## Introduction

This document provides the guidelines for preparing your jupyter notebook towards creating datadocs.

## Datadoc Framework

The datadoc framework consists of a main datadoc parser which comes as a jupyter notebook file. This has the python code to convert your content notebook to a Datadoc in a Plotly Dash application. The file is stored in the following github URL:

<https://github.com/colaberry/datadocgen/blob/master/colaberry-datadoc_parser-new.ipynb>

The input for the datadoc parser come as three (3) different files for ease of maintainability.

1. Input Content Notebook
2. Content Layout Template
3. Global Template (default code)

The datadoc content creators must work on (1) the Input Content Notebook and (2) the Content Layout Template to match the needs of the specific datadoc. The Global Template and the Datadoc parser are part of the framework. The configurations to these global files are made by the Datadoc admin during deployment.

## Input Content Notebook

The main content for a datadoc is placed in a jupyter notebook - namely the Input Content Notebook. The language supported in python.

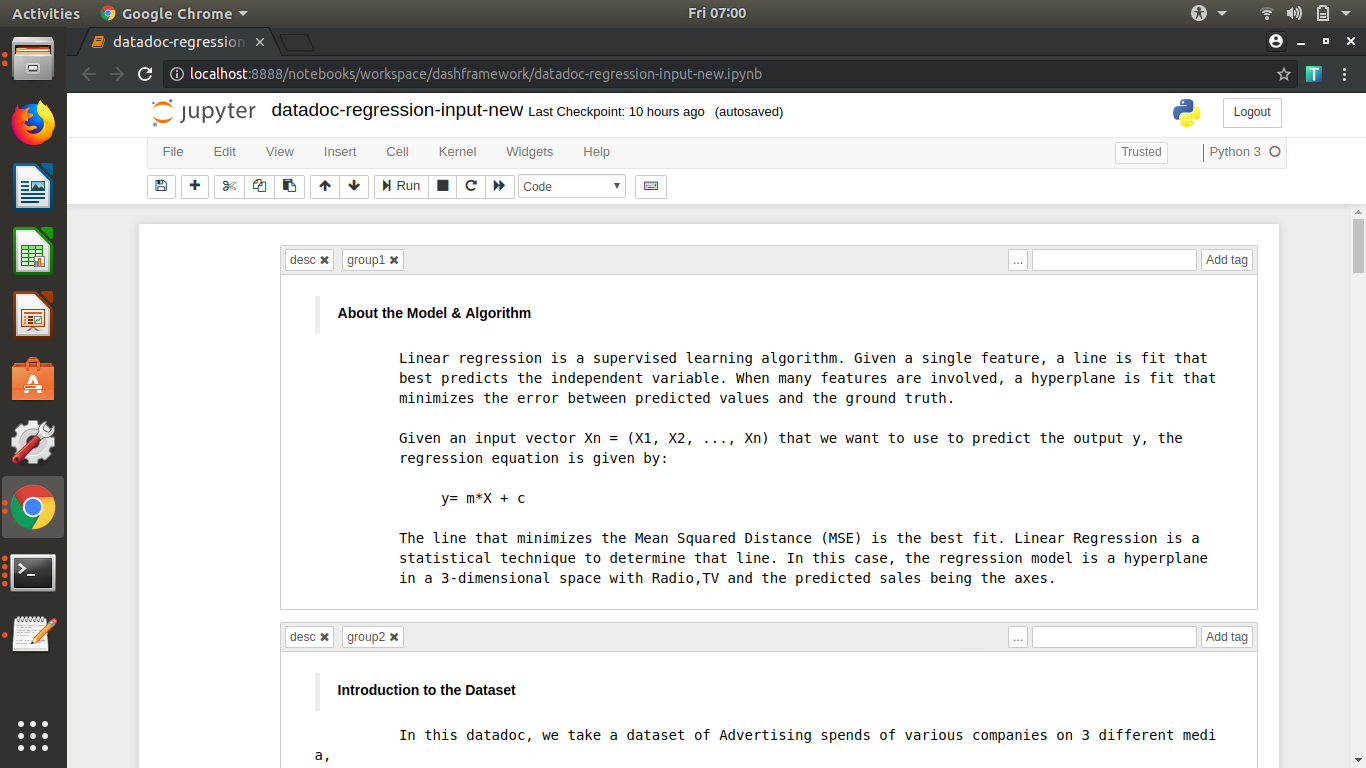
A sample input notebook is available in the below URL:

<https://github.com/colaberry/datadocgen/blob/master/datadoc-regression-input-new.ipynb>

This follows a simple format for text, followed by plot cells. The text cells are Markdown cells in the jupyter notebook. Plot cells are usually code cells with the “plot” tag. The text cells come with the “desc” tag. You could also include code cells with “code” tags where the code is not used for display in a datadoc but just runs in the background.

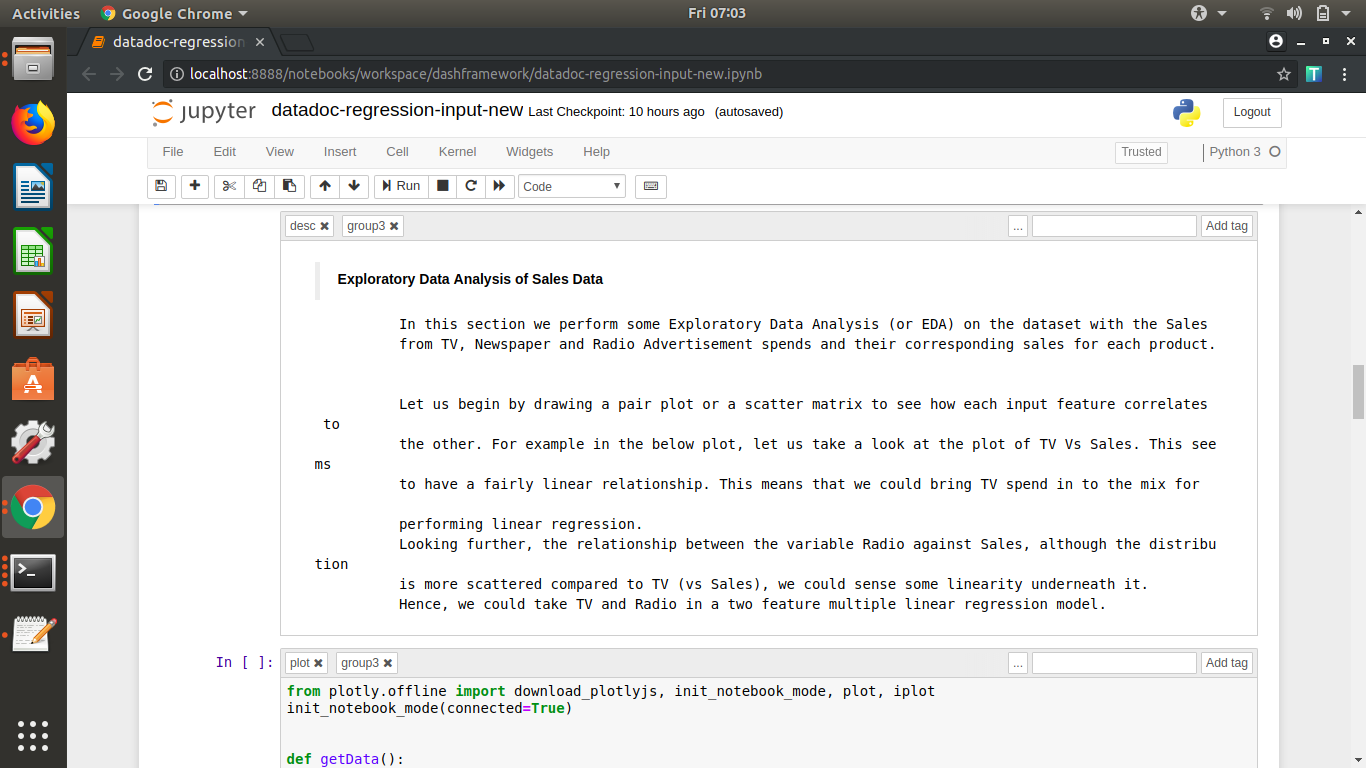
### Grouping of Cells

One important aspect of Input Notebook is the grouping of text and plot cells for sequential display in the datadoc. For that purpose, each text and plot cell must have a second tag “groupX” where X is a number. For example the first two cells could be a “desc” cell followed by a “plot” cell, both of them having “group1” as the second tag.

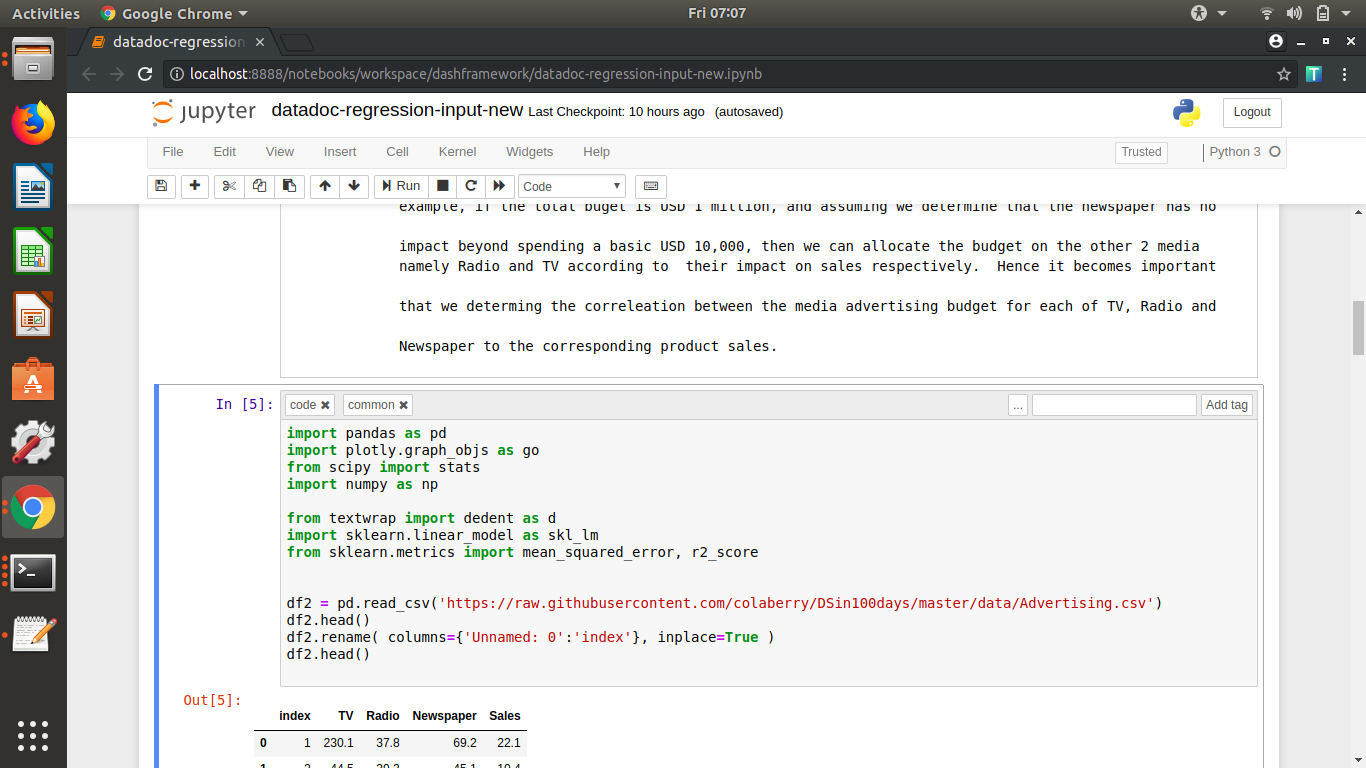


In the above screenshot, the first cell has “desc” as “group1” with only text and not having a plot, as this is the introduction. Hence we do not have a plot yet. The second cell is a group in itself namely “group2”, with just a text “desc” cell.

However in the screenshot below, we have a pair of text “desc” cell and a “plot” cell to be displayed in order in the datadoc as part of a group “group3”.



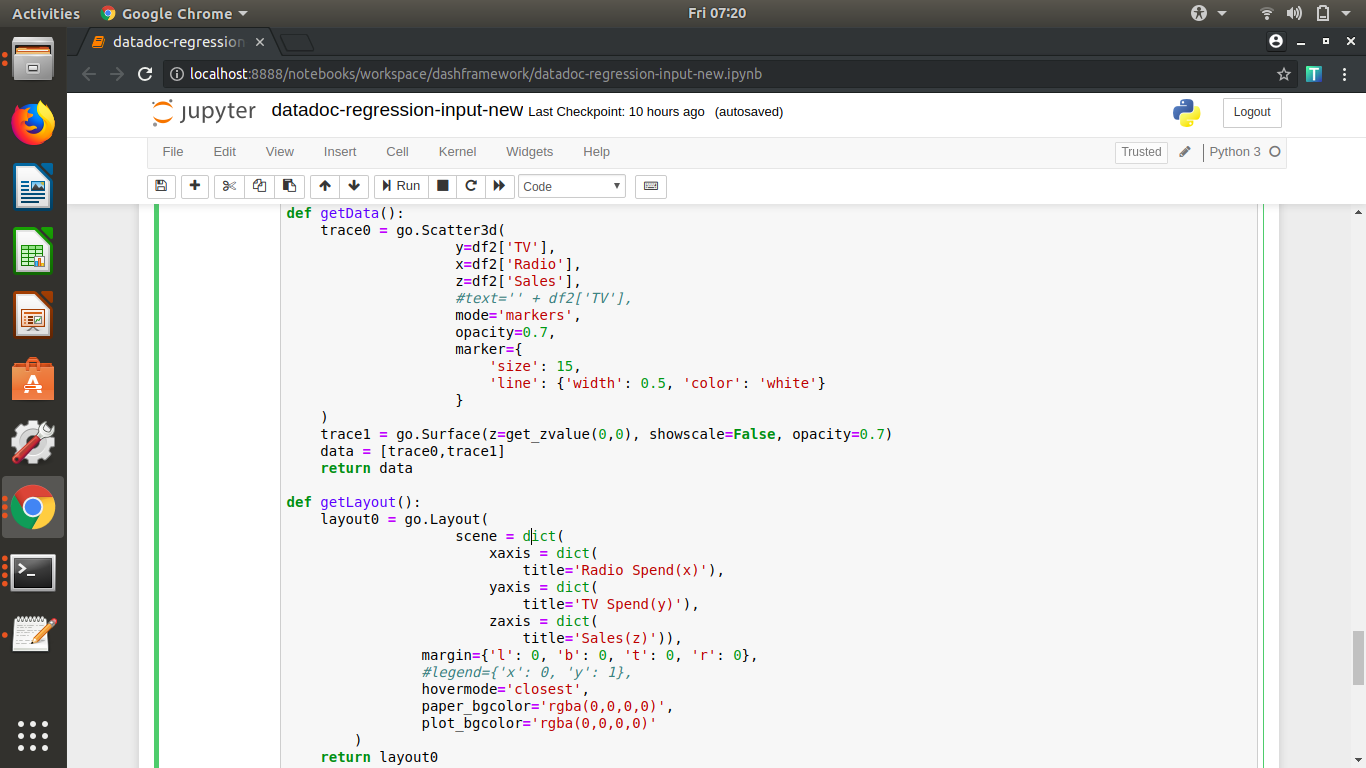
IMPORTANT: Note that the order of the tags is important. The first tag must always be one of “desc”, “code” or “plot”. The second tag must always be “groupX” for text or plot cells. For “code” cells the second tag must be “common”. Please the see the screenshot below for an example.



### Plot Requirements

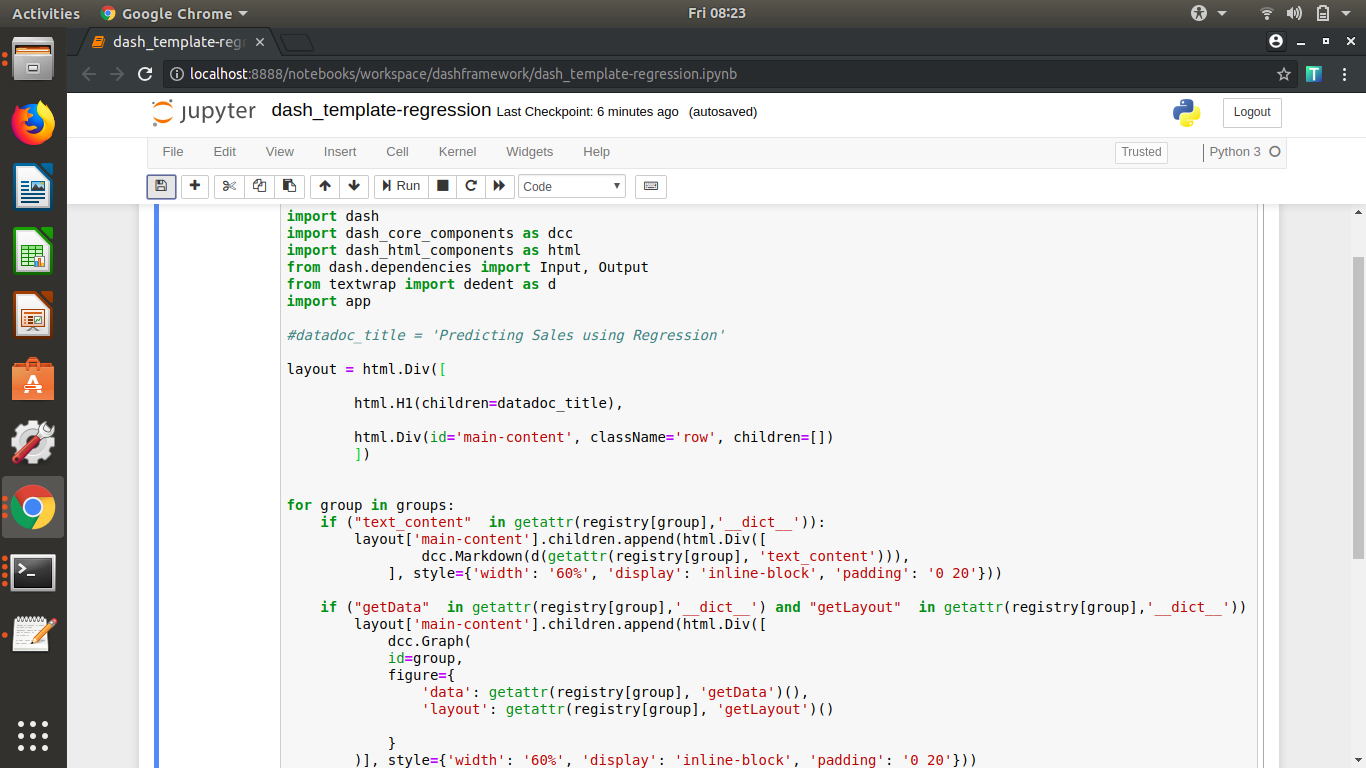
The plots are usually plotly plots. However, we could support Matplotlib in later versions.

The plotly plots in the input content notebook must have two methods with names - getData() and getLayout(). See example screenshot below:



## Content Layout Template

In addition to the Input Content, depending on the specific requirement, you may also need to have a version of the Layout Template. Layout template specifies the look and feel of the datadoc with specific details such as where must the text block appear or where the plot must be displayed or what must be the layout. Given below is a sample screenshot.



You can usually copy the above template for a straightforward input content notebook. Alternatively, you could modify the code to have a specific layout for your datadoc.

The sample layout file is available at the below URL:

<https://github.com/colaberry/datadocgen/blob/master/dash_template-regression.ipynb>

## Repository

The files for the datadoc generator, namely, the parser, sample input, sample layout and other deployment artifacts are available in the below repository.

<https://github.com/colaberry/datadocgen>

IMPORTANT: Please ensure you DO NOT modify anything other than the input content notebook and the layout template notebook.