Module 2 - ICE

Name: Chance Wiese

DATA 3300

Exercise

Import the *masonrybldg.xls* dataset into Notebooks, then complete the data cleaning activities noted below. Once complete, you'll have a single Excel document that includes the scrubbed data. We will be performing all data cleaning activities using the pandas and numpy libraries!

```
# import required libraries – pandas and numpy

import pandas as pd
import numpy as np
```

```
# add in file path and specify read_ type

mason = pd.read_excel('/content/masonrybldg.xlsx')
```

```
# produce a dataframe heading

mason.head()
```

	Unnamed: 0	ObsID	Preliminary Risk Category	Neighborhood	Address	Year Built	No. Stories	Retrofit Level
0	NaN	19	High Risk	Capitol Hill	925 E Pike St	1916	1	Substantial Alteration
1	NaN	40	Medium Risk	Capitol Hill	1621 12th Ave	1917	1	Substantial Alteration
2	NaN	265	Medium Risk	Capitol Hill	1510 Melrose Ave	1930	2	Substantial Alteration
3	NaN	95	Medium Risk	Alki-Admiral	1321 Harbor Ave SW	1915	1	No visible retrofit
4	NaN	49	Medium Risk	Alki/Admiral	2124 California Ave SW	1928	3	No visible retrofit

1. **Remove any leading and trailing spaces from all text columns. *Note: The trim feature does not remove additional spaces between two words.***

```
# remove trailing and leading whitespaces

mason['Preliminary Risk Category'] = mason['Preliminary Risk Category'].str.strip()
mason['Neighborhood'] = mason['Neighborhood'].str.strip()
mason['Address'] = mason['Address'].str.strip()
mason['Retrofit Level'] = mason['Retrofit Level'].str.strip()
mason['Building Use'] = mason['Building Use'].str.strip()
mason['Confirmation Source'] = mason['Confirmation Source'].str.strip()
```

2. **Eliminate any records that have no Address or Retrofit Level data.**

```
# retain only observations that are complete for address and retrofit level

mason_full = mason[mason['Address'].notna()]
mason_full = mason_full[mason_full['Retrofit Level'].notna()]

# print out mason_full

mason_full.head()
```

	Unnamed: 0	ObsID	Preliminary Risk Category	Neighborhood	Address	Year Built	No. Stories	Retrofit Level
0	NaN	19	High Risk	Capitol Hill	925 E Pike St	1916	1	Substantial Alteration
1	NaN	40	Medium Risk	Capitol Hill	1621 12th Ave	1917	1	Substantial Alteration
2	NaN	265	Medium Risk	Capitol Hill	1510 Melrose Ave	1930	2	Substantial Alteration
3	NaN	95	Medium Risk	Alki-Admiral	1321 Harbor Ave SW	1915	1	No visible retrofit
4	NaN	49	Medium Risk	Alki/Admiral	2124 California Ave SW	1928	3	No visible retrofit

3. **Ensure that the labels for Neighborhood and Retrofit Level are consistent (i.e., there's shouldn't be different spellings, abbreviations, or just multiple ways of saying the same thing).**

```
# examine levels of neighborhood via value counts

mason_full['Neighborhood'].value_counts()
```

```
Capitol Hill                139
Duwamish/SODO                79
Cascade/Eastlake             71
Belltown                     68
Ballard                      66
Downtown                     57
First Hill                   45
Greenwood/Phinney Ridge      29
Columbia City                27
Central Area/Squire Park     24
Green Lake                   20
Georgetown                   17
Fremont                      13
Judkins Park                 13
Fauntleroy/Seaview           11
Beacon Hill                   6
Interbay                      5
```

```
Cap Hill                    3
Cedar Park/Meadowbrook      2
Alki–Admiral                2
Broadview/Bitter Lake       2
Alki/Admiral                2
Cascade/Eastlak             1
Highland Park               1
Name: Neighborhood, dtype: int64
```

```
# replace redundant neighborhood values

mason_full = mason_full.replace({'Cap Hill':'Capitol Hill',
                                 'Alki–Admiral':'Alki/Admiral',
                                 'Cascade/Eastlak':'Cascade/Eastlake'})
```

```
# examine levels of retrofit via value counts

mason_full['Retrofit Level'].value_counts()
```

```
No visible retrofit       373
Permitted Retrofit        126
Substantial Alteration     89
Visible retrofit           70
None visible               45
Name: Retrofit Level, dtype: int64
```

```
# replace redundant retrofit levels

mason_full = mason_full.replace('None visible','No visible retrofit')
mason_full['Retrofit Level'].value_counts()
```

```
No visible retrofit       418
Permitted Retrofit        126
Substantial Alteration     89
Visible retrofit           70
Name: Retrofit Level, dtype: int64
```

4. **Many of the buildings are dual-use. This is indicated in the Building Use column. Create two separate columns from the Building Use column, one for the first use listed and the other for the second.**

```
# split building use into two new columns

mason_full[['Primary Use','Secondary Use']] = mason_full['Building Use'].str.split('/', expand=True)
```

5. **Create a new column called "IsCritical". For those buildings shown with a preliminary risk value of "Critical Risk", the value for "IsCritical" should be 1. For all others, the value should be 0.**

```
# create new 'IsCritical' column

mason_full['IsCritical'] = np.where(mason_full['Preliminary Risk Category'] == 'Critical Risk',1,0)
```

6. **We'd like to be able to categorize the buildings' age. Create a new column and name it Era. Populate this column with information reflecting to which of the following 'eras' each building belongs: "before 1920", "1920-1939", "1940-1959", "1960-1979", or "after 1979".**

```
# create new era variable

conditions = [
    (mason_full['Year Built'] < 1920),
    (mason_full['Year Built'] < 1940),
    (mason_full['Year Built'] < 1960),
    (mason_full['Year Built'] < 1980),
    (mason_full['Year Built'] > 1979)
]

values = ['before 1920', '1920-1939', '1940-1959', '1960-1979', 'after 1979']

mason_full['Era'] = np.select(conditions, values)
```

7. **Delete any unnecessary columns and SORT the data by Observation ID.**
8. **Save this .ipynb file, print it to a PDF, and export the cleaned data as an Excel file. You'll upload all three files to this Canvas assignment!**

```
# set index and sort by descending order of ObsID

mason_full = mason_full.set_index('ObsID')
mason_full = mason_full.sort_index()

mason_full.head()
```

```
mason_full.columns
```

```
Index(['Preliminary Risk Category', 'Neighborhood', 'Address', 'Year Built',
       'No. Stories', 'Retrofit Level', 'Building Use',
       'Estimated Number of Occupants', 'Confirmation Source', 'Primary Use',
       'Secondary Use', 'IsCritical', 'Era'],
      dtype='object')
```

```
# drop empty columns

mason_full = mason_full.drop('Unnamed: 0', axis = 1)

mason_full.head()
```

```
# export cleaned data to excel

mason_full.to_excel("Cleaned_mason.xlsx")
```