Part A

# Suman Sourav Bastia

## Problem Statement:

We have a dataset “daily\_offers.xlsx”. It represents steel items offered to customers. Each row represents an item, and has an offered/selling price of that item. We have to do following things:

* Analyse all the features, using statistical and visualization methods of our choice and prepare the data for a regression model.
* Split the data into training and testing sets in the ratio of 90:10 respectively. Build multiple regression models to predict the price of an item (At least 4 models).
* Evaluate the best model using the metric r2 score on the testing set.

## Hardware Information:

|  |  |
| --- | --- |
| Device name | LAPTOP-LK8AF9T1 |
| Processor | 11th Gen Intel(R) Core (TM) i9-11900H @ 2.50GHz 2.50 GHz |
| Installed RAM | 16.0 GB (15.8 GB usable) |
| System Type | 64-bit operating system, x64-based processor |
| Storage | 1TB SSD |

## Software Information:

|  |  |
| --- | --- |
| Operating System | Windows 11 |
| For Python Programming | Jupyter Notebook |

## Concept Development:

* Input Collected Data
* Data Pre-processing
* Data Visualization
* Model Building
* Model Comparison

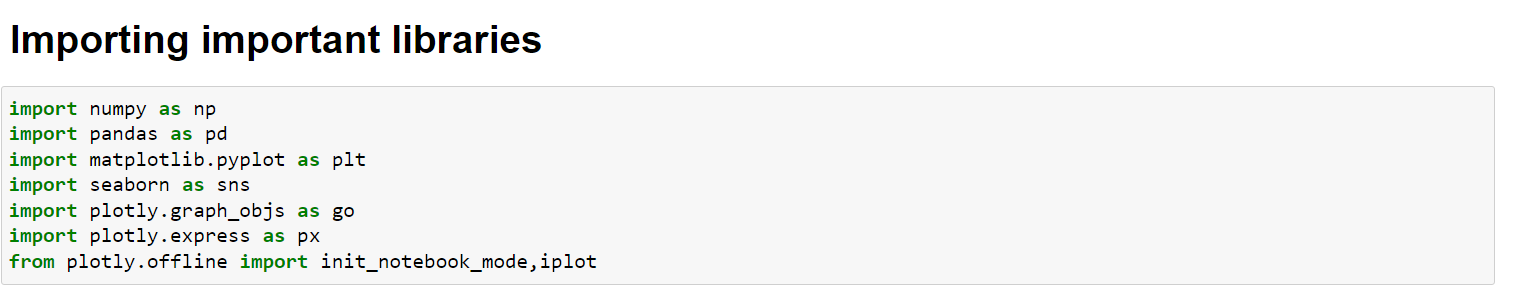
## Libraries:

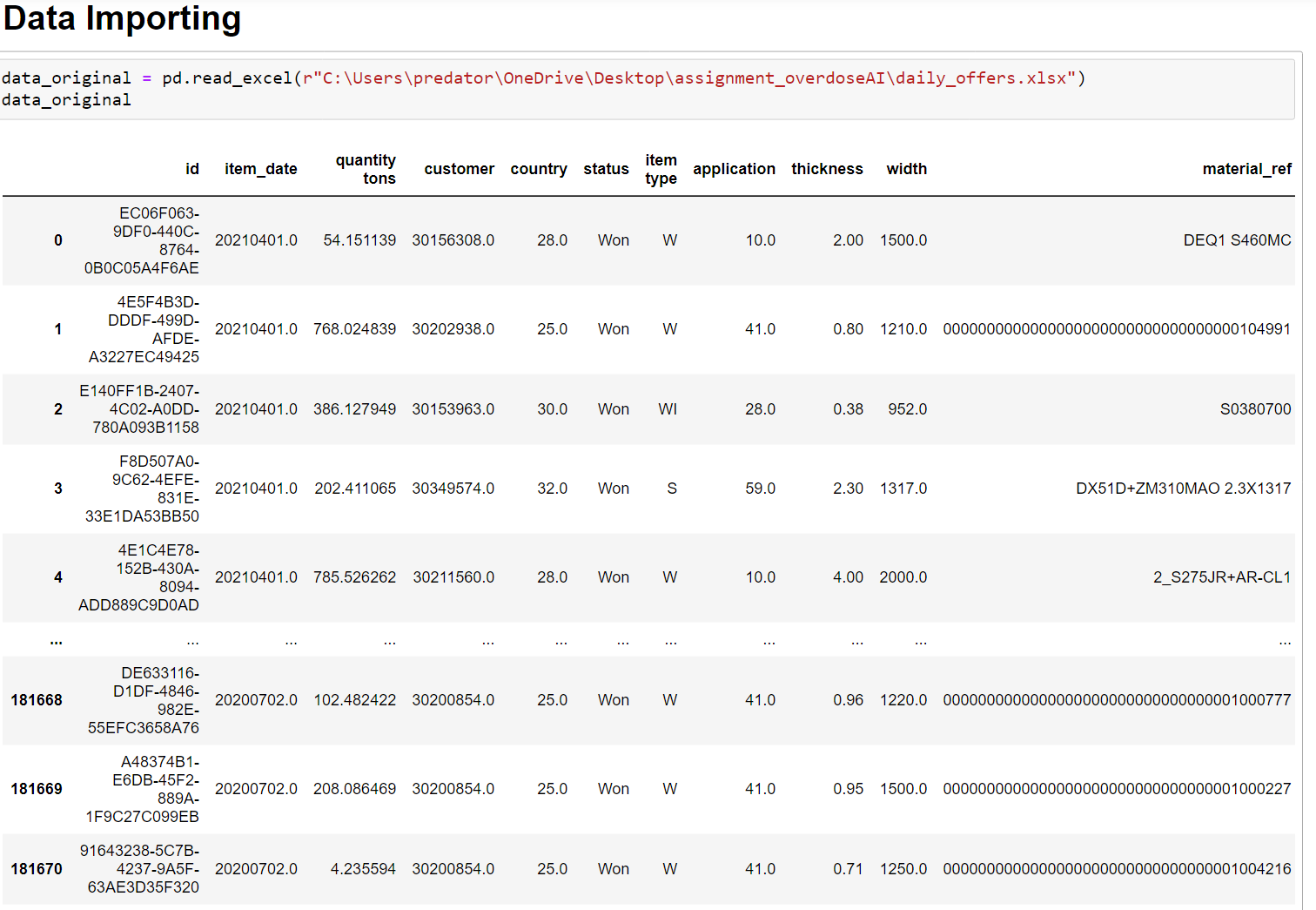
* **Pandas** is used for important activities such as importing files. It also provides read and write data tools between memory data structures and different file formats.
* **Numpy** is a Python library used to work with arrays. It also has functionality in the field of line algebra, Fourier transform, and matrices.
* **Matplotlib** is a python library used to create 2D graphs and sections using python scripts.
* **Seaborn** is an open-source Python library built on top of a matplotlib. It is used to view data and to analyse experimental data. Seaborn works easily with data frames and the Pandas library.
* **Scikit-learn (Sklearn)** is a very useful and powerful mechanical library in Python.

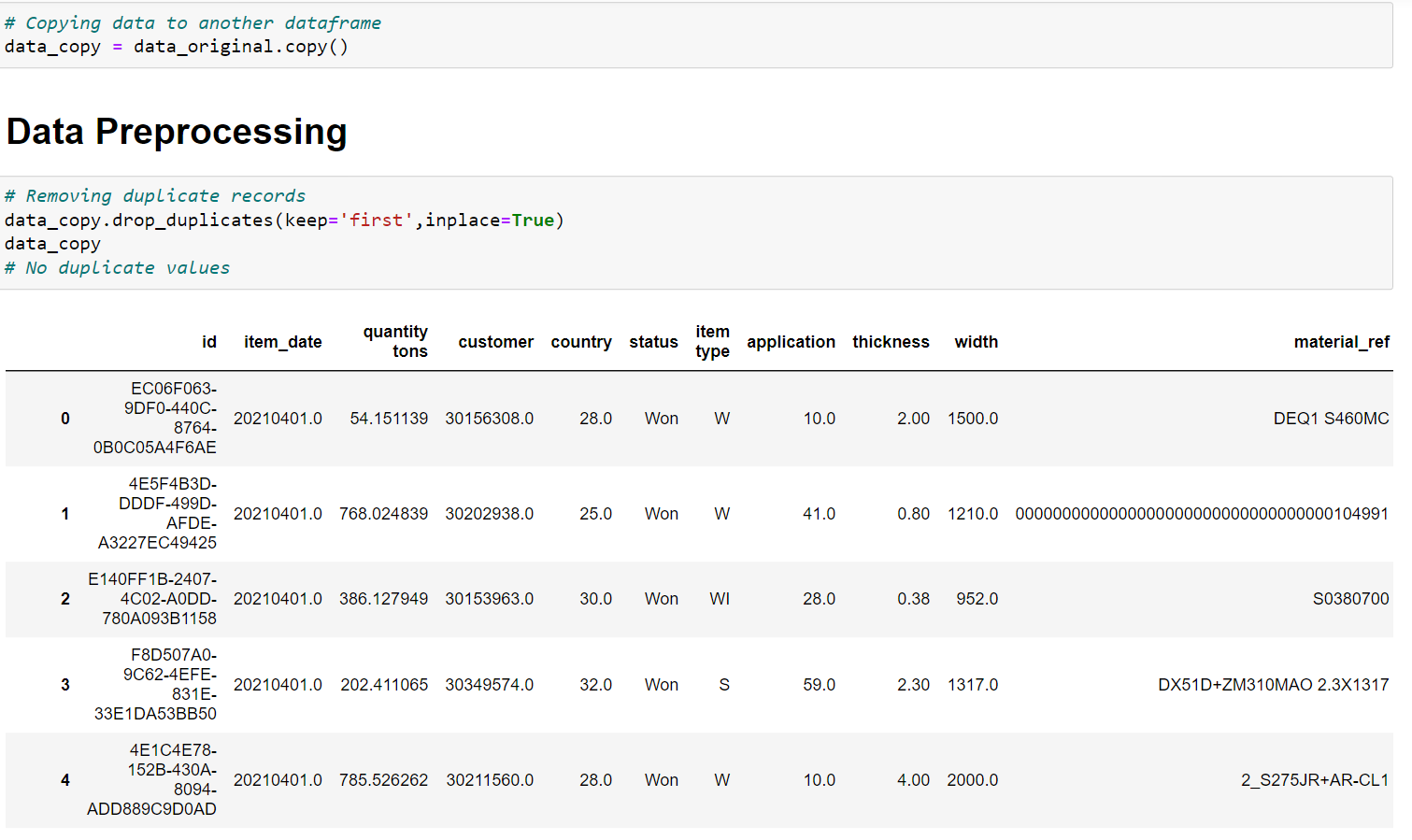
## ML models:

* **Linear Regression**
* **KNN**
* **Decision Tree**
* **Random Forest**
* **Gradient Boosting**
* **Ada Boost**
* **XG Boost**

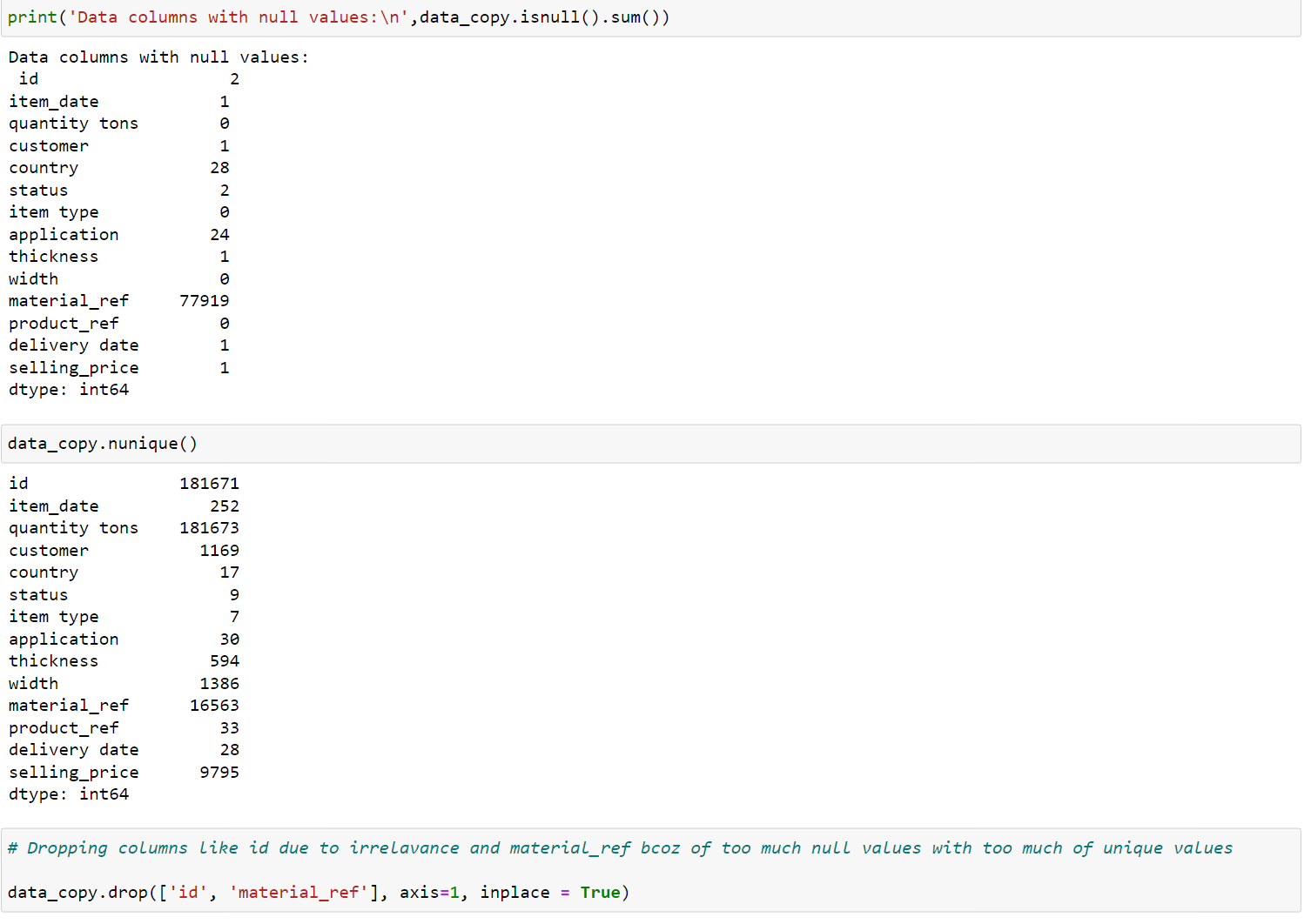
## Code Implementation:

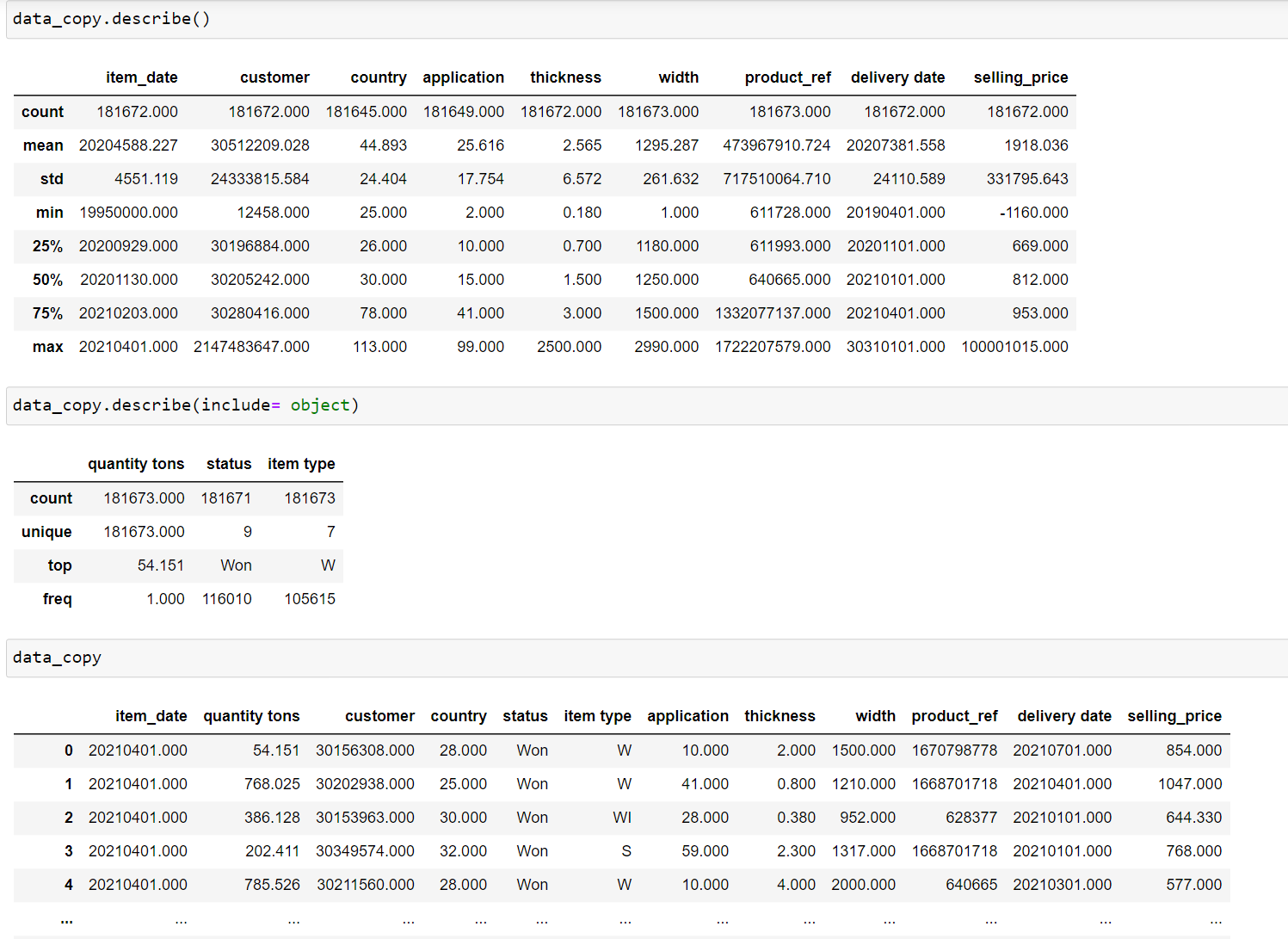


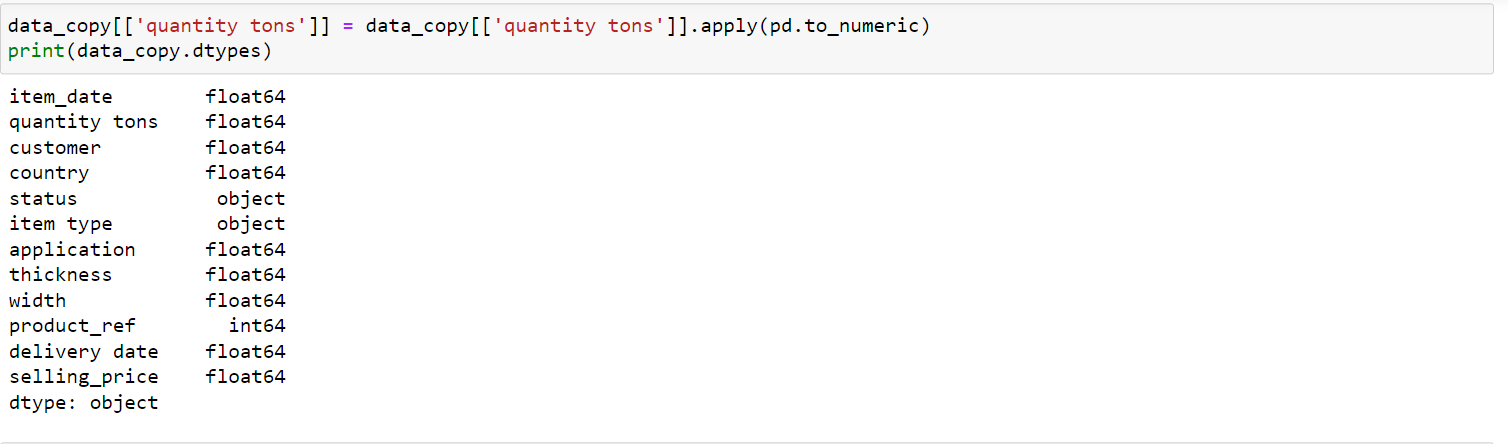


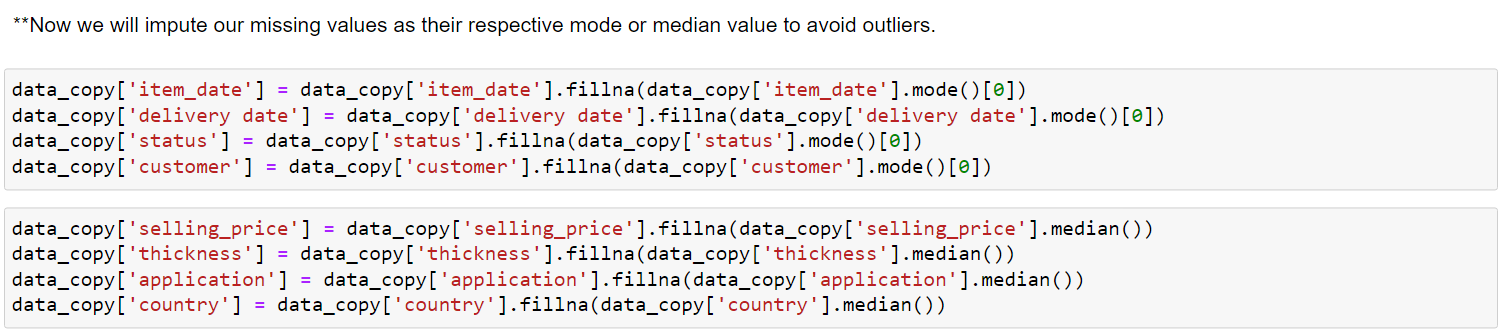


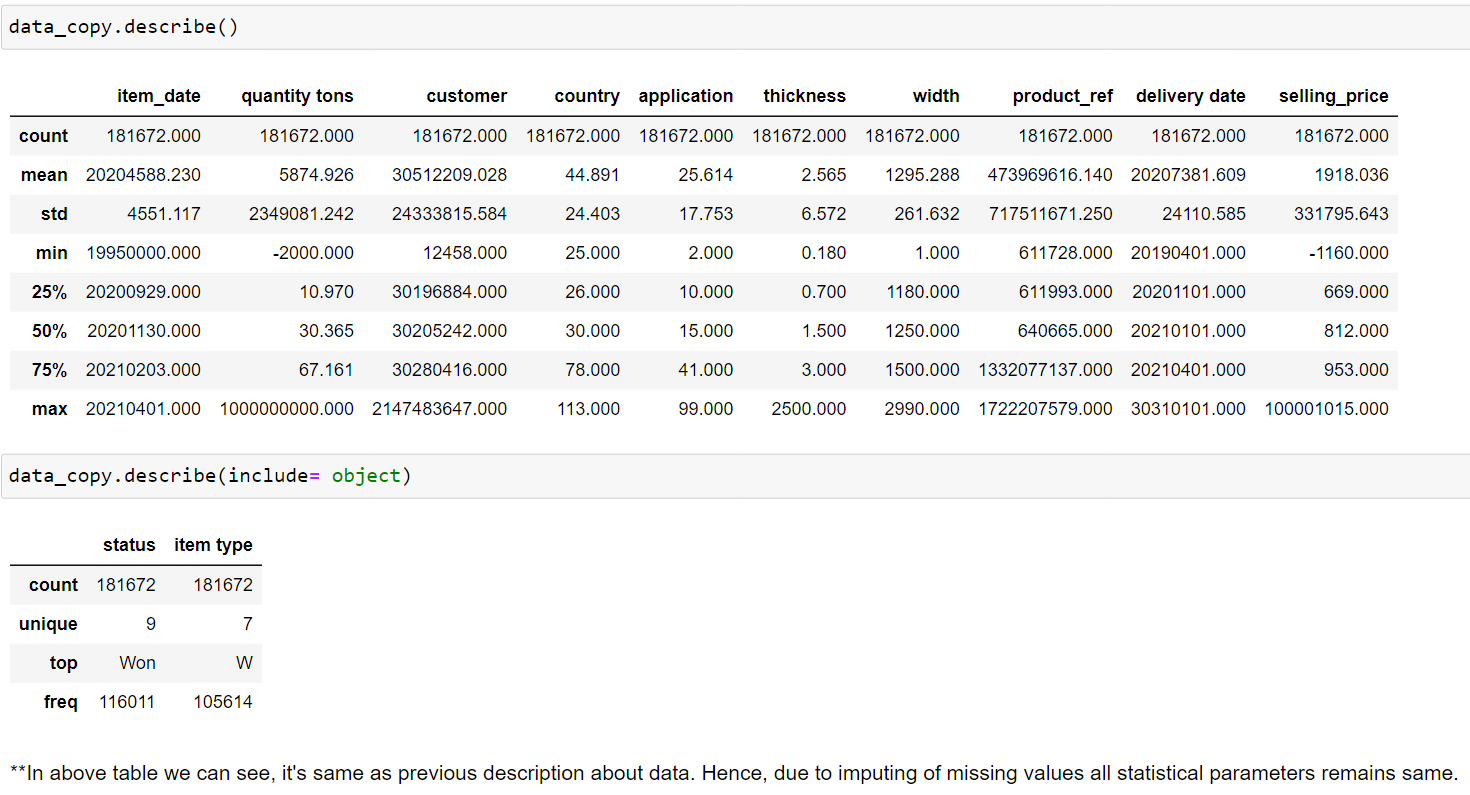


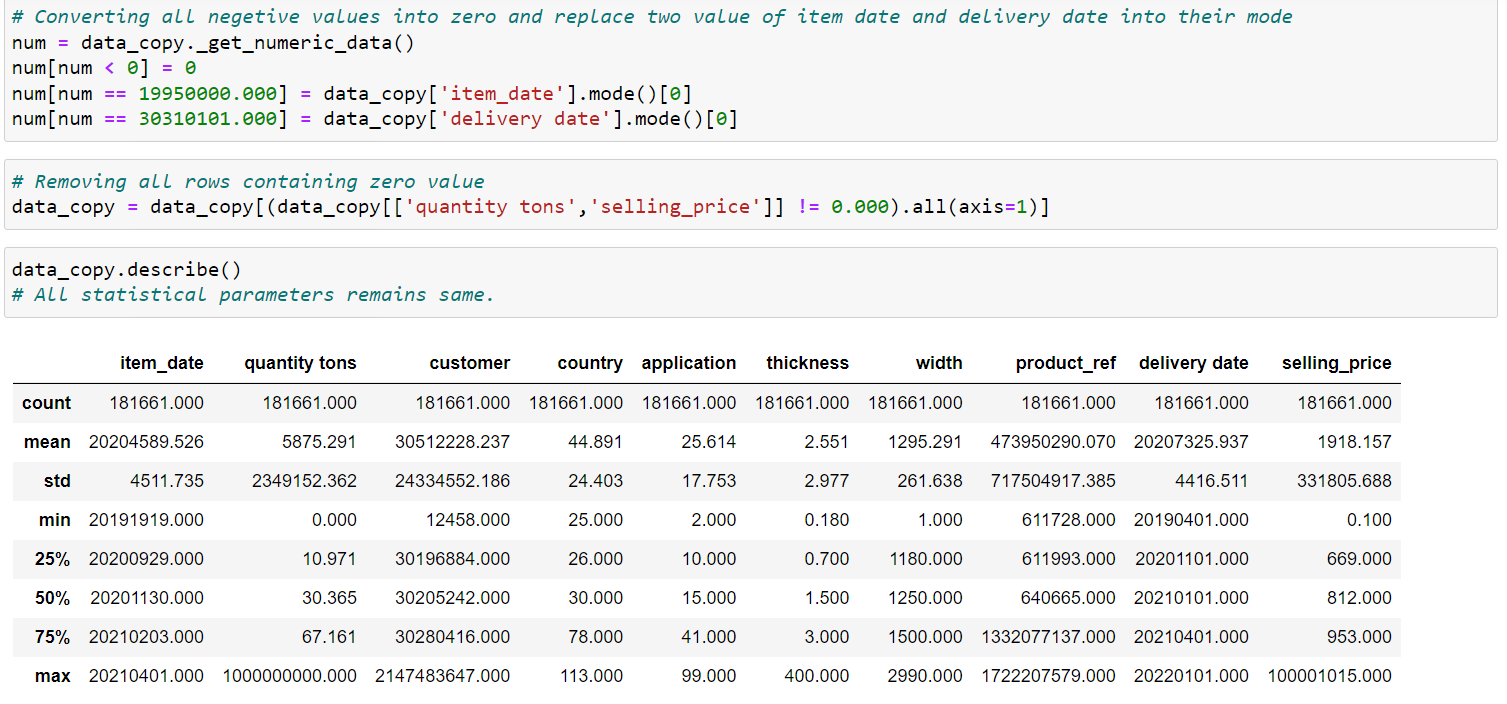


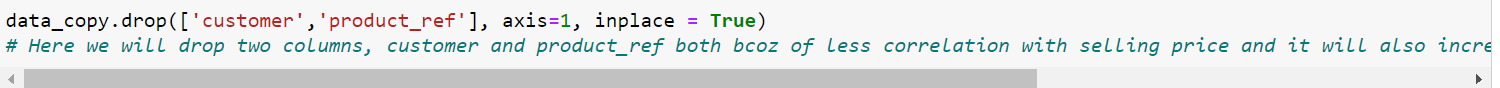


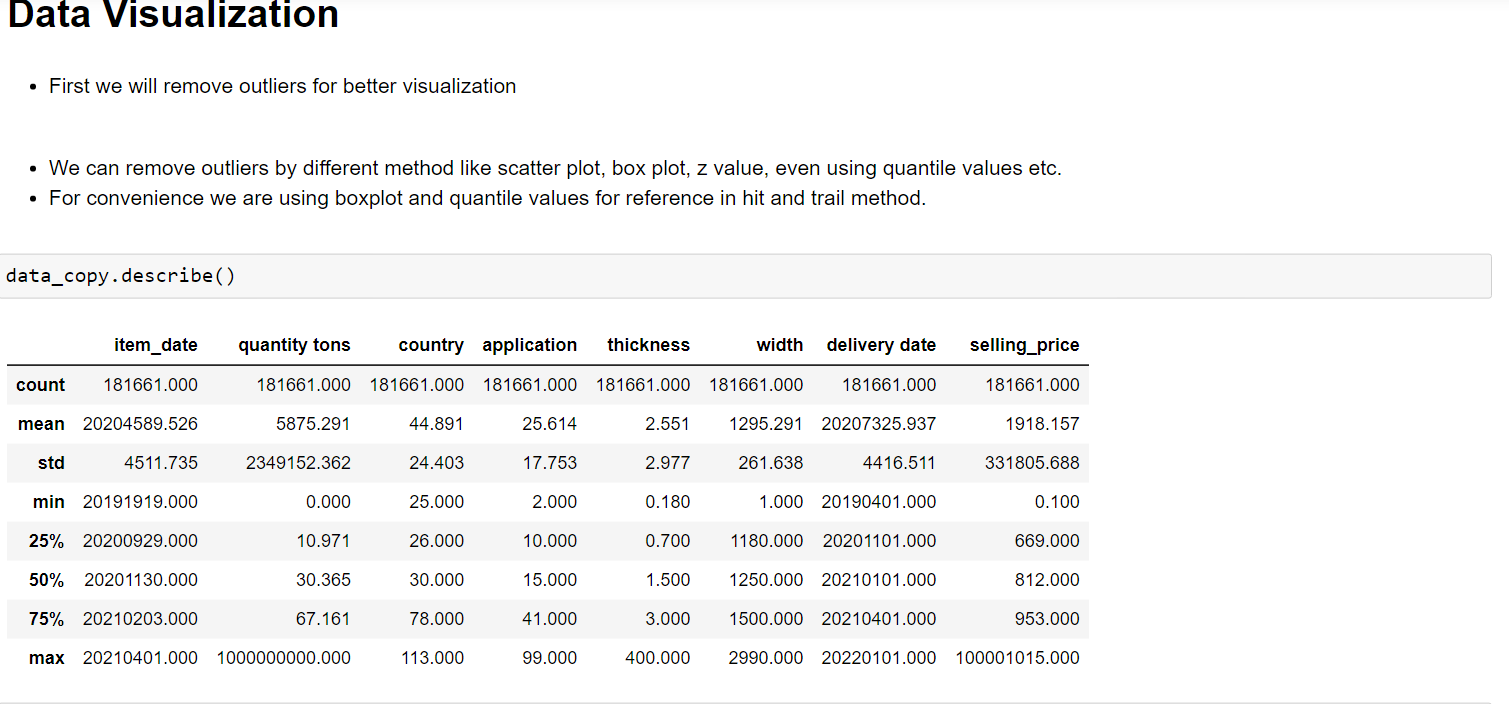


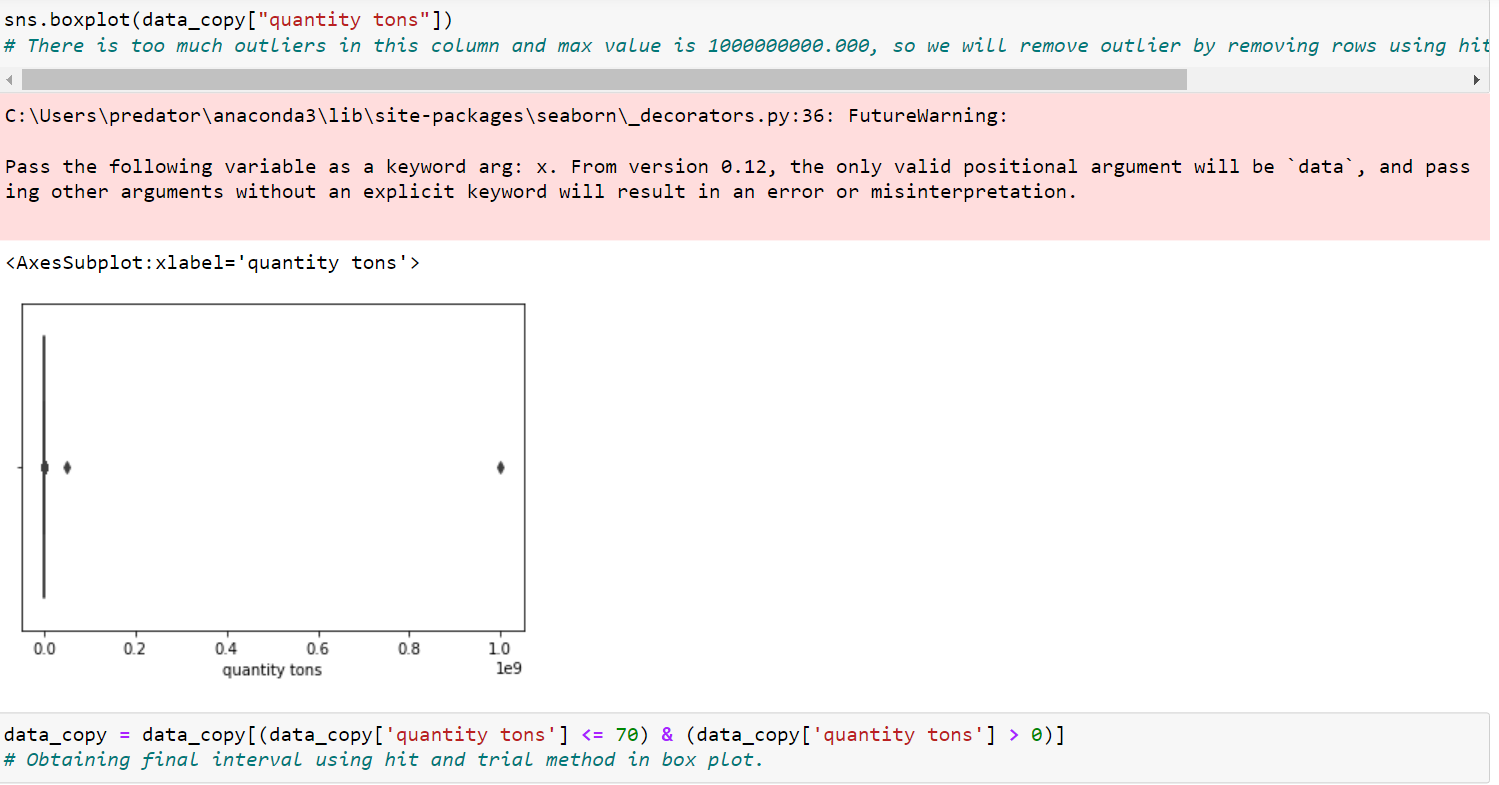


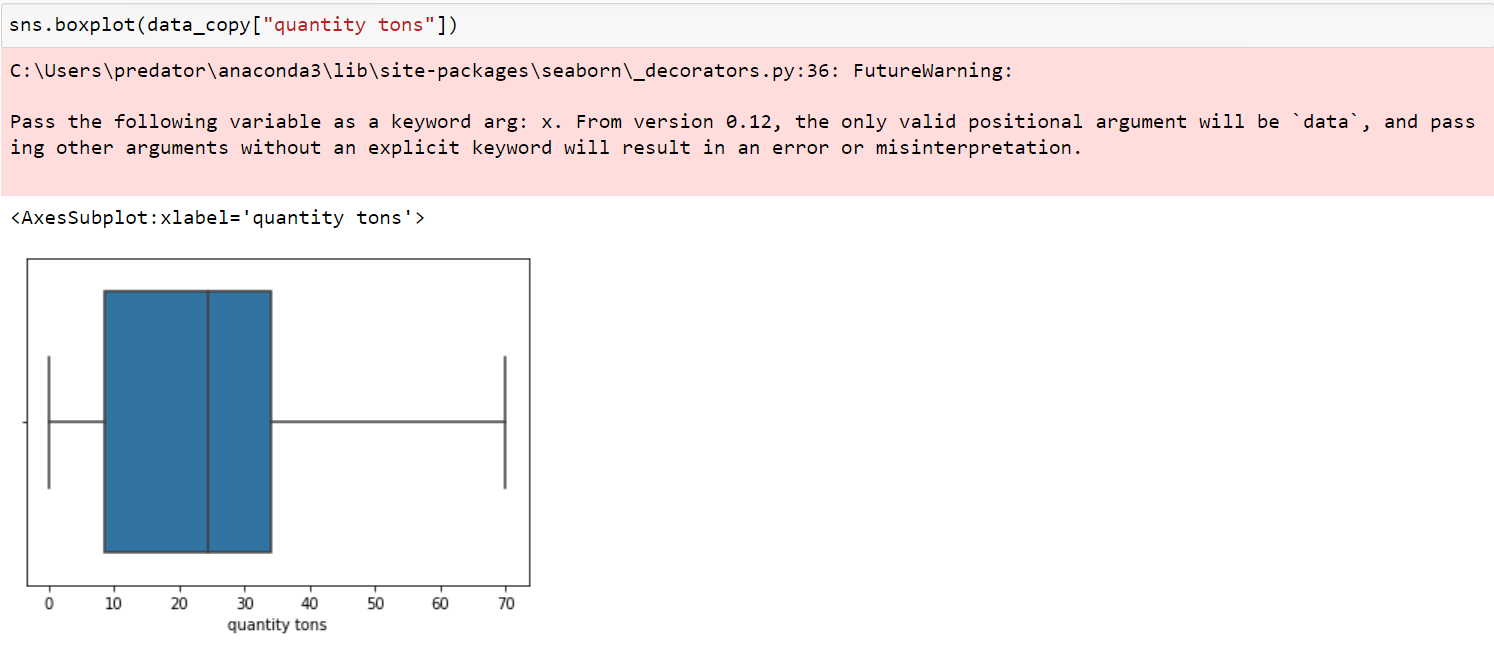


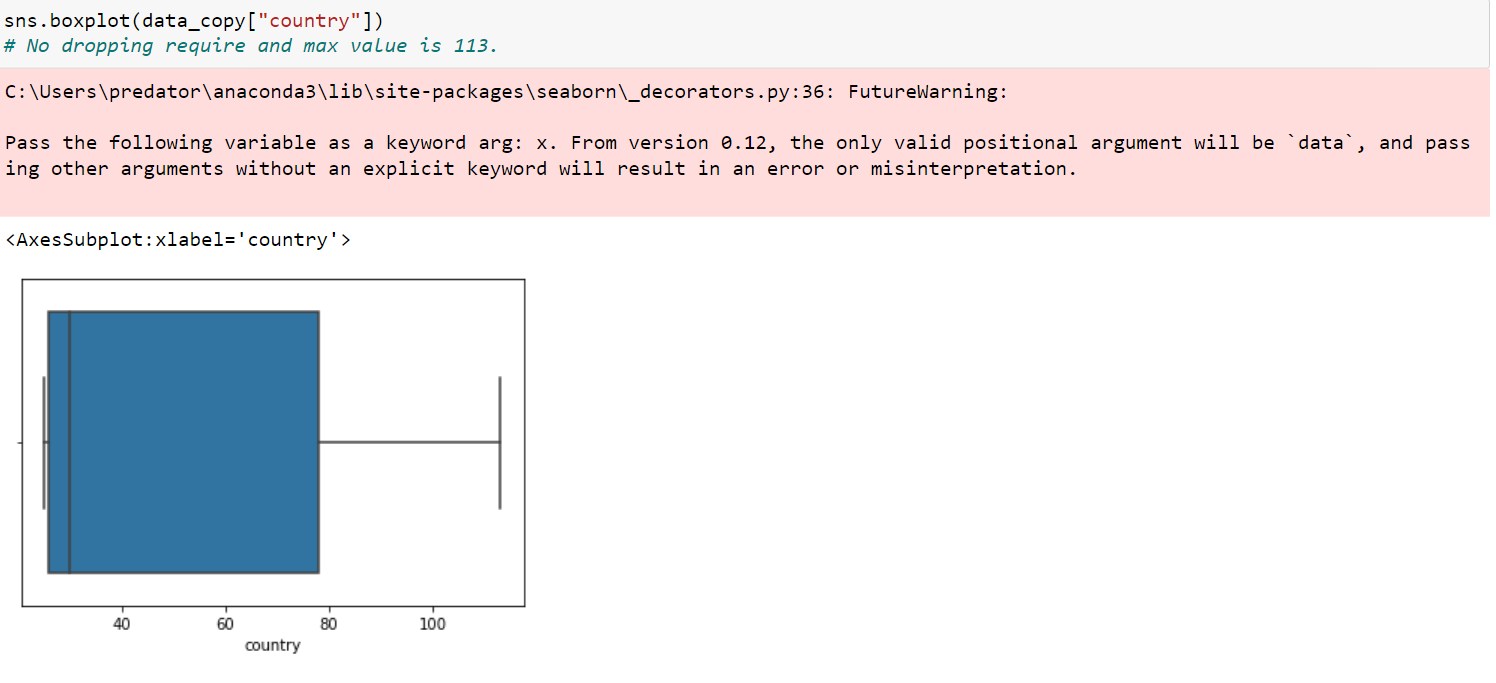


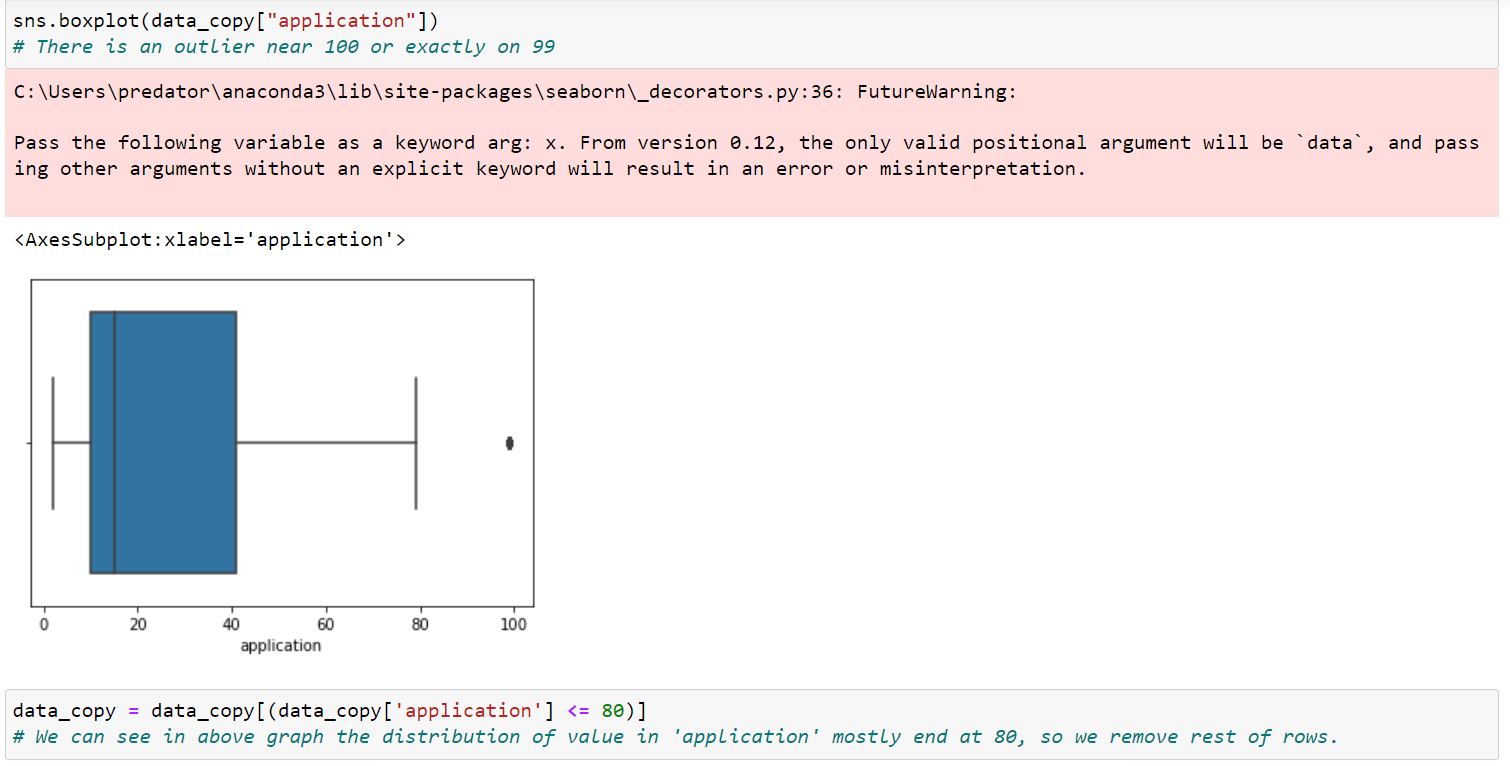


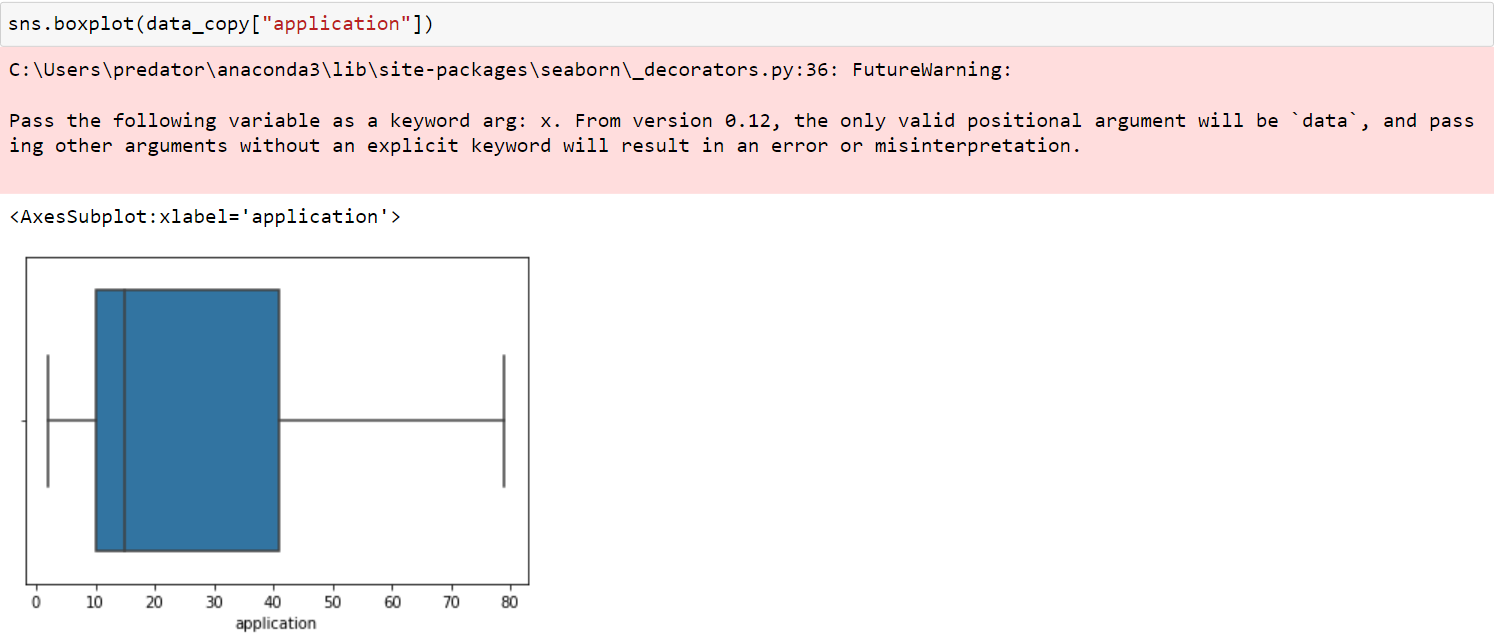


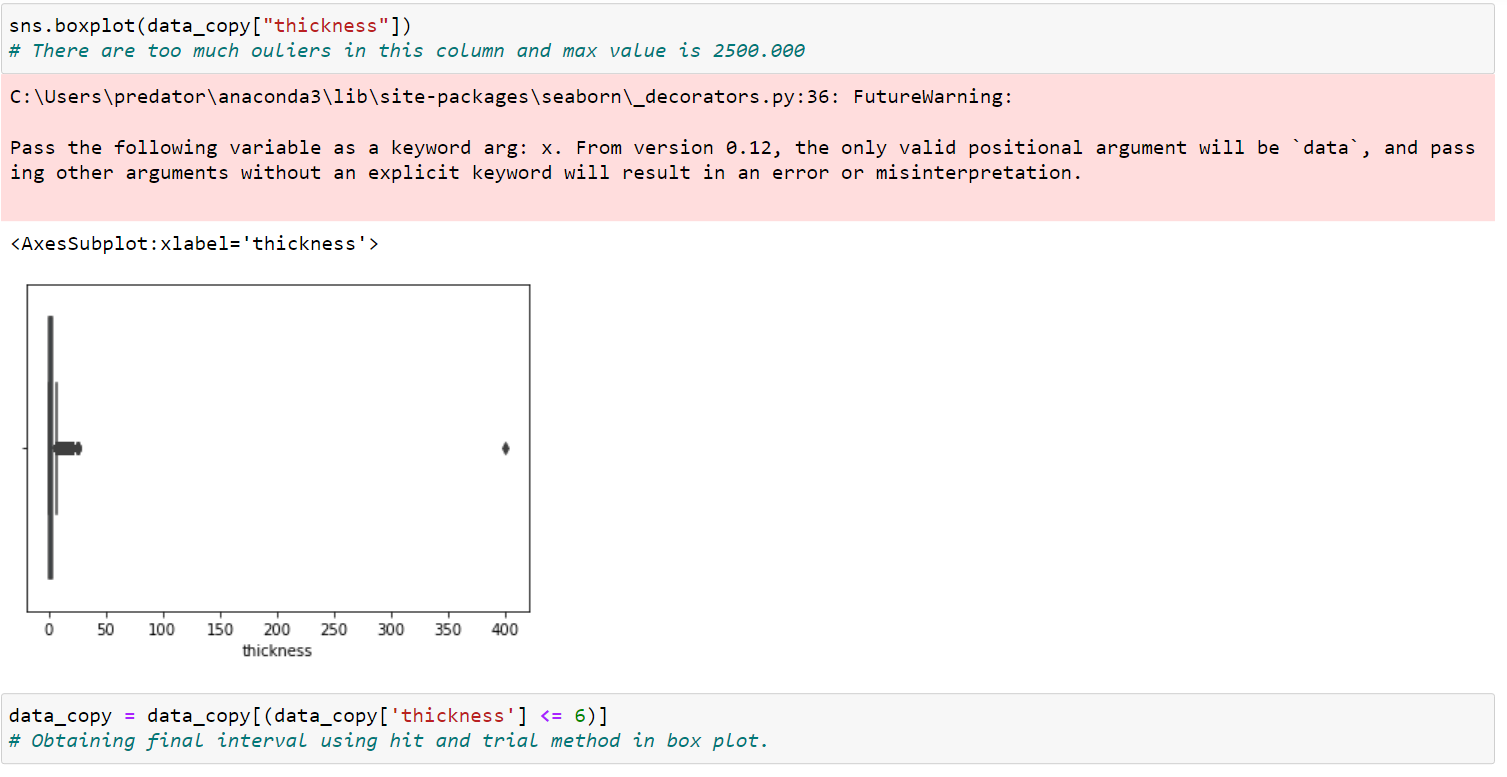


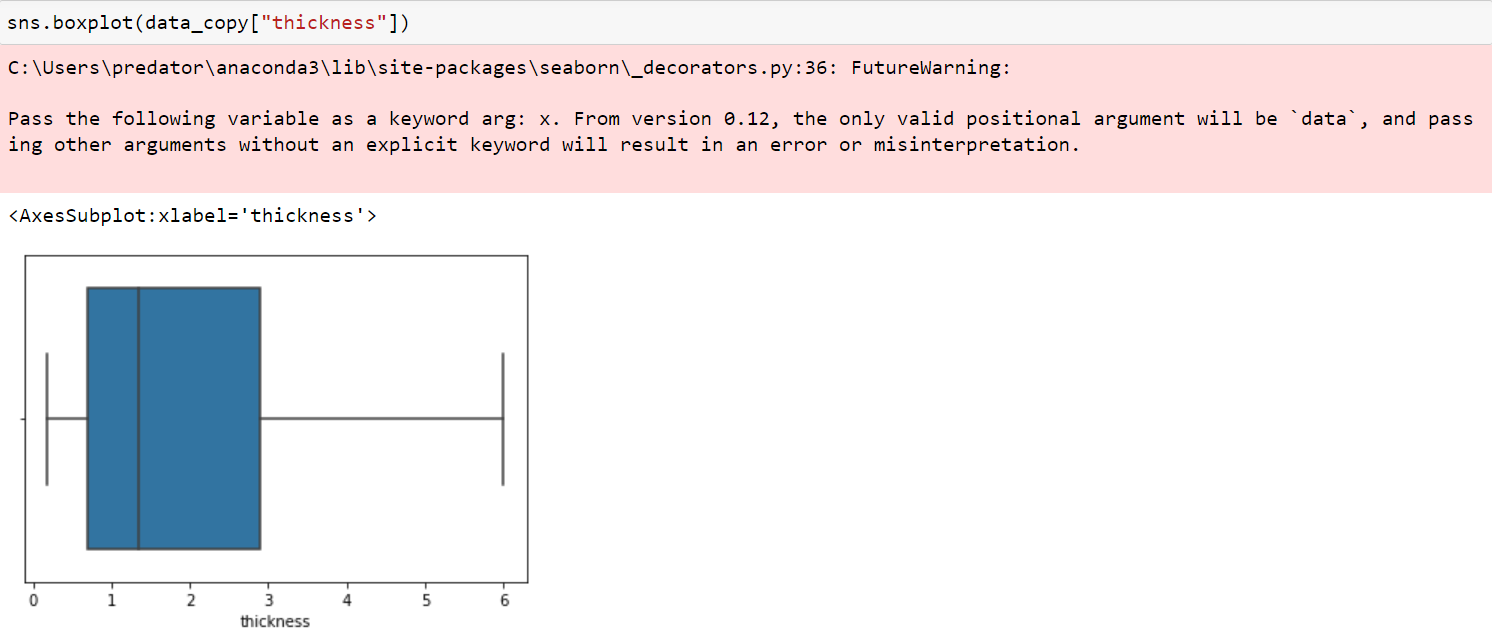


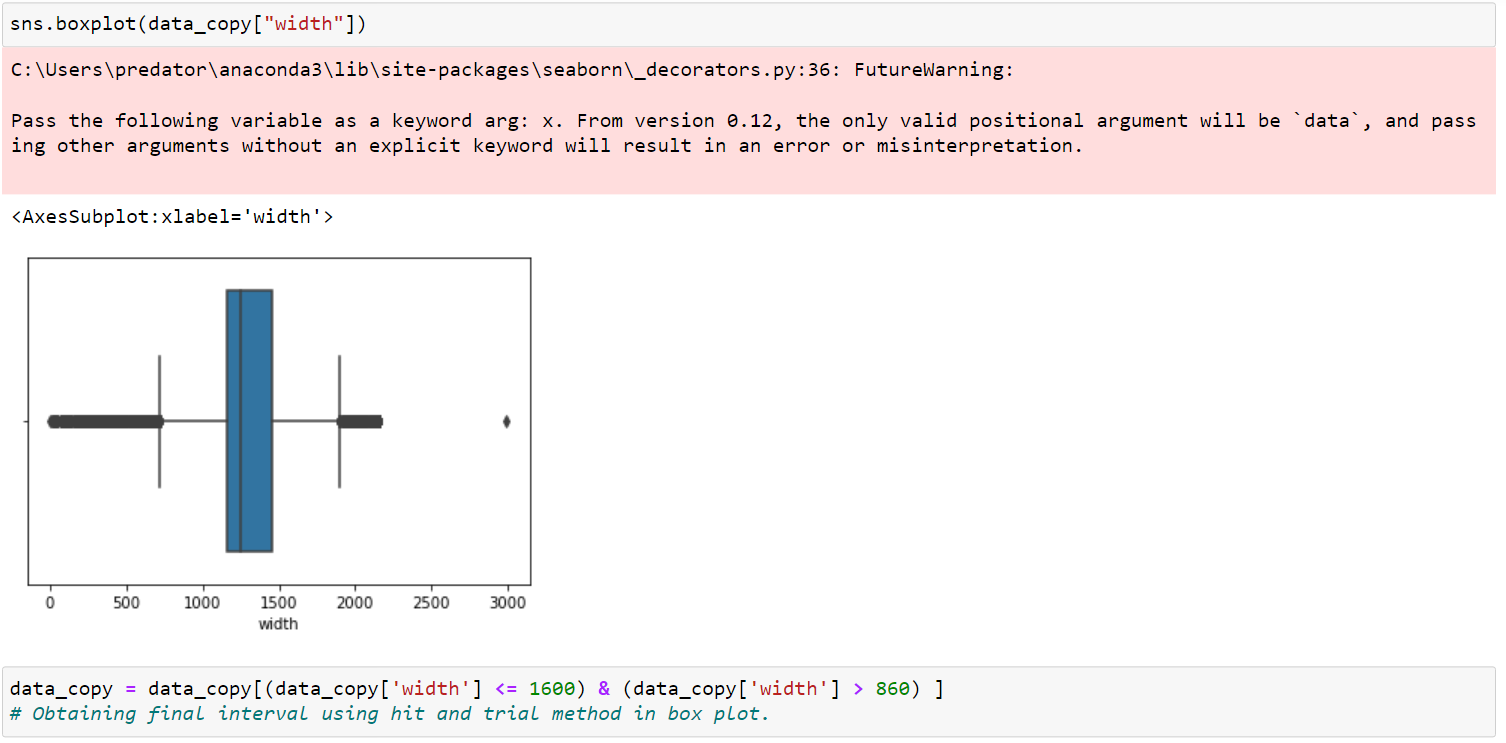


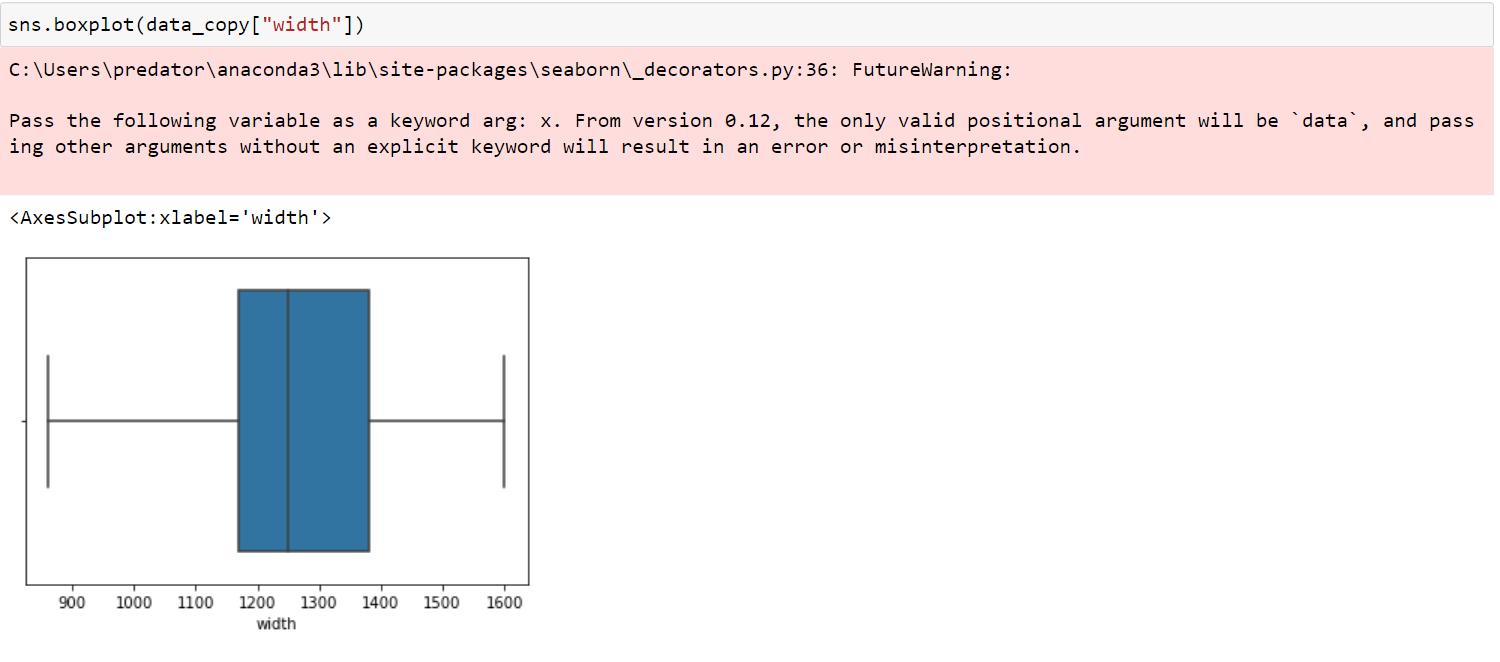


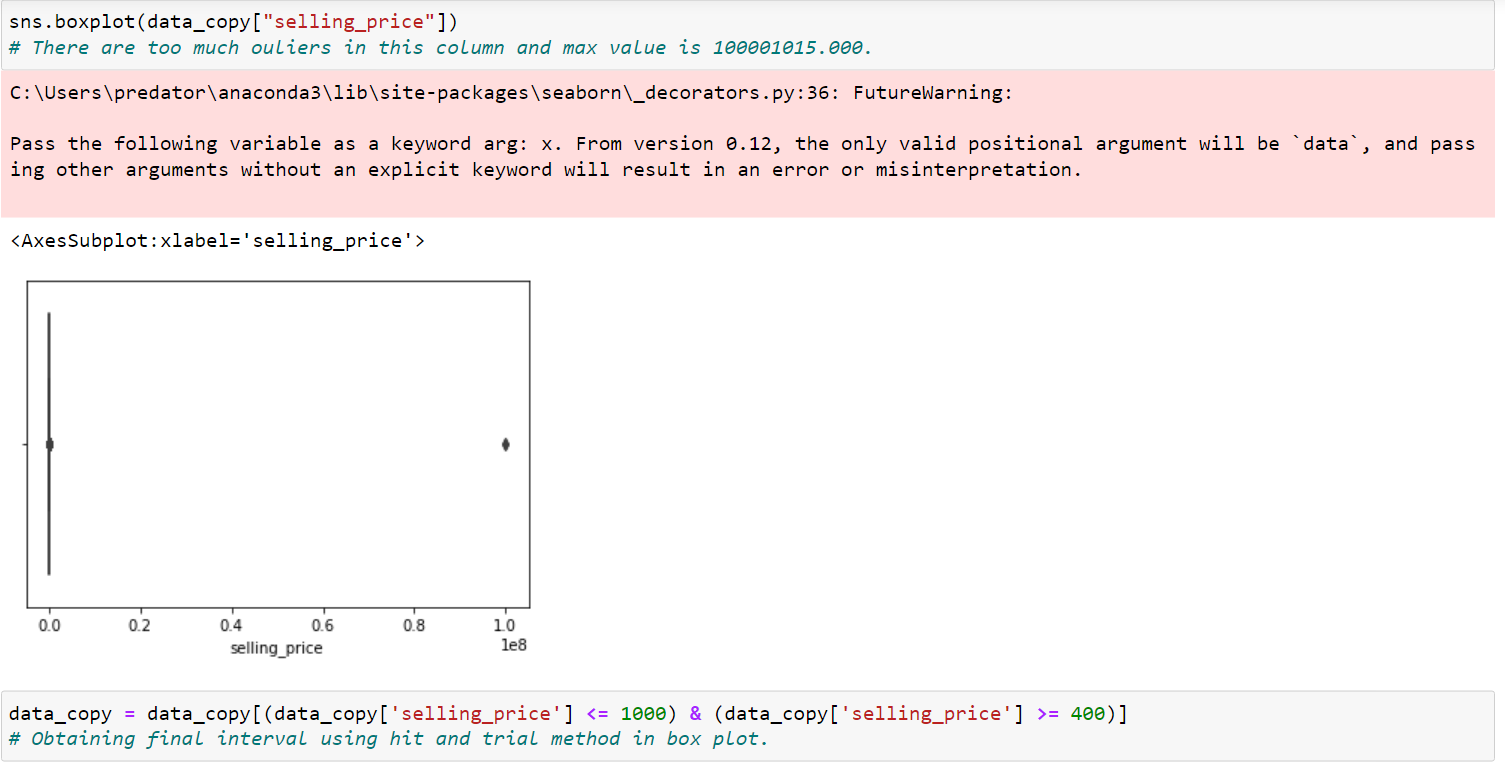


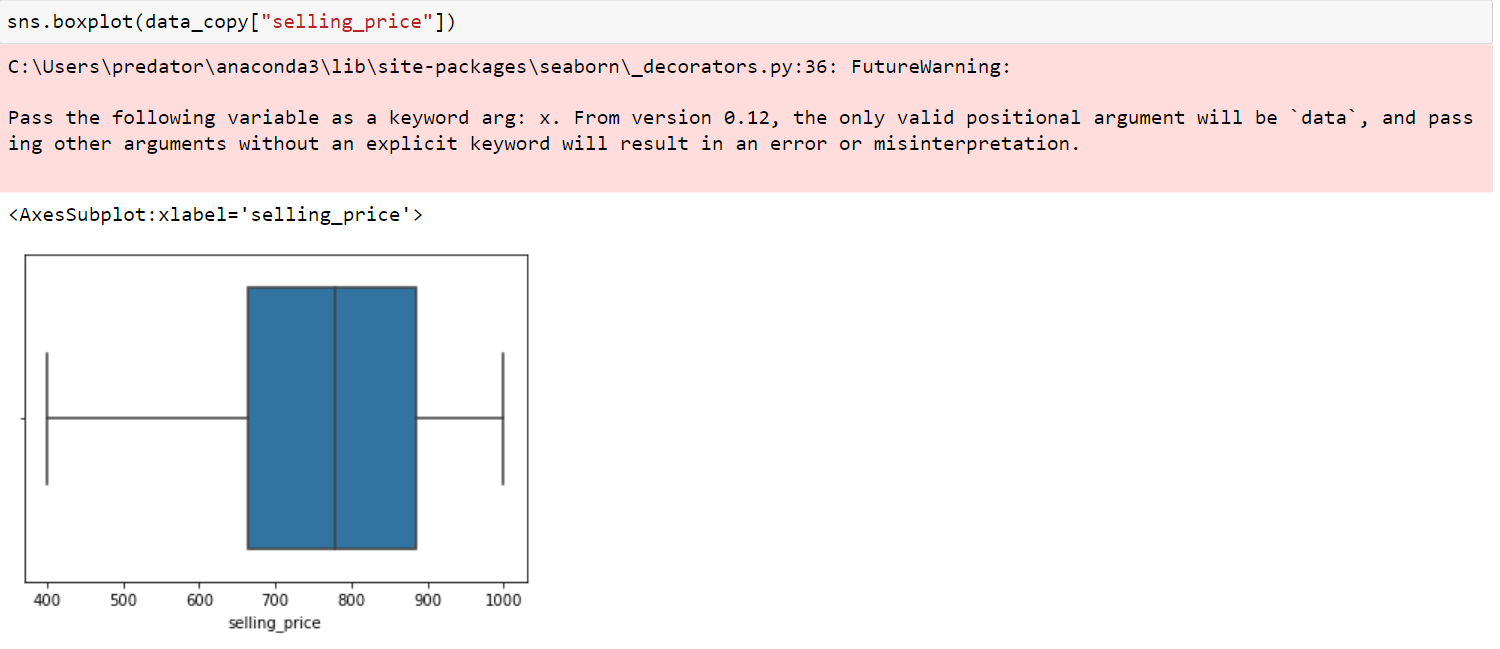


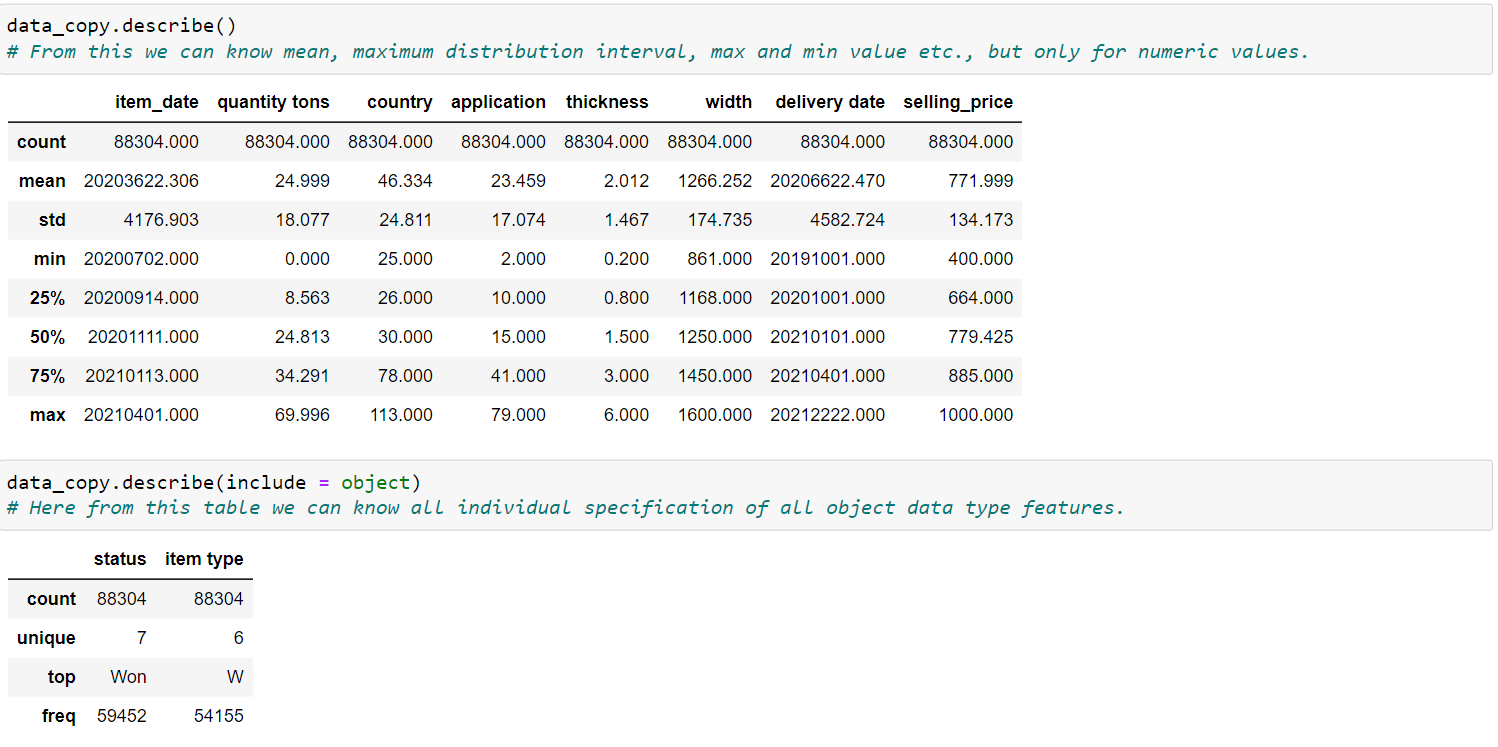


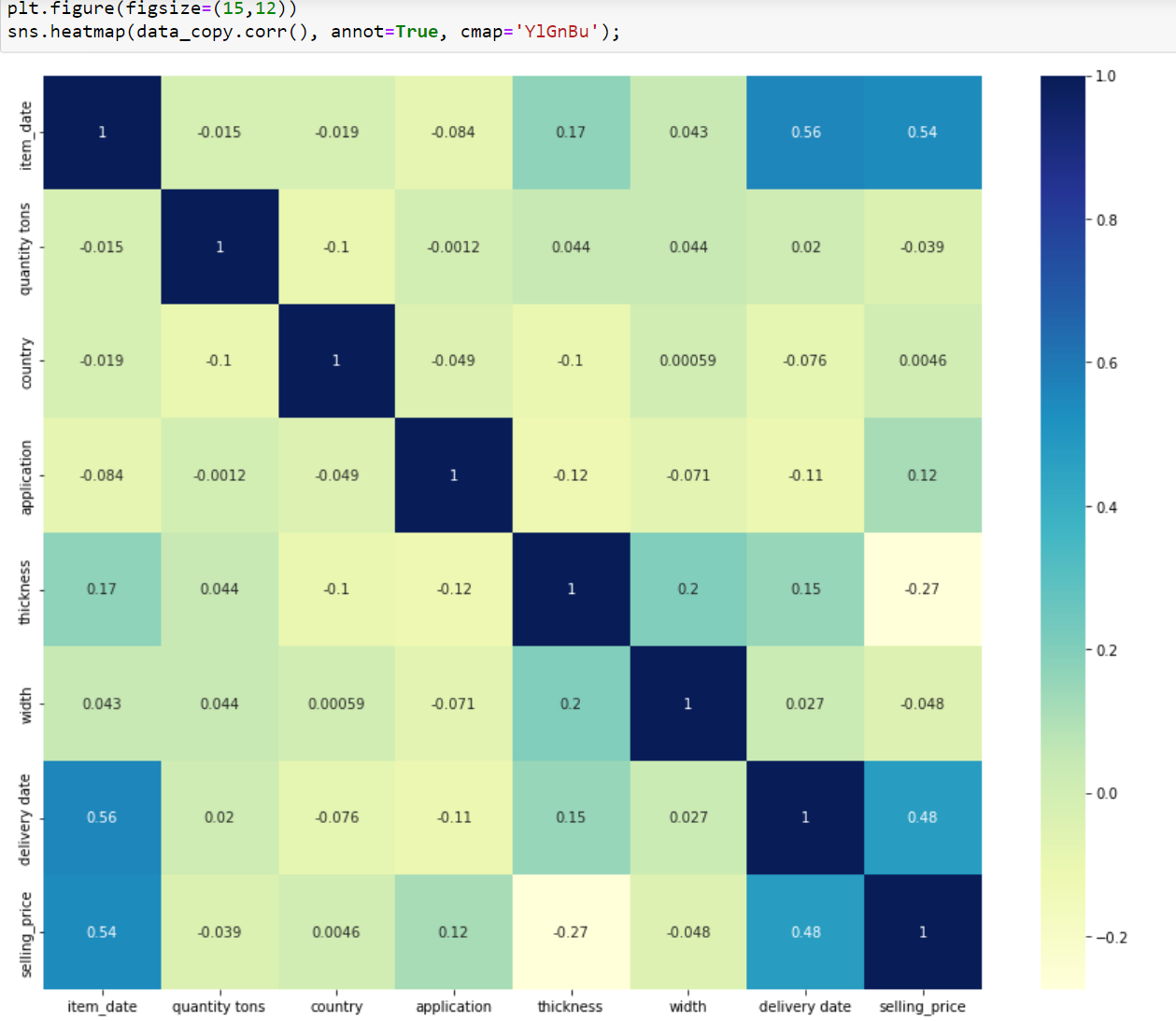


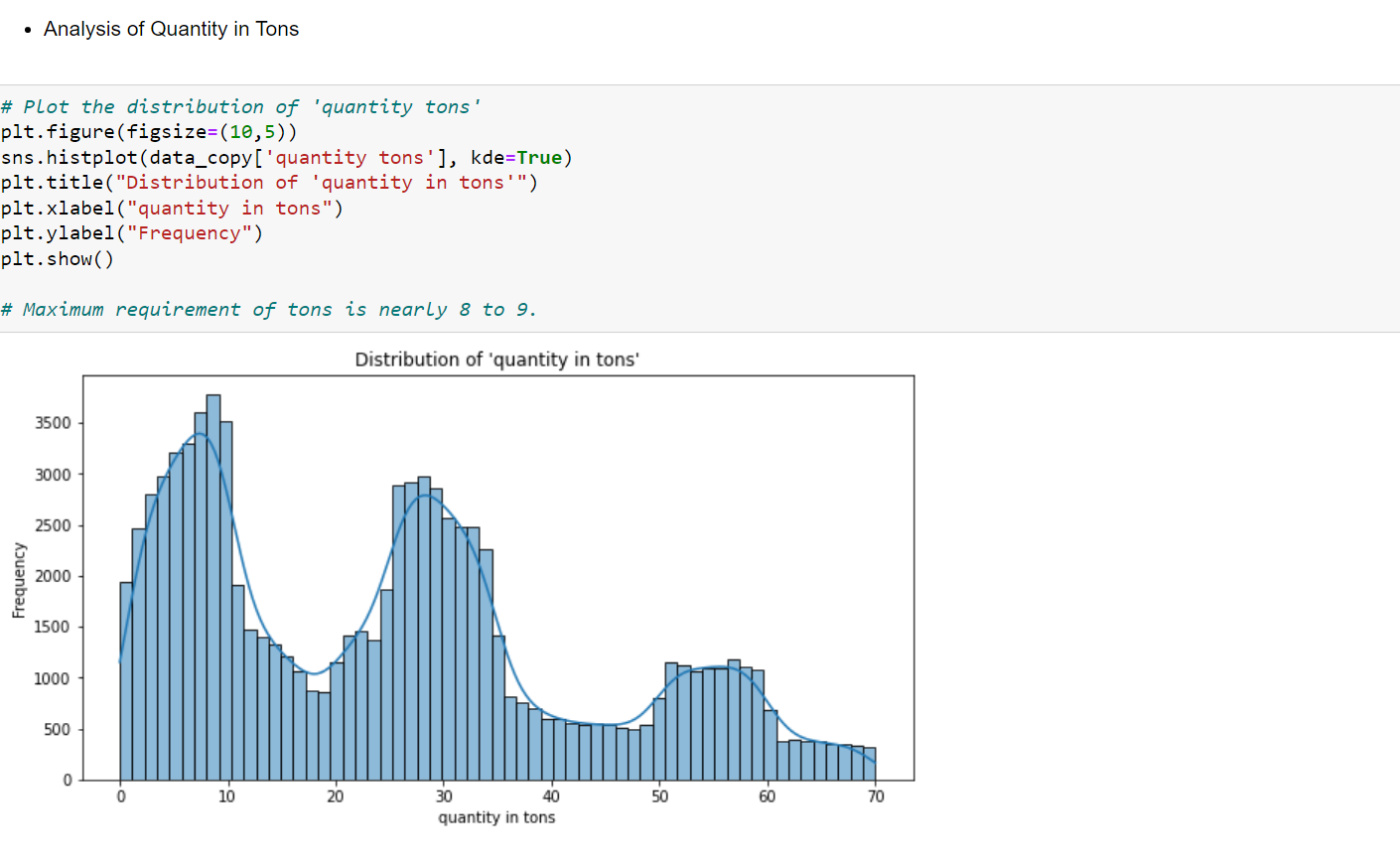


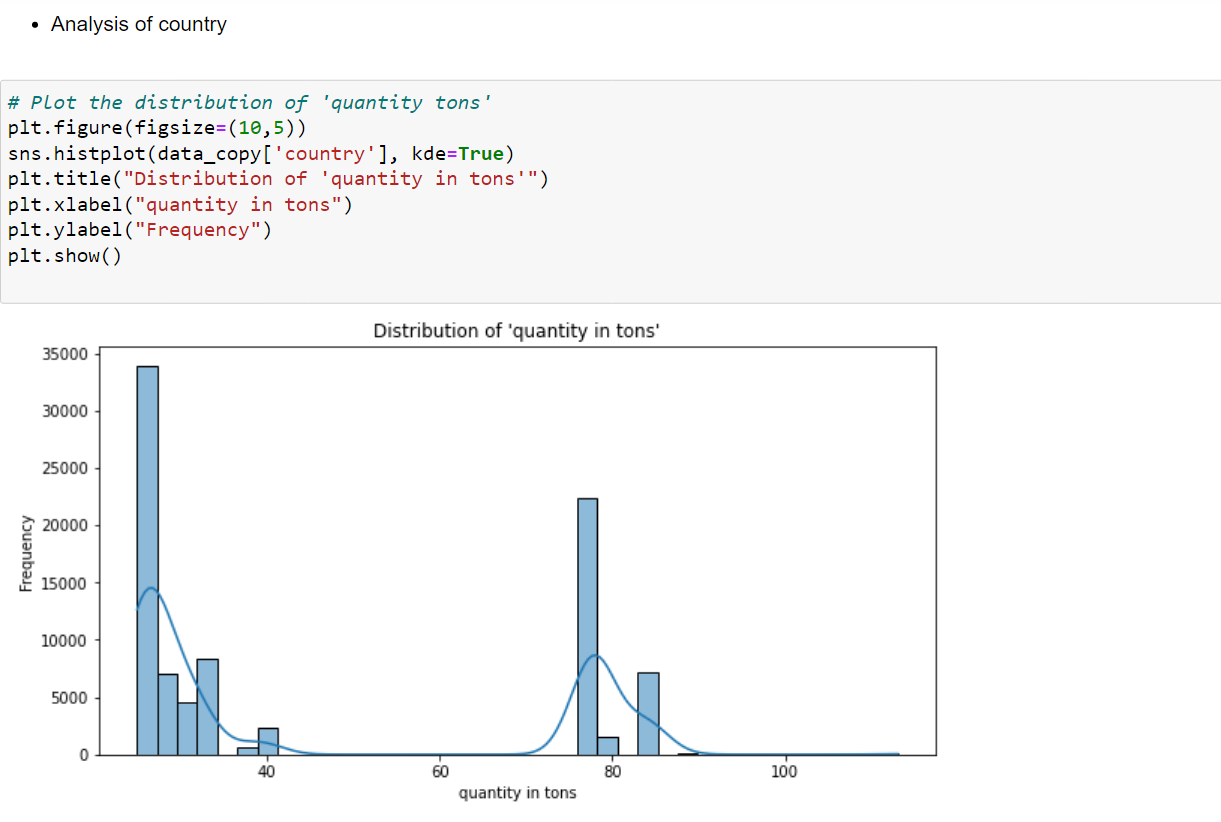


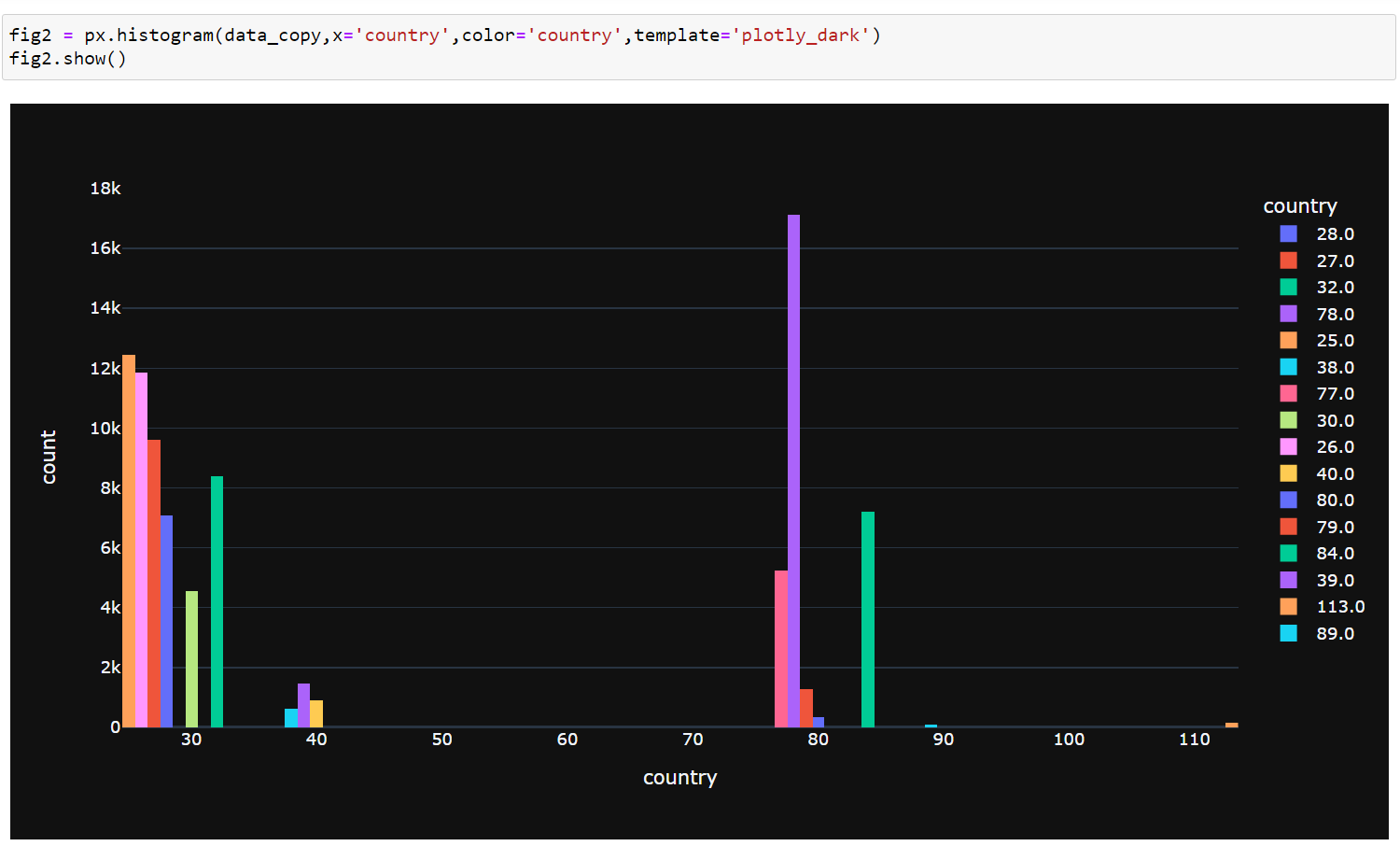


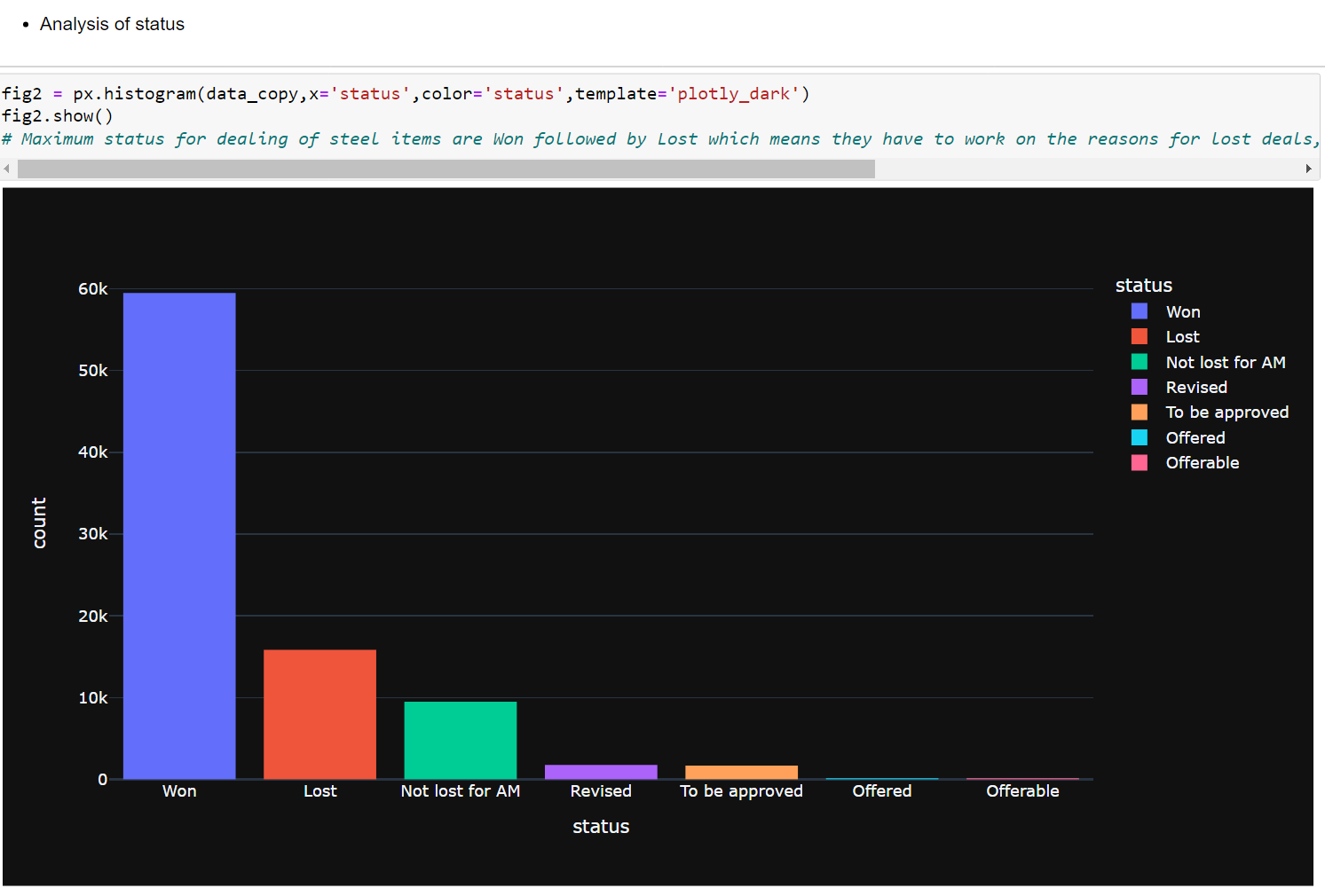


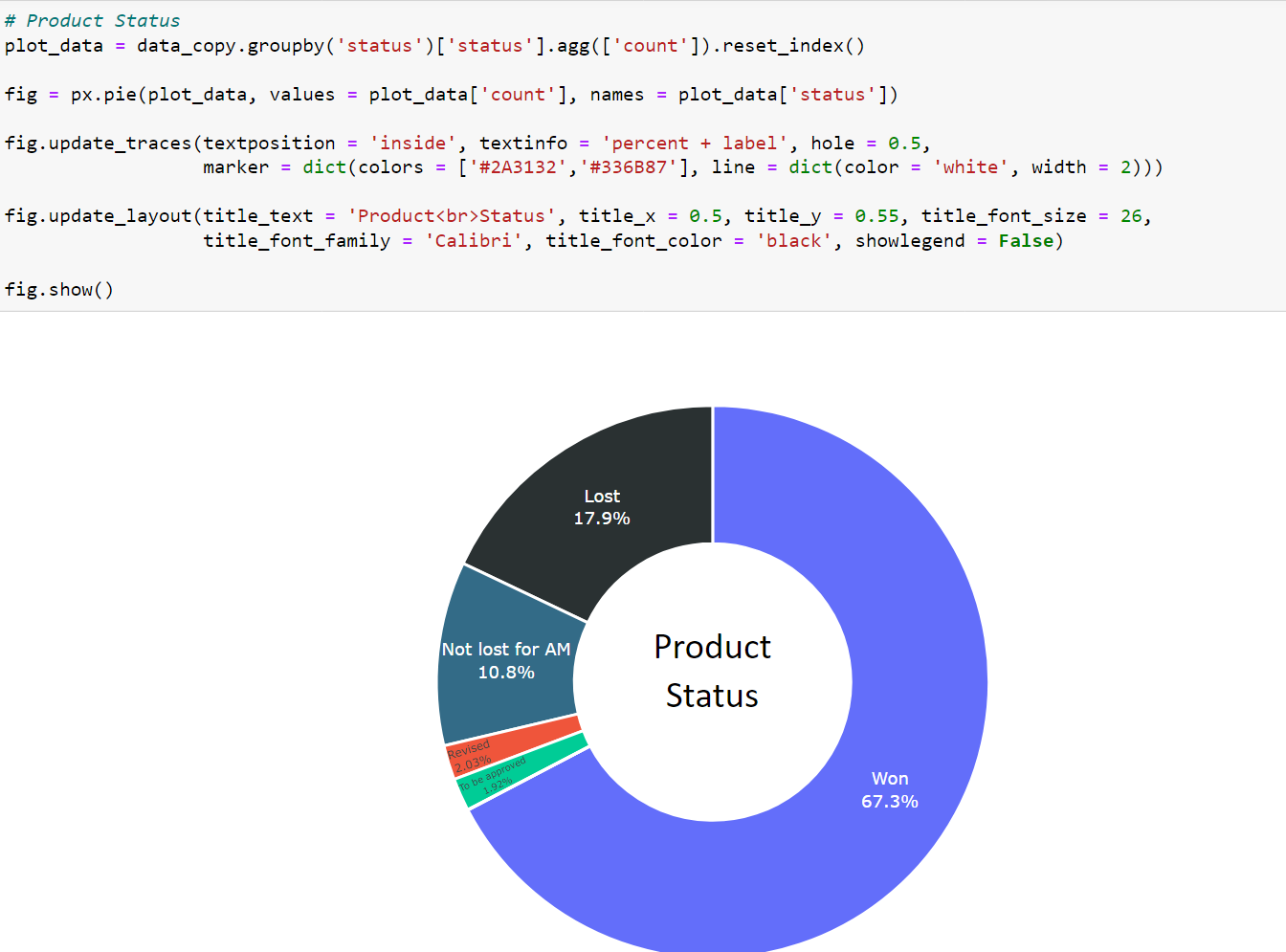


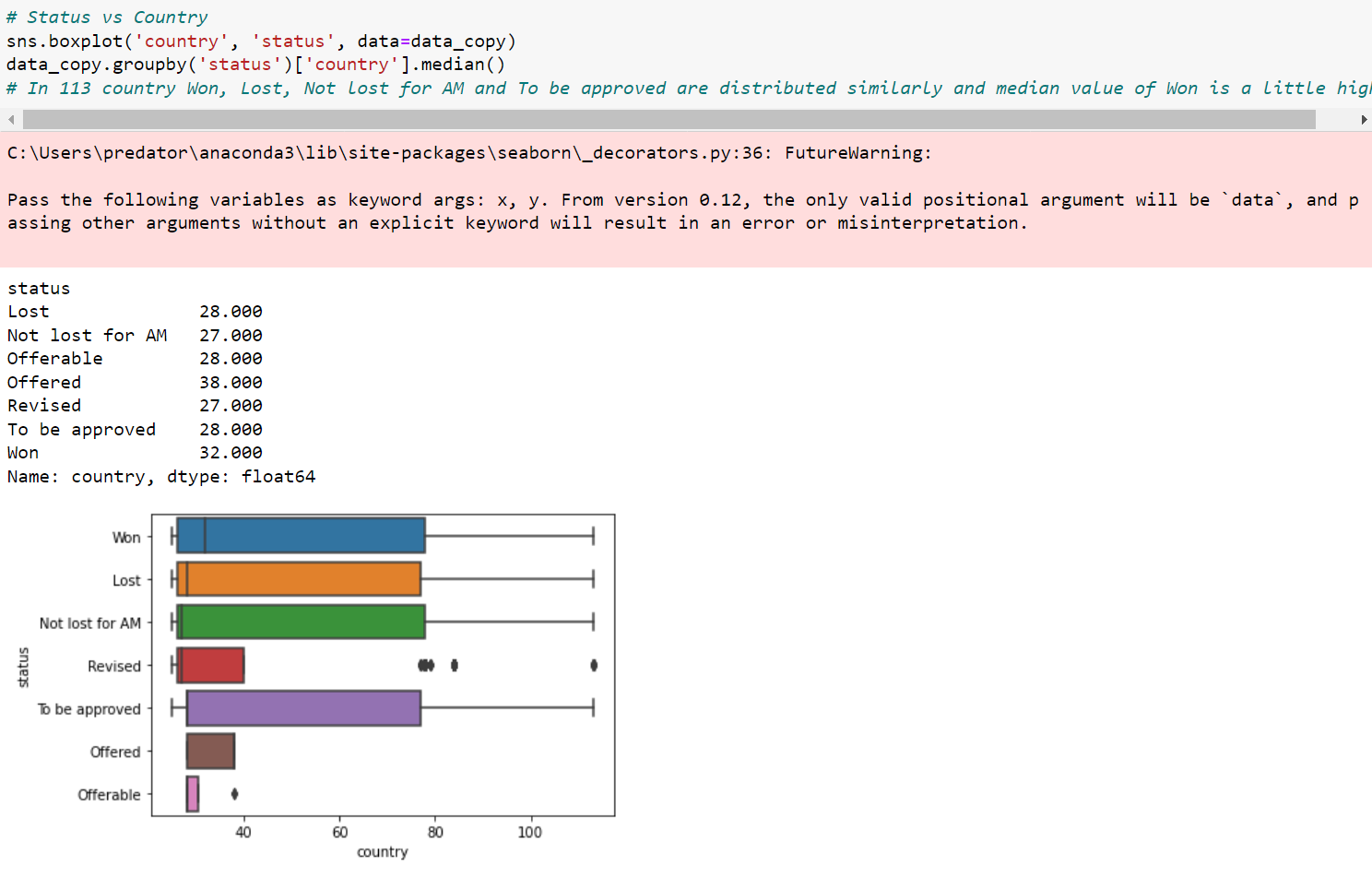


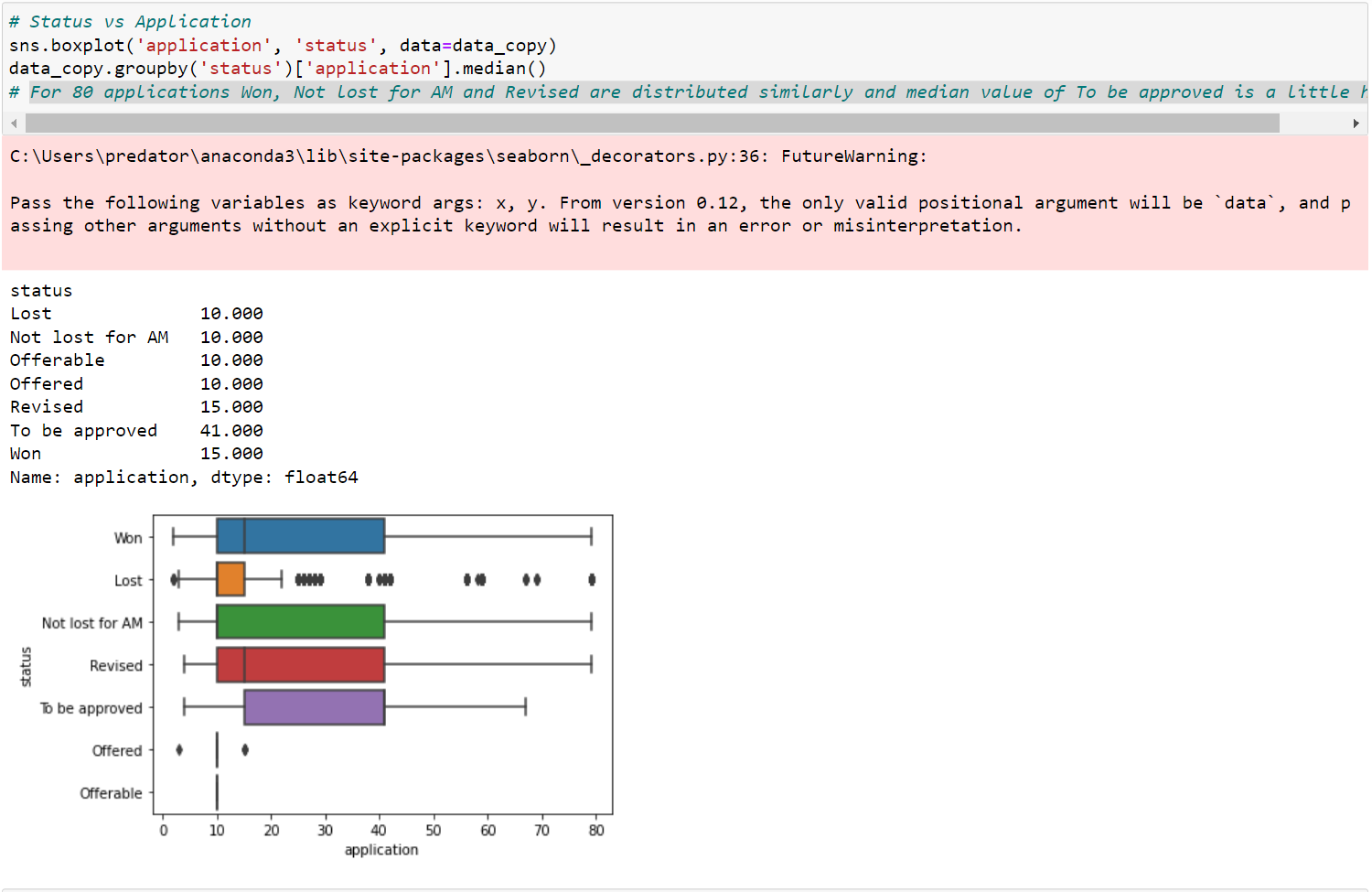


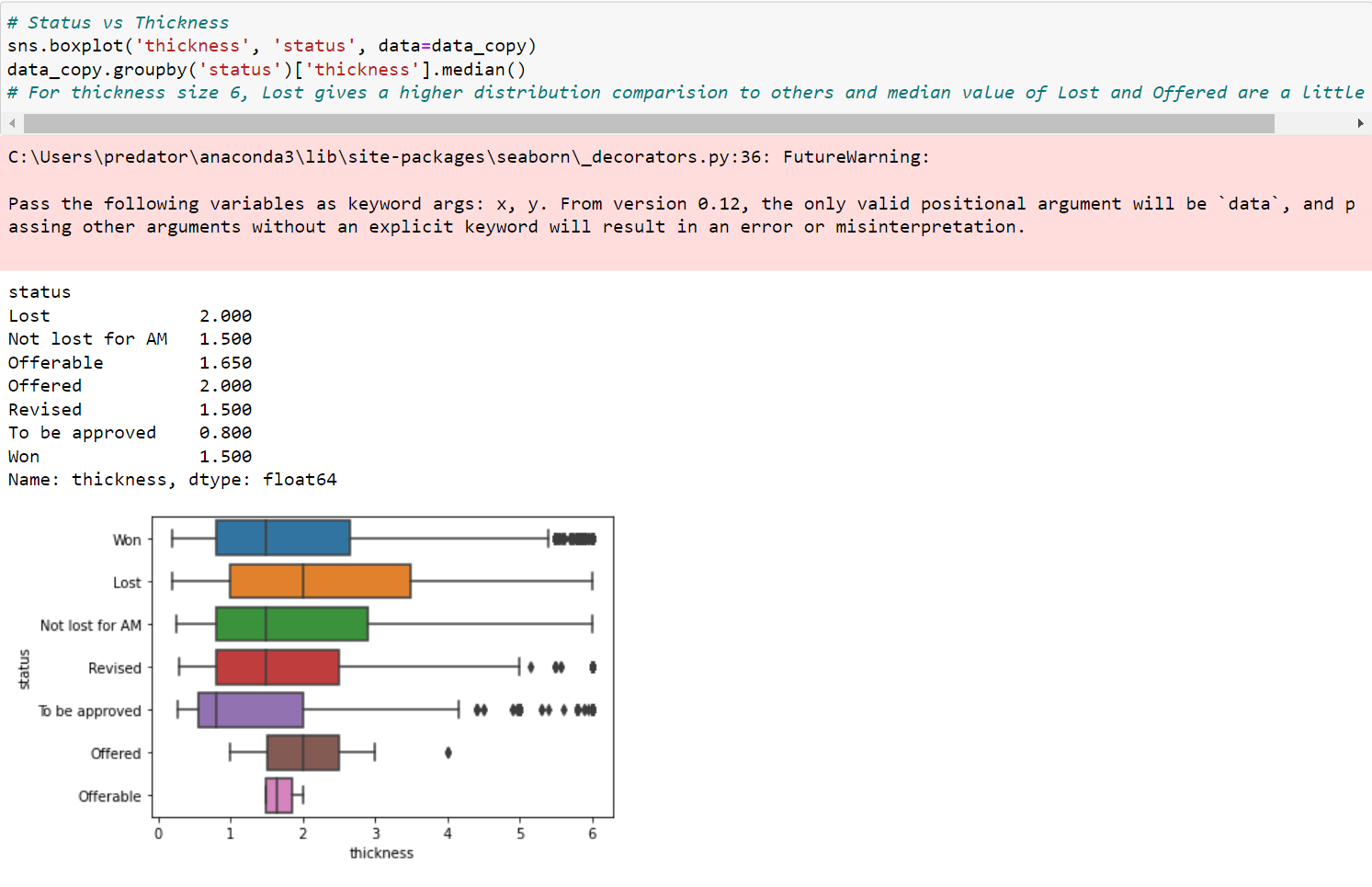


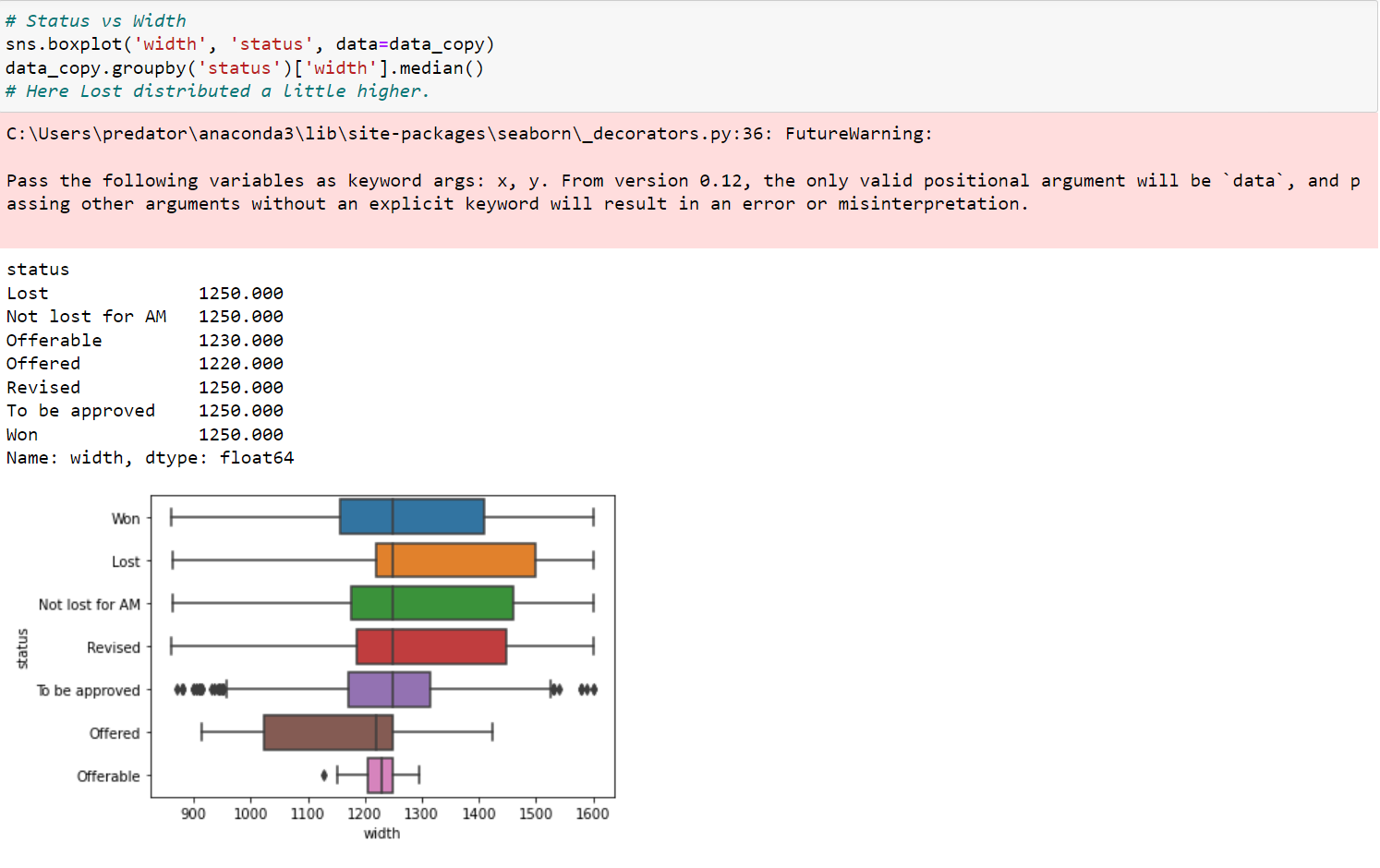


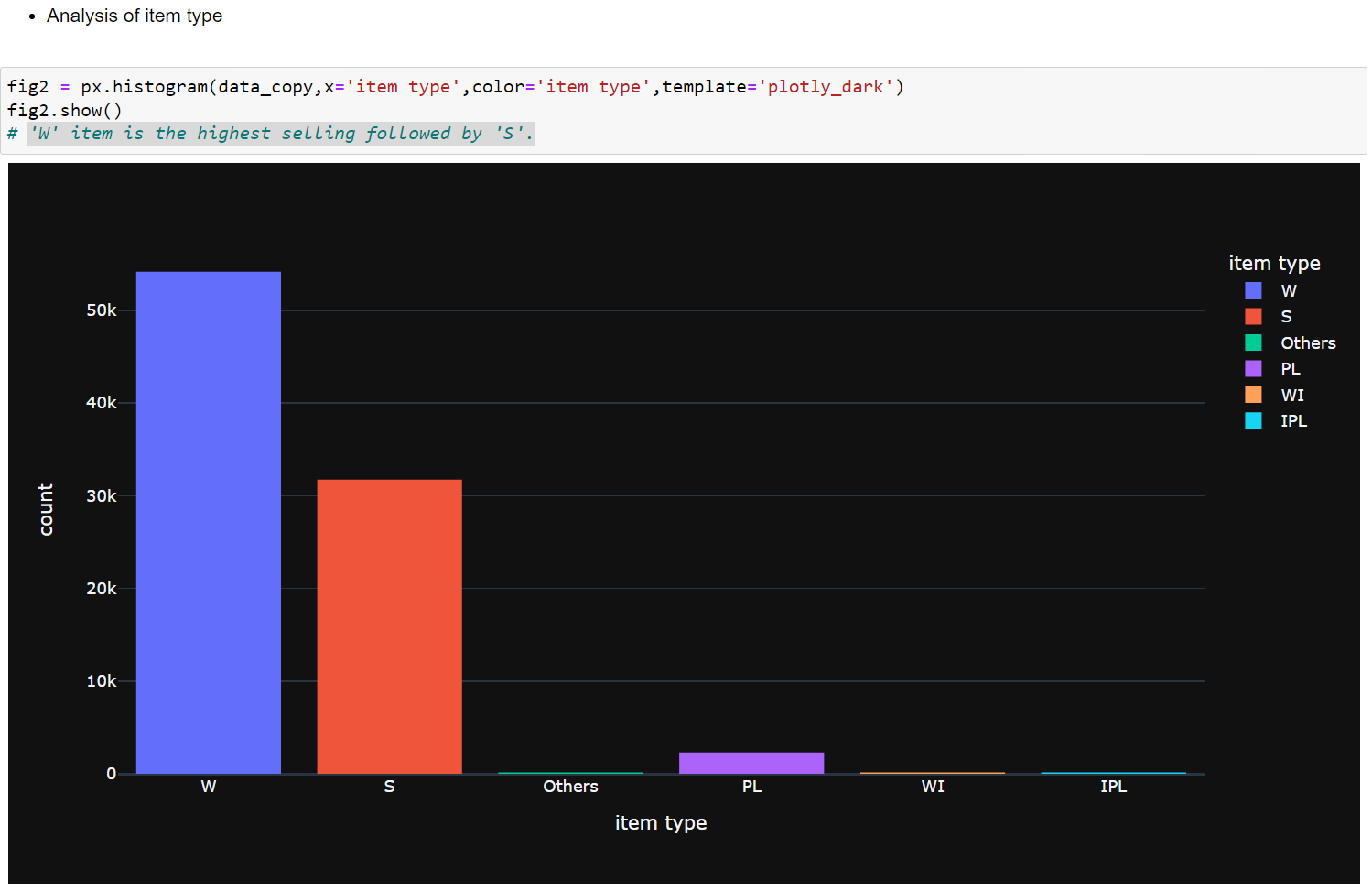


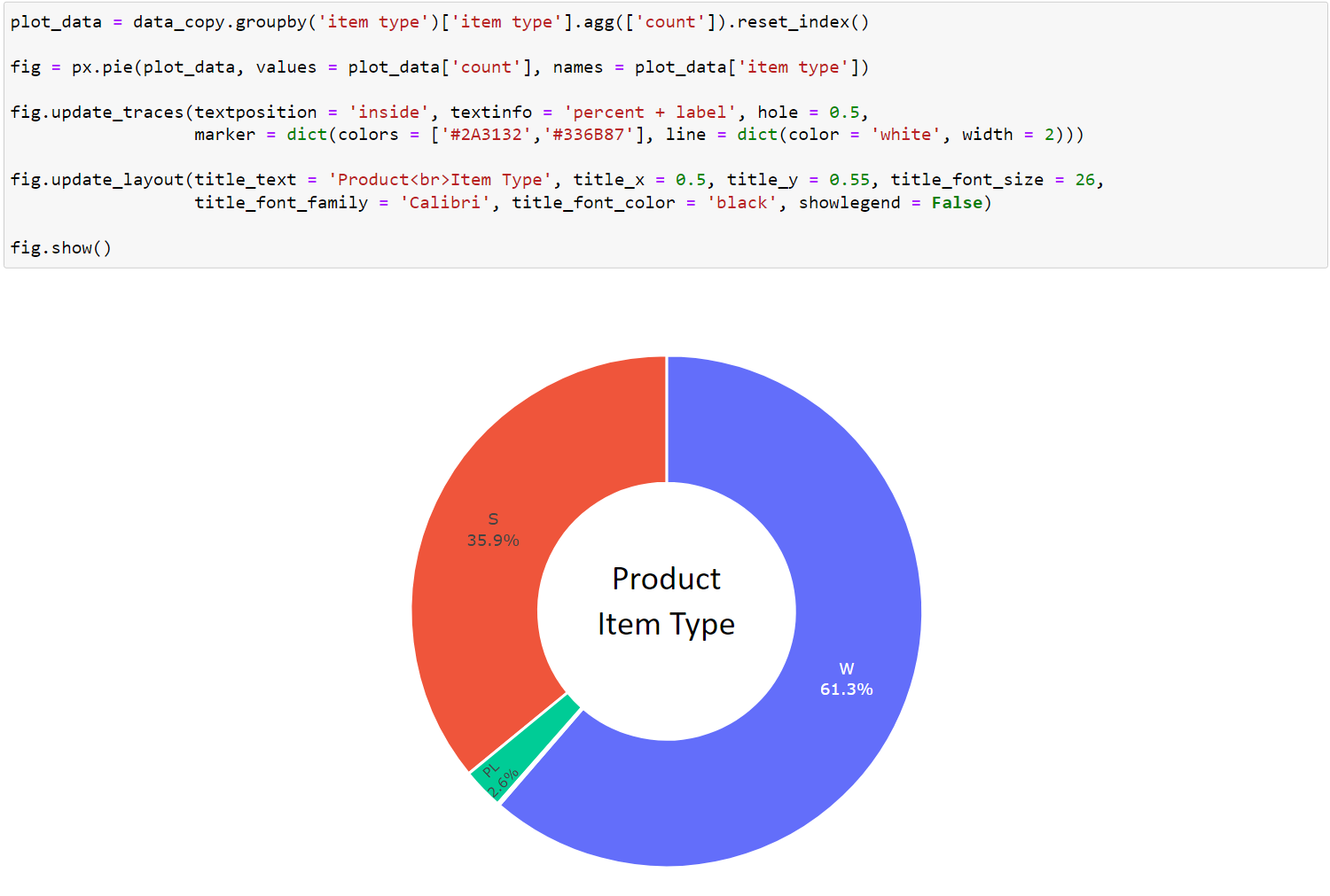


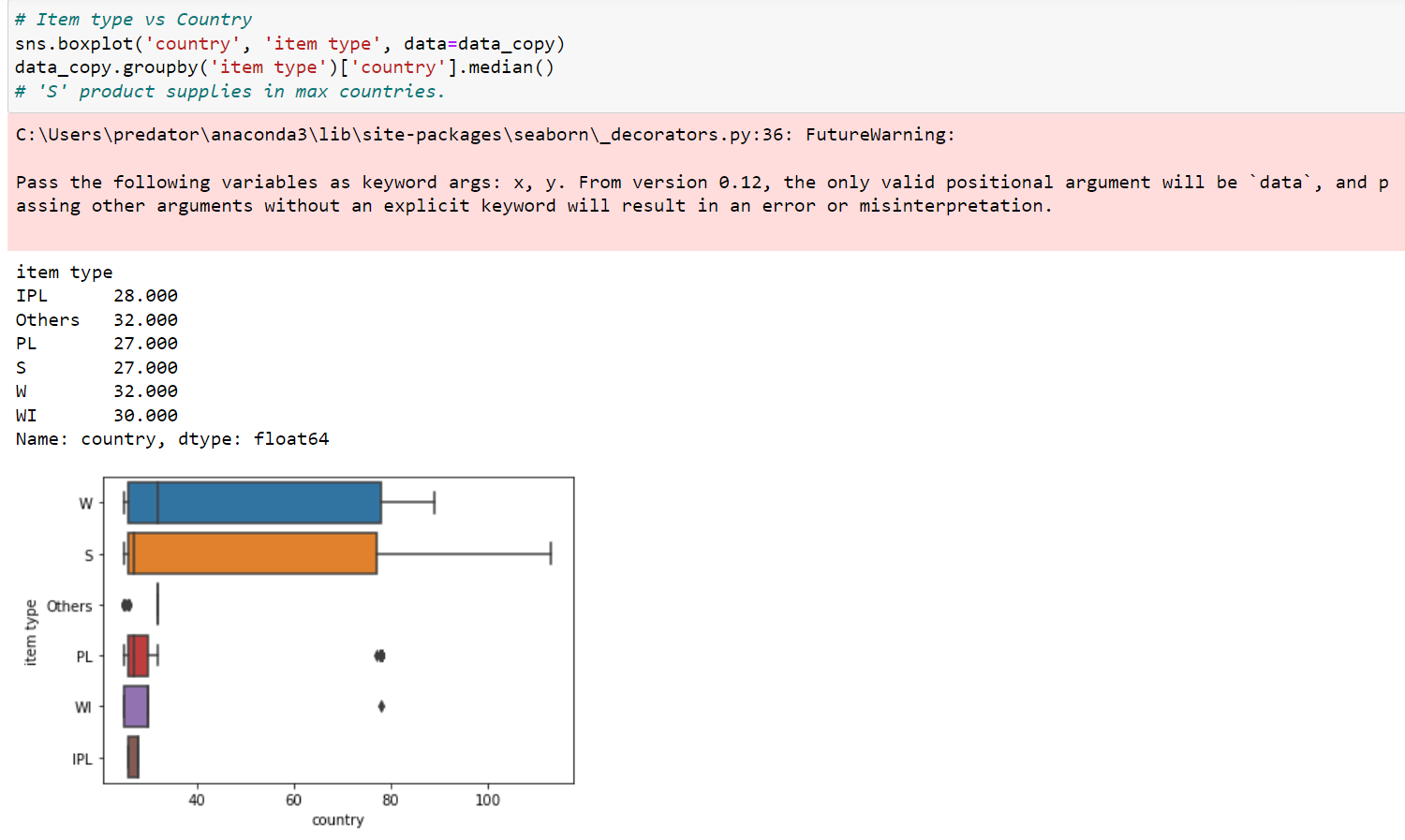




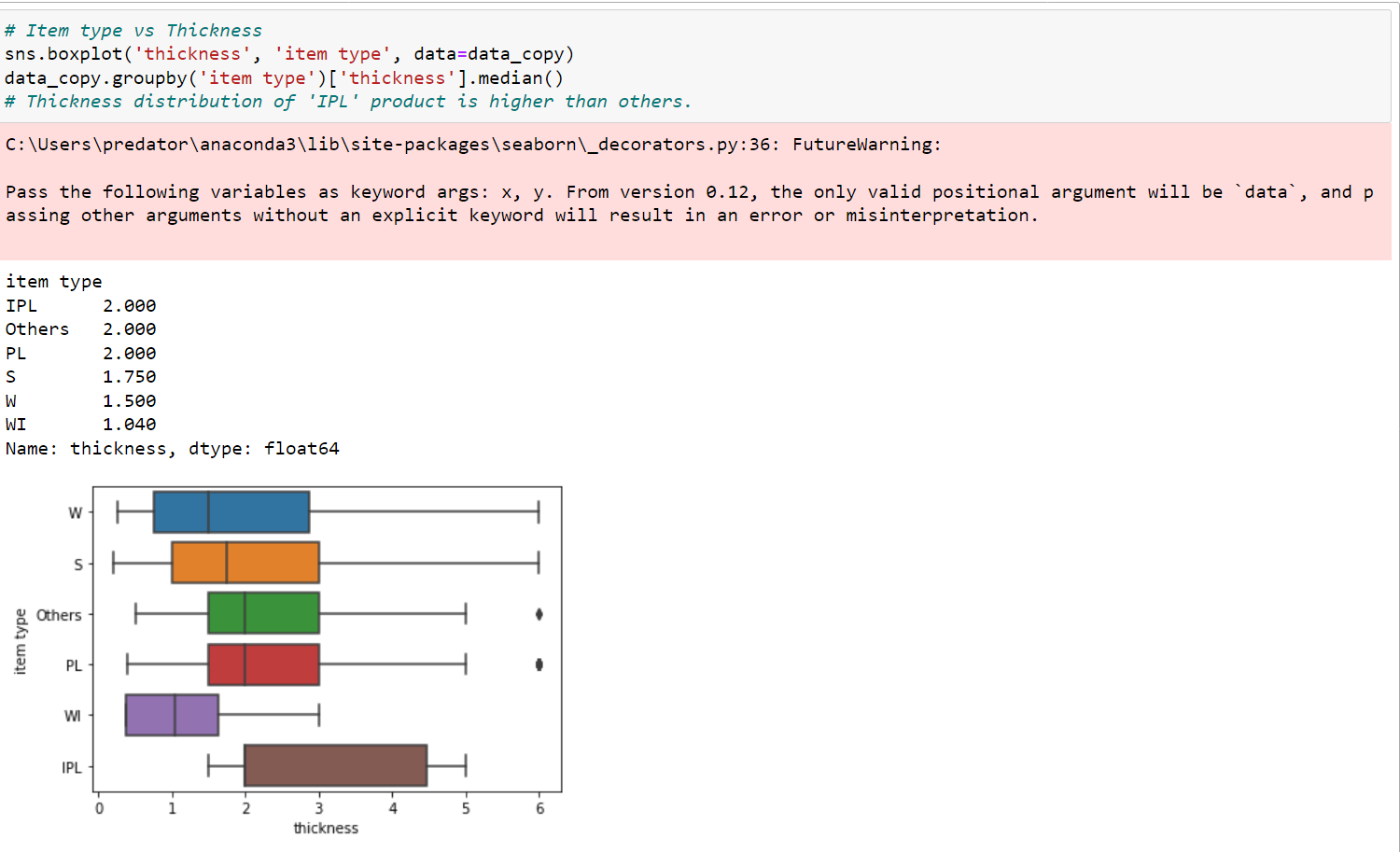


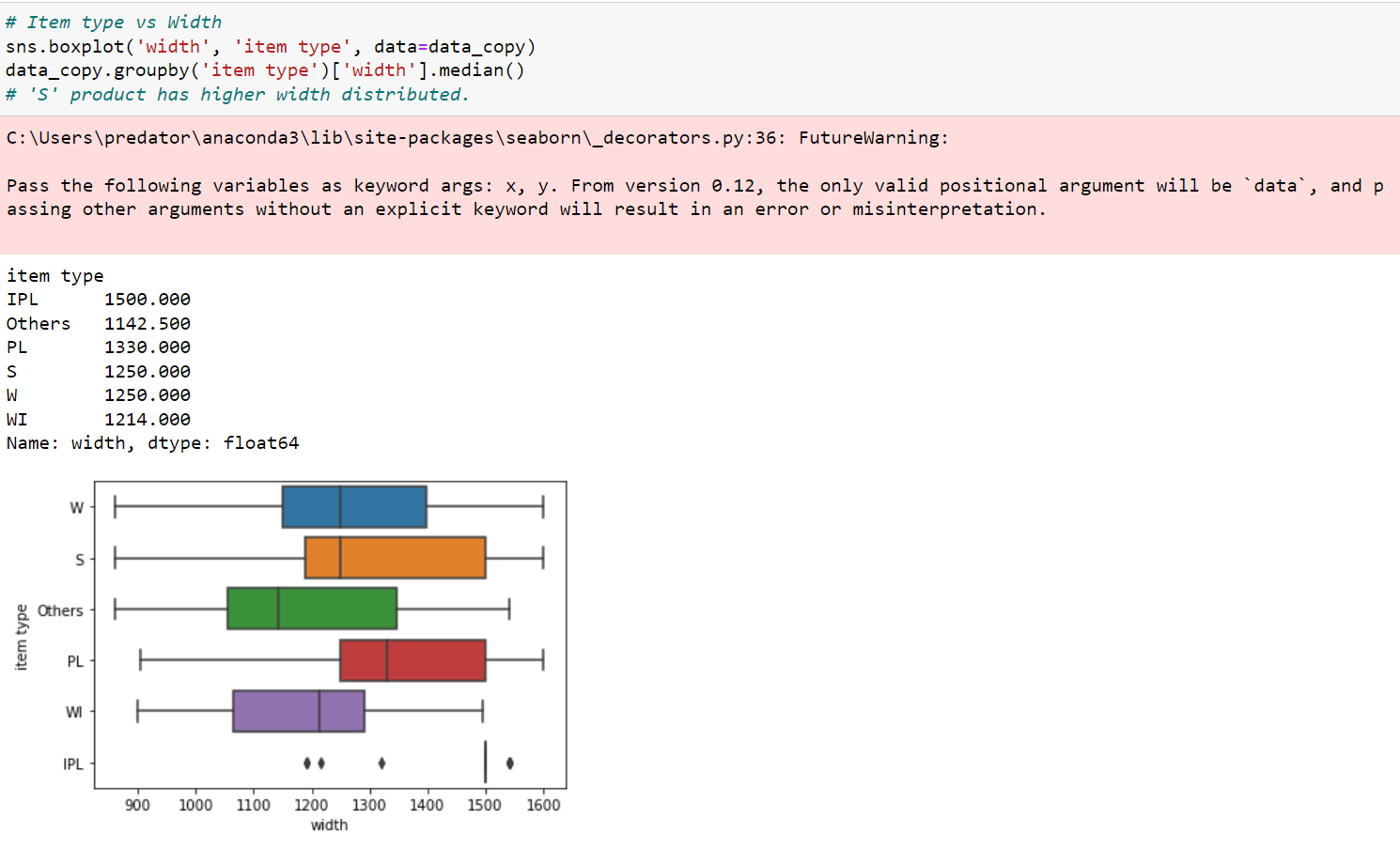


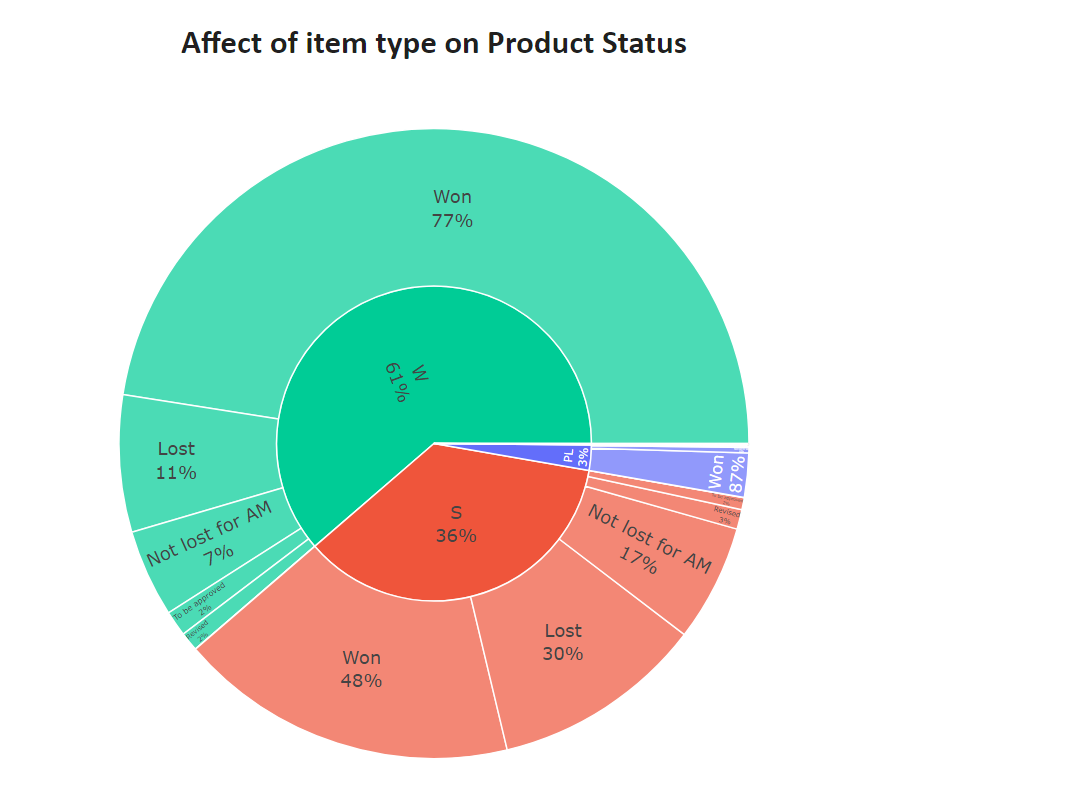


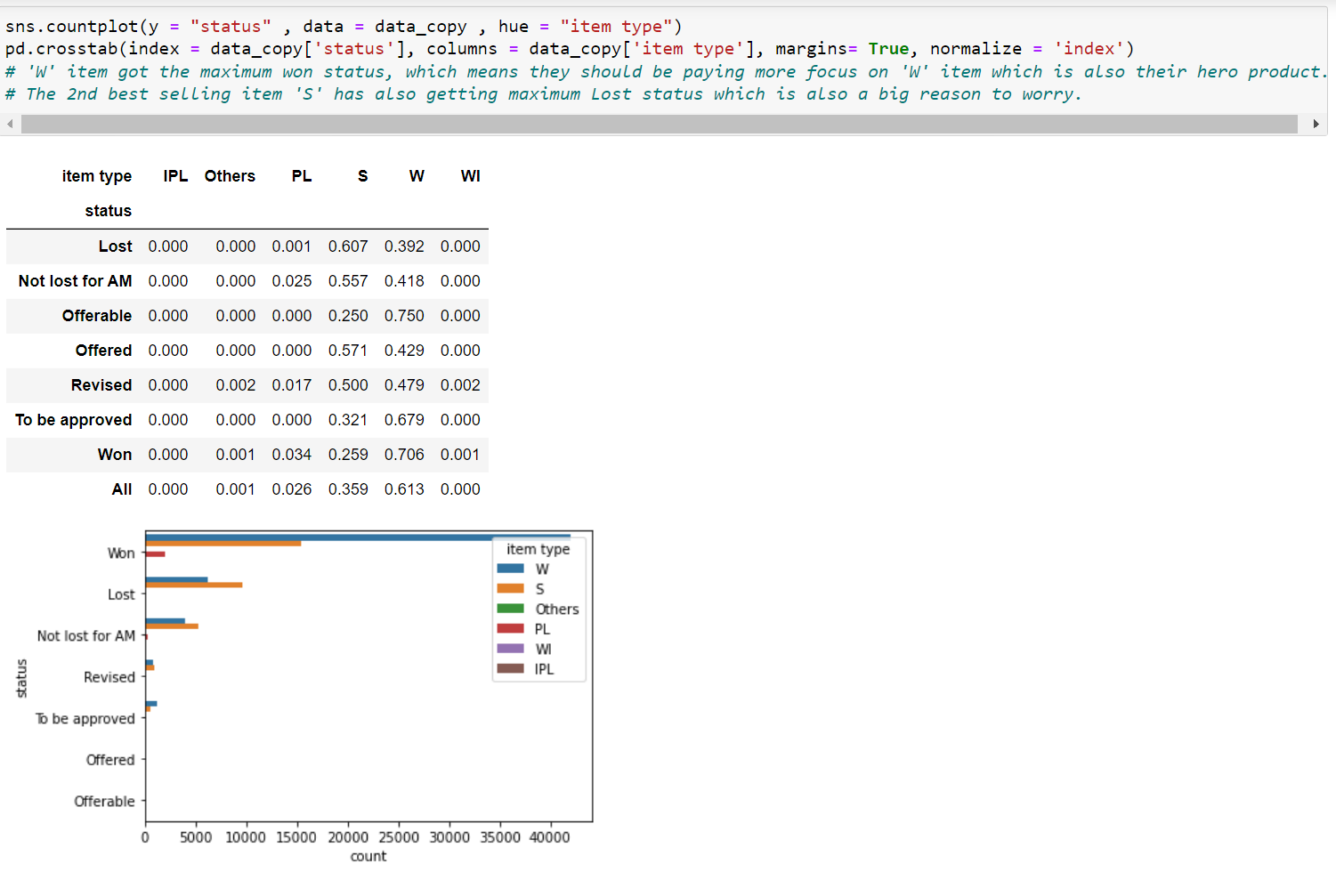


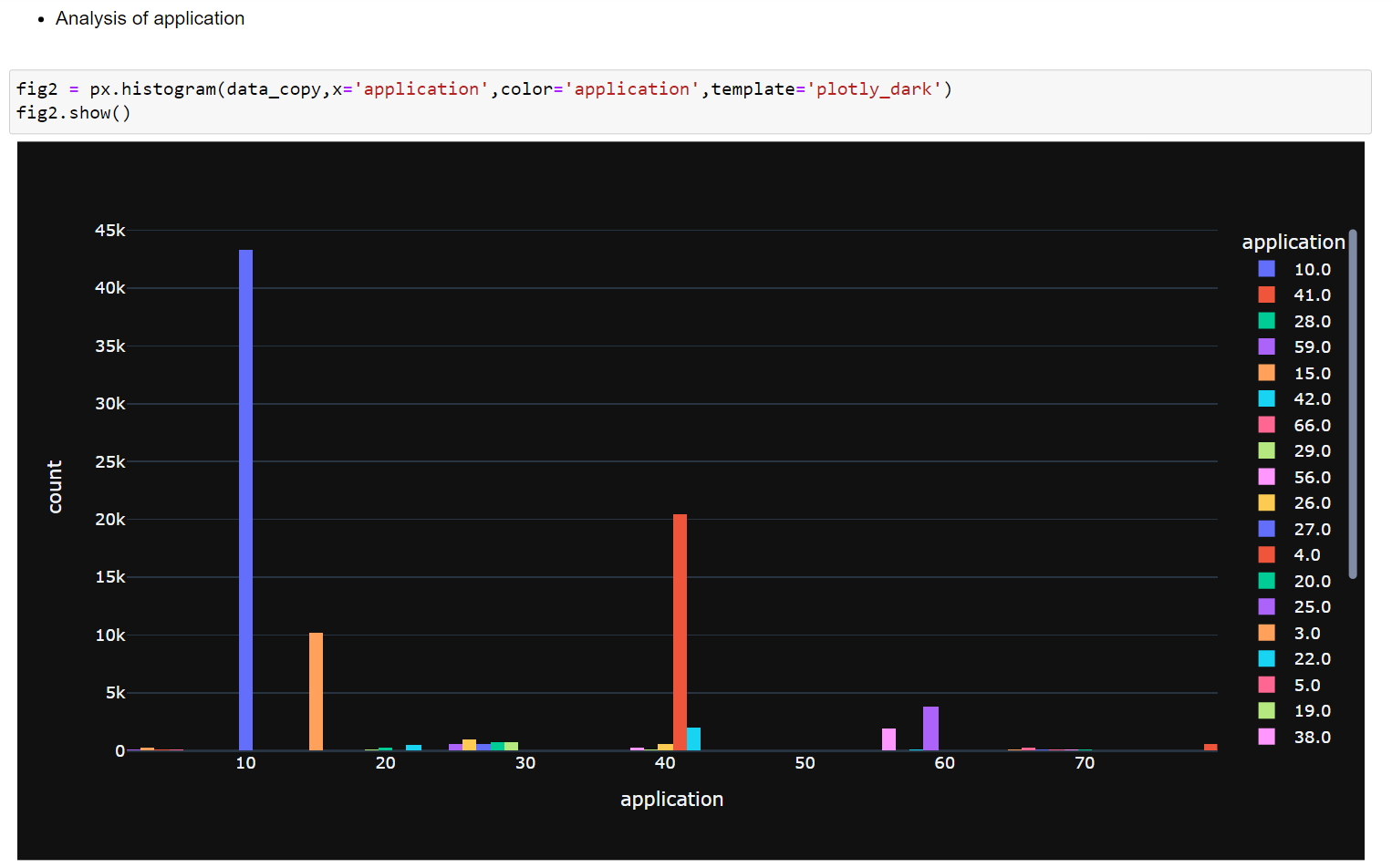


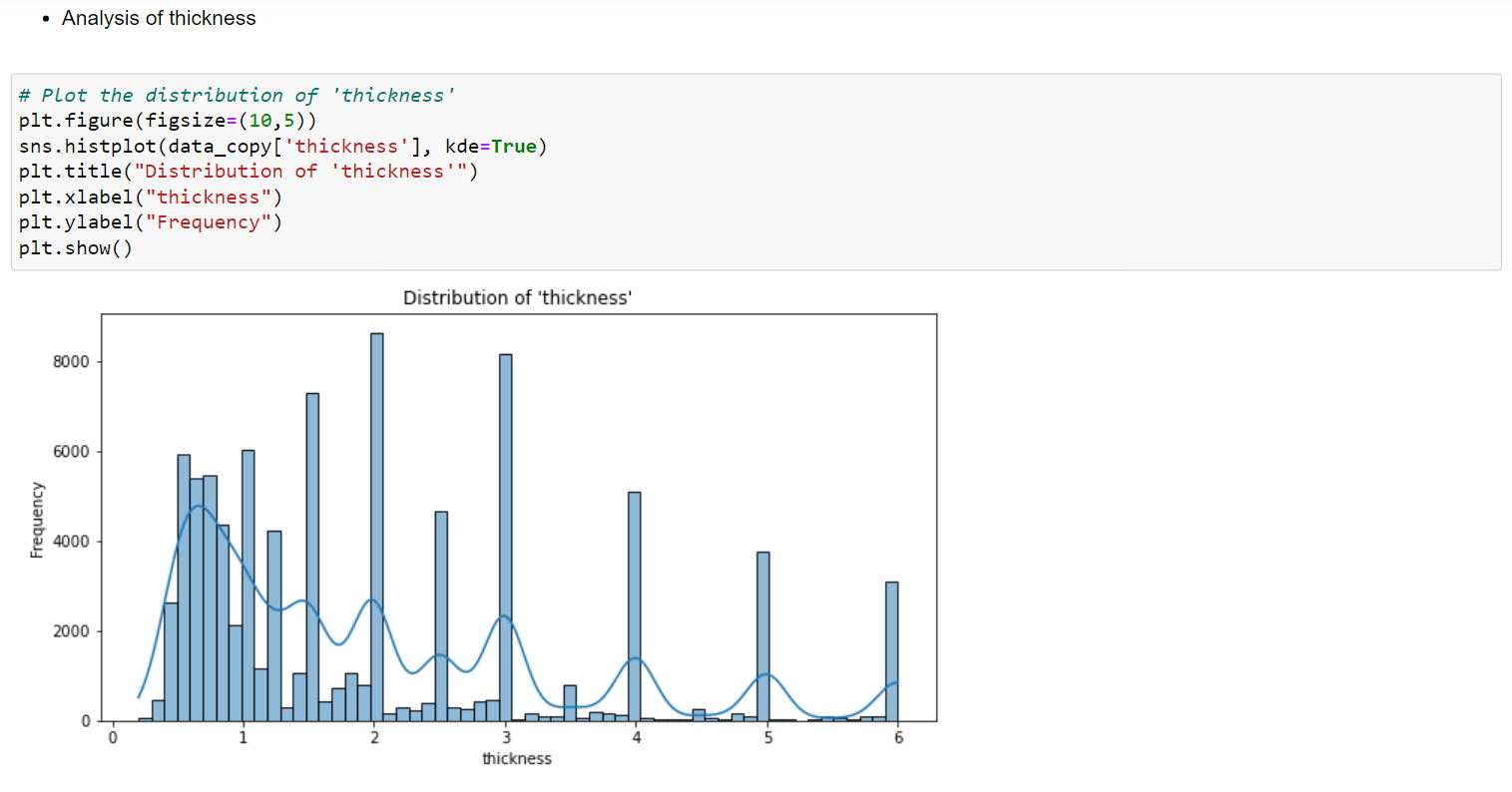


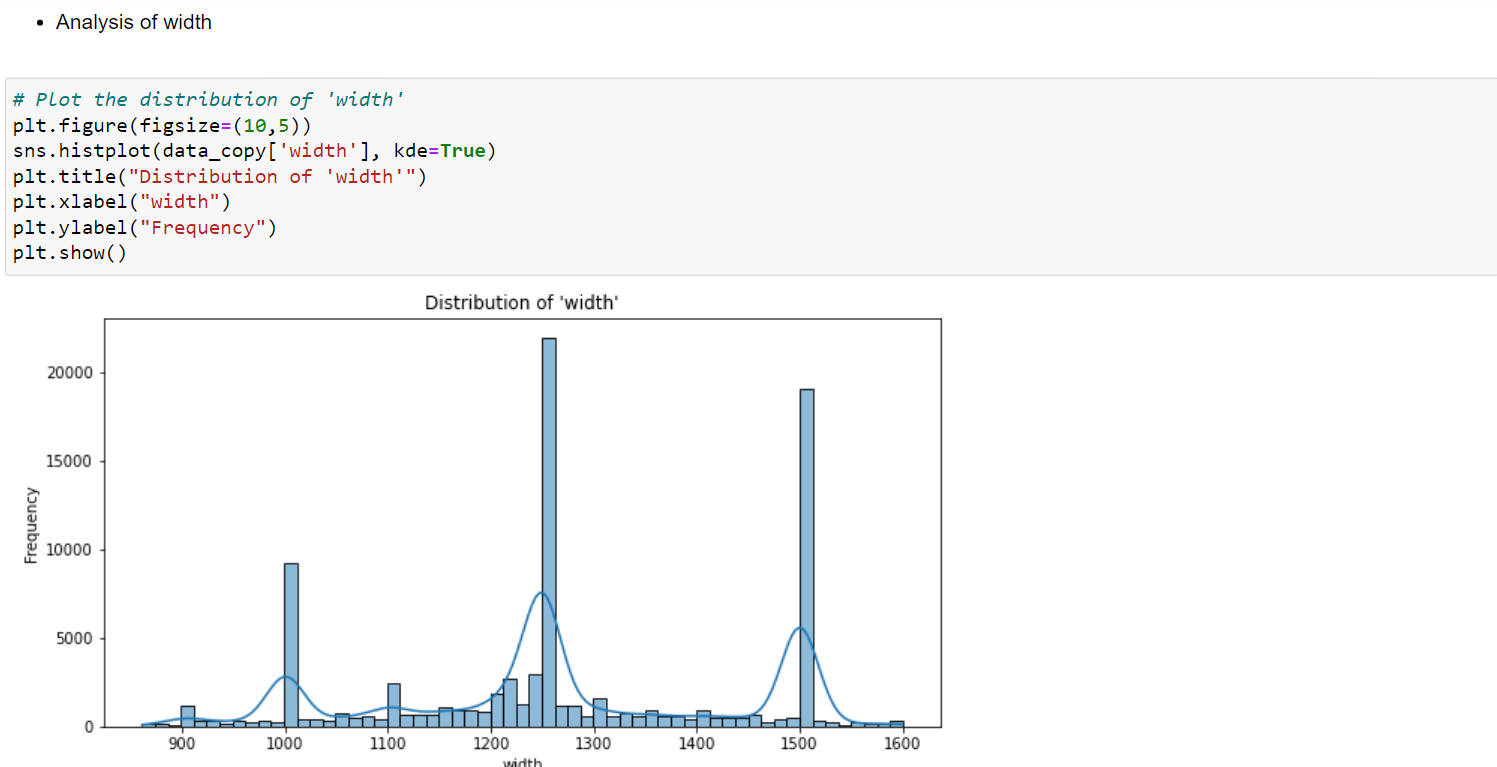


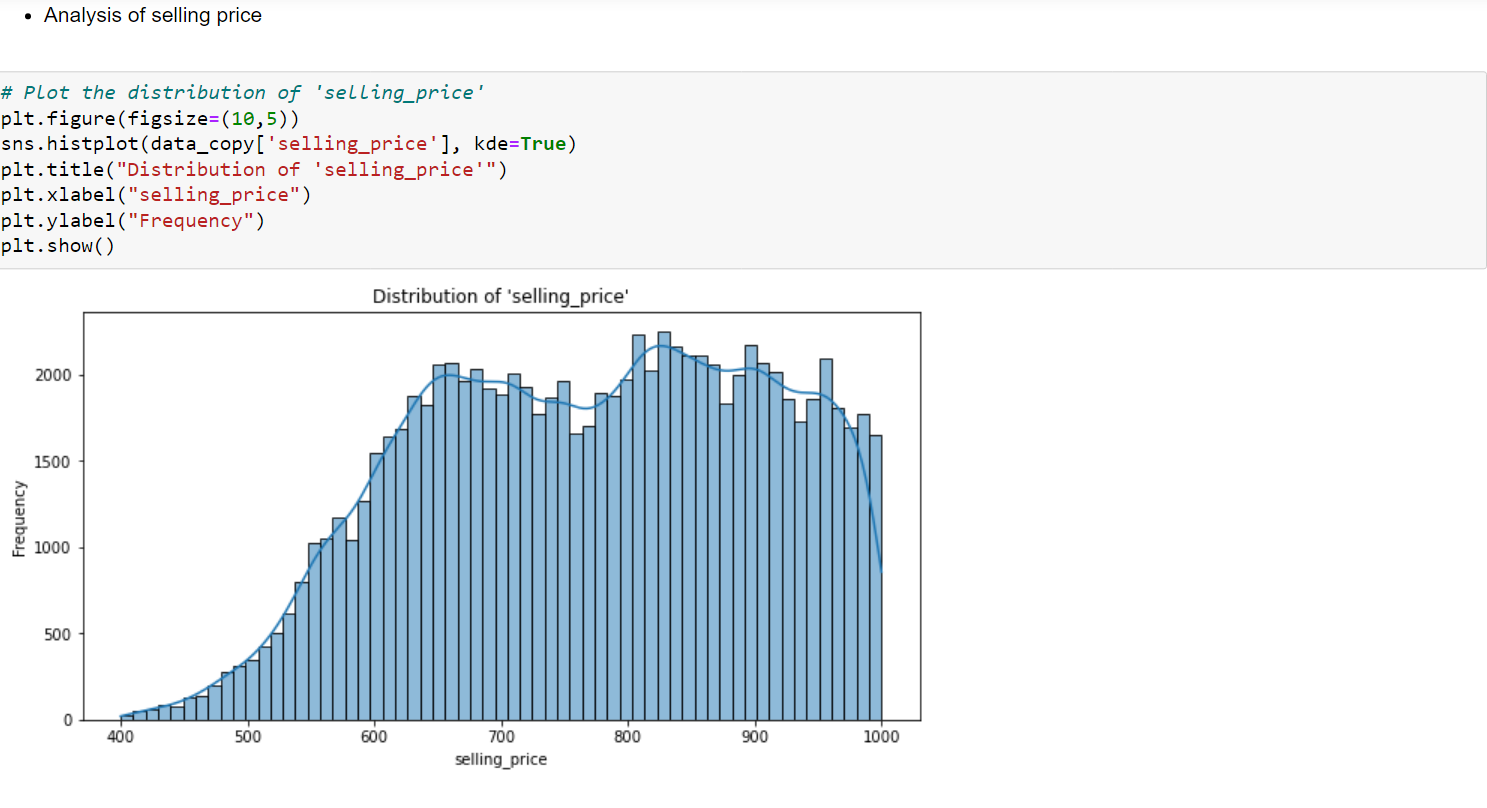


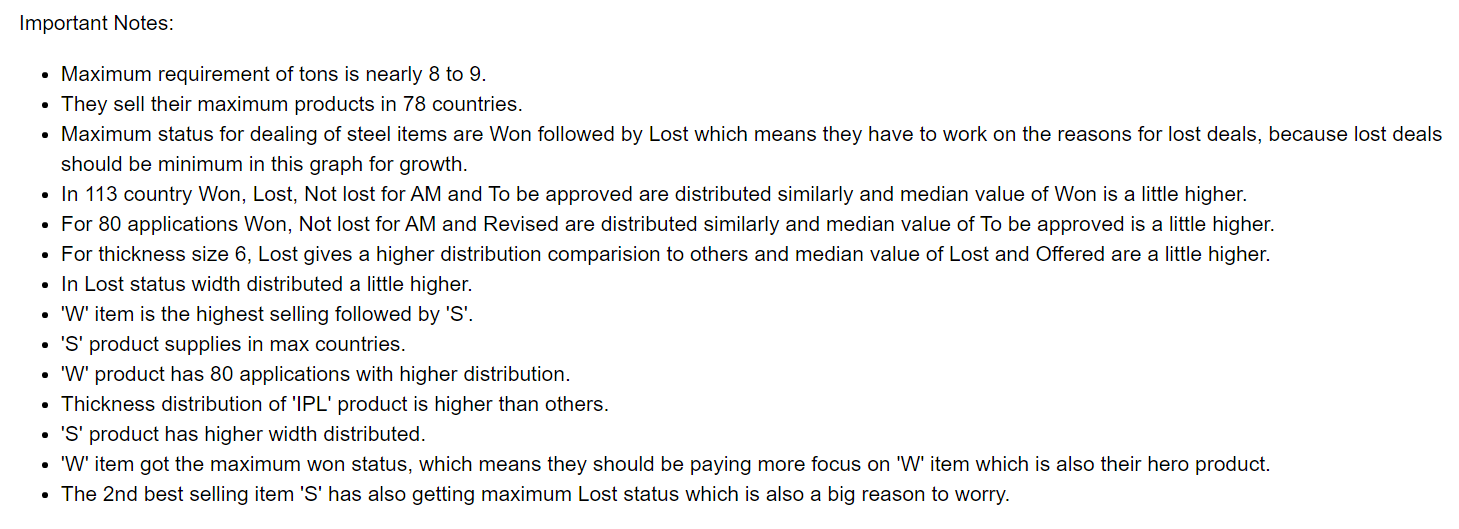




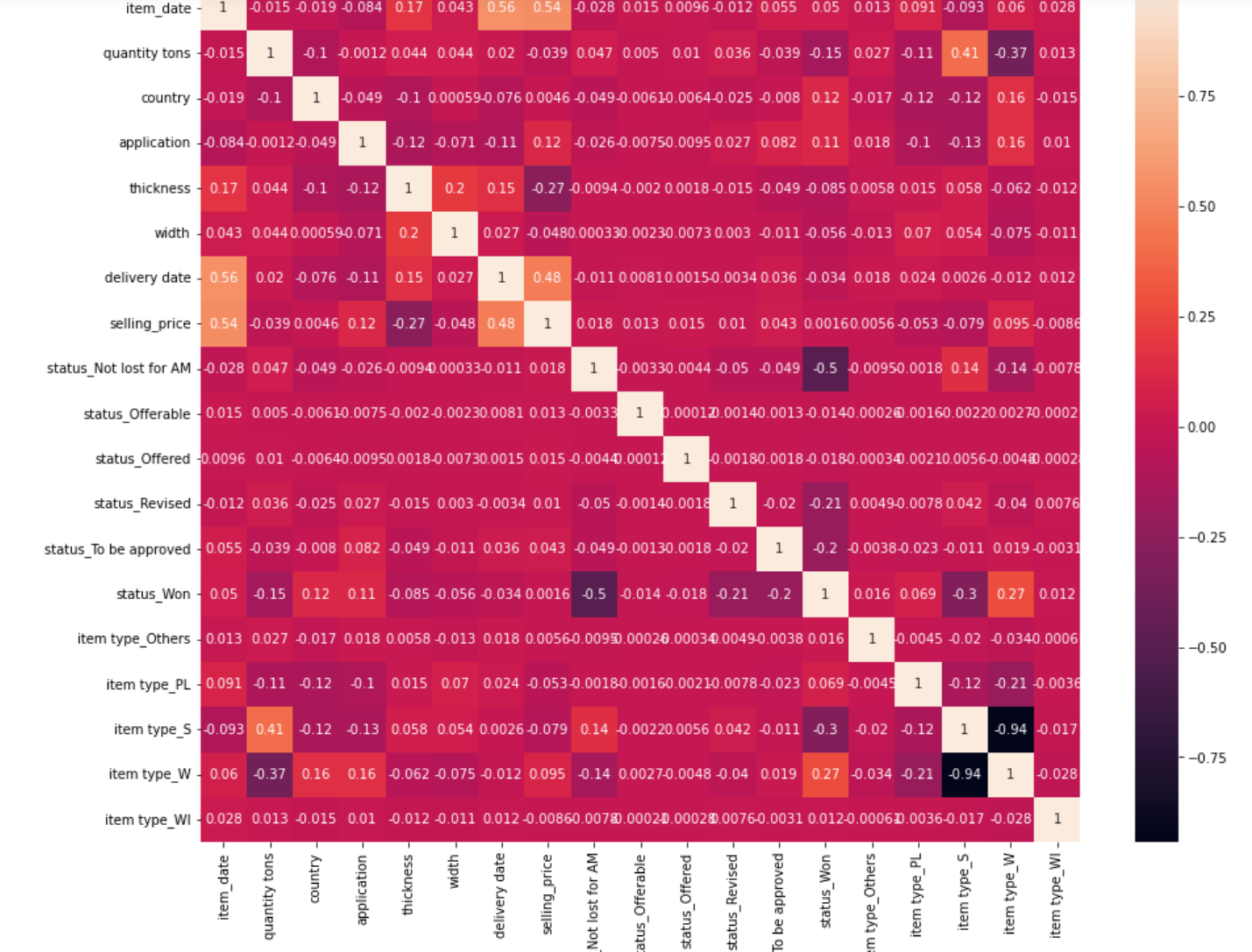






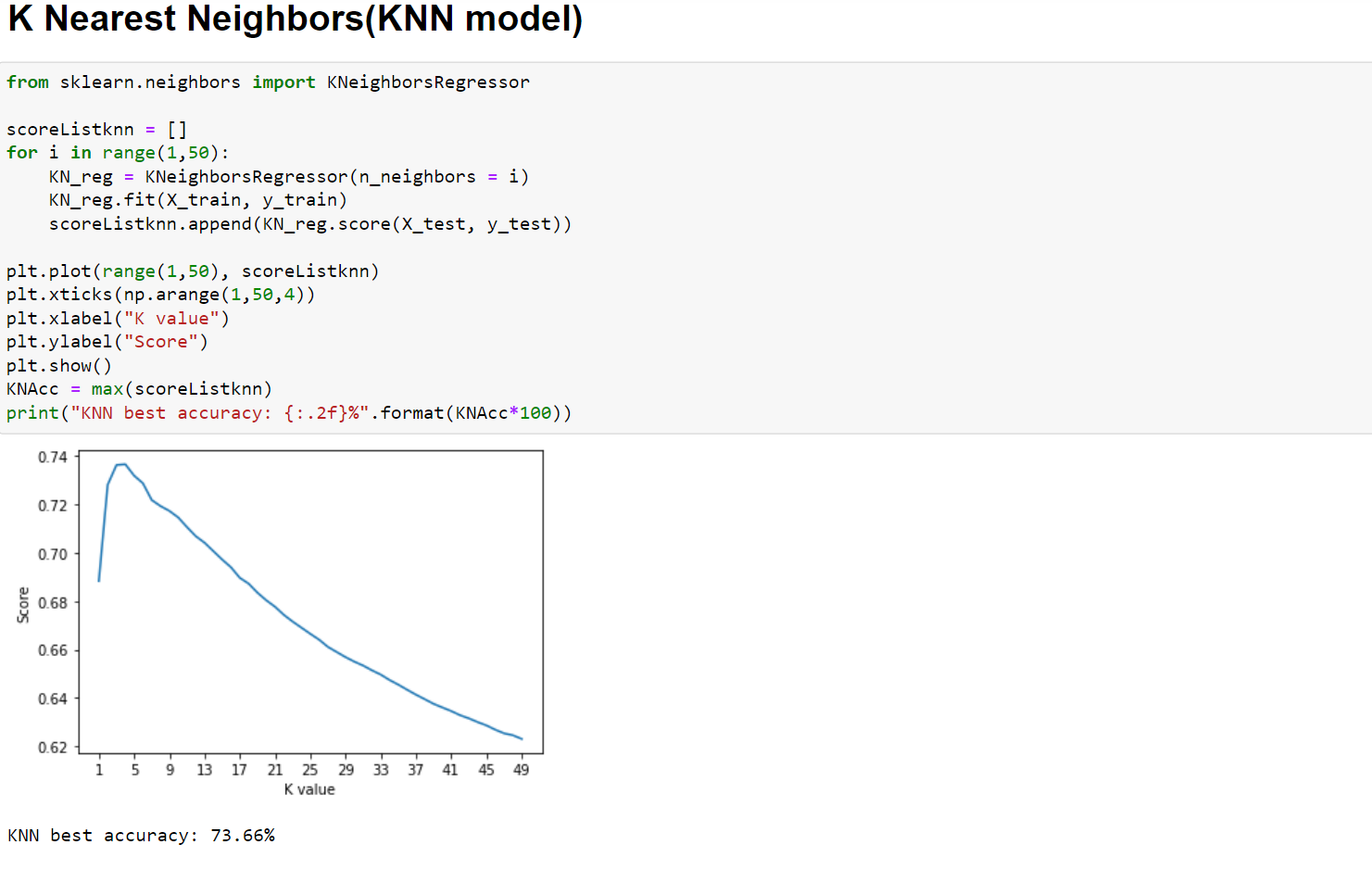


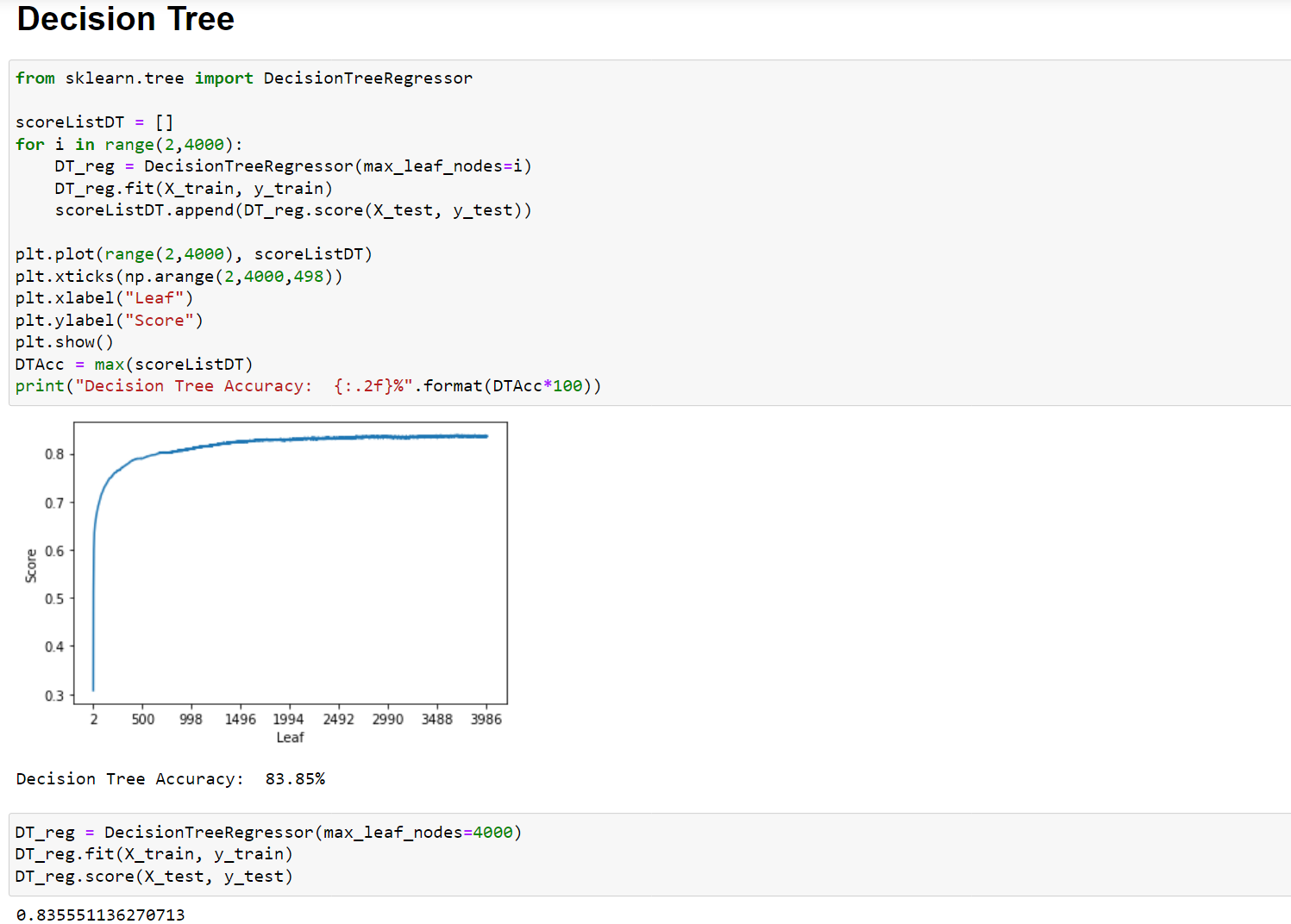


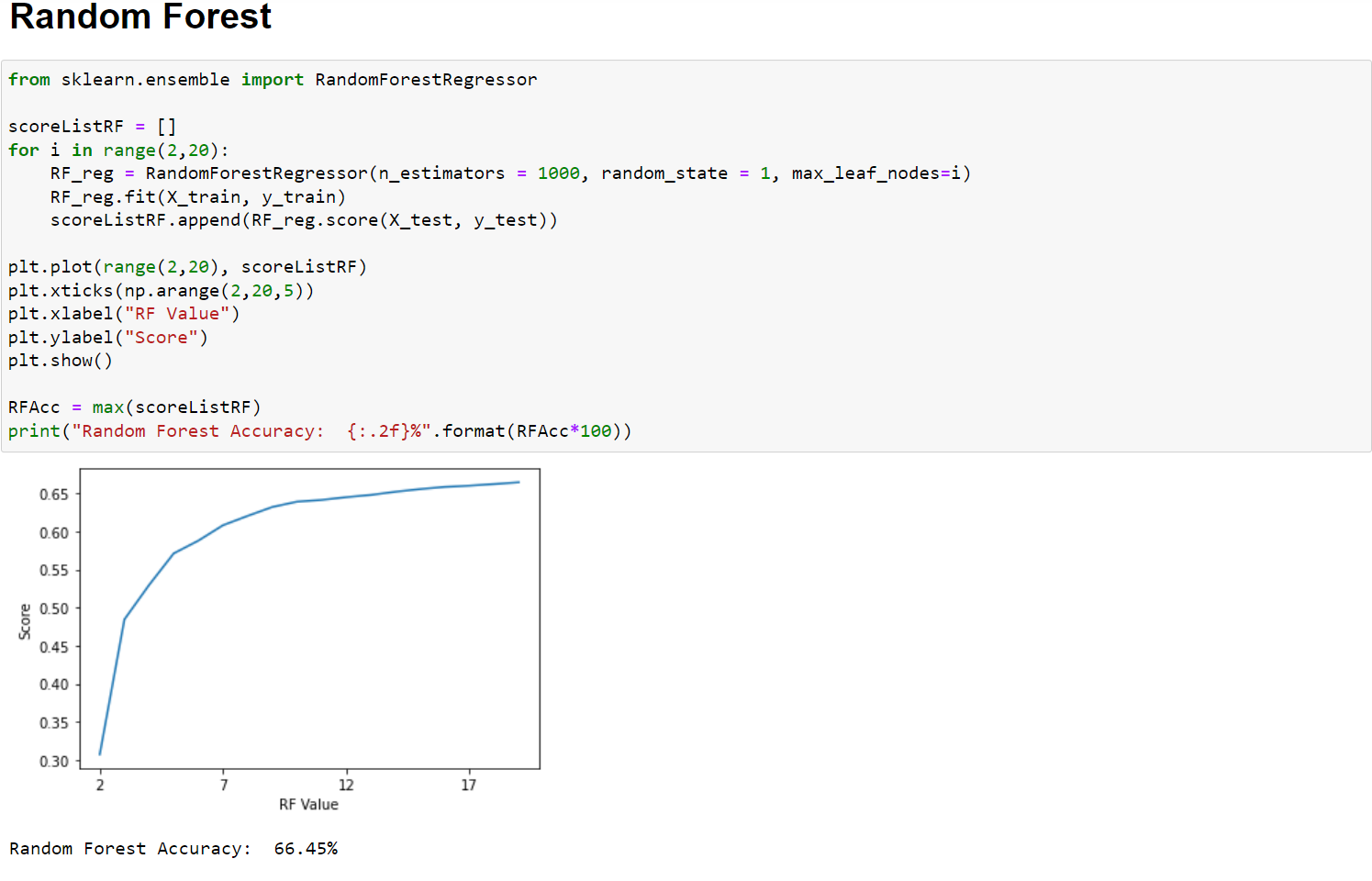


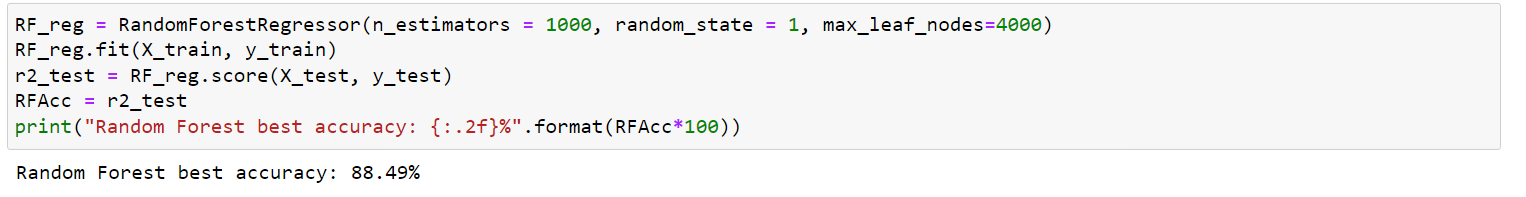










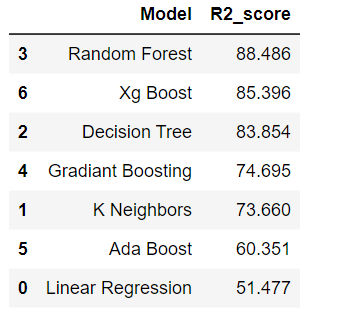








## Model Comparison:



* KNN and SVR are taking too much because of large amount of data.
* We can't get better accuracy because of limited speed of machine.

## Conclusion:

Random Forest gives maximum R2 score of 0.88486 or 88.486 %.