Electric Vehicle Accident Analysis , Israel 2018-19

Importing the libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import osmnx as ox
import networkx as nx
import matplotlib.colors as colors
import  matplotlib.cm as cm
import geopandas as gpd
import plotly.express as px

import plotly
import plotly.offline as plty
plt.rc('axes', labelsize=14)

from shapely.geometry import Point
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

Graph Of Tel Aviv, Israel

In [2]:
```
ox.plot_graph(ox.graph_from_place("Tel Aviv,Israel"))
```

Out[2]: (<Figure size 576x576 with 1 Axes>, <AxesSubplot:>)

In [3]:
```
city = ox.geocode_to_gdf("Tel Aviv,Israel")
ax = ox.project_gdf(city).plot()
_ = ax.axis('off')
```

Administrative boundary vector geometries retrieved for Tel Aviv,Israel

Reading the data file (file-electricvehicleaccidentbs.csv)

In [2]:
```
df = pd.read_csv('/Users/divyabisht/Desktop/electricvehicleaccidentbs.csv')
```

In [104]:
```
df.head()
```

Out[104]:

	accident_id	provider_and_id	provider_code	file_type_police	involved_type	involved_type_hebrew	license_acquiring_date	age_group	age_g
0	2018013760	32018013760	3	3	3	נפגע	0	18	
1	2018013760	32018013760	3	3	1	נהג	0	99	
2	2018042928	32018042928	3	3	1	נהג	2011	7	
3	2018042928	32018042928	3	3	2	נהג נפגע	0	7	
4	2018078809	32018078809	3	3	2	נהג נפגע	2000	8	

5 rows × 118 columns

In [4]:
```
df.shape
```

Out[4]:
```
(7907, 118)
```

Number of Rows in Dataset: 7907

Number of Columns in Dataset:118

In [4]:
```
desc = df.describe()
desc
```

Out[4]:

	accident_id	provider_and_id	provider_code	file_type_police	involved_type	license_acquiring_date	age_group	sex	involve
count	7.907000e+03	7.907000e+03	7907.000000	7907.000000	7907.000000	7907.000000	7907.000000	7083.000000	
mean	2.018520e+09	2.687371e+10	2.485519	2.512837	1.606804	1010.266220	26.900468	1.112099	
std	5.002365e+05	8.742830e+09	0.874281	0.858540	0.626443	999.072692	36.710342	0.567524	
min	2.018000e+09	1.201800e+10	1.000000	1.000000	1.000000	0.000000	1.000000	0.000000	
25%	2.018048e+09	1.201909e+10	1.000000	3.000000	1.000000	0.000000	6.000000	1.000000	
50%	2.018094e+09	3.201806e+10	3.000000	3.000000	2.000000	1963.000000	9.000000	1.000000	
75%	2.019046e+09	3.201903e+10	3.000000	3.000000	2.000000	2000.000000	15.000000	1.000000	
max	2.019100e+09	3.201910e+10	3.000000	3.000000	3.000000	2019.000000	99.000000	2.000000	

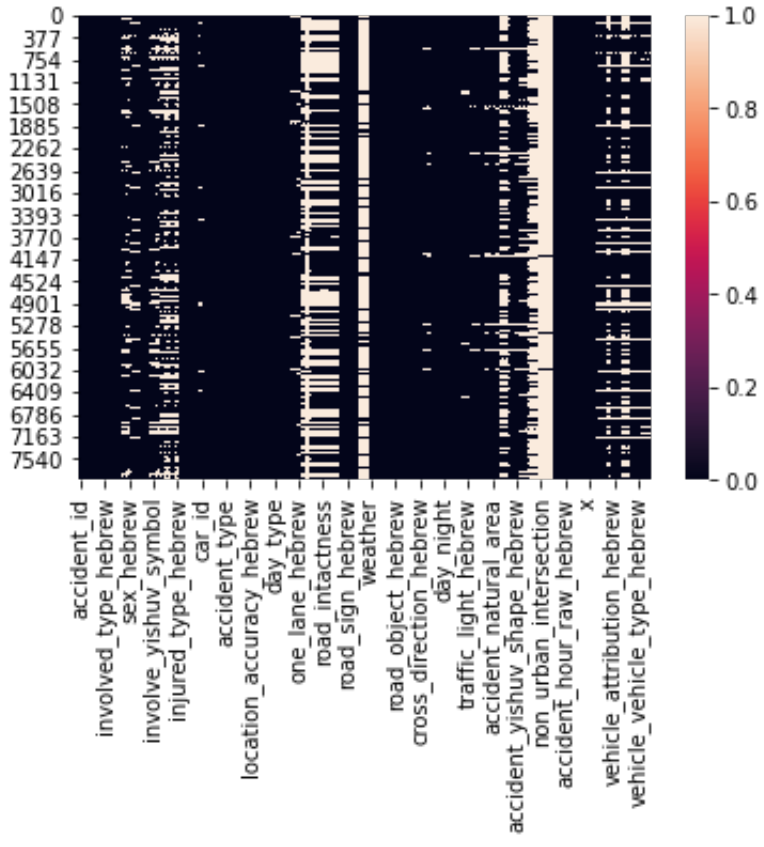
8 rows × 66 columns

In [5]:
```
print(df.dtypes)
```

```
accident_id                    int64
provider_and_id                int64
provider_code                  int64
file_type_police               int64
involved_type                  int64
                               ...
total_weight_hebrew           object
vehicle_vehicle_type         float64
vehicle_vehicle_type_hebrew   object
vehicle_damage               float64
vehicle_damage_hebrew         object
Length: 118, dtype: object
```

In [6]:
```
sns.heatmap(df.isnull())
```

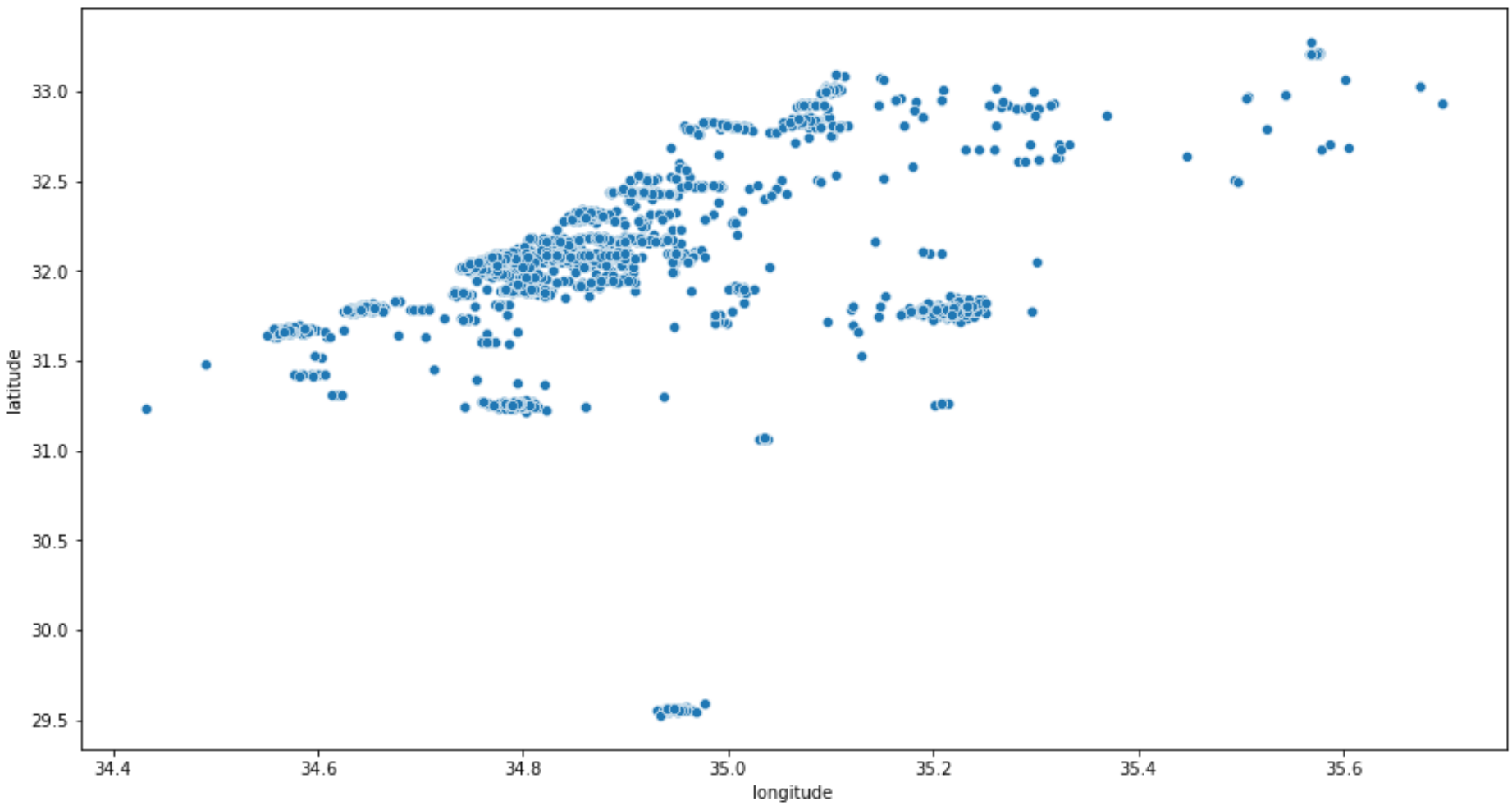
Out[6]: <AxesSubplot:>



The white space shows null values.

In [7]:
```
plt.figure(figsize = (15,8))
sns.scatterplot(df['longitude'], df['latitude'])
```

Out[7]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [8]:
```
df_geo = gpd.GeoDataFrame(df,geometry = gpd.points_from_xy(df.longitude,df.latitude))
```

In [9]:
```
df_geo
```

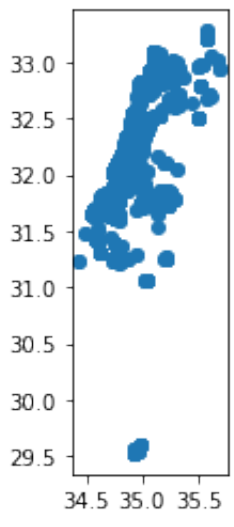
Out[9]:

	accident_id	provider_and_id	provider_code	file_type_police	involved_type	involved_type_hebrew	license_acquiring_date	age_group	ag
0	2018013760	32018013760	3	3	3	נפגע	0	18	
1	2018013760	32018013760	3	3	1	נהג	0	99	
2	2018042928	32018042928	3	3	1	נהג	2011	7	
3	2018042928	32018042928	3	3	2	נהג נפגע	0	7	
4	2018078809	32018078809	3	3	2	נהג נפגע	2000	8	
...	...	...	...	...	...	...	...	...	
7902	2018005642	12018005642	1	1	2	נהג נפגע	2010	7	
7903	2018056690	12018056690	1	1	1	נהג	2003	7	
7904	2018056690	12018056690	1	1	2	נהג נפגע	0	8	
7905	2018084262	12018084262	1	1	1	נהג	1972	15	
7906	2018084262	12018084262	1	1	2	נהג נפגע	0	4	

7907 rows × 119 columns

In [10]:
```
df_geo.plot()
```

Out[10]: <AxesSubplot:>



In [3]:
```
Title = " "
def PlotPiechart(labels, values, columnName):
    fig ={
      "data":[
          {
         "labels": labels,
          "values": values['provider_and_id'],
          #"domain": {"x": [0, 1]},
          "name": columnName,
          "hoverinfo":"label+percent+name",
          "hole": .6,
          "type": "pie"
          },
      ],
      "layout": {
           # "title":"Percentage of Accident happened in situations : " + colum
           "title":Title,
           "annotations": [
{
"font": {
"size": 40 },
                    "showarrow": False,
                    "text": " ",
                    "x": 5.50,
                    "y": 0.5
} ]
} }
    plty.iplot(fig)
```

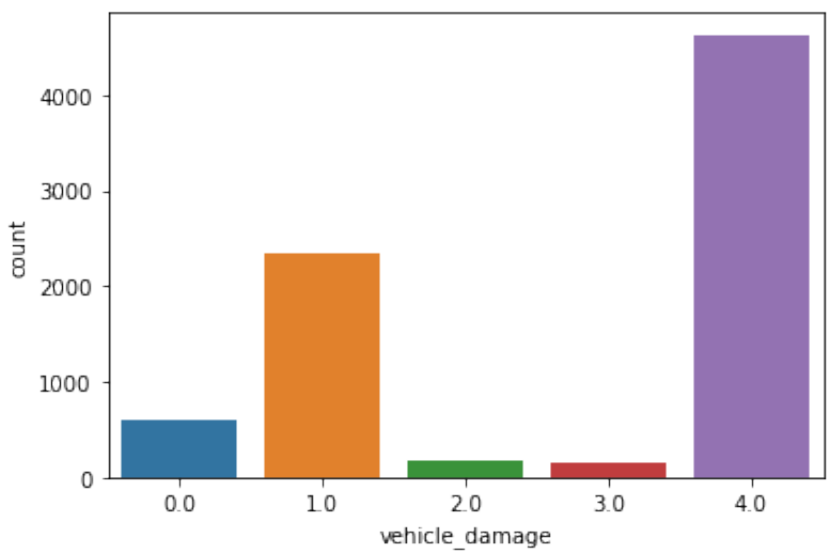
Vehicle Damaged

To fill not a number values to 0.0 so they all belong to same data type for column vehicle damages

In [7]:
```
df.vehicle_damage = df.vehicle_damage.fillna(0.0)
```

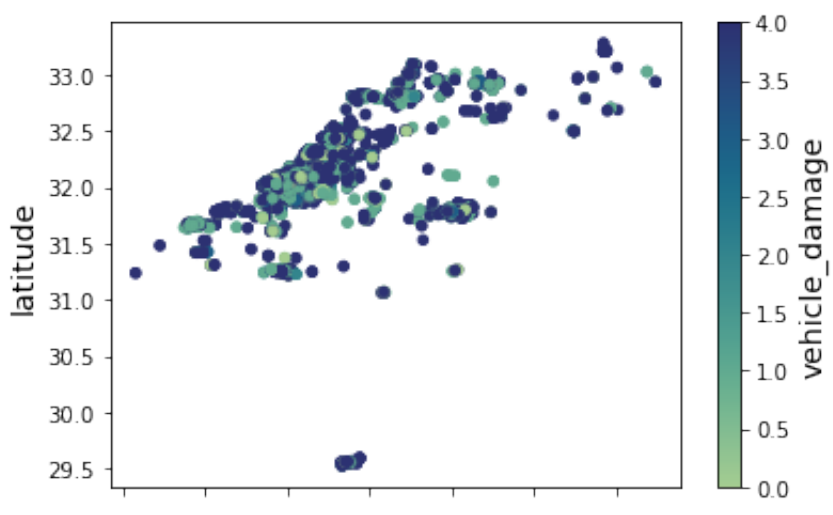
In [8]:
```
sns.countplot(df['vehicle_damage'])
```

Out[8]: <AxesSubplot:xlabel='vehicle_damage', ylabel='count'>

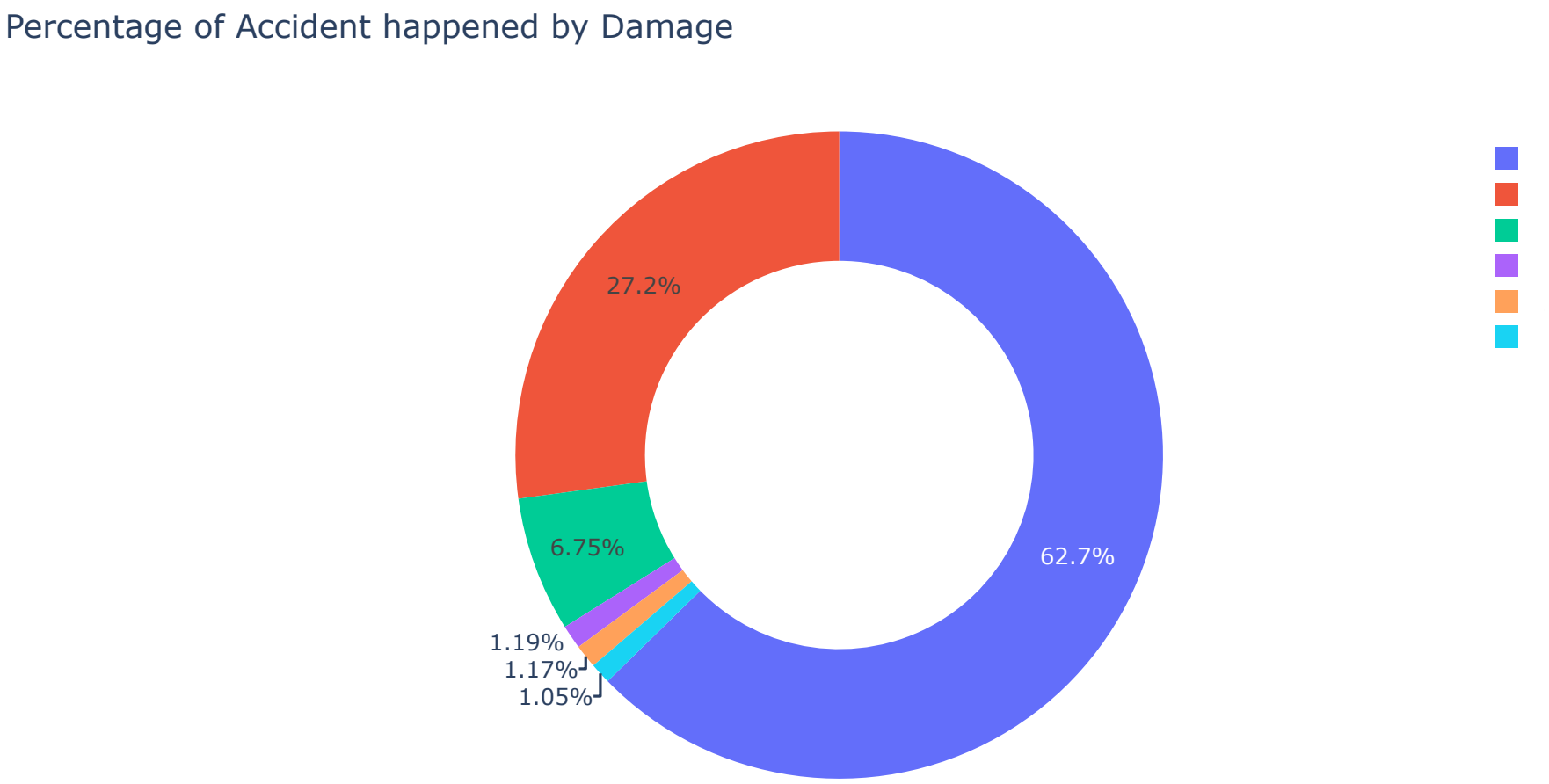


In [82]:
```
df.plot(x="longitude", y="latitude", kind="scatter", c="vehicle_damage",
        colormap="crest")
```

Out[82]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [4]:
```
Title = "Percentage of Accident happened by Damage"
PlotPiechart(df['vehicle_damage_hebrew'].values, df, 'Damage')
```



Vehicle Damaged during accident

0.0 - Unknown/לא ידוע

1.0 - easy/קל

2.0 - medium/בינוני

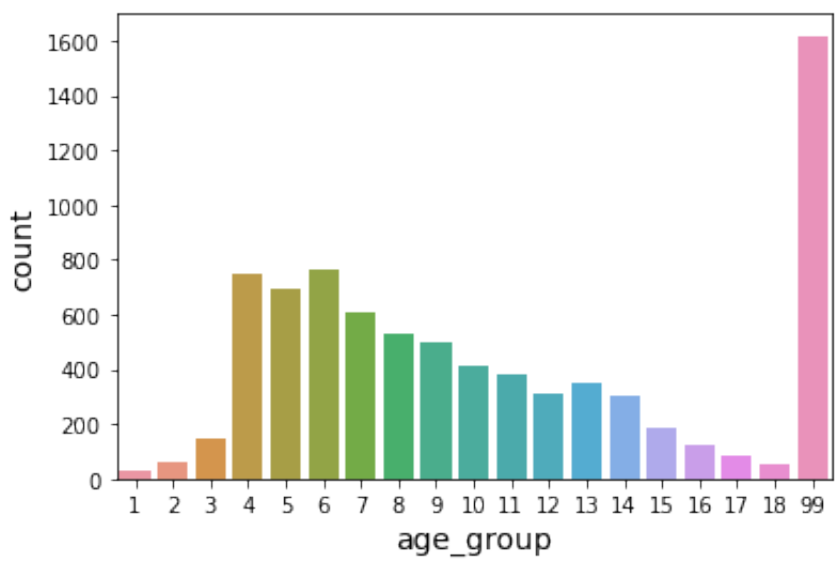
3.0 - Hard/קשה

4.0 - No damage/אין נזק

Age Group Injured during Accident

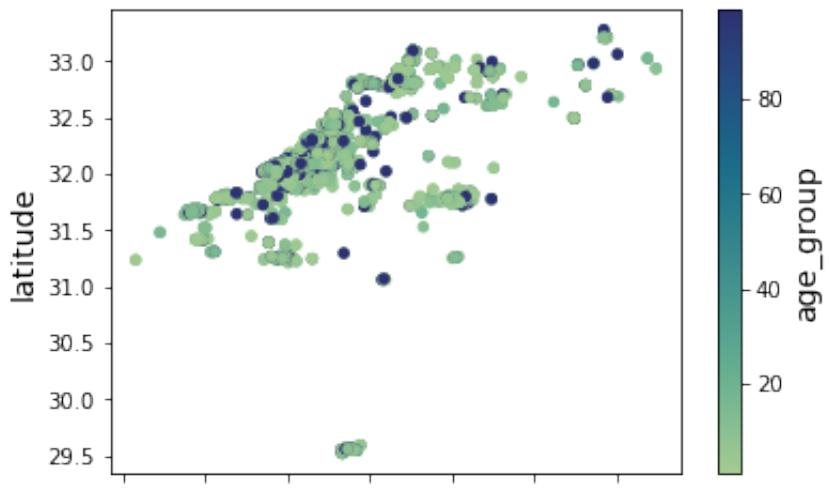
In [109]:
```
sns.countplot(df['age_group'])
```

Out[109]:
```
<AxesSubplot:xlabel='age_group', ylabel='count'>
```

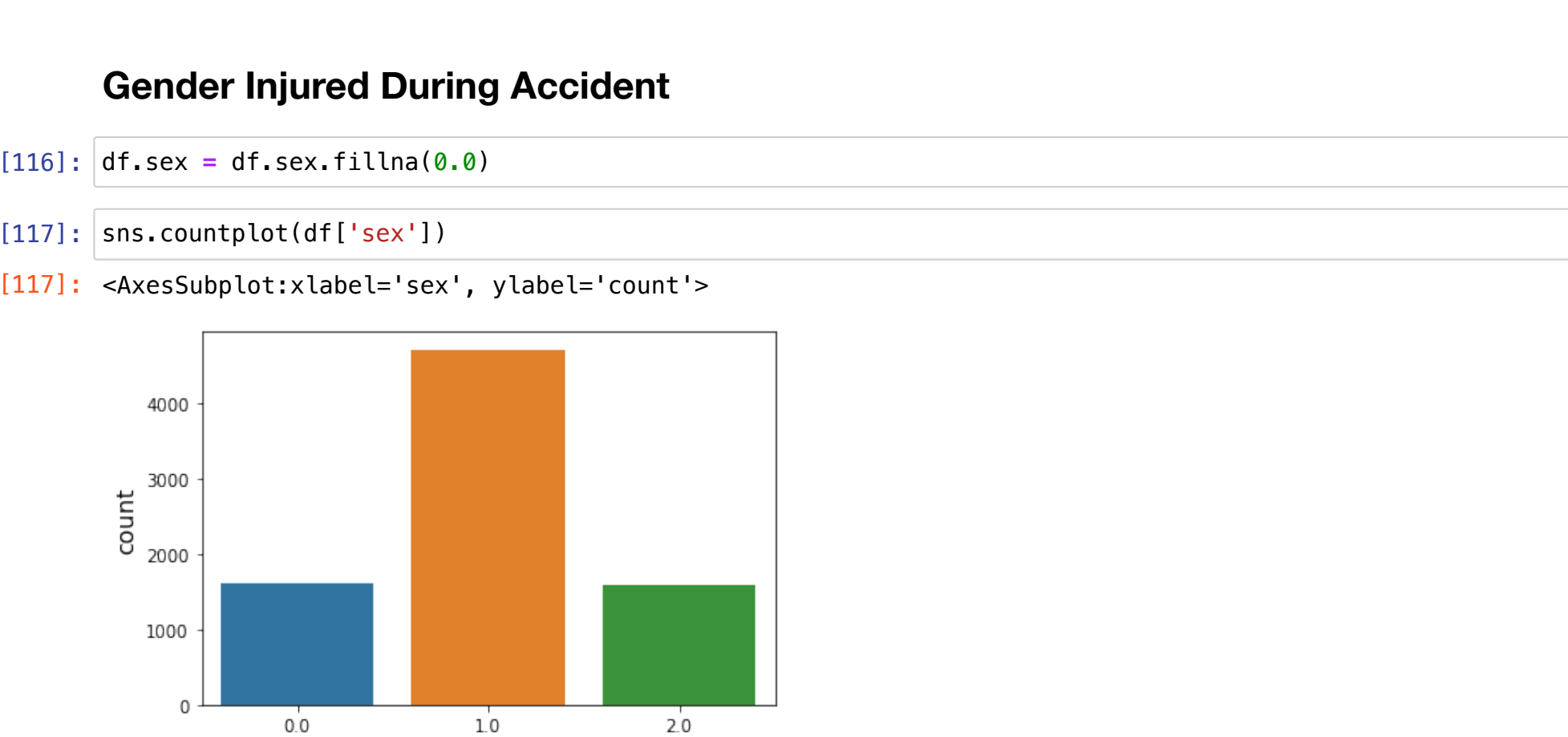


In [110]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="age_group",
        colormap="crest")
```

Out[110]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [5]:
```
Title = "Percentage of Accident happened by Age"
PlotPiechart(df['age_group'].values, df, 'Age')
```



Gender Injured During Accident

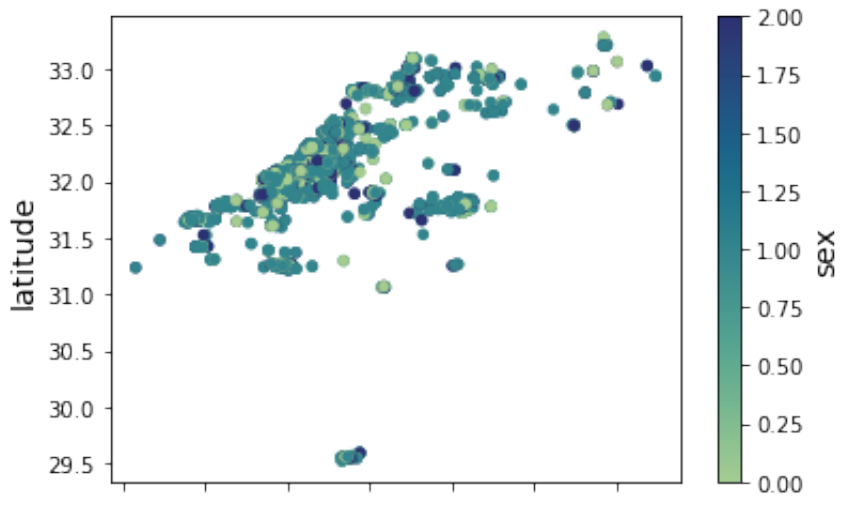
In [116]:
```
df.sex = df.sex.fillna(0.0)
```

In [117]:
```
sns.countplot(df['sex'])
```

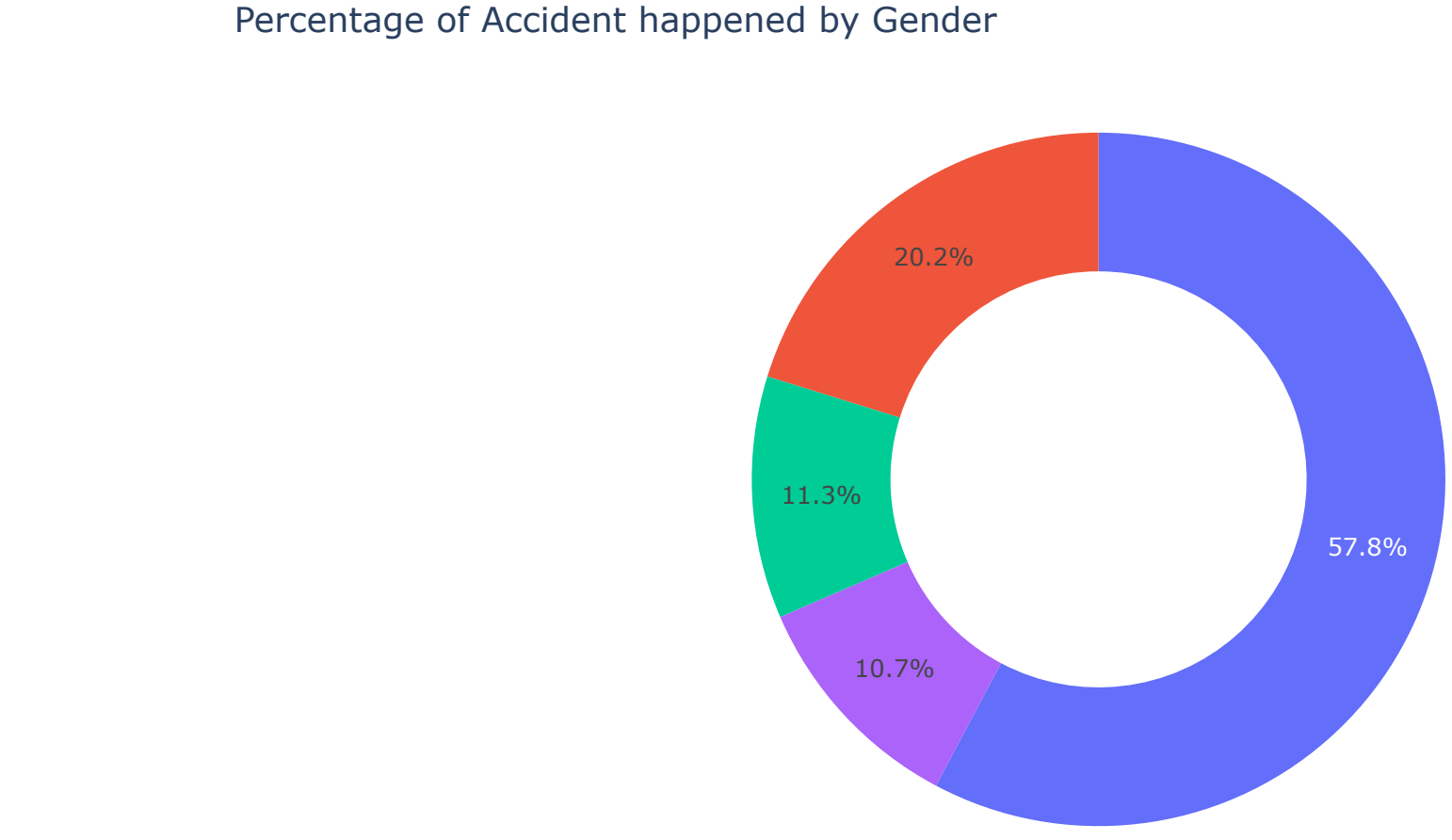
Out[117]: <AxesSubplot:xlabel='sex', ylabel='count'>

In [118]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="sex",
        colormap="crest")
```

Out[118]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [6]:
```
Title = "Percentage of Accident happened by Gender"
PlotPiechart(df['sex'].values, df, 'Gender')
```



0.0 - Not Disclosed

1.0 - Male / זכר

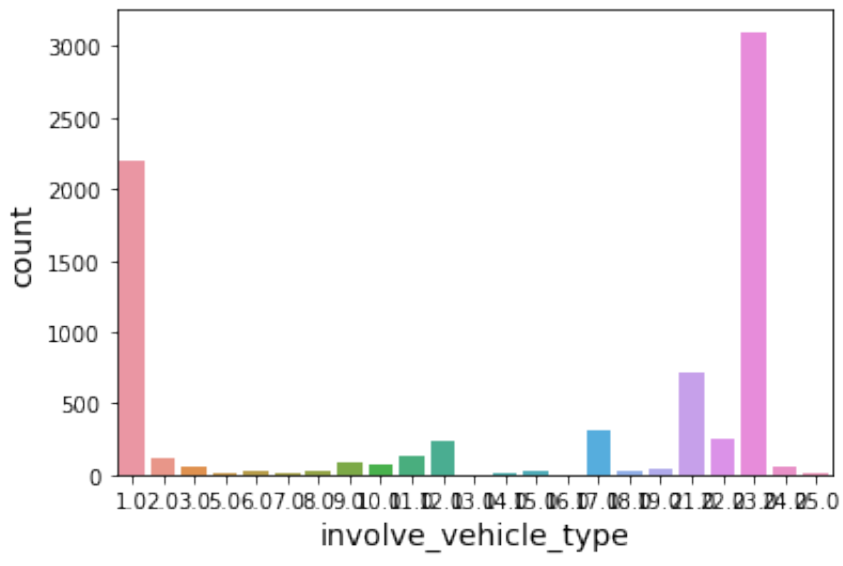
2.0 - Female / נקבה

Male were more involved in Accidents !

Involved Vehicle Type during Accident

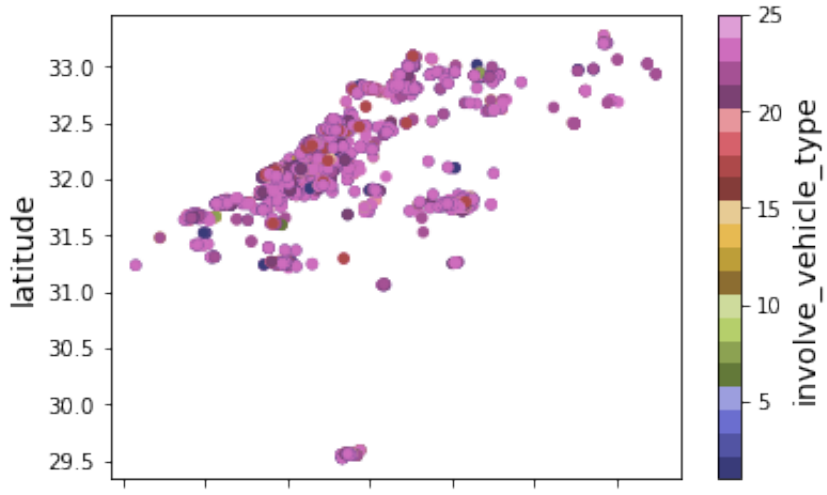
In [120]:
```
sns.countplot(df['involve_vehicle_type'])
```

Out[120]: <AxesSubplot:xlabel='involve_vehicle_type', ylabel='count'>

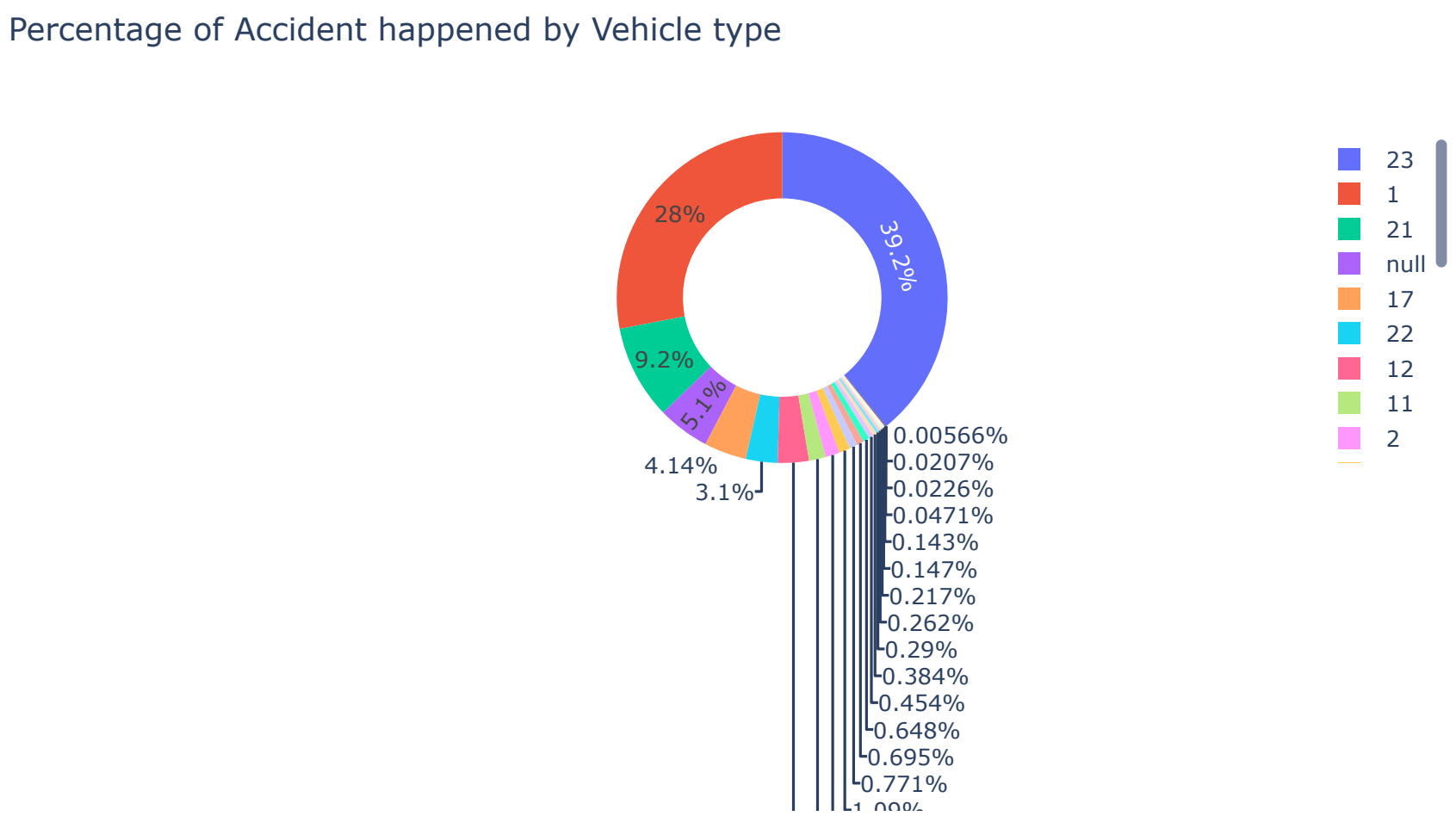


In [121]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="involve_vehicle_type",
        colormap="tab20b")
```

Out[121]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [7]:
```
Title = "Percentage of Accident happened by Vehicle type"
PlotPiechart(df['involve_vehicle_type'].values, df, 'Vehicle type')
```



Vehicle type

23 - Electric Bicycle / אופניים חשמליים

1 - Private passenger car / רכב נוסעים פרטי

21 and 22 - Electric scooter / קורקינט חשמלי

17 and Null - Unknown/ אחר ולא ידוע

12 - Cab / מונית

19 - 401+ cc motorcycle / אופנוע 401+ סמ"ק

10 - 126 to 400 cc motorcycle / אופנוע 126 עד 400 סמ"ק

3 - Freight up to 3.5 tons - non-unified (pickup) / (טנדר לא אחוד - משא עד 3.5 טון)

11 - Bus / אוטובוס

9 - Motorcycle 51 to 125 cc / אופנוע 51 עד 125 סמ"ק

Maximum Number of accidents happend on electric bicycle . Whereas ridesharing platforms cabs and buses have less accidents compared to private vehicles !

Mobility as a Service (MaaS) combines multiple different transportation modes and mobility services, such as public transport, car sharing, taxis / ride hailing and bike sharing to a single integrated platform.

Digitalized mobility services contributions to reducing vehicle kilometres and greenhouse gas emissions

1. Car Sharing :

A form of car rental model is car sharing. Car sharing, unlike regular car rental, allows users to rent cars for short periods of time, generally by the hour. Cars are also available around town and can be picked up outside of business hours.

1.1 Reduction of vehicle kilometers travelled (VKT) from car sharing

The reductive impacts of car sharing on VKT, when compared to the use of a private car, vary considerably between studies, ranging from 18 % to 67 % VKT reduction!

1.2 Reduction of greenhouse gas and other emissions from car sharing

The GHG emissions of a typical petrol car over its lifecycle can be divided into three main sources: production, maintenance, and disposal of the car

1.3 Emission reductions from car sharing from less production, maintenance and disposal of cars

1.4 Emission reductions from car-sharing from less driving and more energy- efficient cars

2. Ride Sharing :

The sharing of a vehicle by passengers travelling to and/or from the same location is known as ride sharing. Trips where the driver undertakes a separate trip to take a passenger somewhere specific are not included in ride sharing (for example taxi, Uber). A web/app platform can help with ride matching (matching a driver with empty car seats and a passenger asking for a ride).

2.1 Reduction of vehicle kilometers travelled from ride sharing

Ride sharing has the potential to reduce vehicle kilometers travelled, as several persons share one vehicle.

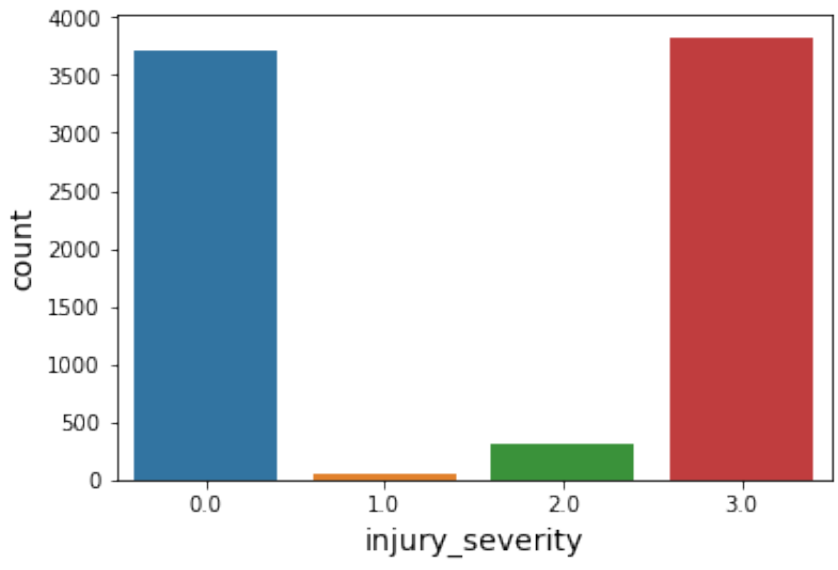
2.2 GHG reductions from ride sharing

Injury Severity during Accidents

In [9]:
```
df.injury_severity = df.injury_severity.fillna(0.0)
```

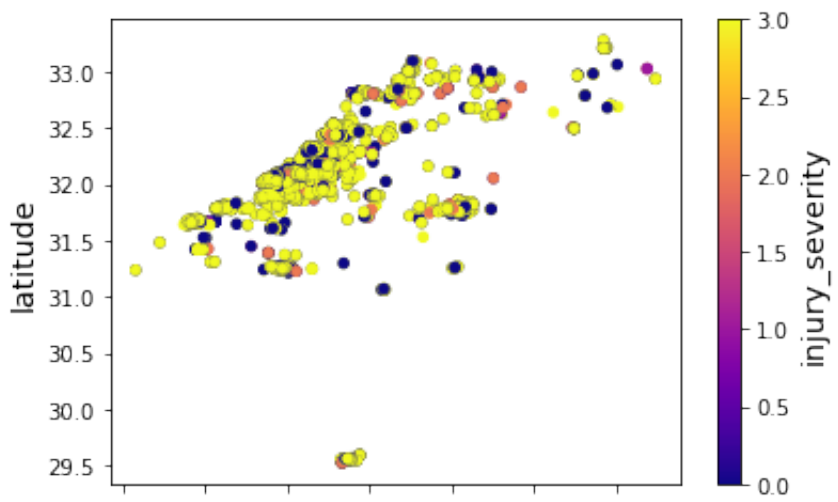
In [130]:
```
sns.countplot(df['injury_severity'])
```

Out[130]: <AxesSubplot:xlabel='injury_severity', ylabel='count'>

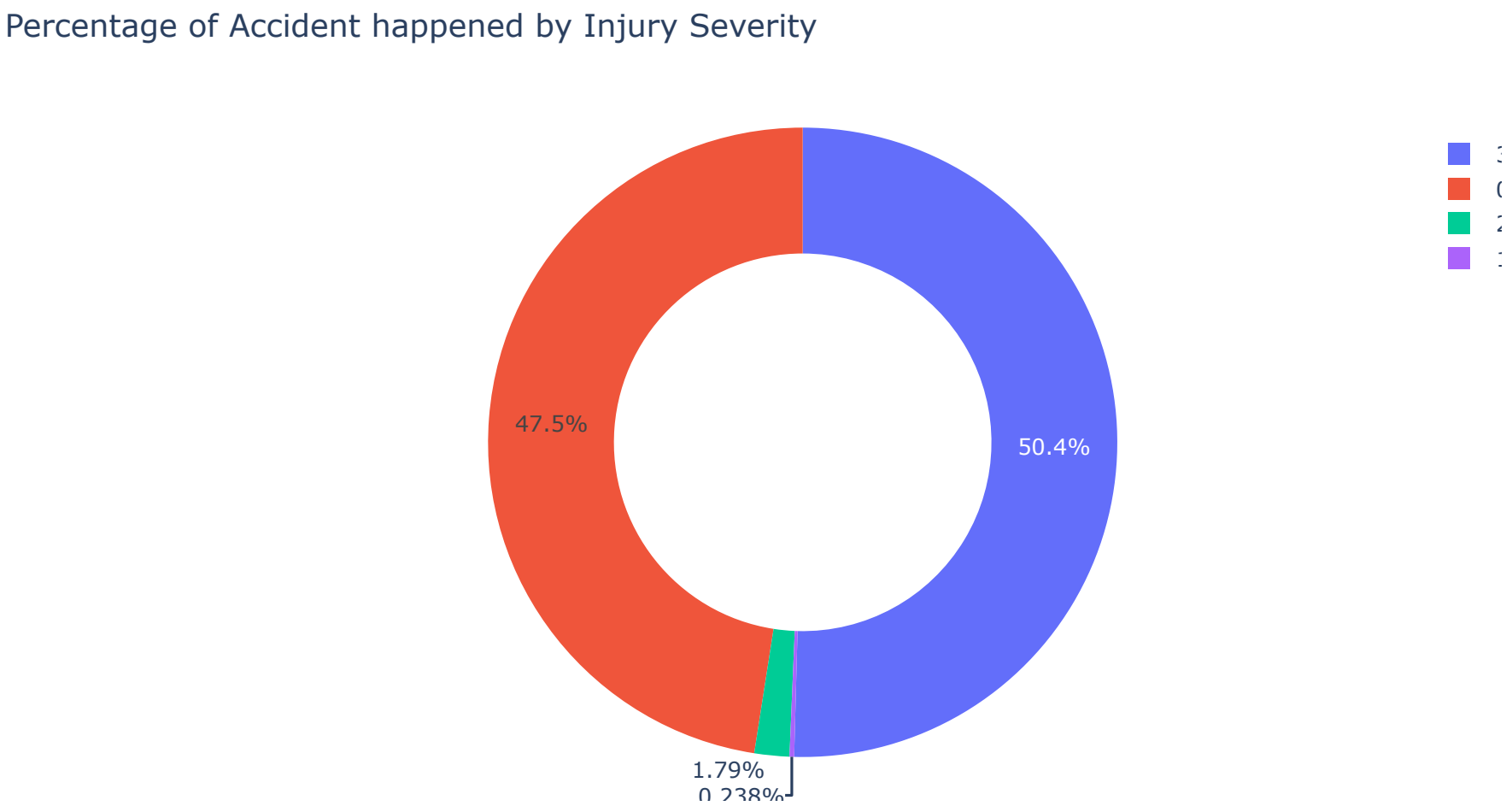


In [133]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="injury_severity",
        colormap="plasma")
```

Out[133]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [10]:
```
Title = "Percentage of Accident happened by Injury Severity"
PlotPiechart(df['injury_severity'].values, df, 'Injury Severity')
```



Injury Severity

0.0 - Status Unknown

1.0 - dead / הרוג

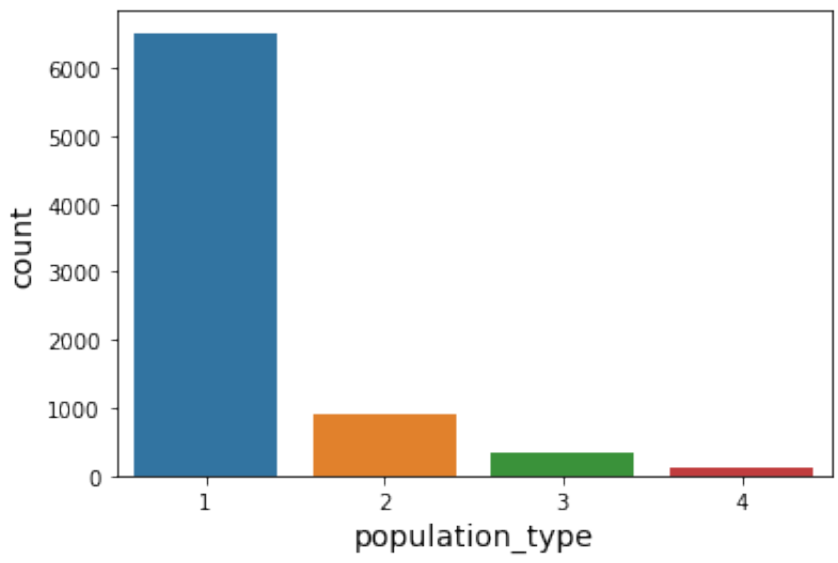
2.0 - Severly injured / פצוע קשה

3.0 - Slightly injured / פצוע קל

Accident happened by Population type

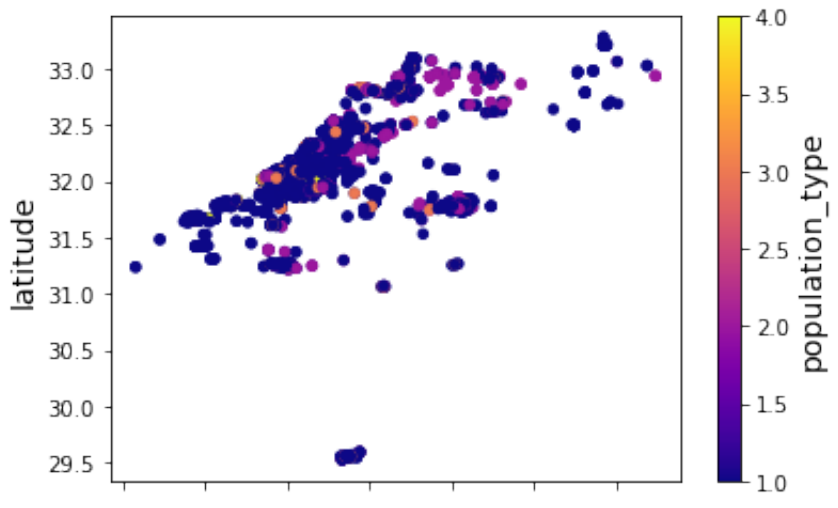
In [135]:
```
sns.countplot(df['population_type'])
```

Out[135]: <AxesSubplot:xlabel='population_type', ylabel='count'>

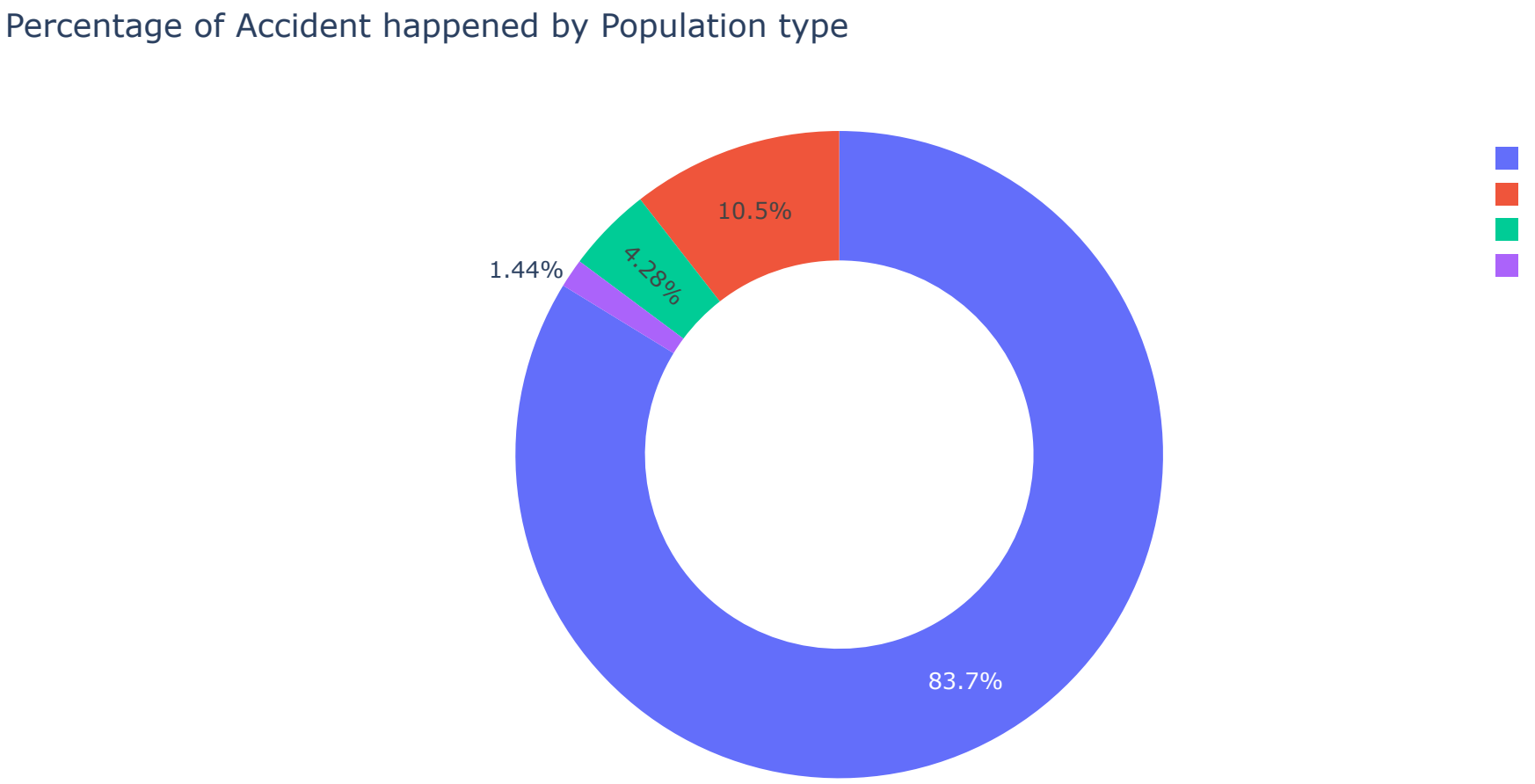


In [136]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="population_type",
        colormap="plasma")
```

Out[136]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [11]:
```
Title = "Percentage of Accident happened by Population type"
PlotPiechart(df['population_type'].values, df, 'Population type')
```



Population Type

1 - Jews / יהודים

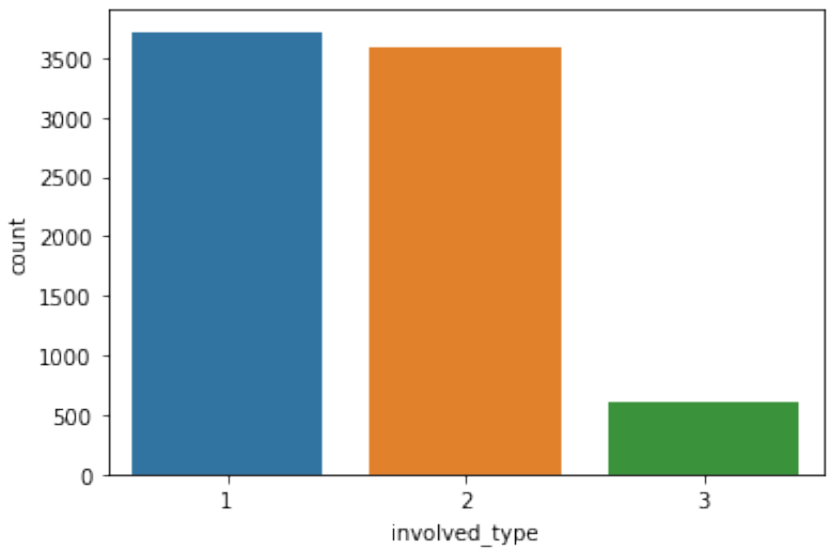
2 - Arabs / ערבים

3 & 4 - Others & Strangers/ זרים אחרים

Involved Type During Accident

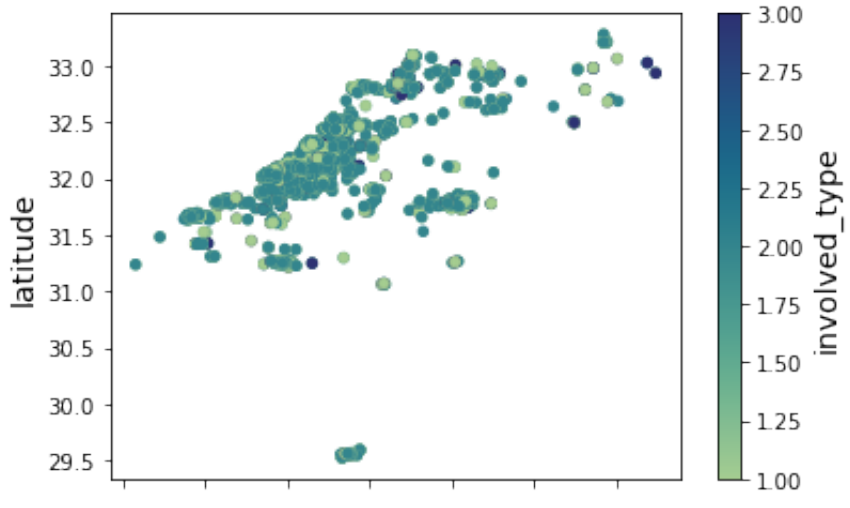
In [12]:
```
sns.countplot(df['involved_type'])
```

Out[12]:
```
<AxesSubplot:xlabel='involved_type', ylabel='count'>
```

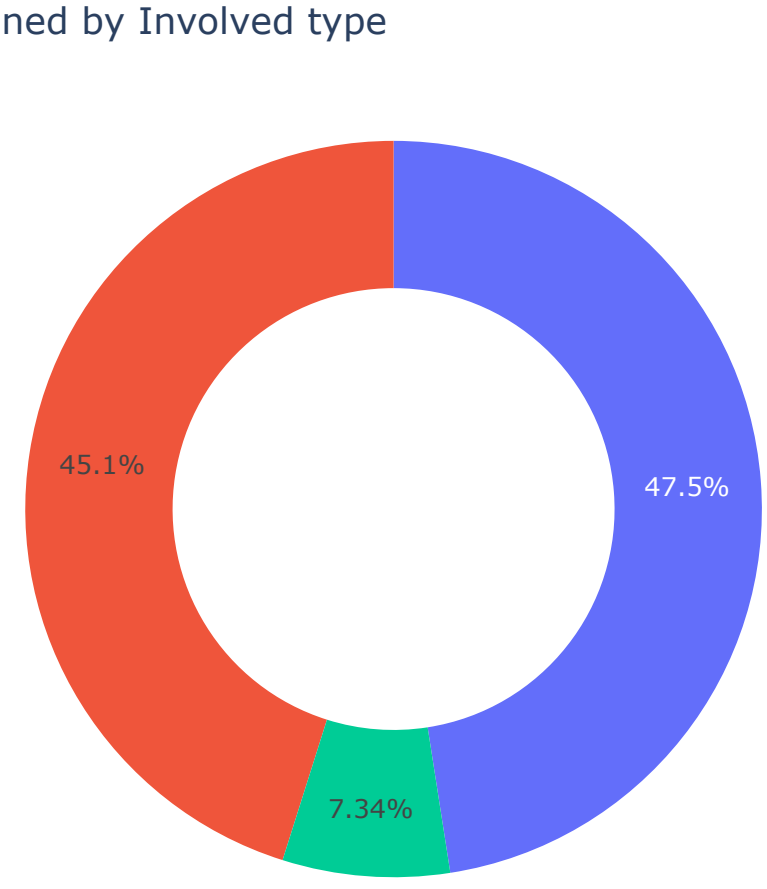


In [89]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="involved_type",
        colormap="crest")
```

Out[89]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [12]:
```
Title = "Percentage of Accident happened by Involved type"
PlotPiechart(df['involved_type'].values, df, 'Involved type')
```



Involved type during accident

1 - Driver/נהג

2 - A driver was injured/נהג נפגע

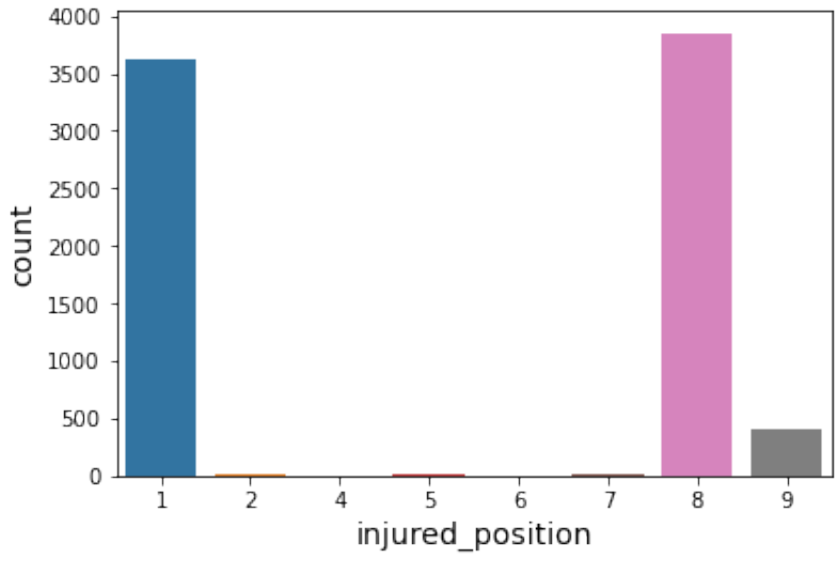
3 - injured/נפגע

Maximum times the driver was injured !

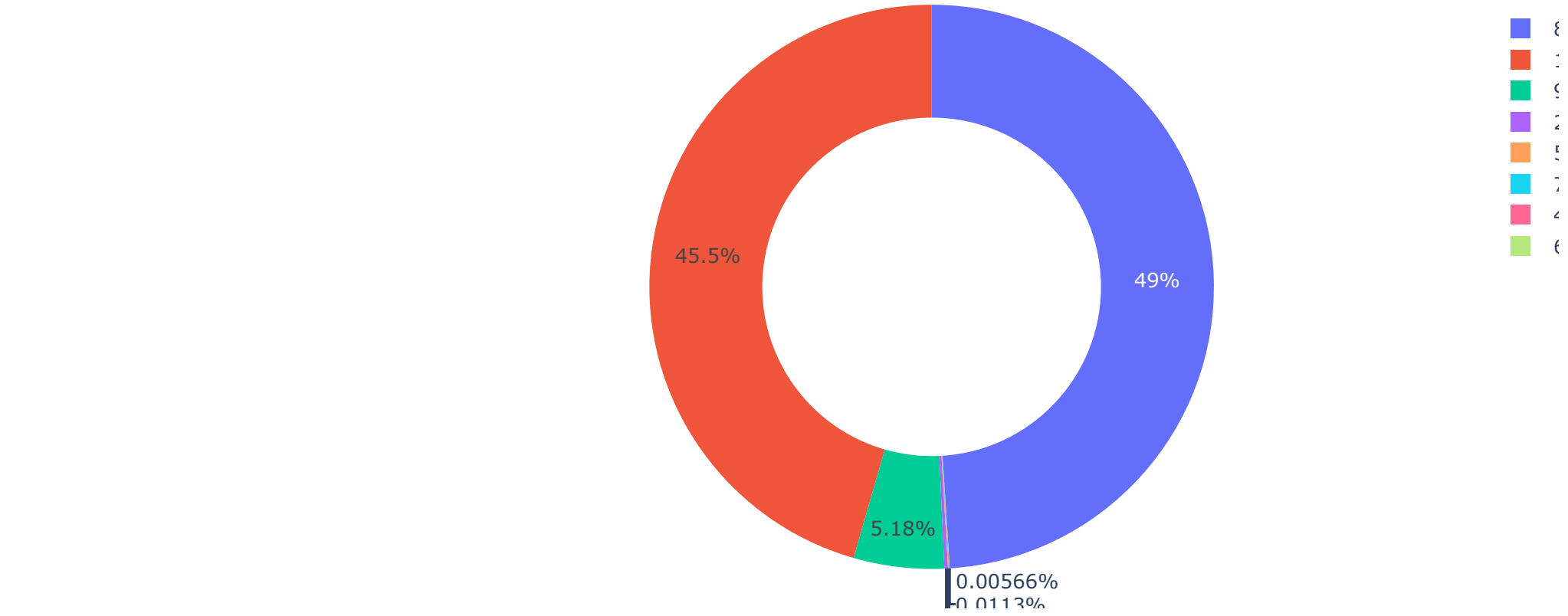
Injured Position During Accident

In [7]:
```
sns.countplot(df['injured_position'])
```

Out[7]: <AxesSubplot:xlabel='injured_position', ylabel='count'>



In [13]:
```
Title = "Injured position during Accident"
PlotPiechart(df['injured_position'].values, df, 'Injured Position')
```



8 - Unknown / לא ידוע

1 - Sitting in the car in the front seat /ישב ברכב במושב קדמי

9 - Pedestrian / הולך רגל

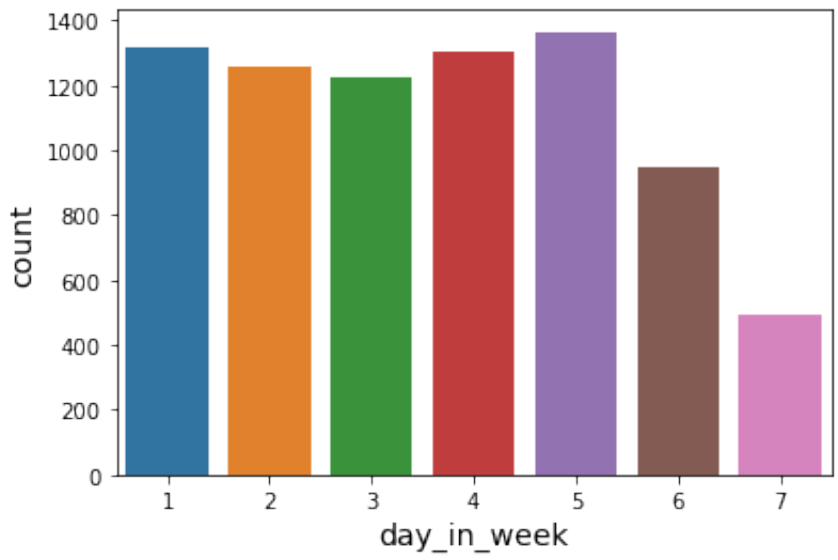
2 - Sat in the car in the back seat/ ישב ברכב במושב אחורי

People in car's front are more vulnerable to accidents followed by pedestrians ! We can see here large number of position involved in accidents is unkown.

Accident happened by Day in week

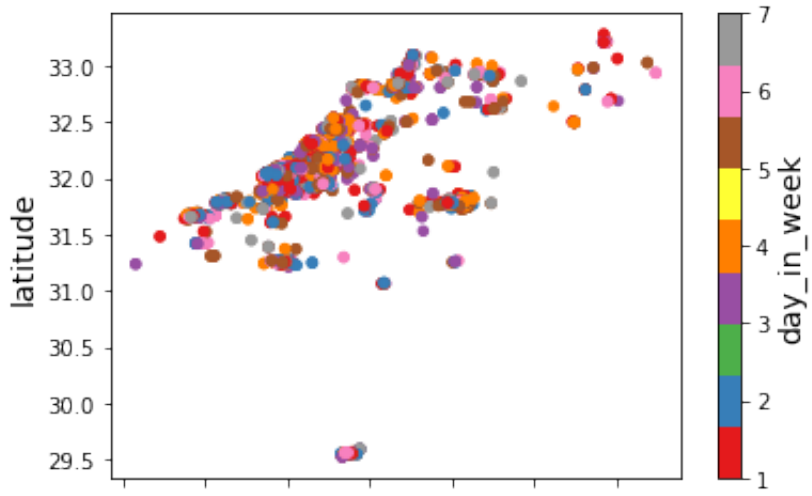
In [139]:
```
sns.countplot(df['day_in_week'])
```

Out[139]: <AxesSubplot:xlabel='day_in_week', ylabel='count'>

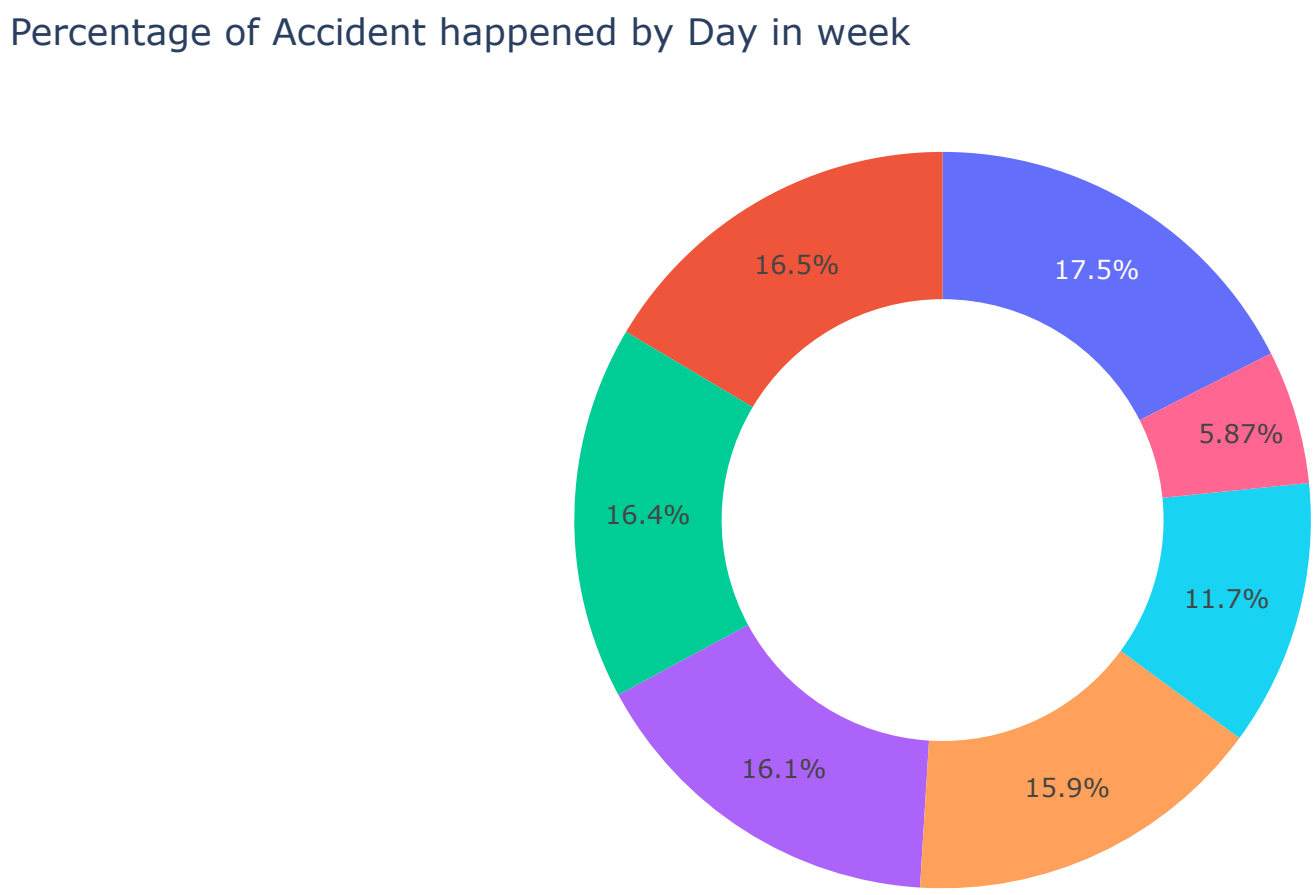


In [141]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="day_in_week",
        colormap="Set1")
```

Out[141]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [14]:
```
Title = "Percentage of Accident happened by Day in week"
PlotPiechart(df['day_in_week'].values, df, 'Day in week')
```



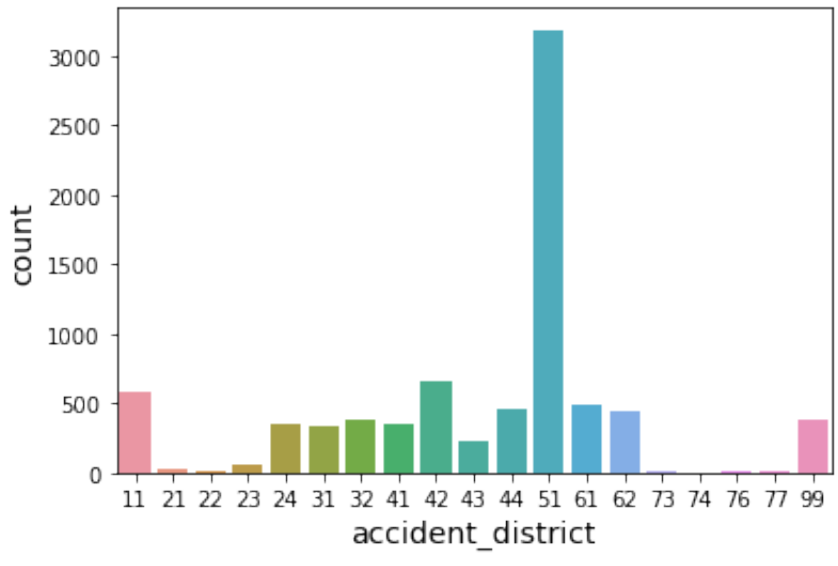
During Weekend there is a drop in Accidents.

Most accidents happens during Fridays.There is less variance between accidents on Weekdays!

Accident happened by District

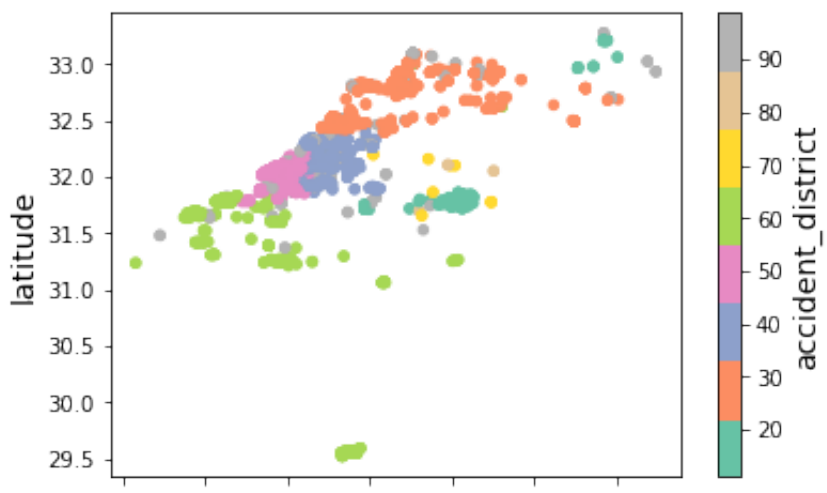
In [144]:
```
sns.countplot(df['accident_district'])
```

Out[144]: <AxesSubplot:xlabel='accident_district', ylabel='count'>

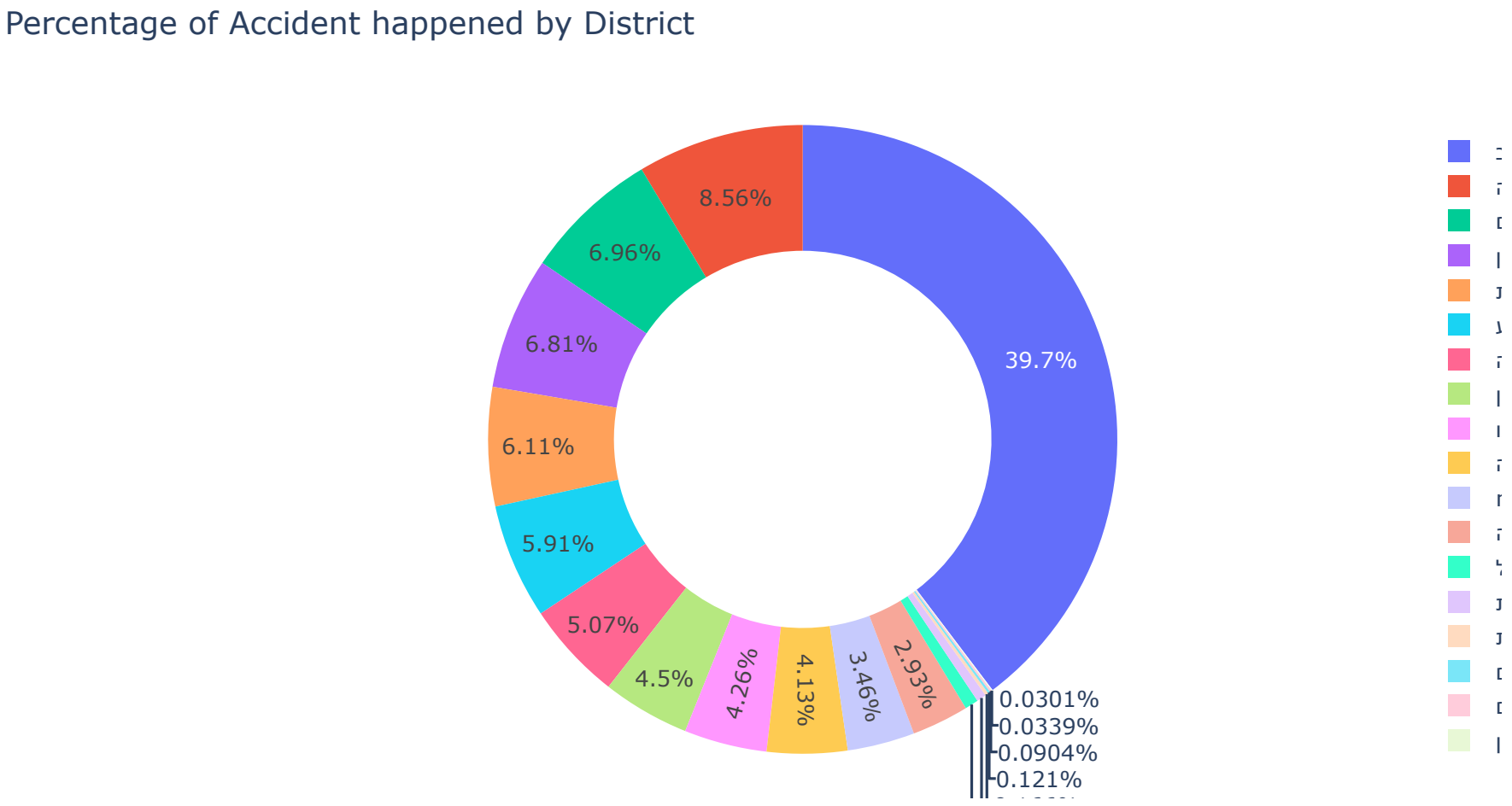


In [146]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="accident_district",
        colormap="Set2")
```

Out[146]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [15]:
```
Title = "Percentage of Accident happened by District"
PlotPiechart(df['accident_district_hebrew'].values, df, 'District')
```



District

51 - Tel Aviv / תל אביב

31 - Haifa /חיפה

24 - Acre / עכו

11 - Jerusalem/ ירושלים

42 - Petah Tiqwa / פתח תקווה

43 - Ramla / רמלה

62 - Beer Sheva / באר שבע

41 - Hasharon/ השרון

61 - Ashkelon / אשקלון

32 - Hadera / חדרה

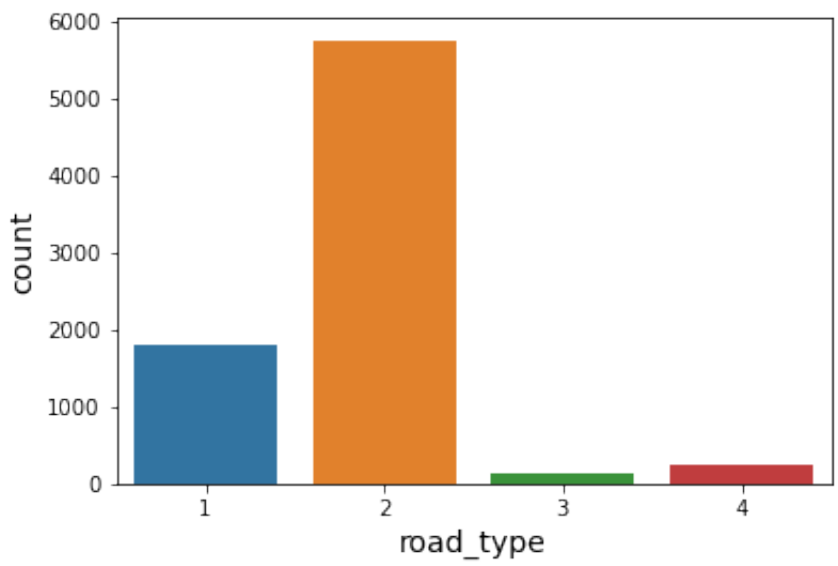
44 - streets / רחובות

Maximum Accidents occur in Cities

Accident happened by Road type

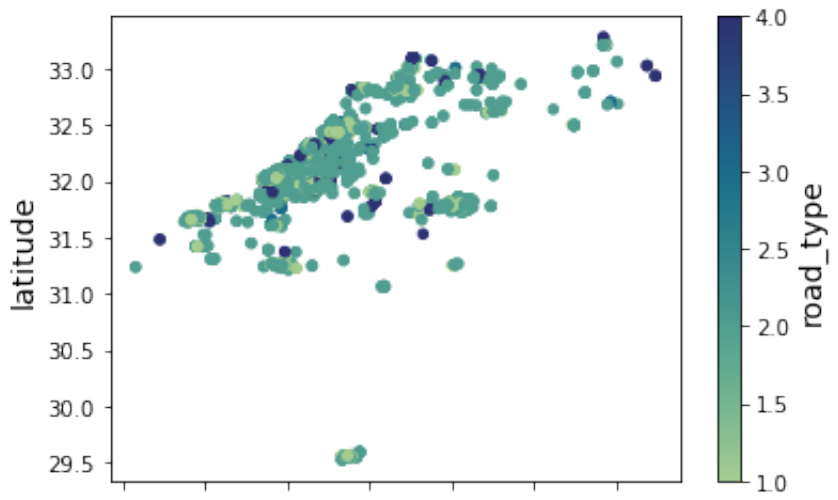
In [150]:
```
sns.countplot(df['road_type'])
```

Out[150]: <AxesSubplot:xlabel='road_type', ylabel='count'>

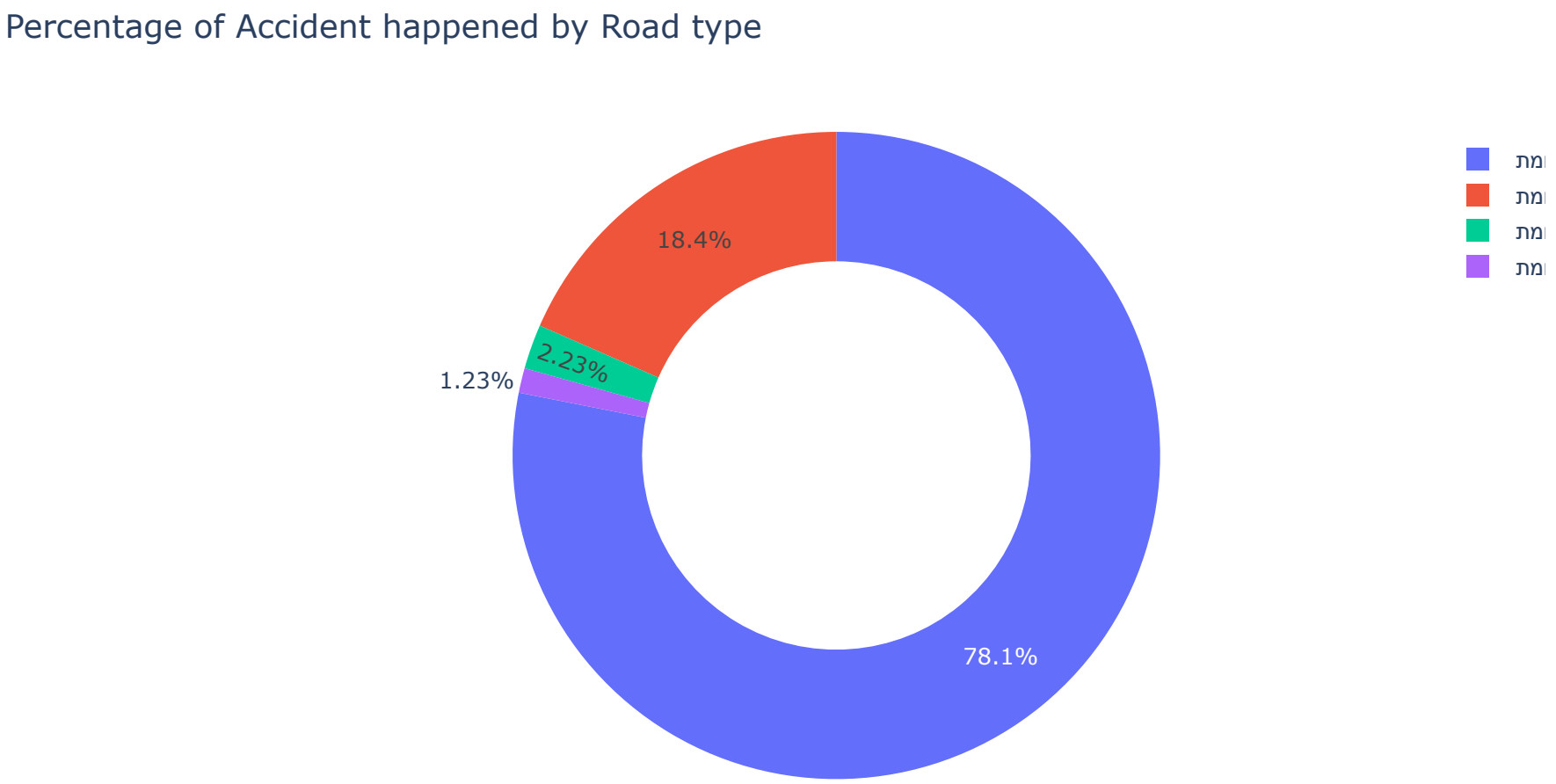


In [151]:
```
df.plot(x="longitude", y="latitude",kind="scatter", c="road_type",
        colormap="crest")
```

Out[151]: <AxesSubplot:xlabel='longitude', ylabel='latitude'>



In [16]:
```
Title = "Percentage of Accident happened by Road type"
PlotPiechart(df['road_type_hebrew'].values, df, 'Road type')
```



Road Type :

1 - Urban at the junction / עירונית בצומת

2 - Urban not at the intersection / עירונית לא בצומת

3 - Non-urban at the intersection/ לא-עירונית בצומת

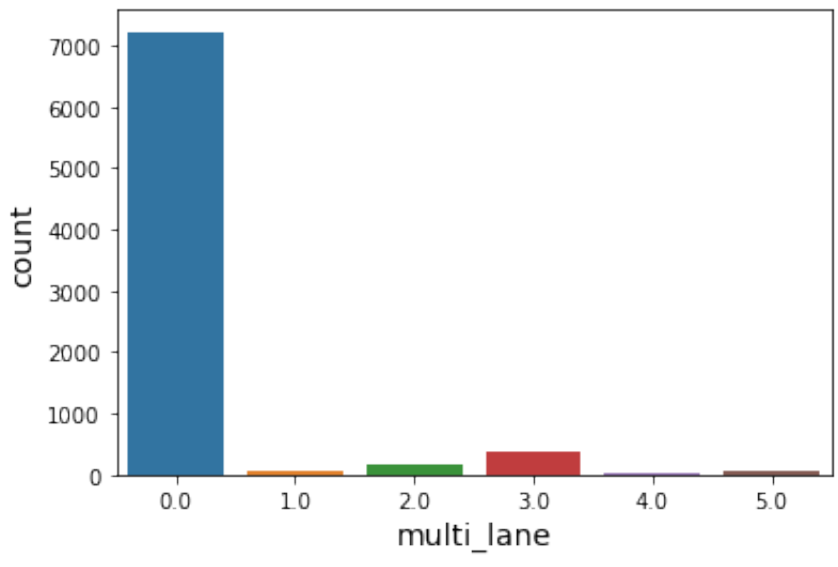
4 - Non-urban not at the junction / לא-עירונית לא בצומת

For Junctions : Signs Alerting drivers of the upcoming junctions, traffic lights, or stop signs would help in some of these areas where they are feasible.

Multi - Lane

In [48]:
```
df.multi_lane = df.multi_lane.fillna(0)
sns.countplot(df['multi_lane'])
```

Out[48]: <AxesSubplot:xlabel='multi_lane', ylabel='count'>



3 - Built-in separator without safety fence/ מיפרדה בנויה ללא גדר בטיחות

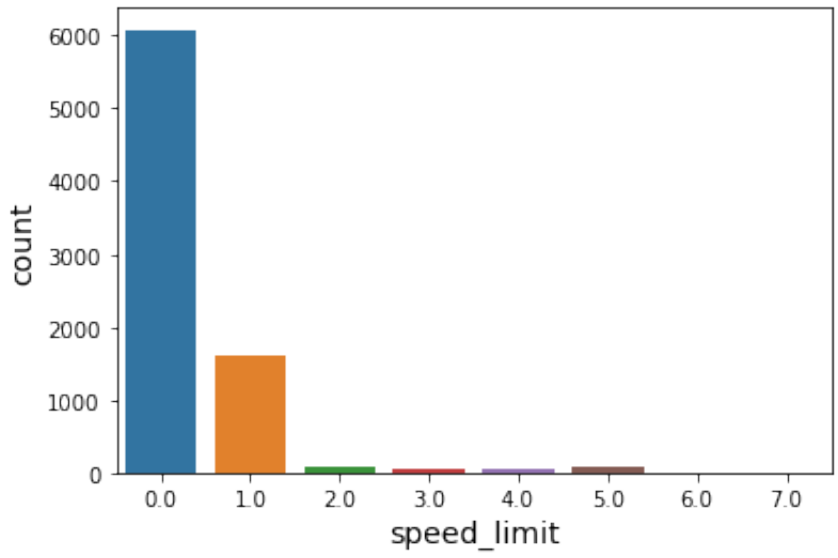
2 -Separator with safety fence / מיפרדה עם גדר בטיחות

0 - Large number of data is Unknown

Speed Limit

In [12]:
```
df.speed_limit = df.speed_limit.fillna(0)
sns.countplot(df['speed_limit'])
```

Out[12]: <AxesSubplot:xlabel='speed_limit', ylabel='count'>



1 - Upto 50 km hr

2 - 60 km hr

Large Number of data is unkown

We can adopt to Intelligent Transportation Systems (ITS) ¶

ITS is a combination of leading-edge information and communication technologies due to the benefits that wireless devices, combined with sensing technologies and ICT smart services.

These technologies optimize ground and maritime transportation by using satellites to share information and master designing a communication environment among infrastructure, vehicles, and passenger's portable gadgets.

It used in traffic management systems to improve the safety, efficiency, and sustainability of transportation networks, to reduce traffic congestion and to enhance drivers’ experiences.