renfrewshire_hygiene_analysis

May 29, 2025

1 Renfrewshire Food Hygiene Data Analysis

1.1 Setup

[32]:

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.patches import Patch
import seaborn as sns
from os import getcwd

# Configure plots
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

# Connect to the SQLite database
#Run fetch_data.py in the data directory to update the database from the FHIS␣
  ↪website
home_path = getcwd()
print(home_path)
conn = sqlite3.connect("/mnt/d/renfrewshire_business_insights/data/
  ↪renfrewshire_hygiene.db") #adjust path accordingly
```

```
/mnt/d/renfrewshire_business_insights/reports
```

1.2 Overview of Data

[33]:

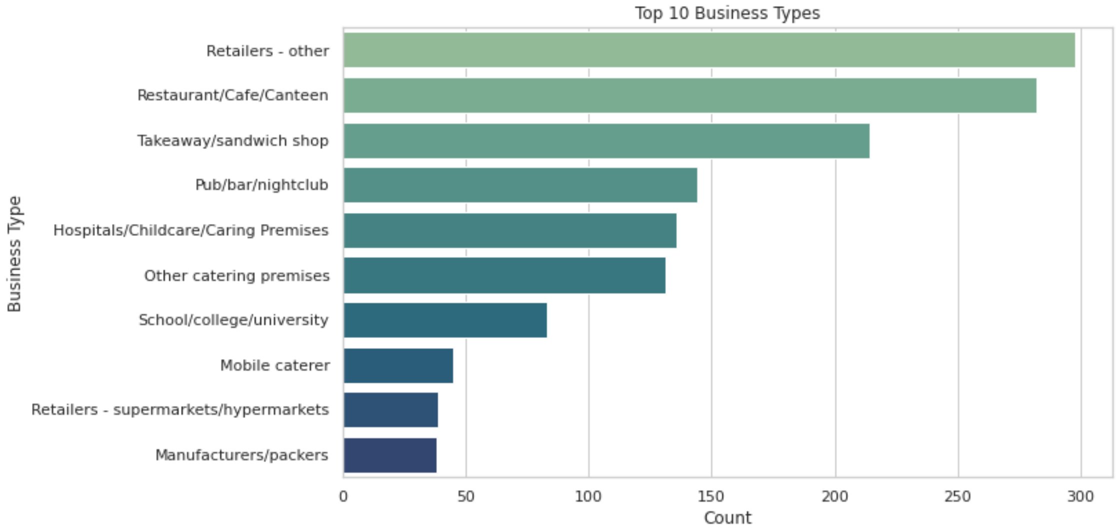
```
#Initial scoping of the SQL database to confirm all is working well
#List all entries
df = pd.read_sql_query("SELECT * FROM establishments;", conn)
#df.head(10)
```

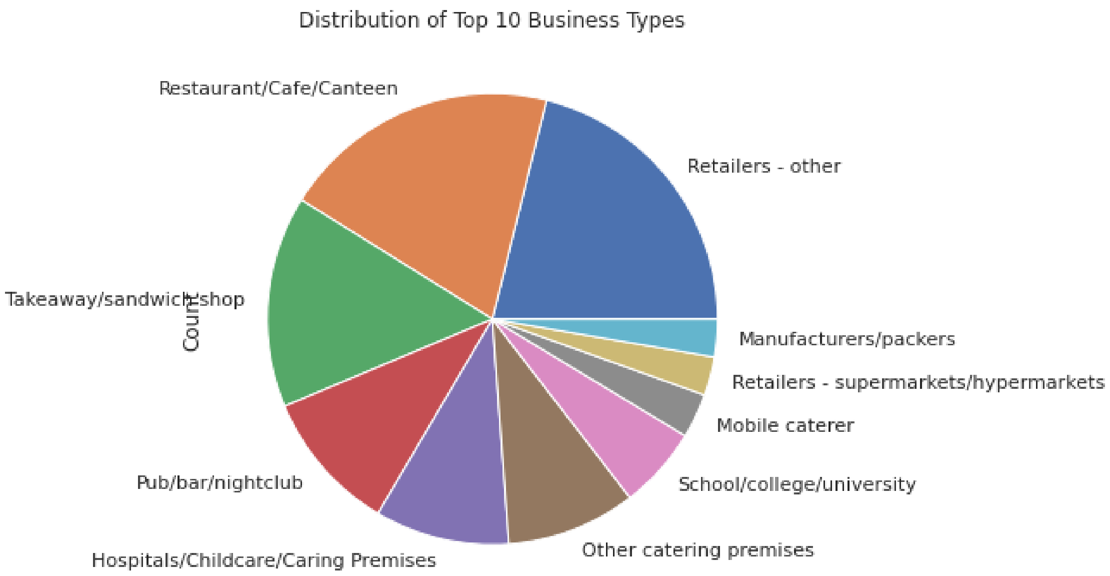
1.3 Top 10 Business Types by Count

[34]:

```
business_counts = pd.read_sql_query("""
SELECT BusinessType, COUNT(*) as Count
FROM establishments
GROUP BY BusinessType
ORDER BY Count DESC
LIMIT 10;
""", conn)

#Plotting
#print(business_counts)
sns.barplot(data=business_counts, x="Count", y="BusinessType", palette="crest")
plt.title("Top 10 Business Types")
plt.xlabel("Count")
plt.ylabel("Business Type")
plt.show()

#Pie chart
business_counts.plot.pie(y = "Count", labels = business_counts["BusinessType"],␣
 ↪legend = False)
plt.title("Distribution of Top 10 Business Types")
#plt.xlabel("Rating")
#plt.ylabel("Number of Establishments")
plt.show()
```

```
/tmp/ipykernel_1585/4050437959.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(data=business_counts, x="Count", y="BusinessType",
palette="crest")
```





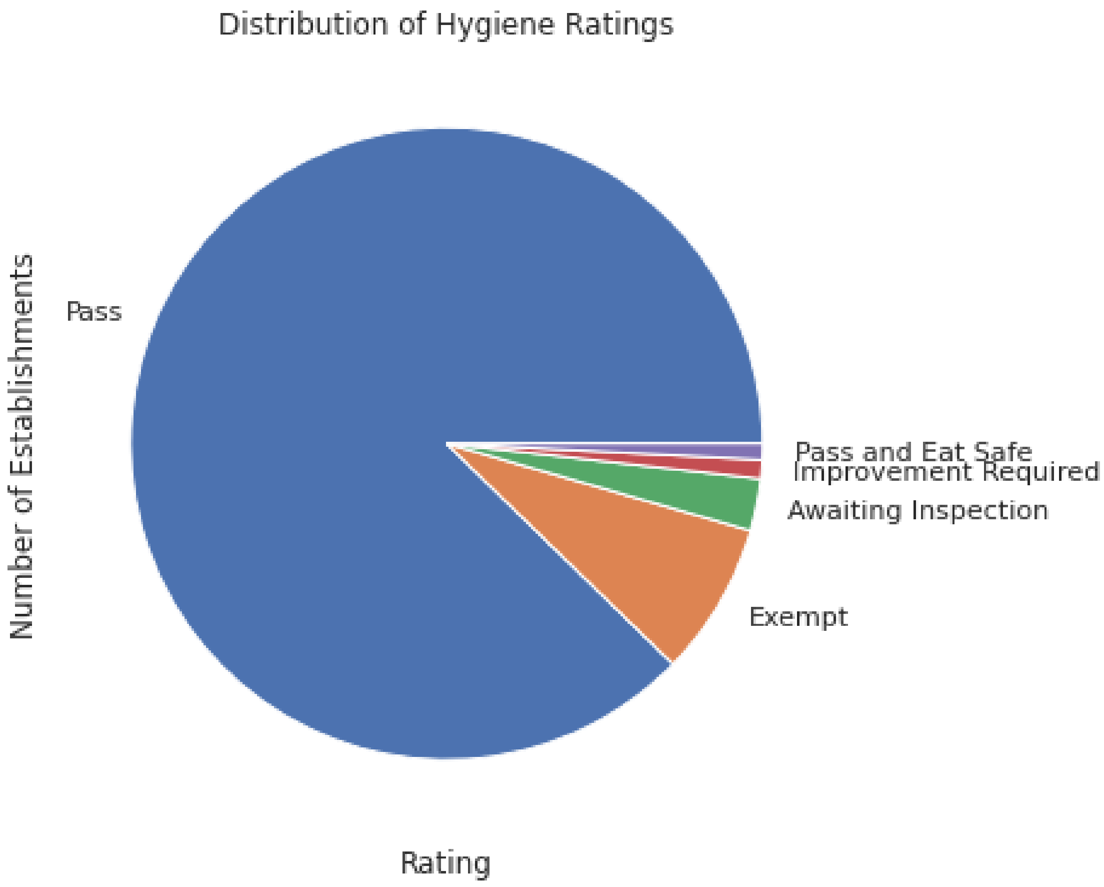
1.4 Hygiene Score Distribution

[35]:

```
# Get rating value counts
rating_counts = pd.read_sql_query("""
SELECT RatingValue, COUNT(*) as Count
FROM establishments
GROUP BY RatingValue
ORDER BY Count DESC
""", conn)

# Plot
print(rating_counts) # print the data frame
total = rating_counts["Count"].sum()
print("Total counts is: ", total)

rating_counts.plot.pie(y = "Count", labels = rating_counts["RatingValue"],␣
  ↪legend = False)
plt.title("Distribution of Hygiene Ratings")
plt.xlabel("Rating")
plt.ylabel("Number of Establishments")
plt.show()
```

```
            RatingValue  Count
0                  Pass   1285
1                Exempt    115
2   Awaiting Inspection     39
3  Improvement Required     14
```

```
4      Pass and Eat Safe     12
Total counts is:  1465
```

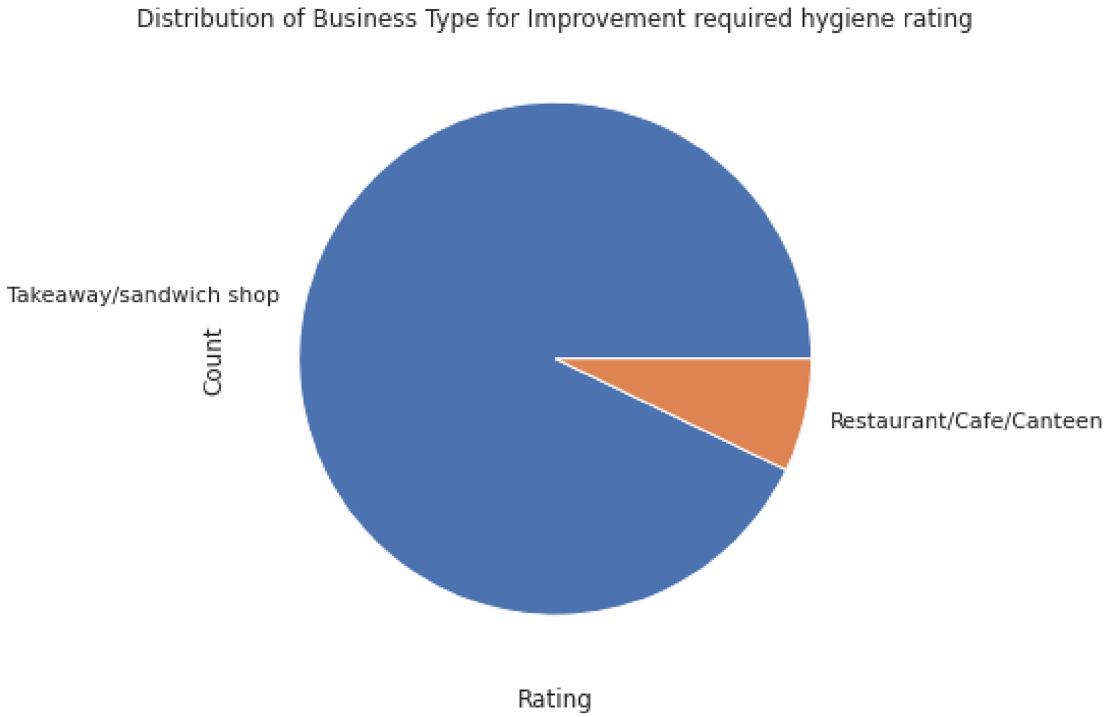


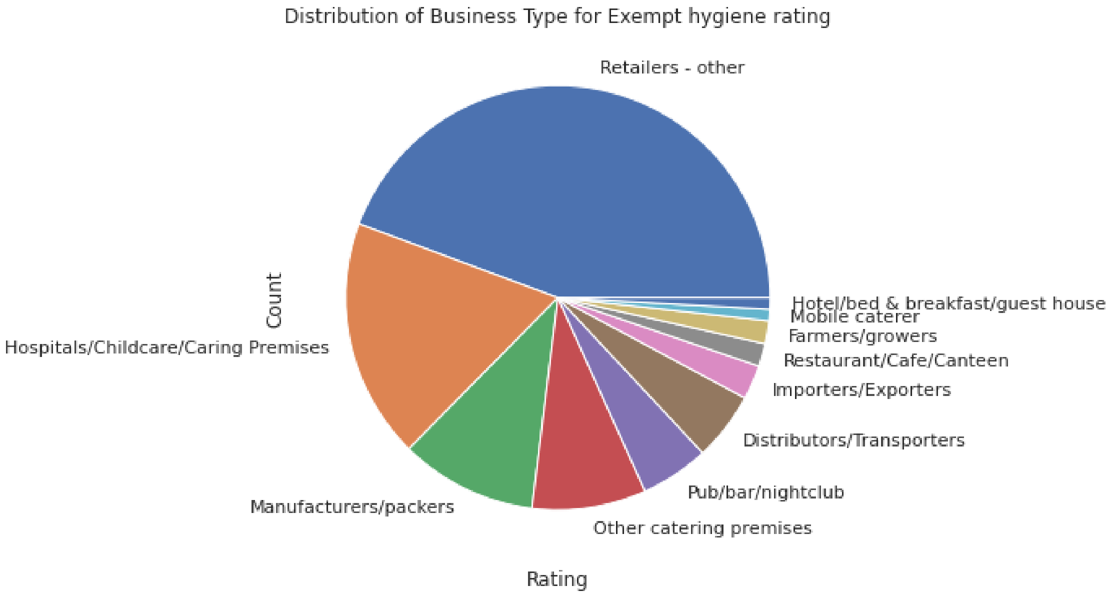
1.5 Deep dive into hygiene ratings

[36]:

```
#Get the improvement required ratings along with various parameters
improvement_required = pd.read_sql_query("""
SELECT RatingValue, BusinessType, COUNT(*) as Count
FROM establishments
WHERE RatingValue = 'Improvement Required'
GROUP BY BusinessType
ORDER BY Count DESC;
""", conn)

#Print
#print(improvement_required) # print the data frame

#Plotting
```

```
improvement_required.plot.pie(y = "Count", labels =␣
 ↪improvement_required["BusinessType"], legend = False)
plt.title("Distribution of Business Type for Improvement required hygiene␣
 ↪rating")
plt.xlabel("Rating")
#plt.ylabel("Number of Establishments")
plt.show()

#Get the Exempt ratings along with various parameters
exempt = pd.read_sql_query("""
SELECT RatingValue, BusinessType, COUNT(*) as Count
FROM establishments
WHERE RatingValue = 'Exempt'
GROUP BY BusinessType
ORDER BY Count DESC;
""", conn)

#Print
#print(exempt) # print the data frame

#Plotting
exempt.plot.pie(y = "Count", labels = exempt["BusinessType"], legend = False)
plt.title("Distribution of Business Type for Exempt hygiene rating")
plt.xlabel("Rating")
#plt.ylabel("Number of Establishments")
plt.show()

#Calculations using the global data frame
takeaway = df[df["BusinessType"] == "Takeaway/sandwich shop"]
#takeaway.head(10)
```





We can see that the majority of businesses that received an ‘Improvement Required’ score are classified as Takeawy/sandwich shop. It should be noted that Takeaway/sandwich shops take up a large proportion of the dataset but when comapred to the Restaurant/Cafe/Canteen category they are behind in food hygiene rating.

For the Exempt status we see a wide variety of business types. Exempt status is granted for businesses that don't produce their own food but do sell pre-packaged products and medicines which is corroborated here: https://essentialfoodhygiene.co.uk/what-are-the-three-food-hygiene-ratings-for-scotland/. This would explain large section being retailers - other.

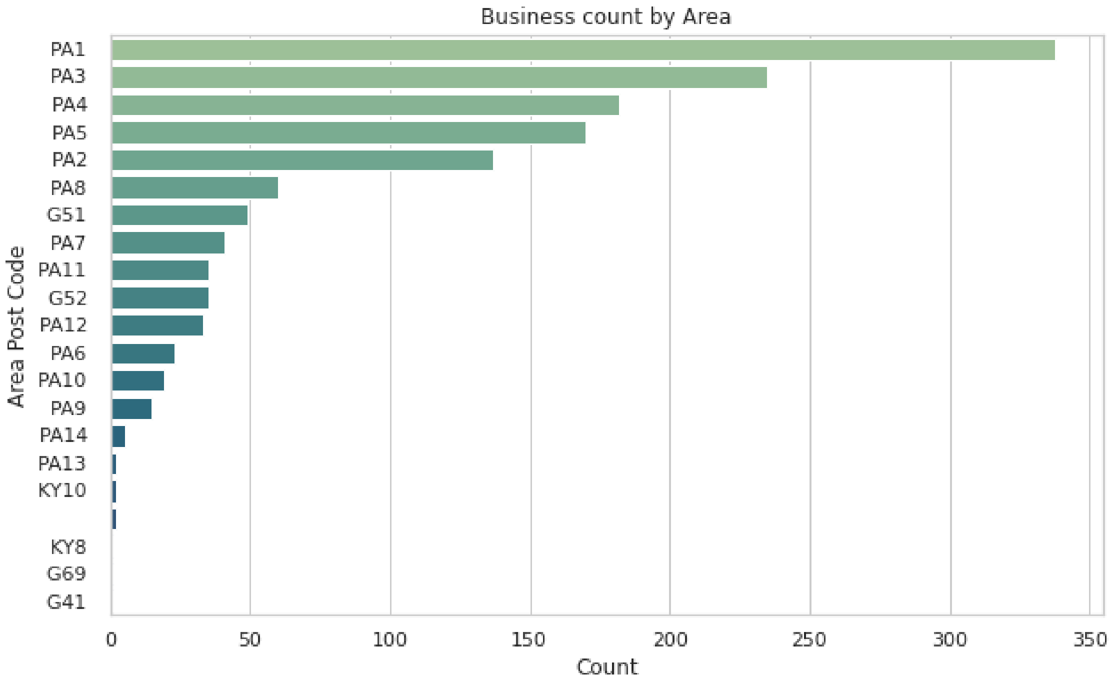
1.6 Businesses by Post Code

Using the intial part of a UK postcode, an indication of geographical area can be found. Lets find how many business fit in these areas.

[37]:

```
business_by_area = pd.read_sql_query("""
SELECT COUNT(*) as "Number of businesses", SUBSTR(PostCode, 1, instr(PostCode,␣
  ↪' ')) as PartPostCode
FROM establishments
GROUP BY PartPostCode
ORDER BY "Number of businesses" DESC;
""", conn)

#print(business_by_area)
sns.barplot(data=business_by_area, x="Number of businesses", y="PartPostCode",␣
  ↪palette="crest")
plt.title("Business count by Area")
plt.xlabel("Count")
plt.ylabel("Area Post Code")
plt.show()
```

```
/tmp/ipykernel_1585/1281992805.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(data=business_by_area, x="Number of businesses", y="PartPostCode",
palette="crest")
```



1.7 Map business location data using Geopandas

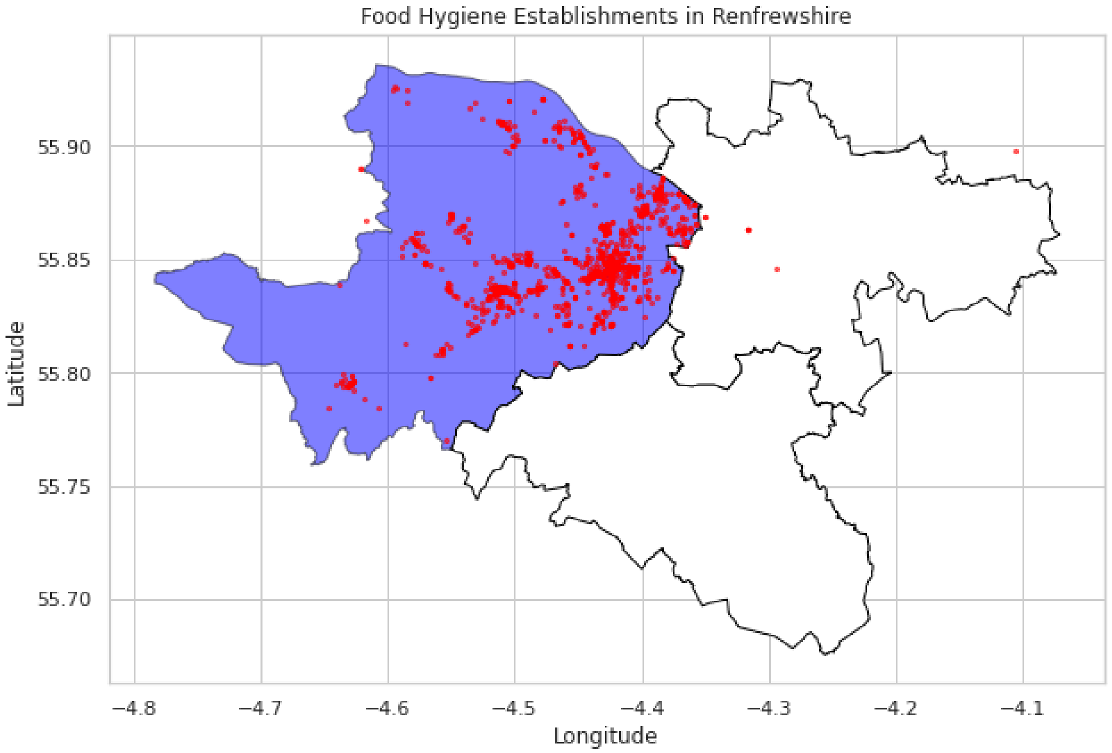
Using a shapefile for the local authority boundaries from the Improvement Service, the business location data can be placed on a map.

"The dataset is provided under Open Government Licence (OGL) for download and use. You are free to copy, publish, distribute and transmit the information as long as you acknowledge the source as coming from Improvement Service under OGL."

[41]:

```
import geopandas as gpd

#Get dataframe with outlier postcodes removed
df_geo = pd.read_sql_query("""
SELECT BusinessName, BusinessType, RatingValue, SUBSTR(PostCode, 1,␣
 ↪instr(PostCode, ' ')) as PartPostCode, Longitude, Latitude
FROM establishments
WHERE PartPostCode LIKE 'PA%' OR PartPostCode LIKE 'G%';
""", conn)

#Load Scottish local authority boundaries
#Please use the README to find instructions on how to download the boundary␣
 ↪shapefiles
la_gdf = gpd.read_file("/mnt/d/renfrewshire_business_insights/data/pub_las.
 ↪shp") #local authority GeoDataFrame
```

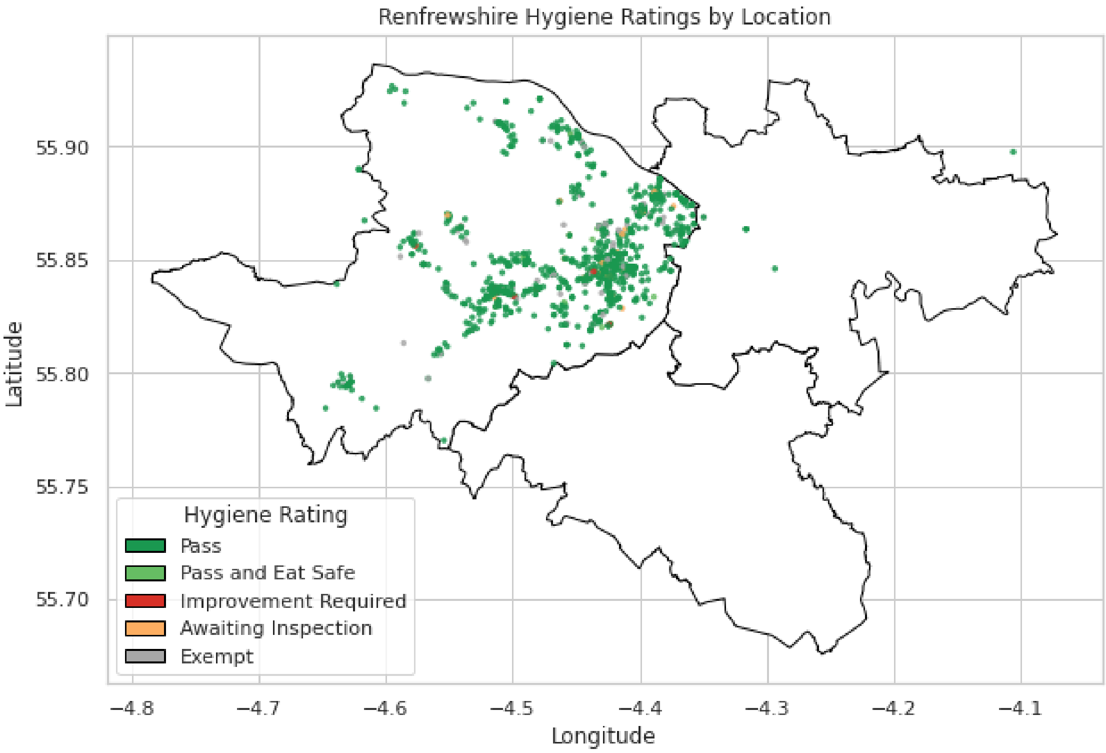
```python
#print(la_gdf.columns.tolist()) #print all available columns in the GeoDataFrame
#print("All available authority names: ") #Check all available authority names
#print(la_gdf["local_auth"].unique())  # Optional: inspect names

#Filter for Renfrewshire, East Renfrewshire and Glasgow City
ren_gdf = la_gdf[la_gdf["local_auth"] == "Renfrewshire"].copy()
east_ren_gdf = la_gdf[la_gdf["local_auth"] == "East Renfrewshire"].copy()
glasgow_gdf = la_gdf[la_gdf["local_auth"] == "Glasgow City"].copy()

#Filter out null coordinates from the hygiene dataframe
df_geo = df_geo[df_geo['Latitude'].notnull() & df_geo['Longitude'].notnull()]

#Convert DataFrame to GeoDataFrame
points_gdf = gpd.GeoDataFrame(
    df_geo,
    geometry=gpd.points_from_xy(df_geo.Longitude.astype(float), df_geo.Latitude.
 ↪astype(float)),
    crs="EPSG:4326"
)

#Ensure CRS matches
ren_gdf = ren_gdf.to_crs(epsg=4326)
east_ren_gdf = east_ren_gdf.to_crs(epsg=4326)
glasgow_gdf = glasgow_gdf.to_crs(epsg=4326)

#Plotting
fig, ax = plt.subplots(figsize=(10, 10))
ren_gdf.plot(ax=ax, color='blue', edgecolor='black', alpha = 0.5)
east_ren_gdf.plot(ax=ax, color='white', edgecolor='black')
glasgow_gdf.plot(ax=ax, color='white', edgecolor='black')
points_gdf.plot(ax=ax, markersize=5, alpha=0.6, color='red')
plt.title("Food Hygiene Establishments in Renfrewshire")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()
```



[39]:

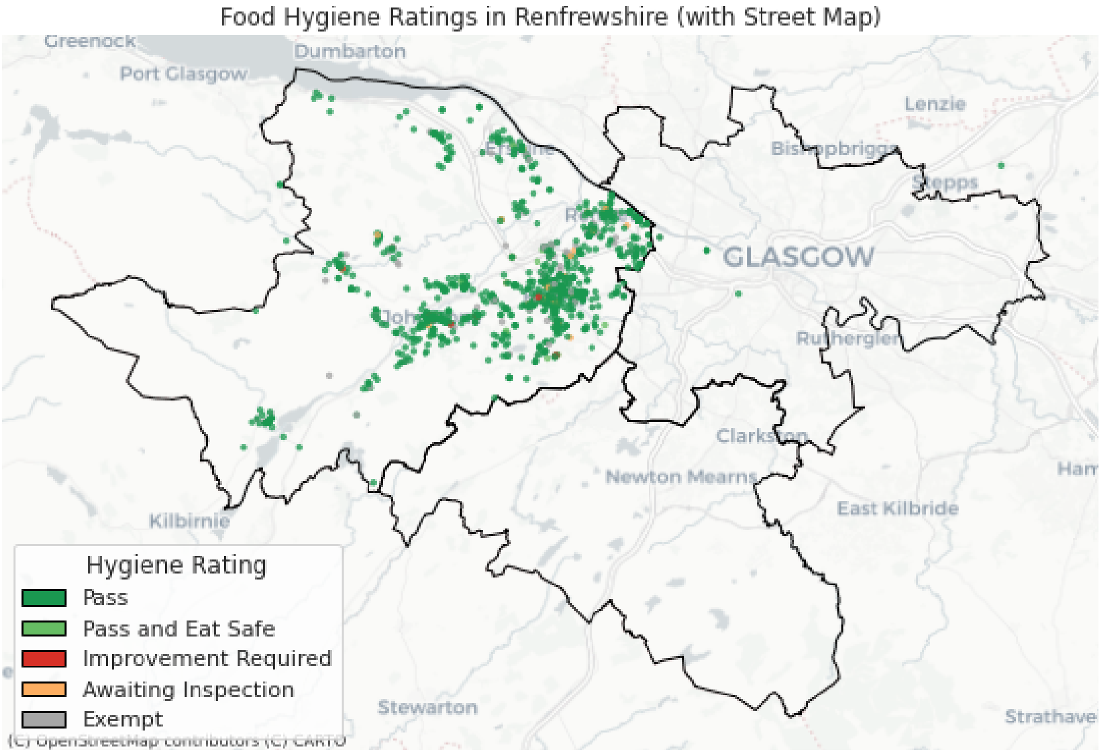
```
#Generate the local authority boundaries to a single GeoDataFrame
included_areas = ["Renfrewshire", "East Renfrewshire", "Glasgow City"]
boundary_gdf = la_gdf[la_gdf["local_auth"].isin(included_areas)].copy()␣
  ↪#boundary dataframe
boundary_gdf = boundary_gdf.to_crs(epsg=4326) #covnert to consitent CRS

# Map each string rating to a colour
rating_colors = {
    "Pass": "#1a9850",                # green
    "Pass and Eat Safe": "#66bd63",   # light green
    "Improvement Required": "#d73027",# red
    "Awaiting Inspection": "#fdae61", # orange
    "Exempt": "#a6a6a6"               # grey
}

#Create a legend for the plot
legend_elements = [
    Patch(facecolor=color, edgecolor='black', label=label)
    for label, color in rating_colors.items()
]

# Map rating to colours by adding a colour column to the dataframe
points_gdf["color"] = points_gdf["RatingValue"].map(rating_colors)
```

```
# #Diagnostics
# print(points_gdf[["Longitude", "Latitude", "geometry"]].head())
# print(points_gdf.geom_type.unique())
# print(points_gdf.crs)

#Plot
fig, ax = plt.subplots(figsize=(10, 10))
boundary_gdf.plot(ax=ax, color="white", edgecolor="black")
points_gdf.plot(ax=ax, markersize=6, color=points_gdf["color"], alpha=0.8)

plt.title("Renfrewshire Hygiene Ratings by Location")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.grid(True)
plt.legend(handles=legend_elements, title="Hygiene Rating", loc='lower left')
plt.show()
```



[40]:
```
#Use contextily to plot street map underneath plots
import contextily as ctx

#Reproject both GeoDataFrames to EPSG:3857 (Web Mercator)
points_web = points_gdf.to_crs(epsg=3857)
```

```
boundary_web = boundary_gdf.to_crs(epsg=3857)

fig, ax = plt.subplots(figsize=(10, 10))

#Plot boundary outline
boundary_web.plot(ax=ax, color='none', edgecolor='black')
points_web.plot(ax=ax, markersize=6, color=points_web['color'], alpha=0.7) #Plot hygiene points (coloured by rating, as before)

#Add basemap tiles
#ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik) #OpenMap full colour
ctx.add_basemap(ax, source=ctx.providers.CartoDB.Positron) #grayscale overlay

plt.title("Food Hygiene Ratings in Renfrewshire (with Street Map)")
plt.axis("off")
plt.legend(handles=legend_elements, title="Hygiene Rating", loc='lower left')
plt.show()
```



1.8 Interactive map using folium based explore method for GeoPandas

[47]:
```
points_web.explore(
    column = "RatingValue",
    tiles = "CartoDB  positron",
    popup = True,
    cmap = "jet_r"
)

#generate cmap later that matches previous colour scheme
```

[47]:
```
<folium.folium.Map at 0x7fc4ba74c520>
```

1.8.1 Outliers

There are three postcodes that start in KY, which is not a post code in the vicinity of Renfrewshire. We can look at these results more closely to determine why this could be.

[11]:
```
outliers = pd.read_sql_query("""
SELECT BusinessName, BusinessType, PostCode, AddressLine1,
AddressLine2, AddressLine3, AddressLine4
FROM establishments
WHERE PostCode LIKE "KY%";
""", conn)

outliers.head()
```

[11]:
```
            BusinessName       BusinessType  PostCode AddressLine1  \
0  DM Fish Merchants Ltd     Mobile caterer   KY8 1HQ         None
1       DNM Fish Mechant  Retailers - other  KY10 3YP         None
2           Sandra Hodge  Retailers - other   KY103HE         None
3            Yvonne Dehn     Mobile caterer  KY10 3YN         None

               AddressLine2 AddressLine3 AddressLine4
0             55 Mavis Bank    Buckhaven         None
1  23 Lindsay Berwick Place   Anstruther         None
2       1 St Ayles Crescent   Anstruther         None
3          13 North Marches   Anstruther         Fife
```

Two of the results are mobile caterers which could explain the lack of a business address within the confines of Renfrewshire.

1.9 Save and Close

[12]:
```
conn.close()
```

[ ]:
```

```