In [1]:

```
import pandas as pd
```

In [2]:

```
ls
```

```
Inspections.csv  Inventroy.csv    Untitled.ipynb   violations.csv
```

In [3]:

```
pd.read_csv('Inspections.csv').head(5)
```

Out[3]:

PROGRAM STATUS	PROGRAM ELEMENT (PE)	PE DESCRIPTION	...	SERVICE DESCRIPTION	SCORE	GRADE	SERIAL NUMBER	EMPL
ACTIVE	1631	RESTAURANT (0-30) SEATS MODERATE RISK	...	ROUTINE INSPECTION	97.0	A	DA2FXQNN6	EE00
ACTIVE	1631	RESTAURANT (0-30) SEATS MODERATE RISK	...	ROUTINE INSPECTION	95.0	A	DACP43IQW	EE00
ACTIVE	1631	RESTAURANT (0-30) SEATS MODERATE RISK	...	ROUTINE INSPECTION	96.0	A	DAEMVMRBY	EE00
ACTIVE	1631	RESTAURANT (0-30) SEATS MODERATE RISK	...	ROUTINE INSPECTION	96.0	A	DANER68S4	EE00
ACTIVE	1632	RESTAURANT (0-30) SEATS HIGH RISK	...	ROUTINE INSPECTION	90.0	A	DACZXQ74W	EE00

In [10]:

```
len(pd.read_csv('Inspections.csv')['Zip Codes'].unique())
```

Out[10]:

```
282
```

In [8]:

```
pd.read_csv('Inspections.csv').columns
```

Out[8]:

```
Index(['ACTIVITY DATE', 'OWNER ID', 'OWNER NAME', 'FACILITY ID',
       'FACILITY NAME', 'RECORD ID', 'PROGRAM NAME', 'PROGRAM STATUS',
       'PROGRAM ELEMENT (PE)', 'PE DESCRIPTION', 'FACILITY ADDRESS',
       'FACILITY CITY', 'FACILITY STATE', 'FACILITY ZIP', 'SERVICE COD
E',
       'SERVICE DESCRIPTION', 'SCORE', 'GRADE', 'SERIAL NUMBER', 'EMPL
OYEE ID',
       'Location', '2011 Supervisorial District Boundaries (Officia
l)',
       'Census Tracts 2010', 'Board Approved Statistical Areas', 'Zip
Codes'],
      dtype='object')
```

In [6]:

```
pd.read_csv('Inventroy.csv').head(5)
```

Out[6]:

	FACILITY ID	FACILITY NAME	RECORD ID	PROGRAM NAME	PROGRAM ELEMENT (PE)	PE DESCRIPTION	FACILITY ADDRES
0	FA0019645	DREAM DINNERS	PR0045642	DREAM DINNERS	1631	RESTAURANT (0-30) SEATS MODERATE RISK	22226 PALO VERDES BLV
1	FA0056432	#1 CAFE	PR0045100	#1 CAFE	1632	RESTAURANT (0-30) SEATS HIGH RISK	2080 CENTURY PAR E STE 10
2	FA0241857	LAUREL TAVERN	PR0189987	LAUREL TAVERN	1638	RESTAURANT (61-150) SEATS HIGH RISK	1220 HERMOSA AV
3	FA0262822	10 SPEED COFFEE	PR0213623	10 SPEED COFFEE	1631	RESTAURANT (0-30) SEATS MODERATE RISK	191 SANTA&#160;MONIC BLVD # 10
4	FA0158893	10 - E	PR0146972	10 - E	1638	RESTAURANT (61-150) SEATS HIGH RISK	811 W 7TH S

5 rows × 23 columns

In [7]:

```
pd.read_csv('Inventroy.csv').columns
```

Out[7]:

```
Index(['FACILITY ID', 'FACILITY NAME', 'RECORD ID', ' PROGRAM NAME',
       'PROGRAM ELEMENT (PE)', 'PE DESCRIPTION', 'FACILITY ADDRESS',
       'FACILITY CITY', 'FACILITY  STATE', 'FACILITY ZIP', 'FACILITY L
ATITUDE',
       'FACILITY LONGITUDE', 'OWNER ID', 'OWNER NAME', 'OWNER ADDRES
S',
       'OWNER CITY', 'OWNER STATE', 'OWNER ZIP', 'Location',
       'Census Tracts 2010',
       '2011 Supervisorial District Boundaries (Official)',
       'Board Approved Statistical Areas', 'Zip Codes'],
      dtype='object')
```

In [5]:

```
pd.read_csv('violations.csv')
```

Out[5]:

	SERIAL NUMBER	VIOLATION STATUS	VIOLATION CODE	VIOLATION DESCRIPTION	POINTS
0	DA0004KIJ	OUT OF COMPLIANCE	F049	# 50. Impoundment of unsanitary equipment or food	0
1	DA0004KIJ	OUT OF COMPLIANCE	F042	# 42. Toilet facilities: properly constructed,...	1
2	DA0004KIJ	OUT OF COMPLIANCE	F037	# 37. Adequate ventilation and lighting; desig...	1
3	DA0004KIJ	OUT OF COMPLIANCE	F015	# 15. Food obtained from approved source	2
4	DA0004KIJ	OUT OF COMPLIANCE	F006	# 06. Adequate handwashing facilities supplied...	2
...	...	...	...	...	...
954076	DAZZZA0P5	OUT OF COMPLIANCE	F034	# 34. Warewashing facilities: Adequate, mainta...	1
954077	DAZZZA0P5	OUT OF COMPLIANCE	F030	# 30. Food properly stored; food storage conta...	1
954078	DAZZZA0P5	OUT OF COMPLIANCE	F024	# 24. Person in charge present and performs du...	1
954079	DAZZZIUVR	OUT OF COMPLIANCE	F035	# 35. Equipment/Utensils - approved; installed...	1
954080	DAZZZIUVR	OUT OF COMPLIANCE	F034	# 34. Warewashing facilities: Adequate, mainta...	1

954081 rows × 5 columns

In [13]:

```
'''Outputs should not include any data from vendors that have a ‘PROGRAM
STATUS’ of INACTIVE.'''

inspections_sample = pd.read_csv('Inspections.csv').head(100)
inspections_sample[~inspections_sample['PROGRAM STATUS'] == "INACTIVE"]
```

Out[13]:

	ACTIVITY DATE	OWNER ID	OWNER NAME	FACILITY ID	FACILITY NAME	RECORD ID	PROGRAM NAME	PROGRAM STATUS
9	03/29/2017	OW0000010	1 EVEN, INC.	FA0015045	CHO MAN WON	PR0021365	CHO MAN WON	INACTIVE
10	10/04/2016	OW0000010	1 EVEN, INC.	FA0015045	CHO MAN WON	PR0021365	CHO MAN WON	INACTIVE
11	02/15/2018	OW0000010	1 EVEN, INC.	FA0015045	CHO MAN WON	PR0021365	CHO MAN WON	INACTIVE
12	11/15/2017	OW0000010	1 EVEN, INC.	FA0015045	CHO MAN WON	PR0021365	CHO MAN WON	INACTIVE
69	09/21/2016	OW0000028	101 GINSENG INC	FA0002829	101 GINSENG UNC	PR0018827	101 GINSENG UNC	INACTIVE
70	04/24/2018	OW0000028	101 GINSENG INC	FA0002829	101 GINSENG UNC	PR0018827	101 GINSENG UNC	INACTIVE

6 rows × 25 columns

In [ ]:

In [46]:

```
'''
2. The ‘PE DESCRIPTION’ column contains information on the number and type of
seating available at the vendor. Extract this out into a new column, retain all
other information within that column. E.g.:
a. ‘FOOD MKT RETAIL (1-1,999 SF) LOW RISK’,
b. ‘RESTAURANT (61-150) SEATS LOW RISK’.
c. Extract the greyed area out and retain the rest in the examples
d. The client initially needs information to generate the following and output
the results using appropriate representation:'''
import re

# extract content inside parantheses
in_paran = lambda x: re.search(r'\((.*?)\)',x).group(1)
inspections_sample['in_paran'] = inspections_sample['PE DESCRIPTION'].apply(in_parar
# extract content outside parantheses
no_paran = lambda x: " ".join(re.findall(r"(.*?)(?:\(.*?\)|$)", x))
inspections_sample['no_paran'] = inspections_sample['PE DESCRIPTION'].apply(no_parar
```

In [80]:

```
# 3. Produce the mean, mode and median for the inspection score per year:
#a. For each type of vendor’s seating
# b. For each ‘zip code’

from dateutil.parser import parse
import numpy as np
from scipy import stats

#create year column
inspections_sample['year'] = inspections_sample['ACTIVITY DATE'].apply(lambda x: par
# inspections_sample.pivot_table(index = 'in_paran', columns = 'year', values = 'SCC

def pivot(index, columns, values, aggfunc, table):
    return table.pivot_table(index = index,
                             columns = columns,
                             values = values,
                             aggfunc = aggfunc )

mode = lambda x: stats.mode(x)[0][0]
mode_table_seating = pivot('in_paran','year','SCORE', mode, inspections_sample)
mean_table_seating = pivot('in_paran','year','SCORE', np.mean, inspections_sample)
median_table_seating = pivot('in_paran','year','SCORE', np.median, inspections_sampl

mode_table_zip = pivot('Zip Codes','year','SCORE', mode, inspections_sample)
```

In [102]:

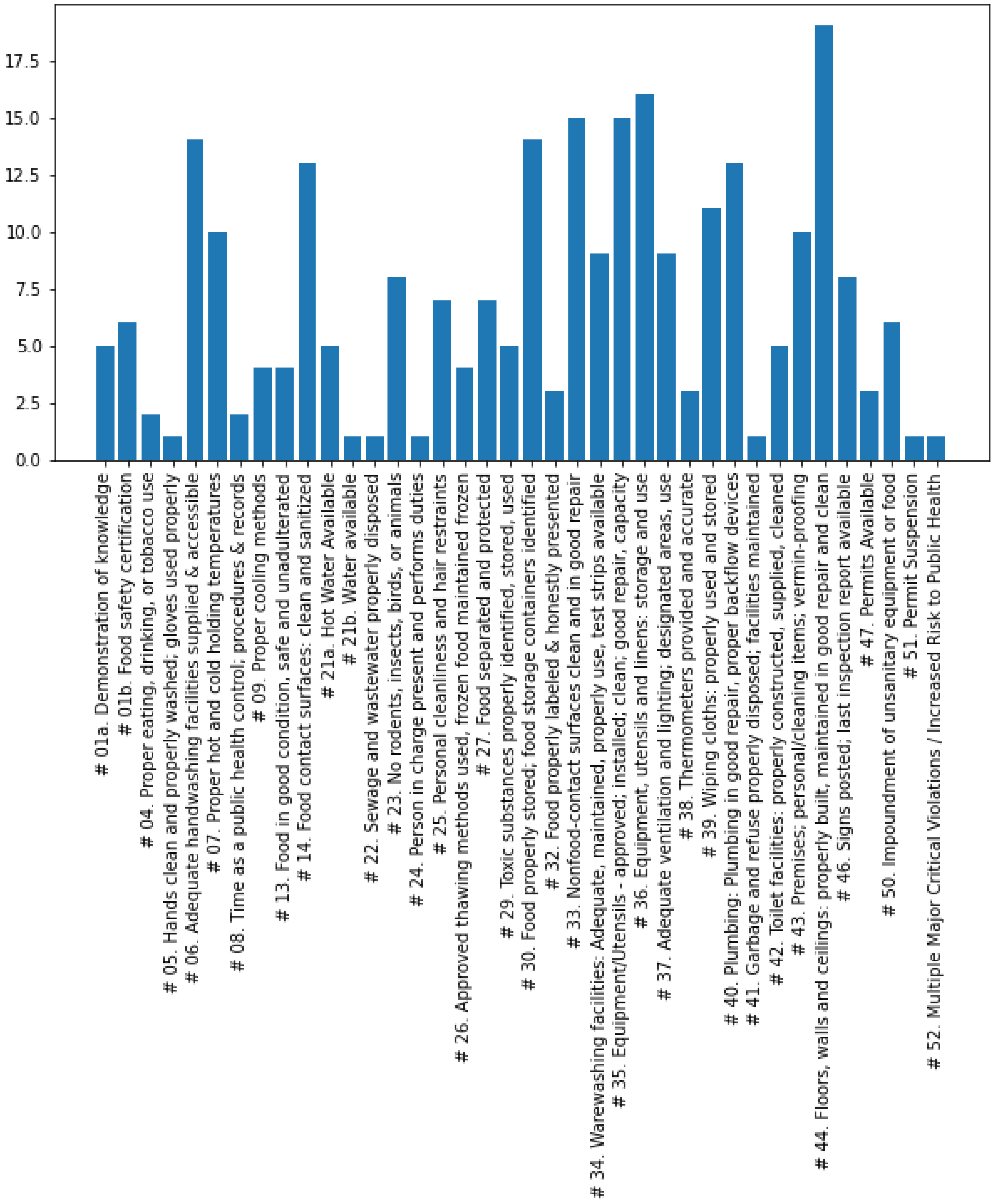
```
'''
Produce a suitable graph that displays the number of establishments that have
committed each type of violation, you may need to consider how you group this
data to make visualisation feasible
'''
violations = pd.read_csv("violations.csv")
merger = pd.merge(inspections_sample, violations, how='inner', on='SERIAL NUMBER')
```

In [142]:

```
import matplotlib.pyplot as plt
%matplotlib inline
violations = merger.groupby('VIOLATION DESCRIPTION')['FACILITY NAME'].nunique()
fig = plt.figure(figsize = (10, 5))
x = np.arange(len(violations.index))
plt.xticks(x, violations.index, rotation='vertical')
plt.bar(violations.index, violations.values)
```

Out[142]:

```
<BarContainer object of 38 artists>
```



In [279]:

```
'''
5. Determine if there is any significant correlation between the number of violatio
committed per vendor and their zip code, ‘Is there a tendency for facilities in
specific locations to have more violations?’. You will need to select an appropriate
visualisation to demonstrate this
'''
```

Out[279]:

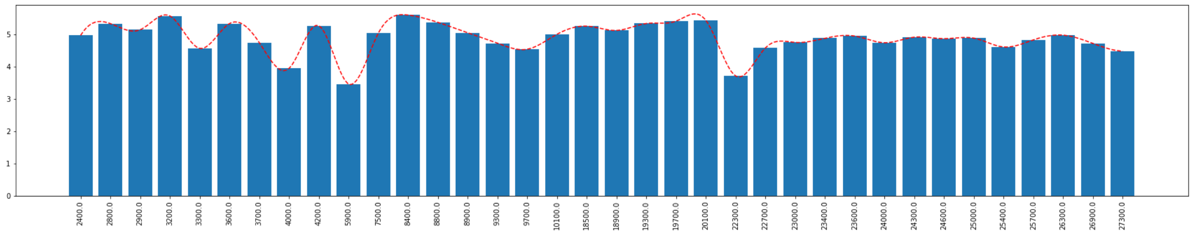
```
'\n5. Determine if there is any significant correlation between the nu
mber of violations\ncommitted per vendor and their zip code, ‘Is there
a tendency for facilities in\nspecific locations to have more violatio
ns?’. You will need to select an appropriate\nvisualisation to demonst
rate this\n'
```

In [269]:

```
from scipy.interpolate import interp1d

inspections = pd.read_csv('Inspections.csv')
inspections['Zip Bin'] = np.floor(inspections['Zip Codes']/100)*100
zipvio = inspections[['FACILITY NAME','Zip Bin']].value_counts().groupby('Zip Bin').
fig, ax = plt.subplots(figsize = (30, 5))
zipindex = [str(x) for x in zipvio.index]
x = np.arange(len(zipvio.index))
plt.xticks(x, zipindex, rotation='vertical')
ax.bar(zipindex, zipvio.values)

f2 = interp1d(x, zipvio.values, kind='cubic')
xnew = np.linspace(0, len(zipindex)-1, num=len(zipindex)*10, endpoint=True)

ax.plot(xnew, f2(xnew), '--', color = 'red')
```

Out[269]:

```
[<matplotlib.lines.Line2D at 0x12e636070>]
```



In [284]:

```
inspections[['FACILITY NAME','Zip Bin']].value_counts().groupby('Zip Bin').mean()
```

Out[284]:

```
Zip Bin
2400.0     4.963100
2800.0     5.327456
2900.0     5.136691
3200.0     5.564935
3300.0     4.560278
3600.0     5.329909
3700.0     4.740000
4000.0     3.941176
4200.0     5.263690
5900.0     3.454545
7500.0     5.035714
8400.0     5.591944
8800.0     5.366045
8900.0     5.043189
9300.0     4.720563
9700.0     4.534539
10100.0    4.998270
18500.0    5.248292
18900.0    5.115346
19300.0    5.331281
19700.0    5.405447
20100.0    5.426614
22300.0    3.703704
22700.0    4.577528
23000.0    4.744530
23400.0    4.874106
23600.0    4.942240
24000.0    4.735835
24300.0    4.907510
24600.0    4.860717
25000.0    4.877540
25400.0    4.604087
25700.0    4.824377
26300.0    4.969935
26900.0    4.712644
27300.0    4.474806
dtype: float64
```

In [286]:

```
test = inspections[['FACILITY NAME','Zip Bin']][inspections['Zip Bin']==2400.0]
```

In [291]:

```
inspections_sample.to_csv('inspections_sample.csv', index=False)
```

In [297]:

```
import os
def find_csv_files(path):
    # Check for csvs in path
    filenames = os.listdir(path)
    csv_files = [x for x in filenames if x.endswith('.csv')]
    return csv_files


cwd = os.getcwd()
find_csv_files(cwd)
```

Out[297]:

```
['Inspections.csv', 'Inventroy.csv', 'violations.csv']
```

In [302]:

```
os.getcwd()
```

Out[302]:

```
'/Users/tarek.lel/Desktop/uni/advanced-programming/formative-assignmen
t/DataSet1'
```

In [ ]:

```

```