Caso de Aplicación: Electromobility Plus

Diplomado en Python y Ciencia de Datos

Curso: Herramientas Básicas de Programación en Python

Docente: Patricia Andrea Moller Acuña

Integrantes:

- Margarita Isabel Barrales Quintero
- Cristofer Edgardo Carvajal García
- Fernando Andrés Gaete Velásquez
- Camilo Roberto Maldonado Valderrama
- Sebastian Andrés Saez Saez

In [1]:
```
# Cargar librerias necesarias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# 2. Cargar los datos desde archivo CSV public_transportation_statistics_by_zip_cod
data = pd.read_csv('public_transportation_statistics_by_zip_code-1.csv')

# Mostrar las primeras filas y verificar el encabezado
print(data.head())
```

```
   zip_code  public_transportation_pct  public_transportation_population
0      1379                        3.3                                13
1      1440                        0.4                                34
2      1505                        0.9                                23
3      1524                        0.5                                20
4      1529                        1.8                                32
```

In [3]:
```
# 3. Verificar el tipo de datos y los valores nulos
print("Información del conjunto de datos:")
print(data.info())
# 3 Descripción estadística del dataset para conocer max y min
print("\nDescripción estadística del dataset:")
print(data.describe())
```

```
Información del conjunto de datos:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33120 entries, 0 to 33119
Data columns (total 3 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   zip_code                        33120 non-null  int64
 1   public_transportation_pct       33120 non-null  float64
 2   public_transportation_population  33120 non-null  int64
dtypes: float64(1), int64(2)
memory usage: 776.4 KB
None

Descripción estadística del dataset:
           zip_code  public_transportation_pct  \
count  33120.000000               3.312000e+04
mean   49666.334209              -1.115137e+07
std    27564.925769               8.549920e+07
min      601.000000              -6.666667e+08
25%    26634.750000               0.000000e+00
50%    49739.000000               0.000000e+00
75%    72123.500000               1.100000e+00
max    99929.000000               1.000000e+02

       public_transportation_population
count                      33120.000000
mean                         230.352748
std                         1313.382153
min                            0.000000
25%                            0.000000
50%                            0.000000
75%                           41.000000
max                        35139.000000
```

In [4]:

```python
# 3. Eliminar valores negativos y nulos en columnas importantes
data = data[data['public_transportation_pct'] >= 0]
data.dropna(subset=['public_transportation_pct', 'public_transportation_population'

# Verificar el tipo de datos y los valores nulos después de la limpieza
print("\nInformación del conjunto de datos después de la limpieza:")
print(data.info())

# Descripción estadística después de la limpieza
description = data.describe()

# Formatear la tabla con pandas Styler
styled_description = description.style\
    .set_caption("Descripción Estadística del Dataset después de la Limpieza")\
    .format("{:.2f}")\
    .set_properties(**{'text-align': 'right'})\
    .set_table_styles([dict(selector="th", props=[('text-align', 'right')])])\
    .background_gradient(cmap='viridis', axis=0)

# Mostrar la tabla formateada
display(styled_description)
```

```
Información del conjunto de datos después de la limpieza:
<class 'pandas.core.frame.DataFrame'>
Index: 32566 entries, 0 to 33119
Data columns (total 3 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   zip_code                          32566 non-null  int64
 1   public_transportation_pct         32566 non-null  float64
 2   public_transportation_population  32566 non-null  int64
dtypes: float64(1), int64(2)
memory usage: 1017.7 KB
None
```

Descripción Estadística del Dataset después de la Limpieza

	zip_code	public_transportation_pct	public_transportation_population
count	32566.00	32566.00	32566.00
mean	49661.29	1.94	234.27
std	27509.63	6.10	1324.16
min	601.00	0.00	0.00
25%	26775.50	0.00	0.00
50%	49775.50	0.00	0.00
75%	72064.75	1.20	43.00
max	99929.00	100.00	35139.00

In [5]:
```
# 4. Definir categorías de ventas potenciales (High/Low) basadas en el porcentaje d
data['potential_sales_category'] = np.where(data['public_transportation_pct'] > 10,
print(data.head())
```

```
   zip_code  public_transportation_pct  public_transportation_population  \
0      1379                        3.3                                13
1      1440                        0.4                                34
2      1505                        0.9                                23
3      1524                        0.5                                20
4      1529                        1.8                                32

  potential_sales_category
0                      Low
1                      Low
2                      Low
3                      Low
4                      Low
```

Explicación de la Aproximación Utilizada para Calcular Ventas Potenciales

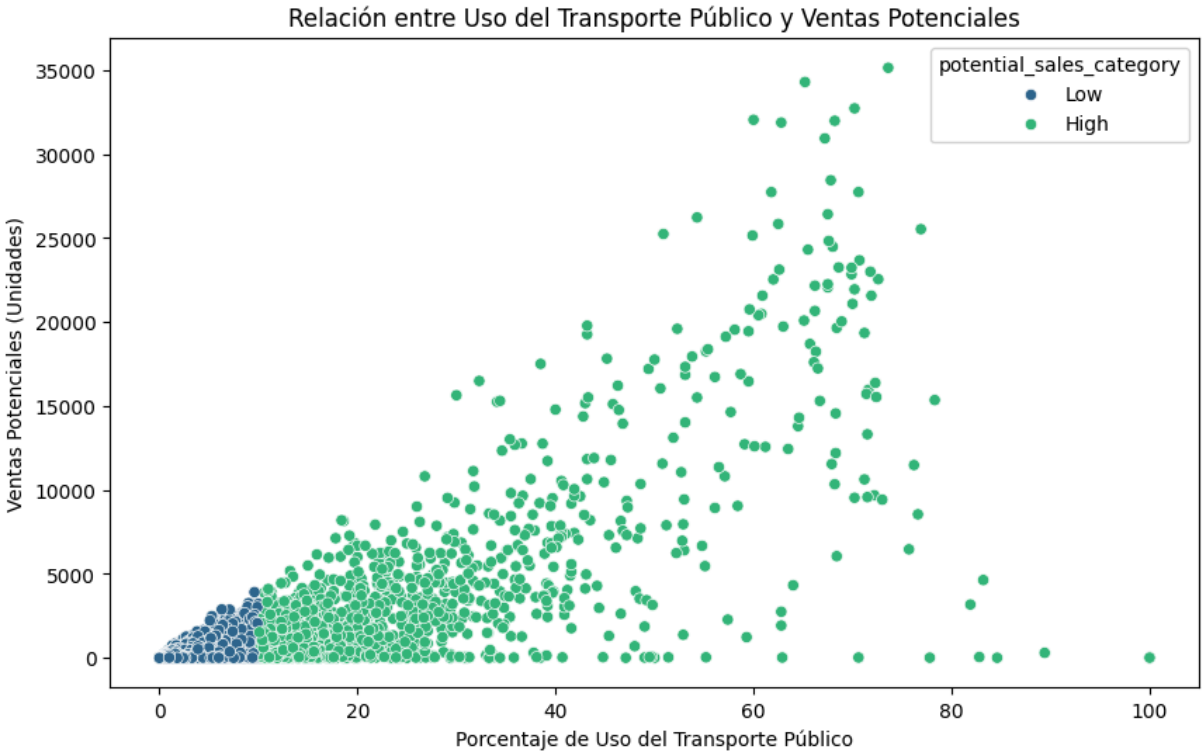
Para abordar la pregunta sobre las ventas potenciales promedio en zonas de alto y bajo uso del transporte público, hemos utilizado el promedio de la población que utiliza transporte

público (`public_transportation_population`) como un proxy para las ventas potenciales de scooters.

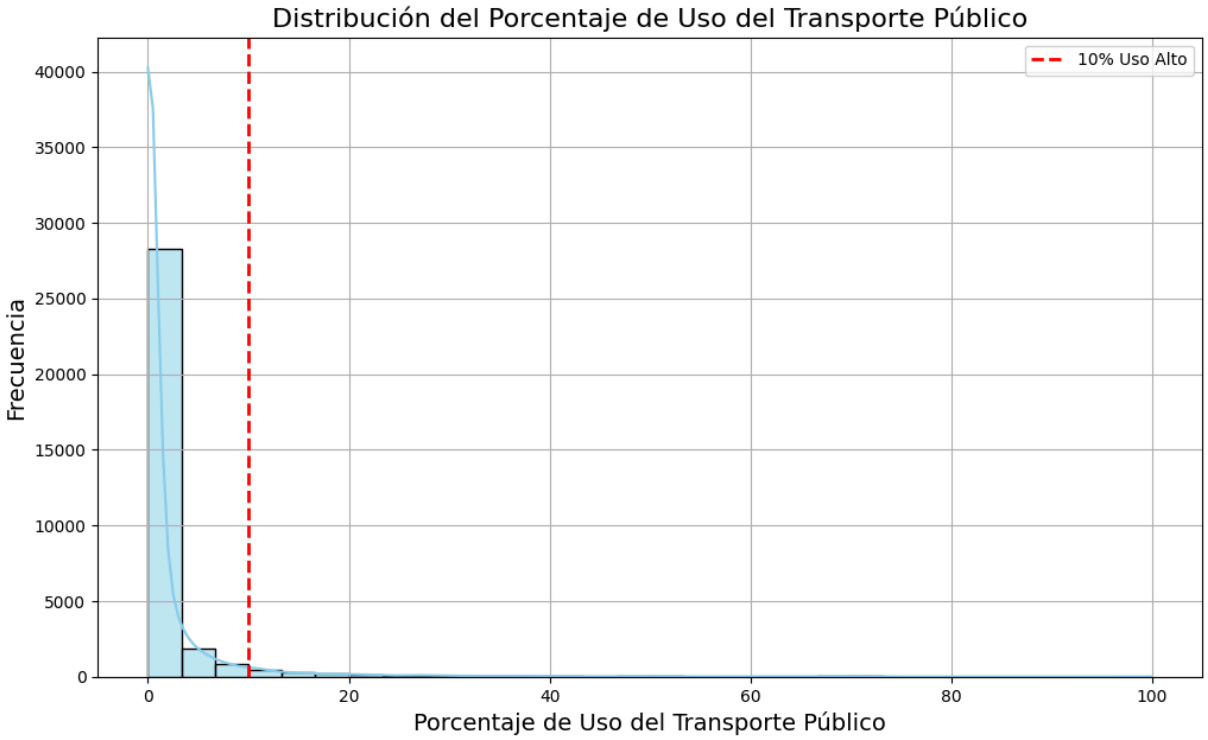
Un 'proxy' es una medida que se utiliza para representar o estimar otra variable que no se puede medir directamente. En este caso, estamos asumiendo que el número de personas que usan transporte público en una zona postal (public_transportation_population) puede servir como una indicación de cuántas ventas potenciales de scooters podría haber en esa zona.

In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 4. Calcular ventas potenciales promedio para zonas de alto y bajo uso del transpo
average_sales_high = data[data['potential_sales_category'] == 'High']['public_trans
average_sales_low = data[data['potential_sales_category'] == 'Low']['public_transpo

print(f'\nVentas potenciales promedio en zonas de alto uso de transporte público: {
print(f'Ventas potenciales promedio en zonas de bajo uso de transporte público: {av

# 6. Crear un diagrama de dispersión para visualizar la relación entre el uso del t
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='public_transportation_pct', y='public_transportation_
plt.title('Relación entre Uso del Transporte Público y Ventas Potenciales')
plt.xlabel('Porcentaje de Uso del Transporte Público')
plt.ylabel('Ventas Potenciales (Unidades)')
plt.show()
```

```
Ventas potenciales promedio en zonas de alto uso de transporte público: 3346.69 unid
ades
Ventas potenciales promedio en zonas de bajo uso de transporte público: 79.47 unidad
es
```



In [7]:
```
# 5. Histograma de distribución del porcentaje de uso del transporte público
plt.figure(figsize=(12, 7))
sns.histplot(data['public_transportation_pct'], kde=True, bins=30, color='skyblue',
plt.axvline(x=10, color='red', linestyle='--', linewidth=2, label='10% Uso Alto')
plt.title('Distribución del Porcentaje de Uso del Transporte Público', fontsize=16)
plt.xlabel('Porcentaje de Uso del Transporte Público', fontsize=14)
plt.ylabel('Frecuencia', fontsize=14)
plt.legend()
plt.grid(True)
plt.show()
```

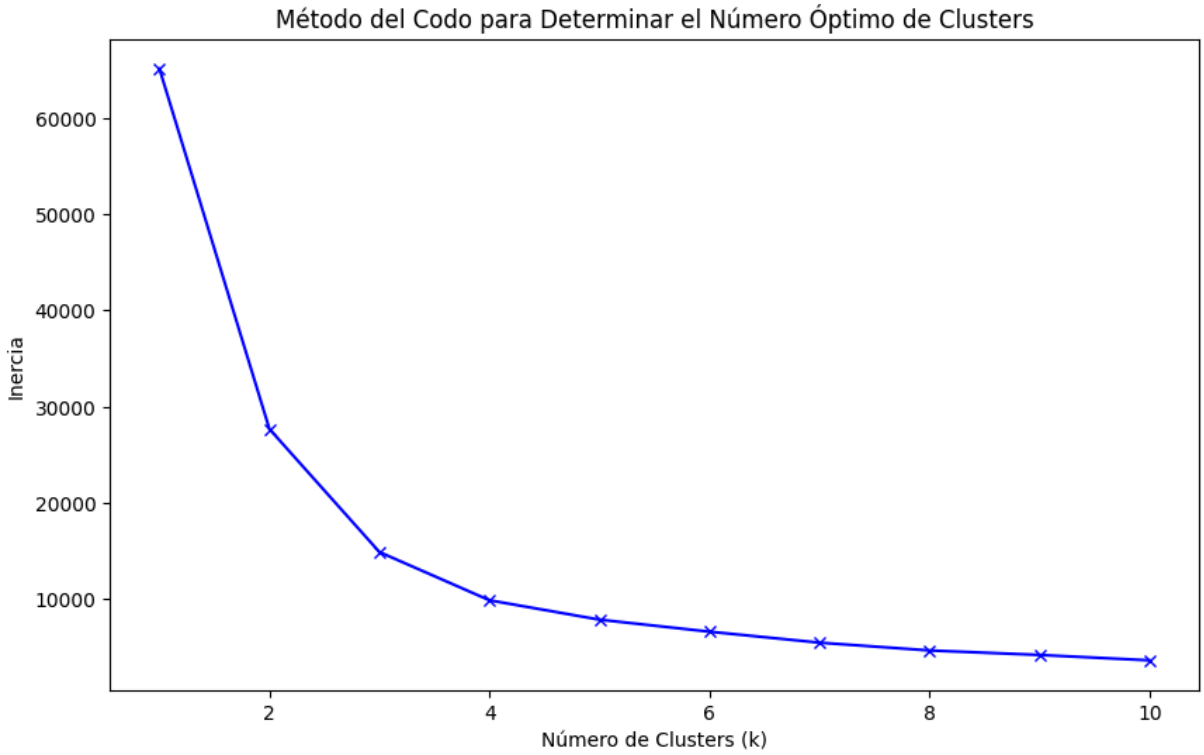


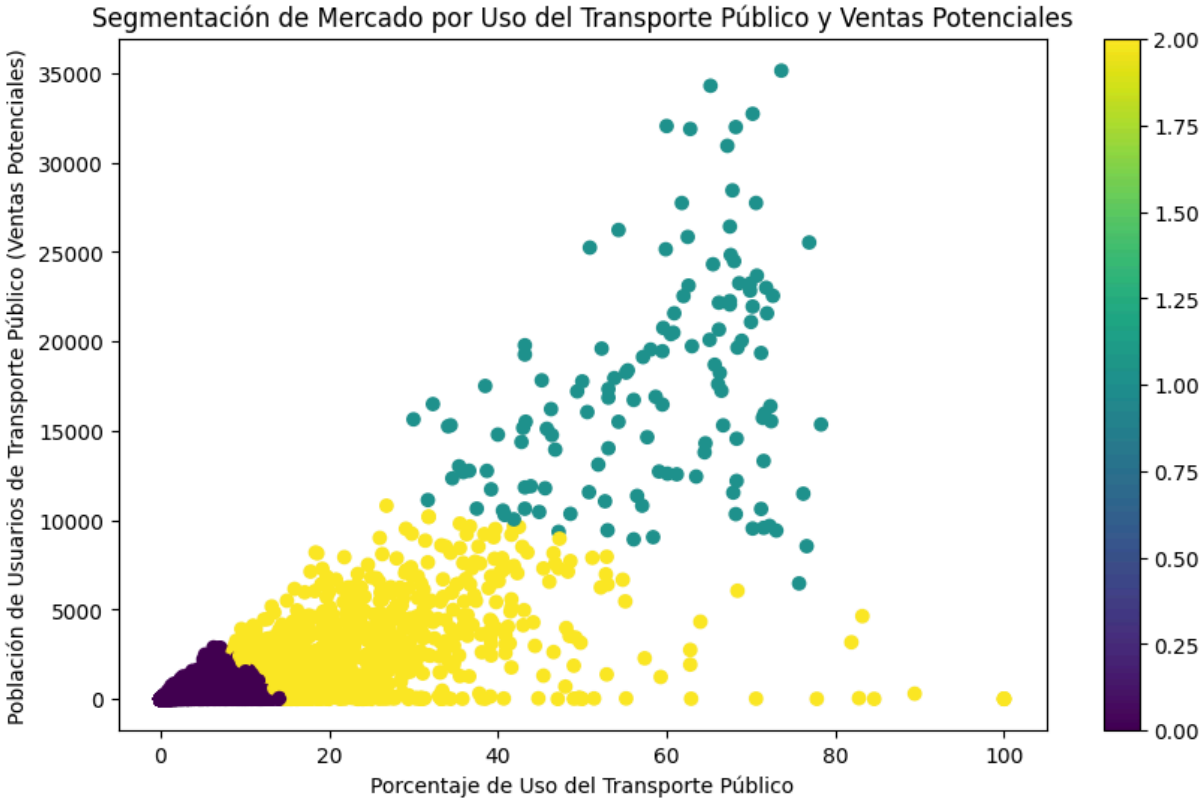
In [8]:
```
# 6. Exportar a excel

# Seleccionar las columnas necesarias
columns_to_export = ['zip_code', 'potential_sales_category', 'public_transportation

# Crear un nuevo dataframe con las columnas seleccionadas
export_data = data[columns_to_export].copy()

# Exportar el dataframe a un archivo Excel
export_data.to_excel('datos_completos.xlsx', index=False)

print("Los datos han sido exportados a 'datos_completos.xlsx'")
```

```
Los datos han sido exportados a 'datos_completos.xlsx'
```

In [9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Supongamos que ya tienes el dataframe 'data' cargado y limpiado

# Seleccionar las variables para el clustering
features = data[['public_transportation_pct', 'public_transportation_population']]

# Estandarizar las variables
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Método del codo para determinar el número óptimo de clusters
inertia = []
K = range(1, 11)
```

```python
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_features)
    inertia.append(kmeans.inertia_)

# Graficar el método del codo
plt.figure(figsize=(10, 6))
plt.plot(K, inertia, 'bx-')
plt.xlabel('Número de Clusters (k)')
plt.ylabel('Inercia')
plt.title('Método del Codo para Determinar el Número Óptimo de Clusters')
plt.show()

# Seleccionar el número óptimo de clusters (por ejemplo, 3)
optimal_k = 3

# Aplicar K-means con el número óptimo de clusters
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
kmeans.fit(scaled_features)
data['cluster'] = kmeans.labels_

# Visualizar los clusters
plt.figure(figsize=(10, 6))
scatter = plt.scatter(data['public_transportation_pct'], data['public_transportatio
plt.colorbar(scatter)
plt.title('Segmentación de Mercado por Uso del Transporte Público y Ventas Potencia
plt.xlabel('Porcentaje de Uso del Transporte Público')
plt.ylabel('Población de Usuarios de Transporte Público (Ventas Potenciales)')
plt.show()

# Mostrar estadísticas de cada cluster
for i in range(optimal_k):
    cluster_data = data[data['cluster'] == i]
    print(f'\nEstadísticas del Cluster {i}:')
    print(cluster_data[['public_transportation_pct', 'public_transportation_populat
```


Método del Codo para Determinar el Número Óptimo de Clusters
Inercia
Número de Clusters (k)
60000
50000
40000
30000
20000
10000
2
4
6
8
10


Segmentación de Mercado por Uso del Transporte Público y Ventas Potenciales
Población de Usuarios de Transporte Público (Ventas Potenciales)
Porcentaje de Uso del Transporte Público
35000
30000
25000
20000
15000
10000
5000
0
0
20
40
60
80
100
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

```
Estadísticas del Cluster 0:
       public_transportation_pct  public_transportation_population
count                31322.000000                      31322.000000
mean                     0.979842                         81.850457
std                      2.034399                        234.313849
min                      0.000000                          0.000000
25%                      0.000000                          0.000000
50%                      0.000000                          0.000000
75%                      0.900000                         31.000000
max                     14.600000                       2891.000000

Estadísticas del Cluster 1:
       public_transportation_pct  public_transportation_population
count                  132.000000                        132.000000
mean                    58.000000                      17417.984848
std                     12.250571                       6173.350755
min                     30.000000                       6468.000000
25%                     48.250000                      12536.750000
50%                     60.050000                      16422.500000
75%                     68.200000                      21206.250000
max                     78.300000                      35139.000000

Estadísticas del Cluster 2:
       public_transportation_pct  public_transportation_population
count                 1112.000000                       1112.000000
mean                    22.324730                       2487.759892
std                     11.809275                       2129.008648
min                      8.400000                          4.000000
25%                     14.700000                        912.250000
50%                     19.000000                       1999.000000
75%                     26.700000                       3398.500000
max                    100.000000                      10813.000000
```

In [10]:
```python
import pandas as pd

# Filtrar el dataframe para obtener solo las filas del cluster 1
cluster_1_data = data[data['cluster'] == 1]

# Mostrar la cantidad de códigos postales del cluster 1
print(f'Cantidad de códigos postales en el Cluster 1: {len(cluster_1_data)}')

# Seleccionar las columnas deseadas y ordenar la tabla de forma descendente en la c
cluster_1_data_sorted = cluster_1_data[['zip_code', 'public_transportation_pct', 'p

# Mostrar los códigos postales del cluster 1 en una tabla formateada
styled_table = cluster_1_data_sorted.style.set_caption('Códigos Postales del Cluste
display(styled_table)
```

```
Cantidad de códigos postales en el Cluster 1: 132
```

Códigos Postales del Cluster 1 Ordenados por Población de Usuarios de Transporte Público

	zip_code	public_transportation_pct	public_transportation_population
2633	11226	73.600000	35139
2827	11368	65.200000	34292
2580	11373	70.200000	32724
2828	11385	60.000000	32041
2836	10025	68.200000	31983
2661	11211	62.800000	31877
2759	11377	67.200000	30936
2630	11215	67.800000	28443
2724	11221	70.600000	27740
2756	11220	61.800000	27737
2721	11208	67.500000	26422
2635	11236	54.300000	26229
3048	10467	62.500000	25842
2932	10031	76.900000	25530
30870	60657	50.900000	25244
2755	11214	59.900000	25152
2743	11207	67.600000	24830
2676	11206	68.000000	24485
2987	10128	65.500000	24312
2871	10024	70.700000	23677
2689	11201	68.600000	23253
3073	10468	69.900000	23234
2974	10462	62.600000	23121
2748	11238	71.800000	22998
2931	10029	69.900000	22840
2745	11216	72.600000	22549
2805	11375	62.000000	22534
3051	10456	67.500000	22254
2921	10458	66.200000	22164

	zip_code	public_transportation_pct	public_transportation_population
2662	11372	67.500000	22063
2626	11233	70.200000	21947
2094	7030	60.900000	21572
2746	11225	71.900000	21570
2973	10452	70.000000	21085
2870	10023	59.600000	20753
2872	10032	66.200000	20662
2577	11218	60.800000	20489
2720	11203	60.500000	20408
2922	10453	65.100000	20089
2983	10033	68.900000	20037
30921	60614	43.200000	19784
2895	10028	63.000000	19731
2723	11212	68.400000	19652
2781	11229	52.300000	19599
2754	11209	58.100000	19544
2892	10009	59.500000	19450
2636	11237	71.200000	19350
2782	11234	43.200000	19266
2913	10463	57.200000	19126
2744	11213	65.700000	18695
2893	10011	55.400000	18381
2578	11223	55.200000	18256
2937	10027	66.300000	18237
2930	10002	53.800000	17946
32173	60640	45.200000	17823
2757	11230	50.000000	17763
3041	10457	66.100000	17610
26853	94110	38.500000	17508
2891	10003	53.100000	17334

	zip_code	public_transportation_pct	public_transportation_population
3043	10472	66.500000	17239
2624	11235	49.400000	17207
2807	11435	58.700000	16902
2576	11204	53.100000	16856
2722	11210	56.100000	16727
30928	60647	32.300000	16495
2030	7302	59.500000	16468
2753	11103	72.300000	16377
30849	60613	46.300000	16214
2944	10466	50.600000	16048
2672	11217	71.600000	15964
2780	11106	71.400000	15721
30851	60618	30.000000	15641
2964	10040	72.400000	15535
2758	11355	43.300000	15510
2789	11432	54.300000	15504
2982	10026	78.300000	15355
30753	60625	34.400000	15308
2625	11222	66.700000	15305
26854	94112	34.100000	15252
1973	7087	43.000000	15166
1974	7093	45.800000	15113
6219	20009	40.000000	14788
2925	10469	46.400000	14763
779	2128	57.700000	14643
2963	10034	68.300000	14558
2632	11219	42.800000	14377
2859	10460	64.600000	14303
2076	7306	53.100000	14024
2730	11434	46.800000	13949

	zip_code	public_transportation_pct	public_transportation_population
2779	11105	64.500000	13798
2741	11102	71.500000	13316
2646	11419	51.900000	13111
26819	94109	35.400000	13016
6332	20011	36.600000	12767
2980	10016	38.700000	12761
2645	11374	59.100000	12723
6387	20002	35.900000	12690
2788	11421	60.100000	12603
2629	11205	61.200000	12568
2690	11231	63.500000	12443
32253	60622	34.600000	12343
2905	10451	68.300000	12198
30853	60626	43.900000	11903
3030	10019	43.200000	11836
3032	10021	45.600000	11787
26915	94117	39.200000	11728
2975	10473	50.800000	11569
2943	10459	67.900000	11544
2628	11101	76.200000	11483
2979	10010	56.500000	11362
27176	94122	31.700000	11129
2869	10014	52.700000	11060
2914	10475	57.100000	10815
1976	7305	37.500000	10649
6011	19143	43.200000	10645
2742	11104	71.200000	10631
2637	11354	40.600000	10541
2877	10461	44.900000	10464
6334	20010	48.600000	10358

	zip_code	public_transportation_pct	public_transportation_population
2924	10455	68.200000	10343
28821	90057	40.800000	10296
32254	60660	41.900000	10052
2837	10030	72.200000	9685
3066	10035	71.500000	9575
2747	11232	70.200000	9537
2787	11418	53.000000	9440
2923	10454	73.000000	9424
2634	11228	47.200000	9352
2579	11224	58.400000	9054
2785	11369	56.100000	8932
3034	10039	76.600000	8551
2121	7310	75.700000	6468