizations, which provide really good insight to data specific problems.

**Size of Dataset:**

Tools limit size of dataset to a particular extent and crossing the limit results in very poor performance as compared to python. Tools use much more resources as compared to simple python hence a good amount of data can be loaded and cleaned in python with a limited use of resources.

**Missing Values:**

Open Refine doesn’t provide any clue about missing values unlike Trifacta, we have to create scatter plot for missing values by using some, functions. Trifacta highlights column header with red color in case it has missing values. Same can be obtained in python with a single line code.



# **Graph Visualization for Data Distribution:**

# Open refine provides icon functionality to draw histograms over data, Trifacta a smarter tool make histograms over column headers by default to visualize data distribution. Python on the other hand requires some lines of code and libraries to do same operation. Tools perform better for this scenario however they provide only specific number of plots, whereas python has numerous libraries to visualize data in required form.

# 

# **Replacing Values:**

# Open refine doesn’t perform well at replacing values. It requires regex for matching Input, and output string for replacement. Regex is generally hard to write for even intermediate users. Same operation can be performed in python with regex too. Tool doesn’t give a better solution for this cleaning problem.

# 

# **String Operations:**

# Trimming, replacing, lower/upper casing has been provided by both tools and python, tools asks to select a column and perform certain action on it, same can be done with python using 1 line of code. In addition some advance string operations like converting to title case, split and select etc. are not provided in tools. Python performs better for variety of string operation with added complexity.

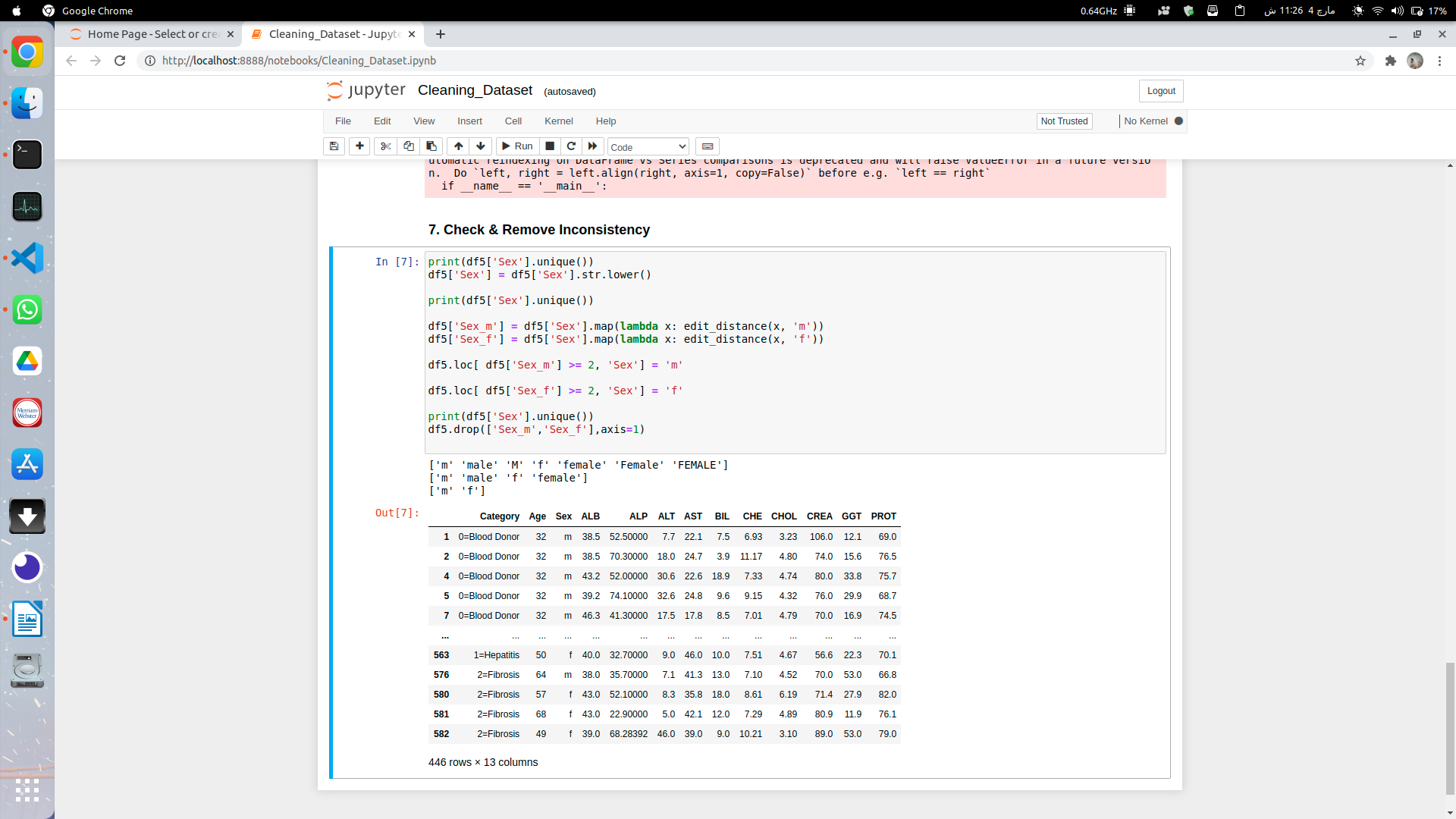
# **Outlier Detection:**

# Only method available for outlier detection in tools is making clusters for data values and finding values which are far spread. Outlier replacement can be then performed manually by hand, hence no tool is provided to perform this action. Python is very handy at outlier detection, a number of metrics can be applied to find outliers, and removal and replacement of outliers can also be done with some lines of code along with a variety of choices.

# Screenshot from 2021-03-04 23-26-06

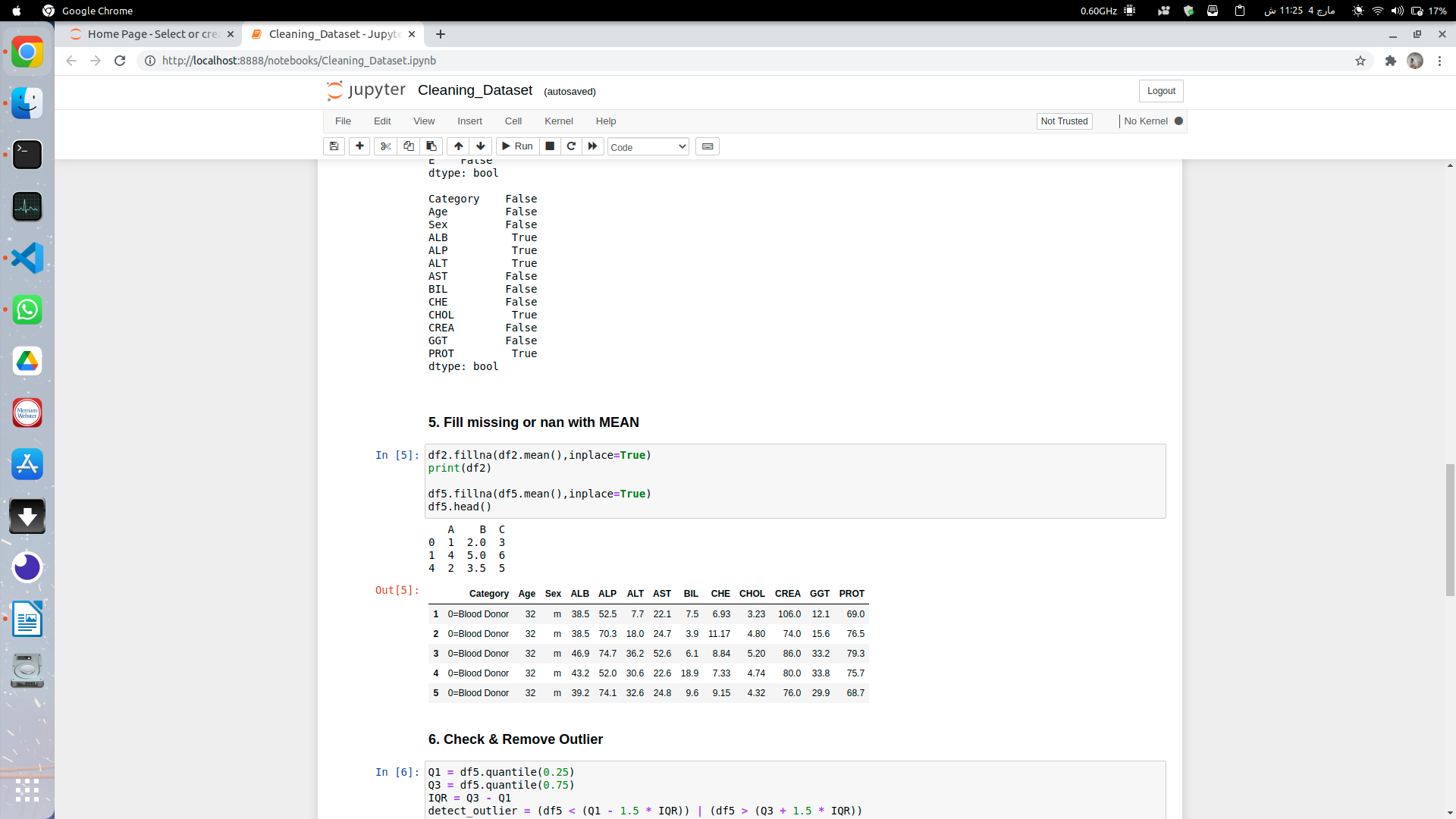
# **Inconsistency Removal:**

# Tools returns unique values for every column and a replacement can be performed to remove these inconsistencies. These operation requires two to three lines of code in python. Hence tools doesn’t provide a better option here.



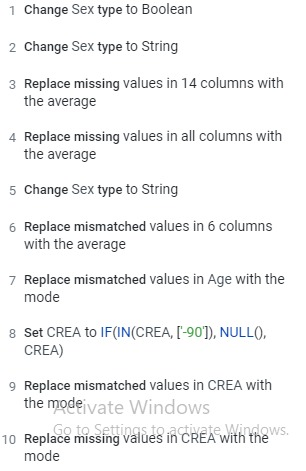
# **Removing Missing Values:**

# Tools are good at removing null/missing values. User has to select a column and provide values to be inserted in place of missing values. Python has a fillna () function which provides this functionality. However python is very flexible at choosing values to replace. So tools are easy python is flexible in contrast.

**Cleaning Recipe:**

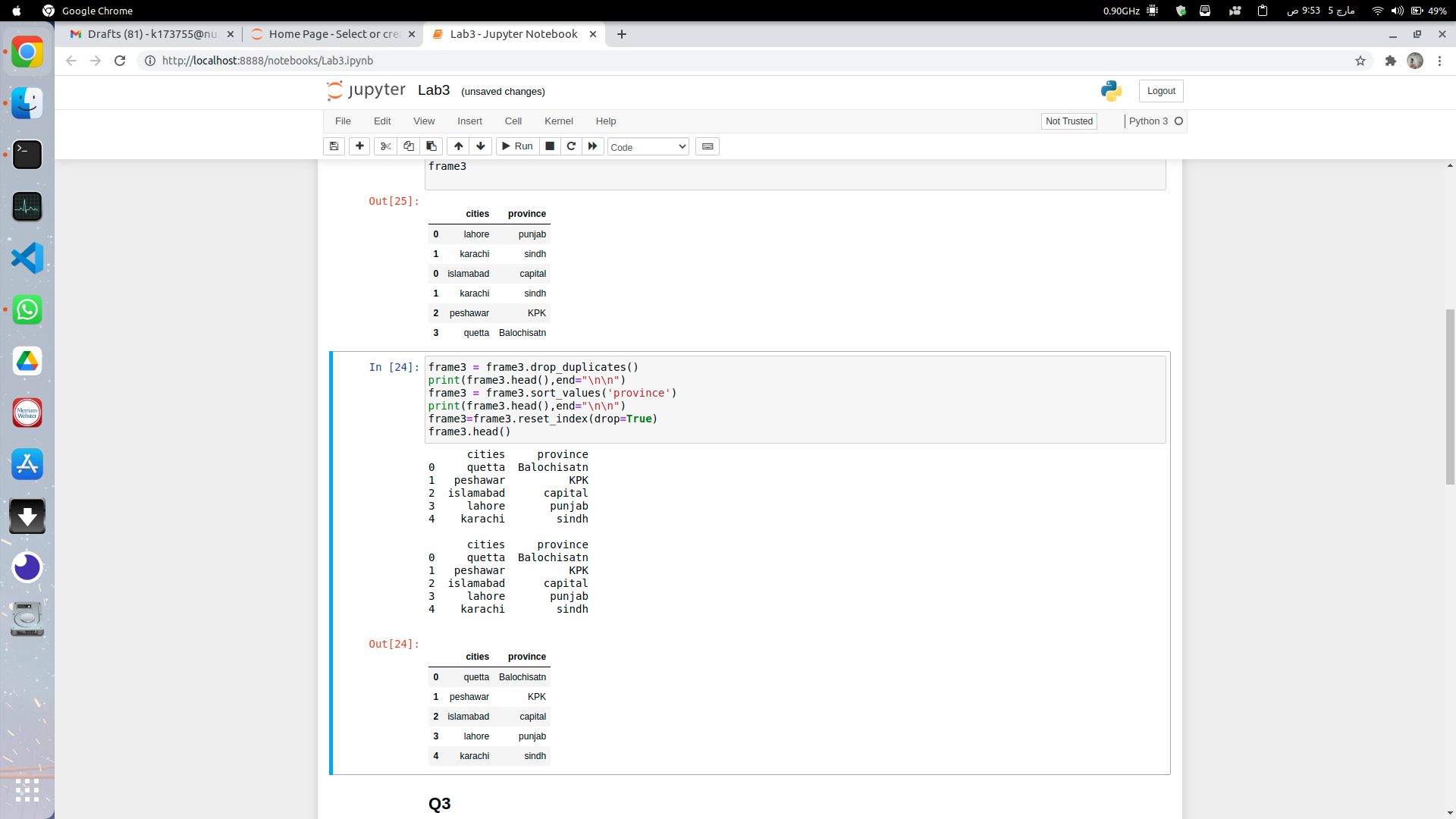
Python code is a bit complex for a naive user hence a user can’t predict operations performed over data while cleaning, Tools on the other hand provide a very simple recipe of cleaning for cumulative operations. Which makes it easy to explain to a naive user.

**Tools:**



# **Python:**

# Screenshot from 2021-03-04 23-26-12



# Screenshot from 2021-03-05 09-54-01

# **Conclusion:**

# Data cleaning tools are great when it comes to simplicity and ease of operations. They have very simple interface which a novice user with a little knowledge of data problems can understand and use efficiently. Easy of visualizations in data is also a plus for tools. However tools are limited to certain functionalities. They don’t have variety of visualizations, replacing functionalities, outlier detectors etc. Python on the other hand is a bit technical and complex. A novice user cannot clean data with the help of python, however python has a clear edge when it comes to flexibility. Hundreds of libraries are available to detect outliers based on different metrics. Hundreds of visualization graphs are available to get better insights of data and related problems. With python code user can mold data in any shape and form and then work accordingly.

# In short tools are great for novice users, while only coding or an extremely high end tool which drains a lot of resources can perform all the task which modern data scientists perform in their daily operations.

1. <https://archive.ics.uci.edu/ml/datasets/HCV+data> [↑](#footnote-ref-1)