DCS 540 Data Preparation (DSC540-T301 2225-1)

Bellevue University

Assignment: Weeks 7 & 8 Exercises

Author: Jake Meyer

Date: 05/07/2022

Perform at least 8 exercises from either of the two datasets.

The *Python for Data Analysis* reading this week includes Chapters 7-8 and 10-11. An optional reading is Chapter 12. The two datasets provided for the assignment are listed below:

- So Much Data Candy, Seriously.csv. (Ng, 2017)
- The Metropolitan Museum of Art Open Access CSV

Two methods from each chapter must be applied on the dataset. The same dataset does not have to be used for all 8 methods that are applied. The transformation and/or cleaning methods for each chapter are outlined below:

Chapter 7

- Filter out missing data
- Fill in missing data
- Remove duplicates
- Transform data using either mapping or a function
- Replace values
- Discretization and Binning
- Manipulate Strings

Chapter 8

- Create hierarchical index
- Combine and Merge Datasets (you will have to either create a new dataset from your existing data or create a relationship between the data I have provided)
- Reshape
- Pivot the data

Chapter 10

- Grouping with Dicts/Series
- Grouping with Functions
- Grouping with Index Levels
- Split/Apply/Combine
- Cross Tabs

Chapter 11

- Convert between string and date time
- Generate date range
- Frequencies and date offsets
- Convert timestamps to periods and back
- Period Frequency conversions

In [406…
```
'''
Import the necessary libraries to complete the assignment.
Other libraries that will be installed for this activity will be NumPy, Pandas, Beautif
'''
import numpy as np

import pandas as pd

# from bs4 import BeautifulSoup - not required

# import requests - not required

# import matplotlib.pyplot as plt

# from scipy import stats
```

In [407…
```
'''
Read the Met Muesuem data from the open access location as met_csv with pd.read.csv().
'''
open_access = "https://media.githubusercontent.com/media/metmuseum/openaccess/cc8ffeafd
met_df = pd.read_csv(open_access)
```

```
C:\Users\jkmey\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: DtypeW
arning: Columns (5,7,10,11,12,13,14,34,35,36,37,38,39,40,41,42,43,44,45,46) have mixed t
ypes.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [408…
```
'''
View met_df to understand the high level structure and content.
'''
met_df.head(10)
```

Out[408…

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Gallery Number	Department	AccessionYear	Object Name	Tit

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Gallery Number	Department	AccessionYear	Object Name	Tit
0	1979.486.1	False	False	False	1	NaN	The American Wing	1979.0	Coin	One doll Liber Hea Co
1	1980.264.5	False	False	False	2	NaN	The American Wing	1980.0	Coin	Ten doll Liber Hea Co
2	67.265.9	False	False	False	3	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co
3	67.265.10	False	False	False	4	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co
4	67.265.11	False	False	False	5	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co
5	67.265.12	False	False	False	6	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co
6	67.265.13	False	False	False	7	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co
7	67.265.14	False	False	False	8	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co
8	67.265.15	False	False	False	9	NaN	The American Wing	1967.0	Coin	Two and-a Ha Doll Co

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Gallery Number	Department	AccessionYear	Object Name	Tit
9	1979.486.3	False	False	False	10	NaN	The American Wing	1979.0	Coin	Two and-a hal doll India Hea Co

10 rows × 54 columns

In [409…
```
'''
Understand the shape of met_df initially.
'''
print('There are {} rows and {} columns in from the met CSV file.'.format(met_df.shape[
```

```
There are 477804 rows and 54 columns in from the met CSV file.
```

In [410…
```
'''
Move a copy of the candy data files into the current working directory.
Read the candyhierarchy2017.csv file as candy_csv with pd.read.csv().
Read the candyhierarchy2016.xlsx, candyhierarchy2015.xlsx, and
candyhierarchy2014.xlsx with pd.read_excel().
'''
candy_df2017 = pd.read_csv("candyhierarchy2017.csv", encoding='latin-1')
candy_df2016 = pd.read_excel('candyhierarchy2016.xlsx')
candy_df2015 = pd.read_excel('candyhierarchy2015.xlsx')
candy_df2014 = pd.read_excel('candyhierarchy2014.xlsx')
```

```
C:\Users\jkmey\anaconda3\lib\site-packages\openpyxl\worksheet\_reader.py:312: UserWarnin
g: Unknown extension is not supported and will be removed
  warn(msg)
```

In [411…
```
'''
View  candy_csv files to understand the high level structure and content.
candy_df2017 will be shown below:
'''
candy_df2017.head(5)
```

Out[411…

	Internal ID	Q1: GOING OUT?	Q2: GENDER	Q3: AGE	Q4: COUNTRY	Q5: STATE, PROVINCE, COUNTY, ETC	Q6 | 100 Grand Bar	Q6 | Anonymous brown globs that come in black and orange wrappers\t(a.k.a. Mary Janes)	Q6 | Any full-sized candy bar	Q6 | Black Jacks
0	90258773	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	Internal ID	Q1: GOING OUT?	Q2: GENDER	Q3: AGE	Q4: COUNTRY	Q5: STATE, PROVINCE, COUNTY, ETC	Q6 | 100 Grand Bar	Q6 | Anonymous brown globs that come in black and orange wrappers\t(a.k.a. Mary Janes)	Q6 | Any full-sized candy bar	Q6 | Black Jacks
1	90272821	No	Male	44	USA	NM	MEH	DESPAIR	JOY	MEH
2	90272829	NaN	Male	49	USA	Virginia	NaN	NaN	NaN	NaN
3	90272840	No	Male	40	us	or	MEH	DESPAIR	JOY	MEH
4	90272841	No	Male	23	usa	exton pa	JOY	DESPAIR	JOY	DESPAIR

5 rows × 120 columns

In [412…

```
'''
View  candy_csv files to understand the high level structure and content.
candy_df2016 will be shown below:
'''
candy_df2016.head(5)
```

Out[412…

	Timestamp	Are you going actually going trick or treating yourself?	Your gender:	How old are you?	Which country do you live in?	Which state, province, county do you live in?	[100 Grand Bar]	[Anonymous brown globs that come in black and orange wrappers]	[Any full-sized candy bar]	[Black Jacks]	...
0	2016-10-24 05:09:23.033	No	Male	22	Canada	Ontario	JOY	DESPAIR	JOY	MEH	...

1	2016-10-24 05:09:54.798	No	Male	45	usa	il	MEH	MEH	JOY	JOY	...
2	2016-10-24 05:13:06.734	No	Female	48	US	Colorado	JOY	DESPAIR	JOY	MEH	...
3	2016-10-24 05:14:17.192	No	Male	57	usa	il	JOY	MEH	JOY	MEH	...
4	2016-10-24 05:14:24.625	Yes	Male	42	USA	South Dakota	MEH	DESPAIR	JOY	DESPAIR	...

5 rows × 123 columns

In [413…

```
'''
View  candy_csv files to understand the high level structure and content.
candy_df2015 will be shown below:
'''
candy_df2015.head(5)
```

Out[413…

	Timestamp	How old are you?	Are you going actually going trick or treating yourself?	[Butterfinger]	[100 Grand Bar]	[Anonymous brown globs that come in black and orange wrappers]	[Any full-sized candy bar]	[Black Jacks]	[Bonkers]	[Bo Ca
0	2015-10-23 08:46:20.451	35	No	JOY	NaN	DESPAIR	JOY	NaN	NaN	N
1	2015-10-23 08:46:51.583	41	No	JOY	JOY	DESPAIR	JOY	DESPAIR	DESPAIR	
2	2015-10-23 08:47:34.285	33	No	DESPAIR	DESPAIR	DESPAIR	JOY	DESPAIR	DESPAIR	DESI
3	2015-10-23 08:47:58.964	31	No	JOY	JOY	DESPAIR	JOY	DESPAIR	DESPAIR	
4	2015-10-23 08:48:11.719	30	No	NaN	JOY	DESPAIR	JOY	NaN	NaN	N

5 rows × 124 columns

In [414…
```
'''
View  candy_csv files to understand the high level structure and content.
candy_df2014 will be shown below:
'''
candy_df2014.head(5)
```

Out[414…

	ITEM	JOY	DESPAIR	NET FEELIES	NET CLOUT	DESPAIR (NEG)
0	York Peppermint Patties	634	78	556.0	1.639118	-78.0
1	Whole Wheat anything	21	419	-398.0	1.012938	-419.0
2	White Bread	15	473	-458.0	1.123440	-473.0
3	Vicodin	323	210	113.0	1.227036	-210.0
4	Twix	770	26	744.0	1.832497	-26.0

In [415…
```
'''
Understand the shape of each of the candy_df's.
'''
print('candy_df2017 contains {} rows and {} columns'.format(candy_df2017.shape[0],candy
print('candy_df2016 contains {} rows and {} columns'.format(candy_df2016.shape[0],candy
print('candy_df2015 contains {} rows and {} columns'.format(candy_df2015.shape[0],candy
print('candy_df2014 contains {} rows and {} columns'.format(candy_df2014.shape[0],candy
```

```
candy_df2017 contains 2460 rows and 120 columns
candy_df2016 contains 1259 rows and 123 columns
candy_df2015 contains 5630 rows and 124 columns
candy_df2014 contains 87 rows and 6 columns
```

Transformations from Chapter 7

In [416…
```
'''
met_df will be considered for the Chapter 7 transformations. The transformations that w
1) Filter out missing data.
2) Fill in missing data.
3) Remove duplicates.
Based from the output for met_df above:
There are 477804 rows and 54 columns in from the met CSV file.
Begin by understanding where the missing data is located for met_df.
'''
# Find out which columns have missing values with a count.
met_df.isna().sum()
```

Out[416…
```
Object Number                0
Is Highlight                 0
Is Timeline Work             0
Is Public Domain             0
Object ID                    0
Gallery Number          426028
Department                   0
AccessionYear             3556
Object Name               1691
Title                    29185
Culture                 270425
```

```
Period                     386848
Dynasty                    454571
Reign                      466578
Portfolio                  454274
Constituent ID             202269
Artist Role                204368
Artist Prefix              202269
Artist Display Name        202269
Artist Display Bio         204368
Artist Suffix              202317
Artist Alpha Sort          202269
Artist Nationality         202269
Artist Begin Date          202269
Artist End Date            202269
Artist Gender              374743
Artist ULAN URL            255783
Artist Wikidata URL        260072
Object Date                 13867
Object Begin Date               0
Object End Date                 0
Medium                       7120
Dimensions                  75294
Credit Line                   451
Geography Type             418035
City                       445397
State                      475254
County                     469354
Country                    402053
Region                     446444
Subregion                  455680
Locale                     462095
Locus                      470311
Excavation                 461246
River                      475709
Classification              78206
Rights and Reproduction    453606
Link Resource                   0
Object Wikidata URL        455539
Metadata Date              477804
Repository                      0
Tags                       277404
Tags AAT URL               277404
Tags Wikidata URL          277404
dtype: int64
```

In [417…

```
'''
First transformation step will be to filter out missing data using dropna().
Drop columns from met_df if missing data exceeds 90% of the overall rows (Thresh is 477
'''
met_df.dropna(thresh=430024, axis = 1, inplace = True)
```

In [418…

```
'''
Review the columns that have missing values after the dropna() step.
'''
met_df.isna().sum()
```

Out[418…

```
Object Number           0
Is Highlight            0
```

```
Is Timeline Work         0
Is Public Domain         0
Object ID                0
Department               0
AccessionYear         3556
Object Name           1691
Title                29185
Object Date          13867
Object Begin Date        0
Object End Date          0
Medium                7120
Credit Line            451
Link Resource            0
Repository               0
dtype: int64
```

In [419…

```
'''
Review the shape of met_df after the dropna() step.
'''
print('There are {} rows and {} columns in from the met CSV file.'.format(met_df.shape[
diff_col = 54 - met_df.shape[1]
print('There were {} columns dropped.'.format(diff_col))
```

```
There are 477804 rows and 16 columns in from the met CSV file.
There were 38 columns dropped.
```

In [420…

```
'''
Second transformation step is to fill in missing data with fillna().
First understand the type of data in the remaining columns.
'''
met_df.dtypes
```

Out[420…

```
Object Number        object
Is Highlight           bool
Is Timeline Work       bool
Is Public Domain       bool
Object ID             int64
Department           object
AccessionYear        object
Object Name          object
Title                object
Object Date          object
Object Begin Date     int64
Object End Date       int64
Medium               object
Credit Line          object
Link Resource        object
Repository           object
dtype: object
```

In [421…

```
'''
View the met_df dataframe in its current state.
'''
met_df.head(30)
```

Out[421…

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Department	AccessionYear	Object Name	Title	C
0	1979.486.1	False	False	False	1	The American Wing	1979.0	Coin	One-dollar Liberty Head Coin	
1	1980.264.5	False	False	False	2	The American Wing	1980.0	Coin	Ten-dollar Liberty Head Coin	
2	67.265.9	False	False	False	3	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	
3	67.265.10	False	False	False	4	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	
4	67.265.11	False	False	False	5	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	
5	67.265.12	False	False	False	6	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	
6	67.265.13	False	False	False	7	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	
7	67.265.14	False	False	False	8	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	
8	67.265.15	False	False	False	9	The American Wing	1967.0	Coin	Two-and-a-Half Dollar Coin	

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Department	AccessionYear	Object Name	Title	C
9	1979.486.3	False	False	False	10	The American Wing	1979.0	Coin	Two-and-a-half-dollar Indian Head Coin	
10	1979.486.2	False	False	False	11	The American Wing	1979.0	Coin	Two-and-a-half-dollar Liberty Head Coin	
11	1979.486.7	False	False	False	12	The American Wing	1979.0	Coin	Twenty-dollar Liberty Head Coin	
12	1979.486.4	False	False	False	13	The American Wing	1979.0	Coin	Five-dollar Indian Head Coin	
13	1979.486.5	False	False	False	14	The American Wing	1979.0	Coin	Five-dollar Liberty Head Coin	
14	16.74.49	False	False	False	15	The American Wing	1916.0	Coin	Coin, 1/2 Real	
15	16.74.27	False	False	False	16	The American Wing	1916.0	Peso	Coin, 1/4 Peso	
16	16.74.28	False	False	False	17	The American Wing	1916.0	Peso	Coin, 1/4 Peso	
17	16.74.29	False	False	False	18	The American Wing	1916.0	Peso	Coin, 1/4 Peso	
18	16.74.30	False	False	False	19	The American Wing	1916.0	Peso	Coin, 1/4 Peso	

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Department	AccessionYear	Object Name	Title	C
19	16.74.31	False	False	False	20	The American Wing	1916.0	Peso	Coin, 1/4 Peso	
20	16.74.32	False	False	False	21	The American Wing	1916.0	Peso	Coin, 1/4 Peso	
21	16.74.43	False	False	False	22	The American Wing	1916.0	Coin	Coin, 1/4 Real	
22	16.74.44	False	False	False	23	The American Wing	1916.0	Coin	Coin, 1/4 Real	
23	16.74.33	False	False	False	24	The American Wing	1916.0	Centavos	Coin, 10 Centavos	
24	16.74.34	False	False	False	25	The American Wing	1916.0	Centavos	Coin, 10 Centavos	
25	16.74.35	False	False	False	26	The American Wing	1916.0	Centavos	Coin, 10 Centavos	
26	16.74.36	False	False	False	27	The American Wing	1916.0	Centavos	Coin, 10 Centavos	
27	16.74.38	False	False	False	28	The American Wing	1916.0	Centavos	Coin, 10 Centavos	
28	16.74.39	False	False	False	29	The American Wing	1916.0	Centavos	Coin, 10 Centavos	
29	16.74.37	False	False	False	30	The American Wing	1916.0	Centavos	Coin, 10 Centavos	

In [422…

```
...
```

```
Fill in the missing string values with fillna() and the verbiage "No Data"
'''
met_df.fillna({'Credit Line': "No Data", 'Medium': "No Data", 'Title': "No Data",
               'Object Name': "No Data"}, inplace=True)
```

In [423…

```
'''
Review the coulmns that have missing values after the fillna() step for the string colu
'''
met_df.isna().sum()
```

Out[423…

```
Object Number            0
Is Highlight             0
Is Timeline Work         0
Is Public Domain         0
Object ID                0
Department               0
AccessionYear         3556
Object Name              0
Title                    0
Object Date          13867
Object Begin Date        0
Object End Date          0
Medium                   0
Credit Line              0
Link Resource            0
Repository               0
dtype: int64
```

In [424…

```
'''
Last two columns with NaN values pertains to date data.These columns will be dropped fr
'''
met_df.dropna(axis = 1, inplace=True)
```

In [425…

```
'''
Review the df to show there are no missing values.
'''
met_df.isna().sum()
```

Out[425…

```
Object Number        0
Is Highlight         0
Is Timeline Work     0
Is Public Domain     0
Object ID            0
Department           0
Object Name          0
Title                0
Object Begin Date    0
Object End Date      0
Medium               0
Credit Line          0
Link Resource        0
Repository           0
dtype: int64
```

In [426…

```
'''
The third transformation from Chapter 7 will be removal of duplicates.
```

```
First, check for any duplications within the DataFrame.
'''
met_df.duplicated().any()
```

Out[426…
```
False
```

No duplicates exist for the met_df DataFrame as seen above. If duplicates did exist, then I would have used drop_duplicates() to remove the duplicate data.

In [427…
```
'''
Show the intitial information of met_df after the Chapter 7 data transformations.
'''
met_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 477804 entries, 0 to 477803
Data columns (total 14 columns):
 #   Column             Non-Null Count   Dtype 
---  ------             --------------   ----- 
 0   Object Number      477804 non-null  object
 1   Is Highlight       477804 non-null  bool  
 2   Is Timeline Work   477804 non-null  bool  
 3   Is Public Domain   477804 non-null  bool  
 4   Object ID          477804 non-null  int64 
 5   Department         477804 non-null  object
 6   Object Name        477804 non-null  object
 7   Title              477804 non-null  object
 8   Object Begin Date  477804 non-null  int64 
 9   Object End Date    477804 non-null  int64 
 10  Medium             477804 non-null  object
 11  Credit Line        477804 non-null  object
 12  Link Resource      477804 non-null  object
 13  Repository         477804 non-null  object
dtypes: bool(3), int64(3), object(8)
memory usage: 41.5+ MB
```

In [428…
```
'''
View the first 10 rows of the met_df DataFrame after the Chapter 7 Transformations.
'''
met_df.head(10)
# met_df.tail(10)
```

Out[428…

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Department	Object Name	Title	Object Begin Date	Object End Date	Medi
0	1979.486.1	False	False	False	1	The American Wing	Coin	One-dollar Liberty Head Coin	1853	1853	G
1	1980.264.5	False	False	False	2	The American Wing	Coin	Ten-dollar Liberty Head Coin	1901	1901	G

	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Object ID	Department	Object Name	Title	Object Begin Date	Object End Date	Medi
2	67.265.9	False	False	False	3	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
3	67.265.10	False	False	False	4	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
4	67.265.11	False	False	False	5	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
5	67.265.12	False	False	False	6	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
6	67.265.13	False	False	False	7	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
7	67.265.14	False	False	False	8	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
8	67.265.15	False	False	False	9	The American Wing	Coin	Two-and-a-Half Dollar Coin	1909	1927	G
9	1979.486.3	False	False	False	10	The American Wing	Coin	Two-and-a-half-dollar Indian Head Coin	1912	1912	G

Transformations from Chapter 8

In [429…

```
'''
The transformations that will take place for Chapter 8 will be:
1) Hierarchical Indexing. - met_df dataframe set to have multiple indexes.
```

```
2) Reshape Data. - met_df dataframe will be reshaped and stored under another dataframe
The steps will be commented and labeled throughout the code for clarity.
Initially, understand the unique content within 'Object Name'.
'''
met_df['Object Name'].unique()
```

Out[429…
```
array(['Coin', 'Peso', 'Centavos', ..., 'Book, print, ephemera',
       'Book; prints', 'Ephemera; postcard'], dtype=object)
```

In [430…
```
'''
Understand the unique categories within 'Department'.
Trying to understand if Department would be a good candiate for indexing.
'''
met_df['Department'].unique()
```

Out[430…
```
array(['The American Wing', 'European Sculpture and Decorative Arts',
       'Modern and Contemporary Art', 'Arms and Armor', 'Medieval Art',
       'Asian Art', 'Islamic Art', 'Costume Institute',
       'Arts of Africa, Oceania, and the Americas', 'Drawings and Prints',
       'Greek and Roman Art', 'Photographs', 'Ancient Near Eastern Art',
       'Egyptian Art', 'European Paintings', 'Robert Lehman Collection',
       'The Cloisters', 'Musical Instruments', 'The Libraries'],
      dtype=object)
```

In [431…
```
'''
Use hierarchical indexing for met_df by setting the index based on 'Object Name', 'Depa
'''
met_df.set_index(['Object Name','Department', 'Object ID'], inplace = True)
```

In [432…
```
'''
View the met_df DataFrame to observe the multiple indexes.
'''
met_df.head(50)
```

Out[432…

Object Name	Department	Object ID	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date
Coin	The American Wing	1	1979.486.1	False	False	False	One-dollar Liberty Head Coin	1853
		2	1980.264.5	False	False	False	Ten-dollar Liberty Head Coin	1901
		3	67.265.9	False	False	False	Two-and-a-Half Dollar Coin	1909

Object Name	Department	Object ID	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date
		4	67.265.10	False	False	False	Two-and-a-Half Dollar Coin	1909
		5	67.265.11	False	False	False	Two-and-a-Half Dollar Coin	1909
		6	67.265.12	False	False	False	Two-and-a-Half Dollar Coin	1909
		7	67.265.13	False	False	False	Two-and-a-Half Dollar Coin	1909
		8	67.265.14	False	False	False	Two-and-a-Half Dollar Coin	1909
		9	67.265.15	False	False	False	Two-and-a-Half Dollar Coin	1909
		10	1979.486.3	False	False	False	Two-and-a-half-dollar Indian Head Coin	1912
		11	1979.486.2	False	False	False	Two-and-a-half-dollar Liberty Head Coin	1907
		12	1979.486.7	False	False	False	Twenty-dollar Liberty Head Coin	1876
		13	1979.486.4	False	False	False	Five-dollar Indian Head Coin	1910
		14	1979.486.5	False	False	False	Five-dollar Liberty Head Coin	1907

Object Name	Department	Object ID	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date
		15	16.74.49	False	False	False	Coin, 1/2 Real	1665
Peso	**The American Wing**	**16**	16.74.27	False	False	False	Coin, 1/4 Peso	1800
		17	16.74.28	False	False	False	Coin, 1/4 Peso	1867
		18	16.74.29	False	False	False	Coin, 1/4 Peso	1860
		19	16.74.30	False	False	False	Coin, 1/4 Peso	1859
		20	16.74.31	False	False	False	Coin, 1/4 Peso	1860
		21	16.74.32	False	False	False	Coin, 1/4 Peso	1859
Coin	**The American Wing**	**22**	16.74.43	False	False	False	Coin, 1/4 Real	1881
		23	16.74.44	False	False	False	Coin, 1/4 Real	1878
Centavos	**The American Wing**	**24**	16.74.33	False	False	False	Coin, 10 Centavos	1860
		25	16.74.34	False	False	False	Coin, 10 Centavos	1860

Object Name	Department	Object ID	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date
		26	16.74.35	False	False	False	Coin, 10 Centavos	1860
		27	16.74.36	False	False	False	Coin, 10 Centavos	1860
		28	16.74.38	False	False	False	Coin, 10 Centavos	1860
		29	16.74.39	False	False	False	Coin, 10 Centavos	1860
		30	16.74.37	False	False	False	Coin, 10 Centavos	1885
		31	16.74.40	False	False	False	Coin, 10 Centavos	1885
Pesos	**The American Wing**	**32**	09.9.15	False	False	False	Coin, 20 Pesos	1866
Bust	**The American Wing**	**33**	64.62	False	False	False	Bust of Abraham Lincoln	1876
Clock	**The American Wing**	**34**	1970.289.6	False	False	True	Acorn Clock	1847
Vase	**The American Wing**	**35**	04.1a–c	True	True	False	The Adams Vase	1893
Side Chair	**The American Wing**	**36**	1976.319	False	False	False	Side Chair	1884

Object Name	Department	Object ID	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date
Figure	**The American Wing**	**37**	38.165.51	False	False	True	Figure of Admiral George Rodney	1782
		38	38.165.50	False	False	True	Figure of Admiral Samuel Hood	1782
Advertisement	**The American Wing**	**39**	18.11.10	False	False	True	Advertisement for Norwich Stone Ware Factory	1770
Ale glass	**The American Wing**	**40**	46.140.143	False	False	True	Ale Glass	1830
		41	46.140.864	False	False	True	Ale Glass	1850
Andiron	**The American Wing**	**42**	60.58.1	False	False	True	Andiron	1795
		43	60.58.2	False	False	True	Andiron	1795
		44	10.125.444a	False	False	True	Andiron	1787
		45	10.125.444b	False	False	True	Andiron	1787
		46	10.125.445a	False	False	True	Andiron	1700
		47	10.125.445b	False	False	True	Andiron	1700

Object Name	Department	Object ID	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date
		48	10.125.446a	False	False	False	Andiron	1770
		49	10.125.446b	False	False	False	Andiron	1770
		50	10.125.447a	False	False	True	Andiron	1770

In [433…

```
'''
Reshape the met_df data with stack to pivot the columns of the data into rows.
Store the newly shaped data under a new DataFrame called met_df_shape.
'''
met_df_shape = met_df.stack()
met_df_shape.head(50)
```

Out[433…

```
Object Name  Department         Object ID
Coin         The American Wing  1          Object Number
1979.486.1

                                           Is Highlight
False

                                           Is Timeline Work
False

                                           Is Public Domain
False

                                           Title                                One-
dollar Liberty Head Coin

                                           Object Begin Date
1853

                                           Object End Date
1853

                                           Medium
Gold

                                           Credit Line                        Gift of He
inz L. Stoppelmann, 1979

                                           Link Resource        http://www.metmuseum.or
g/art/collection/search/1

                                           Repository                   Metropolitan Mus
eum of Art, New York, NY

                                  2        Object Number
1980.264.5

                                           Is Highlight
False

                                           Is Timeline Work
False
```

False

dollar Liberty Head Coin

1901

1901

Gold

inz L. Stoppelmann, 1980

g/art/collection/search/2

eum of Art, New York, NY

3

67.265.9

False

False

False

o-and-a-Half Dollar Coin

1909

1927

Gold

C. Ruxton Love Jr., 1967

g/art/collection/search/3

eum of Art, New York, NY

4

67.265.10

False

False

False

o-and-a-Half Dollar Coin

1909

1927

Gold

C. Ruxton Love Jr., 1967

g/art/collection/search/4

eum of Art, New York, NY

Is Public Domain

Title Ten-

Object Begin Date

Object End Date

Medium

Credit Line Gift of He

Link Resource http://www.metmuseum.or

Repository Metropolitan Mus

Object Number

Is Highlight

Is Timeline Work

Is Public Domain

Title Tw

Object Begin Date

Object End Date

Medium

Credit Line Gift of

Link Resource http://www.metmuseum.or

Repository Metropolitan Mus

Object Number

Is Highlight

Is Timeline Work

Is Public Domain

Title Tw

Object Begin Date

Object End Date

Medium

Credit Line Gift of

Link Resource http://www.metmuseum.or

Repository Metropolitan Mus

```
                                5          Object Number
67.265.11
                                           Is Highlight
False
                                           Is Timeline Work
False
                                           Is Public Domain
False
                                           Title                                    Tw
o-and-a-Half Dollar Coin
                                           Object Begin Date
1909
dtype: object
```

Transformations from Chapter 10

In [434…

```
'''
The Met data will be considered again for Chapter 10 Transformations. The transformatio
1) Grouping by Index Levels.
2) Cross-Tabulation
Start with Grouping by Index Levels with 'Object Name' using .groupby().
'''
met_df.groupby(level='Object Name', axis = 0).count()
```

Out[434…

Object Name	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date	Object End Date	Medium	Credit Line	Link Resource
"Autophone" Organette	1	1	1	1	1	1	1	1	1	1
"Basso"	1	1	1	1	1	1	1	1	1	1
"Chanot Model" Violin	1	1	1	1	1	1	1	1	1	1
"Humantone" Nose Flute	1	1	1	1	1	1	1	1	1	1
"Japanese Fiddle"	1	1	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...
Śrnga	1	1	1	1	1	1	1	1	1	1
Ūd	8	8	8	8	8	8	8	8	8	8
Ūd (converted to mandora)	1	1	1	1	1	1	1	1	1	1
Ṣanāṣel (sistrum)	1	1	1	1	1	1	1	1	1	1
Ṭāśā	1	1	1	1	1	1	1	1	1	1

28450 rows × 11 columns

In [435…

```
'''
Next, try Grouping by Index Levels with 'Department' using .groupby().
Use .count() to understand the number of observations for each category across the colu
'''
met_df.groupby(level='Department', axis = 0).count()
```

Out[435…

Department	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date	Object End Date	Medium	Credit Line	Reso
Ancient Near Eastern Art	6222	6222	6222	6222	6222	6222	6222	6222	6222	
Arms and Armor	13611	13611	13611	13611	13611	13611	13611	13611	13611	1
Arts of Africa, Oceania, and the Americas	12335	12335	12335	12335	12335	12335	12335	12335	12335	1
Asian Art	36817	36817	36817	36817	36817	36817	36817	36817	36817	3
Costume Institute	31412	31412	31412	31412	31412	31412	31412	31412	31412	3
Drawings and Prints	167152	167152	167152	167152	167152	167152	167152	167152	167152	16
Egyptian Art	27962	27962	27962	27962	27962	27962	27962	27962	27962	2
European Paintings	2617	2617	2617	2617	2617	2617	2617	2617	2617	
European Sculpture and Decorative Arts	42934	42934	42934	42934	42934	42934	42934	42934	42934	4
Greek and Roman Art	33750	33750	33750	33750	33750	33750	33750	33750	33750	3
Islamic Art	15473	15473	15473	15473	15473	15473	15473	15473	15473	1
Medieval Art	7150	7150	7150	7150	7150	7150	7150	7150	7150	
Modern and Contemporary Art	14241	14241	14241	14241	14241	14241	14241	14241	14241	1
Musical Instruments	5210	5210	5210	5210	5210	5210	5210	5210	5210	
Photographs	37192	37192	37192	37192	37192	37192	37192	37192	37192	3
Robert Lehman Collection	2586	2586	2586	2586	2586	2586	2586	2586	2586	
The American Wing	18412	18412	18412	18412	18412	18412	18412	18412	18412	1

Department	Object Number	Is Highlight	Is Timeline Work	Is Public Domain	Title	Object Begin Date	Object End Date	Medium	Credit Line	Reso
The Cloisters	2338	2338	2338	2338	2338	2338	2338	2338	2338	
The Libraries	390	390	390	390	390	390	390	390	390	

In [436…

```
'''
Perform Cross-Tabulation with pd.crosstab() function.
Cross-Tabulate the Object Number with Is Public Domain to identify True or False
'''
cross_tabs = pd.crosstab(met_df['Object Number'], met_df['Is Public Domain'], margins =
cross_tabs.head(30)
```

Out[436…

Object Number	Is Public Domain	False	True	All
00.1.1		0	1	1
00.1.10		0	1	1
00.1.11		0	1	1
00.1.14		0	1	1
00.1.15		0	1	1
00.1.18		0	1	1
00.1.2		0	1	1
00.1.20		0	1	1
00.1.21		1	0	1
00.1.22		0	1	1
00.1.23		1	0	1
00.1.3		0	1	1
00.1.4		0	1	1
00.1.5		0	1	1
00.1.6		0	1	1
00.1.8		0	1	1
00.1.9		0	1	1
00.10		0	1	1
00.11.1		0	1	1
00.12.1		0	1	1
00.12.10		0	1	1

Is Public Domain	False	True	All
Object Number			
00.12.11	0	1	1
00.12.12	0	1	1
00.12.13	1	0	1
00.12.14	1	0	1
00.12.15	1	0	1
00.12.16	1	0	1
00.12.17	0	1	1
00.12.18	1	0	1
00.12.19	0	1	1

Transformations from Chapter 11

In [437…

```
'''
The Candy Hierarchy data will be considered  for Chapter 11 Transformations. The transf
1) Convert between string and date time. - 2015 and 2016 data set timestamps
2) Convert timestamps to periods and back. - 2015 and 2016 data set timestamps
Start by importing datetime from datetime module.
'''
from datetime import datetime
```

In [438…

```
'''
Verify the current data type for 'Timestamp' is datetime64 as shown below for 2016 data
'''
candy_df2016['Timestamp']
```

Out[438…

```
0      2016-10-24 05:09:23.033
1      2016-10-24 05:09:54.798
2      2016-10-24 05:13:06.734
3      2016-10-24 05:14:17.192
4      2016-10-24 05:14:24.625
                  ...
1254   2016-10-29 16:53:52.516
1255   2016-10-30 06:53:54.735
1256   2016-10-30 11:06:10.827
1257   2016-10-30 16:07:26.539
1258   2016-10-30 17:06:45.660
Name: Timestamp, Length: 1259, dtype: datetime64[ns]
```

In [439…

```
'''
Convert the candy_df2016['Timestamp'] column to a string data type.
'''
candy_df2016['Timestamp'] = candy_df2016['Timestamp'].dt.strftime('%Y-%m-%d')
```

In [440…

```
'''
View the conversion from datetime to string by showing the first 5 columns of candy_df2
```

```
'''
candy_df2016['Timestamp'].head(5)
```

Out[440…
```
0    2016-10-24
1    2016-10-24
2    2016-10-24
3    2016-10-24
4    2016-10-24
Name: Timestamp, dtype: object
```

In [441…
```
'''
Convert the candy_df2015['Timestamp'] column to a string data type.
'''
candy_df2015['Timestamp'] = candy_df2015['Timestamp'].dt.strftime('%Y-%m-%d')
```

In [442…
```
'''
View the conversion from DateTime to String by showing the first 5 columns of candy_df2
'''
candy_df2015['Timestamp'].head(5)
```

Out[442…
```
0    2015-10-23
1    2015-10-23
2    2015-10-23
3    2015-10-23
4    2015-10-23
Name: Timestamp, dtype: object
```

In [443…
```
'''
Typically, wouldn't convert the 'Timestamp' column again. However, I'll convert candy_d
back to DateTime data type.
'''
candy_df2016['Timestamp'] = pd.to_datetime(candy_df2016['Timestamp'],format='%Y-%m-%d')
```

In [444…
```
'''
View the candy_df2016['Timestamp'] datatype to confirm datetime.
'''
candy_df2016['Timestamp'].head(5)
```

Out[444…
```
0   2016-10-24
1   2016-10-24
2   2016-10-24
3   2016-10-24
4   2016-10-24
Name: Timestamp, dtype: datetime64[ns]
```

In [445…
```
'''
Perform a similar step for candy_df2015['Timestamp'] to convert back to datetime.
'''
candy_df2015['Timestamp'] = pd.to_datetime(candy_df2015['Timestamp'],format='%Y-%m-%d')
```

In [446…
```
'''
View the candy_df2015['Timestamp'] datatype to confirm datetime.
'''
candy_df2015['Timestamp'].head(5)
```

Out[446…
```
0   2015-10-23
1   2015-10-23
2   2015-10-23
3   2015-10-23
4   2015-10-23
Name: Timestamp, dtype: datetime64[ns]
```

In [447…
```
'''
Convert the candy_df2016['Timestamp'] data to a revised period (monthly).
'''
candy_df2016['Timestamp'] = candy_df2016['Timestamp'].dt.to_period(freq ='M')
```

In [448…
```
'''
View the candy_df2016['Timestamp'] column to show the data is now based on months.
The datatype should also show as period.
'''
candy_df2016['Timestamp'].head(5)
```

Out[448…
```
0    2016-10
1    2016-10
2    2016-10
3    2016-10
4    2016-10
Name: Timestamp, dtype: period[M]
```

In [449…
```
'''
Convert the candy_df2015['Timestamp'] data to a revised period (monthly).
'''
candy_df2015['Timestamp'] = candy_df2015['Timestamp'].dt.to_period(freq ='M')
```

In [450…
```
'''
View the candy_df2015['Timestamp'] column to show the data is now based on months.
The datatype should also show as period.
'''
candy_df2015['Timestamp'].head(5)
```

Out[450…
```
0    2015-10
1    2015-10
2    2015-10
3    2015-10
4    2015-10
Name: Timestamp, dtype: period[M]
```

In [451…
```
'''
Convert candy_df2016['Timestamp'] back to DateTime data type.
'''
candy_df2016['Timestamp'] = candy_df2016['Timestamp'].dt.to_timestamp(freq = 'M')
```

In [452…
```
'''
View the candy_df2016['Timestamp'] datatype to confirm datetime.
Note that default will be month end date.
'''
candy_df2016['Timestamp'].head(5)
```

Out[452…
```
0   2016-10-31
1   2016-10-31
2   2016-10-31
3   2016-10-31
4   2016-10-31
Name: Timestamp, dtype: datetime64[ns]
```

In [453…
```
'''
Convert candy_df2015['Timestamp'] back to DateTime data type.
'''
candy_df2015['Timestamp'] = candy_df2015['Timestamp'].dt.to_timestamp(freq = 'M')
```

In [454…
```
'''
View the candy_df2015['Timestamp'] datatype to confirm datetime.
Note that the default will be month end date.
'''
candy_df2015['Timestamp'].head(5)
```

Out[454…
```
0   2015-10-31
1   2015-10-31
2   2015-10-31
3   2015-10-31
4   2015-10-31
Name: Timestamp, dtype: datetime64[ns]
```