



Landscapes

At the Metropolitan Museum of Art

Amanda Anderson-You



Background

- Multiple concepts before landing here
- Inspired by scope of landscape keywords in CSV
- Worked with Pandas to format the dataset
- Used plotly.js for visualizing, which is D3.js based



Process

- Reviewed CSV and filtered by keyword “landscapes”
 - Includes objects where landscape is included anywhere in the list of tags, not just as a primary tag
- Reviewed departments and chose to include only departments with over 25 objects, keeping 8 of 16 departments.
- Used object end date for year created, decided to limit to years AD and remove 13 records (of almost 5K total) of years BC

Met_QuantVisual.csv
Met_StructureData.py

```

1 # import csv
2 # input_file = csv.DictReader(open("MetQuant_GoodData.csv"))
3 # for row in input_file:
4 #     print(row)
5
6 # import csv
7 # import collections
8 # import matplotlib.pyplot as plt
9 # import pandas as pd
10
11
12 with open('Departments/American_DecoArts.csv') as csvfile:
13     readCSV = csv.reader(csvfile, delimiter=',')
14     years = []
15     for row in readCSV:
16         year = row[1]
17         years.append(year)
18
19     # print(years)
20
21
22 dictionary = (collections.Counter(years))
23 # print(dictionary)
24
25 keys = dictionary.keys()
26 values = dictionary.values()
27
28 # print(keys)
29 # print(values)
30
31 keys = [int(i) for i in keys]
32 values = [int(i) for i in values]
33 print(keys)
34 print(values)
35
36
37 x = [keys]
38 y = [values]
39 plt.plot(x,y)
40 plt.show()
41
42
43
44
45
46
47

```

with open('Departments/American...') for row in readCSV

Met_StructureData

```

/usr/local/bin/python3.7 /Users/a.andersonyou/Documents/School/Parsons/MajorStudio_1/Met_StructureData.py
1704, 1727, 1730, 1735, 1758, 1760, 1761, 1776, 1778, 1790, 1795, 1796, 1798, 1800, 1806, 1810, 1815, 1816, 1820, 1823, 1825, 1827, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868, 1869, 1870, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879, 1880, 1881, 1882, 1883, 1884, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1892, 1893, 1894, 1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911, 1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950

```

Met_PythonVis.py
Met_StructureData.py
testADA.json

```

1 # import csv
2 # input_file = csv.DictReader(open("MetQuant_GoodData.csv"))
3 # for row in input_file:
4 #     print(row)
5
6 # import csv
7 # import collections
8 # import matplotlib.pyplot as plt
9 # import pandas as pd
10 # import numpy as np
11
12 # Use pandas to read the .csv file with approximately 5k object records of landscape art at the Met.
13 # Columns include department and end date year, I'll later use counter to record the quantity of pieces
14 # per year per department.
15 data = pd.read_csv('MetQuant_GoodData.csv', delimiter=',')
16 # create variables that hold individual tables of data for each of the 8 total departments.
17 american_decoarts = data[data.Department == 'American Decorative Arts']
18 asian_arts = data[data.Department == 'Asian Art']
19
20 american_decoarts = american_decoarts.groupby('Year').count()
21 asian_arts = asian_arts.groupby('Year').count()
22
23 print(asian_arts)
24
25 # american_decoarts.plot.line()
26 asian_arts.plot.line()
27 plt.show()
28
29
30
31
32

```

Run: Met_PythonVis

```

/usr/local/bin/python3.7 /Users/a.andersonyou/Documents/School/Parsons/MajorStudio_1/Departments/Met_PythonVis.py
Department
Year
999 1
1060 1
1090 1
1133 1
1159 1
...
1997 9
1999 3
2008 1
2009 1
2015 1

```

[266 rows x 1 columns]

Work in Progress...

	Department	Year
0	American Decorative Arts	1704
1	American Decorative Arts	1727
2	American Decorative Arts	1730
3	American Decorative Arts	1730
4	American Decorative Arts	1730
..
757	American Decorative Arts	1941
758	American Decorative Arts	1944
759	American Decorative Arts	1957
760	American Decorative Arts	1962
761	American Decorative Arts	1969

[762 rows x 2 columns]

```
# when department == "department name"
df['count'] = df.groupby('Year')['Department'].transform('count')
|
```

```
/usr/local/bin/python3.7 /Users/a.andersonyou/D
Department Year count
0 American Decorative Arts 1704 2
1 American Decorative Arts 1727 2
2 American Decorative Arts 1730 10
3 American Decorative Arts 1730 10
4 American Decorative Arts 1730 10
...
4861 Robert Lehman Collection 1958 6
4862 Robert Lehman Collection 1963 3
4863 Robert Lehman Collection 1964 4
4864 Robert Lehman Collection 1965 3
4865 Robert Lehman Collection 1969 19
```

[4866 rows x 3 columns]

Process finished with exit code 0

```
data = pd.read_csv('MetQuant_GoodData.csv', delimiter=',', names=['Department', 'Year', 'Count'])
# create variables that hold individual tables of data for each of the 8 total departments.
american_decoarts = data[data.Department == 'American Decorative Arts']
year_counts = pd.value_counts(american_decoarts.values.ravel())
# year_counts = year_counts[1:]
# print(type(year_counts))
df = pd.DataFrame([year_counts])
print(df)
```

	American Decorative Arts	1870	1890	1904	...	1796	1842	1941	1921
0	762	47	46	39	...	1	1	1	1

[1 rows x 128 columns]

Work in Progress...


```

6 import pandas as pd
7 import plotly.express as px
8 import plotly.graph_objects as go
9
10 app = dash.Dash(__name__)
11
12 # Use pandas to read the .csv file with approximately 5k object records of landscape art at the Met.
13 # Existing csv columns used include departments & end date year. CSV contains additional object level information in
14 # case needed.
15 df = pd.read_csv('MetLandscape_Details.csv', delimiter=',')
16
17 # Create a tidy Pandas DataFrame from the csv file.
18 # Use reset_index to make an Artworks column totaling the unique instances of both department and year.
19 df = df.groupby(['Department', 'Year']).size().reset_index(name='Artworks')
20 # print(df)
21
22 # Using vectors to plot all points at once
23 # Plotly uses D3, the user is able to hover over each point for additional detail
24

```

Structuring the Data

```

25 # Plotly Scatter Plot:
26 fig = px.scatter(df, x="Year", y="Artworks", color="Department")
27 fig.update_layout(
28     title='Landscapes at the Metropolitan Museum of Art',
29     font=dict(
30         family="Times, monospace",
31         size=20,
32         color="#7f7f7f"
33     )
34 )
35 fig.update_layout(
36     xaxis=go.layout.XAxis(
37         title=go.layout.xaxis.Title(
38             text="Years of Artwork Creation",
39             font=dict(
40                 family="Times, monospace",
41                 size=16,
42                 color="#7f7f7f"
43             )
44         ),
45     ),
46     yaxis=go.layout.YAxis(
47         title=go.layout.yaxis.Title(
48             text="Artwork Quantity",
49             font=dict(
50                 family="Times, monospace",
51                 size=16,
52                 color="#7f7f7f"
53             )
54         ),
55     )
56 )
57 )
58 fig.update_layout(
59     legend=go.layout.Legend(
60         traceorder="normal",
61         font=dict(
62             family="Times, monospace",
63             size=14,
64             color="#7f7f7f"
65         ),
66     ),
67 )
68 )
69 )
70 )
71 fig.show()

```

```

72 # Plotly Line Plot:
73 fig = px.line(df, x='Year', y='Artworks', color='Department')
74 fig.update_layout(
75     title='Landscapes at the Metropolitan Museum of Art',
76     font=dict(
77         family="Times, monospace",
78         size=20,
79         color="#7f7f7f"
80     )
81 )
82 )
83 )
84 )
85 )
86 fig.update_layout(
87     xaxis=go.layout.XAxis(
88         title=go.layout.xaxis.Title(
89             text="Years of Artwork Creation",
90             font=dict(
91                 family="Times, monospace",
92                 size=16,
93                 color="#7f7f7f"
94             )
95         ),
96     ),
97     yaxis=go.layout.YAxis(
98         title=go.layout.yaxis.Title(
99             text="Artwork Quantity",
100             font=dict(
101                 family="Times, monospace",
102                 size=16,
103                 color="#7f7f7f"
104             )
105         ),
106     )
107 )
108 )
109 fig.update_layout(
110     legend=go.layout.Legend(
111         traceorder="normal",
112         font=dict(
113             family="Times, monospace",
114             size=14,
115             color="#7f7f7f"
116         ),
117     ),
118 )
119 )
120 fig.show()
121 )
122 )

```

Plotting the Scatter and Line Graphs



Click to view and interact: **Scatter & Line Plots**

Tips

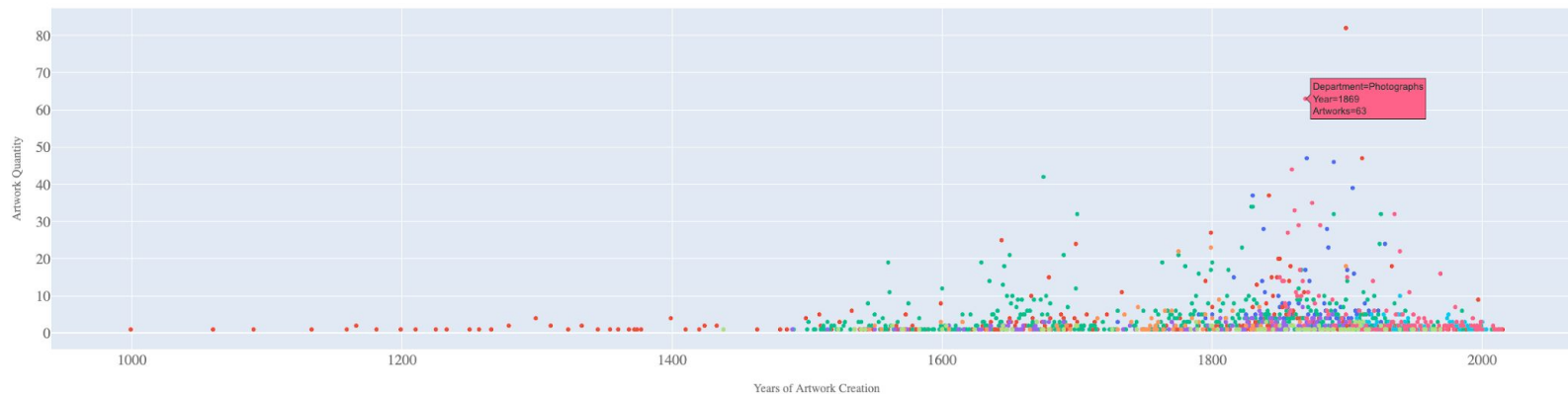
Hover over points to see more information

Select specific departments to turn them on/off

Landscapes at the Metropolitan Museum of Art

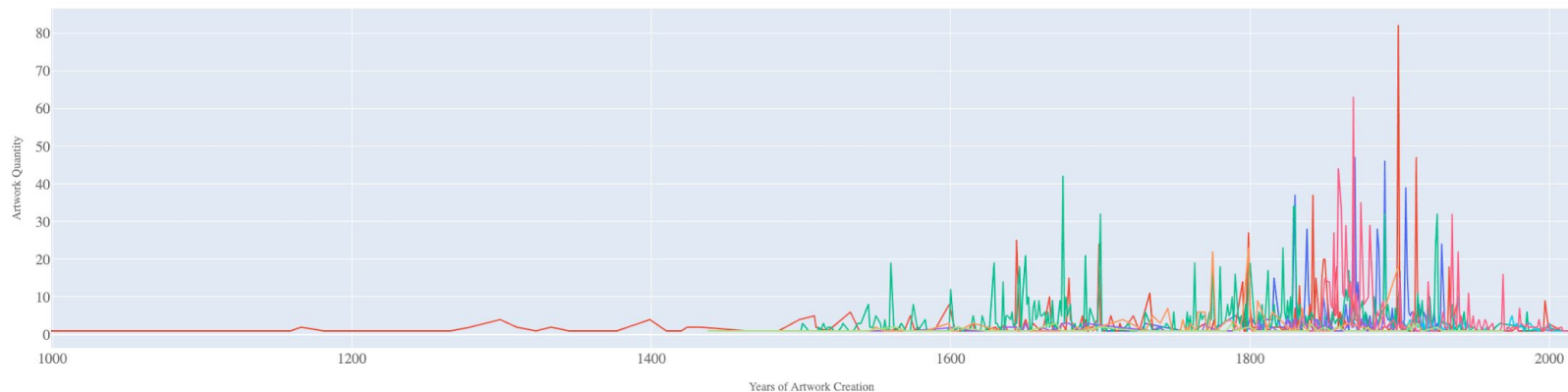


- Departments=American Decorative Arts
- Departments=Asian Art
- Departments=Drawings and Prints
- Departments-European Paintings
- Departments-European Sculpture and Decorative Arts
- Departments-Modern and Contemporary Art
- Departments=Photographs
- Departments=Robert Lehman Collection



Landscapes at the Metropolitan Museum of Art

- Departments=American Decorative Arts
- Departments=Asian Art
- Departments=Drawings and Prints
- Departments-European Paintings
- Departments-European Sculpture and Decorative Arts
- Departments-Modern and Contemporary Art
- Departments=Photographs
- Departments=Robert Lehman Collection



Thank
you!



Photo by Ines Castellano