Wk 1 - Homework - KenwanCheung

Prompt: Use Python Regular Expressions to identify top-10 most frequent causes of failed food inspections in Chicago. You can download the dataset here: https://data.cityofchicago.org/Health-Human-Services/Food-Inspections/4ijn-s7e5

In [2]:
```
import nltk
import re
import pandas as pd
import numpy as np
```

In [3]:
```
# import csv
fi = pd.read_csv('../wk1/Food_Inspections.csv')
```

In [5]:
```
fi.head(10)
```

Out[5]:

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Addr
0	2130049	JOE & THE JUICE ILLINOIS, LLC	JOE & THE JUICE	2564512.0	Restaurant	Risk 2 (Medium)	10 E DELAWAR PL
1	2130022	BOO BAE TEA INC	BOO BAE TEA INC	2570290.0	NaN	Risk 2 (Medium)	1013 W Webster A

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Addr
2	2130018	FRESHII	FRESHII	2446395.0	Restaurant	Risk 1 (High)	1166 W MADISON
3	2129964	LINCOLN PARK PRESCHOOL	LINCOLN PARK PRESCHOOL & KINDERGARTEN	2215624.0	Daycare (2 - 6 Years)	Risk 1 (High)	108 W GERMANI PL
4	2129963	ORIGINAL STEAM	ORIGINAL STEAM	2574892.0	NaN	Risk 1 (High)	2428 S WALLACE AVE
5	2129953	JERK TACO MAN 1	JERK TACO MAN	2529063.0	Restaurant	Risk 1 (High)	4001 W JACKSON BLVD
6	2129949	LA CEBOLLITA GRILL #2, INC.	LA CEBOLLITA GRILL	2184654.0	Restaurant	Risk 1 (High)	1807 S ASHLAND AVE
7	2129931	CLIFTON GRILL,INC.	CLIFTON GRILL	2164148.0	Restaurant	Risk 1 (High)	6410-12 N CLAREMO AVE

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Addr
8	2129924	BALU SMOOTHIES AND MOOR, LLC	LA FITNESS HEALTH CLUB	2559797.0	Restaurant	Risk 2 (Medium)	55 E RANDOLP ST
9	2129920	KEDZIE DISCOUNT MART	KEDZIE DISCOUNT MART	2506863.0	Grocery Store	Risk 3 (Low)	2410 S KEDZIE AV

In [6]:
```
np.unique(fi.Results)
```

Out[6]:
```
array(['Business Not Located', 'Fail', 'No Entry', 'Not Ready',
       'Out of Business', 'Pass', 'Pass w/ Conditions'], dtype=object)
```

In [89]:
```
# segment into own df
fail = fi[fi["Results"] == "Fail"].reset_index()
fail = fail.Violations
```

In [91]:
```
# check
fail.head(10)
```

Out[91]:
```
0    9. WATER SOURCE: SAFE, HOT & COLD UNDER CITY P...
1    3. POTENTIALLY HAZARDOUS FOOD MEETS TEMPERATUR...
2    12. HAND WASHING FACILITIES: WITH SOAP AND SAN...
3    18. NO EVIDENCE OF RODENT OR INSECT OUTER OPEN...
4    41. PREMISES MAINTAINED FREE OF LITTER, UNNECE...
5    16. FOOD PROTECTED DURING STORAGE, PREPARATION...
6    33. FOOD AND NON-FOOD CONTACT EQUIPMENT UTENSI...
7    2. FACILITIES TO MAINTAIN PROPER TEMPERATURE -...
8    13. NO EVIDENCE OF RODENT OR INSECT INFESTATIO...
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
9    18. NO EVIDENCE OF RODENT OR INSECT OUTER OPEN...
Name: Violations, dtype: object
```

In [93]:
```
# reading longer strings of reasons
fail[0]
```

Out[93]:
```
'9. WATER SOURCE: SAFE, HOT & COLD UNDER CITY PRESSURE - Comments: OBSE
RVE NO HOT RUNNING WATER ON THE PREMISES; THAT INCLUDES EXPOSED HAND SI
NKS IN FRONT & REAR PREP AREAS, 3-COMPARTMENT SINK IN REAR PREP, AND HA
ND SINKS IN BOTH TOILET ROOMS. INSTRUCTED TO CONTACT PLUMMER TO HAVE HO
T WATER RESTORED. CRITICAL VIOLATION 7-38-030. | 29. PREVIOUS MINOR VIO
LATION(S) CORRECTED 7-42-090 - Comments: PREVIOUS MINOR VIOLATION NOT C
ORRECTED FROM INSPECTION REPORT 1989320, DATED 2/17/2017. VIOLATION INC
LUDES; #38; NO RUNNING HOT AND COLD WATER TO TOP LOADING SOFT SERVE MAC
HINE, INSTRUCTED TO PROVIDE,  \n \nVIOLATION STILL EXISTS. SERIOUS VIOL
ATION 7-42-090 | 32. FOOD AND NON-FOOD CONTACT SURFACES PROPERLY DESIGN
ED, CONSTRUCTED AND MAINTAINED - Comments: MUST DISCONTINUE USING MILK
CRATES AS STORAGE RACKS THROUGHOUT FRONT AND REAR PREP AREAS, AND IN TH
E WALK IN COOLER. INSTALL CORRECT STORAGE RACKS. | 34. FLOORS: CONSTRUC
TED PER CODE, CLEANED, GOOD REPAIR, COVING INSTALLED, DUST-LESS CLEANIN
G METHODS USED - Comments: MUST REPAIR COVING ON WALL ACROSS FROM THE E
XPOSED HAND SINK  \nIN THE REAR PREP AREA.'
```

Parsing

It seems like we need to pull the section past the . and space, which seems to be the automated comment.

We should parse that string until the hyphen "-"

In [259]:
```
fail.tail()
```

Out[259]:
```
31545    41. PREMISES MAINTAINED FREE OF LITTER, UNNECE...
31546    21. * CERTIFIED FOOD MANAGER ON SITE WHEN POTE...
31547    18. NO EVIDENCE OF RODENT OR INSECT OUTER OPEN...
31548    41. PREMISES MAINTAINED FREE OF LITTER, UNNECE...
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
31549    34. FLOORS: CONSTRUCTED PER CODE, CLEANED, GOO...
Name: Violations, dtype: object
```

In [ ]:
```
# test pattern
reason_codes = pd.DataFrame(columns = ['reason'])

for i in range(0,len(fail)):
#     print(i)
    test_text = str(fail[i])
    m = re.findall(r"\d\.\s(.*?)\s-\sComments:",test_text)
#     print(m)
    if len(m) > 0:
        reason_codes.loc[i] = m[0]
    else:
        reason_codes.loc[i] = None
```

In [328]:
```
reason_codes[0:5]
```

Out[328]:

	reason
0	WATER SOURCE: SAFE, HOT & COLD UNDER CITY PRES...
1	POTENTIALLY HAZARDOUS FOOD MEETS TEMPERATURE R...
2	HAND WASHING FACILITIES: WITH SOAP AND SANITAR...
3	NO EVIDENCE OF RODENT OR INSECT OUTER OPENINGS...
4	PREMISES MAINTAINED FREE OF LITTER, UNNECESSAR...

In [329]:
```
# remove NAs
reason_codes_clean = reason_codes.dropna(axis=0, how='any')
reason_codes_clean[0:10]
```

Out[329]:

	reason
0	WATER SOURCE: SAFE, HOT & COLD UNDER CITY PRES...

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	reason
1	POTENTIALLY HAZARDOUS FOOD MEETS TEMPERATURE R...
2	HAND WASHING FACILITIES: WITH SOAP AND SANITAR...
3	NO EVIDENCE OF RODENT OR INSECT OUTER OPENINGS...
4	PREMISES MAINTAINED FREE OF LITTER, UNNECESSAR...
5	FOOD PROTECTED DURING STORAGE, PREPARATION, DI...
6	FOOD AND NON-FOOD CONTACT EQUIPMENT UTENSILS C...
7	FACILITIES TO MAINTAIN PROPER TEMPERATURE
8	NO EVIDENCE OF RODENT OR INSECT INFESTATION, N...
9	NO EVIDENCE OF RODENT OR INSECT OUTER OPENINGS...

Now we have a cleaned list of just the reason codes. We should thus find the highest frequency on this list

Let's use nltk to get the highest frequency

In [330]:
```
reason_freq = reason_codes_clean.reason.value_counts()
reason_freq[0:10]
```

Out[330]:
```
NO EVIDENCE OF RODENT OR INSECT OUTER OPENINGS PROTECTED/RODENT PROOFE
D, A WRITTEN LOG SHALL BE MAINTAINED AVAILABLE TO THE INSPECTORS    754
1
FACILITIES TO MAINTAIN PROPER TEMPERATURE
                                                                   2012
PREVIOUS MINOR VIOLATION(S) CORRECTED 7-42-090
                                                                   1621
FOOD PROTECTED DURING STORAGE, PREPARATION, DISPLAY, SERVICE AND TRANSP
ORTATION                                                           1352
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ADEQUATE NUMBER, CONVENIENT, ACCESSIBLE, DESIGNED, AND MAINTAINED
                                                                1332
POTENTIALLY HAZARDOUS FOOD MEETS TEMPERATURE REQUIREMENT DURING STORAG
E, PREPARATION DISPLAY AND SERVICE                               120
6
FOOD AND NON-FOOD CONTACT SURFACES PROPERLY DESIGNED, CONSTRUCTED AND M
AINTAINED                                                        1050
* CERTIFIED FOOD MANAGER ON SITE WHEN POTENTIALLY HAZARDOUS FOODS ARE
PREPARED AND SERVED                                               936
VENTILATION: ROOMS AND EQUIPMENT VENTED AS REQUIRED: PLUMBING: INSTALLE
D AND MAINTAINED                                                  933
OUTSIDE GARBAGE WASTE GREASE AND STORAGE AREA; CLEAN, RODENT PROOF, ALL
CONTAINERS COVERED                                               925
Name: reason, dtype: int64
```

Most common results:

The most common reason codes are in the 10 above. The most common reason code above all else seems to be protection from rodents, which is not shocking!

In [ ]:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD