Multiple Linear Regression and Geospatial

In [1]:
```
# Check the dataset directory
%pwd
```

Out[1]:
```
'C:\\Users\\SK\\Desktop\\Python\\Python\\Python Project\\5. Multiple Linear R
egression and Geospatial'
```

In [2]:
```
# Change the working directory
import os
os.chdir("/Users/SK/Desktop/SK/NUS EBA/Semester 2/Statistical BootCamp/WK4")
```

In [3]:
```
# Import the functions
import pandas as pd
import numpy as np
from pandas import DataFrame, read_csv
```

In [4]:
```
# Read the csv file
housing = pd.read_csv("housing.csv")
```

In [5]:
```
housing.head()
```

Out[5]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0

In [6]:
```
housing.describe()
```

Out[6]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	popul
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.0C
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.47
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.46
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.0C
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.0C
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.0C
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.0C
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.0C

In [7]:
```
housing.shape
```

Out[7]:
```
(20640, 10)
```

In [8]:
```
# Check the missing value
housing.isnull().sum()
```

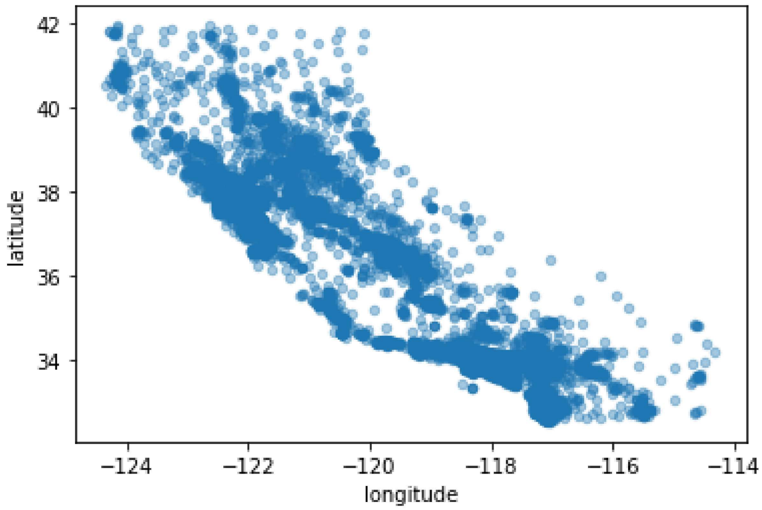
Out[8]:
```
longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
```

In [13]:
```
# Remove the missing value from the dataset
house_no_missing = housing.dropna()
```

In [14]:
```
house_no_missing.isnull().sum()
```

Out[14]:
```
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
median_house_value    0
ocean_proximity       0
dtype: int64
```

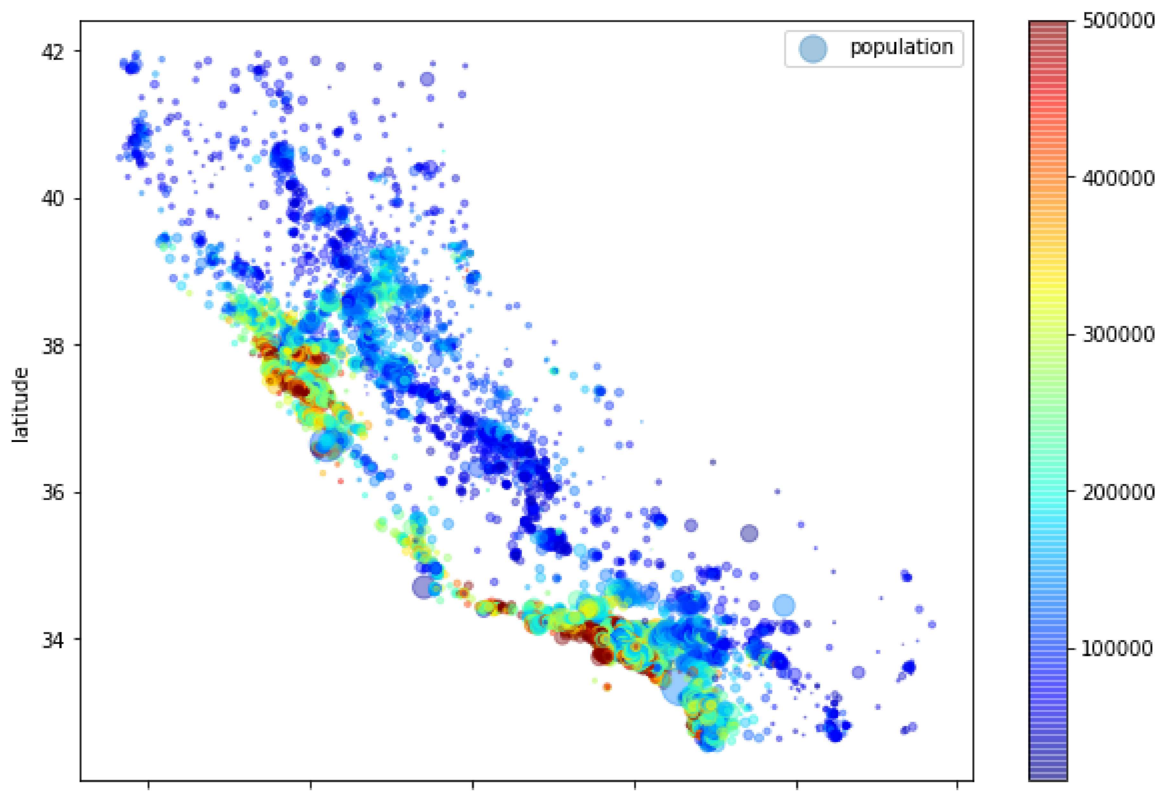
In [16]:
```
import matplotlib.pyplot as plt
house_no_missing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.4)
plt.show()
```



In [18]:

```
house_no_missing.plot(kind="scatter", x="longitude", y="latitude",
    s=house_no_missing['population']/100, label="population",
    c="median_house_value", cmap=plt.get_cmap("jet"),
    colorbar=True, alpha=0.4, figsize=(10,7),
)
plt.legend()
plt.show()
```

```
C:\Users\SK\Anaconda3\lib\site-packages\pandas\plotting\_tools.py:307: Matplo
tlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
C:\Users\SK\Anaconda3\lib\site-packages\pandas\plotting\_tools.py:307: Matplo
tlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
C:\Users\SK\Anaconda3\lib\site-packages\pandas\plotting\_tools.py:313: Matplo
tlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
C:\Users\SK\Anaconda3\lib\site-packages\pandas\plotting\_tools.py:313: Matplo
tlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
```



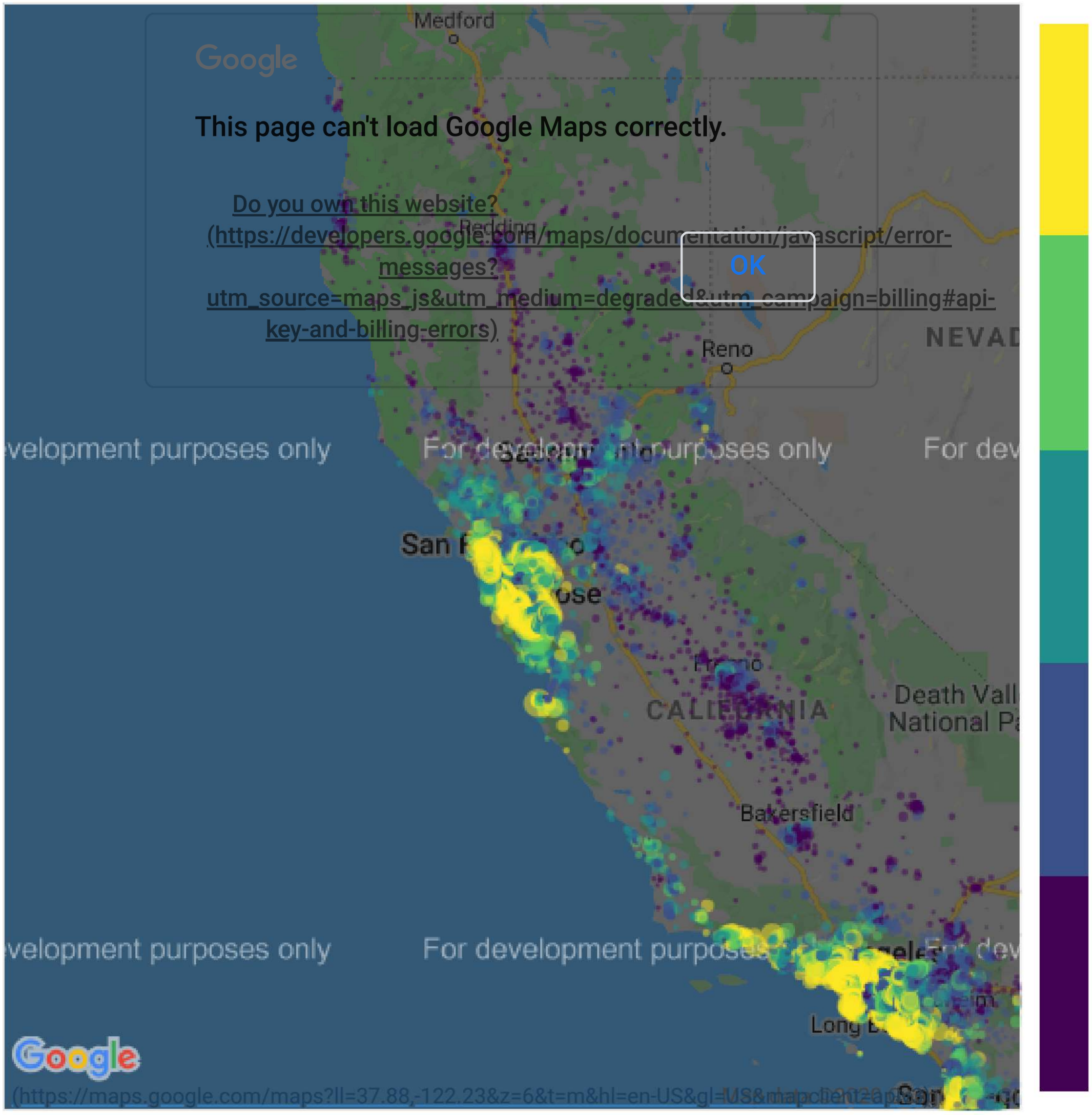
In [19]:
```
from bokeh.io import output_file, output_notebook, show
from bokeh.models import (
    GMapPlot, GMapOptions, ColumnDataSource, Circle, LogColorMapper, BasicTick
er,
ColorBar, Range1d, PanTool, WheelZoomTool, BoxSelectTool)
from bokeh.models.mappers import ColorMapper, LinearColorMapper
from bokeh.palettes import Viridis5
```

In [20]:
```
map_options = GMapOptions(lat=37.88, lng=-122.23, map_type="roadmap", zoom=6)

plot = GMapPlot(
    x_range=Range1d(), y_range=Range1d(),
    map_options=map_options
)
plot.title.text = "Hey look! It's a scatter plot on a map!"
```

In [21]:
```
# For GMaps to function, Google requires you obtain and enable an API key:
#
#     https://developers.google.com/maps/documentation/javascript/get-api-key
#
# Replace the value below with your personal API key:
plot.api_key = "AIzaSyBYrbp34OohAHsX1cub8ZeHlMEFajv15fY"

source = ColumnDataSource(
    data=dict(
        lat=house_no_missing.latitude.tolist(),
        lon=house_no_missing.longitude.tolist(),
        size=house_no_missing.median_income.tolist(),
        color=house_no_missing.median_house_value.tolist()
    )
)
max_median_house_value = house_no_missing.loc[house_no_missing['median_house_v
alue'].idxmax()]['median_house_value']
min_median_house_value = house_no_missing.loc[house_no_missing['median_house_v
alue'].idxmin()]['median_house_value']

#color_mapper = CategoricalColorMapper(factors=['hi', 'lo'], palette=[RdBu3
[2], RdBu3[0]])
#color_mapper = LogColorMapper(palette="Viridis5", low=min_median_house_value,
high=max_median_house_value)
color_mapper = LinearColorMapper(palette=Viridis5)

circle = Circle(x="lon", y="lat", size="size", fill_color={'field': 'color',
'transform': color_mapper}, fill_alpha=0.5, line_color=None)
plot.add_glyph(source, circle)

color_bar = ColorBar(color_mapper=color_mapper, ticker=BasicTicker(),
                     label_standoff=12, border_line_color=None, location=(0,0
))
plot.add_layout(color_bar, 'right')

plot.add_tools(PanTool(), WheelZoomTool(), BoxSelectTool())
#output_file("gmap_plot.html")
output_notebook()

show(plot)
```

(https://bokeh.pydata.org) BokehJS 1.2.0 successfully loaded.

Hey look! It's a scatter plot on a map!



(http

In [22]:
```python
## Check the Correlation between variable
house_no_missing.corr()
```

Out[22]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	po
longitude	1.000000	-0.924616	-0.109357	0.045480	0.069608	
latitude	-0.924616	1.000000	0.011899	-0.036667	-0.066983	-
housing_median_age	-0.109357	0.011899	1.000000	-0.360628	-0.320451	-
total_rooms	0.045480	-0.036667	-0.360628	1.000000	0.930380	
total_bedrooms	0.069608	-0.066983	-0.320451	0.930380	1.000000	
population	0.100270	-0.108997	-0.295787	0.857281	0.877747	
households	0.056513	-0.071774	-0.302768	0.918992	0.979728	
median_income	-0.015550	-0.079626	-0.118278	0.197882	-0.007723	
median_house_value	-0.045398	-0.144638	0.106432	0.133294	0.049686	-

In [51]:
```python
## Multiple Linear Regression
x = house_no_missing.drop(['median_house_value', 'ocean_proximity'], axis =1)
y = house_no_missing['median_house_value']
```

In [52]:
```
from sklearn.linear_model import LinearRegression
import statsmodels.formula.api as smf
model = smf.ols('y~x', data = house_no_missing).fit()
model.summary()
```

Out[52]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.637
Model:	OLS	**Adj. R-squared:**	0.637
Method:	Least Squares	**F-statistic:**	4478.
Date:	Wed, 15 Jul 2020	**Prob (F-statistic):**	0.00
Time:	23:46:43	**Log-Likelihood:**	-2.5682e+05
No. Observations:	20433	**AIC:**	5.137e+05
Df Residuals:	20424	**BIC:**	5.137e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3.585e+06	6.29e+04	-57.001	0.000	-3.71e+06	-3.46e+06
x[0]	-4.273e+04	717.087	-59.588	0.000	-4.41e+04	-4.13e+04
x[1]	-4.251e+04	676.952	-62.796	0.000	-4.38e+04	-4.12e+04
x[2]	1157.9003	43.389	26.687	0.000	1072.855	1242.945
x[3]	-8.2497	0.794	-10.387	0.000	-9.807	-6.693
x[4]	113.8207	6.931	16.423	0.000	100.236	127.405
x[5]	-38.3856	1.084	-35.407	0.000	-40.511	-36.261
x[6]	47.7014	7.547	6.321	0.000	32.909	62.493
x[7]	4.03e+04	337.207	119.504	0.000	3.96e+04	4.1e+04

Omnibus:	4898.534	**Durbin-Watson:**	0.975
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	18260.733
Skew:	1.166	**Prob(JB):**	0.00
Kurtosis:	7.002	**Cond. No.**	5.10e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.1e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [55]:
```
## Check the VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor
x['Intercept'] = 1

vif = pd.DataFrame()
vif["variables"] = x.columns
vif['VIF'] = [variance_inflation_factor(x.values, i) for i in range(x.shape[1
])]

print(vif)
```

```
            variables           VIF
0           longitude      8.713740
1            latitude      8.828919
2  housing_median_age      1.260015
3         total_rooms     12.717000
4      total_bedrooms     36.003726
5          population      6.371238
6          households     35.136045
7       median_income      1.731511
8           Intercept  16702.386835
```

In [56]:
```
## Eliminate 'total_bedrooms' variable due to high VIF value
x = house_no_missing.drop(['median_house_value', 'ocean_proximity' ,'total_bed
rooms'], axis =1)
y = house_no_missing['median_house_value']
```

In [57]:
```
model = smf.ols('y~x', data = house_no_missing).fit()
model.summary()
```

Out[57]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.632
Model:	OLS	**Adj. R-squared:**	0.632
Method:	Least Squares	**F-statistic:**	5014.
Date:	Wed, 15 Jul 2020	**Prob (F-statistic):**	0.00
Time:	23:51:44	**Log-Likelihood:**	-2.5695e+05
No. Observations:	20433	**AIC:**	5.139e+05
Df Residuals:	20425	**BIC:**	5.140e+05
Df Model:	7		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-3.497e+06	6.31e+04	-55.434	0.000	-3.62e+06	-3.37e+06
x[0]	-4.197e+04	720.300	-58.273	0.000	-4.34e+04	-4.06e+04
x[1]	-4.222e+04	681.159	-61.983	0.000	-4.36e+04	-4.09e+04
x[2]	1126.4985	43.631	25.819	0.000	1040.979	1212.018
x[3]	-1.7751	0.694	-2.558	0.011	-3.135	-0.415
x[4]	-43.0960	1.052	-40.952	0.000	-45.159	-41.033
x[5]	148.9776	4.378	34.025	0.000	140.396	157.560
x[6]	3.838e+04	318.479	120.518	0.000	3.78e+04	3.9e+04

Omnibus:	5263.565	**Durbin-Watson:**	0.941
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	20453.770
Skew:	1.241	**Prob(JB):**	0.00
Kurtosis:	7.227	**Cond. No.**	5.01e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.01e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [58]:
```
## Check the VIF
x['Intercept'] = 1

vif = pd.DataFrame()
vif["variables"] = x.columns
vif['VIF'] = [variance_inflation_factor(x.values, i) for i in range(x.shape[1
])]

print(vif)
```

```
            variables           VIF
0           longitude      8.677840
1            latitude      8.822928
2  housing_median_age      1.257568
3         total_rooms      9.583814
4          population      5.925295
5          households     11.673860
6       median_income      1.524467
7           Intercept  16579.355378
```

In [59]:
```
## Eliminate 'households' variable due to high VIF value
x = house_no_missing.drop(['median_house_value', 'ocean_proximity' ,'total_bed
rooms', 'households'], axis =1)
y = house_no_missing['median_house_value']
```

In [60]:
```
model = smf.ols('y~x', data = house_no_missing).fit()
model.summary()
```

Out[60]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.611
Model:	OLS	**Adj. R-squared:**	0.611
Method:	Least Squares	**F-statistic:**	5353.
Date:	Wed, 15 Jul 2020	**Prob (F-statistic):**	0.00
Time:	23:53:28	**Log-Likelihood:**	-2.5752e+05
No. Observations:	20433	**AIC:**	5.150e+05
Df Residuals:	20426	**BIC:**	5.151e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-3.968e+06	6.33e+04	-62.723	0.000	-4.09e+06	-3.84e+06
x[0]	-4.774e+04	719.613	-66.345	0.000	-4.92e+04	-4.63e+04
x[1]	-4.777e+04	679.805	-70.271	0.000	-4.91e+04	-4.64e+04
x[2]	1118.2926	44.848	24.935	0.000	1030.386	1206.199
x[3]	15.0384	0.501	30.020	0.000	14.057	16.020
x[4]	-25.4074	0.941	-27.014	0.000	-27.251	-23.564
x[5]	3.431e+04	303.318	113.100	0.000	3.37e+04	3.49e+04

Omnibus:	4604.156	**Durbin-Watson:**	0.816
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	12192.156
Skew:	1.215	**Prob(JB):**	0.00
Kurtosis:	5.902	**Cond. No.**	4.83e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.83e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [61]:
```
## Check the VIF
x['Intercept'] = 1

vif = pd.DataFrame()
vif["variables"] = x.columns
vif['VIF'] = [variance_inflation_factor(x.values, i) for i in range(x.shape[1
])]

print(vif)
```

```
            variables           VIF
0           longitude      8.197084
1            latitude      8.316903
2  housing_median_age      1.257530
3         total_rooms      4.725402
4          population      4.479289
5       median_income      1.308668
6           Intercept  15780.636633
```