

# Task 0

```
In [1]: pip install geopandas altair pandas jupyter
```

```
Requirement already satisfied: ipywidgets in ./anaconda3/lib/python3.11/site-packages (from jupyter) (8.0.4)
Requirement already satisfied: attrs>=17.4.0 in ./anaconda3/lib/python3.11/site-packages (from jsonschema>=3.0->altair) (22.1.0)
Requirement already satisfied: pyparsing!=0.1.0,!=0.1.1,!=0.1.2,!=0.1.3,!=0.1.4,!=0.1.5,!=0.1.6,!=0.1.7,!=0.1.8,!=0.1.9,!=0.1.10,!=0.1.11,!=0.1.12,!=0.1.13,!=0.1.14,!=0.1.15,!=0.1.16,!=0.1.17,!=0.1.18,!=0.1.19,!=0.1.20,!=0.1.21,!=0.1.22,!=0.1.23,!=0.1.24,!=0.1.25,!=0.1.26,!=0.1.27,!=0.1.28,!=0.1.29,!=0.1.30,!=0.1.31,!=0.1.32,!=0.1.33,!=0.1.34,!=0.1.35,!=0.1.36,!=0.1.37,!=0.1.38,!=0.1.39,!=0.1.40,!=0.1.41,!=0.1.42,!=0.1.43,!=0.1.44,!=0.1.45,!=0.1.46,!=0.1.47,!=0.1.48,!=0.1.49,!=0.1.50,!=0.1.51,!=0.1.52,!=0.1.53,!=0.1.54,!=0.1.55,!=0.1.56,!=0.1.57,!=0.1.58,!=0.1.59,!=0.1.60,!=0.1.61,!=0.1.62,!=0.1.63,!=0.1.64,!=0.1.65,!=0.1.66,!=0.1.67,!=0.1.68,!=0.1.69,!=0.1.70,!=0.1.71,!=0.1.72,!=0.1.73,!=0.1.74,!=0.1.75,!=0.1.76,!=0.1.77,!=0.1.78,!=0.1.79,!=0.1.80,!=0.1.81,!=0.1.82,!=0.1.83,!=0.1.84,!=0.1.85,!=0.1.86,!=0.1.87,!=0.1.88,!=0.1.89,!=0.1.90,!=0.1.91,!=0.1.92,!=0.1.93,!=0.1.94,!=0.1.95,!=0.1.96,!=0.1.97,!=0.1.98,!=0.1.99,!=0.1.100,!=0.1.101,!=0.1.102,!=0.1.103,!=0.1.104,!=0.1.105,!=0.1.106,!=0.1.107,!=0.1.108,!=0.1.109,!=0.1.110,!=0.1.111,!=0.1.112,!=0.1.113,!=0.1.114,!=0.1.115,!=0.1.116,!=0.1.117,!=0.1.118,!=0.1.119,!=0.1.120,!=0.1.121,!=0.1.122,!=0.1.123,!=0.1.124,!=0.1.125,!=0.1.126,!=0.1.127,!=0.1.128,!=0.1.129,!=0.1.130,!=0.1.131,!=0.1.132,!=0.1.133,!=0.1.134,!=0.1.135,!=0.1.136,!=0.1.137,!=0.1.138,!=0.1.139,!=0.1.140,!=0.1.141,!=0.1.142,!=0.1.143,!=0.1.144,!=0.1.145,!=0.1.146,!=0.1.147,!=0.1.148,!=0.1.149,!=0.1.150,!=0.1.151,!=0.1.152,!=0.1.153,!=0.1.154,!=0.1.155,!=0.1.156,!=0.1.157,!=0.1.158,!=0.1.159,!=0.1.160,!=0.1.161,!=0.1.162,!=0.1.163,!=0.1.164,!=0.1.165,!=0.1.166,!=0.1.167,!=0.1.168,!=0.1.169,!=0.1.170,!=0.1.171,!=0.1.172,!=0.1.173,!=0.1.174,!=0.1.175,!=0.1.176,!=0.1.177,!=0.1.178,!=0.1.179,!=0.1.180,!=0.1.181,!=0.1.182,!=0.1.183,!=0.1.184,!=0.1.185,!=0.1.186,!=0.1.187,!=0.1.188,!=0.1.189,!=0.1.190,!=0.1.191,!=0.1.192,!=0.1.193,!=0.1.194,!=0.1.195,!=0.1.196,!=0.1.197,!=0.1.198,!=0.1.199,!=0.1.200,!=0.1.201,!=0.1.202,!=0.1.203,!=0.1.204,!=0.1.205,!=0.1.206,!=0.1.207,!=0.1.208,!=0.1.209,!=0.1.210,!=0.1.211,!=0.1.212,!=0.1.213,!=0.1.214,!=0.1.215,!=0.1.216,!=0.1.217,!=0.1.218,!=0.1.219,!=0.1.220,!=0.1.221,!=0.1.222,!=0.1.223,!=0.1.224,!=0.1.225,!=0.1.226,!=0.1.227,!=0.1.228,!=0.1.229,!=0.1.230,!=0.1.231,!=0.1.232,!=0.1.233,!=0.1.234,!=0.1.235,!=0.1.236,!=0.1.237,!=0.1.238,!=0.1.239,!=0.1.240,!=0.1.241,!=0.1.242,!=0.1.243,!=0.1.244,!=0.1.245,!=0.1.246,!=0.1.247,!=0.1.248,!=0.1.249,!=0.1.250,!=0.1.251,!=0.1.252,!=0.1.253,!=0.1.254,!=0.1.255,!=0.1.256,!=0.1.257,!=0.1.258,!=0.1.259,!=0.1.260,!=0.1.261,!=0.1.262,!=0.1.263,!=0.1.264,!=0.1.265,!=0.1.266,!=0.1.267,!=0.1.268,!=0.1.269,!=0.1.270,!=0.1.271,!=0.1.272,!=0.1.273,!=0.1.274,!=0.1.275,!=0.1.276,!=0.1.277,!=0.1.278,!=0.1.279,!=0.1.280,!=0.1.281,!=0.1.282,!=0.1.283,!=0.1.284,!=0.1.285,!=0.1.286,!=0.1.287,!=0.1.288,!=0.1.289,!=0.1.290,!=0.1.291,!=0.1.292,!=0.1.293,!=0.1.294,!=0.1.295,!=0.1.296,!=0.1.297,!=0.1.298,!=0.1.299,!=0.1.300,!=0.1.301,!=0.1.302,!=0.1.303,!=0.1.304,!=0.1.305,!=0.1.306,!=0.1.307,!=0.1.308,!=0.1.309,!=0.1.310,!=0.1.311,!=0.1.312,!=0.1.313,!=0.1.314,!=0.1.315,!=0.1.316,!=0.1.317,!=0.1.318,!=0.1.319,!=0.1.320,!=0.1.321,!=0.1.322,!=0.1.323,!=0.1.324,!=0.1.325,!=0.1.326,!=0.1.327,!=0.1.328,!=0.1.329,!=0.1.330,!=0.1.331,!=0.1.332,!=0.1.333,!=0.1.334,!=0.1.335,!=0.1.336,!=0.1.337,!=0.1.338,!=0.1.339,!=0.1.340,!=0.1.341,!=0.1.342,!=0.1.343,!=0.1.344,!=0.1.345,!=0.1.346,!=0.1.347,!=0.1.348,!=0.1.349,!=0.1.350,!=0.1.351,!=0.1.352,!=0.1.353,!=0.1.354,!=0.1.355,!=0.1.356,!=0.1.357,!=0.1.358,!=0.1.359,!=0.1.360,!=0.1.361,!=0.1.362,!=0.1.363,!=0.1.364,!=0.1.365,!=0.1.366,!=0.1.367,!=0.1.368,!=0.1.369,!=0.1.370,!=0.1.371,!=0.1.372,!=0.1.373,!=0.1.374,!=0.1.375,!=0.1.376,!=0.1.377,!=0.1.378,!=0.1.379,!=0.1.380,!=0.1.381,!=0.1.382,!=0.1.383,!=0.1.384,!=0.1.385,!=0.1.386,!=0.1.387,!=0.1.388,!=0.1.389,!=0.1.390,!=0.1.391,!=0.1.392,!=0.1.393,!=0.1.394,!=0.1.395,!=0.1.396,!=0.1.397,!=0.1.398,!=0.1.399,!=0.1.400,!=0.1.401,!=0.1.402,!=0.1.403,!=0.1.404,!=0.1.405,!=0.1.406,!=0.1.407,!=0.1.408,!=0.1.409,!=0.1.410,!=0.1.411,!=0.1.412,!=0.1.413,!=0.1.414,!=0.1.415,!=0.1.416,!=0.1.417,!=0.1.418,!=0.1.419,!=0.1.420,!=0.1.421,!=0.1.422,!=0.1.423,!=0.1.424,!=0.1.425,!=0.1.426,!=0.1.427,!=0.1.428,!=0.1.429,!=0.1.430,!=0.1.431,!=0.1.432,!=0.1.433,!=0.1.434,!=0.1.435,!=0.1.436,!=0.1.437,!=0.1.438,!=0.1.439,!=0.1.440,!=0.1.441,!=0.1.442,!=0.1.443,!=0.1.444,!=0.1.445,!=0.1.446,!=0.1.447,!=0.1.448,!=0.1.449,!=0.1.450,!=0.1.451,!=0.1.452,!=0.1.453,!=0.1.454,!=0.1.455,!=0.1.456
```

## Task 1:

```
In [2]: import pandas as pd
import geopandas as gpd
import altair as alt
from shapely.geometry import Point
```

```
df = pd.read_csv( /Users/manasadattakandimalla/Downloads/CS424-Lab 3/Trip_Mileage.csv)
geometry = [Point(xy) for xy in zip(df['Pickup_Centroid_Longitude'], df['Pickup_Centroid_Latitude'])]
gdf = gpd.GeoDataFrame(df, geometry=geometry, crs=4326).sample(1000)
gdf = gdf.rename(columns={"Trip_Seconds": "Trip_Seconds", "Trip_Miles": "Trip_Miles" })
```

```
In [3]: gdf.head()
```

Out[3]:

Trip ID		
63234	8d0da516d29134a1fe38a3ff38e7bffc1c73fb4	f5b5de5c0c42601f6878d82eeb1984d06c977aaf
66108	6b64bb409a807416fdeb55f48dffedf0e82ba53c	e637e1afe0dab0e2d04c635335602caf1045d43a
88101	9405c3f8f04e16f4d45e1cf7d660a187f68ac091	791d74df896226a452a8e223e2ec9fa0df7d80b1
19184	8ea27993206011b779446035822e877741cc9d12	b515261b90c81a5966755dbbb08f7cf0ff3928a
41466	2fa0e111fa9bf33628b7a5130f94b1a44a8b9b13	3beee2ac6f2972630c37deb940e1b66e774e228f

5 rows × 24 columns

## Task 2:

```
In [4]: chicago = gpd.read_file('/Users/manasadattakandimalla/Downloads/CS424-
```

```
In [5]: joined = gpd.sjoin(gdf, chicago, predicate='within')
joined = joined.groupby('zip').mean(numeric_only=True)
joined = joined.filter(['Fare'])
```

```
In [6]: merged = chicago.merge(joined, on='zip')
```

## Task 3:

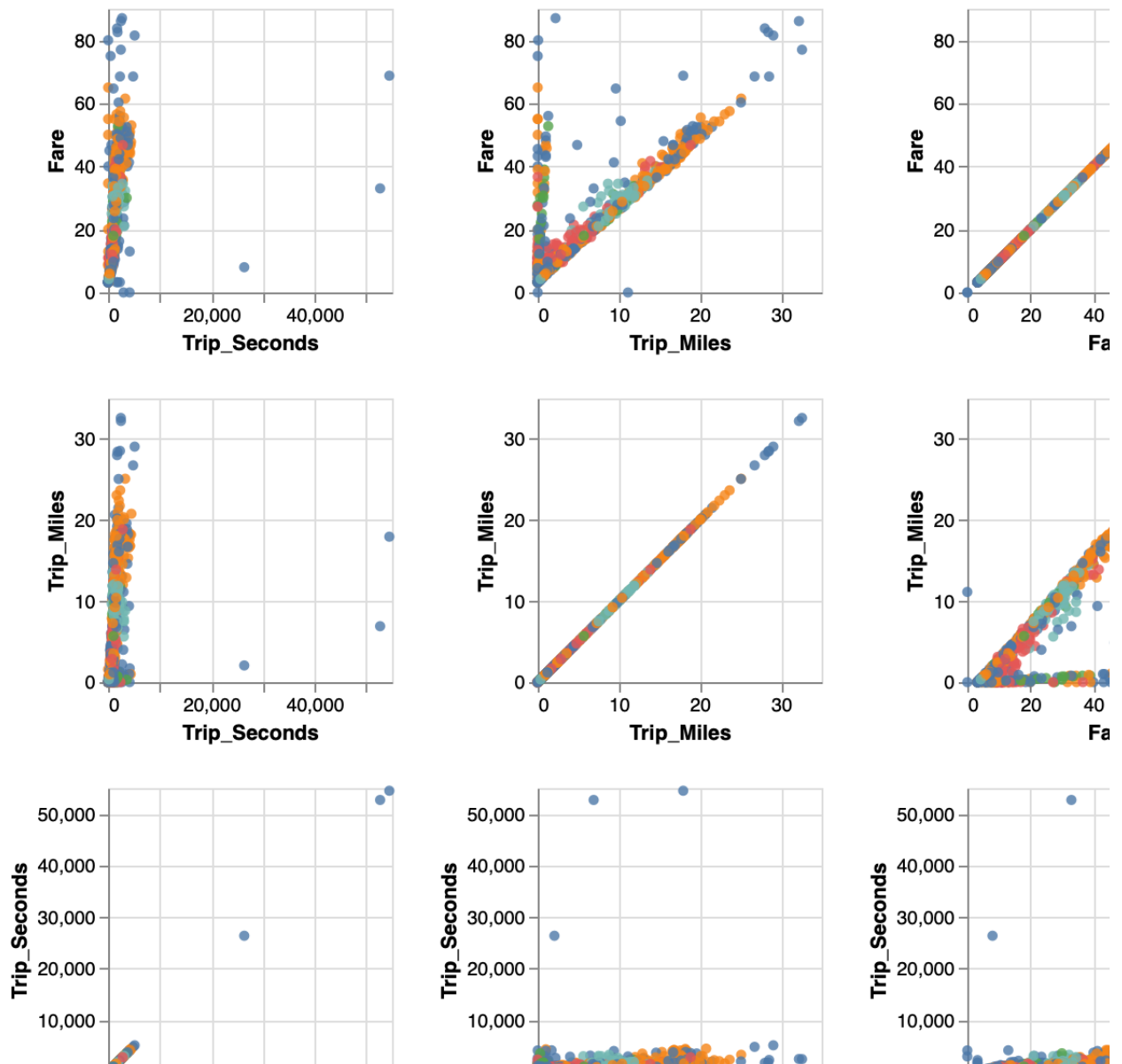
```
In [7]:
```

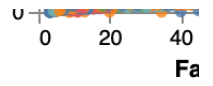
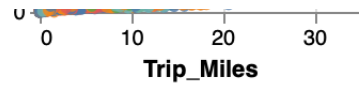
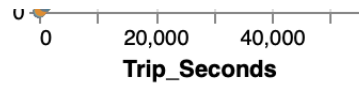
```
brush = alt.selection_interval()

matrix = alt.Chart(gdf).mark_circle().add_params(brush).encode(
    alt.X(alt.repeat("column"), type='quantitative'),
    alt.Y(alt.repeat("row"), type='quantitative'),
    color=alt.condition(brush, 'Payment Type:N', alt.value('grey')),
    opacity=alt.condition(brush, alt.value(0.8), alt.value(0.1))
).properties(
    width=150,
    height=150
).repeat(
    row=['Fare', 'Trip_Miles', 'Trip_Seconds'],
    column=['Trip_Seconds', 'Trip_Miles', 'Fare']
)

matrix
```

Out[7]:

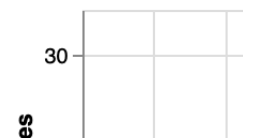
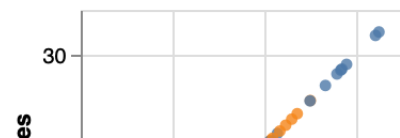
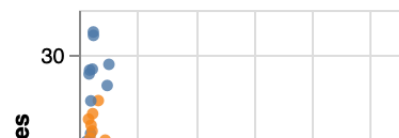
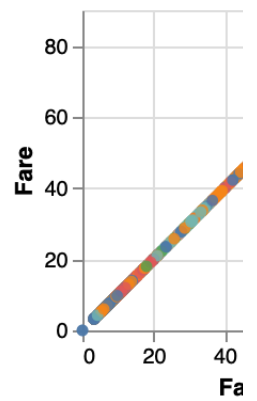
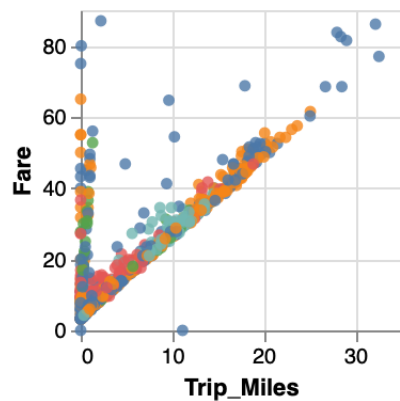
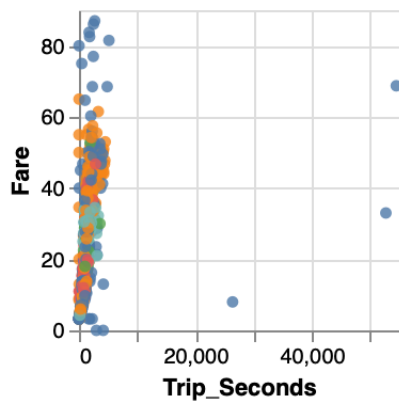
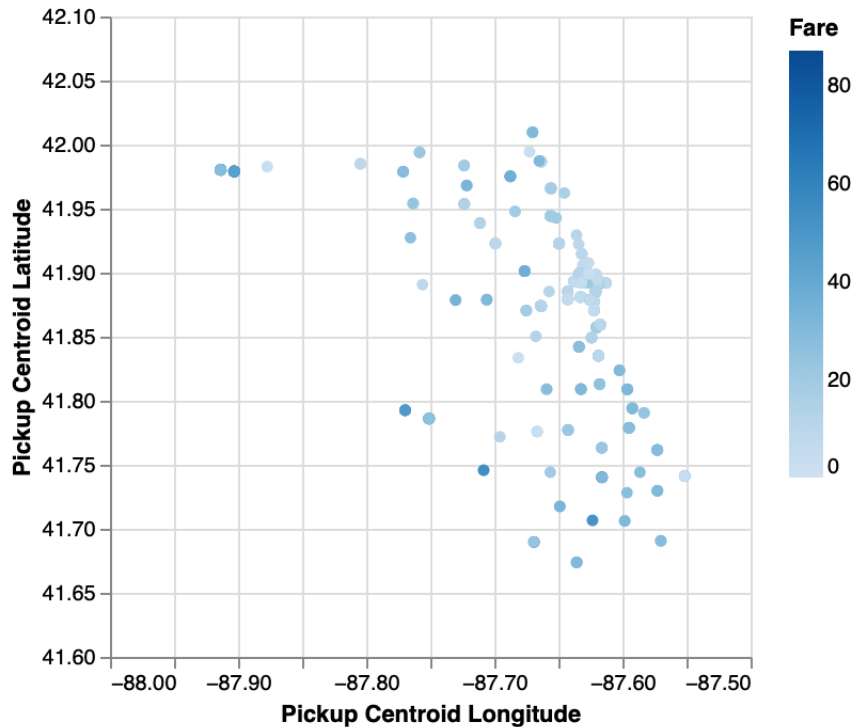


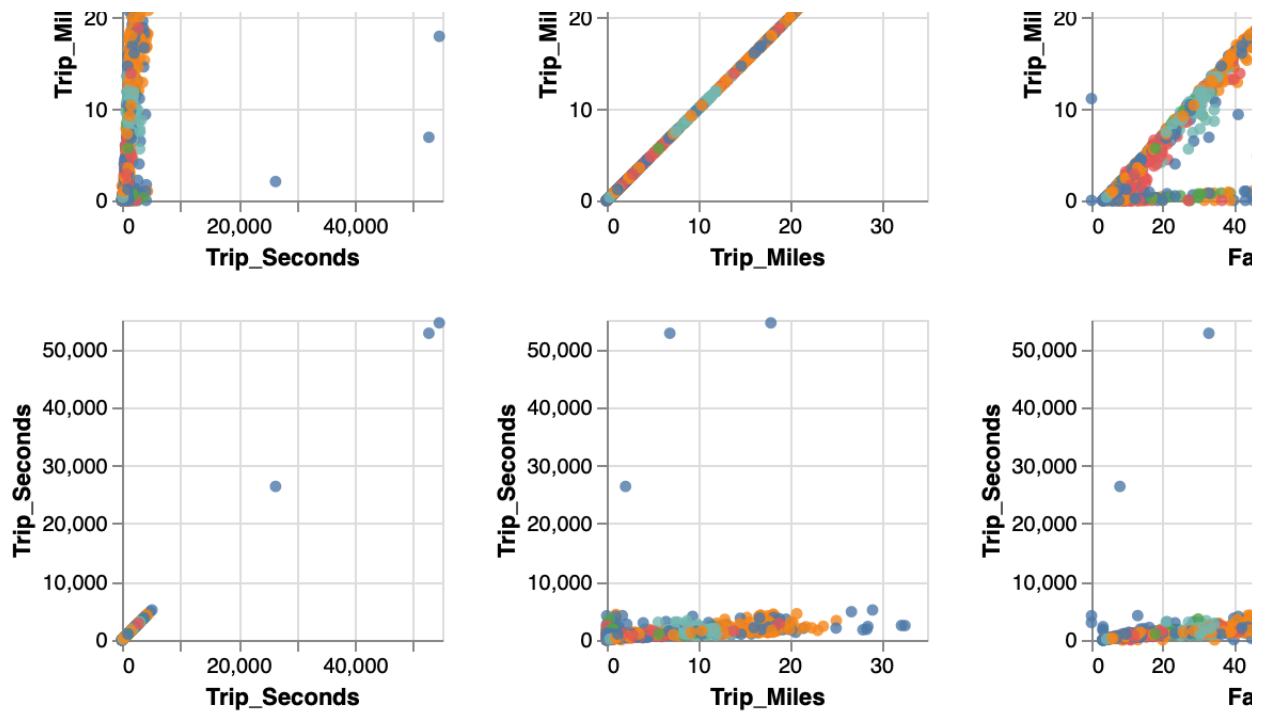


```
In [8]: scatter = alt.Chart(gdf).mark_circle().encode(
    x=alt.Y('Pickup Centroid Longitude',scale=alt.Scale(domain=[-88.0,
    y=alt.Y('Pickup Centroid Latitude',scale=alt.Scale(domain=[41.6, 4
    color='Fare',
    opacity=alt.condition(brush, alt.value(1), alt.value(0))
    )
```

```
In [9]: (scatter & matrix).add_params(brush)
```

Out [9]:





## Task 4 (Bubble Plot)

In [10]:

```

import altair as alt

brush = alt.selection_interval()

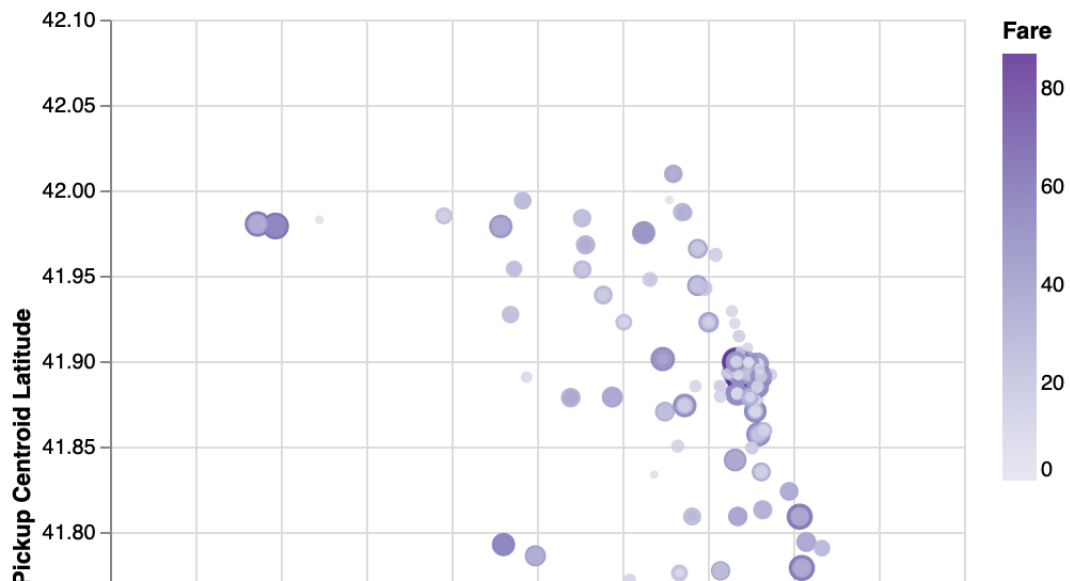
# Bubble plot for spatial attributes (Pickup locations)
bubble_scatter = alt.Chart(gdf).mark_circle().encode(
    x=alt.X('Pickup Centroid Longitude', scale=alt.Scale(domain=[-88.0, -87.5])),
    y=alt.Y('Pickup Centroid Latitude', scale=alt.Scale(domain=[41.6, 42.1])),
    size=alt.Size('Fare', scale=alt.Scale(range=[10, 200]), legend=Non),
    color=alt.Color('Fare', scale=alt.Scale(scheme='purples')),
    opacity=alt.condition(brush, alt.value(0.8), alt.value(0.1))
).add_params(brush).properties(
    width=400,
    height=400
)

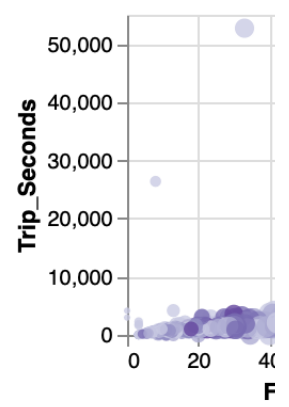
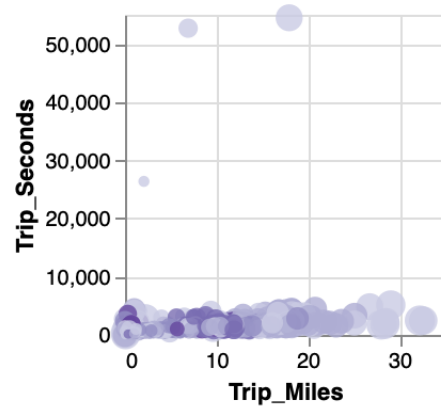
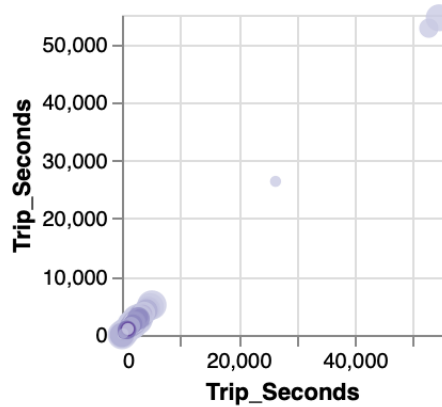
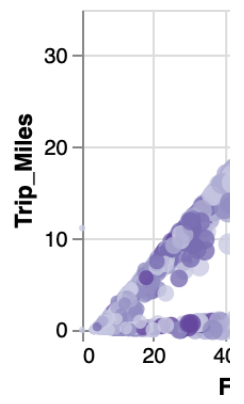
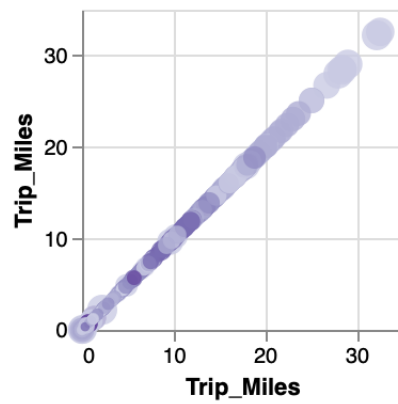
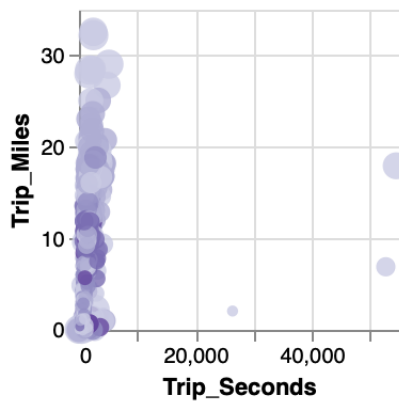
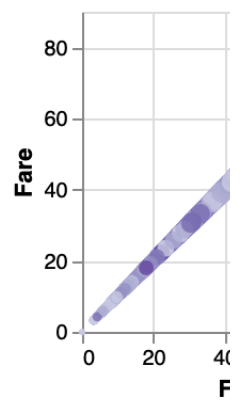
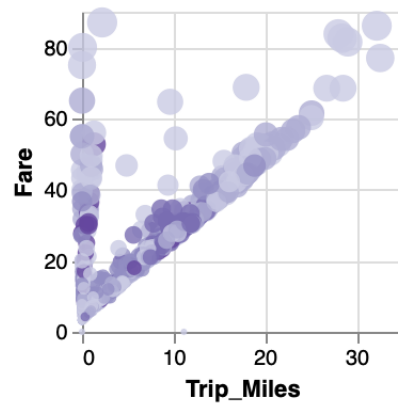
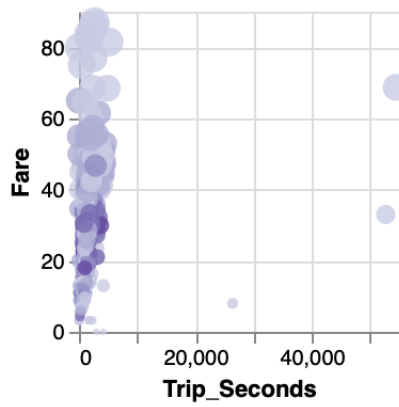
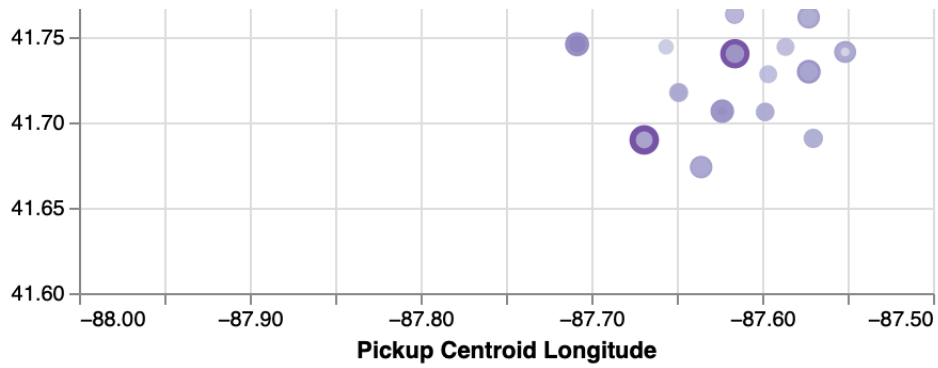
# Bubble matrix for linking
bubble_matrix = alt.Chart(gdf).mark_circle().encode(
    alt.X(alt.repeat("column"), type='quantitative'),
    alt.Y(alt.repeat("row"), type='quantitative'),
    size=alt.Size('Fare', scale=alt.Scale(range=[10, 200]), legend=Non),
    color=alt.condition(brush, alt.Color('Payment Type:N', scale=alt.S),
    opacity=alt.condition(brush, alt.value(0.8), alt.value(0.1))
).properties(
    width=150,
    height=150
).repeat(
    row=['Fare', 'Trip_Miles', 'Trip_Seconds'],
    column=['Trip_Seconds', 'Trip_Miles', 'Fare']
).add_params(brush)

# Combine both charts
(bubble_scatter & bubble_matrix)

```

Out[10]:





In [ ]:

