

Data Description: The Metropolitan Museum of Art presents over 5,000 years of art from around the world for everyone to experience and enjoy. The Museum lives in three iconic sites in New York City—The Met Fifth Avenue, The Met Breuer, and The Met Cloisters. Millions of people also take part in The Met experience online. Since it was founded in 1870, The Met has always aspired to be more than a treasury of rare and beautiful objects. Every day, art comes alive in the Museum's galleries and through its exhibitions and events, revealing both new ideas and unexpected connections across time and across cultures. The Metropolitan Museum of Art provides select datasets of information on more than 470,000 artworks in its Collection for unrestricted commercial and noncommercial use.

Critical Details and Instructions:

i. Included with these directions should be a .csv file (**MetObjects_Subset.csv**) that consists of only a small subset of objects (~17.3k) in the museum. **You should use this file as a basis for all instructions that follow.**

ii. You must use either a .ipynb notebook with separate cells per problem or a .py file with separate functions per problem in your submission. In the latter case, you should also provide clear documentation as to how to use your code.

iii. For problems 1-5, you can manipulate the data-frames/dictionaries as you see fit and using whatever functions/libraries you want. **However, it is critically important that your end results for each problem match the provided variable name (ex: the result of problem 1 is called `df_prob1`) so that they are accessible for grading.**

iv. With the exception of problem 1 (which is trivial) you should include a few comments in your notebook or code that make it clear what your thought process and/or code does to address each problem. Failure to do so will result in penalties (2 points) per problem.

1. Load the .csv file into a pandas data-frame (DataFrame) called **`df_prob1`** with appropriate rows and columns. Each row must represent an individual object, while each column must represent a particular attribute/variable (Object Number, Title, Dynasty, etc.)

2. Many columns of this data are missing *entirely* (i.e. no values are present for *any* objects). Use Python to determine which columns are missing for **all** of the provided objects and create a modified copy of **`df_prob1`** that *excludes* these columns entirely called **`df_prob2`**.

Hint: There are multiple ways to do this, but you may want to look into the pandas member function **`isna`**.

3. Suppose we are only interested in objects in the public domain. You are to create a copy of **`df_prob2`** called **`df_prob3`** that consists only of those objects where the “Is Public Domain” column is “TRUE”; **note that if the “Is Public Domain” column does not contain *any* string for a given object, it *should not* be present in `df_prob3`.**

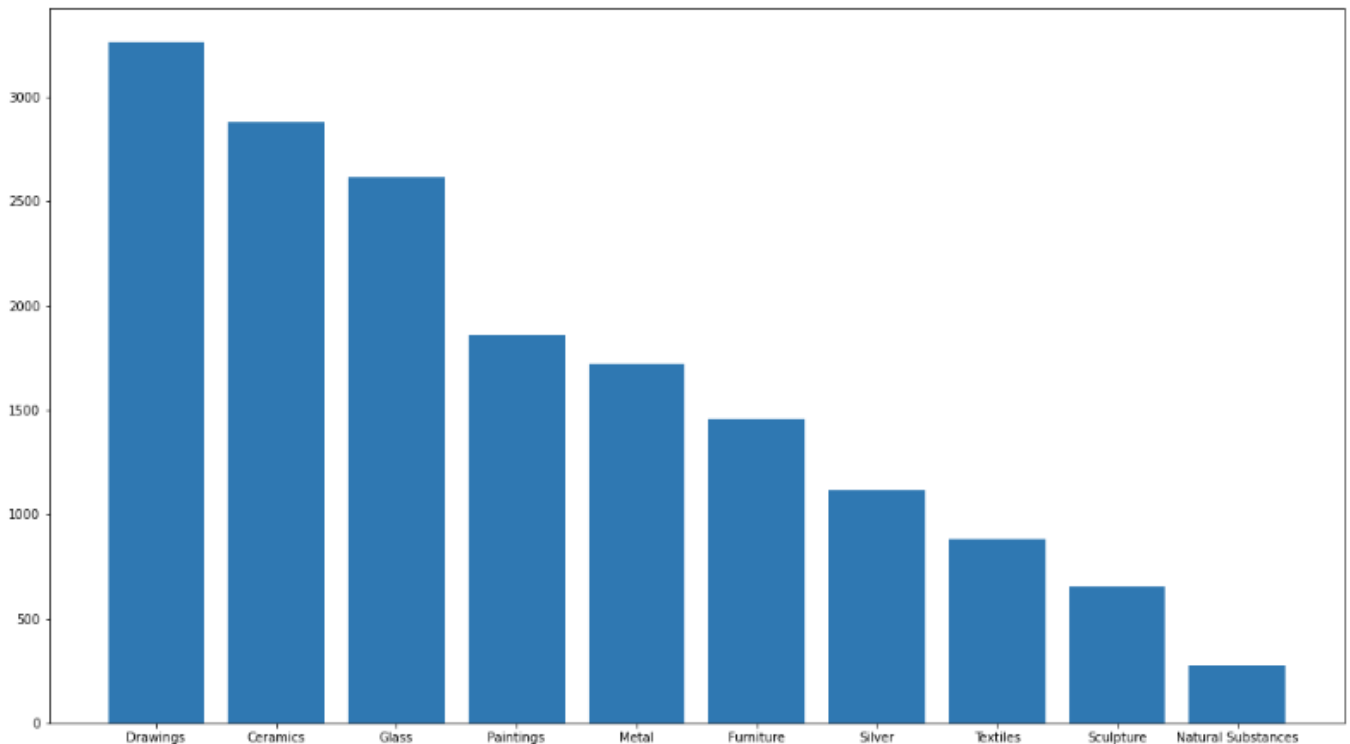
4. The objects in the dataset include a column “Artist Role.” You are to create a copy of **`df_prob3`** called **`df_prob4`** that includes the string “Maker” within the “Artist Role” column. Note – while this may seem trivial, the “Artist Role” column can contain *multiple roles* for a given object. You must be certain that if ***at least one*** of the roles is Maker, then the object/row will appear in the **`df_prob4`** data-frame.

Hint: you *can* use regular expressions for this problem, but it is not required, nor is it likely the easiest way to handle the problem.

Note: for part 5 you will want to utilize the “Object Begin Date” and “Object End Date” columns in your data manipulation. The “Object Date” column is unreliable and largely missing in information.

5. For this part, you are to create a copy of `df_prob4` called `df_prob5` and add a *new column* to `df_prob5` called *DecadesToMake*, which represents the number of decades that have elapsed between the start year and end year of the object’s creation **rounded down**. For example, an object with “Object Begin Date” of 1890 and “Object End Date” of 1910 would have a DecadesToMake value of 2, while another object with begin date of 1932 and end date of 1941 would have a DecadesToMake value of 0. **Note: Any object missing either start date or end date, or with a start date and end date in the same year, should have a DecadesToMake value of 0.**

6. Most objects in the dataset have a *Classification*. For this problem, you are to produce a bar plot depicting the ten most common Classifications for objects in `df_prob5`, with classification labels on the x-axis and counts on the y-axis, sorted in descending order of counts from left to right. The example below should give you an idea of the format you want to use, but note that this was generated using the basic DataFrame (`df_prob1`) and will not reflect the expected results you will observe for `df_prob5`. In fact, you may not see some of these same classifications at all on your plot, for example.



You should upload your submission just as you would any standard assignment via Blackboard – but if you *cannot* do so, email it to me ASAP. Note that if you are submitting a .py file you are highly encouraged to include a README to explain what should be run to produce the required structures for problems 1-5 and graphs for problem 6. Failure to do so, should your code be confusing or difficult to use will result in penalties.