Parking Tickets

Jonathan Kacvinsky

2026-01-31

"This submission is my work alone and complies with the 30538 integrity policy." Add your initials to indicate your agreement: JK

Before you begin this problem set, please rename `ps3_template.qmd` as `ps3.qmd`.

Please submit correctly and rename ps3_template.qmd as ps3.qmd. Knit your completed solution `ps3.qmd` to make `ps3.pdf`. Push both `ps3.qmd` and `ps3.pdf` to your Github repo.

```python
import os

print(os.getcwd())
new_path = r"c:\Users\home-admin\Documents\University of Chicago\Data
 ↪  Analytics and Visualization for Public Policy"
os.chdir(new_path)
```

```
C:\Users\home-admin\Documents\University of Chicago\Data Analytics and
Visualization for Public Policy
```

```python
import pandas as pd
import numpy as np
import altair as alt
import time

import warnings
warnings.filterwarnings('ignore')
alt.renderers.enable("png")
alt.data_transformers.disable_max_rows()

park = pd.read_csv("parking_tickets_one_percent.csv", index_col=0)
park
```

	ticket_number	issue_date	violation_location	license_plate_number
1	5.148290e+07	2007-01-01 01:25:00	5762 N AVONDALE	d41ee9a4cb0676e641399ad14aaa
2	5.068150e+07	2007-01-01 01:51:00	2724 W FARRAGUT	3395fd3f71f18f9ea4f0a8e1f13bf0a
3	5.157970e+07	2007-01-01 02:22:00	1748 W ESTES	302cb9c55f63ff828d7315c5589d9
4	5.126220e+07	2007-01-01 02:35:00	4756 N SHERIDAN	94d018f52c7990cea326d1810a327
5	5.189800e+07	2007-01-01 03:50:00	7134 S CAMPBELL	876dd3a95179f4f1d720613f6e32a
...	...	...	...	...
287454	9.190000e+09	2018-05-14 14:51:00	1128 W MONROE	31f8a07f5ffc423447ceb2e99a5dc5
287455	9.190000e+09	2018-05-14 16:34:00	1820 N MILWAUKEE AVE	7533cbfcb6cd16743d6f3c77af51b
287456	9.190000e+09	2018-05-14 16:52:00	122 E 21ST ST	3af6ff915e8335ea396cb09761a1f3
287457	9.190000e+09	2018-05-14 18:04:00	10 S DEARBORN ST	fa986644c31a2c97f4c260334175e
287458	9.190000e+09	2018-05-14 20:56:00	2201 W ARTHUR	5fdb8dd85edbb1d262388d0b33e8

Data cleaning continued

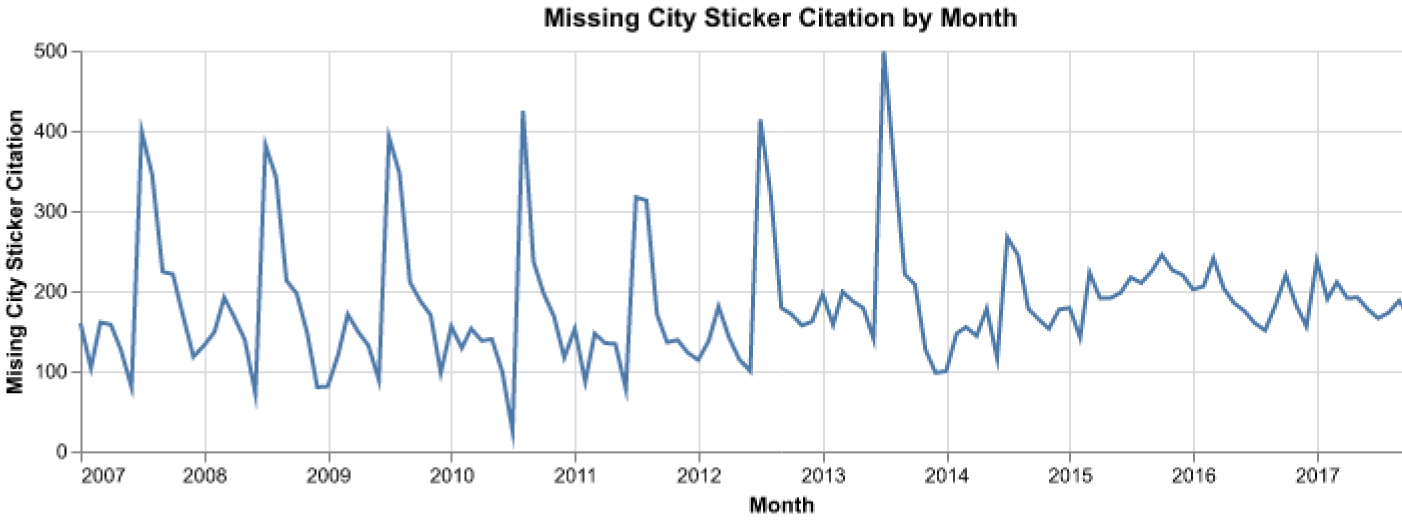
We would like to hear how you used AI (if at all) on this section. Where did you find it useful/not useful? We are asking for knowledge-sharing purposes; your answer will not impact your grade on the assignment.
Answer:
1. The old violation code was a $120 fine, the new is a $200 fine. 2. $60 for street cleaning citation, $50 for expired meter citation, $75-$135 for large passenger vehicles, $60 for late city sticker purchase

Revenue increase from "missing city sticker" tickets

We would like to hear how you used AI (if at all) on this section. Where did you find it useful/not useful? We are asking for knowledge-sharing purposes; your answer will not impact your grade on the assignment.
Answer: I used AI to fix my to_period("M").dt.to_timestamp() organization of monthly rows. I also used it to help calculate repayment rates and to translate that to revenue gained per ticket.
1.

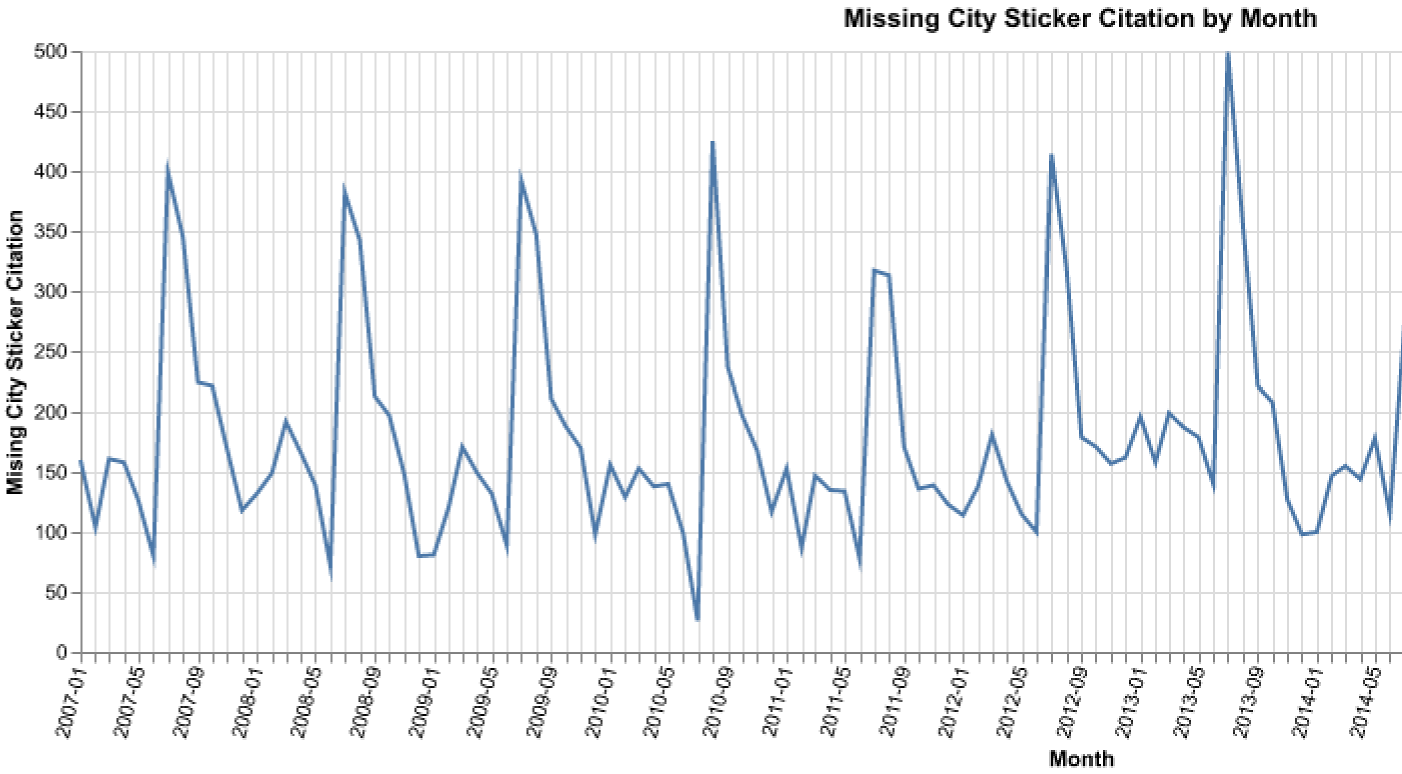
```
park["violation_missing"] = np.where(
    park["violation_code"].isin(["0964125", "0964125B"]),
    "Missing City Sticker",
    "Other"
)
```

```
park["issue_date"] = pd.to_datetime(park["issue_date"])
park["month"] = park["issue_date"].dt.to_period("M").dt.to_timestamp()

month_missing = (park[park["violation_missing"] == "Missing City
 ↪  Sticker"].groupby("month").size().reset_index(name="count"))

alt.Chart(month_missing).mark_line().encode(
    alt.X("month:T", title="Month"),
    alt.Y("count:Q", title = "Mising City Sticker Citation"),
).properties(
    width=700,
    height=200,
    title="Missing City Sticker Citation by Month"
)
```



2.

```
alt.Chart(month_missing).mark_line().encode(
    alt.X("month:T", title="Month", axis=alt.Axis(format="%Y-%m",
 ↪  tickCount="month", labelAngle=-75)),
    alt.Y("count:Q", title="Mising City Sticker Citation"),
).properties(
    width=1000,
    height=300,
    title="Missing City Sticker Citation by Month"
)
```



#Website used: altair-viz.github.io/user_guide/customization.html

3.

```
park_2013 = park[
    (park["issue_date"] >= "2013-01-01") &
    (park["issue_date"] < "2014-01-01") &
    (park["violation_code"].isin(["0964125", "0964125B"]))
]
print(f"Projected Revenue After Citation Increase: ${len(park_2013) * 80 *
↪  100}")
```

```
Projected Revenue After Citation Increase: $20536000
```

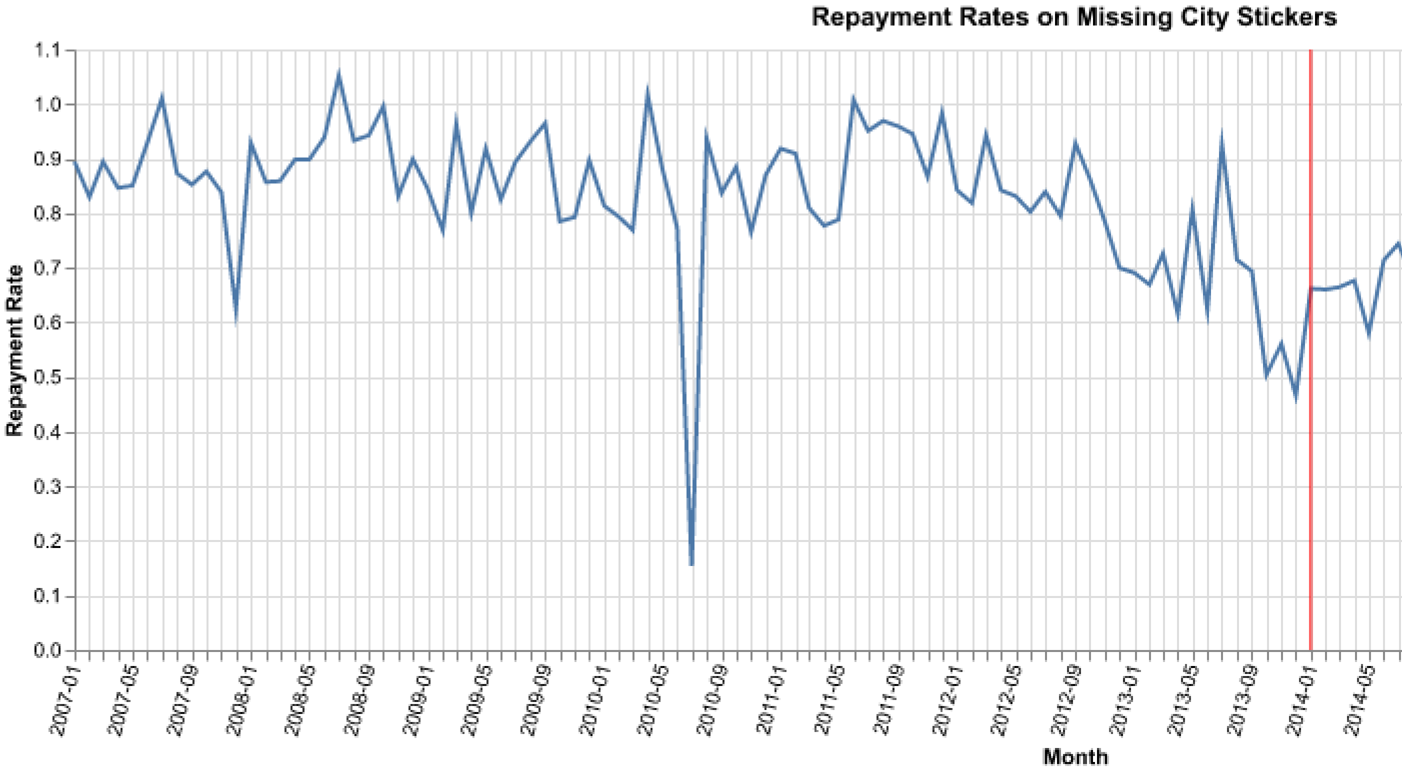
4.

```
park_2014 = park[
    (park["issue_date"] >= "2014-01-01") &
    (park["issue_date"] < "2015-01-01") &
    (park["violation_code"].isin(["0964125", "0964125B"])) &
    (park["ticket_queue"] == "Paid")
]

print(f"Actual Revenue After Citation Increase:
 ↪  ${park_2014["total_payments"].sum() * 100}")
```

```
Actual Revenue After Citation Increase: $25243600.0
```

5.

```
miss_stick = park[park["violation_code"].isin(["0964125",
 ↪  "0964125B"])].copy()
miss_stick["month"] =
 ↪  miss_stick["issue_date"].dt.to_period("M").dt.to_timestamp()
repay_rate =
 ↪  (miss_stick.groupby("month").agg(fine_amount=("fine_level1_amount","sum"),
 ↪  tot_pay=("total_payments", "sum")).reset_index())
repay_rate["repayment_rate"] =
 ↪  (repay_rate["tot_pay"]/repay_rate["fine_amount"])

re_ra = alt.Chart(repay_rate).mark_line().encode(
    alt.X("month:T", title="Month", axis=alt.Axis(format="%Y-%m",
 ↪  tickCount="month", labelAngle=-75)),
    alt.Y("repayment_rate:Q", title = "Repayment Rate"),
).properties(
    width=1000,
    height=300,
    title="Repayment Rates on Missing City Stickers"
)

pol_change = (
    alt.Chart(pd.DataFrame({"month": ["2014-01-01"]}))
    .mark_rule(
        color="red"
    )
    .encode(
        x = "month:T"
    )
)

re_ra + pol_change
```

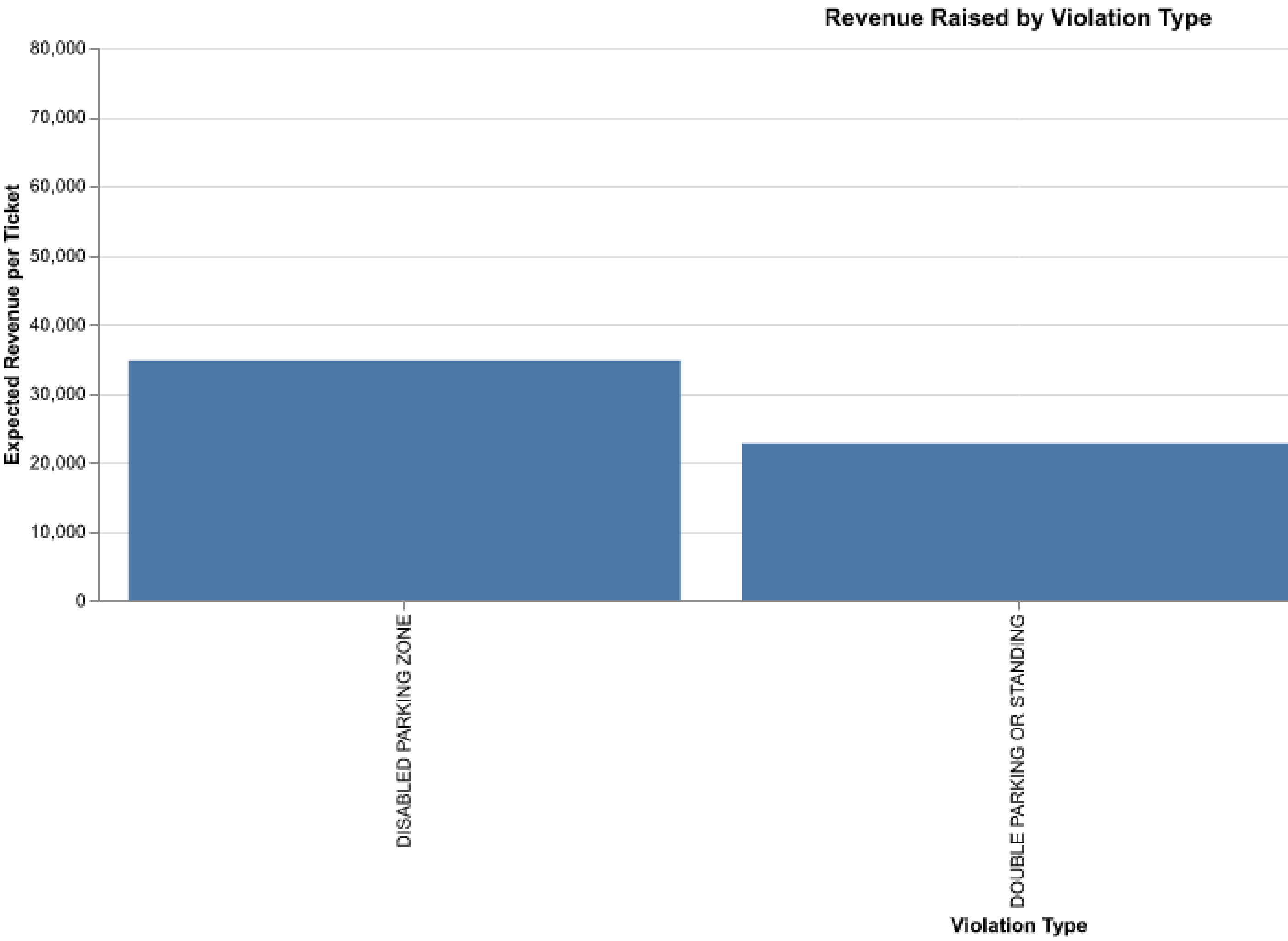


#After the policy was introduced, repayment rates dropped.

6.

```
viol_types = park[
    (park["issue_date"] >= "2014-01-01") &
    (park["issue_date"] < "2015-01-01") &
    (park["violation_description"].isin(["DISABLED PARKING ZONE", "DOUBLE
 ↪  PARKING OR STANDING", "WITHIN 15' OF FIRE HYDRANT"]))
].copy()

recommended =
 ↪  (viol_types.groupby("violation_description").agg(tickets_issued=("violation_code",
 ↪  "count"), total_fines=("fine_level1_amount", "sum"),
 ↪  total_paid=("total_payments", "sum")).reset_index())
```

```
recommended["repayment_rate"] =
 ↪  (recommended["total_paid"]/recommended["total_fines"])
recommended["revenue"] = (recommended["total_fines"] *
 ↪  recommended["repayment_rate"])

recommended

rec_graph = alt.Chart(recommended).mark_bar().encode(
    alt.X("violation_description:N", title="Violation Type"),
    alt.Y("revenue:Q", title = "Expected Revenue per Ticket"),
).properties(
    width=1000,
    height=300,
    title="Revenue Raised by Violation Type"
)

rec_graph
```



#I would recommend increasing the price of parking in disabled parking zones, double parking or standing, and parking within 15' of a fire hydrant. #These are projections for if the tickets were the same price, but each of these violation types make substantial revenue, and an increase in any of them would make substantial money. #Even if the money raised isn't as much as it is for city sticker citations, each of these prevent potential security risks for vulnerable populations.

Headlines and sub-messages

We would like to hear how you used AI (if at all) on this section. Where did you find it useful/not useful? We are asking for knowledge-sharing purposes; your answer will not impact your grade on the assignment.

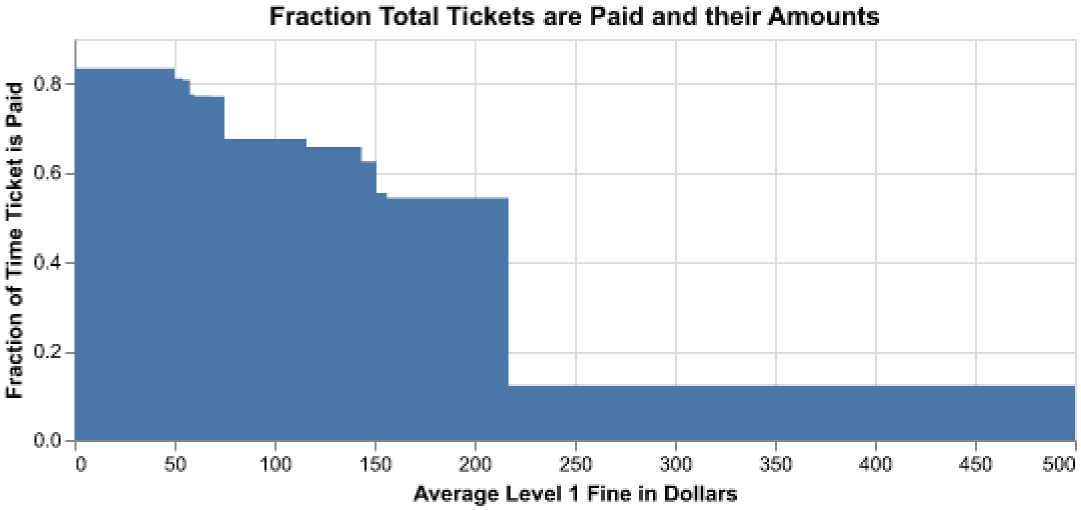
Answer: I used ai to help choose a good format for Tooltip.
1.

```
new_park = park.copy()
new_park["paid"] = np.where(park["ticket_queue"] == "Paid", 1, 0)
tot_tick =
 ↪  new_park.groupby("violation_description").agg(tickets_issued=("violation_description",
 ↪  "size"), fraction_paid=("paid", "mean"), avg_fine=("fine_level1_amount",
 ↪  "mean")).reset_index().sort_values("tickets_issued", ascending=False)
print(tot_tick.head())
```

```
                        violation_description  tickets_issued  fraction_paid  
                        \
23   EXPIRED PLATES OR TEMPORARY REGISTRATION           44811       0.604361   
101                           STREET CLEANING           28712       0.811612   
90                  RESIDENTIAL PERMIT PARKING           23683       0.742262   
19   EXP. METER NON-CENTRAL BUSINESS DISTRICT           20600       0.792913   
81        PARKING/STANDING PROHIBITED ANYTIME           19753       0.705817   

      avg_fine  
23   54.968869  
101  54.004249  
90   66.338302  
19   46.598058  
81   66.142864  
```

2.

```
tot_tick_100 = tot_tick[tot_tick["tickets_issued"] >= 100]

alt.Chart(tot_tick_100).mark_circle().encode(
    alt.X("avg_fine:Q", title="Average Level 1 Fine in Dollars"),
    alt.Y("fraction_paid:Q", title="Fraction of Time Ticket is Paid"),
    tooltip=[
        "violation_description:N",
        "tickets_issued:Q",
        alt.Tooltip("fraction_paid:Q", format=".2f"),
        alt.Tooltip("avg_fine:Q", format="$.0f")
    ],
).properties(
    width=500,
    height=200,
    title="Fraction Total Tickets are Paid and their Amounts"
)

#Headline and sub-message: The average amount of level 1 fines in dollars
 ↪ decreases as the fraction of time they are paid.

alt.Chart(tot_tick_100).mark_bar().encode(
    alt.X("avg_fine:Q", title="Average Level 1 Fine in Dollars"),
    alt.Y("fraction_paid:Q", title="Fraction of Time Ticket is Paid"),
    tooltip=[
        "violation_description:N",
        "tickets_issued:Q",
        alt.Tooltip("fraction_paid:Q", format=".2f"),
        alt.Tooltip("avg_fine:Q", format="$.0f")
    ],
).properties(
    width=500,
    height=200,
    title="Fraction Total Tickets are Paid and their Amounts"
)

#Headline and sub-message: The average amount of level 1 fines in dollars
 ↪ decreases as the fraction of time they are paid.

alt.Chart(tot_tick_100).mark_rect().encode(
    alt.X("avg_fine:Q", title="Average Level 1 Fine in Dollars"),
    alt.Y("fraction_paid:Q", title="Fraction of Time Ticket is Paid"),
    tooltip=[
        "violation_description:N",
        "tickets_issued:Q",
        alt.Tooltip("fraction_paid:Q", format=".2f"),
        alt.Tooltip("avg_fine:Q", format="$.0f")
    ],
).properties(
    width=500,
    height=200,
    title="Fraction Total Tickets are Paid and their Amounts"
)

#Headline and sub-message: The average amount of level 1 fines in dollars
 ↪ decreases as the fraction of time they are paid.
```



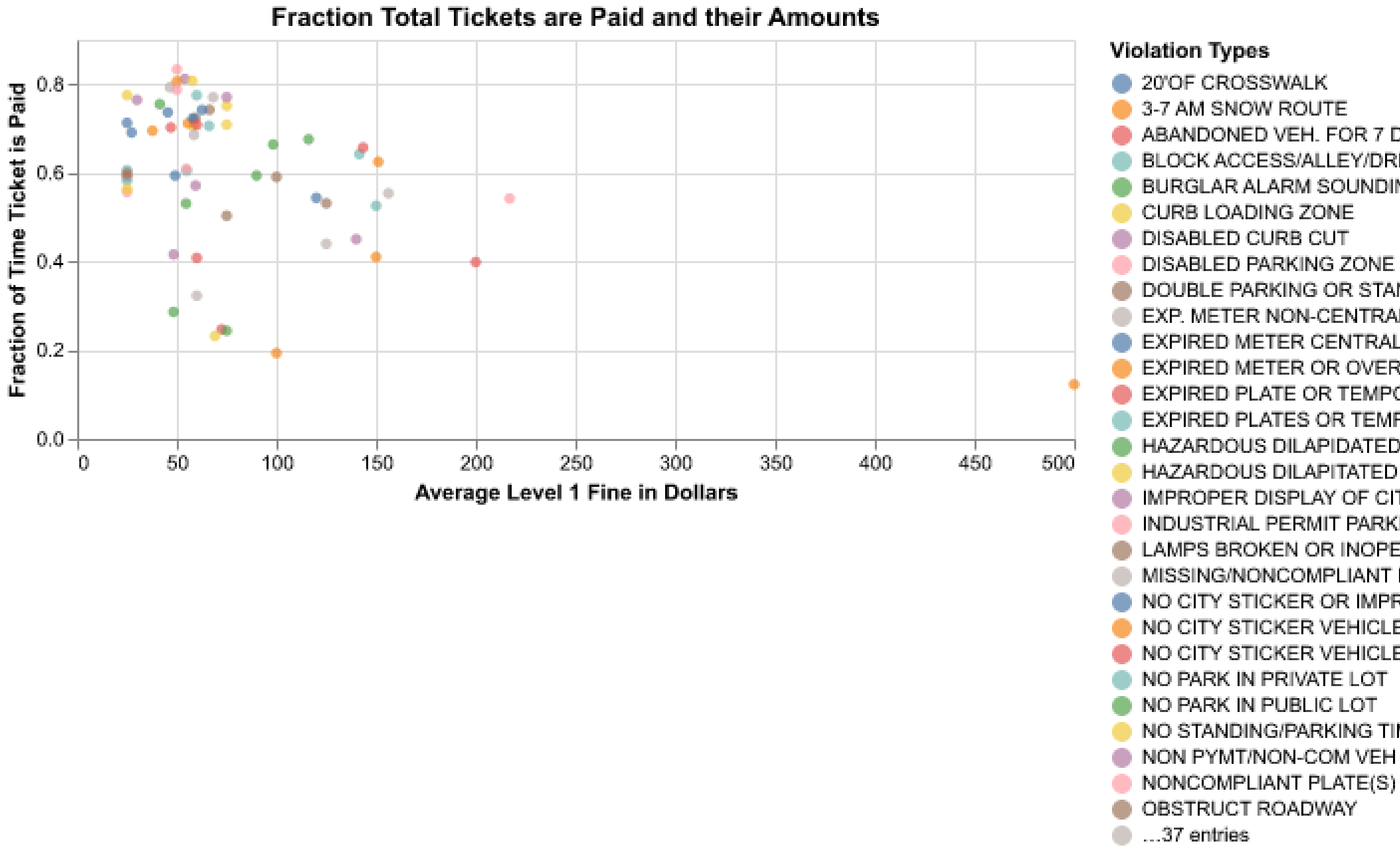
3.

#I would choose the bar chart. It has a clear delination of the average level 1 fines, and represents a comparison of their magnitudes well.

Exploration vs. Production

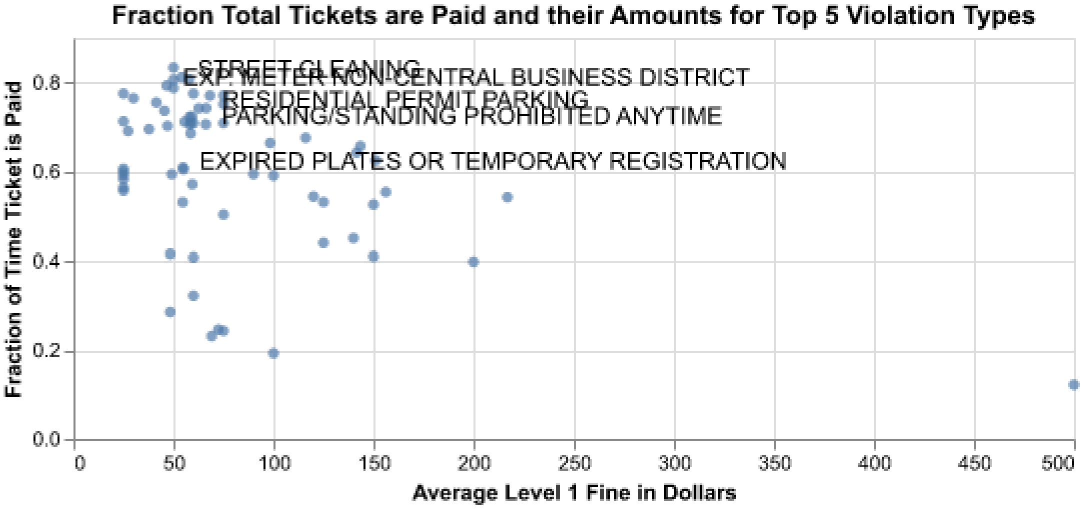
We would like to hear how you used AI (if at all) on this section. Where did you find it useful/not useful? We are asking for knowledge-sharing purposes; your answer will not impact your grade on the assignment.
Answer: I used AI to help differentiate my labels and format it properly to highlight just a few violation types.
1.

```
alt.Chart(tot_tick_100).mark_circle().encode(
    alt.X("avg_fine:Q", title="Average Level 1 Fine in Dollars"),
    alt.Y("fraction_paid:Q", title="Fraction of Time Ticket is Paid"),
    color=alt.Color(
        "violation_description:N",
        title="Violation Types"
    ),
    tooltip=[
        "violation_description:N",
        "tickets_issued:Q",
        alt.Tooltip("fraction_paid:Q", format=".2f"),
        alt.Tooltip("avg_fine:Q", format="$.0f")
    ],
).properties(
    width=500,
    height=200,
    title="Fraction Total Tickets are Paid and their Amounts"
)
```



2.

```
tot_tick_100
highlights = (tot_tick_100.sort_values("tickets_issued",
↪   ascending=False).head()["violation_description"].tolist())
tot_tick_100["highlights"] =
↪   tot_tick_100["violation_description"].isin(highlights)

point = alt.Chart(tot_tick_100).mark_circle().encode(
    alt.X("avg_fine:Q", title="Average Level 1 Fine in Dollars"),
    alt.Y("fraction_paid:Q", title="Fraction of Time Ticket is Paid"),
    tooltip=[
        "violation_description:N",
        "tickets_issued:Q",
        alt.Tooltip("fraction_paid:Q", format=".2f"),
    ]
)

label = alt.Chart(tot_tick_100[tot_tick_100["highlights"]]
).mark_text(
    align="left",
    dx=8,
    dy=-4,
    fontSize=12
).encode(
    alt.X("avg_fine:Q"),
    alt.Y("fraction_paid:Q"),
    text="violation_description:N"
)

(point + label).properties(
    width=500,
    height=200,
    title="Fraction Total Tickets are Paid and their Amounts for Top 5
 ↪  Violation Types"
)

#A useful way to reduce the clutter of labels is to reduce it to just a
 ↪  highlighted few, so that the reader can interpret the results in a
 ↪  cleaner more efficient manner.
```


Fraction Total Tickets are Paid and their Amounts for Top 5 Violation Types
STREET CLEANING
EXP. METER NON-CENTRAL BUSINESS DISTRICT
RESIDENTIAL PERMIT PARKING
PARKING/STANDING PROHIBITED ANYTIME
EXPIRED PLATES OR TEMPORARY REGISTRATION
Fraction of Time Ticket is Paid
0.0
0.2
0.4
0.6
0.8
0
50
100
150
200
250
300
350
400
450
500
Average Level 1 Fine in Dollars