COMP4447 Final Project

Peter Strimbu

May 16, 2022

METROPOLITAN MUSEUM OF ART DATA ANALYSIS

COLUMNS

1. Is Highlight
2. Object ID
3. AccessionYear
4. Culture
5. Reign
6. Artist Role
7. Artist Display Bio
8. Artist Nationality
9. Artist Gender
10. Object Date
11. Medium
12. Geography Type
13. County
14. Subregion
15. Excavation
16. Rights and Reproduction
17. Metadata Date
18. Tags AAT URL

COLUMN TYPES / RANGE / CLEANING TASKS

OBJECT ID

- int64
- 1 through 860873
- No cleanup, ok as-is

ACCESSIONYEAR

- 4 digit year
- Ex: nan, 1889.0, 1928, 19171917.0, 1956-08-24
- Clean: keep only first 4 digits will fix all 4 issues

CULTURE

- category
- Ex: nan, 'reign of Amenhotep III', 'reign of Xerxes I'
- remove duplicate spaces
- possibly remove 'reign of', 'reigns of', 'or later', 'possibly', 'or slightly later', ',early', ',probably', (anything after comma), (anything in parens), (question marks), (space at end), (split on slash,dash/'and'/'or' and process individually), 'to xyz' and group by remaining names

ARTIST ROLE

- object
- Ex: 'nan', 'Maker', 'Designer|Manufacturer'
- Clean: split on vert bar?

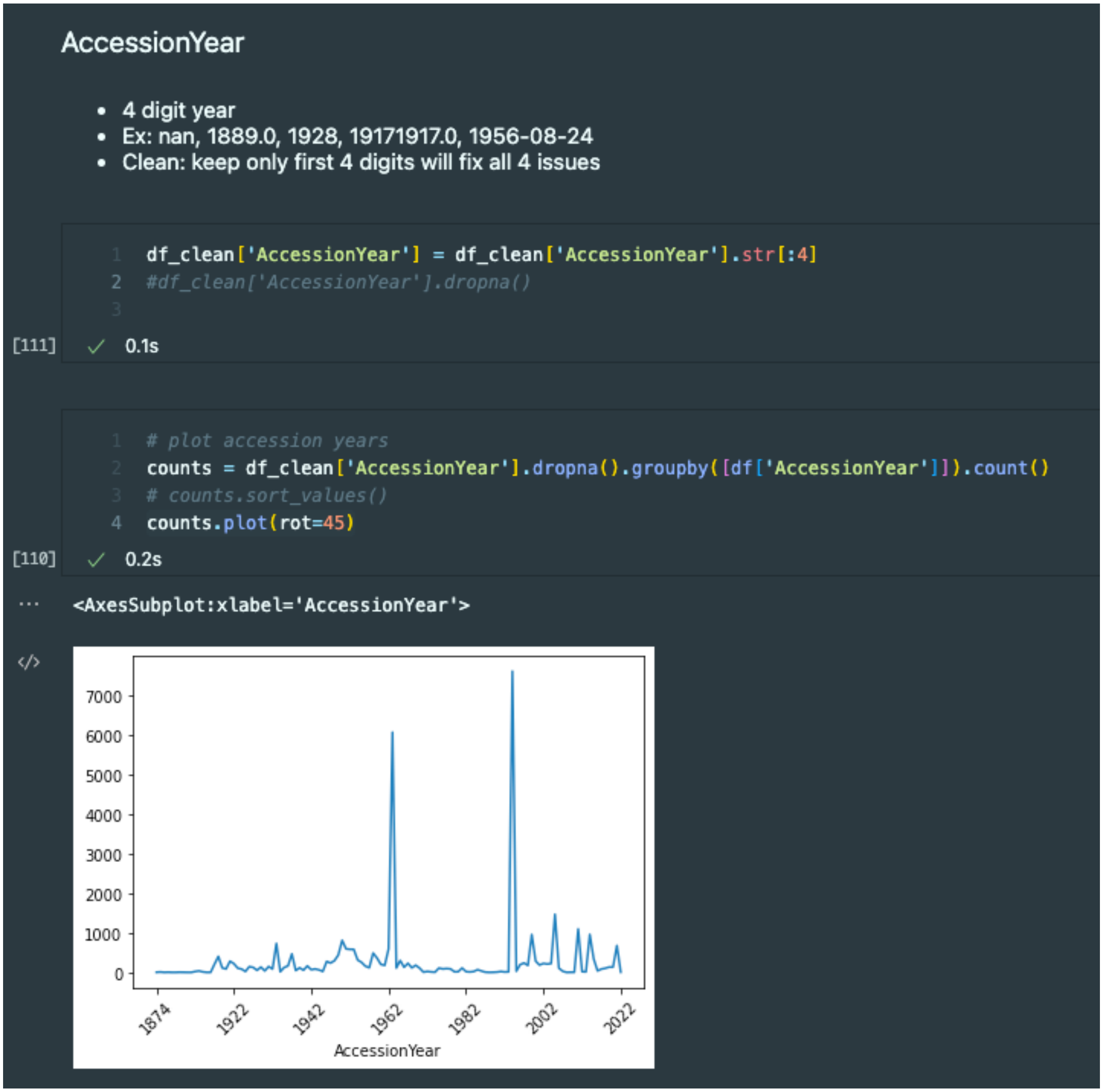
ARTIST DISPLAY BIO

- object
- Ex: '1785–1844', 'nan', 'British, London 1873–1952 Hailsham, Sussex|British, Wiltshire 1877–1952 Oxford', '||||Female|Female'
- Issues: not sure what the vert bars separate
- None - don't use or use as-is

OBJECT DATE

- Numeric
- Ex: '1853', '1901', '1909–27', '1782-1784', 'December 1, 1925', 'after 1773'
- Medium
- object
- Ex: 'Gold' 'Silver' 'Bronze or copper' ... 'Overlay for 23.112.2889, graphite and ink on glazed linen tracing paper'
- Don't use or use as-is

ACCESSIONYEAR



AccessionYear

- 4 digit year
- Ex: nan, 1889.0, 1928, 19171917.0, 1956-08-24
- Clean: keep only first 4 digits will fix all 4 issues

```
df_clean['AccessionYear'] = df_clean['AccessionYear'].str[:4]
#df_clean['AccessionYear'].dropna()

```

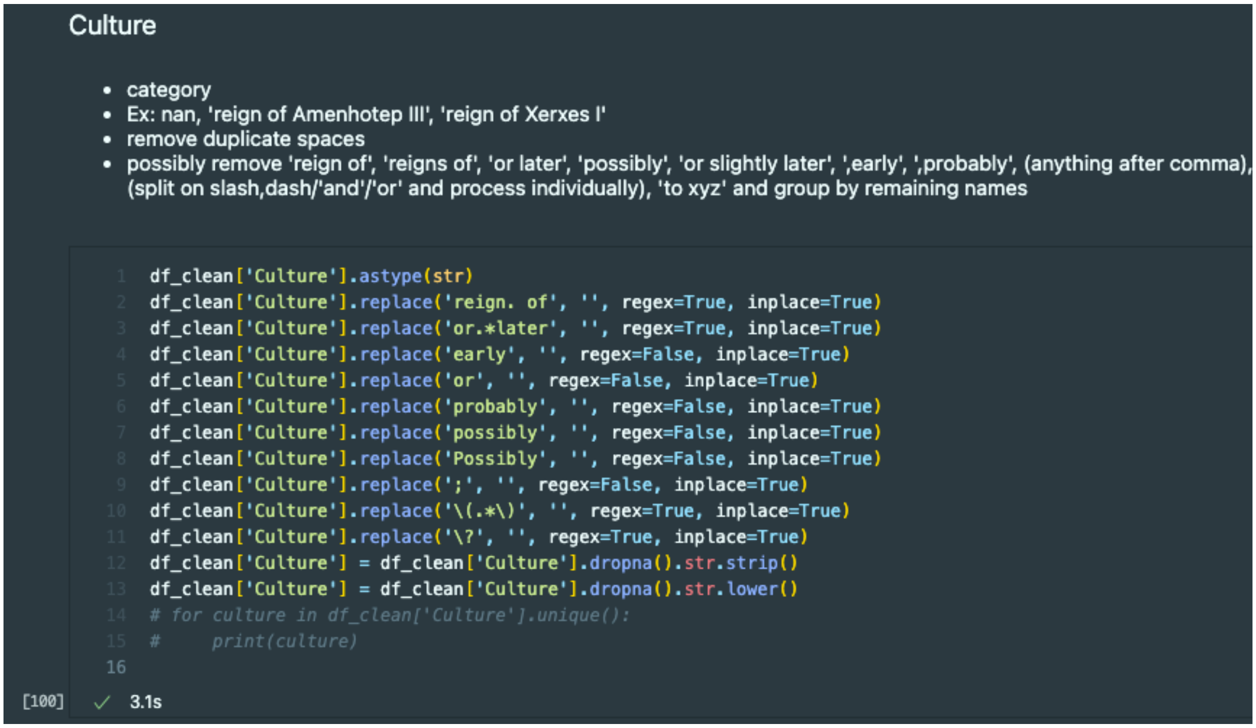
[111] ✓ 0.1s

```
# plot accession years
counts = df_clean['AccessionYear'].dropna().groupby([df['AccessionYear']]).count()
# counts.sort_values()
counts.plot(rot=45)
```

[110] ✓ 0.2s

```
<AxesSubplot:xlabel='AccessionYear'>
```

CULTURE

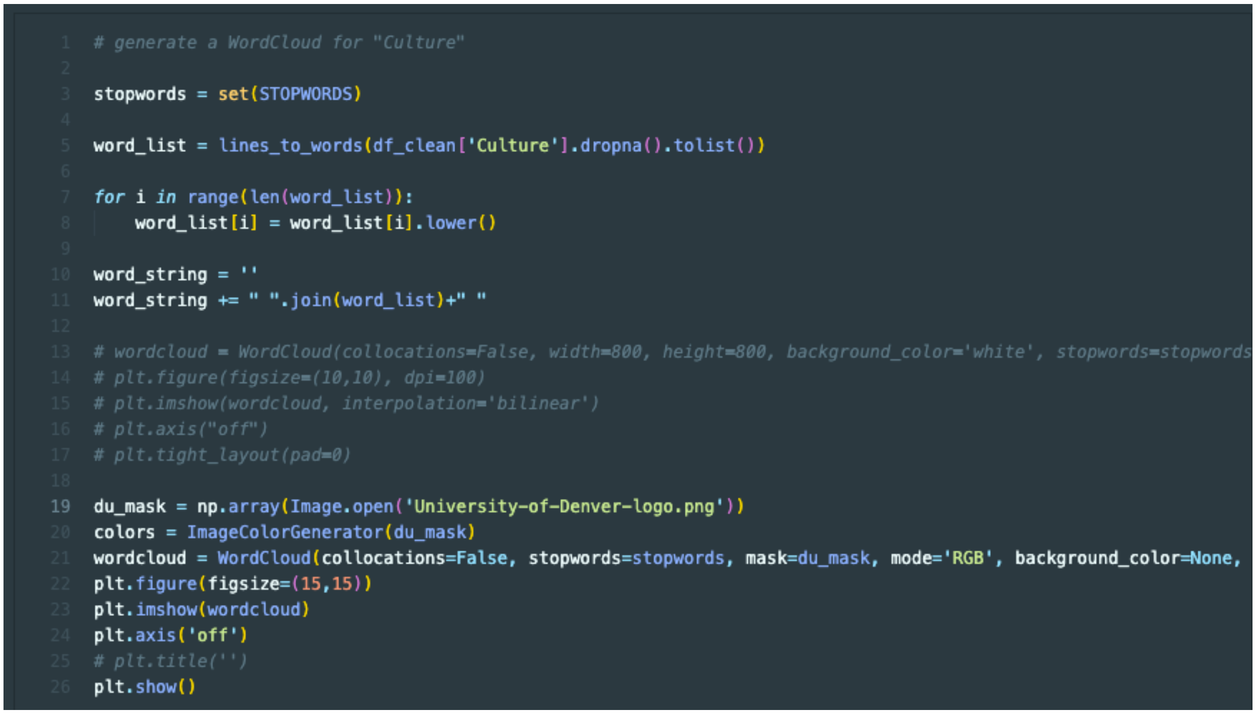


Culture

- category
- Ex: nan, 'reign of Amenhotep III', 'reign of Xerxes I'
- remove duplicate spaces
- possibly remove 'reign of', 'reigns of', 'or later', 'possibly', 'or slightly later', ',early', ',probably', (anything after comma), (split on slash,dash/'and'/'or' and process individually), 'to xyz' and group by remaining names

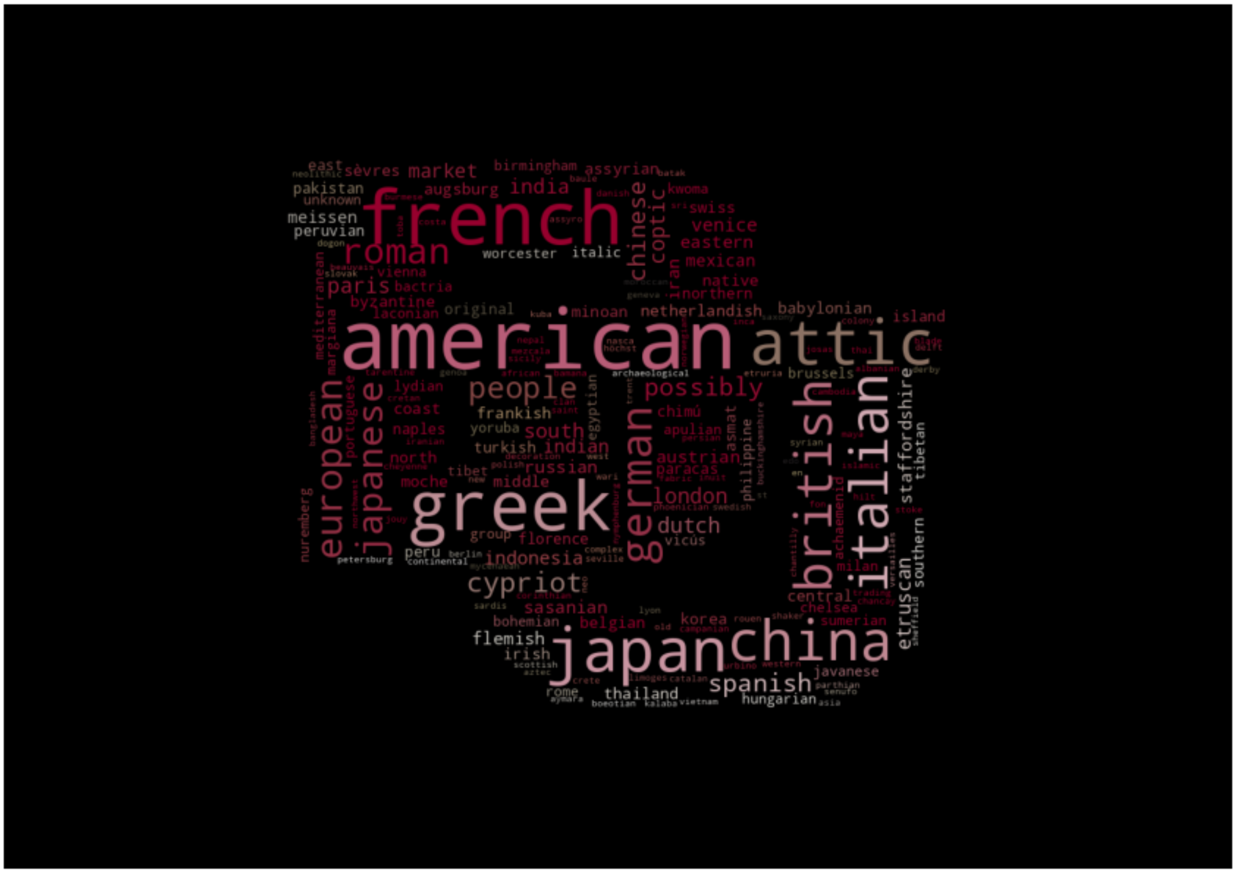
```python
df_clean['Culture'].astype(str)
df_clean['Culture'].replace('reign. of', '', regex=True, inplace=True)
df_clean['Culture'].replace('or.*later', '', regex=True, inplace=True)
df_clean['Culture'].replace('early', '', regex=False, inplace=True)
df_clean['Culture'].replace('or', '', regex=False, inplace=True)
df_clean['Culture'].replace('probably', '', regex=False, inplace=True)
df_clean['Culture'].replace('possibly', '', regex=False, inplace=True)
df_clean['Culture'].replace('Possibly', '', regex=False, inplace=True)
df_clean['Culture'].replace(';', '', regex=False, inplace=True)
df_clean['Culture'].replace('\(.*\)', '', regex=True, inplace=True)
df_clean['Culture'].replace('\?', '', regex=True, inplace=True)
df_clean['Culture'] = df_clean['Culture'].dropna().str.strip()
df_clean['Culture'] = df_clean['Culture'].dropna().str.lower()
# for culture in df_clean['Culture'].unique():
#     print(culture)

```

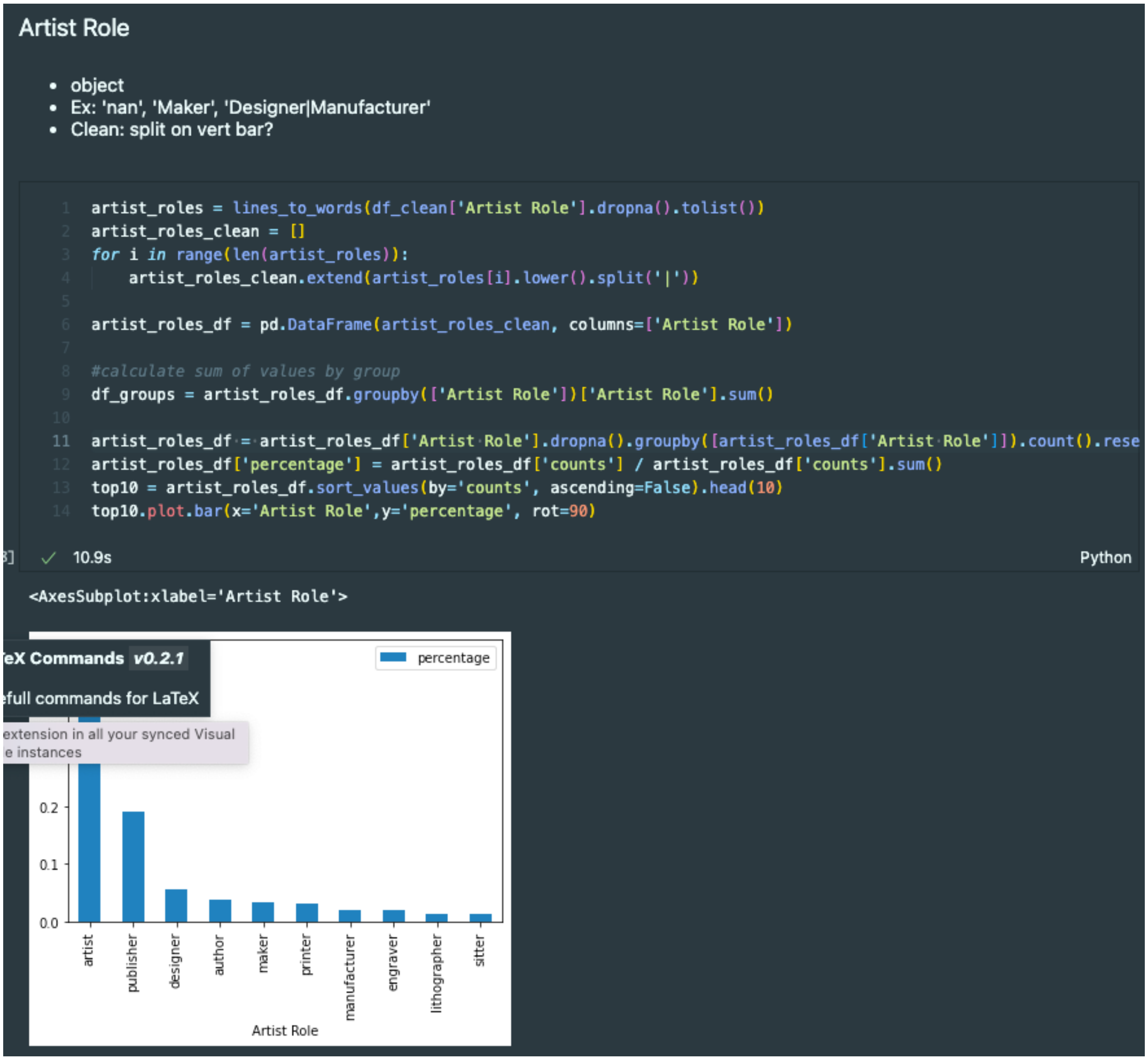
[100] ✓ 3.1s



```python
# generate a WordCloud for "Culture"

stopwords = set(STOPWORDS)

word_list = lines_to_words(df_clean['Culture'].dropna().tolist())

for i in range(len(word_list)):
    word_list[i] = word_list[i].lower()

word_string = ''
word_string += " ".join(word_list)+" "

# wordcloud = WordCloud(collocations=False, width=800, height=800, background_color='white', stopwords=stopwords
# plt.figure(figsize=(10,10), dpi=100)
# plt.imshow(wordcloud, interpolation='bilinear')
# plt.axis("off")
# plt.tight_layout(pad=0)

du_mask = np.array(Image.open('University-of-Denver-logo.png'))
colors = ImageColorGenerator(du_mask)
wordcloud = WordCloud(collocations=False, stopwords=stopwords, mask=du_mask, mode='RGB', background_color=None,
plt.figure(figsize=(15,15))
plt.imshow(wordcloud)
plt.axis('off')
# plt.title('')
plt.show()
```

MASK:


east
neolithic
sèvres
market
birmingham
assyrian
batak
baule
pakistan
unknown
augsburg
india
danish
kwoma
burmese
meissen
peruvian
french
costa
toba
assyro
chinese
coptic
sri
swiss
venice
eastern
dogon
roman
worcester
italic
mexican
iran
moroccan
native
geneva
northern
beauvais
slovak
vienna
paris
bactria
mediterranean
margiana
byzantine
laconian
original
kuba
minoan
netherlandish
babylonian
island
saxony
inca
colony
american
attic
nepal
mezcala
sicily
nasca
höchst
norwegian
josas
thai
blade
delft
tarentine
genoa
african
bamana
archaeological
etruria
brussels
albanian
derby
european
japanese
portuguese
lydian
cretan
people
egyptian
trent
possibly
cambodia
british
italian
staffordshire
tibetan
bangladesh
coast
clan
saint
frankish
chimú
asmat
philippine
buckinghamshire
naples
iranian
yoruba
south
apulian
persian
maya
syrian
turkish
indian
west
german
decoration
north
cheyenne
polish
russian
austrian
paracas
edo
islamic
tibet
new
middle
wari
fabric
inuit
en
moche
nuremberg
northwest
greek
nymphenburg
london
phoenician
swedish
st
hilt
fon
achaemenid
stoke
jouy
dutch
vicús
group
florence
complex
seville
chantilly
peru
berlin
petersburg
continental
indonesia
mycenaean
milan
versailles
southern
etruscan
sheffield
cypriot
neo
corinthian
central
trading
chancay
sardis
sasanian
lyon
chelsea
bohemian
belgian
old
korea
campanian
rouen
shaker
sumerian
flemish
irish
scottish
aztec
japan
china
urbino
western
javanese
crete
limoges
catalan
spanish
parthian
senufo
rome
aymara
thailand
boeotian
kalaba
vietnam
hungarian
asia

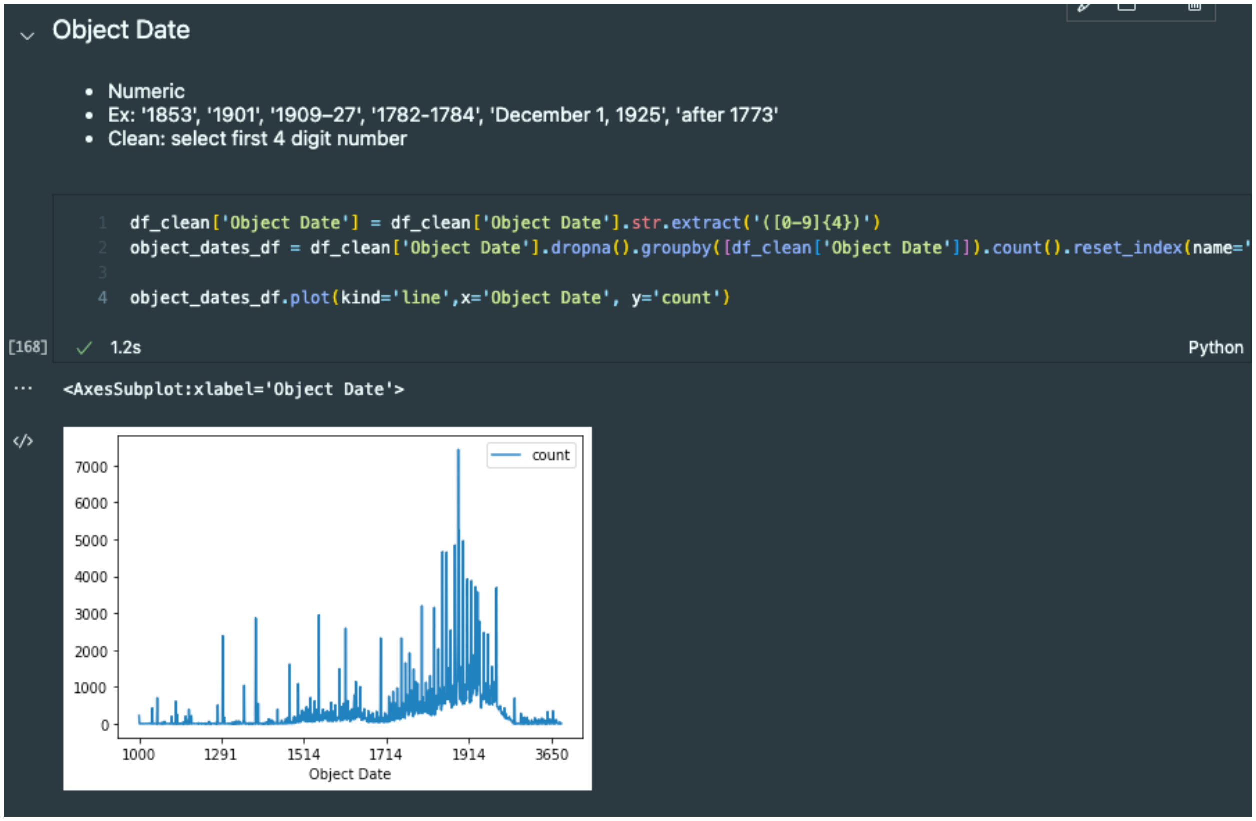


Artist Role

- object
- Ex: 'nan', 'Maker', 'Designer|Manufacturer'
- Clean: split on vert bar?

```python
artist_roles = lines_to_words(df_clean['Artist Role'].dropna().tolist())
artist_roles_clean = []
for i in range(len(artist_roles)):
    artist_roles_clean.extend(artist_roles[i].lower().split('|'))

artist_roles_df = pd.DataFrame(artist_roles_clean, columns=['Artist Role'])

#calculate sum of values by group
df_groups = artist_roles_df.groupby(['Artist Role'])['Artist Role'].sum()

artist_roles_df = artist_roles_df['Artist Role'].dropna().groupby([artist_roles_df['Artist Role']]).count().rese
artist_roles_df['percentage'] = artist_roles_df['counts'] / artist_roles_df['counts'].sum()
top10 = artist_roles_df.sort_values(by='counts', ascending=False).head(10)
top10.plot.bar(x='Artist Role',y='percentage', rot=90)
```

```
<AxesSubplot:xlabel='Artist Role'>
```



Object Date

- Numeric
- Ex: '1853', '1901', '1909–27', '1782-1784', 'December 1, 1925', 'after 1773'
- Clean: select first 4 digit number

```
df_clean['Object Date'] = df_clean['Object Date'].str.extract('([0-9]{4})')
object_dates_df = df_clean['Object Date'].dropna().groupby([df_clean['Object Date']]).count().reset_index(name='

object_dates_df.plot(kind='line',x='Object Date', y='count')
```

[168] ✓ 1.2s Python

```
<AxesSubplot:xlabel='Object Date'>
```