Link of dataset

https://www.toronto.ca/city-government/data-research-maps/open-data/open-data-catalogue/health/#e3c15b0f-5f83-0f12-fabb-c84018395c38 (https://www.toronto.ca/city-government/data-research-maps/open-data/open-data-catalogue/health/#e3c15b0f-5f83-0f12-fabb-c84018395c38)

Importing Needed libraries

In [1]:

```
%matplotlib inline
import pandas as pd
from pandas import Series, DataFrame

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.core.interactiveshell import InteractiveShell
from pandas.plotting import autocorrelation_plot

import datetime

# For fetching data from google map api
import googlemaps

import concurrent.futures
import multiprocessing
import tqdm
```

Configurations

In [2]:

```
np.random.seed(12345)
plt.rc('figure', figsize=(10, 6))
PREVIOUS_MAX_ROWS = pd.options.display.max_rows
pd.options.display.max_rows = 15
np.set_printoptions(precision=4, suppress=True)
pd.options.display.max_columns = None
```

Data Loading

In [162]:

```
df = pd.read_csv('dinesafe.csv')
```

Data Dictionary

- ROW_ID - Represents the Row Number
- ESTABLISHMENT_ID – Unique identifier for an establishment
- INSPECTION_ID - Unique identifier for each Inspection
- ESTABLISHMENT_NAME – Business name of the establishment
- ESTABLISHMENTTYPE – Establishment type ie restaurant, mobile cart
- ESTABLISHMENT_ADDRESS – Municipal address of the establishment
- LONG/LAT– Longitude & Latitude coordinates of an establishment
- ESTABLISHMENT_STATUS – Pass, Conditional Pass, Closed
- MINIMUM_INSPECTIONS_PERYEAR – Every eating and drinking establishment in the City of Toronto receives a minimum of 1, 2, or 3 inspections each year depending on the specific type of establishment, the food preparation processes, volume and type of food served and other related criteria
- INFRACTION_DETAILS – Description of the Infraction
- INSPECTION_DATE – Calendar date the inspection was conducted
- SEVERITY – Level of the infraction, i.e. S – Significant, M – Minor, C – Crucial
- ACTION – Enforcement activity based on the infractions noted during a food safety inspection
- COURT_OUTCOME – The registered court decision resulting from the issuance of a ticket or summons for outstanding infractions to the Health Protection and Promotion Act
- AMOUNT_FINED – Fine determined in a court outcome

In [167]:

```
df.columns= map(str.lower, df.columns)
```

Adding date dimensions

In [168]:

```
df['inspection_date'] = pd.to_datetime(df['inspection_date'])
df['year'] = pd.DatetimeIndex(df['inspection_date']).year
df['quarter'] = pd.DatetimeIndex(df['inspection_date']).quarter
df['month'] = pd.DatetimeIndex(df['inspection_date']).month
df['week'] = pd.DatetimeIndex(df['inspection_date']).week
df['year_quarter'] = df['year'].astype(str) +'-'+ df['quarter'].astype(str)
df['year_month'] = df['year'].astype(str) +'-'+ df['month'].astype(str)
df['year_week'] = df['year'].astype(str) +'-'+ df['week'].astype(str)
df['week_day'] = pd.DatetimeIndex(df['inspection_date']).weekday #the day of the
week with monday=0, sunday=6
df.head()
```

Out[168]:

	row_id	establishment_id	inspection_id	establishment_name	establishmenttype	es
0	1	1222579	103868579	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
1	2	1222579	104063869	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
2	3	1222579	104246429	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
3	4	1222579	104246429	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
4	5	1222579	104246429	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87

Adding postal_code and area

In [169]:

```
loc = pd.read_csv("locations_2.csv")
loc.columns= map(str.lower, loc.columns)
loc.head()
```

Out[169]:

	unnamed: 0	latitude	longitude	postal_code	area
0	0	43.586770	-79.542082	M8W 3P2	Etobicoke
1	1	43.587910	-79.538666	M8W 1C1	Etobicoke
2	2	43.590227	-79.543784	M8W 3P1	Etobicoke
3	3	43.591013	-79.545156	M8W 1R3	Etobicoke
4	4	43.591901	-79.543077	M8W 1R3	Etobicoke

In [170]:

```
df = pd.merge(df,loc,on=['latitude','longitude'],how='left')
df.head()
```

Out[170]:

	row_id	establishment_id	inspection_id	establishment_name	establishmenttype	es
0	1	1222579	103868579	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
1	2	1222579	104063869	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
2	3	1222579	104246429	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
3	4	1222579	104246429	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87
4	5	1222579	104246429	SAI-LILA KHAMAN DHOKLA HOUSE	Food Take Out	87

Data Profiling

In [8]:

```
cnt_value =[]
for col_nm in df.columns:
    a=len(df.groupby(col_nm).size())
    cnt_value.append(a)

cnt_null = df.isnull().sum()
pd.DataFrame({
    '# of distinct value': cnt_value,
    '# of null value': cnt_null
    })
```

Out[8]:

	# of distinct value	# of null value
row_id	90520	0
establishment_id	16291	0
inspection_id	55589	0
establishment_name	12780	0
establishmenttype	55	0
establishment_address	11284	0
latitude	10686	0
...	...	...
year_quarter	9	0
year_month	25	0
year_week	105	0
week_day	7	0
unnamed: 0	9649	11761
postal_code	5348	11761
area	6	11761

27 rows × 2 columns

In [9]:

```
tmp1 = df[df['infraction_details'].notna()]
tmp1.groupby(['establishment_status']).size()
```

Out[9]:

```
establishment_status
Closed                 352
Conditional Pass     17237
Pass                 44109
dtype: int64
```

In [10]:

```
tmp1 = df[df['infraction_details'].isna()]
tmp1.groupby(['establishment_status']).size()
```

Out[10]:

```
establishment_status
Pass    28822
dtype: int64
```

In [11]:

```
#ESTABLISHMENT_NAME
df_est = df.establishment_name.value_counts()
df_est.head(15)
```

Out[11]:

```
TIM HORTONS            1305
SUBWAY                  854
MCDONALD'S              457
PIZZA PIZZA             432
SECOND CUP              309
FRESHII                 275
AROMA ESPRESSO BAR      266
STARBUCKS COFFEE        234
THAI EXPRESS            228
STARBUCKS               224
METRO                   212
BOOSTER JUICE           208
PIZZA NOVA              175
SHOPPERS DRUG MART      170
PIZZAIOLO               166
Name: establishment_name, dtype: int64
```

In [12]:

```
#ESTABLISHMENTTYPE
df_est = df.establishmenttype.value_counts()
df_est.head(15)
```

Out[12]:

```
Restaurant                            49016
Food Take Out                         13119
Food Store (Convenience / Variety)     4448
Supermarket                            3209
Food Court Vendor                      3196
Bakery                                 2454
Child Care - Catered                   2027
Child Care - Food Preparation          1738
Food Caterer                            938
Banquet Facility                        901
Butcher Shop                            895
Food Processing Plant                   885
Cafeteria                               818
Retirement Homes(Licensed)              697
Nursing Home / Home for the Aged        505
Name: establishmenttype, dtype: int64
```

In [13]:

```
#ESTABLISHMENT_STATUS
df.establishment_status.value_counts()
```

Out[13]:

```
Pass                    72931
Conditional Pass        17237
Closed                    352
Name: establishment_status, dtype: int64
```

In [14]:

```
#MINIMUM_INSPECTIONS_PERYEAR >>> Establishment Risk Category
df.minimum_inspections_peryear.value_counts()
```

Out[14]:

```
2    44295
3    38629
1     7596
Name: minimum_inspections_peryear, dtype: int64
```

In [15]:

```
#INFRACTION_DETAILS
df_inf = df.infraction_details.value_counts()
df_inf.head(15)
```

Out[15]:

```
Operator fail to properly wash surfaces in rooms
8822
Operator fail to properly maintain rooms
7145
Operator fail to properly wash equipment
5961
Operator fail to properly maintain equipment(NON-FOOD)
2350
Operator fail to provide proper equipment
1909
Fail to Ensure the Presence of the Holder of a Valid Food Handler's
Certificate. Muncipal Code Chapter 545-157(17)(a)    1540
FAIL TO PROVIDE THERMOMETER IN STORAGE COMPARTMENT O. REG  562/90 SE
C. 21                                                1297
FAIL TO HAVE TEST REAGENT AVAILABLE AT  PLACE OF SANITIZATION O. REG
562/90 SEC. 75(2)                                   1252
Operator fail to use proper procedure(s) to ensure food safety
1196
Food handler fail to wear headgear
1133
OPERATOR FAIL TO ENSURE COVER WILL PREVENT CONTAMINATION OR ADULTERA
TION O. REG  562/90 SEC. 59(C)(II)                    949
Operator fail to provide adequate pest control
937
FAIL TO STORE FOOD ON RACKS OR SHELVES O. REG  562/90 SEC. 23
914
STORE UTENSILS IN MANNER NOT PREVENTING CONTAMINATION O. REG  562/90
SEC. 81                                              864
FAIL TO PROVIDE TOWELS IN FOOD PREPARATION AREA O. REG  562/90 SEC.
20(1)(C)                                              794
Name: infraction_details, dtype: int64
```

In [16]:

```
#SEVERITY
df.severity.value_counts()
```

Out[16]:

```
M - Minor               32280
S - Significant         22970
NA - Not Applicable      4048
C - Crucial              2400
Name: severity, dtype: int64
```

In [17]:

```
#ACTION
df.action.value_counts()
```

Out[17]:

```
Notice to Comply                   46529
Corrected During Inspection        14292
Ticket                               693
Summons                               89
Summons and Health Hazard Order       58
Not in Compliance                     25
Education Provided                     3
Prohibition Order Requested            3
Recommendations                        3
Closure Order                          2
Warning Letter                         1
Name: action, dtype: int64
```

In [18]:

```
#COURT_OUTCOME
df.court_outcome.value_counts()
```

Out[18]:

```
Pending                                          478
Conviction - Fined                               242
Charges Withdrawn                                 38
Cancelled                                         10
Charges Quashed                                    5
Conviction - Fined & Order to Close by Court       5
Conviction - Suspended Sentence                    3
Name: court_outcome, dtype: int64
```

In [19]:

```
#AMOUNT_FINED
df_fined = df.amount_fined.value_counts()
df_fined.head(10)
```

Out[19]:

```
60.0     65
120.0    50
305.0    24
0.0      14
300.0    13
460.0     8
115.0     5
180.0     5
95.0      4
310.0     4
Name: amount_fined, dtype: int64
```

In [20]:

```
#year
df.year.value_counts()
```

Out[20]:

```
2017    44132
2018    34607
2016    11781
Name: year, dtype: int64
```

In [21]:

```
#year + quarter
df.groupby(['year','quarter']).size()
```

Out[21]:

```
year  quarter
2016  3            879
      4          10902
2017  1          10158
      2          11222
      3          10775
      4          11977
2018  1          12061
      2          12227
      3          10319
dtype: int64
```

In [22]:

```
#year + month
save_max_rows = pd.options.display.max_rows
pd.options.display.max_rows = 0
display(df.groupby(['year','month']).size())
pd.options.display.max_rows = save_max_rows
```

```
year  month
2016  9         879
      10       3815
      11       3442
      12       3645
2017  1        2746
      2        3369
      3        4043
      4        3572
      5        3464
      6        4186
      7        3256
      8        4093
      9        3426
      10       4043
      11       4170
      12       3764
2018  1        3980
      2        3888
      3        4193
      4        4259
      5        4234
      6        3734
      7        3768
      8        4359
      9        2192
dtype: int64
```

In [23]:

```
df.groupby(['area']).size()
```

Out[23]:

```
area
East York        1801
Etobicoke        6480
North York      17327
Old Toronto     36781
Scarborough     13255
York             3115
dtype: int64
```

A. Data Preparation

In [24]:

```
start = datetime.date(2016, 9, 1)
end = datetime.date(2018, 9, 30)
bm_rng = pd.date_range(start, end, freq='M')
df_yrmth = pd.DataFrame({
    'date': bm_rng,
    'year': bm_rng.year,
    'month': bm_rng.month,
    'day': bm_rng.day,
    'year_month': bm_rng.year.astype(str) + '-' + bm_rng.month.astype(str)
})
```

Count of Inspection by Year_Month

In [25]:

```
def count_inspection(dim, val, interval):
    cnt_insp = df[df[dim]==val].groupby([interval]).inspection_id.nunique()
    return cnt_insp

df_inspection = pd.DataFrame({
#Total
'Inspection: Total': df.groupby(['year_month']).inspection_id.nunique(),

#Risk Category
'Inspection: High Risk': count_inspection('minimum_inspections_peryear',3,'year_month'),
'Inspection: Medium Risk': count_inspection('minimum_inspections_peryear',2,'year_month'),
'Inspection: Low Risk': count_inspection('minimum_inspections_peryear',1,'year_month'),

#ESTABLISHMENTTYPE
'Inspection: Restaurant': count_inspection('establishmenttype','Restaurant','year_month'),
'Inspection: Food_Take_Out': count_inspection('establishmenttype','Food Take Out','year_month'),
'Inspection: Food_Store': count_inspection('establishmenttype','Food Store (Convenience / Variety)','year_month'),
'Inspection: Supermarket': count_inspection('establishmenttype','Supermarket','year_month'),
'Inspection: Food_Court': count_inspection('establishmenttype','Food Court Vendor','year_month'),
'Inspection: Bakery': count_inspection('establishmenttype','Bakery','year_month'),
'Inspection: Child_Care_Catered': count_inspection('establishmenttype','Child Care - Catered','year_month'),
'Inspection: Child_Care_Food_Prep': count_inspection('establishmenttype','Child

Care - Food Preparation','year_month'),
'Inspection: Food_Caterer': count_inspection('establishmenttype','Food Caterer',
'year_month'),
'Inspection: Banquet': count_inspection('establishmenttype','Banquet Facility','
year_month'),
'Inspection: Butcher_Shop': count_inspection('establishmenttype','Butcher Shop',
'year_month'),
'Inspection: Food_Plant': count_inspection('establishmenttype','Food Processing
Plant','year_month'),
'Inspection: Cafeteria': count_inspection('establishmenttype','Cafeteria','year_
month'),
'Inspection: Retirement': count_inspection('establishmenttype','Retirement Homes
(Licensed)','year_month'),
'Inspection: Nursing_Home': count_inspection('establishmenttype','Nursing Home /
Home for the Aged','year_month'),

#ESTABLISHMENT_NAME
'Inspection: TIM HORTONS': count_inspection('establishment_name','TIM HORTONS','
year_month'),
'Inspection: SUBWAY': count_inspection('establishment_name','SUBWAY','year_month
'),
"Inspection: MCDONALD'S": count_inspection('establishment_name',"MCDONALD'S",'ye
ar_month'),
'Inspection: PIZZA PIZZA': count_inspection('establishment_name','PIZZA PIZZA','
year_month'),
'Inspection: SECOND CUP': count_inspection('establishment_name','SECOND CUP','ye
ar_month'),
'Inspection: FRESHII': count_inspection('establishment_name','FRESHII','year_mon
th'),
'Inspection: AROMA ESPRESSO BAR': count_inspection('establishment_name','AROMA E
SPRESSO BAR','year_month'),
'Inspection: STARBUCKS COFFEE': count_inspection('establishment_name','STARBUCKS
COFFEE','year_month'),
'Inspection: THAI EXPRESS': count_inspection('establishment_name','THAI EXPRESS'
,'year_month'),
'Inspection: STARBUCKS': count_inspection('establishment_name','STARBUCKS','year
_month')

})
df_inspection.index = df_yrmth['year_month']
df_inspection = df_inspection.fillna(0)
df_inspection.head()
```

Out[25]:

	Inspection: Total	Inspection: High Risk	Inspection: Medium Risk	Inspection: Low Risk	Inspection: Restaurant	Inspect Food_Take_
year_month						
2016-9	2562	729	1490	343	1168	427
2016-10	2303	701	1332	270	1094	312
2016-11	2330	730	1330	270	1187	245
2016-12	598	145	396	57	265	122
2017-1	1834	474	1027	333	910	275

In [26]:

```
def count_infraction(dim,val,interval):
    df_infr = df[df['infraction_details'].notna()]
    cnt_infr = df_infr[df_infr[dim]==val].groupby([interval]).inspection_id.size
()
    return cnt_infr

df_infraction = pd.DataFrame({
#pass_rate
'Infraction: Total': df[df['infraction_details'].notna()].groupby(['year_month']
).inspection_id.size(),

#Risk Category
'Infraction: High Risk': count_infraction('minimum_inspections_peryear',3,'year_
month'),
'Infraction: Medium Risk': count_infraction('minimum_inspections_peryear',2,'yea
r_month'),
'Infraction: Low Risk': count_infraction('minimum_inspections_peryear',1,'year_m
onth'),

#ESTABLISHMENTTYPE
'Infraction: Restaurant': count_infraction('establishmenttype','Restaurant','yea
r_month'),
'Infraction: Food_Take_Out': count_infraction('establishmenttype','Food Take Out
','year_month'),
'Infraction: Food_Store': count_infraction('establishmenttype','Food Store (Conv
enience / Variety)','year_month'),
'Infraction: Supermarket': count_infraction('establishmenttype','Supermarket','y
ear_month'),
'Infraction: Food_Court': count_infraction('establishmenttype','Food Court Vendo
r','year_month'),
'Infraction: Bakery': count_infraction('establishmenttype','Bakery','year_month'
),
'Infraction: Child Care Catered': count_infraction('establishmenttype','Child Ca
'Infraction: Child_Care_Catered': count_infraction('establishmenttype','Child Care - Catered','year_month'),
'Infraction: Child_Care_Food_Prep': count_infraction('establishmenttype','Child Care - Food Preparation','year_month'),
'Infraction: Food_Caterer': count_infraction('establishmenttype','Food Caterer','year_month'),
'Infraction: Banquet': count_infraction('establishmenttype','Banquet Facility','year_month'),
'Infraction: Butcher_Shop': count_infraction('establishmenttype','Butcher Shop','year_month'),
'Infraction: Food_Plant': count_infraction('establishmenttype','Food Processing Plant','year_month'),
'Infraction: Cafeteria': count_infraction('establishmenttype','Cafeteria','year_month'),
'Infraction: Retirement': count_infraction('establishmenttype','Retirement Homes (Licensed)','year_month'),
'Infraction: Nursing_Home': count_infraction('establishmenttype','Nursing Home / Home for the Aged','year_month'),

#ESTABLISHMENT_NAME
'Infraction: TIM HORTONS': count_infraction('establishment_name','TIM HORTONS','year_month'),
'Infraction: SUBWAY': count_infraction('establishment_name','SUBWAY','year_month'),
"Infraction: MCDONALD'S": count_infraction('establishment_name',"MCDONALD'S",'year_month'),
'Infraction: PIZZA PIZZA': count_infraction('establishment_name','PIZZA PIZZA','year_month'),
'Infraction: SECOND CUP': count_infraction('establishment_name','SECOND CUP','year_month'),
'Infraction: FRESHII': count_infraction('establishment_name','FRESHII','year_month'),
'Infraction: AROMA ESPRESSO BAR': count_infraction('establishment_name','AROMA ESPRESSO BAR','year_month'),
'Infraction: STARBUCKS COFFEE': count_infraction('establishment_name','STARBUCKS COFFEE','year_month'),
'Infraction: THAI EXPRESS': count_infraction('establishment_name','THAI EXPRESS','year_month'),
'Infraction: STARBUCKS': count_infraction('establishment_name','STARBUCKS','year_month')

})
df_infraction = df_infraction.fillna(0)
df_infraction.index = df_yrmth['year_month']
df_infraction.head()
```

Out[26]:

	Infraction: Total	Infraction: High Risk	Infraction: Medium Risk	Infraction: Low Risk	Infraction: Restaurant	Infraction: Food_Take_Out
year_month						
2016-9	2384	1072	1161	151	1426	331
2016-10	2162	925	1115	122	1275	297
2016-11	2395	1076	1208	111	1528	239
2016-12	532	213	305	14	300	72
2017-1	1715	765	825	125	1013	212

Infraction/Inspection Ratio by Year_Month

In [27]:

```
def ratio_infr_insp(dim,val,interval):
    df_infr = df[df['infraction_details'].notna()]
    cnt_infr = df_infr[df_infr[dim]==val].groupby([interval]).inspection_id.size
()
    cnt_insp = df[df[dim]==val].groupby([interval]).inspection_id.nunique()
    ratio_infr_insp = cnt_infr/cnt_insp
    return ratio_infr_insp

df_ratio_infr_insp = pd.DataFrame({
#pass_rate
'Ratio: Total': df[df['infraction_details'].notna()].groupby(['year_month']).ins
pection_id.size() / df.groupby(['year_month']).inspection_id.nunique(),

#Risk Category
'Ratio: High Risk': ratio_infr_insp('minimum_inspections_peryear',3,'year_month'
),
'Ratio: Medium Risk': ratio_infr_insp('minimum_inspections_peryear',2,'year_mont
h'),
'Ratio: Low Risk': ratio_infr_insp('minimum_inspections_peryear',1,'year_month')
,

#ESTABLISHMENTTYPE
'Ratio: Restaurant': ratio_infr_insp('establishmenttype','Restaurant','year_mont
h'),
'Ratio: Food_Take_Out': ratio_infr_insp('establishmenttype','Food Take Out','yea
r_month'),
'Ratio: Food_Store': ratio_infr_insp('establishmenttype','Food Store (Convenienc
e / Variety)','year_month'),
'Ratio: Supermarket': ratio_infr_insp('establishmenttype','Supermarket','year_mo
nth'),
'Ratio: Food_Court': ratio_infr_insp('establishmenttype','Food Court Vendor','ye
ar_month'),
'Ratio: Bakery': ratio_infr_insp('establishmenttype','Bakery','year_month'),
'Ratio: Child_Care_Catered': ratio_infr_insp('establishmenttype','Child Care - C
atered','year_month'),
'Ratio: Child_Care_Food_Prep': ratio_infr_insp('establishmenttype','Child Care -
Food Preparation','year_month'),
'Ratio: Food_Caterer': ratio_infr_insp('establishmenttype','Food Caterer','year_
month'),
'Ratio: Banquet': ratio_infr_insp('establishmenttype','Banquet Facility','year_m
onth'),
'Ratio: Butcher_Shop': ratio_infr_insp('establishmenttype','Butcher Shop','year_
month'),
'Ratio: Food_Plant': ratio_infr_insp('establishmenttype','Food Processing Plant'
,'year_month'),
'Ratio: Cafeteria': ratio_infr_insp('establishmenttype','Cafeteria','year_month'
),
'Ratio: Retirement': ratio_infr_insp('establishmenttype','Retirement Homes(Licen
sed)','year_month'),
'Ratio: Nursing_Home': ratio_infr_insp('establishmenttype','Nursing Home / Home
for the Aged','year_month'),

#ESTABLISHMENT_NAME
'Ratio: TIM HORTONS': ratio_infr_insp('establishment_name','TIM HORTONS','year_m
onth'),
'Ratio: SUBWAY': ratio_infr_insp('establishment_name','SUBWAY','year_month'),
"Ratio: MCDONALD'S": ratio_infr_insp('establishment_name',"MCDONALD'S",'year_mon
th'),
'Ratio: PIZZA PIZZA': ratio_infr_insp('establishment_name','PIZZA PIZZA','year_m
onth'),
'Ratio: SECOND CUP': ratio_infr_insp('establishment_name','SECOND CUP','year_mon
th'),
'Ratio: FRESHII': ratio_infr_insp('establishment_name','FRESHII','year_month'),
'Ratio: AROMA ESPRESSO BAR': ratio_infr_insp('establishment_name','AROMA ESPRESS
O BAR','year_month'),
'Ratio: STARBUCKS COFFEE': ratio_infr_insp('establishment_name','STARBUCKS COFFE
E','year_month'),
'Ratio: THAI EXPRESS': ratio_infr_insp('establishment_name','THAI EXPRESS','year
_month'),
'Ratio: STARBUCKS': ratio_infr_insp('establishment_name','STARBUCKS','year_month
')

})
df_ratio_infr_insp = df_ratio_infr_insp.fillna(0)
df_ratio_infr_insp.index = df_yrmth['year_month']
df_ratio_infr_insp.head()
```

```
Out[27]:
```

	Ratio: Total	Ratio: High Risk	Ratio: Medium Risk	Ratio: Low Risk	Ratio: Restaurant	Ratio: Food_Take_Out	Foo
year_month							
2016-9	0.930523	1.470508	0.779195	0.440233	1.220890	0.775176	0.54
2016-10	0.938776	1.319544	0.837087	0.451852	1.165448	0.951923	0.58
2016-11	1.027897	1.473973	0.908271	0.411111	1.287279	0.975510	0.34
2016-12	0.889632	1.468966	0.770202	0.245614	1.132075	0.590164	0.35
2017-1	0.935115	1.613924	0.803311	0.375375	1.113187	0.770909	0.58

```
In [28]:
```

```
def pass_rate_cal(dim, val, interval):
    df_pass = df[df['establishment_status']=='Pass']
    cnt_pass = df_pass[df_pass[dim]==val].groupby([interval]).inspection_id.nuni
que()
    cnt_insp = df[df[dim]==val].groupby([interval]).inspection_id.nunique()
    pass_rate = pd.Series(cnt_pass/cnt_insp*100)
    return pass_rate
df_pass_rate = pd.DataFrame({
#pass_rate
'Pass Rate: Total': (df[df['establishment_status']=='Pass'].groupby(['year_month
']).inspection_id.nunique()/df.groupby(['year_month']).inspection_id.nunique())*
100,

#Risk Category
'Pass Rate: High Risk': pass_rate_cal('minimum_inspections_peryear',3,'year_mont
h'),
'Pass Rate: Medium Risk': pass_rate_cal('minimum_inspections_peryear',2,'year_mo
nth'),
'Pass Rate: Low Risk': pass_rate_cal('minimum_inspections_peryear',1,'year_month
'),

#ESTABLISHMENTTYPE
'Pass Rate: Restaurant': pass_rate_cal('establishmenttype','Restaurant','year_mo
nth'),
'Pass Rate: Food_Take_Out': pass_rate_cal('establishmenttype','Food Take Out','y
ear_month'),
'Pass Rate: Food_Store': pass_rate_cal('establishmenttype','Food Store (Convenie
nce / Variety)','year_month'),
'Pass Rate: Supermarket': pass_rate_cal('establishmenttype','Supermarket','year_
month'),
'Pass Rate: Food_Court': pass_rate_cal('establishmenttype','Food Court Vendor','
year_month'),
'Pass Rate: Bakery': pass_rate_cal('establishmenttype','Bakery','year_month'),
'Pass Rate: Bakery': pass_rate_cal('establishmenttype','Bakery','year_month'),
'Pass Rate: Child_Care_Catered': pass_rate_cal('establishmenttype','Child Care -
Catered','year_month'),
'Pass Rate: Child_Care_Food_Prep': pass_rate_cal('establishmenttype','Child Care
- Food Preparation','year_month'),
'Pass Rate: Food_Caterer': pass_rate_cal('establishmenttype','Food Caterer','yea
r_month'),
'Pass Rate: Banquet': pass_rate_cal('establishmenttype','Banquet Facility','year
_month'),
'Pass Rate: Butcher_Shop': pass_rate_cal('establishmenttype','Butcher Shop','yea
r_month'),
'Pass Rate: Food_Plant': pass_rate_cal('establishmenttype','Food Processing Plan
t','year_month'),
'Pass Rate: Cafeteria': pass_rate_cal('establishmenttype','Cafeteria','year_mont
h'),
'Pass Rate: Retirement': pass_rate_cal('establishmenttype','Retirement Homes(Lic
ensed)','year_month'),
'Pass Rate: Nursing_Home': pass_rate_cal('establishmenttype','Nursing Home / Hom
e for the Aged','year_month'),

#ESTABLISHMENT_NAME
'Pass Rate: TIM HORTONS': pass_rate_cal('establishment_name','TIM HORTONS','year
_month'),
'Pass Rate: SUBWAY': pass_rate_cal('establishment_name','SUBWAY','year_month'),
"Pass Rate: MCDONALD'S": pass_rate_cal('establishment_name',"MCDONALD'S",'year_m
onth'),
'Pass Rate: PIZZA PIZZA': pass_rate_cal('establishment_name','PIZZA PIZZA','year
_month'),
'Pass Rate: SECOND CUP': pass_rate_cal('establishment_name','SECOND CUP','year_m
onth'),
'Pass Rate: FRESHII': pass_rate_cal('establishment_name','FRESHII','year_month')
,
'Pass Rate: AROMA ESPRESSO BAR': pass_rate_cal('establishment_name','AROMA ESPRE
SSO BAR','year_month'),
'Pass Rate: STARBUCKS COFFEE': pass_rate_cal('establishment_name','STARBUCKS COF
FEE','year_month'),
'Pass Rate: THAI EXPRESS': pass_rate_cal('establishment_name','THAI EXPRESS','ye
ar_month'),
'Pass Rate: STARBUCKS': pass_rate_cal('establishment_name','STARBUCKS','year_mon
th')
})
df_pass_rate.index = df_yrmth['year_month']
df_pass_rate.head()
```

```
Out[28]:
```

	Pass Rate: Total	Pass Rate: High Risk	Pass Rate: Medium Risk	Pass Rate: Low Risk	Pass Rate: Restaurant	Pass Rate: Food_Take_Out
year_month						
2016-9	94.964871	91.769547	95.838926	97.959184	94.349315	94.847775
2016-10	93.834129	90.584879	94.894895	97.037037	92.778793	95.833333
2016-11	94.592275	91.369863	95.563910	98.518519	93.513058	96.734694
2016-12	94.983278	93.103448	95.454545	96.491228	92.830189	97.540984
2017-1	94.274809	87.763713	95.618306	99.399399	93.296703	95.272727

Plot Functions by Year_Month

```
In [29]:
```

```
def infraction_ratio_bar(insp,infr,ratio):
    df_grf = pd.DataFrame({
        '# of Inspection':df_inspection[insp],
        '# of Infraction': df_infraction[infr]
    })
    f = df_grf.plot.bar(figsize=(20,5))
    ts = list(df_ratio_infr_insp[ratio])
    dt = df_yrmth['year_month']
    s = pd.Series(ts, index=dt)
    f = s.plot(secondary_y=True, kind='line', label=ratio, legend=True, figsize=
(20,5), color ='k', marker='o')

#inspection plot
def inspection_plot(var):
    dt = df_yrmth['date']
    ts = list(df_inspection[var])
    s = pd.Series(ts, index=dt)
    f = s.plot(label=var, legend=True, figsize=(20,5), marker='o')
    plt.xlabel('Inspection Date')
    plt.ylabel('Inspection')

#infraction plot
def infraction_plot(var):
    dt = df_yrmth['date']
    ts = list(df_infraction[var])
    s = pd.Series(ts, index=dt)
    f = s.plot(label=var, legend=True, figsize=(20,5), marker='o')
    plt.xlabel('Inspection Date')
    plt.ylabel('Infraction')
```

```
#df_ratio_infr_insp
def ratio_infr_insp_plot(var):
    dt = df_yrmth['date']
    ts = list(df_ratio_infr_insp[var])
    s = pd.Series(ts, index=dt)
    f = s.plot(label=var, legend=True, figsize=(20,5), marker='o')
    plt.xlabel('Inspection Date')
    plt.ylabel('Infraction/Inspection Ratio')

def ratio_infr_insp_plot_y(var):
    dt = df_yrmth['date']
    ts = list(df_ratio_infr_insp[var])
    s = pd.Series(ts, index=dt)
    f = s.plot(secondary_y=True, label=var, legend=True, figsize=(20,5), marker=
'o')
    plt.xlabel('Inspection Date')
    plt.ylabel('Infraction/Inspection Ratio')

#pass_rate plot
def pass_rate_plot(var):
    dt = df_yrmth['date']
    ts = list(df_pass_rate[var])
    s = pd.Series(ts, index=dt)
    f = s.plot(label=var, legend=True, figsize=(20,5), marker='o')
    plt.xlabel('Inspection Date')
    plt.ylabel('Pass Rate (%)')

def pass_rate_plot_y(var):
    dt = df_yrmth['date']
    ts = list(df_pass_rate[var])
    s = pd.Series(ts, index=dt)
    f = s.plot(secondary_y=True, label=var, legend=True, figsize=(20,5), marker=
'o')
    plt.xlabel('Inspection Date')
    plt.ylabel('Pass Rate (%)')
```

Top15 Establishment Type

In [30]:

```
df_est_type = df.establishmenttype.value_counts()
t15 = df_est_type.head(15)
df_t15 = pd.DataFrame({'establishmenttype':t15.index})
df_est_t15 = pd.merge(df_t15, df, on='establishmenttype', how='left')

df_all = df_est_t15.copy()
df_infr = df_all[df_all['infraction_details'].notna()]
cnt_infr_all = df_infr.groupby(['establishmenttype']).inspection_id.size()
cnt_insp_all = df_all.groupby(['establishmenttype']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
```

```
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['establishmenttype']).inspection_id.nunique(
)
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_17 = df_est_t15[df_est_t15['year']==2017]
df_infr_17 = df_all[df_all['infraction_details'].notna()].query('year==2017')
cnt_infr_17 = df_infr_17.groupby(['establishmenttype']).inspection_id.size()
cnt_insp_17 = df_17.groupby(['establishmenttype']).inspection_id.nunique()
ratio_infr_insp_17 = pd.Series(cnt_infr_17/cnt_insp_17)
df_pass_17 = df_17[df_17['establishment_status']=='Pass']
cnt_pass_17 = df_pass_17.groupby(['establishmenttype']).inspection_id.nunique()
pass_rate_17 = pd.Series(cnt_pass_17/cnt_insp_17*100)

df_18 = df_est_t15[df_est_t15['year']==2018]
df_infr_18 = df_all[df_all['infraction_details'].notna()].query('year==2018')
cnt_infr_18 = df_infr_18.groupby(['establishmenttype']).inspection_id.size()
cnt_insp_18 = df_18.groupby(['establishmenttype']).inspection_id.nunique()
ratio_infr_insp_18 = pd.Series(cnt_infr_18/cnt_insp_18)
df_pass_18 = df_18[df_18['establishment_status']=='Pass']
cnt_pass_18 = df_pass_18.groupby(['establishmenttype']).inspection_id.nunique()
pass_rate_18 = pd.Series(cnt_pass_18/cnt_insp_18*100)

df_est_type = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all,

'infr_17': cnt_infr_17,
'insp_17': cnt_insp_17,
'ratio_infr_insp_17': ratio_infr_insp_17,
'pass_17': cnt_pass_17,
'pass_rate_17': pass_rate_17,

'infr_18': cnt_infr_18,
'insp_18': cnt_insp_18,
'ratio_infr_insp_18': ratio_infr_insp_18,
'pass_18': cnt_pass_18,
'pass_rate_18': pass_rate_18
})

df_est_type.head()
```

```
Out[30]:
```

	infr	insp	ratio_infr_insp	pass	pass_rate	infr_17	insp_17	ratio_
establishmenttype								
Bakery	1831	1330	1.376692	1181	88.796992	899	666	1.349
Banquet Facility	551	590	0.933898	553	93.728814	302	304	0.993
Butcher Shop	636	518	1.227799	476	91.891892	309	255	1.211
Cafeteria	448	603	0.742952	578	95.854063	220	297	0.740
Child Care - Catered	426	1894	0.224921	1843	97.307286	228	991	0.230

Top15 Establishment Name

```
In [70]:
```

```
df_est_name = df.establishment_name.value_counts()
t15 = df_est_name.head(15)
df_t15 = pd.DataFrame({'establishment_name':t15.index})
df_est_t15 = pd.merge(df_t15, df, on='establishment_name', how='left')

df_all = df_est_t15.copy()
df_infr = df_all[df_all['infraction_details'].notna()]
cnt_infr_all = df_infr.groupby(['establishment_name']).inspection_id.size()
cnt_insp_all = df_all.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['establishment_name']).inspection_id.nunique
()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_17 = df_est_t15[df_est_t15['year']==2017]
df_infr_17 = df_all[df_all['infraction_details'].notna()].query('year==2017')
cnt_infr_17 = df_infr_17.groupby(['establishment_name']).inspection_id.size()
cnt_insp_17 = df_17.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_17 = pd.Series(cnt_infr_17/cnt_insp_17)
df_pass_17 = df_17[df_17['establishment_status']=='Pass']
cnt_pass_17 = df_pass_17.groupby(['establishment_name']).inspection_id.nunique()
pass_rate_17 = pd.Series(cnt_pass_17/cnt_insp_17*100)

df_18 = df_est_t15[df_est_t15['year']==2018]
df_infr_18 = df_all[df_all['infraction_details'].notna()].query('year==2018')
cnt_infr_18 = df_infr_18.groupby(['establishment_name']).inspection_id.size()
cnt_insp_18 = df_18.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_18 = pd.Series(cnt_infr_18/cnt_insp_18)
df_pass_18 = df_18[df_18['establishment_status']=='Pass']
cnt_pass_18 = df_pass_18.groupby(['establishment_name']).inspection_id.nunique()
```

```
pass_rate_18 = pd.Series(cnt_pass_18/cnt_insp_18*100)


df_est_name = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all,

'infr_17': cnt_infr_17,
'insp_17': cnt_insp_17,
'ratio_infr_insp_17': ratio_infr_insp_17,
'pass_17': cnt_pass_17,
'pass_rate_17': pass_rate_17,

'infr_18': cnt_infr_18,
'insp_18': cnt_insp_18,
'ratio_infr_insp_18': ratio_infr_insp_18,
'pass_18': cnt_pass_18,
'pass_rate_18': pass_rate_18
})

df_est_name.head()
```

Out[70]:

	infr	**insp**	**ratio_infr_insp**	**pass**	**pass_rate**	**infr_17**	**insp_17**	**ratio**
establishment_name								
AROMA ESPRESSO BAR	221	126	1.753968	115	91.269841	112	59	1.898
BOOSTER JUICE	142	137	1.036496	129	94.160584	64	66	0.969
FRESHII	192	172	1.116279	156	90.697674	83	79	1.050
MCDONALD'S	266	342	0.777778	325	95.029240	132	174	0.758
METRO	178	103	1.728155	97	94.174757	86	48	1.791

Top15 Infraction Details

In [72]:

```
df_inf_dtl = df.infraction_details.value_counts()
t15 = df_inf_dtl.head(15)
df_t15 = pd.DataFrame({'infraction_details':t15.index})
df_est_t15 = pd.merge(df_t15, df, on='infraction_details', how='left')

df_all = df_est_t15.copy()
df_infr = df_all[df_all['infraction_details'].notna()]
```

```
cnt_infr_all = df_infr.groupby(['infraction_details']).inspection_id.size()

cnt_insp_all = df_all.groupby(['infraction_details']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='c']
cnt_pass_all = df_pass_all.groupby(['infraction_details']).inspection_id.nunique
()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_17 = df_est_t15[df_est_t15['year']==2017]
df_infr_17 = df_all[df_all['infraction_details'].notna()].query('year==2017')
cnt_infr_17 = df_infr_17.groupby(['infraction_details']).inspection_id.size()
cnt_insp_17 = df_17.groupby(['infraction_details']).inspection_id.nunique()
ratio_infr_insp_17 = pd.Series(cnt_infr_17/cnt_insp_17)
df_pass_17 = df_17[df_17['establishment_status']=='Pass']
cnt_pass_17 = df_pass_17.groupby(['infraction_details']).inspection_id.nunique()
pass_rate_17 = pd.Series(cnt_pass_17/cnt_insp_17*100)

df_18 = df_est_t15[df_est_t15['year']==2018]
df_infr_18 = df_all[df_all['infraction_details'].notna()].query('year==2018')
cnt_infr_18 = df_infr_18.groupby(['infraction_details']).inspection_id.size()
cnt_insp_18 = df_18.groupby(['infraction_details']).inspection_id.nunique()
ratio_infr_insp_18 = pd.Series(cnt_infr_18/cnt_insp_18)
df_pass_18 = df_18[df_18['establishment_status']=='Pass']
cnt_pass_18 = df_pass_18.groupby(['infraction_details']).inspection_id.nunique()
pass_rate_18 = pd.Series(cnt_pass_18/cnt_insp_18*100)

df_inf_dtl = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all,

'infr_17': cnt_infr_17,
'insp_17': cnt_insp_17,
'ratio_infr_insp_17': ratio_infr_insp_17,
'pass_17': cnt_pass_17,
'pass_rate_17': pass_rate_17,

'infr_18': cnt_infr_18,
'insp_18': cnt_insp_18,
'ratio_infr_insp_18': ratio_infr_insp_18,
'pass_18': cnt_pass_18,
'pass_rate_18': pass_rate_18
})

df_inf_dtl.head()
```

Out[72]:

	infr	insp	ratio_infr_insp	pass	pass_rate	infr_17	insp_17	ratio_in
FAIL TO HAVE TEST REAGENT AVAILABLE AT PLACE OF SANITIZATION O. REG 562/90 SEC. 75(2)	1252	1252	1.0	NaN	NaN	654	654	1.0
FAIL TO PROVIDE THERMOMETER IN STORAGE COMPARTMENT O. REG 562/90 SEC. 21	1297	1297	1.0	NaN	NaN	669	669	1.0
FAIL TO PROVIDE TOWELS IN FOOD PREPARATION AREA O. REG 562/90 SEC. 20(1)(C)	794	794	1.0	NaN	NaN	402	402	1.0
FAIL TO STORE FOOD ON RACKS OR SHELVES O. REG 562/90 SEC. 23	914	914	1.0	NaN	NaN	506	506	1.0
Fail to Ensure the Presence of the Holder of a Valid Food Handler's Certificate. Muncipal Code Chapter 545-157(17)(a)	1540	1540	1.0	NaN	NaN	839	839	1.0

Week Day

In [128]:

```
cnt_infr_all = df[df['infraction_details'].notna()].groupby(['week_day']).inspection_id.size()
cnt_insp_all = df.groupby(['week_day']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df[df['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['week_day']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_wkdy = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})

df_wkdy.head()
```

Out[128]:

	infr	insp	ratio_infr_insp	pass	pass_rate
week_day					
0	10585	9241	1.145439	8551	92.533276
1	13260	11531	1.149944	10704	92.828029
2	13430	11670	1.150814	10815	92.673522
3	13555	12235	1.107887	11372	92.946465
4	10041	10395	0.965945	9787	94.151034

Week

In [73]:

```
cnt_infr_all = df[df['infraction_details'].notna()].groupby(['week']).inspection
_id.size()
cnt_insp_all = df.groupby(['week']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df[df['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['week']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_wk = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})

df_wk.head()
```

Out[73]:

	infr	**insp**	**ratio_infr_insp**	**pass**	**pass_rate**
week					
1	545	502	1.085657	480	95.617530
2	977	994	0.982897	932	93.762575
3	1111	1061	1.047125	995	93.779453
4	1130	1036	1.090734	957	92.374517
5	1469	1218	1.206076	1138	93.431856

Month

In [74]:

```
cnt_infr_all = df[df['infraction_details'].notna()].groupby(['month']).inspection_id.size()
cnt_insp_all = df.groupby(['month']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df[df['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['month']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_mth = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})

df_mth.head()
```

Out[74]:

	infr	insp	ratio_infr_insp	pass	pass_rate
month					
1	4524	4199	1.077399	3929	93.569898
2	5064	4307	1.175760	4009	93.081031
3	5810	4840	1.200413	4487	92.706612
4	5737	4353	1.317942	4016	92.258213
5	5335	4651	1.147065	4280	92.023221

Top 100 Establishment Name

In [75]:

```
df_est_name_t100 = df.establishment_name.value_counts()
t15 = df_est_name_t100.head(100)
df_t15 = pd.DataFrame({'establishment_name':t15.index})
df_est_t15 = pd.merge(df_t15, df, on='establishment_name', how='left')

df_all = df_est_t15.copy()
cnt_infr_all = df_all[df_all['infraction_details'].notna()].groupby(['establishment_name']).inspection_id.size()
cnt_insp_all = df_all.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['establishment_name']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_est_name_t100 = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})
df_est_name_t100 = df_est_name_t100.fillna(0)
df_est_name_t100.head()
```

Out[75]:

	infr	insp	ratio_infr_insp	pass	pass_rate
establishment_name					
2-4-1 PIZZA	41	39	1.051282	37	94.871795
7-ELEVEN	81	67	1.208955	62	92.537313
A & W	52	96	0.541667	92	95.833333
ALI BABA'S	69	40	1.725000	35	87.500000
AMAYA EXPRESS	67	41	1.634146	36	87.804878

Top 1000 Establishment Name

In [76]:

```
df_est_name_t1000 = df[df['infraction_details'].notna()].establishment_name.value_counts()
t15 = df_est_name_t1000.head(1000)
df_t15 = pd.DataFrame({'establishment_name':t15.index})
df_est_t15 = pd.merge(df_t15, df, on='establishment_name', how='left')

df_all = df_est_t15.copy()
cnt_infr_all = df_all[df_all['infraction_details'].notna()].groupby(['establishment_name']).inspection_id.size()
cnt_insp_all = df_all.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['establishment_name']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_est_name_t1000 = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})
df_est_name_t1000 = df_est_name_t1000.fillna(0)
df_est_name_t1000.head()
```

Out[76]:

	infr	insp	ratio_infr_insp	pass	pass_rate
147 ELDER ST INC.	20	6	3.333333	5.0	83.333333
2-4-1 PIZZA	41	39	1.051282	37.0	94.871795
241 PIZZA	27	20	1.350000	18.0	90.000000
3 EGGS ALL DAY PUB & GRILL	19	7	2.714286	5.0	71.428571
4C Broast Chicken	30	11	2.727273	9.0	81.818182

Top 10000 Establishment Name

In [77]:

```
df_est_name_t10000 = df[df['infraction_details'].notna()].establishment_name.value_counts()
t15 = df_est_name_t10000.head(10000)
df_t15 = pd.DataFrame({'establishment_name':t15.index})
df_est_t15 = pd.merge(df_t15, df, on='establishment_name', how='left')

df_all = df_est_t15.copy()
cnt_infr_all = df_all[df_all['infraction_details'].notna()].groupby(['establishment_name']).inspection_id.size()
cnt_insp_all = df_all.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['establishment_name']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_est_name_t10000 = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})
df_est_name_t10000 = df_est_name_t10000.fillna(0)
df_est_name_t10000.head()
```

Out[77]:

	infr	insp	ratio_infr_insp	pass	pass_rate
'K' STORE	2	3	0.666667	3.0	100.000000
1 PLUS 1 PIZZA	11	6	1.833333	5.0	83.333333
1-SATELLITE KITCHEN	2	4	0.500000	4.0	100.000000
100% Korean	3	2	1.500000	2.0	100.000000
100% MEXICANO	7	5	1.400000	5.0	100.000000

ALL Establishment Name

In [78]:

```
df_all = df.copy()
cnt_infr_all = df_all[df_all['infraction_details'].notna()].groupby(['establishment_name']).inspection_id.size()
cnt_insp_all = df_all.groupby(['establishment_name']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['establishment_name']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_est_name_all = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all
})
df_est_name_all = df_est_name_all.fillna(0)
df_est_name_all.head()
```

Out[78]:

	infr	insp	ratio_infr_insp	pass	pass_rate
'K' STORE	2.0	3	0.666667	3.0	100.000000
(RAWF 2017) HAWBERRY FARMS	0.0	1	0.000000	1.0	100.000000
1 PLUS 1 PIZZA	11.0	6	1.833333	5.0	83.333333
1-SATELLITE KITCHEN	2.0	4	0.500000	4.0	100.000000
100% Korean	3.0	2	1.500000	2.0	100.000000

Area

In [79]:

```
df_all = df.copy()
cnt_infr_all = df_all[df_all['infraction_details'].notna()].groupby(['area']).inspection_id.size()
cnt_insp_all = df_all.groupby(['area']).inspection_id.nunique()
ratio_infr_insp_all = pd.Series(cnt_infr_all/cnt_insp_all)
df_pass_all = df_all[df_all['establishment_status']=='Pass']
cnt_pass_all = df_pass_all.groupby(['area']).inspection_id.nunique()
pass_rate_all = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_17 = df[df['year'] == 2017]
cnt_infr_17 = df_17[df_17['infraction_details'].notna()].groupby(['area']).inspection_id.size()
cnt_insp_17 = df_17.groupby(['area']).inspection_id.nunique()
ratio_infr_insp_17 = pd.Series(cnt_infr_17/cnt_insp_17)
df_pass_17 = df_17[df_17['establishment_status']=='Pass']
cnt_pass_17 = df_pass_all.groupby(['area']).inspection_id.nunique()
pass_rate_17 = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_18 = df[df['year'] == 2018]
cnt_infr_18 = df_18[df_18['infraction_details'].notna()].groupby(['area']).inspection_id.size()
cnt_insp_18 = df_18.groupby(['area']).inspection_id.nunique()
ratio_infr_insp_18 = pd.Series(cnt_infr_18/cnt_insp_18)
df_pass_18 = df_18[df_18['establishment_status']=='Pass']
cnt_pass_18 = df_pass_all.groupby(['area']).inspection_id.nunique()
pass_rate_18 = pd.Series(cnt_pass_all/cnt_insp_all*100)

df_area = pd.DataFrame({
'infr': cnt_infr_all,
'insp': cnt_insp_all,
'ratio_infr_insp': ratio_infr_insp_all,
'pass': cnt_pass_all,
'pass_rate': pass_rate_all,

'infr_17': cnt_infr_17,
'insp_17': cnt_insp_17,
'ratio_infr_insp_17': ratio_infr_insp_17,
'pass_17': cnt_pass_17,
'pass_rate_17': pass_rate_17,

'infr_18': cnt_infr_18,
'insp_18': cnt_insp_18,
'ratio_infr_insp_18': ratio_infr_insp_18,
'pass_18': cnt_pass_18,
'pass_rate_18': pass_rate_18
})
df_area = df_area.fillna(0)
df_area.head()
```

Out[79]:

	infr	insp	ratio_infr_insp	pass	pass_rate	infr_17	insp_17	ratio_in
area								
East York	1031	1214	0.849259	1149	94.645799	554	597	0.92797
Etobicoke	2959	5122	0.577704	4969	97.012886	1490	2499	0.59623
North York	13525	8685	1.557283	7793	89.729419	6555	4339	1.51071
Old Toronto	25619	23024	1.112708	21568	93.676164	12155	11162	1.08896
Scarborough	8919	7926	1.125284	7202	90.865506	4532	4087	1.10888

B. Data Analysis

Question: Has "DineSafe Program" been effective to reduce incidence of foodborne illness?

Metrics used to measure effectiveness of the program:

- Infraction/Inspecation Ratio = # of infraction / # of inspection
- Pass Rate (a.k.a. Compliance Rate) = # of inspection passed / # of total inspection

DineSafe program is aimed at reducing the incidence of foodborne illness among consumers in Toronto. Increased pass rates and decreases in infraction/inspection Ratio indicate a trend to enhanced food safety.

In [80]:

```
#Data Analysis Outline:

#1. Infraction/Inspection Ratio vs Pass Rate
#1.1. Relationship between Infraction/Inspection Ratio vs Pass Rate
#1.2. Infraction/Inspection Ratio vs Pass Rate - High Risk Establishment
#1.3. Infraction/Inspection Ratio vs Pass Rate - Medium Risk Establishment
#1.4. Infraction/Inspection Ratio vs Pass Rate - Low Risk Establishment

#2. Infraction/Inspection Ratio Trend
#2.1. Infraction/Inspection Ratio Trend by Establishment Risk Category
#2.1.1. Infraction/Inspection Ratio Trend - High Risk Category
#2.1.2. Infraction/Inspection Ratio Trend - Medium Risk Category
#2.1.3. Infraction/Inspection Ratio Trend - Low Risk Category

#2.2. Infraction/Inspection Ratio by Establishment Type
#2.2.1. Infraction/Inspection Ratio by Establishment Type - Restaurant
#2.2.2. Infraction/Inspection Ratio by Establishment Type - Food Take Out
#2.2.3. Infraction/Inspection Ratio by Establishment Type - Food Court
```

```
#2.2.3. Infraction/Inspection Ratio by Establishment Type - Food Court
#2.2.4. Infraction/Inspection Ratio by Establishment Type - Supermarket
#2.2.5. Infraction/Inspection Ratio by Establishment Type - Bakery

#2.3. Infraction/Inspection Ratio on Top 15 Establishment Type
#2.3.1 Infraction/Inspection Ratio on Top 15 Establishment Type - 2017CY
#2.3.2 Infraction/Inspection Ratio on Top 15 Establishment Type - 2018CY
#2.3.3 Infraction/Inspection Ratio Year over Year (YoY) Change

#2.4. Infraction/Inspection Ratio on Top 15 Establishment Name
#2.4.1. Infraction/Inspection Ratio on Top 15 Establishment Name - 2017CY
#2.4.2. Infraction/Inspection Ratio on Top 15 Establishment Name - 2018CY
#2.4.3. Infraction/Inspection Ratio Year over Year (YoY) Change - Top 15 Establishment Name

#3. Pass Rate (Compliance Rate)
#3.1. Pass Rate Trend by Risk Category
#3.2. Pass Rate Trend by Establishment Type

#3.3. Pass Rates on Top 15 Establishment Type
#3.3.1. Pass Rates on Top 15 Establishment Type - 2017CY
#3.3.2. Pass Rates on Top 15 Establishment Type - 2018CY
#3.3.3. Pass Rate Year over Year (YoY) Change - Top 15 Establishment Type

#3.4. Pass Rates on Top 15 Establishment Name
#3.4.1. Pass Rates on Top 15 Establishment Name - 2017CY
#3.4.2. Pass Rates on Top 15 Establishment Name - 2018CY
#3.4.3. Pass Rate Year over Year (YoY) Change - Top 15 Establishment Name

#3.5. Pass Rates on Top 15 Infraction Details
#3.5.1. Pass Rates on Top 15 Infraction Details - 2017CY
#3.5.2. Pass Rates on Top 15 Infraction Details - 2018CY
#3.5.3. Pass Rate Year over Year (YoY) Change - Top 15 Infraction Details

#4. Week Day / Week / Month
#4.1. Infraction/Inspection Rate on Week Day
#4.2. Pass Rate on Week Day
#4.3. Infraction/Inspection Rate on Week
#4.4. Pass Rate on Week
#4.5. Infraction/Inspection Rate on Month
#4.6. Pass Rate on Month
```

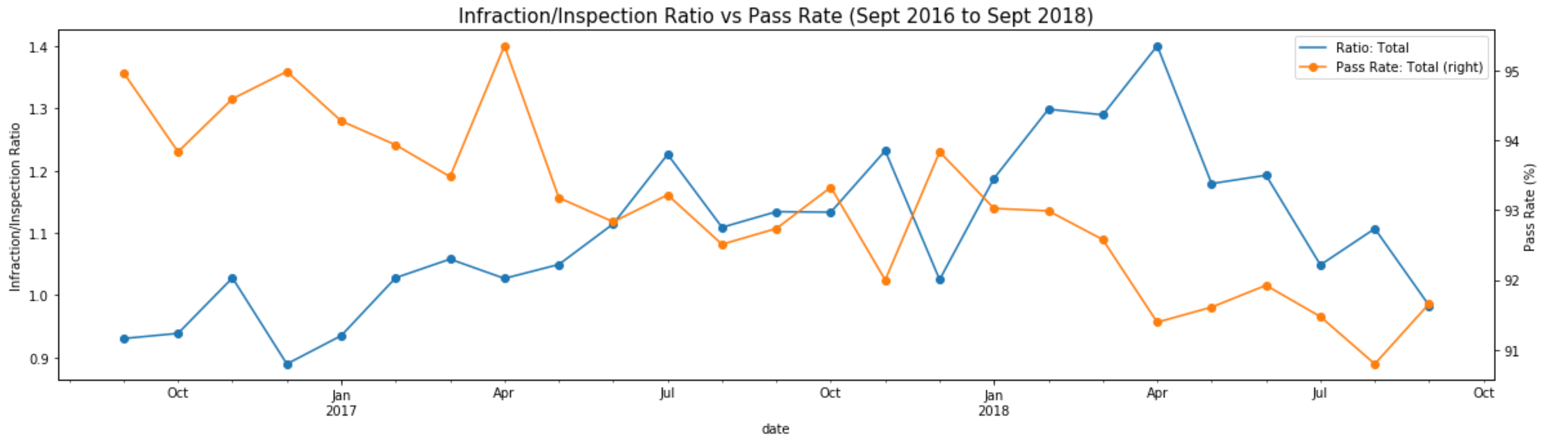
1. Infraction/Inspection Ratio vs Pass Rate

In [81]:

```
ratio_infr_insp_plot('Ratio: Total')
pass_rate_plot_y('Pass Rate: Total')
plt.title('Infraction/Inspection Ratio vs Pass Rate (Sept 2016 to Sept 2018)', f
ontsize=15)
```

Out[81]:

```
Text(0.5, 1.0, 'Infraction/Inspection Ratio vs Pass Rate (Sept 2016
to Sept 2018)')
```

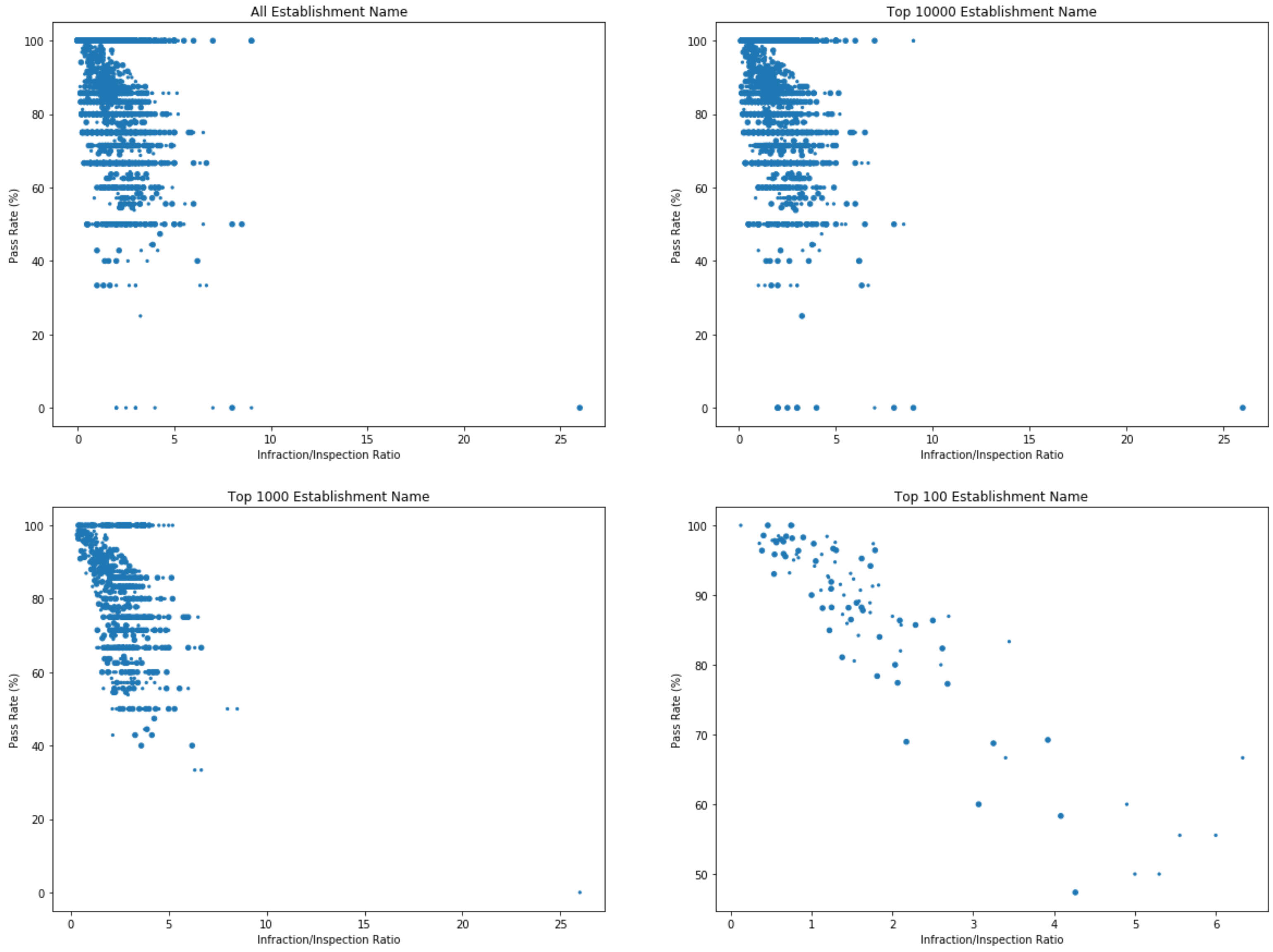


- Between Sept 2016 and Apr 2018, infraction/inspection ratio had increased. Since Apr 2018, the ratio has declined. However, the pass rates have declined in last 25 months.

1.1. Relationship between Infraction/Inspection Ratio vs Pass Rate

In [82]:

```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(20,15))

ax1 = fig.add_subplot(2,2,4)
plt.xlabel('Infraction/Inspection Ratio')
plt.ylabel('Pass Rate (%)')
plt.title('Top 100 Establishment Name', fontsize=12)

ax2 = fig.add_subplot(2,2,3)
plt.xlabel('Infraction/Inspection Ratio')
plt.ylabel('Pass Rate (%)')
plt.title('Top 1000 Establishment Name', fontsize=12)

ax3 = fig.add_subplot(2,2,2)
plt.xlabel('Infraction/Inspection Ratio')
plt.ylabel('Pass Rate (%)')
plt.title('Top 10000 Establishment Name', fontsize=12)

ax4 = fig.add_subplot(2,2,1)
plt.xlabel('Infraction/Inspection Ratio')
plt.ylabel('Pass Rate (%)')
plt.title('All Establishment Name', fontsize=12)

ax1.scatter(x = df_est_name_t100['ratio_infr_insp'], y = df_est_name_t100['pass_rate'],sizes=(20,5))
ax2.scatter(x = df_est_name_t1000['ratio_infr_insp'], y = df_est_name_t1000['pass_rate'],sizes=(20,5))
ax3.scatter(x = df_est_name_t10000['ratio_infr_insp'], y = df_est_name_t10000['pass_rate'],sizes=(20,5))
ax4.scatter(x = df_est_name_all['ratio_infr_insp'], y = df_est_name_all['pass_rate'],sizes=(20,5))

plt.show()
```



In [83]:

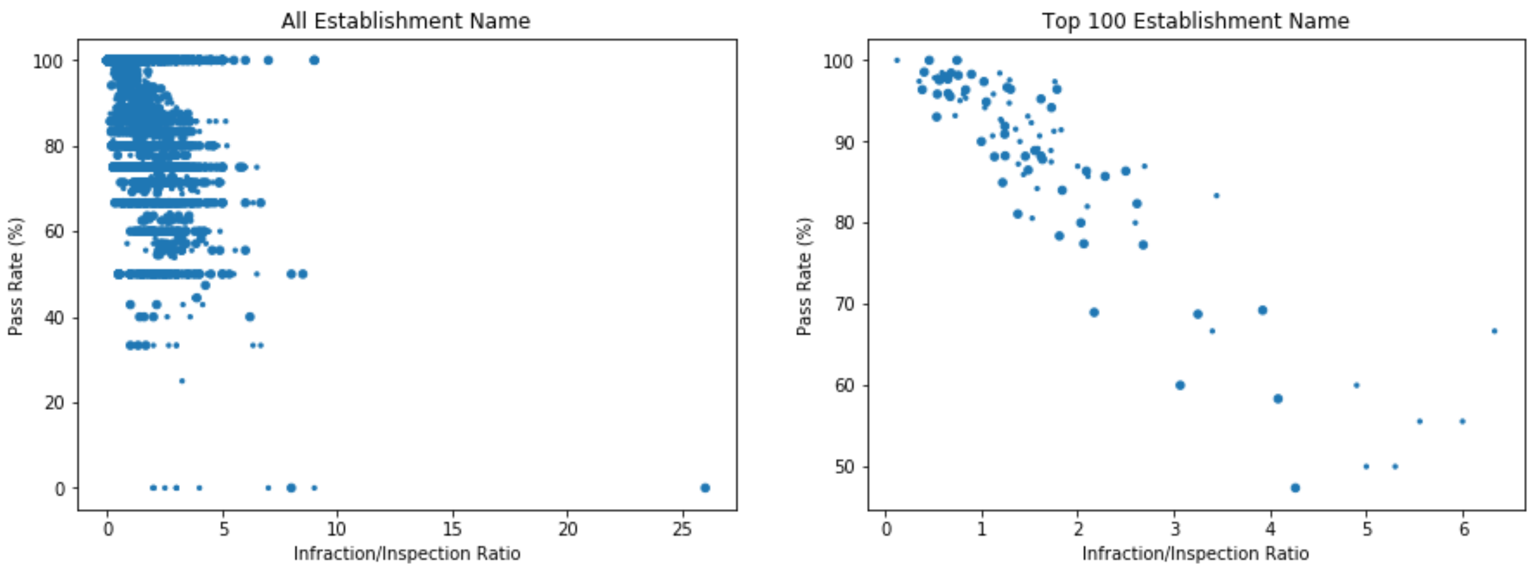
```
display(np.corrcoef(df_est_name_t100['ratio_infr_insp'],df_est_name_t100['pass_rate'])[0,1])
display(np.corrcoef(df_est_name_t1000['ratio_infr_insp'],df_est_name_t1000['pass_rate'])[0,1])
display(np.corrcoef(df_est_name_t10000['ratio_infr_insp'],df_est_name_t10000['pass_rate'])[0,1])
display(np.corrcoef(df_est_name_all['ratio_infr_insp'],df_est_name_all['pass_rate'])[0,1])
```

```
-0.9041595831737087

-0.4652592624941136

-0.4061134115416236

-0.4878948597623952
```

- Top 100 Establishment Name has strong negative correlation.

In [84]:

```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(15,5))

ax1 = fig.add_subplot(1,2,2)
plt.xlabel('Infraction/Inspection Ratio')
plt.ylabel('Pass Rate (%)')
plt.title('Top 100 Establishment Name', fontsize=12)

ax4 = fig.add_subplot(1,2,1)
plt.xlabel('Infraction/Inspection Ratio')
plt.ylabel('Pass Rate (%)')
plt.title('All Establishment Name', fontsize=12)

ax1.scatter(x = df_est_name_t100['ratio_infr_insp'], y = df_est_name_t100['pass_
rate'],sizes=(20,5))
ax4.scatter(x = df_est_name_all['ratio_infr_insp'], y = df_est_name_all['pass_ra
te'],sizes=(20,5))

plt.show()
```



In [85]:

```
print(np.corrcoef(df_est_name_all['ratio_infr_insp'],df_est_name_all['pass_rate'
])[0,1])
np.corrcoef(df_est_name_t100['ratio_infr_insp'],df_est_name_t100['pass_rate'])[0
,1]
```

```
-0.4878948597623952
```

Out[85]:

```
-0.9041595831737087
```

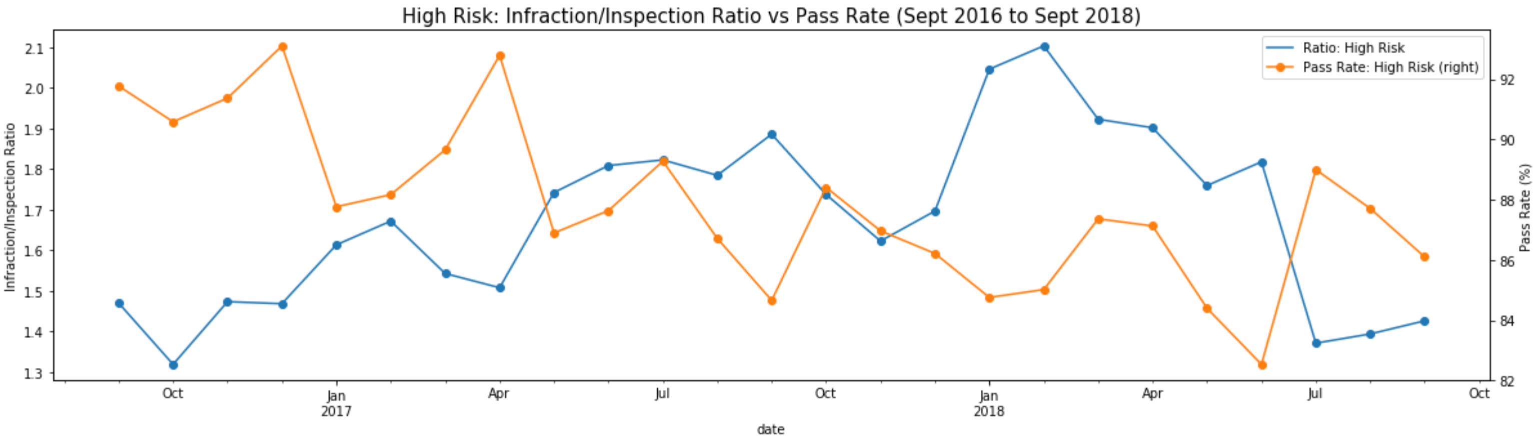
1.2. Infraction/Inspection Ratio vs Pass Rate - High Risk Establishment

In [86]:

```
ratio_infr_insp_plot('Ratio: High Risk')
pass_rate_plot_y('Pass Rate: High Risk')
plt.title('High Risk: Infraction/Inspection Ratio vs Pass Rate (Sept 2016 to Sep
t 2018)', fontsize=15)
```

Out[86]:

```
Text(0.5, 1.0, 'High Risk: Infraction/Inspection Ratio vs Pass Rate
(Sept 2016 to Sept 2018)')
```



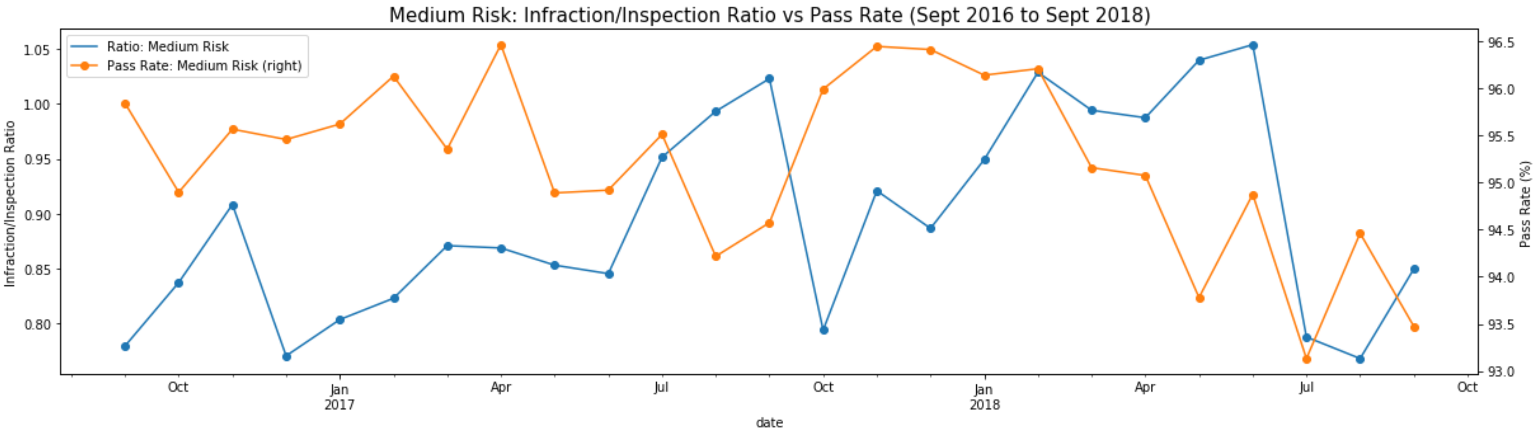
1.3. Infraction/Inspection Ratio vs Pass Rate - Medium Risk Establishment

In [87]:

```
ratio_infr_insp_plot('Ratio: Medium Risk')
pass_rate_plot_y('Pass Rate: Medium Risk')
plt.title('Medium Risk: Infraction/Inspection Ratio vs Pass Rate (Sept 2016 to S
ept 2018)', fontsize=15)
```

Out[87]:

```
Text(0.5, 1.0, 'Medium Risk: Infraction/Inspection Ratio vs Pass Rat
e (Sept 2016 to Sept 2018)')
```



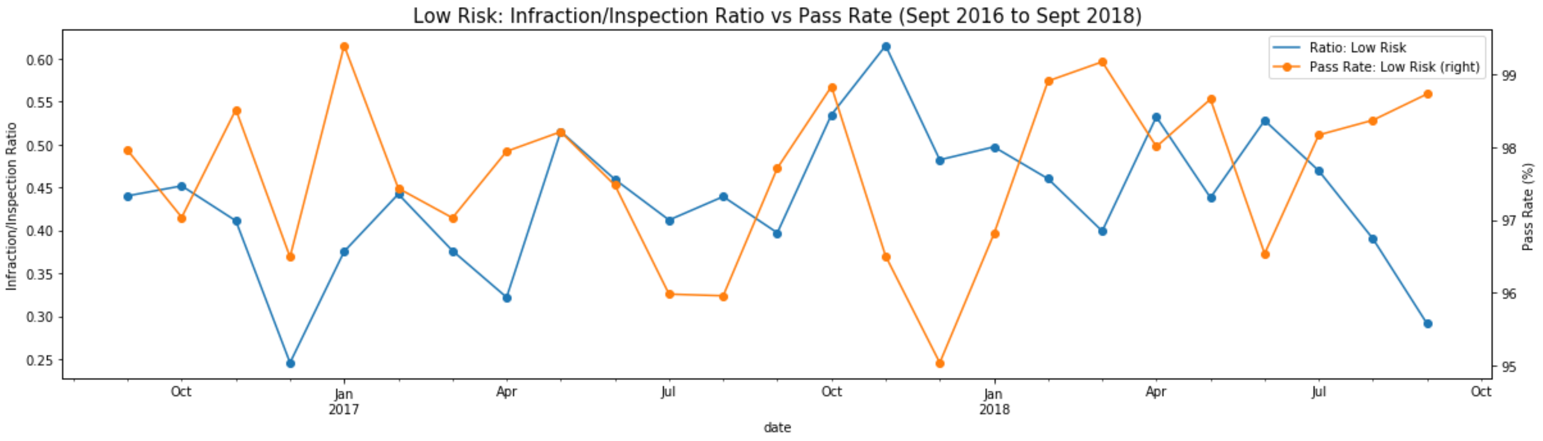
1.4. Infraction/Inspection Ratio vs Pass Rate - Low Risk Establishment

In [88]:

```
ratio_infr_insp_plot('Ratio: Low Risk')
pass_rate_plot_y('Pass Rate: Low Risk')
plt.title('Low Risk: Infraction/Inspection Ratio vs Pass Rate (Sept 2016 to Sept
2018)', fontsize=15)
```

Out[88]:

```
Text(0.5, 1.0, 'Low Risk: Infraction/Inspection Ratio vs Pass Rate (
Sept 2016 to Sept 2018)')
```



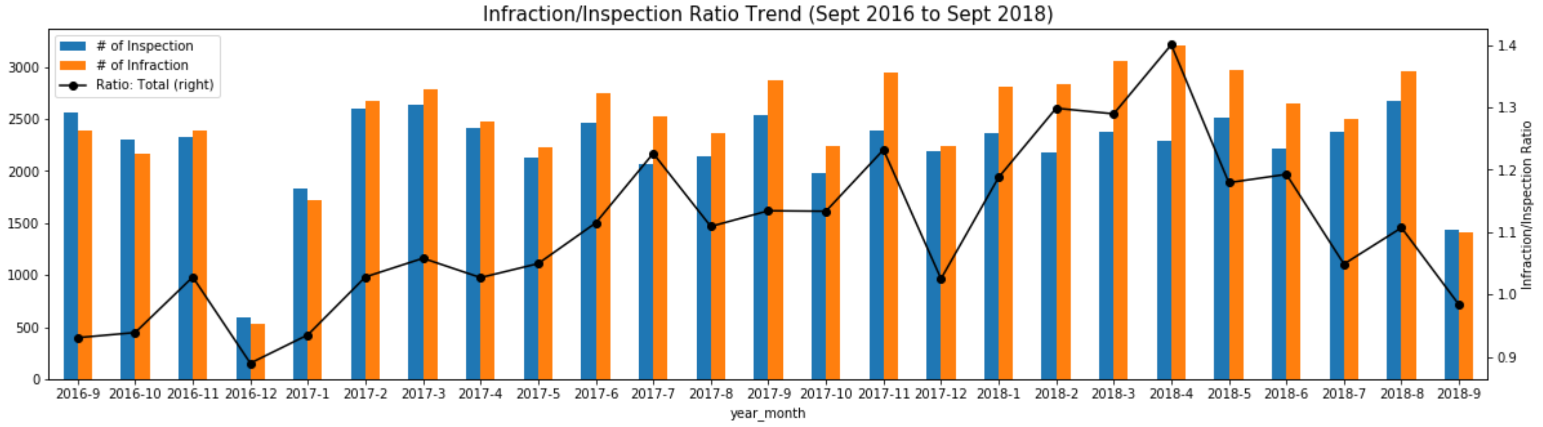
2. Infraction/Inspection Ratio Trend

In [89]:

```
infraction_ratio_bar('Inspection: Total','Infraction: Total','Ratio: Total')
plt.title('Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018)', fontsize
=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[89]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



- Infraction Ratio has been improved since April 2018.

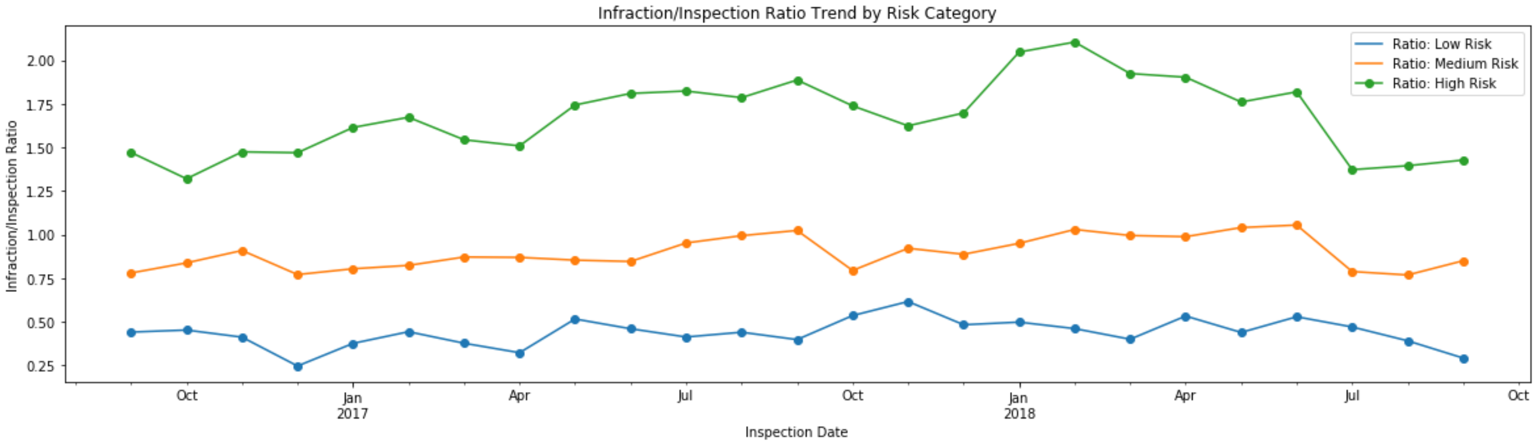
2.1. Infraction/Inspection Ratio Trend by Establishment Risk Category

In [90]:

```
ratio_infr_insp_plot('Ratio: Low Risk')
ratio_infr_insp_plot('Ratio: Medium Risk')
ratio_infr_insp_plot('Ratio: High Risk')
plt.title("Infraction/Inspection Ratio Trend by Risk Category")
```

Out[90]:

```
Text(0.5, 1.0, 'Infraction/Inspection Ratio Trend by Risk Category')
```



- High Risk establishments were driving the increases in infractions 2018Q1. Low and Medium Risk establishments indicate the stable infraction ratio trend.

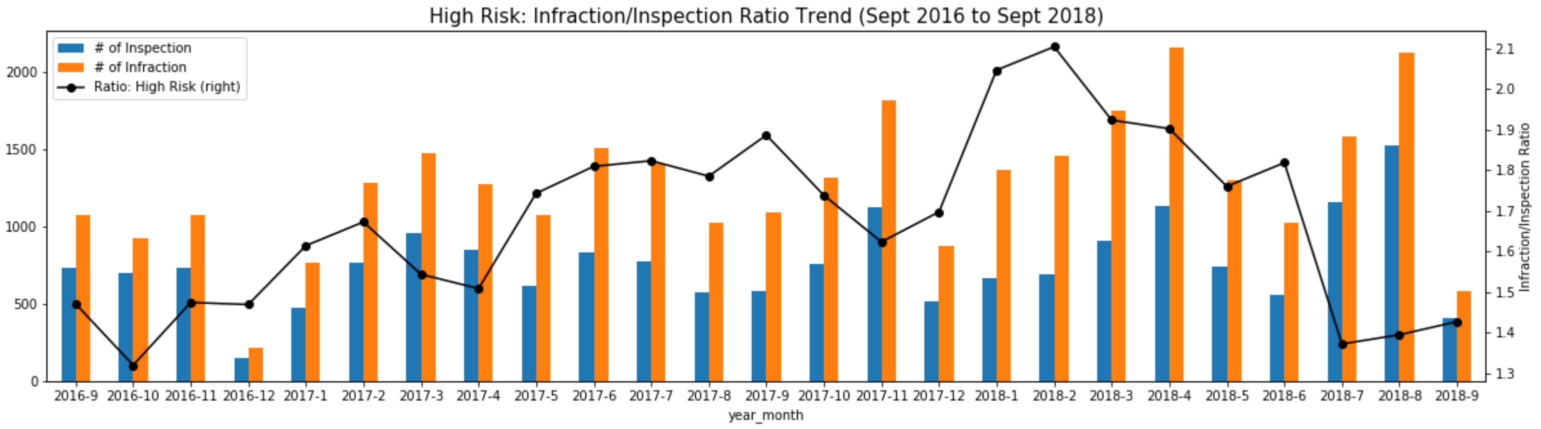
2.1.1. Infraction/Inspection Ratio Trend - High Risk Category

In [91]:

```
infraction_ratio_bar('Inspection: High Risk','Infraction: High Risk','Ratio: Hig
h Risk')
plt.title('High Risk: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018)
', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[91]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



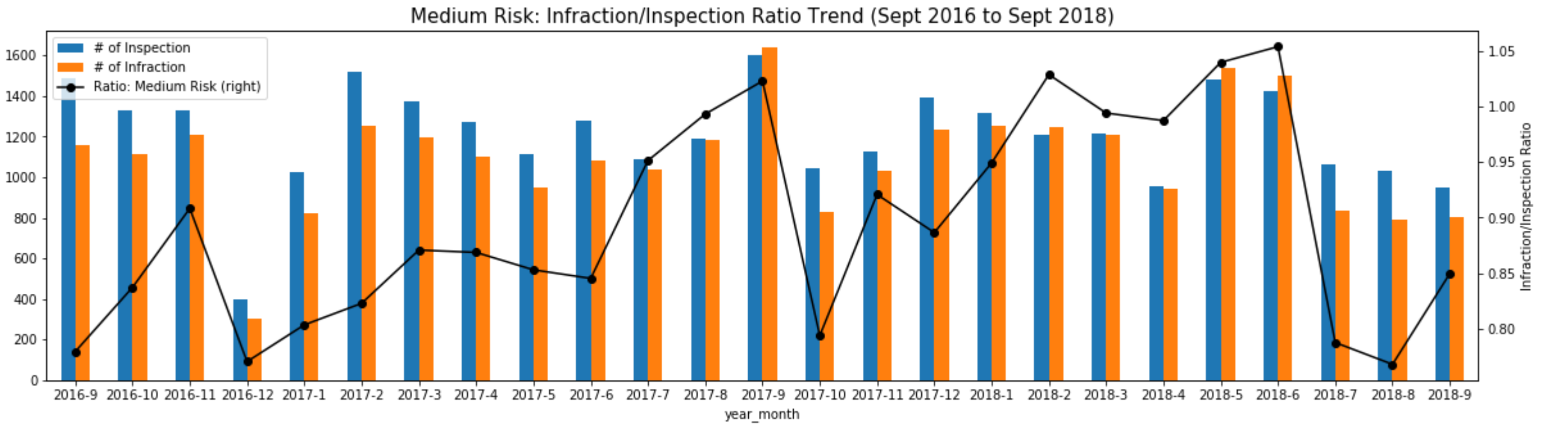
2.1.2. Infraction/Inspection Ratio Trend - Medium Risk Category

In [92]:

```
infraction_ratio_bar('Inspection: Medium Risk','Infraction: Medium Risk','Ratio:
Medium Risk')
plt.title('Medium Risk: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 201
8)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[92]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



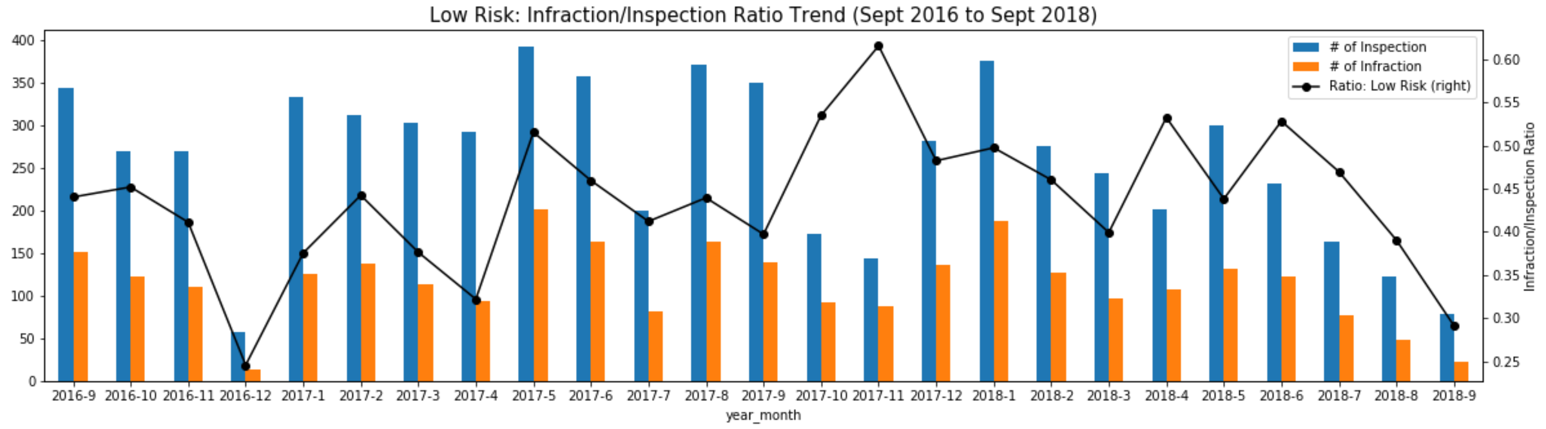
2.1.3. Infraction/Inspection Ratio Trend - Low Risk Category

In [93]:

```
infraction_ratio_bar('Inspection: Low Risk','Infraction: Low Risk','Ratio: Low R
isk')
plt.title('Low Risk: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018)'
, fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[93]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



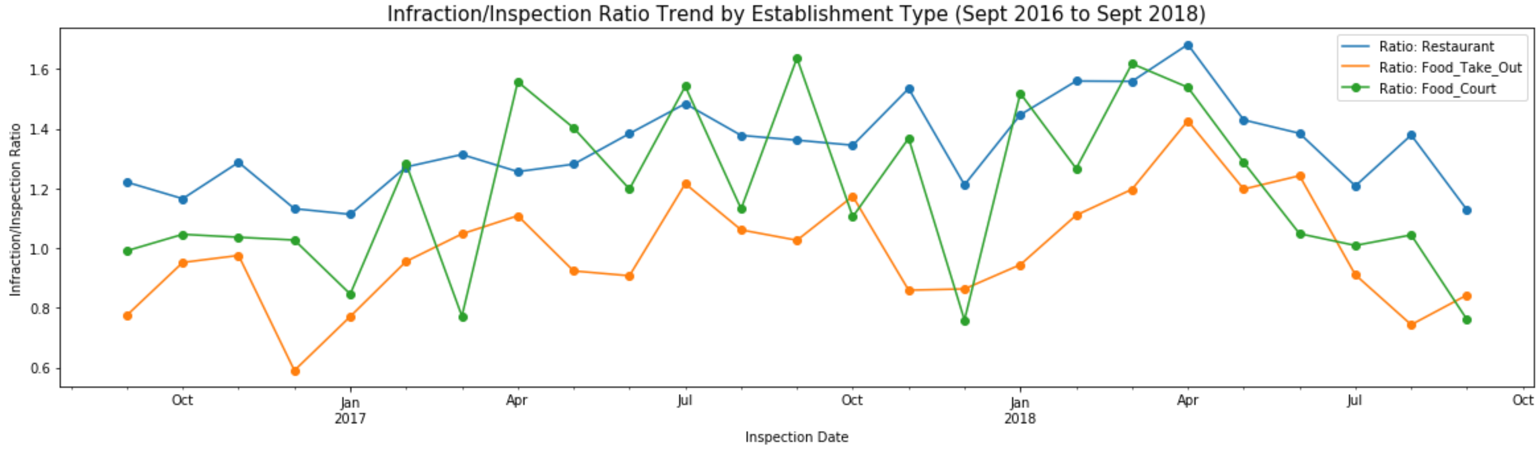
2.2. Infraction/Inspection Ratio by Establishment Type

In [94]:

```
### Figure 2.2a
ratio_infr_insp_plot('Ratio: Restaurant')
ratio_infr_insp_plot('Ratio: Food_Take_Out')
ratio_infr_insp_plot('Ratio: Food_Court')
#ratio_infr_insp_plot('Ratio: Sumpermarket')
#ratio_infr_insp_plot('Ratio: Bakery')
plt.title('Infraction/Inspection Ratio Trend by Establishment Type (Sept 2016 to
Sept 2018)', fontsize=15)
```

Out[94]:

```
Text(0.5, 1.0, 'Infraction/Inspection Ratio Trend by Establishment T
ype (Sept 2016 to Sept 2018)')
```

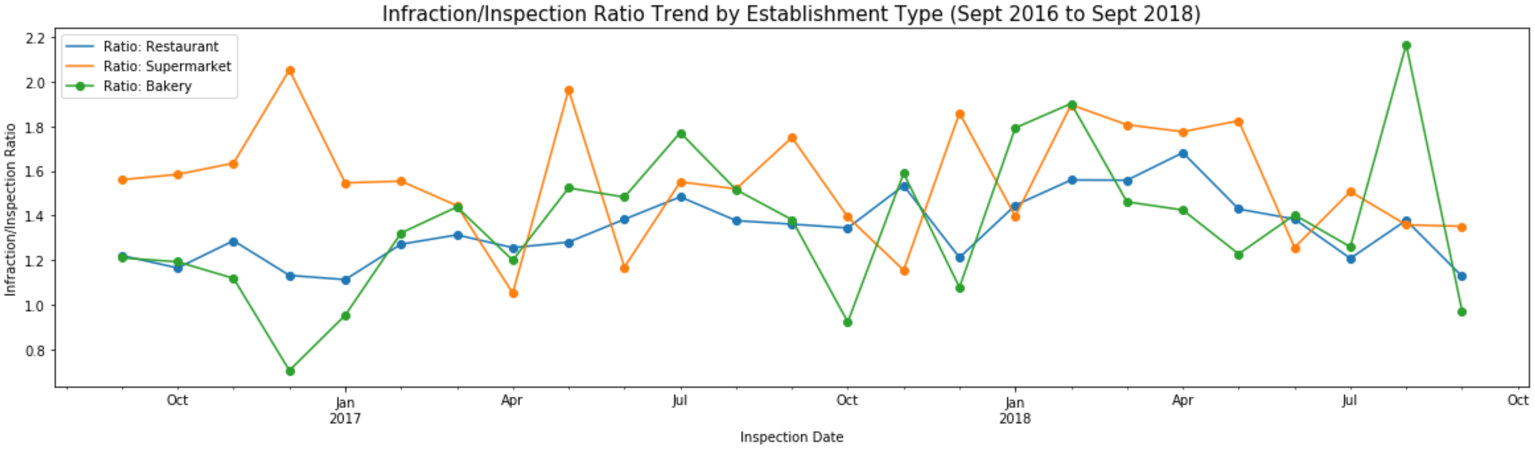


In [95]:

```
### Figure 2.2b
ratio_infr_insp_plot('Ratio: Restaurant')
#ratio_infr_insp_plot('Ratio: Food_Take_Out')
#ratio_infr_insp_plot('Ratio: Food_Court')
ratio_infr_insp_plot('Ratio: Supermarket')
ratio_infr_insp_plot('Ratio: Bakery')
plt.title('Infraction/Inspection Ratio Trend by Establishment Type (Sept 2016 to Sept 2018)', fontsize=15)
```

Out[95]:

```
Text(0.5, 1.0, 'Infraction/Inspection Ratio Trend by Establishment T
ype (Sept 2016 to Sept 2018)')
```



- Supermarket's nfraction/inspection ratio is the higest followed by Barkery's.

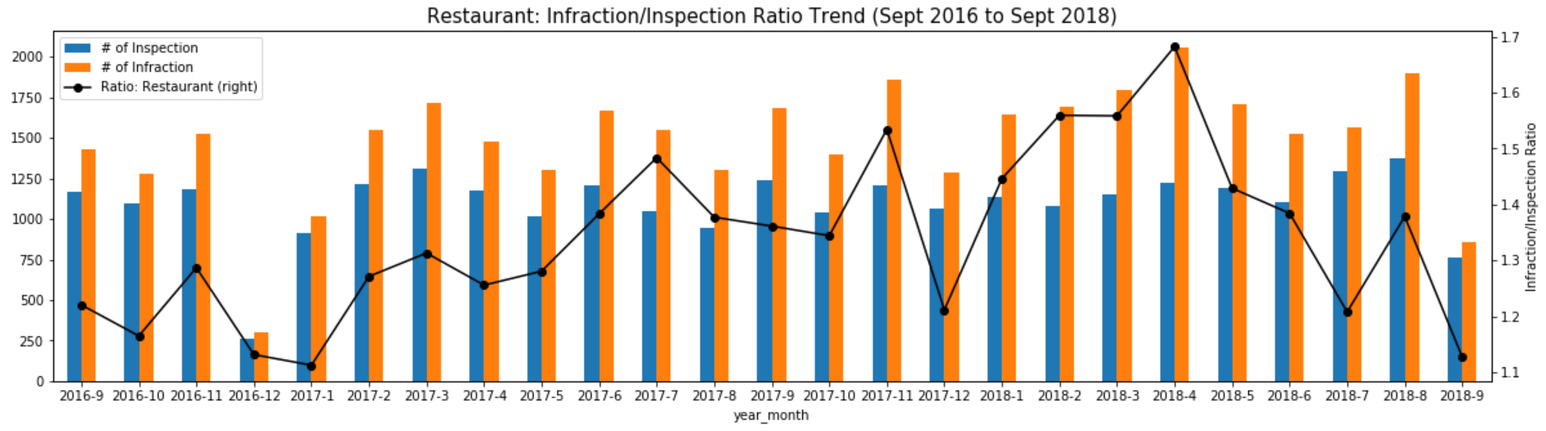
2.2.1. Infraction/Inspection Ratio by Establishment Type - Restaurant

In [96]:

```
infraction_ratio_bar('Inspection: Restaurant','Infraction: Restaurant','Ratio: Restaurant')
plt.title('Restaurant: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[96]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



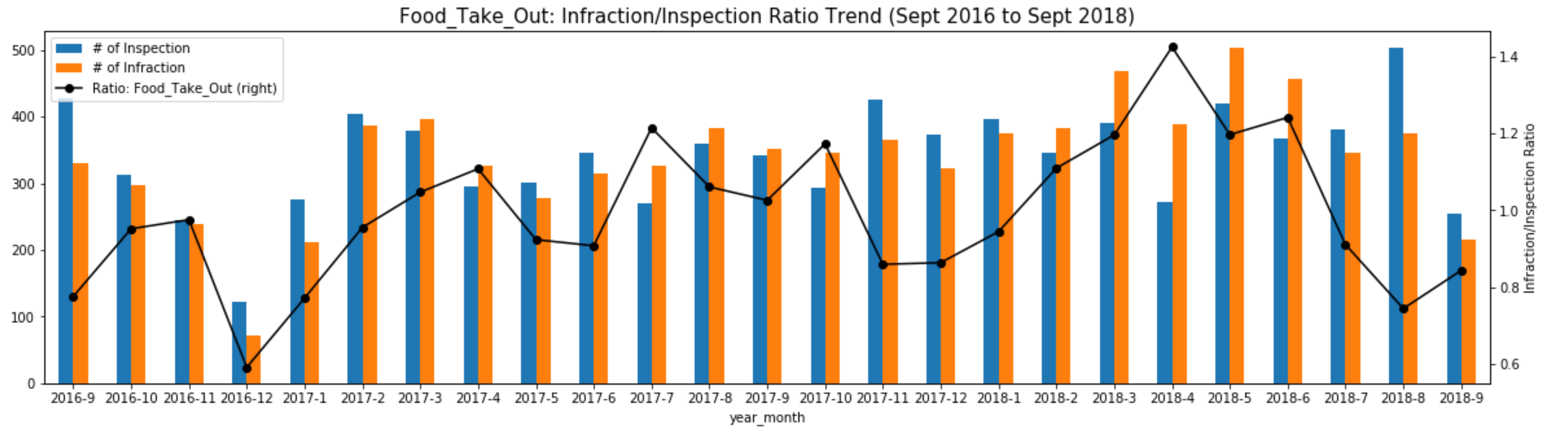
2.2.2. Infraction/Inspection Ratio by Establishment Type - Food Take Out

In [97]:

```
infraction_ratio_bar('Inspection: Food_Take_Out','Infraction: Food_Take_Out','Ratio: Food_Take_Out')
plt.title('Food_Take_Out: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[97]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



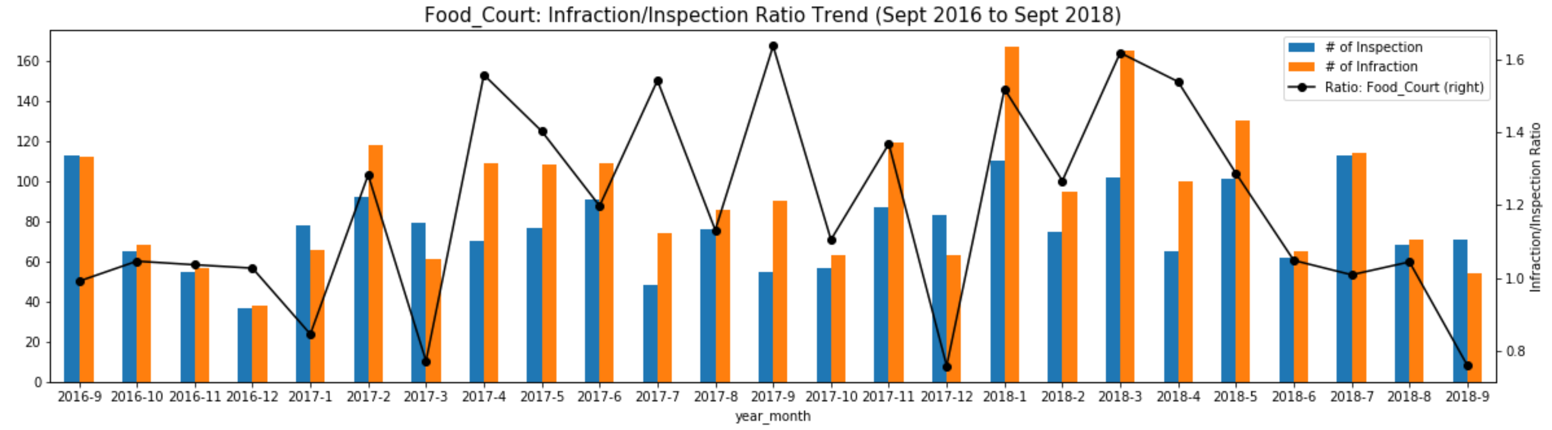
2.2.3. Infraction/Inspection Ratio by Establishment Type - Food Court

In [98]:

```
infraction_ratio_bar('Inspection: Food_Court','Infraction: Food_Court','Ratio: F
ood_Court')
plt.title('Food_Court: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018
)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[98]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



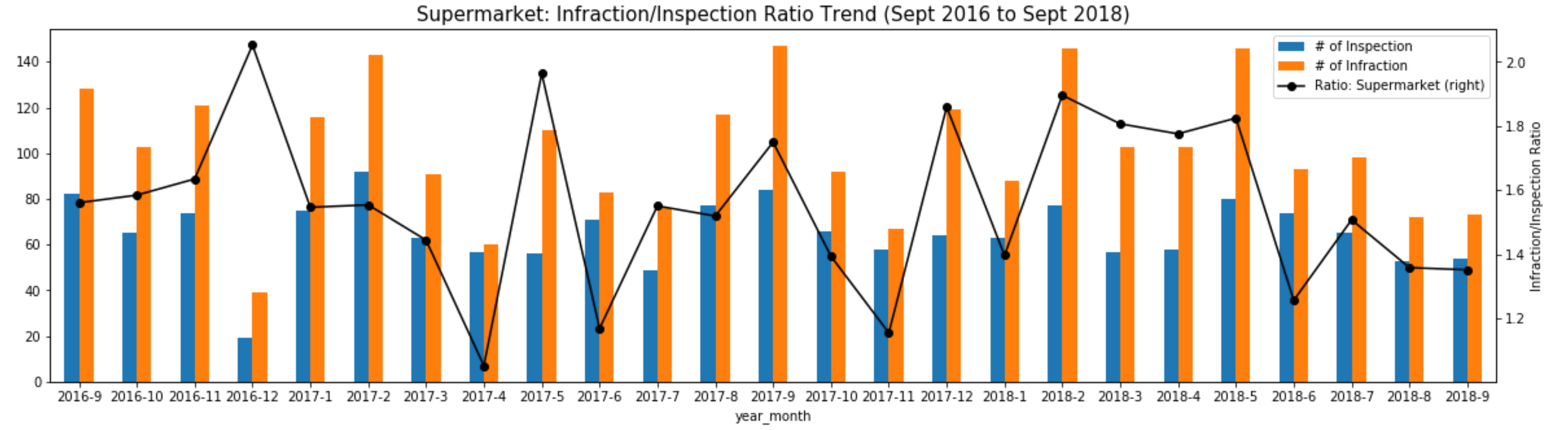
2.2.4. Infraction/Inspection Ratio by Establishment Type - Supermarket

In [99]:

```
infraction_ratio_bar('Inspection: Supermarket','Infraction: Supermarket','Ratio:
Supermarket')
plt.title('Supermarket: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 201
8)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[99]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```

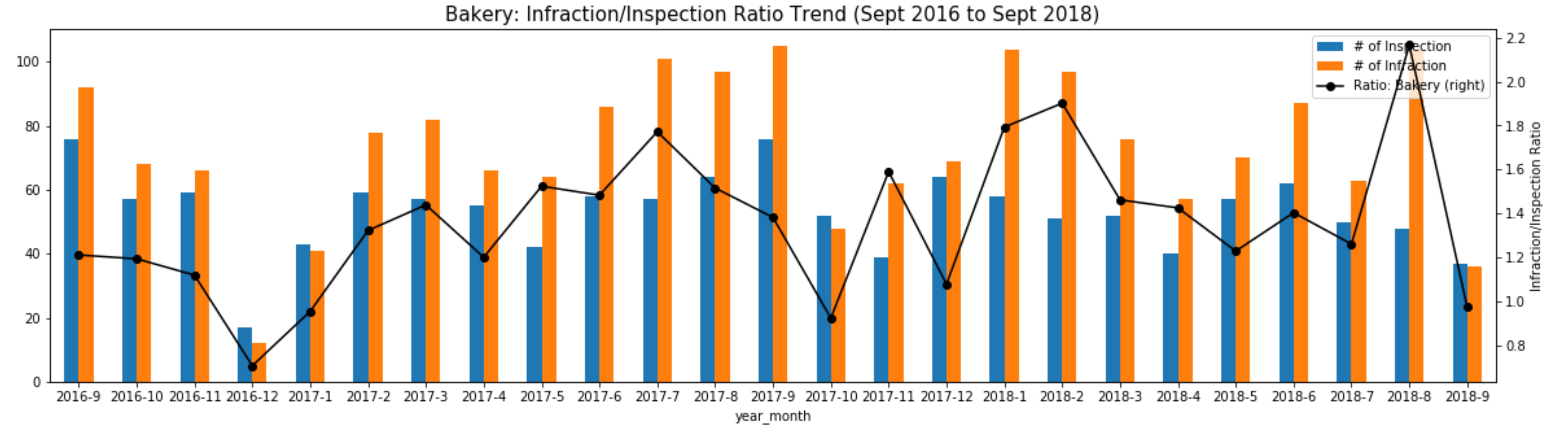


2.2.5. Infraction/Inspection Ratio by Establishment Type - Bakery

```
In [100]:
```

```
infraction_ratio_bar('Inspection: Bakery','Infraction: Bakery','Ratio: Bakery')
plt.title('Bakery: Infraction/Inspection Ratio Trend (Sept 2016 to Sept 2018)',
fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

```
Out[100]:

Text(0, 0.5, 'Infraction/Inspection Ratio')
```



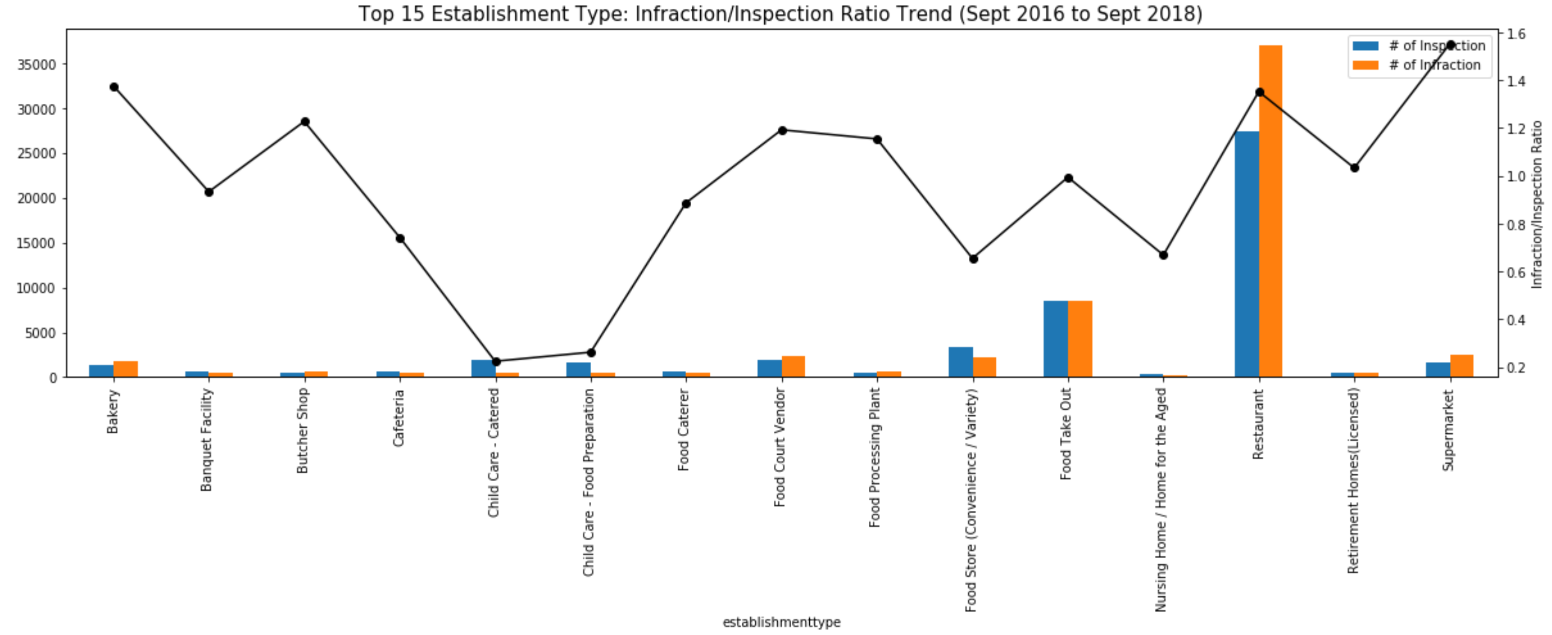
2.3. Infraction/Inspection Ratio on Top 15 Establishment Type

In [101]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_type['insp'],
        '# of Infraction': df_est_type['infr']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_type['ratio_infr_insp'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)

plt.title('Top 15 Establishment Type: Infraction/Inspection Ratio Trend (Sept 20
16 to Sept 2018)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[101]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



- Child care has the lowest infraction ratio while supermarket has the highest.

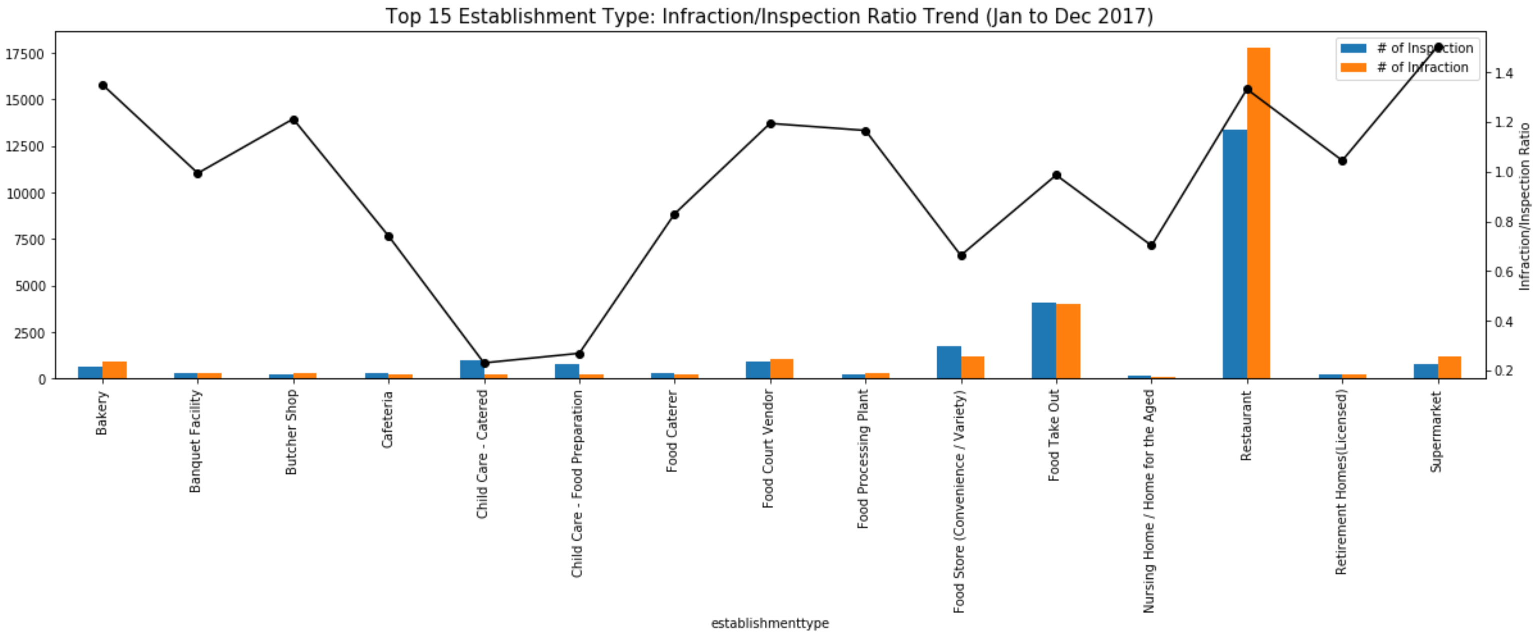
2.3.1 Infraction/Inspection Ratio on Top 15 Establishment Type - 2017CY

In [102]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_type['insp_17'],
        '# of Infraction': df_est_type['infr_17']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_type['ratio_infr_insp_17'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Top 15 Establishment Type: Infraction/Inspection Ratio Trend (Jan to
Dec 2017)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[102]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



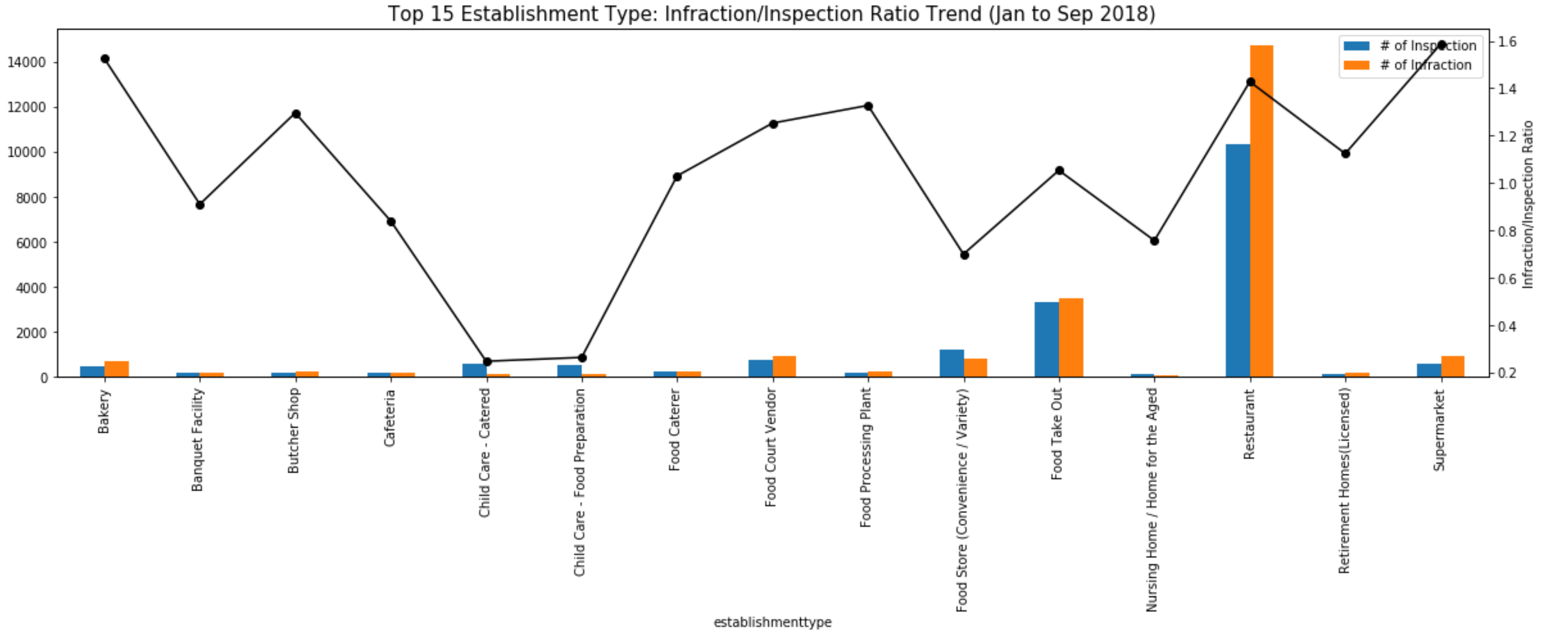
2.3.2 Infraction/Inspection Ratio on Top 15 Establishment Type - 2018CY

In [103]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_type['insp_18'],
        '# of Infraction': df_est_type['infr_18']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_type['ratio_infr_insp_18'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Top 15 Establishment Type: Infraction/Inspection Ratio Trend (Jan to
Sep 2018)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[103]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



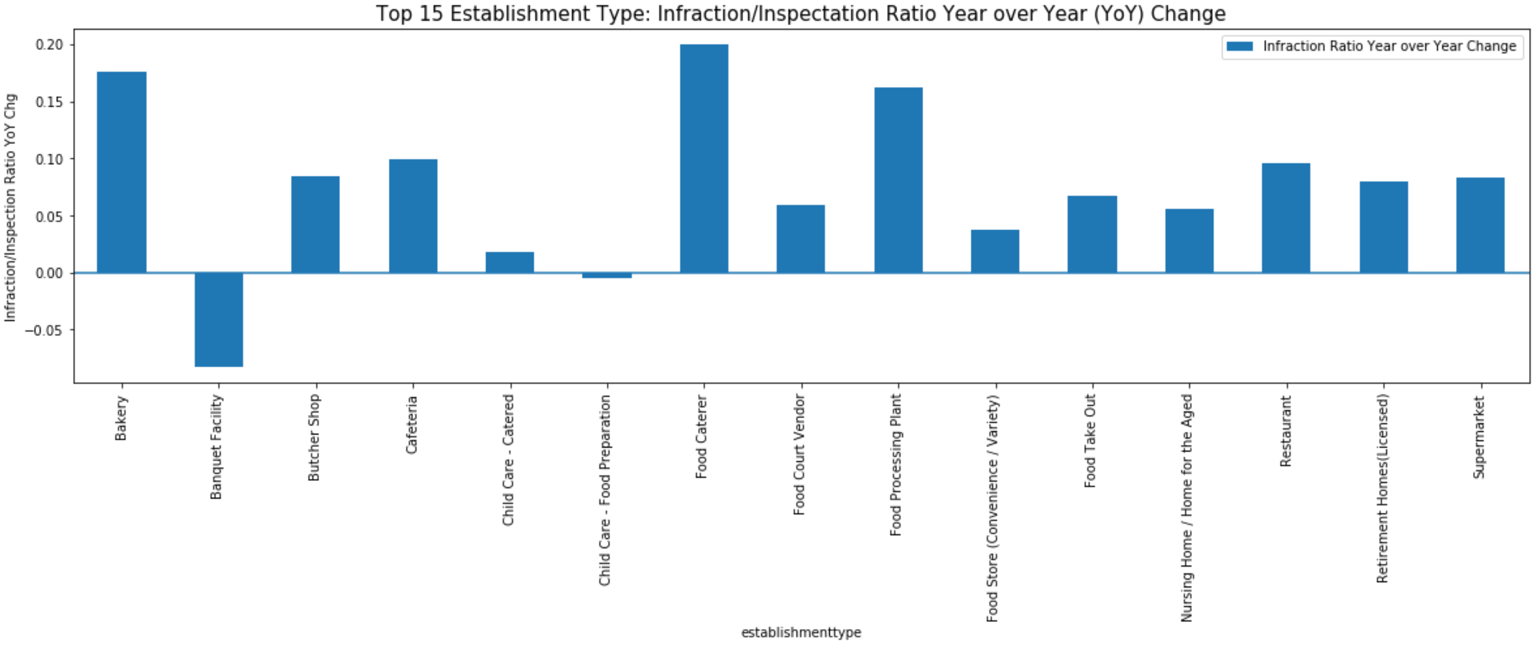
2.3.3 Infraction/Inspecation Ratio Year over Year (YoY) Change

In [104]:

```
df_grf = pd.DataFrame({'Infraction Ratio Year over Year Change':df_est_type['rat
io_infr_insp_18'] - df_est_type['ratio_infr_insp_17']})
ax1 = df_grf.plot.bar(figsize=(20,5))
plt.axhline(0)
plt.title('Top 15 Establishment Type: Infraction/Inspectation Ratio Year over Ye
ar (YoY) Change', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio YoY Chg')
```

Out[104]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio YoY Chg')
```



- Overall increases in top 15 establishment type
- Banquet Facility decreased by 0.05
- Food Caterer increased by 0.20

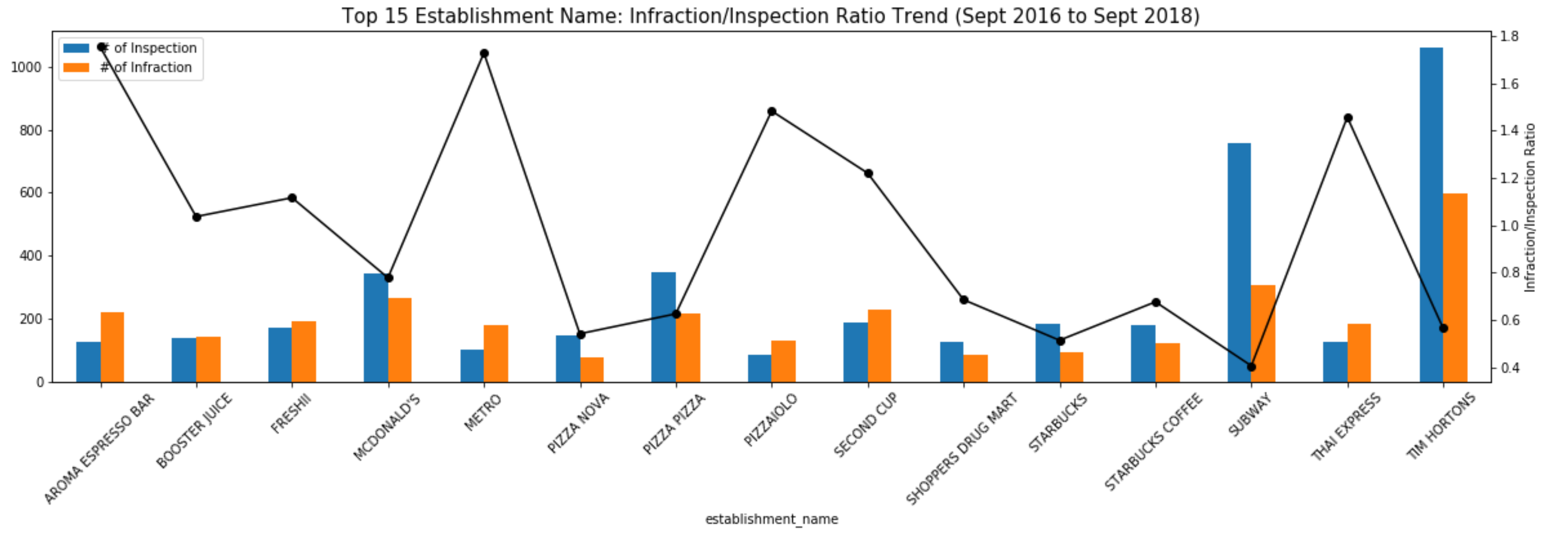
2.4. Infraction/Inspection Ratio on Top 15 Establishment Name

In [105]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_name['insp'],
        '# of Infraction': df_est_name['infr']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_name['ratio_infr_insp'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(45)
plt.title('Top 15 Establishment Name: Infraction/Inspection Ratio Trend (Sept 20
16 to Sept 2018)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[105]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



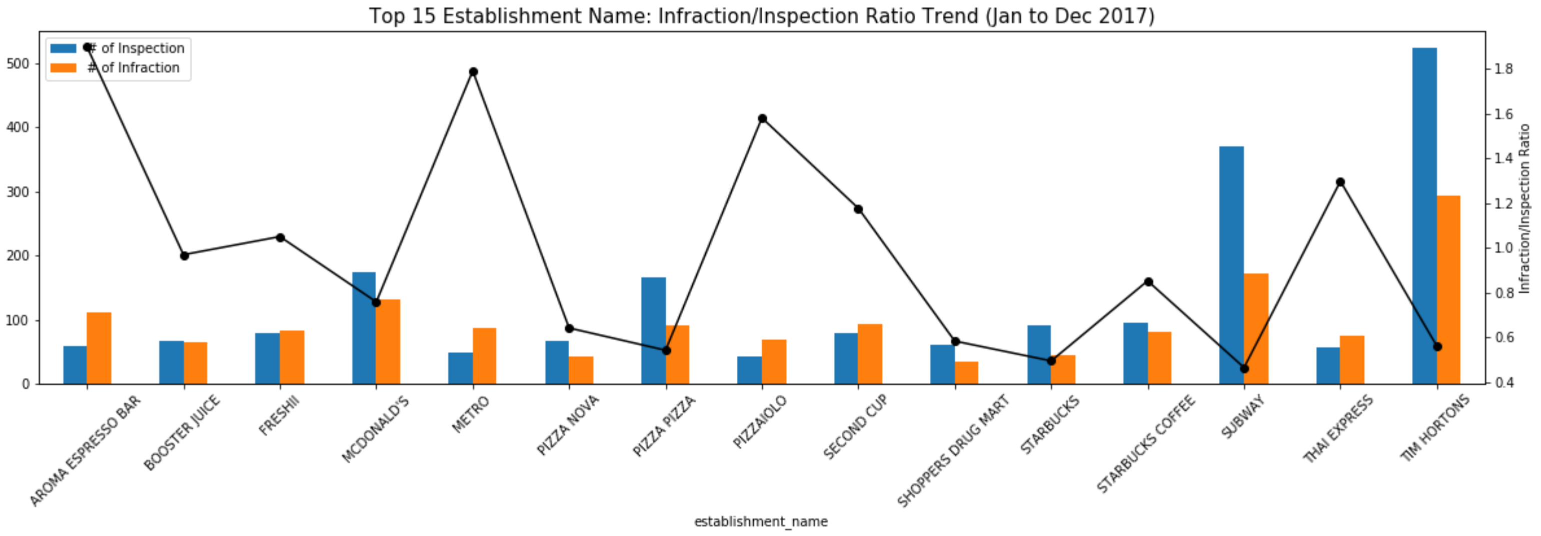
2.4.1. Infraction/Inspection Ratio on Top 15 Establishment Name - 2017CY

In [106]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_name['insp_17'],
        '# of Infraction': df_est_name['infr_17']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_name['ratio_infr_insp_17'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(45)
plt.title('Top 15 Establishment Name: Infraction/Inspection Ratio Trend (Jan to
Dec 2017)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[106]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```

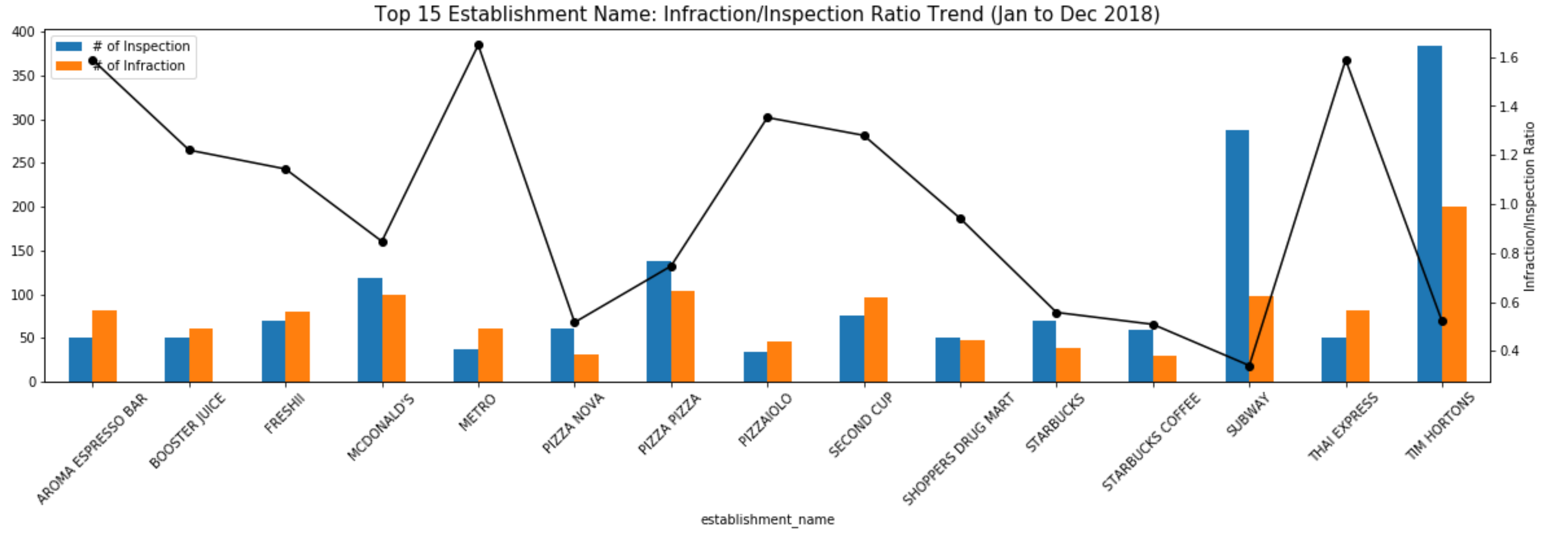


2.4.2. Infraction/Inspection Ratio on Top 15 Establishment Name - 2018CY

```
In [107]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_name['insp_18'],
        '# of Infraction': df_est_name['infr_18']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_name['ratio_infr_insp_18'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(45)
plt.title('Top 15 Establishment Name: Infraction/Inspection Ratio Trend (Jan to
Dec 2018)', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

```
Out[107]:

Text(0, 0.5, 'Infraction/Inspection Ratio')
```



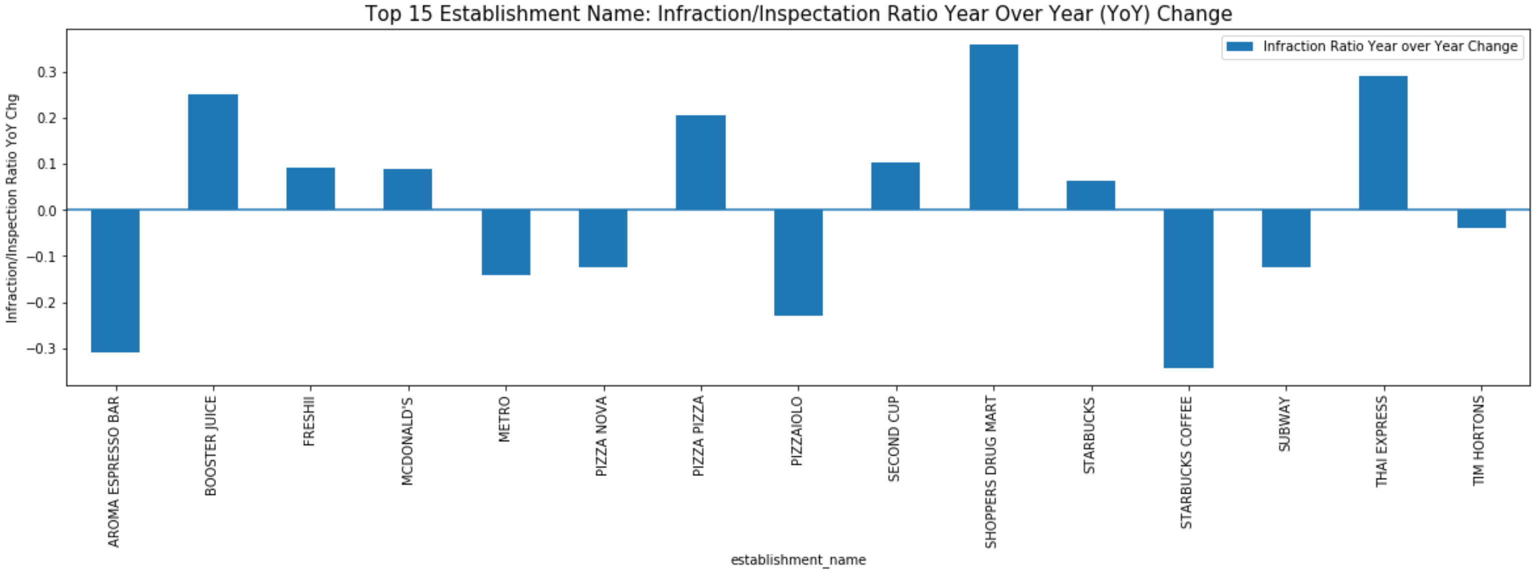
2.4.3. Infraction/Inspection Ratio Year over Year (YoY) Change - Top 15 Establishment Name

In [108]:

```
df_grf = pd.DataFrame({'Infraction Ratio Year over Year Change':df_est_name['rat
io_infr_insp_18'] - df_est_name['ratio_infr_insp_17']})
ax1 = df_grf.plot.bar(figsize=(20,5))
plt.axhline(0)
plt.title('Top 15 Establishment Name: Infraction/Inspectation Ratio Year Over Ye
ar (YoY) Change', fontsize=15)
plt.ylabel('Infraction/Inspection Ratio YoY Chg')
```

Out[108]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio YoY Chg')
```

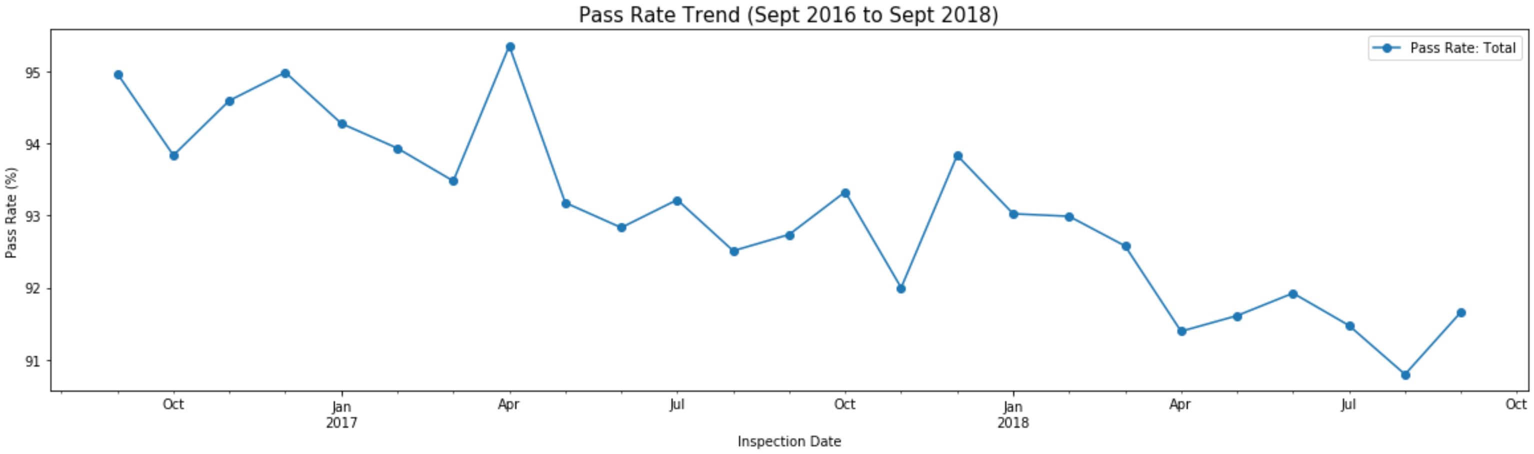


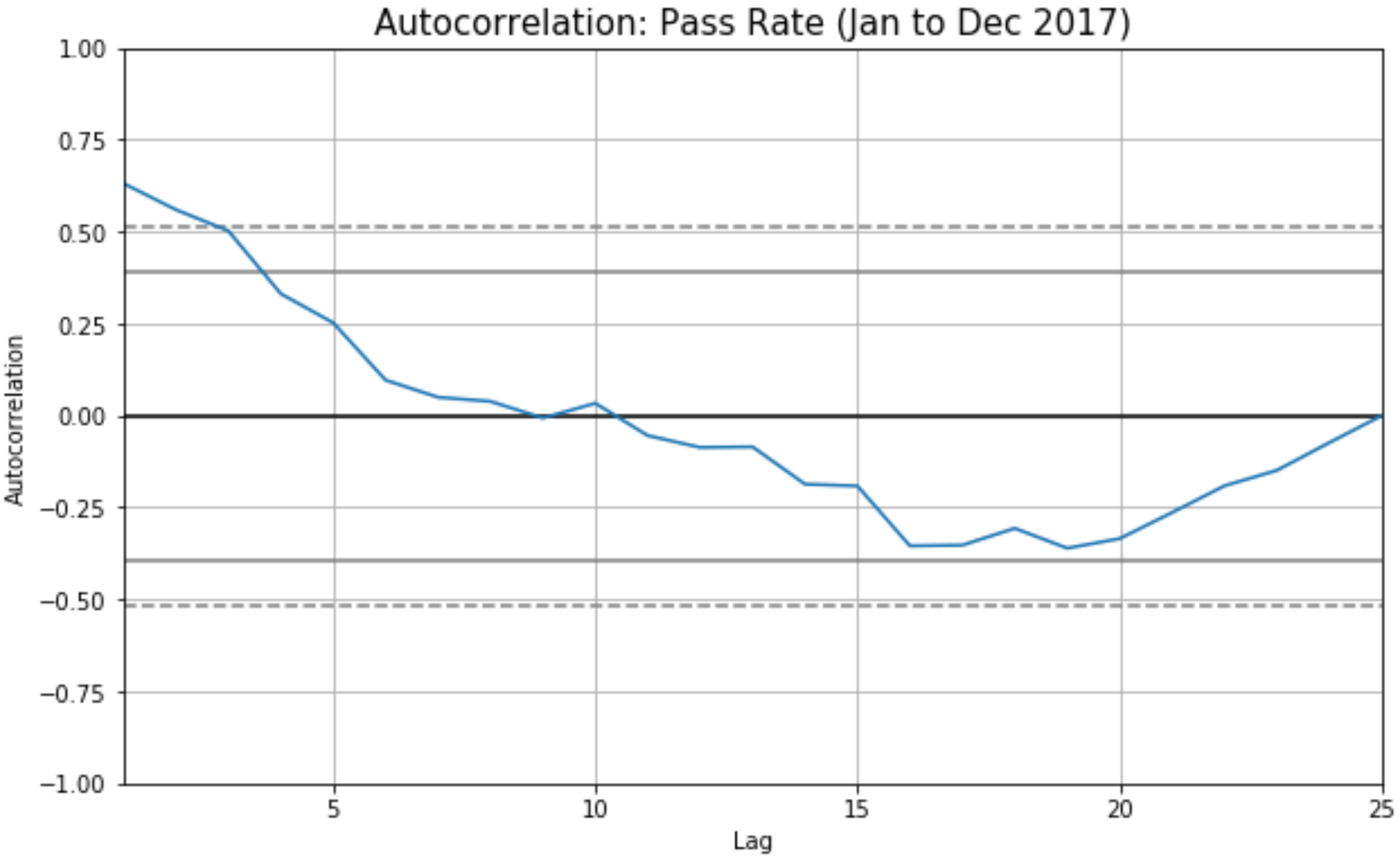
- Aroma Expresso Bar and Pizzailolo improved infraction ratio.

3. Pass Rate (Compliance Rate)

In [110]:

```
#overall pass rate
pass_rate_plot('Pass Rate: Total')
plt.title('Pass Rate Trend (Sept 2016 to Sept 2018)', fontsize=15)
plt.show()

#autocorrelation
cnt_pass = df[df['establishment_status']=='Pass'].groupby(['year_month']).inspec
tion_id.nunique()
cnt_insp = df.groupby(['year_month']).inspection_id.nunique()
ts = cnt_pass/cnt_insp
autocorrelation_plot(ts)
plt.title('Autocorrelation: Pass Rate (Jan to Dec 2017)', fontsize=15)
plt.show()
```



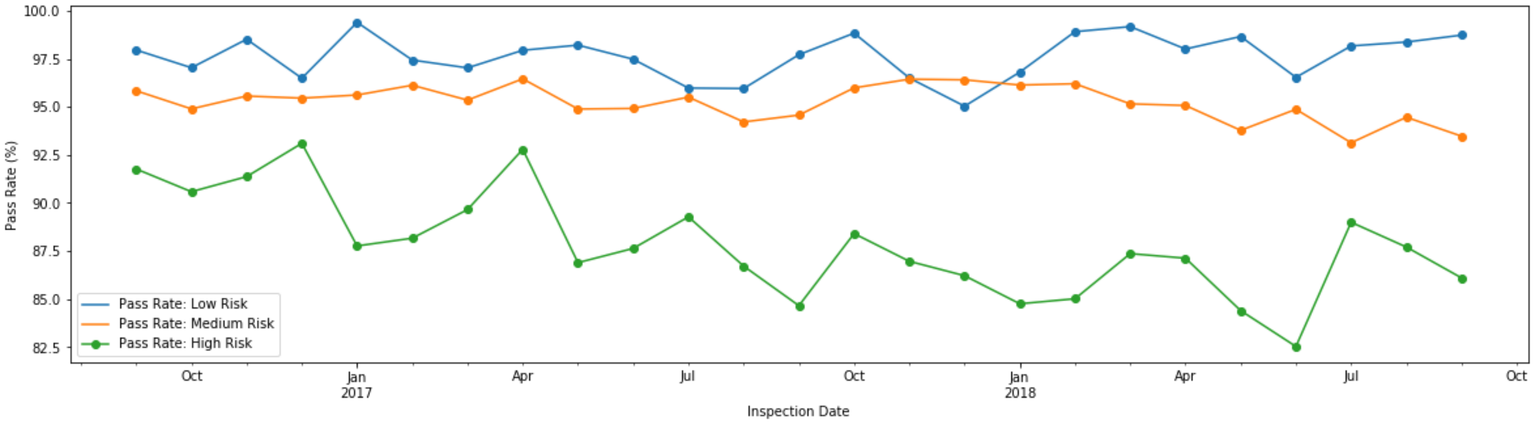


- Overall pass rate is trennding down.
- No autocorrelation

3.1. Pass Rate Trend by Risk Category

In [111]:

```
pass_rate_plot('Pass Rate: Low Risk')
pass_rate_plot('Pass Rate: Medium Risk')
pass_rate_plot('Pass Rate: High Risk')
```



- Pass rates of High Risk establishments are trending down while Low and medium risk establishments are stable.

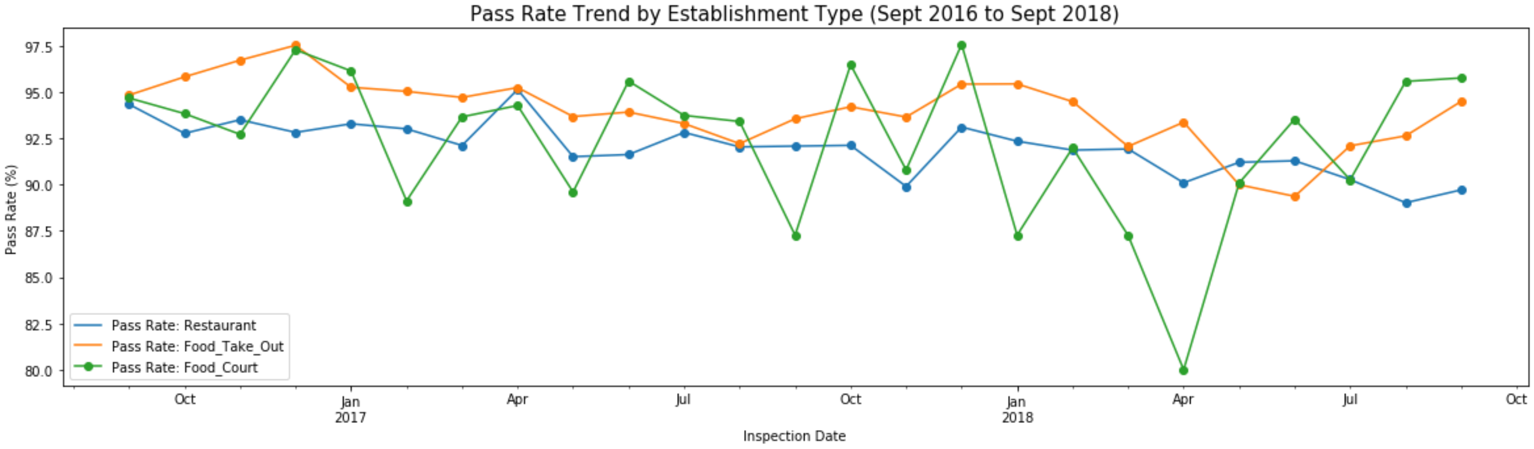
3.2. Pass Rate Trend by Establishment Type

In [112]:

```
pass_rate_plot('Pass Rate: Restaurant')
pass_rate_plot('Pass Rate: Food_Take_Out')
pass_rate_plot('Pass Rate: Food_Court')
#pass_rate_plot('Pass Rate: Sumpermarket')
#pass_rate_plot('Pass Rate: Bakery')
plt.title('Pass Rate Trend by Establishment Type (Sept 2016 to Sept 2018)', font
size=15)
```

Out[112]:

```
Text(0.5, 1.0, 'Pass Rate Trend by Establishment Type (Sept 2016 to
Sept 2018)')
```



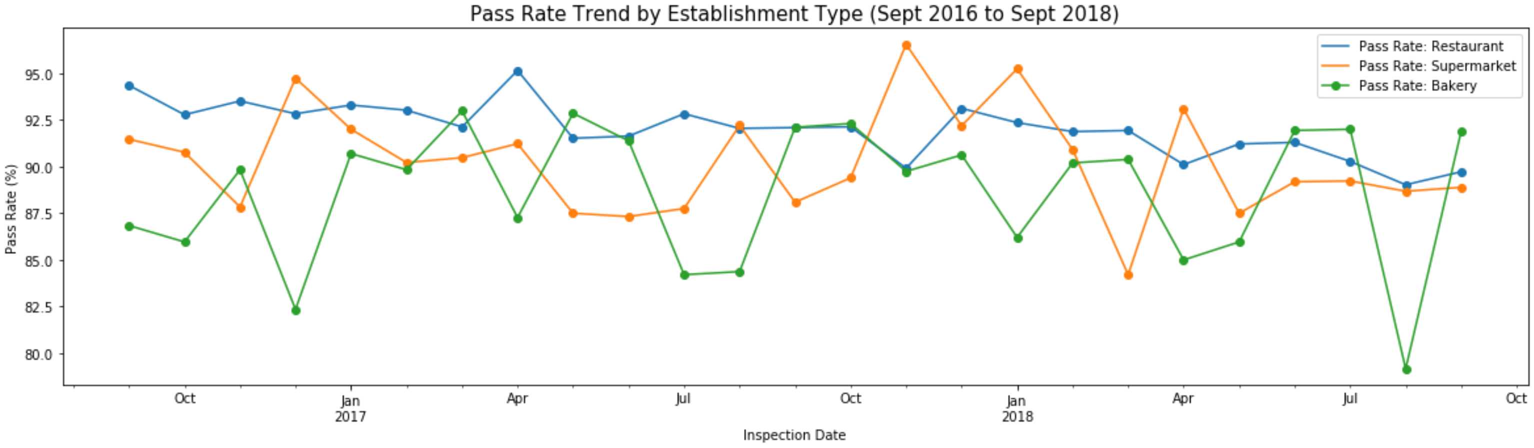
- 'Food Take Out' places had less infraction ratio and more stable ratio than restaurant and Food Court.

In [113]:

```
pass_rate_plot('Pass Rate: Restaurant')
#pass_rate_plot('Pass Rate: Food_Take_Out')
#pass_rate_plot('Pass Rate: Food_Court')
pass_rate_plot('Pass Rate: Supermarket')
pass_rate_plot('Pass Rate: Bakery')
plt.title('Pass Rate Trend by Establishment Type (Sept 2016 to Sept 2018)', font
size=15)
```

Out[113]:

```
Text(0.5, 1.0, 'Pass Rate Trend by Establishment Type (Sept 2016 to
Sept 2018)')
```

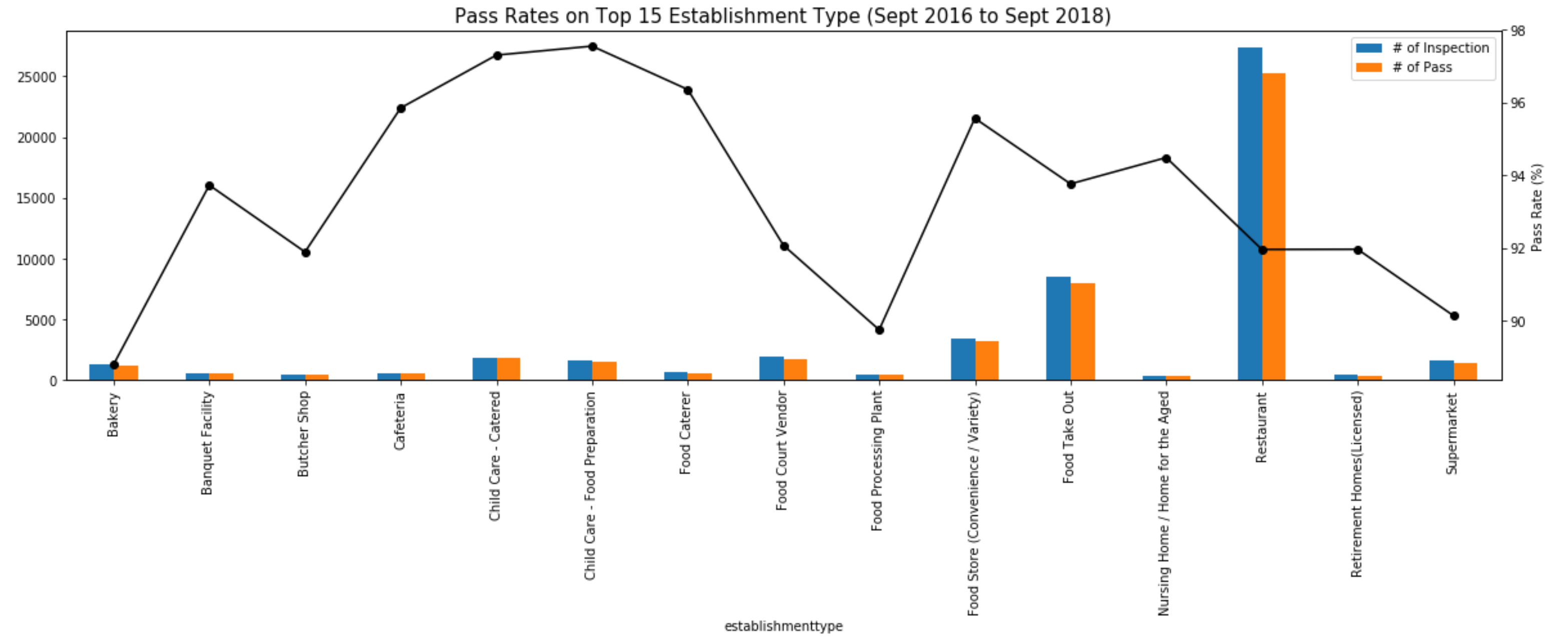


3.3. Pass Rates on Top 15 Establishment Type

```
In [114]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_type['insp'],
        '# of Pass': df_est_type['pass']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_type['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Establishment Type (Sept 2016 to Sept 2018)', fo
ntsize=15)
plt.ylabel('Pass Rate (%)')
```

```
Out[114]:

Text(0, 0.5, 'Pass Rate (%)')
```



- Child care, Cafeteria, and Food Caterer had higher pass rates than others while Bakery and Supermarket had the lower pass rates.

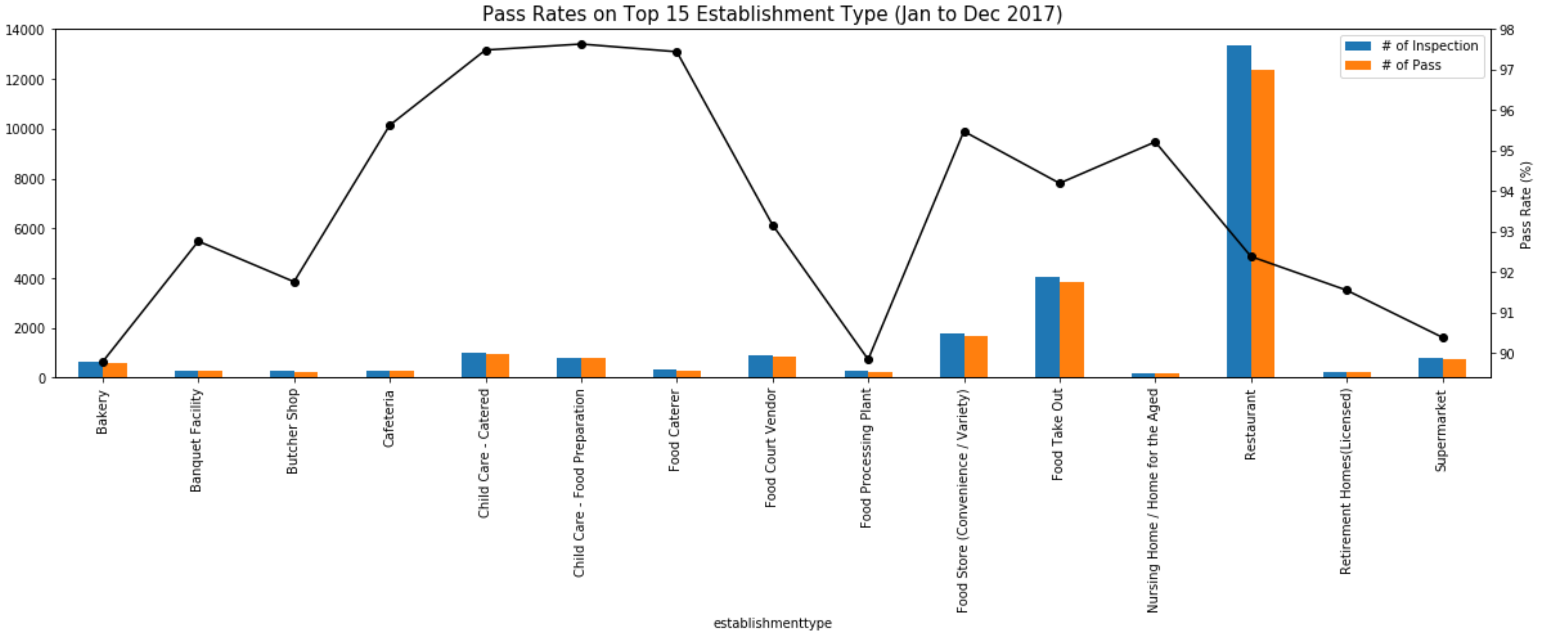
3.3.1. Pass Rates on Top 15 Establishment Type - 2017CY

In [115]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_type['insp_17'],
        '# of Pass': df_est_type['pass_17']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_type['pass_rate_17'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Establishment Type (Jan to Dec 2017)', fontsize=
15)
plt.ylabel('Pass Rate (%)')
```

Out[115]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



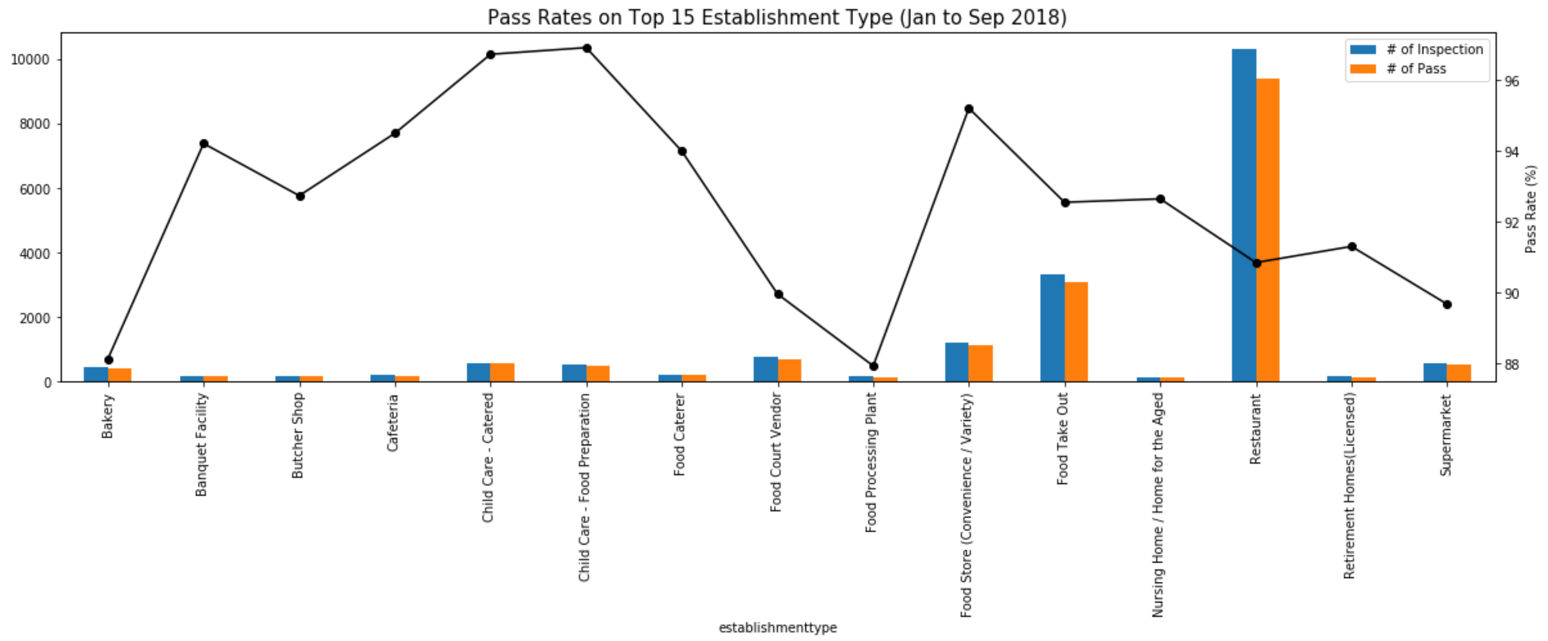
3.3.2. Pass Rates on Top 15 Establishment Type - 2018CY

In [116]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_type['insp_18'],
        '# of Pass': df_est_type['pass_18']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_type['pass_rate_18'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Establishment Type (Jan to Sep 2018)', fontsize=
15)
plt.ylabel('Pass Rate (%)')
```

Out[116]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



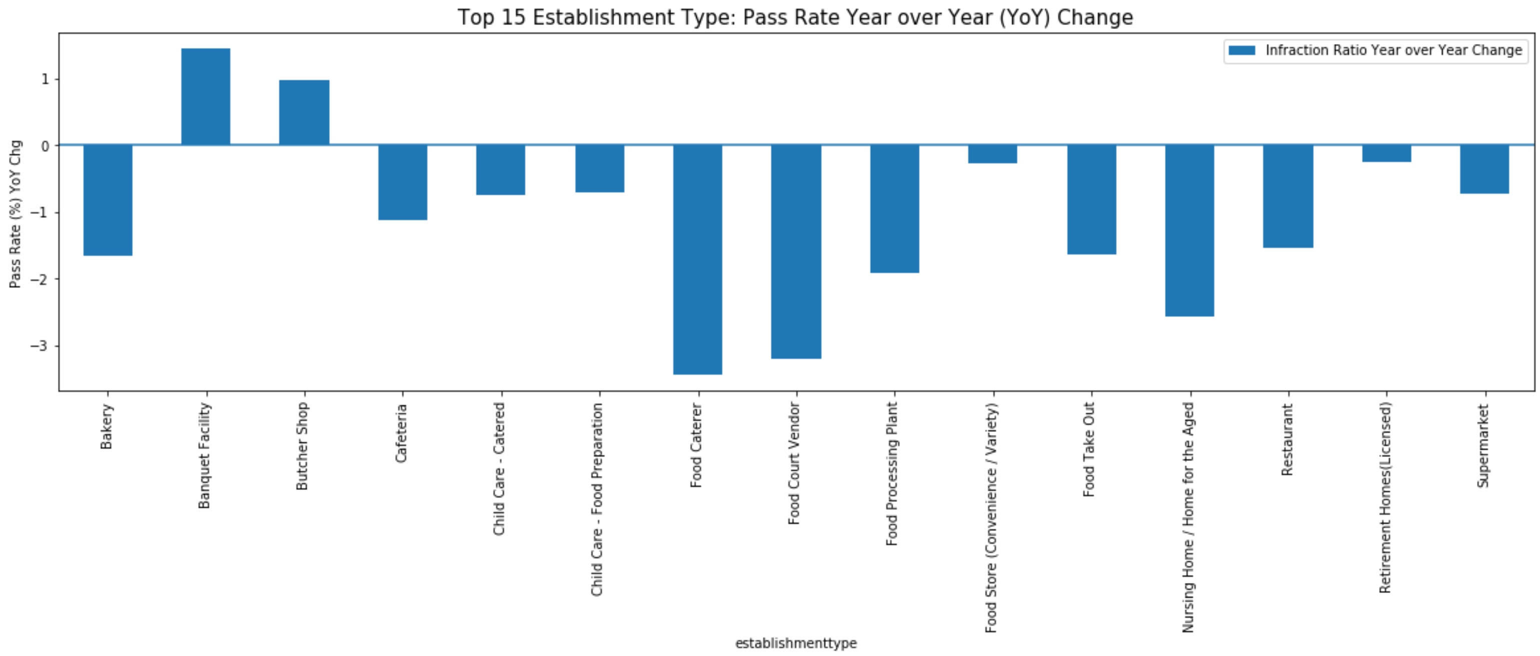
3.3.3. Pass Rate Year over Year (YoY) Change - Top 15 Establishment Type

In [117]:

```
df_grf = pd.DataFrame({'Infraction Ratio Year over Year Change':df_est_type['pass_rate_18'] - df_est_type['pass_rate_17']})
ax1 = df_grf.plot.bar(figsize=(20,5))
plt.axhline(0)
plt.title('Top 15 Establishment Type: Pass Rate Year over Year (YoY) Change', fontsize=15)
plt.ylabel('Pass Rate (%) YoY Chg')
```

Out[117]:

```
Text(0, 0.5, 'Pass Rate (%) YoY Chg')
```



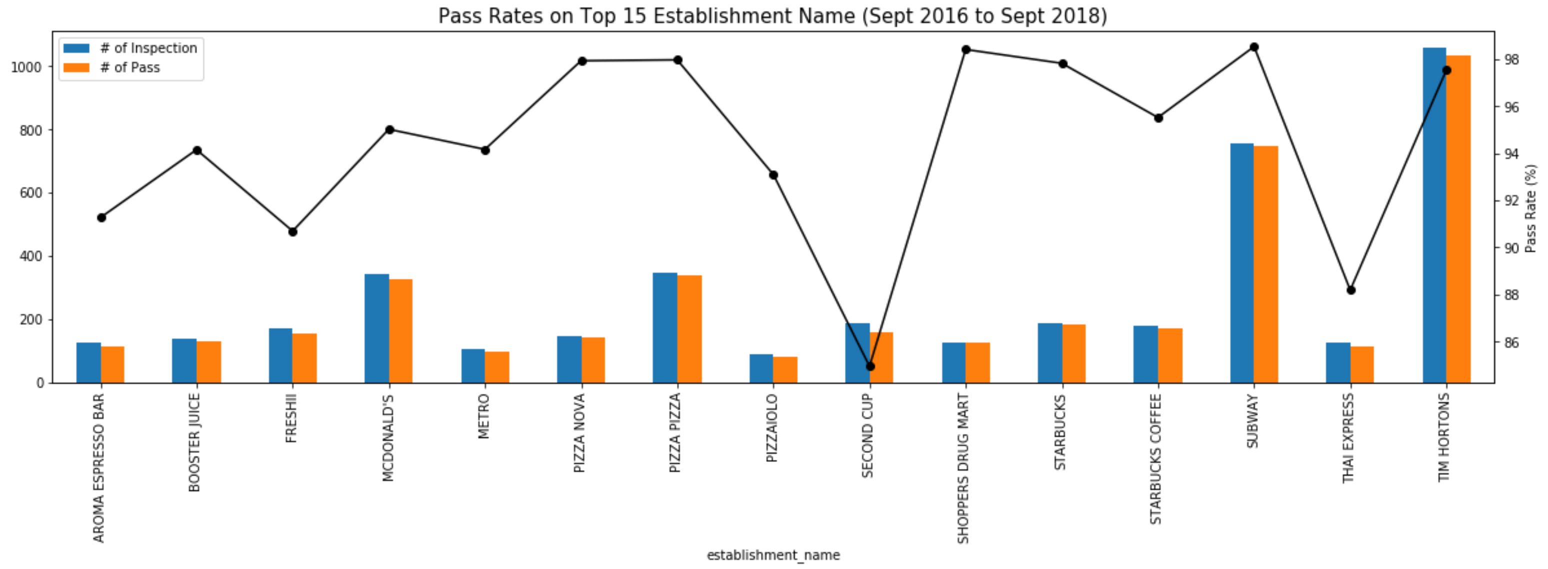
- Most pass rates of top 15 establishment type had decreased except Banquet Facility and Butcher Shop.

3.4. Pass Rates on Top 15 Establishment Name

```
In [118]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_name['insp'],
        '# of Pass': df_est_name['pass']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_name['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Establishment Name (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Pass Rate (%)')
```

```
Out[118]:

Text(0, 0.5, 'Pass Rate (%)')
```



- Second Cup has the lowest pass rate from top 15 Establishment Name.

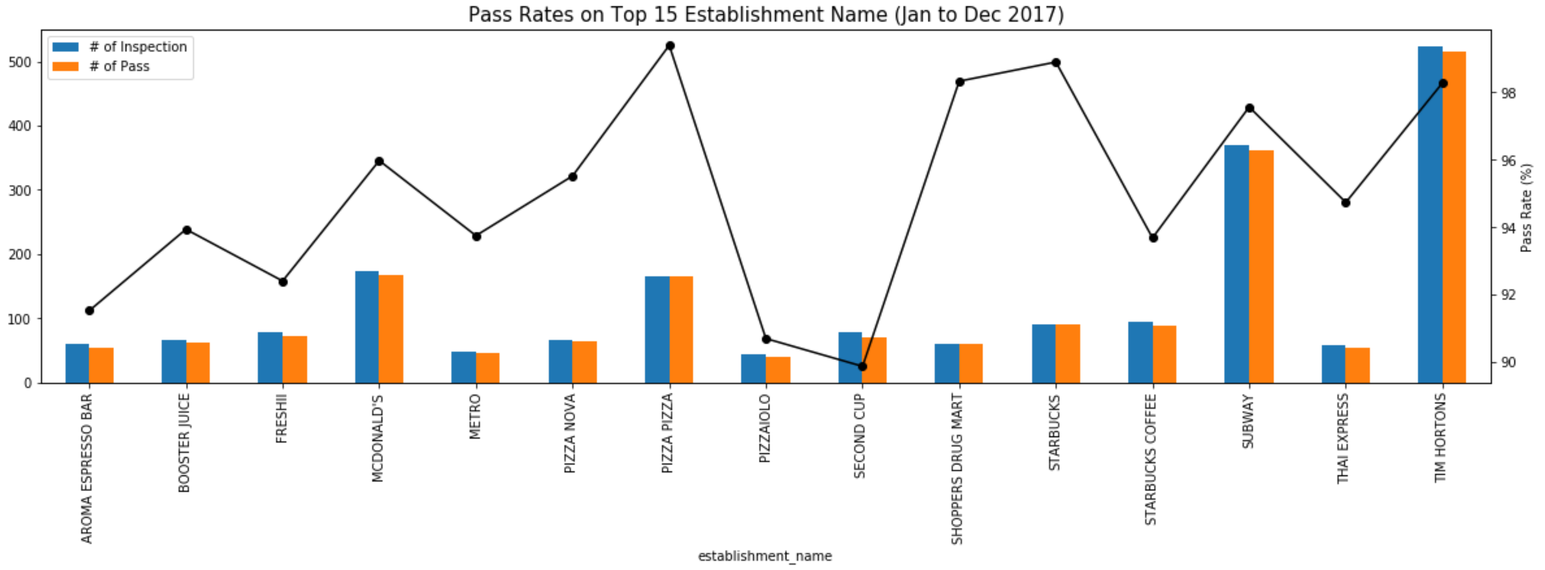
3.4.1. Pass Rates on Top 15 Establishment Name - 2017CY

In [119]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_name['insp_17'],
        '# of Pass': df_est_name['pass_17']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_name['pass_rate_17'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Establishment Name (Jan to Dec 2017)', fontsize=
15)
plt.ylabel('Pass Rate (%)')
```

Out[119]:

```
Text(0, 0.5, 'Pass Rate (%)')
```

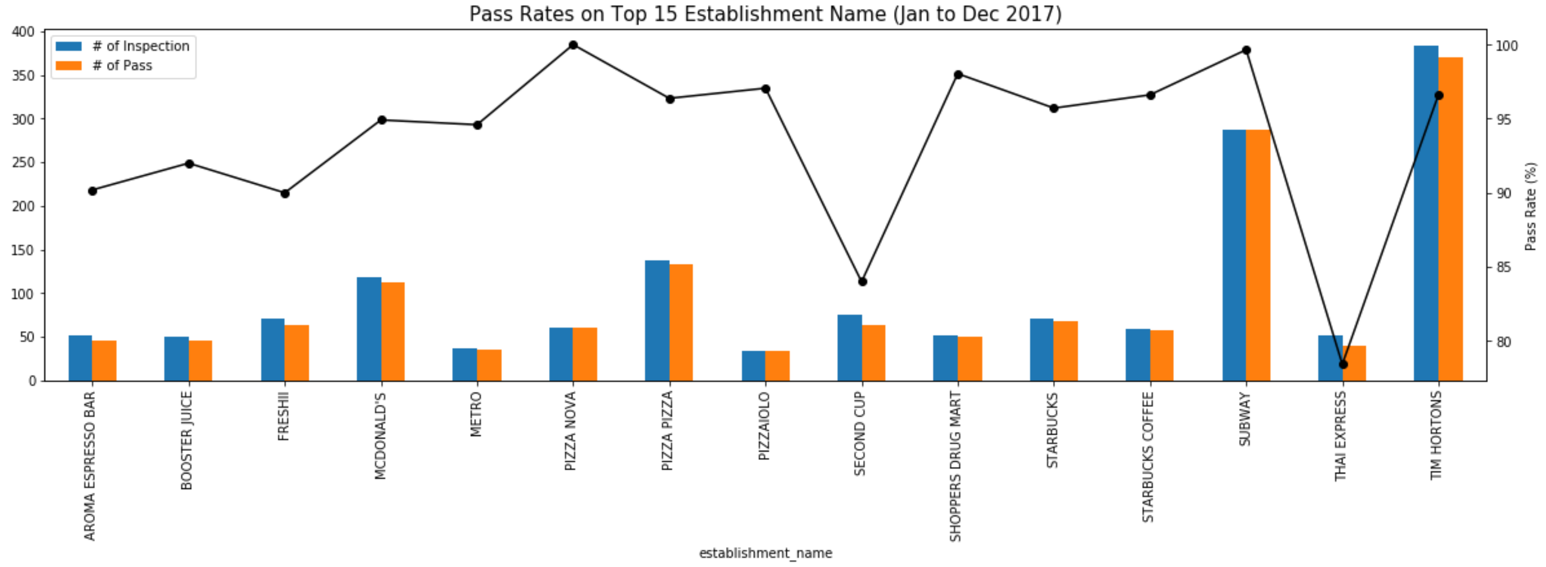


3.4.2. Pass Rates on Top 15 Establishment Name - 2018CY

```
In [120]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_est_name['insp_18'],
        '# of Pass': df_est_name['pass_18']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_est_name['pass_rate_18'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Establishment Name (Jan to Dec 2017)', fontsize=
15)
plt.ylabel('Pass Rate (%)')
```

```
Out[120]:

Text(0, 0.5, 'Pass Rate (%)')
```



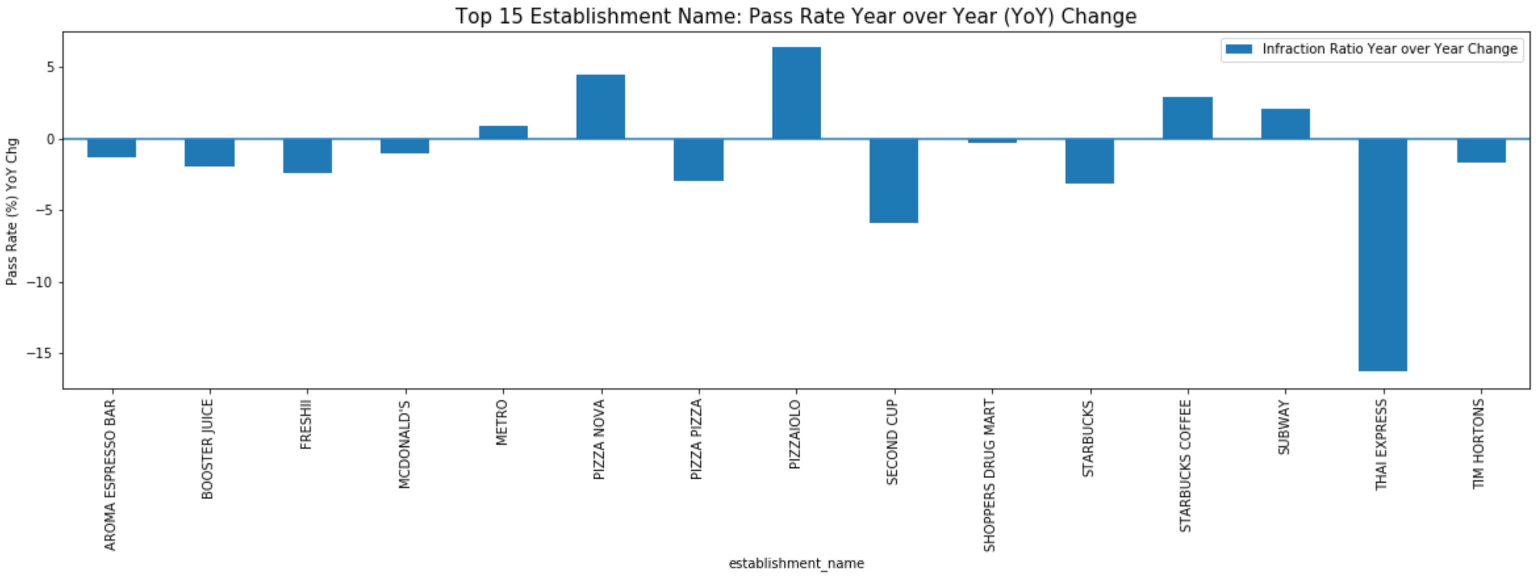
3.4.3. Pass Rate Year over Year (YoY) Change - Top 15 Establishment Name

In [121]:

```
df_grf = pd.DataFrame({'Infraction Ratio Year over Year Change':df_est_name['pas
s_rate_18'] - df_est_name['pass_rate_17']})
ax1 = df_grf.plot.bar(figsize=(20,5))
plt.axhline(0)
plt.title('Top 15 Establishment Name: Pass Rate Year over Year (YoY) Change', fo
ntsize=15)
plt.ylabel('Pass Rate (%) YoY Chg')
```

Out[121]:

```
Text(0, 0.5, 'Pass Rate (%) YoY Chg')
```



- Thai Express pass rate decreased by 0.15.

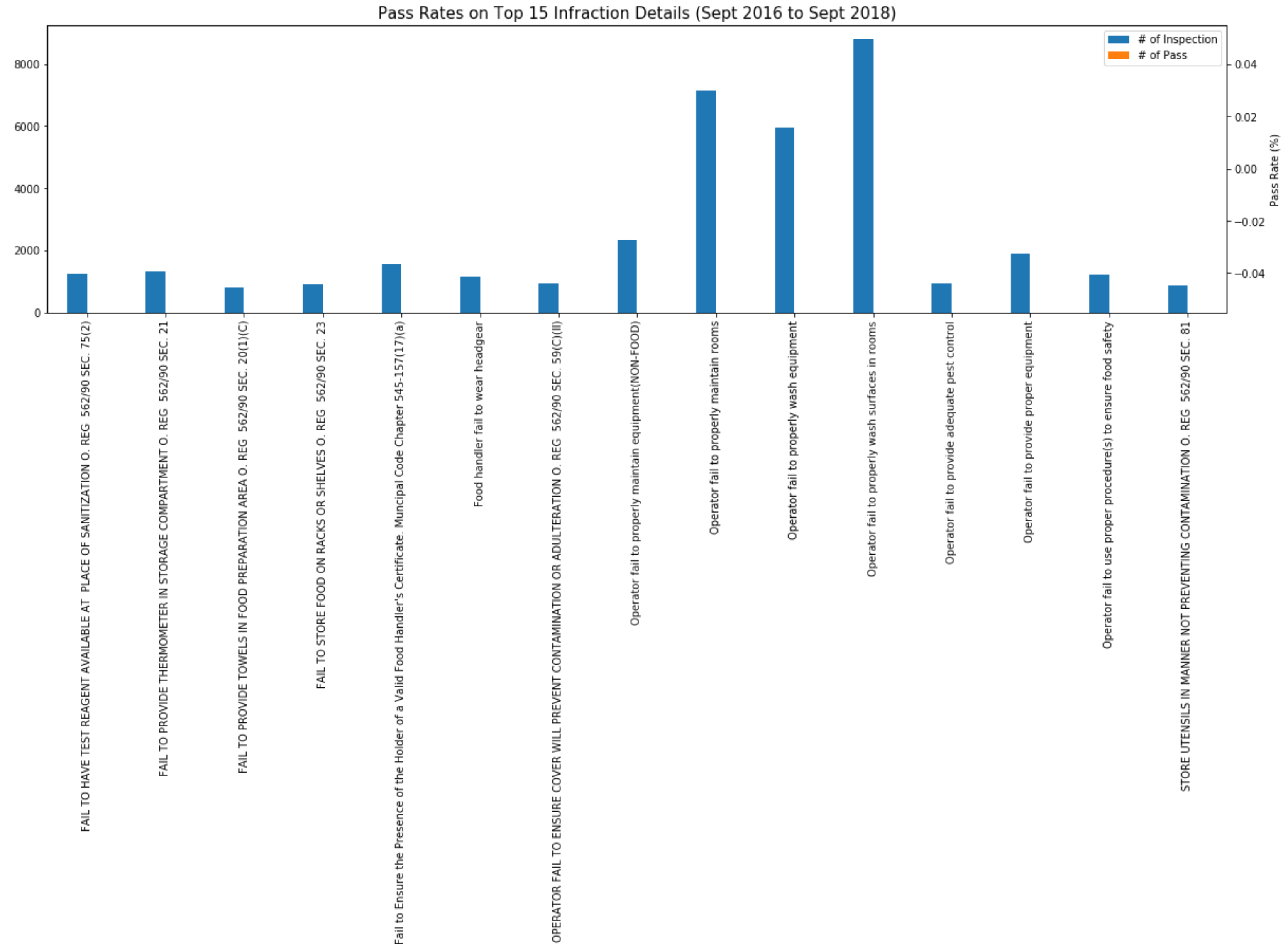
3.5. Pass Rates on Top 15 Infraction Details

In [122]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_inf_dtl['insp'],
        '# of Pass': df_inf_dtl['pass']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_inf_dtl['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Infraction Details (Sept 2016 to Sept 2018)', fo
ntsize=15)
plt.ylabel('Pass Rate (%)')
```

Out[122]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



Note:

- Operator fail to provide adequate pest control has the lowest pass rate (almost 0%).

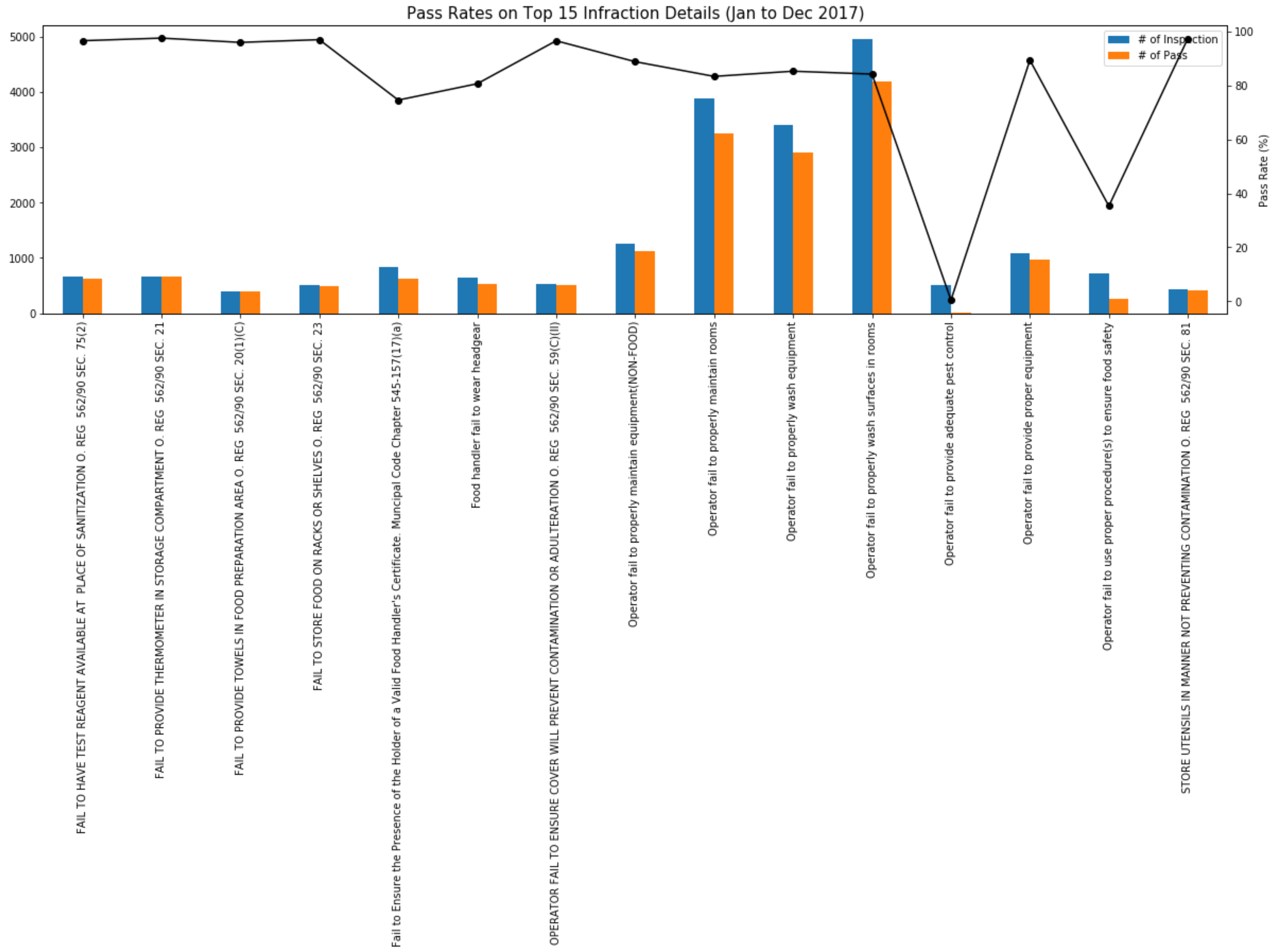
3.5.1. Pass Rates on Top 15 Infraction Details - 2017CY

In [123]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_inf_dtl['insp_17'],
        '# of Pass': df_inf_dtl['pass_17']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_inf_dtl['pass_rate_17'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Infraction Details (Jan to Dec 2017)', fontsize=
15)
plt.ylabel('Pass Rate (%)')
```

Out[123]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



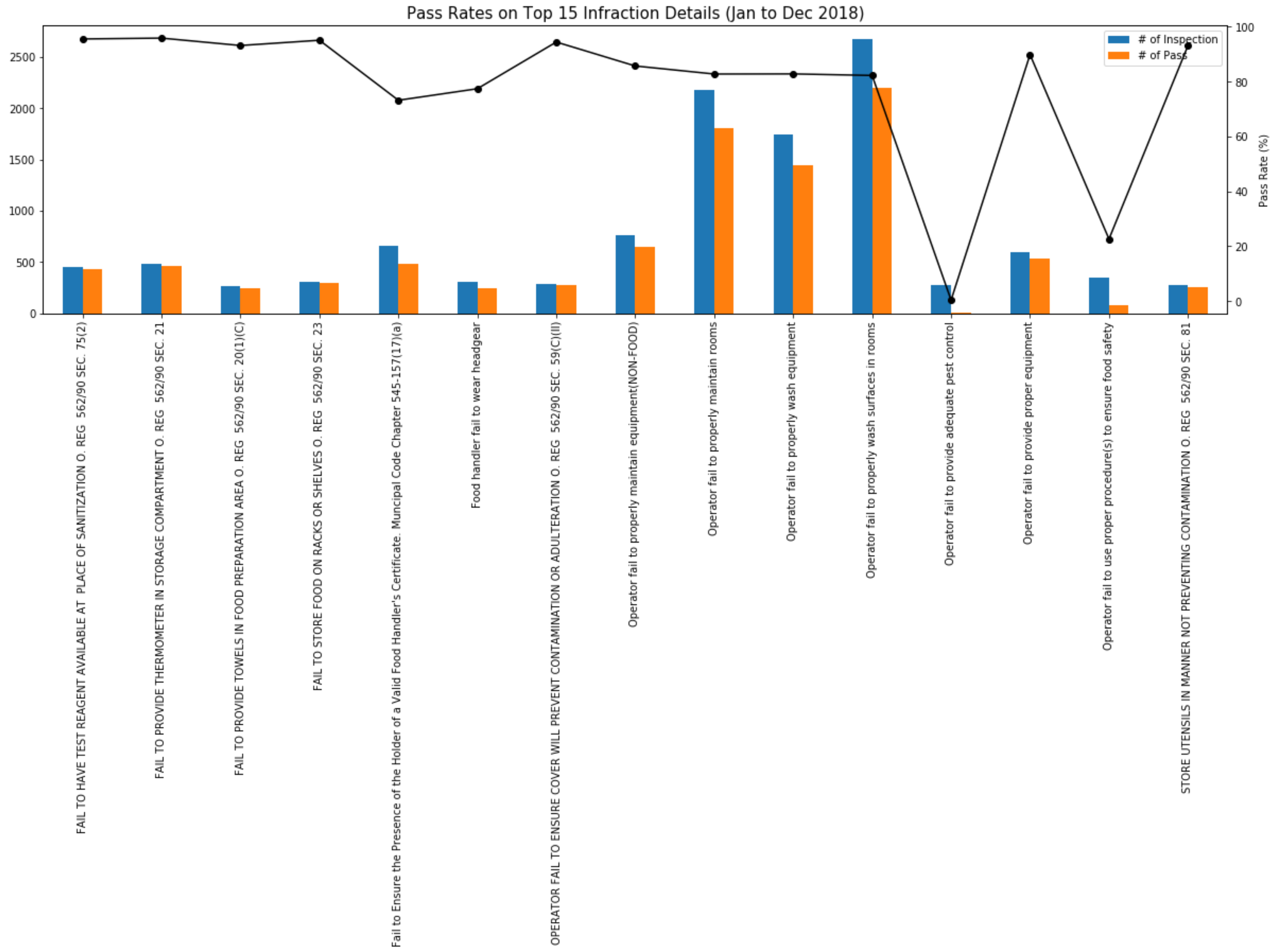
3.5.2. Pass Rates on Top 15 Infraction Details - 2018CY

In [124]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_inf_dtl['insp_18'],
        '# of Pass': df_inf_dtl['pass_18']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_inf_dtl['pass_rate_18'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(90)
plt.title('Pass Rates on Top 15 Infraction Details (Jan to Dec 2018)', fontsize=
15)
plt.ylabel('Pass Rate (%)')
```

Out[124]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



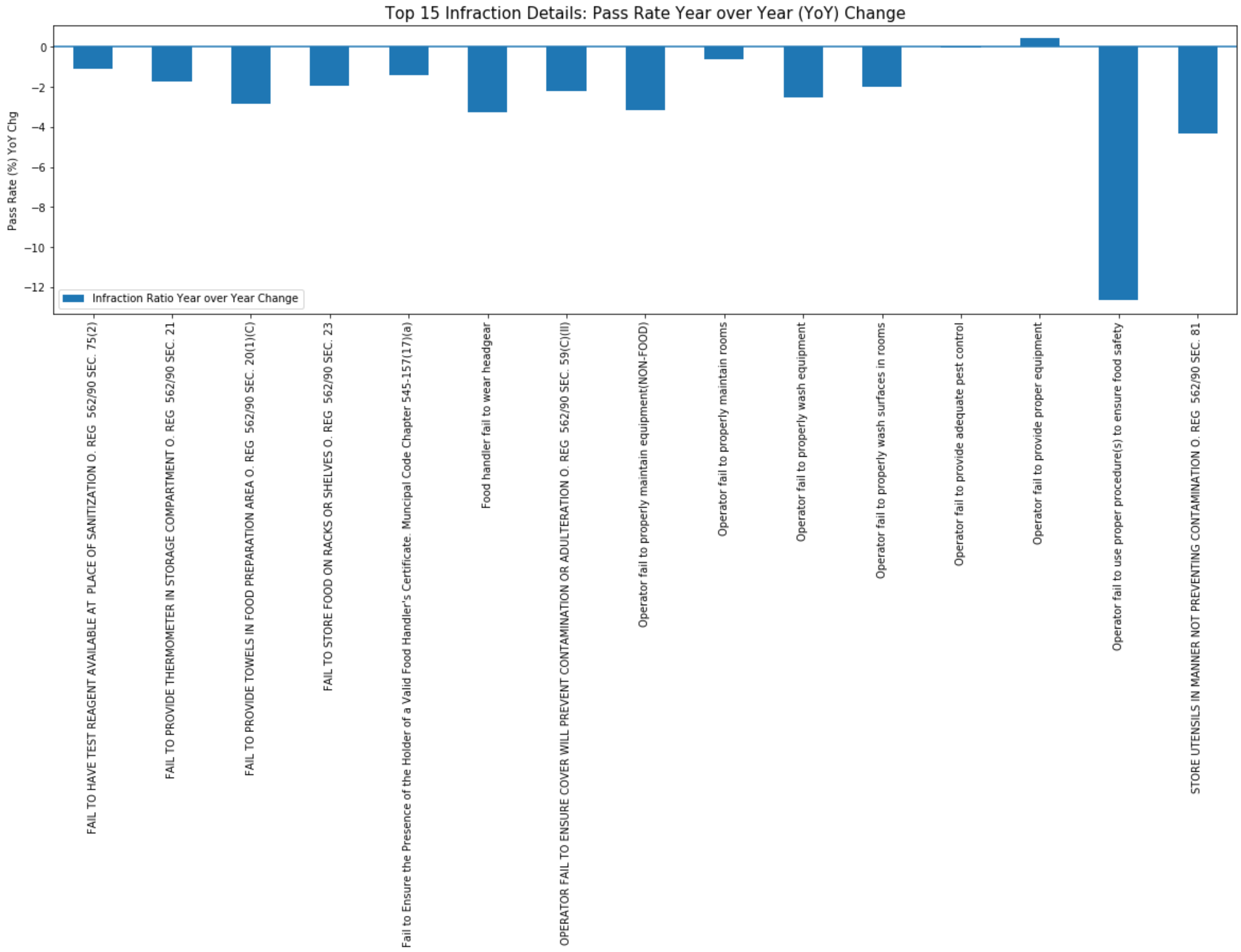
3.5.3. Pass Rate Year over Year (YoY) Change - Top 15 Infraction Details

```
In [125]:
```

```
df_grf = pd.DataFrame({'Infraction Ratio Year over Year Change':df_inf_dtl['pass
_rate_18'] - df_inf_dtl['pass_rate_17']})
ax1 = df_grf.plot.bar(figsize=(20,5))
plt.axhline(0)
plt.title('Top 15 Infraction Details: Pass Rate Year over Year (YoY) Change', fo
ntsize=15)
plt.ylabel('Pass Rate (%) YoY Chg')
```

```
Out[125]:
```

```
Text(0, 0.5, 'Pass Rate (%) YoY Chg')
```



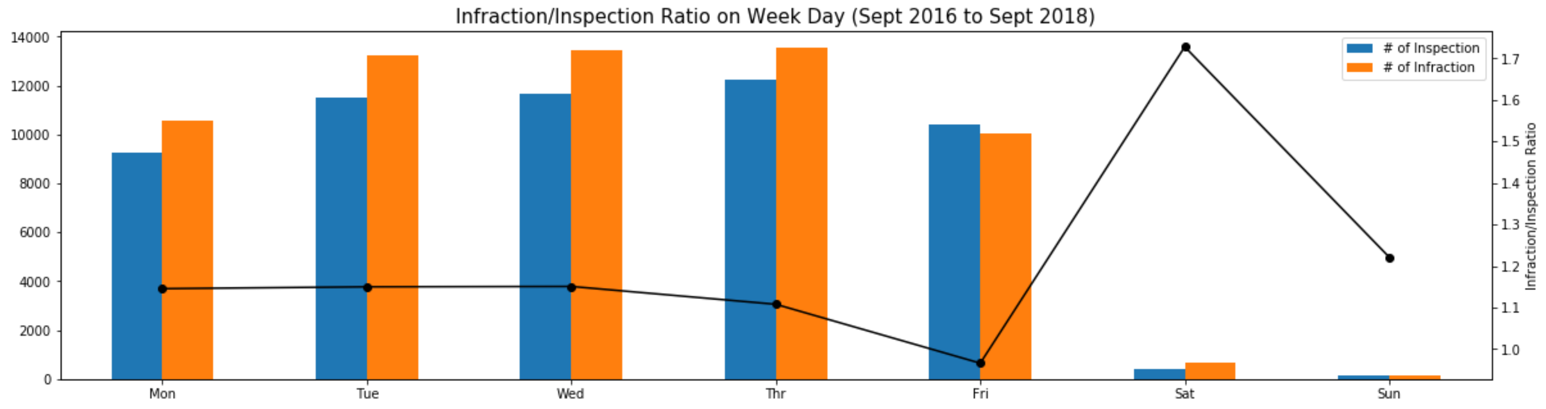
4. Week Day / Week / Month

4.1. Infraction/Inspection Rate on Week Day

```
In [129]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_wkdy['insp'],
        '# of Infraction': df_wkdy['infr']
    })
df_grf.index = ['Mon','Tue','Wed','Thr','Fri','Sat','Sun']
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_wkdy['ratio_infr_insp'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(0)
plt.title('Infraction/Inspection Ratio on Week Day (Sept 2016 to Sept 2018)', fo
ntsize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

```
Out[129]:

Text(0, 0.5, 'Infraction/Inspection Ratio')
```



- Lowest infraction/inspection ratio on Friday

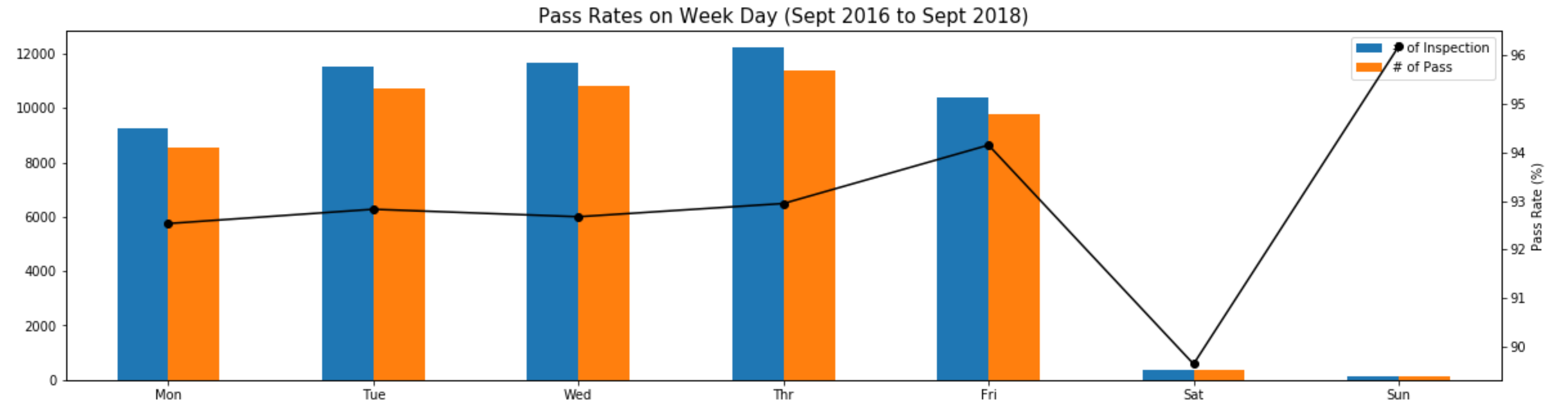
4.2. Pass Rate on Week Day

In [130]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_wkdy['insp'],
        '# of Pass': df_wkdy['pass']
    })
df_grf.index = ['Mon','Tue','Wed','Thr','Fri','Sat','Sun']
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_wkdy['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(0)
plt.title('Pass Rates on Week Day (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Pass Rate (%)')
```

Out[130]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



- Higher pass rate on Friday

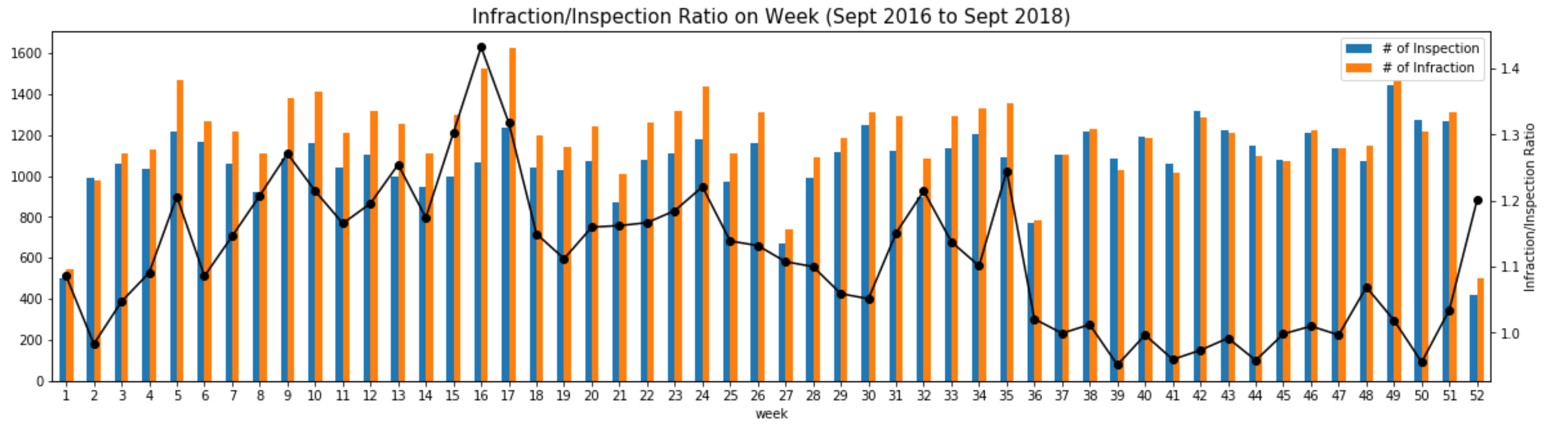
4.3. Infraction/Inspection Rate on Week

In [131]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_wk['insp'],
        '# of Infraction': df_wk['infr']
    })

ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_wk['ratio_infr_insp'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(0)
plt.title('Infraction/Inspection Ratio on Week (Sept 2016 to Sept 2018)', fontsi
ze=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[131]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```

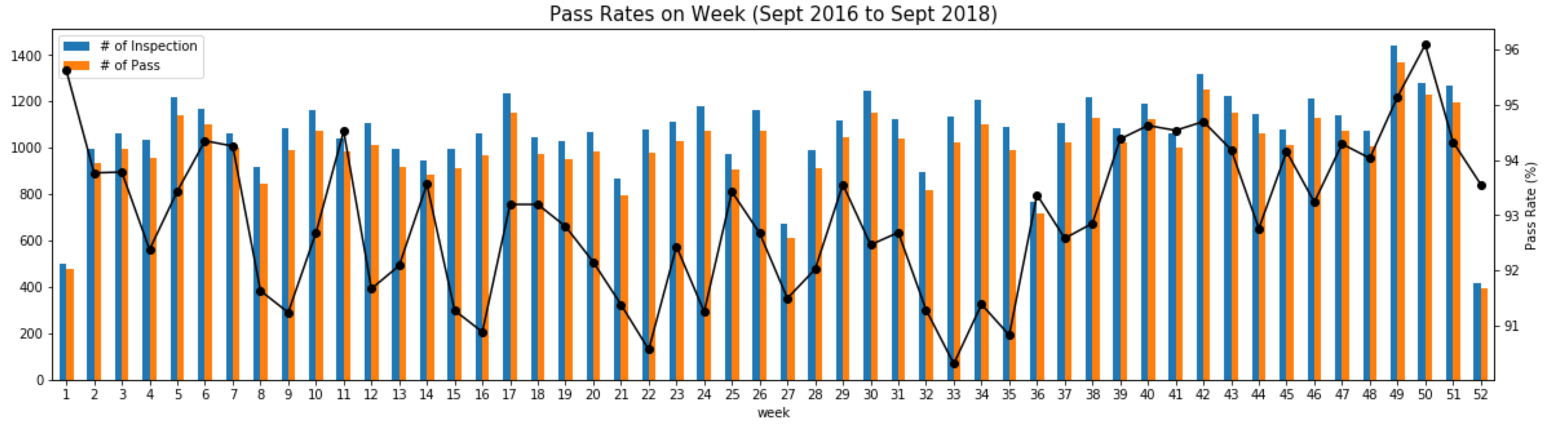


4.4. Pass Rate on Week

```
In [132]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_wk['insp'],
        '# of Pass': df_wk['pass']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_wk['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(0)
plt.title('Pass Rates on Week (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Pass Rate (%)')
```

```
Out[132]:

Text(0, 0.5, 'Pass Rate (%)')
```



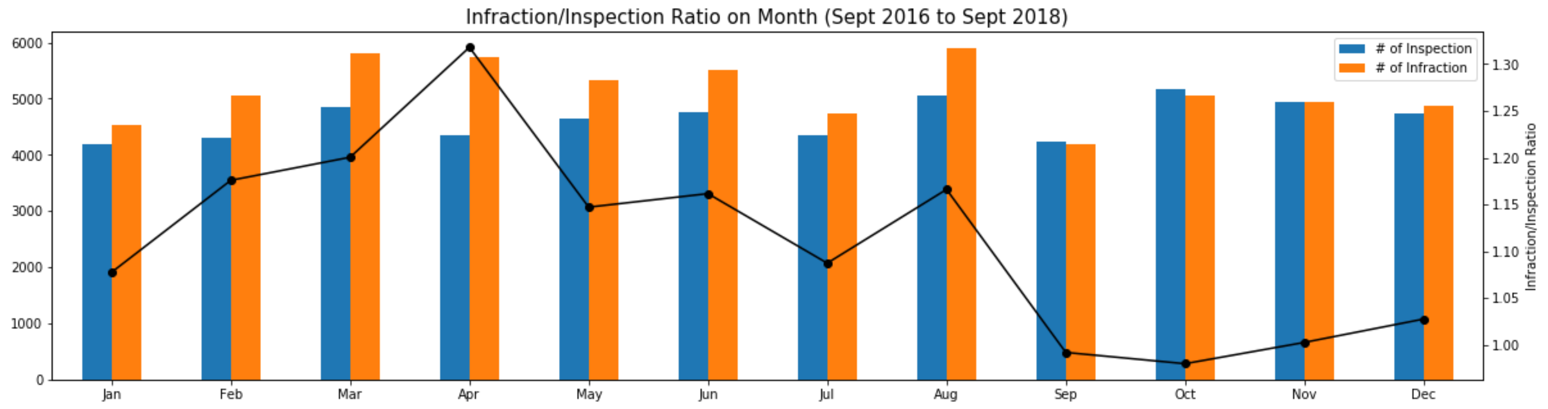
4.5. Infraction/Inspection Rate on Month

In [133]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_mth['insp'],
        '# of Infraction': df_mth['infr']
    })
df_grf.index = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
','Dec']
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_mth['ratio_infr_insp'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(0)
plt.title('Infraction/Inspection Ratio on Month (Sept 2016 to Sept 2018)', fonts
ize=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[133]:

```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



- Highest infraction/inspection ratio on April

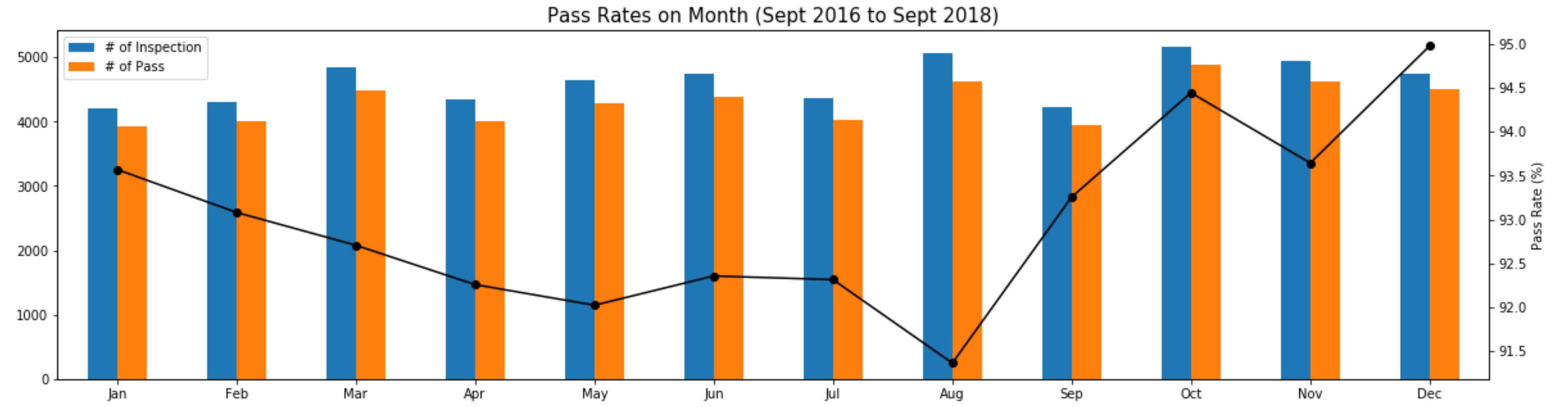
4.6. Pass Rate on Month

In [134]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_mth['insp'],
        '# of Pass': df_mth['pass']
    })
df_grf.index = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
','Dec']
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_mth['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(0)
plt.title('Pass Rates on Month (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Pass Rate (%)')
```

Out[134]:

```
Text(0, 0.5, 'Pass Rate (%)')
```



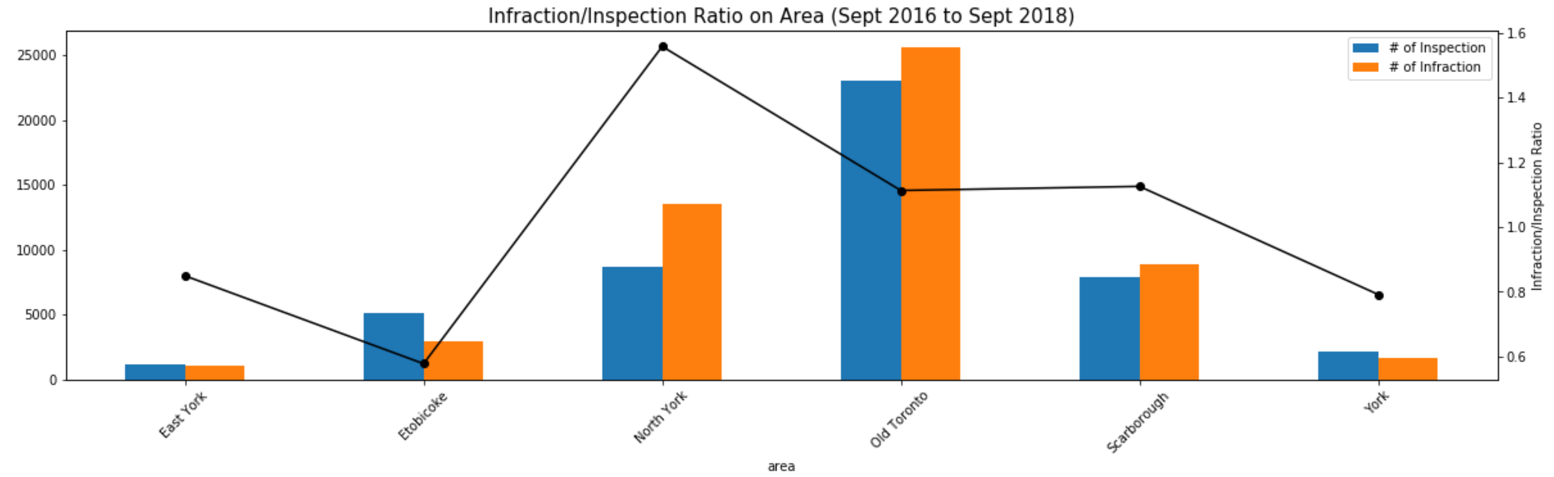
Infraction Rate on Area

In [135]:

```
df_grf = pd.DataFrame({
        '# of Inspection':df_area['insp'],
        '# of Infraction': df_area['infr']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_area['ratio_infr_insp'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(45)
plt.title('Infraction/Inspection Ratio on Area (Sept 2016 to Sept 2018)', fontsi
ze=15)
plt.ylabel('Infraction/Inspection Ratio')
```

Out[135]:

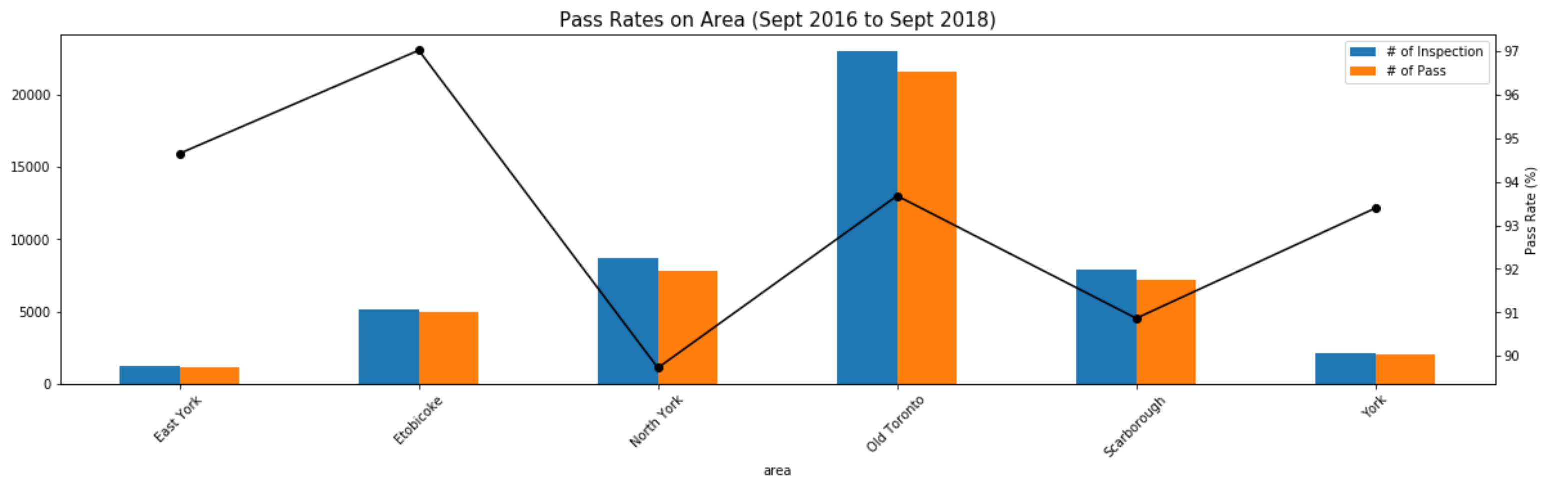
```
Text(0, 0.5, 'Infraction/Inspection Ratio')
```



```
In [136]:
```

```
df_grf = pd.DataFrame({
        '# of Inspection':df_area['insp'],
        '# of Pass': df_area['pass']
    })
ax1 = df_grf.plot.bar(figsize=(20,5))
l = list(df_area['pass_rate'])
s = pd.Series(l)
ax2 = s.plot(secondary_y=True, kind='line', figsize=(20,5), color ='k', marker='
o')
for tick in ax1.get_xticklabels():
    tick.set_rotation(45)
plt.title('Pass Rates on Area (Sept 2016 to Sept 2018)', fontsize=15)
plt.ylabel('Pass Rate (%)')
```

```
Out[136]:

Text(0, 0.5, 'Pass Rate (%)')
```



5. Other Analysis

6. Closed Establishment

In [137]:

```
closed = df.query("establishment_status == 'Closed'")
len(closed)
closed.head()
```

Out[137]:

	row_id	establishment_id	inspection_id	establishment_name	establishmenttype
1250	1251	9002014	104292592	BUN KING BAKERY	Bakery
1251	1252	9002014	104292592	BUN KING BAKERY	Bakery
1252	1253	9002014	104292592	BUN KING BAKERY	Bakery
1253	1254	9002014	104292592	BUN KING BAKERY	Bakery
1254	1255	9002014	104292592	BUN KING BAKERY	Bakery

In [138]:

```
df[['infraction_details','establishment_status']].query("establishment_status ==
'Closed'").head()
```

Out[138]:

	infraction_details	establishment_status
1250	FAIL TO PROVIDE REQUIRED ILLUMINATION DURING A...	Closed
1251	Fail to maintain handwashing stations (liquid ...	Closed
1252	Fail to maintain records of pest control measu...	Closed
1253	Fail to protect against harbouring of pests - ...	Closed
1254	Fail to protect food from contamination or adu...	Closed

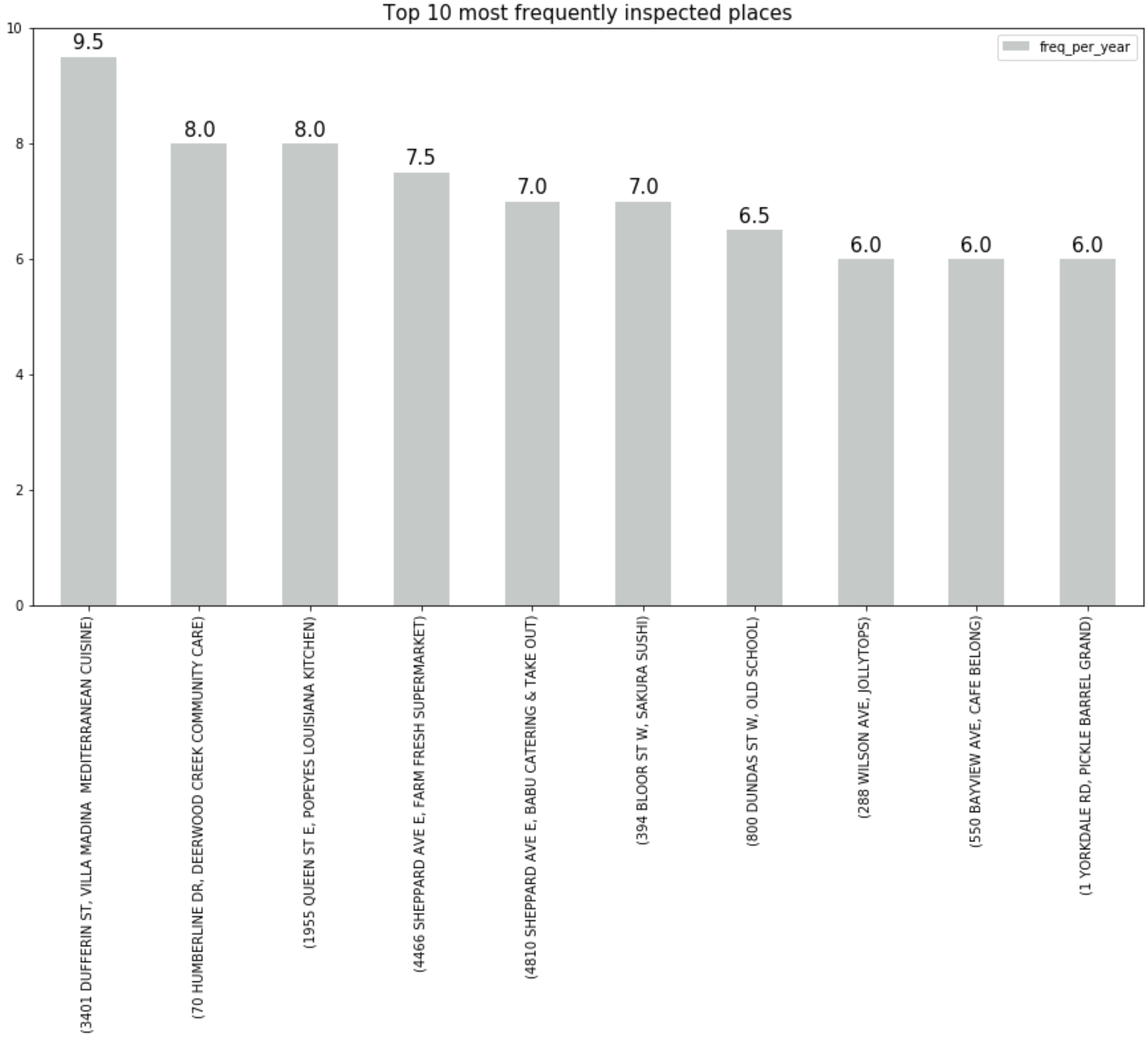
Top 10 most inspected places

In [139]:

```
#Count the number of inspections for each establishment
count_establishment = df.groupby(['establishment_address', 'establishment_name']
)['inspection_id'].nunique()
#Transfer the series to a dataframe
freq = count_establishment.to_frame(name = 'freq_per_year')
#Order the dataframe by 'freq per year'
freq_order=freq.sort_values(['freq_per_year'],ascending = False)
freq_order['freq_per_year']=freq_order['freq_per_year']/2
freq_top_10 = freq_order[:10]
```

In [140]:

```
ax1=freq_top_10.plot.bar(color='xkcd:silver',figsize=(15,8))
plt.title('Top 10 most frequently inspected places', fontsize = 15)
plt.xlabel('',fontsize=15)
plt.ylabel('',fontsize=15)
ax1.set_ylim(0,10)
for p in ax1.patches:
    ax1.annotate("%.1f" % p.get_height(), (p.get_x() + p.get_width() / 2., p.get_height()),
                 ha='center', va='center', fontsize=15, color='black', xytext=(0, 10),
                 textcoords='offset points')
```



In [141]:

```
#Assign different infraction scores to establishments based on the category of severity
def label_severity(row):
 if row['severity'] == 'M - Minor':
   return 1
 if row['severity']== 'S - Significant':
   return 2
 if row['severity']== 'C - Crucial':
   return 3
 return 0

df['score']=df.apply(lambda row: label_severity(row),axis=1)
```

In [142]:

```
# get the total score per establishment
sum_score = df.groupby(['establishment_address', 'establishment_name'])['score'].sum()
total_score = sum_score.to_frame(name = 'score')
total_score=total_score.sort_values(['score'],ascending = False)
# get the total number of inspection per establishment
total_inspection = df.groupby(['establishment_address', 'establishment_name'])['inspection_id'].nunique()
total_ins = total_inspection.to_frame(name = 'num_inspection')
total_ins=total_ins.sort_values(['num_inspection'],ascending = False)
# join total score table with the total inspection table
temp = pd.merge(total_ins, total_score, on=('establishment_address','establishment_name'), how='inner')
# create  a avg_score column
temp['Avg_Score'] = temp['score']/temp['num_inspection']
avgScore = temp.sort_values(['Avg_Score'],ascending = False)
# round avg score to 1 decimal
avgScore=avgScore.round(1)
# reset index for both freq_order and avgScore table
freq_order.reset_index()
avgScore.reset_index()

# Join freq_order and avgScore table to get the scoring system table
ScoreData = pd.merge(freq_order, avgScore, on=('establishment_address','establishment_name'), how='inner')
ScoreData=ScoreData.sort_values(['score'],ascending = False)
ScoreData
top20_score_data=ScoreData[:20]
top20_score_data
```

Out[142]:

		freq_per_year	num_inspection	score	A
establishment_address	establishment_name				
3401 DUFFERIN ST	VILLA MADINA MEDITERRANEAN CUISINE	9.5	19	132	6
866 WILSON AVE	MUSTAFA	4.5	9	104	1
3555 DON MILLS RD	TASTY BBQ SEAFOOD RESTAURANT	5.0	10	89	8
531 WILSON HEIGHTS BLVD	TIMES SQUARE DINER	5.0	10	89	8
1008 WILSON AVE	PHO MI ASIA	5.0	10	86	8
3220 DUFFERIN ST	SEOUL HOUSE	4.5	9	85	9
2450 DUFFERIN ST	YUM YUM RESTAURANT	4.5	9	83	9
...	...	...	...	...	..
3200 DUFFERIN ST	KRYSTOS MODERN GREEK CUISINE	4.5	9	69	7
2350 YONGE ST	STAR KING	4.5	9	69	7
1027 STEELES AVE W	KIVA'S BAGEL BAKERY & RESTAURANT	4.0	8	68	8
1955 QUEEN ST E	POPEYES LOUISIANA KITCHEN	8.0	16	68	4
1090 WILSON AVE	METRO	3.5	7	68	9
3203 DUFFERIN ST	LUCKY DRAGON RESTAURANT	4.5	9	65	7
2555 VICTORIA PARK AVE	HONG TAI SUPERMARKET	4.0	8	61	7

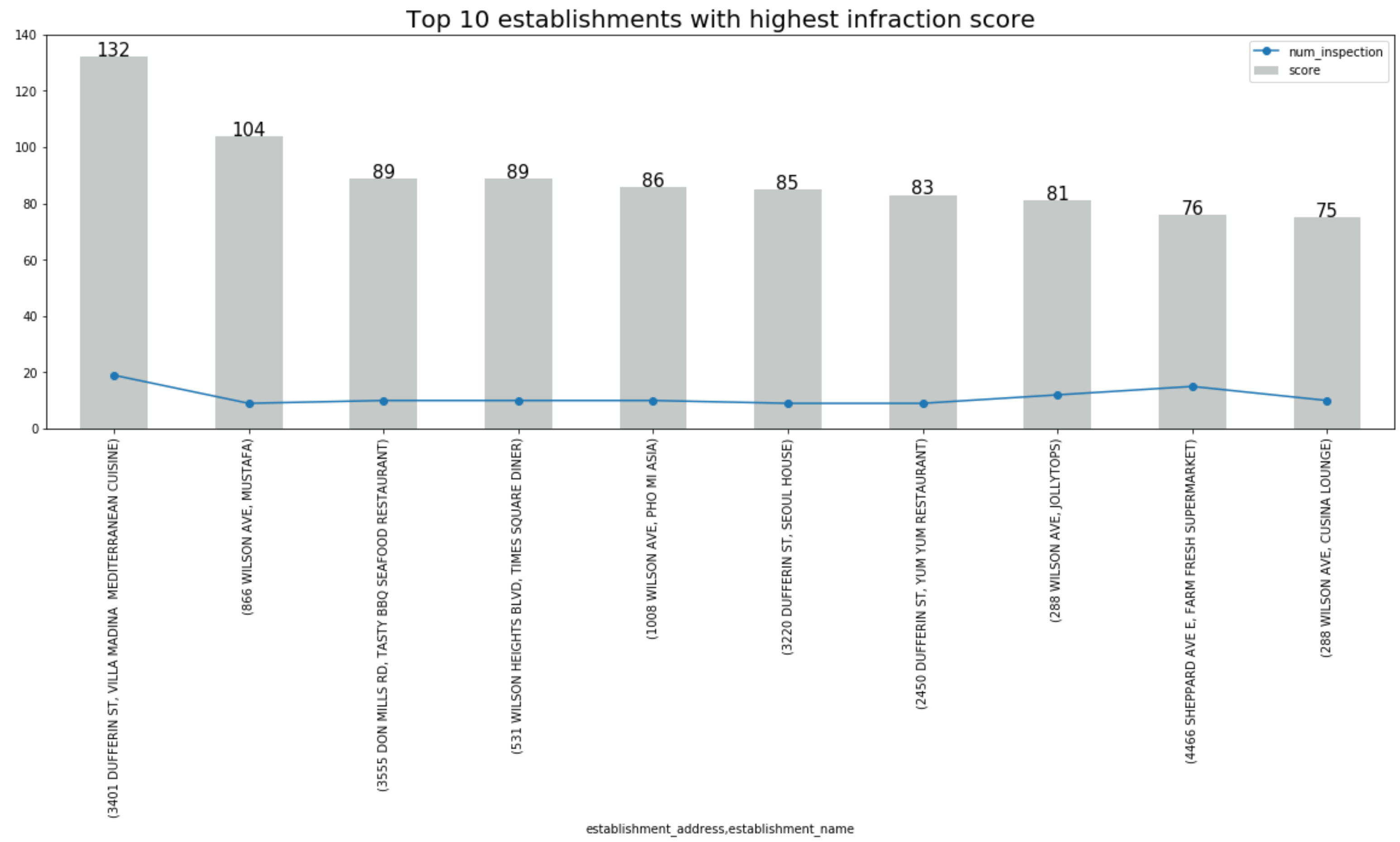
20 rows × 4 columns

In [143]:

```
top_10_score= ScoreData[:10]
ax = top_10_score['num_inspection'].plot(linestyle='-', marker='o',figsize=(20,6
))
top_10_score['score'].plot(kind='bar',ax=ax,color='xkcd:silver')
ax.set_title("Top 10 establishments with highest infraction score", fontsize=20)
ax.set_ylim(0,140)
for p in ax.patches:
    ax.annotate("%.0f" % p.get_height(), (p.get_x() + p.get_width() / 2., p.get_
height()),
                 ha='center', va='center', fontsize=15, color='black', xytext=(0
, 5),
                 textcoords='offset points')
plt.legend()
```

Out[143]:

```
<matplotlib.legend.Legend at 0x115bf5d68>
```



In [144]:

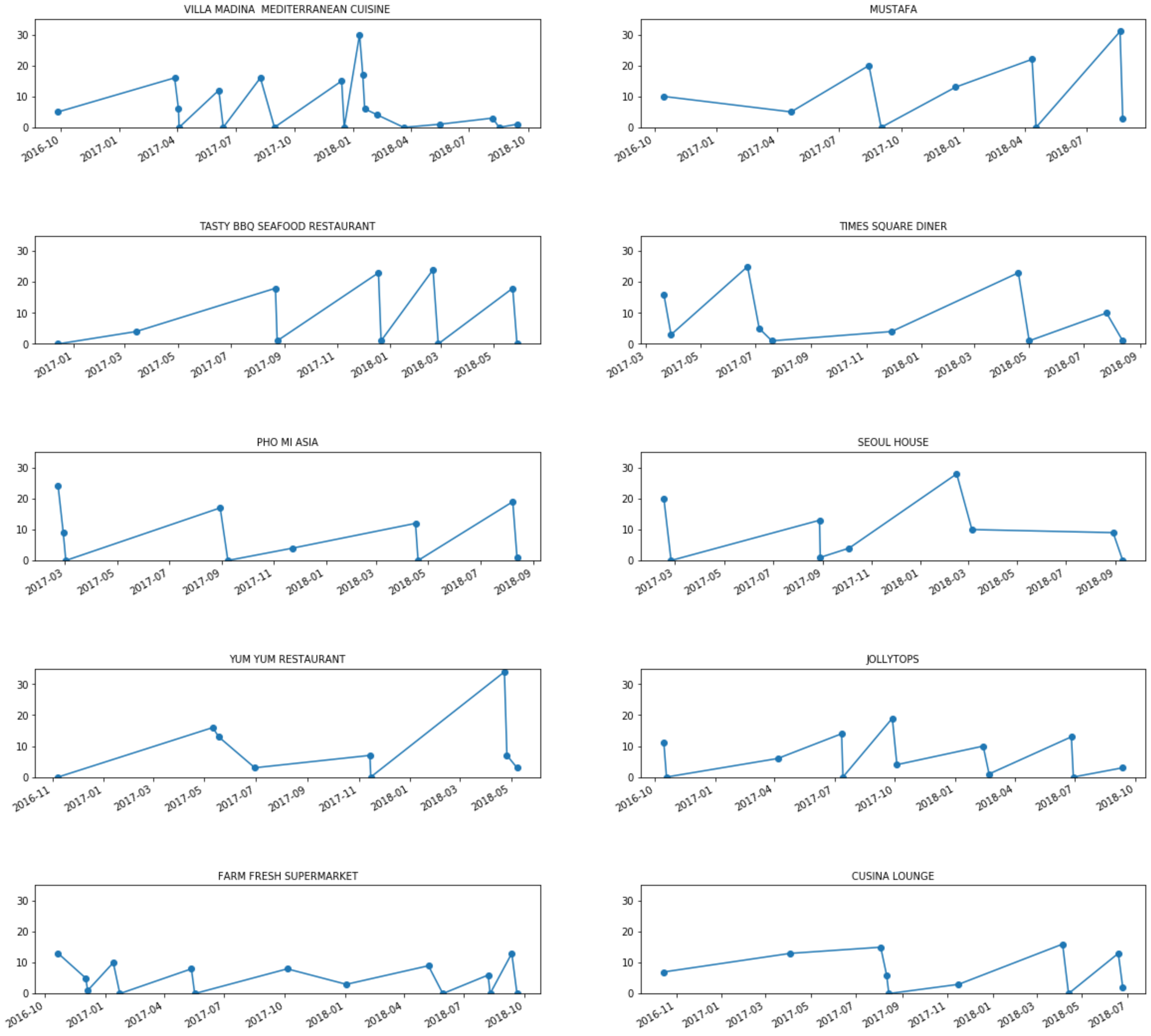
```
loc_index = top20_score_data.index.values[:10]
len(loc_index)
```

Out[144]:

```
10
```

In [145]:

```
num=1
#var_data=pd.DataFrame(columns=['establishment_name','variance'])
std_list=[]
str_list=[]
for i in range(len(loc_index)):
    plt.subplot(5,2, num)
    num+=1
    #if num in range(14) :
    #    plt.tick_params(labelbottom=False)
    #if num not in [1,4,7] :
     #   plt.tick_params(labelleft=True)
    plt.subplots_adjust(hspace=1)
    #plt.tick_params(labelbottom=True)
    Villa_Trend = df[(df['establishment_address']==loc_index[i][0]) & (df['establishment_name']==loc_index[i][1])].sort_values(['inspection_date'],ascending = True)
    Villa_Trend=Villa_Trend.groupby('inspection_date').sum()
    x=np.std(Villa_Trend['score'])
    #list1=[loc_index[i][1],x]
    std_list.append(x)
    str_list.append(loc_index[i][1])
    ax_v=Villa_Trend['score'].plot(linestyle='-', marker='o',figsize=(20,20))
    ax_v.set_ylim(0,35)
    plt.title(loc_index[i][1], fontsize = 10)
    ax_v.xaxis.label.set_visible(False)
```



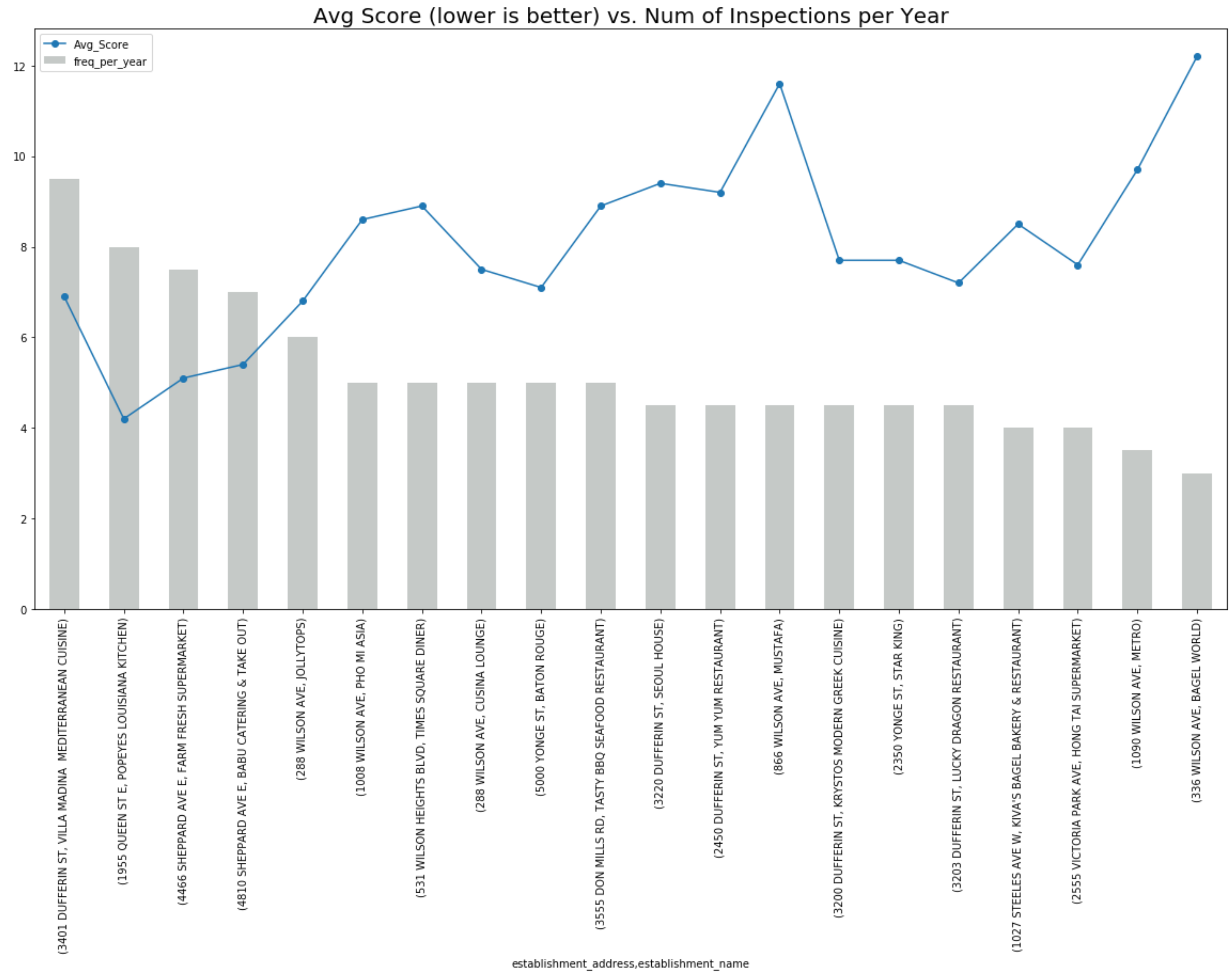
Avg. Score vs. Num of inspections per year

In [146]:

```
sort_score_data = top20_score_data.sort_values(['num_inspection'],ascending = Fa
lse)
ax = sort_score_data[['Avg_Score']].plot(linestyle='-', marker='o',figsize=(20,1
0))
sort_score_data[['freq_per_year']].plot(kind='bar', ax=ax,color='xkcd:silver')
ax.set_title("Avg Score (lower is better) vs. Num of Inspections per Year", font
size=20)
plt.legend()
```

Out[146]:

```
<matplotlib.legend.Legend at 0x11515da90>
```



In [147]:

```
std_df = pd.DataFrame({'establishment_name':str_list, 'std':std_list})
```

In [148]:

```
std_ins_corr = pd.merge(ScoreData, std_df, on=('establishment_name'), how='inner')
std_ins_corr1=std_ins_corr.sort_values(['Avg_Score'],ascending = False)
std_ins_corr1
```

Out[148]:

	establishment_name	freq_per_year	num_inspection	score	Avg_Score	std
1	MUSTAFA	4.5	9	104	11.6	10.253575
5	SEOUL HOUSE	4.5	9	85	9.4	9.117884
6	YUM YUM RESTAURANT	4.5	9	83	9.2	10.173798
2	TASTY BBQ SEAFOOD RESTAURANT	5.0	10	89	8.9	9.893938
3	TIMES SQUARE DINER	5.0	10	89	8.9	8.780091
4	PHO MI ASIA	5.0	10	86	8.6	8.534635
9	CUSINA LOUNGE	5.0	10	75	7.5	5.953990
0	VILLA MADINA MEDITERRANEAN CUISINE	9.5	19	132	6.9	8.198034
7	JOLLYTOPS	6.0	12	81	6.8	6.206515
8	FARM FRESH SUPERMARKET	7.5	15	76	5.1	4.711216

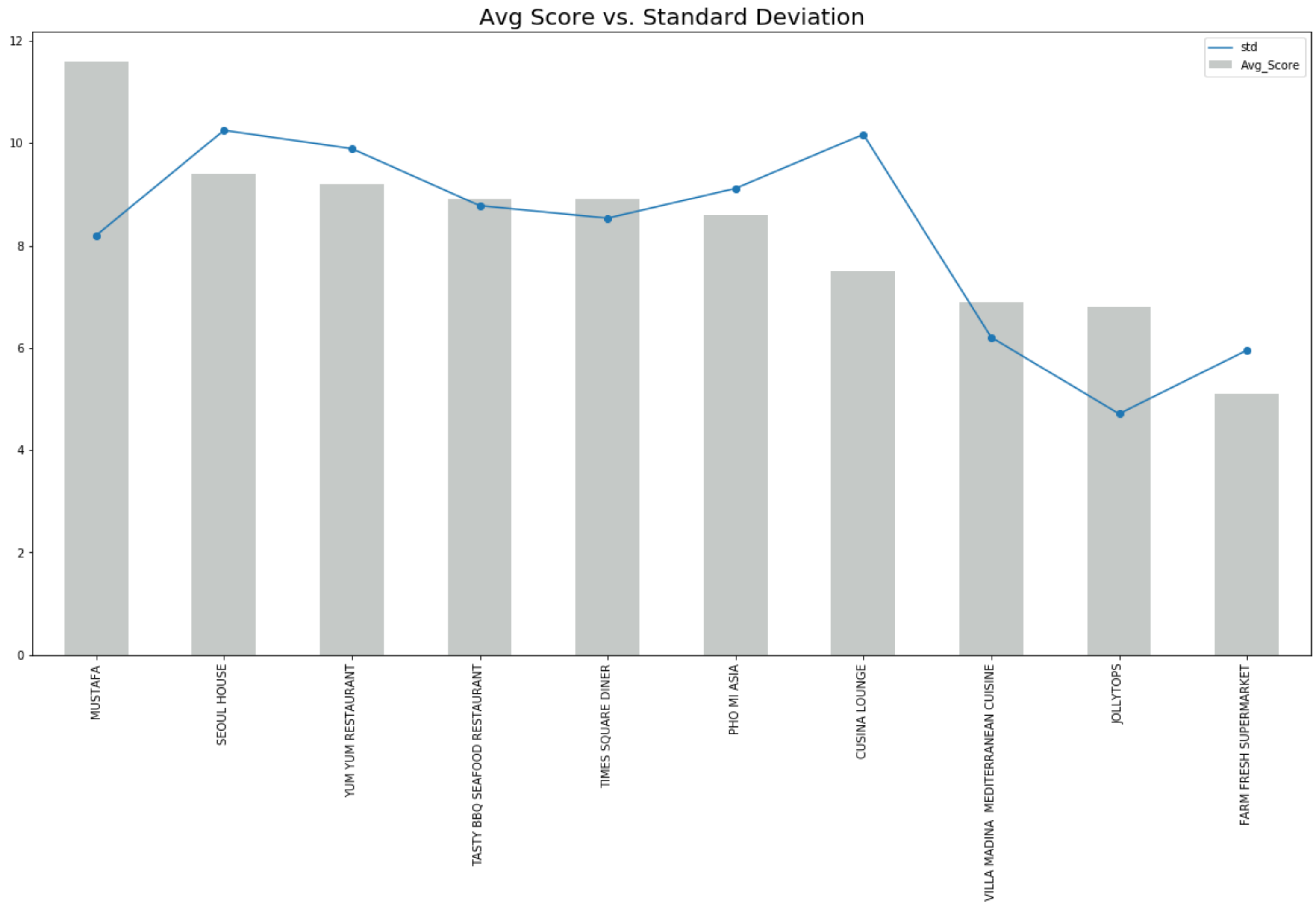
Avg. Score vs. Std

In [149]:

```
ax_std1 = std_ins_corr[['std']].plot(linestyle='-', marker='o',figsize=(20,10))
std_ins_corr1[['Avg_Score']].plot(kind='bar',ax = ax_std1, color='xkcd:silver')
ax_std1.set_title("Avg Score vs. Standard Deviation", fontsize=20)
ax_std1.set_xticklabels(std_ins_corr1.establishment_name)
```

Out[149]:

```
[Text(0, 0, 'MUSTAFA'),
 Text(0, 0, 'SEOUL HOUSE'),
 Text(0, 0, 'YUM YUM RESTAURANT'),
 Text(0, 0, 'TASTY BBQ SEAFOOD RESTAURANT'),
 Text(0, 0, 'TIMES SQUARE DINER'),
 Text(0, 0, 'PHO MI ASIA'),
 Text(0, 0, 'CUSINA LOUNGE'),
 Text(0, 0, 'VILLA MADINA  MEDITERRANEAN CUISINE'),
 Text(0, 0, 'JOLLYTOPS'),
 Text(0, 0, 'FARM FRESH SUPERMARKET')]
```



In [150]:

```
std_ins_corr1['Avg_Score'].corr(std_ins_corr1['std'])
```

Out[150]:

```
0.8689779472791854
```

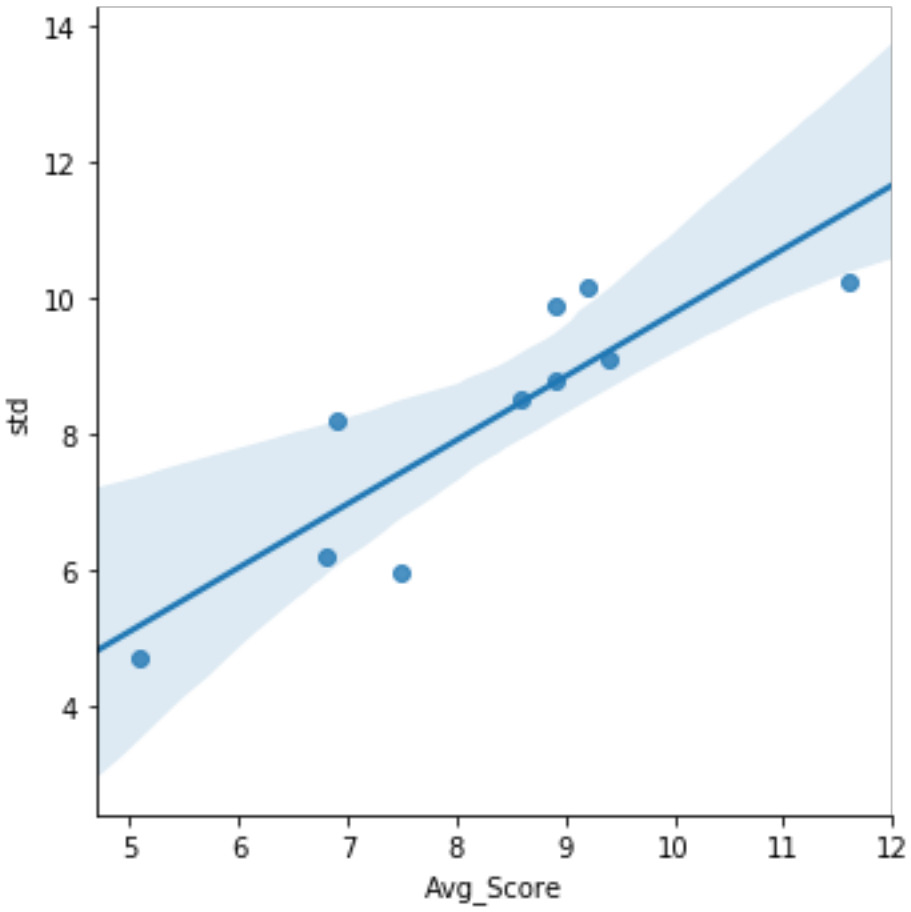
In [151]:

```
#fit a regression line to the scatter plot
sns.lmplot(x='Avg_Score', y='std', data=std_ins_corr1)
```

```
/usr/local/lib/python3.7/site-packages/scipy/stats/stats.py:1713: Fu
tureWarning: Using a non-tuple sequence for multidimensional indexin
g is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the
future this will be interpreted as an array index, `arr[np.array(seq
)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```

Out[151]:

```
<seaborn.axisgrid.FacetGrid at 0x115c777b8>
```



Find Locations based on Latitude and Longitude from Google API

Before calling the google api, please consider that it's a bit less than 96K records

So it's better to:

1. Collect the unique list of Lat and long and save it in another data frame
2. Search for the address but it's not suggeted to just apply the lambda for the hole dataframe, if any record fails, it rolls back the hole dataframe and will loose the hole api calls it made
3. It takes around 2 hours to fetch the hole data from a PC in sequential calls
4. Save the data frame as a separate data frame, so for each analysis there is no need to call the api again, it saves time and cost
5. Check if there is no bad data in the dataset, otherwise google api throws an error and the process will be stopped
6. For make it faster, it's better to use async calls in parallel which drops the time from 2 hours to 15 minutes
7. For showing the progress bar, tqdm has been used

In [198]:

```
gmaps = googlemaps.Client(key='AIzaSyClxI-vvB3yT4Ic_pO76hon9WSbBMpghiM')
def find_first_postal_code(reverse_geocode_result):
    for google_result in reverse_geocode_result:
        for item in google_result.get("address_components"):
            if((item["types"][0]=='postal_code')):
                return item['short_name']

def find_first_locality_name(reverse_geocode_result):
    for google_result in reverse_geocode_result:
        for item in google_result.get("address_components"):
            if(("sublocality" in item["types"])& ("sublocality_level_1" in item[
"types"])):
                return item["short_name"]


def get_postal_code_and_locality(lat, long):
    reverse_geocode_result = gmaps.reverse_geocode((lat, long))
    postal_code = find_first_postal_code(reverse_geocode_result)
    city = find_first_locality_name(reverse_geocode_result)
    return[lat, long, postal_code, city]
```

In [199]:

```
df[df['longitude']<-90]
```

Out[199]:

	row_id	establishment_id	inspection_id	establishment_name	establishmenttyp
89797	89798	10638787	104258049	FRESCO	Supermarket

In [200]:

```
def fixLogtitude(df):
    df.longitude.iloc[89797]= df.longitude.iloc[89797] / 10**9
    df[df['longitude']<-90]
    df.LONGITUDE.iloc[89797]
```

In [201]:

```
locs = df.groupby(['latitude', 'longitude']).size().reset_index(name='Count')
len(locs)
```

Out[201]:

```
11003
```

- There are 11003 unique point, so it helps to have a less query

In [212]:

```
def createLocationList(locs):

    with concurrent.futures.ProcessPoolExecutor(4) as pool:
        location_list = list(tqdm.tqdm(pool.map(get_postal_code_and_locality, lo
cs['latitude'], locs['longitude'], chunksize=20), total=df.shape[0])) # Without
a progressbar
        return location_list
```

In [213]:

```
def put_location_lits_to_csv_file(location_list):
    new_locs = pd.DataFrame(columns=['latitude', 'longitude',"Postal_Code","Area
"], data=location_list)
    new_locs.to_csv("locations_2.csv", sep=',', encoding='utf-8')
```

In [214]:

```
df['score']=df.apply(lambda row: label_severity(row),axis=1)
df.groupby(['area'])['score'].mean().reset_index(name="ave").sort_values(by="ave",ascending=False)
```

Out[214]:

	area	ave
2	North York	1.107693
4	Scarborough	0.982874
3	Old Toronto	0.949077
0	East York	0.764020
5	York	0.738684
1	Etobicoke	0.567130