main

October 14, 2022

[177]:
```
import numpy as np
import pandas as pd
```

[178]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("whitegrid")
```

[179]:
```
data = pd.read_csv("fire_cases_in_uk_last_3_years.csv")
data.head()
```

[179]:

	IncidentNumber	DateOfCall	CalYear	TimeOfCall	HourOfCall	IncidentGroup	StopCodeDescription	SpecialServiceType	PropertyCategory	PropertyType	…	FirstPumpArriving_AttendanceTime	FirstPumpArriving_DeployedFromStation
0	000006-01012019	01 Jan 2019	2019	00:01:45	0	Special Service	Special Service	Lift Release	Dwelling	Purpose Built Flats/Maisonettes - 4 to 9 storeys	…	NaN	NaN
1	000019-01012019	01 Jan 2019	2019	00:04:33	0	Fire	Secondary Fire	NaN	Outdoor	Tree scrub	…	357.0	Edmonton
2	000020-01012019	01 Jan 2019	2019	00:04:39	0	False Alarm	False alarm - Good intent	NaN	Outdoor	Domestic garden (vegetation not equipment)	…	318.0	Southgate
3	000021-01012019	01 Jan 2019	2019	00:04:44	0	False Alarm	AFA	NaN	Dwelling	Stately Home (part not open to public)	…	210.0	Kensington
4	000024-01012019	01 Jan 2019	2019	00:05:00	0	Special Service	Special Service	Lift Release	Dwelling	Purpose Built Flats/Maisonettes - 4 to 9 storeys	…	329.0	Bethnal Green

```
  SecondPumpArriving_AttendanceTime  SecondPumpArriving_DeployedFromStation  \
0                               NaN                                     NaN
1                               NaN                                     NaN
2                               NaN                                     NaN
3                               NaN                                     NaN
4                               NaN                                     NaN

   NumStationsWithPumpsAttending NumPumpsAttending PumpCount PumpHoursRoundUp  \
0                            1.0               1.0       1.0              1.0
1                            1.0               1.0       1.0              1.0
2                            1.0               1.0       1.0              1.0
3                            1.0               1.0       1.0              1.0
4                            1.0               1.0       1.0              1.0

  Notional Cost (£) NumCalls
0             333.0      2.0
1             333.0      1.0
2             333.0      1.0
3             333.0      1.0
4             333.0      1.0

[5 rows x 39 columns]
```

[180]:
```
data.columns
```

[180]:
```
Index(['IncidentNumber', 'DateOfCall', 'CalYear', 'TimeOfCall', 'HourOfCall',
       'IncidentGroup', 'StopCodeDescription', 'SpecialServiceType',
       'PropertyCategory', 'PropertyType', 'AddressQualifier', 'Postcode_full',
       'Postcode_district', 'UPRN', 'USRN', 'IncGeo_BoroughCode',
       'IncGeo_BoroughName', 'ProperCase', 'IncGeo_WardCode',
       'IncGeo_WardName', 'IncGeo_WardNameNew', 'Easting_m', 'Northing_m',
       'Easting_rounded', 'Northing_rounded', 'Latitude', 'Longitude', 'FRS',
       'IncidentStationGround', 'FirstPumpArriving_AttendanceTime',
       'FirstPumpArriving_DeployedFromStation',
       'SecondPumpArriving_AttendanceTime',
       'SecondPumpArriving_DeployedFromStation',
       'NumStationsWithPumpsAttending', 'NumPumpsAttending', 'PumpCount',
       'PumpHoursRoundUp', 'Notional Cost (£)', 'NumCalls'],
      dtype='object')
```

[181]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331570 entries, 0 to 331569
Data columns (total 39 columns):
 #   Column                                  Non-Null Count   Dtype
```

```
---  ------                                    --------------   -----
 0   IncidentNumber                            331570 non-null  object
 1   DateOfCall                                331570 non-null  object
 2   CalYear                                   331570 non-null  int64
 3   TimeOfCall                                331570 non-null  object
 4   HourOfCall                                331570 non-null  int64
 5   IncidentGroup                             331570 non-null  object
 6   StopCodeDescription                       331570 non-null  object
 7   SpecialServiceType                        112570 non-null  object
 8   PropertyCategory                          331570 non-null  object
 9   PropertyType                              331570 non-null  object
 10  AddressQualifier                          331570 non-null  object
 11  Postcode_full                             150622 non-null  object
 12  Postcode_district                         331570 non-null  object
 13  UPRN                                      331570 non-null  int64
 14  USRN                                      331570 non-null  int64
 15  IncGeo_BoroughCode                        331570 non-null  object
 16  IncGeo_BoroughName                        331570 non-null  object
 17  ProperCase                                331570 non-null  object
 18  IncGeo_WardCode                           331569 non-null  object
 19  IncGeo_WardName                           331569 non-null  object
 20  IncGeo_WardNameNew                        331569 non-null  object
 21  Easting_m                                 150622 non-null  float64
 22  Northing_m                                150622 non-null  float64
 23  Easting_rounded                           331570 non-null  int64
 24  Northing_rounded                          331570 non-null  int64
 25  Latitude                                  150622 non-null  float64
 26  Longitude                                 150622 non-null  float64
 27  FRS                                       331570 non-null  object
 28  IncidentStationGround                     331570 non-null  object
 29  FirstPumpArriving_AttendanceTime          311815 non-null  float64
 30  FirstPumpArriving_DeployedFromStation     311810 non-null  object
 31  SecondPumpArriving_AttendanceTime         125981 non-null  float64
 32  SecondPumpArriving_DeployedFromStation    125980 non-null  object
 33  NumStationsWithPumpsAttending             327357 non-null  float64
 34  NumPumpsAttending                         327357 non-null  float64
 35  PumpCount                                 329467 non-null  float64
 36  PumpHoursRoundUp                          329362 non-null  float64
 37  Notional Cost (£)                         329362 non-null  float64
 38  NumCalls                                  331566 non-null  float64
dtypes: float64(12), int64(6), object(21)
memory usage: 98.7+ MB
```

0.1 Prepare data

[182]:
```
data["time"] = pd.to_datetime(data["TimeOfCall"])
data["date"] = pd.to_datetime(data["DateOfCall"])
data.drop(columns = ["TimeOfCall", "DateOfCall"], inplace = True)
```

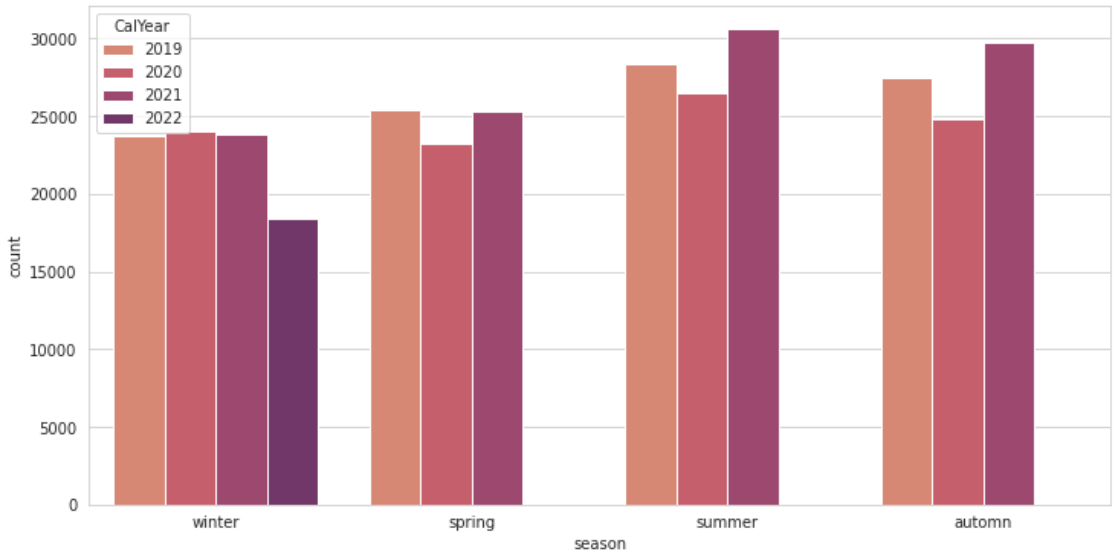
[183]:
```
def season(month):
    if month < 3 or month == 12:
        return "winter"
    elif 3 <= month < 6:
        return "spring"
    elif 6 <= month < 9:
        return "summer"
    else:
        return "automn"
def day_time(hour):
    if hour <= 6:
        return "night"
    if hour <= 12:
        return "morning"
    elif hour <= 18:
        return "afternoon"
    else:
        return "evening"
```

[184]:
```
dayofweek = {0: "Mon", 1: "Tue", 2: "Wed", 3: "Thu", 4: "Fri", 5: "Sat", 6:␣
 ↪"Sun"}
data["season"] = data["date"].apply(lambda x: season(x.month))
data["DayTime"] = data["HourOfCall"].apply(lambda x: day_time(x))
data["DayOfWeek"] = data["date"].apply(lambda x: dayofweek[x.dayofweek])
```

0.2 Visualize

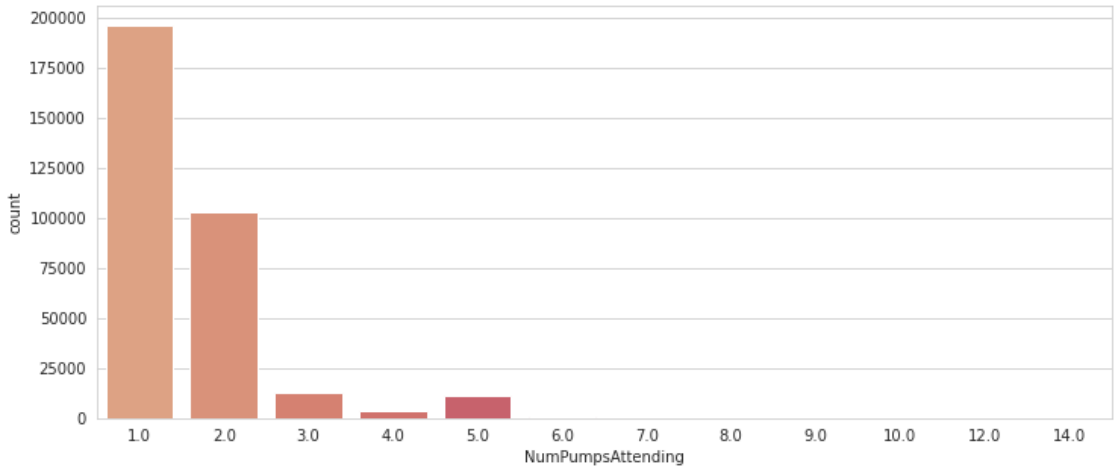
[185]:
```
plt.figure(figsize = (12, 6))
sns.countplot(x = "season", data = data, hue = "CalYear",   palette= "flare")
```

[185]:
```
<AxesSubplot:xlabel='season', ylabel='count'>
```



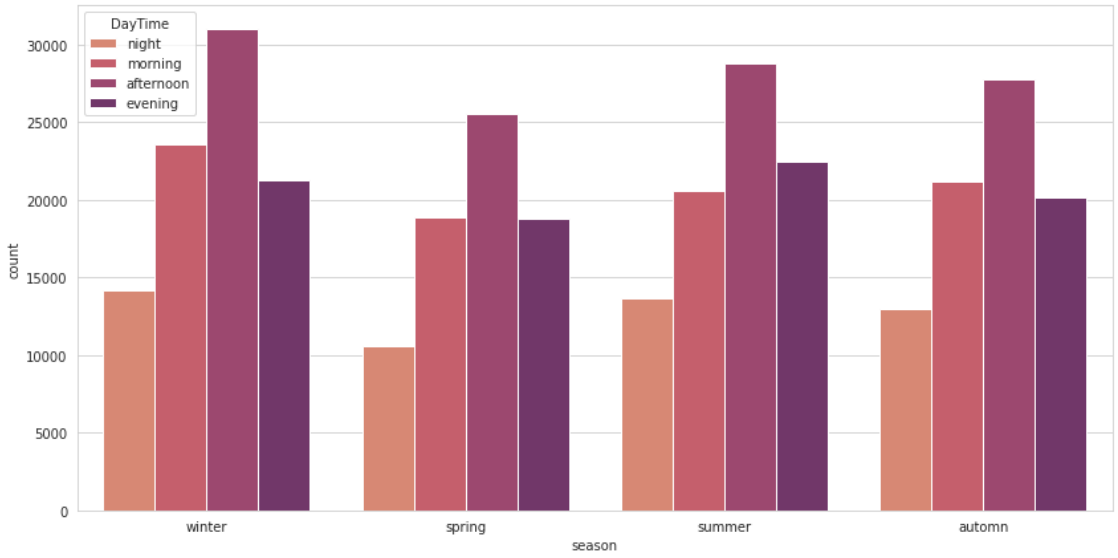
[186]:
```
plt.figure(figsize = (12, 5))
sns.countplot(x = "NumPumpsAttending",hue = None ,data = data,  palette=␣
↪"flare")
```

[186]: <AxesSubplot:xlabel='NumPumpsAttending', ylabel='count'>



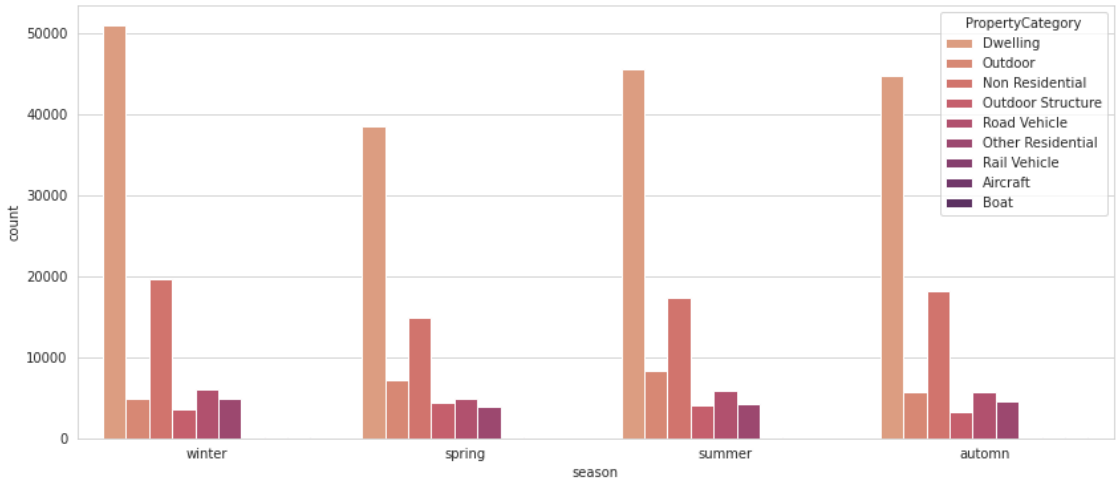
[187]:
```
plt.figure(figsize = (14, 7))
sns.countplot(x = "season", data = data, hue = "DayTime", palette= "flare")
```

[187]: <AxesSubplot:xlabel='season', ylabel='count'>



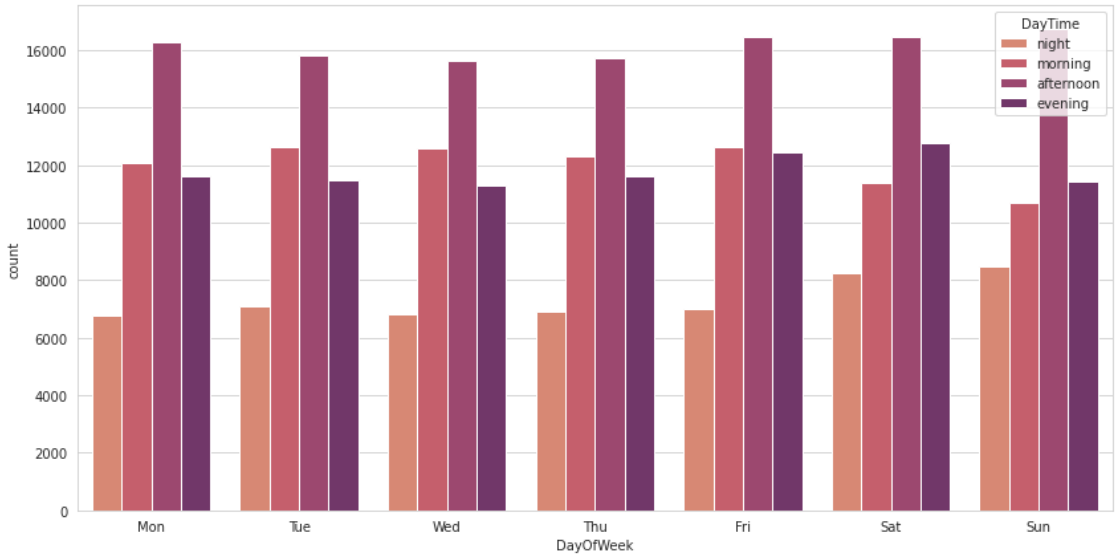
[188]:
```
plt.figure(figsize = (14, 6))
sns.countplot(x = data["season"], hue = data["PropertyCategory"],  palette=␣
↪"flare")
```

[188]: <AxesSubplot:xlabel='season', ylabel='count'>

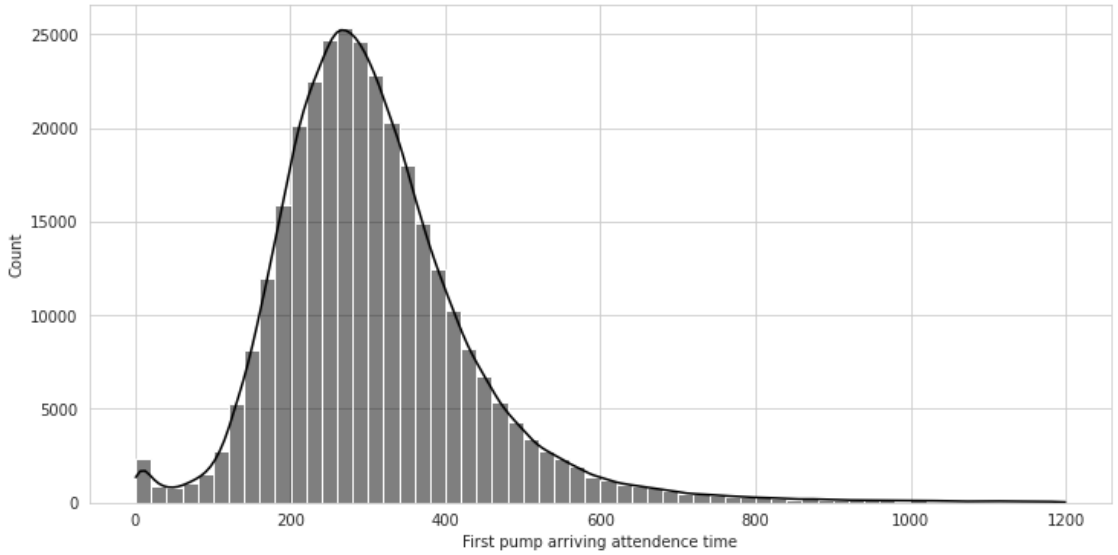


[189]:
```
plt.figure(figsize = (14, 7))
sns.countplot(x = "DayOfWeek", hue = "DayTime", data = data, order = "Mon Tue␣
↪Wed Thu Fri Sat Sun".split(),  palette= "flare")
```

[189]: <AxesSubplot:xlabel='DayOfWeek', ylabel='count'>



[190]:
```
plt.figure(figsize = (12, 6))
plt.xlabel("First pump arriving attendence time")
s = sns.histplot(x = "FirstPumpArriving_AttendanceTime", data = data, kde =␣
 ↪True, color = "black", bins = 60)
```



[194]:
```
data1 = data[data["Latitude"] > 0] #data with longitude and latitude
x = data1["Longitude"]
y = data1["Latitude"]
```

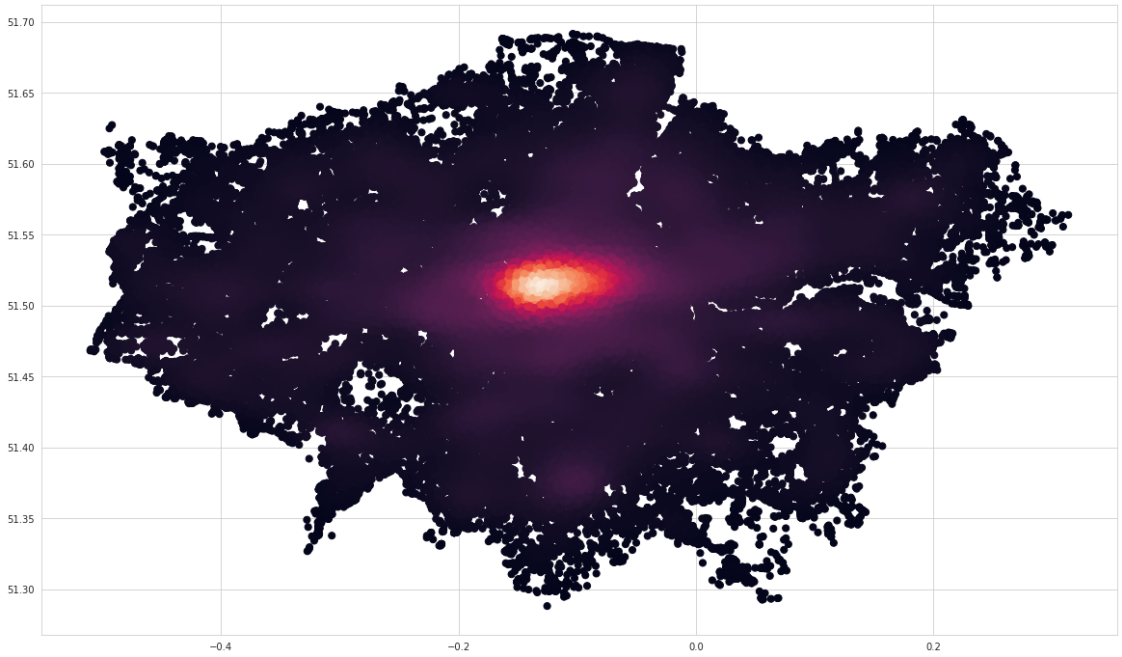
[195]:
```
from scipy.stats import gaussian_kde
plt.figure(figsize = (20, 12))
xy = np.vstack([x,y])

z = gaussian_kde(xy)(xy)
print(z)

plt.scatter(x = "Longitude", y = "Latitude", data = data1, c = z, s = 50)
```

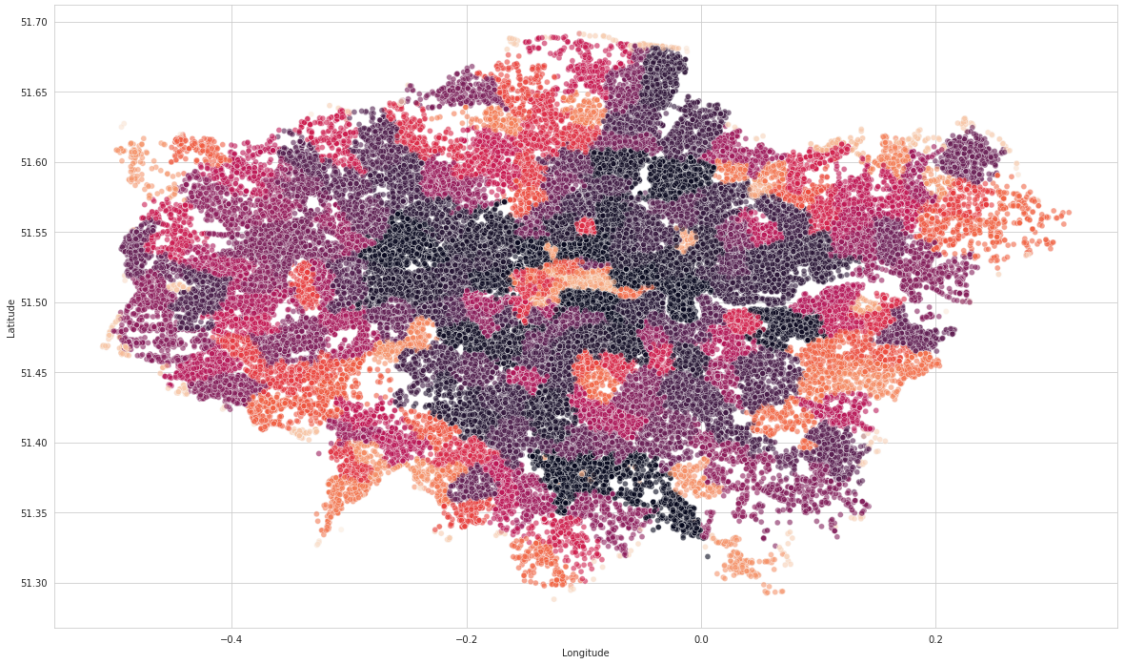
```
[ 4.63769736  2.51926884  7.20319826 …  1.99557644  8.96463305
 34.3203302 ]
```

[195]:
```
<matplotlib.collections.PathCollection at 0x7f1a75f3b520>
```



[197]:
```
plt.figure(figsize = (20, 12))
sns.scatterplot(x = "Longitude", y = "Latitude", data = data1, hue =␣
 ↪"Postcode_district",
                alpha = 0.6 ,legend = False, palette = "rocket",
                hue_order = data["Postcode_district"].value_counts().index)
# The darkest areas are areas with the biggest amount of fires
```

[197]:
```
<AxesSubplot:xlabel='Longitude', ylabel='Latitude'>
```


51.70
51.65
51.60
51.55
51.50
51.45
51.40
51.35
51.30
Latitude
−0.4
−0.2
0.0
0.2
Longitude

[ ]:

[ ]: