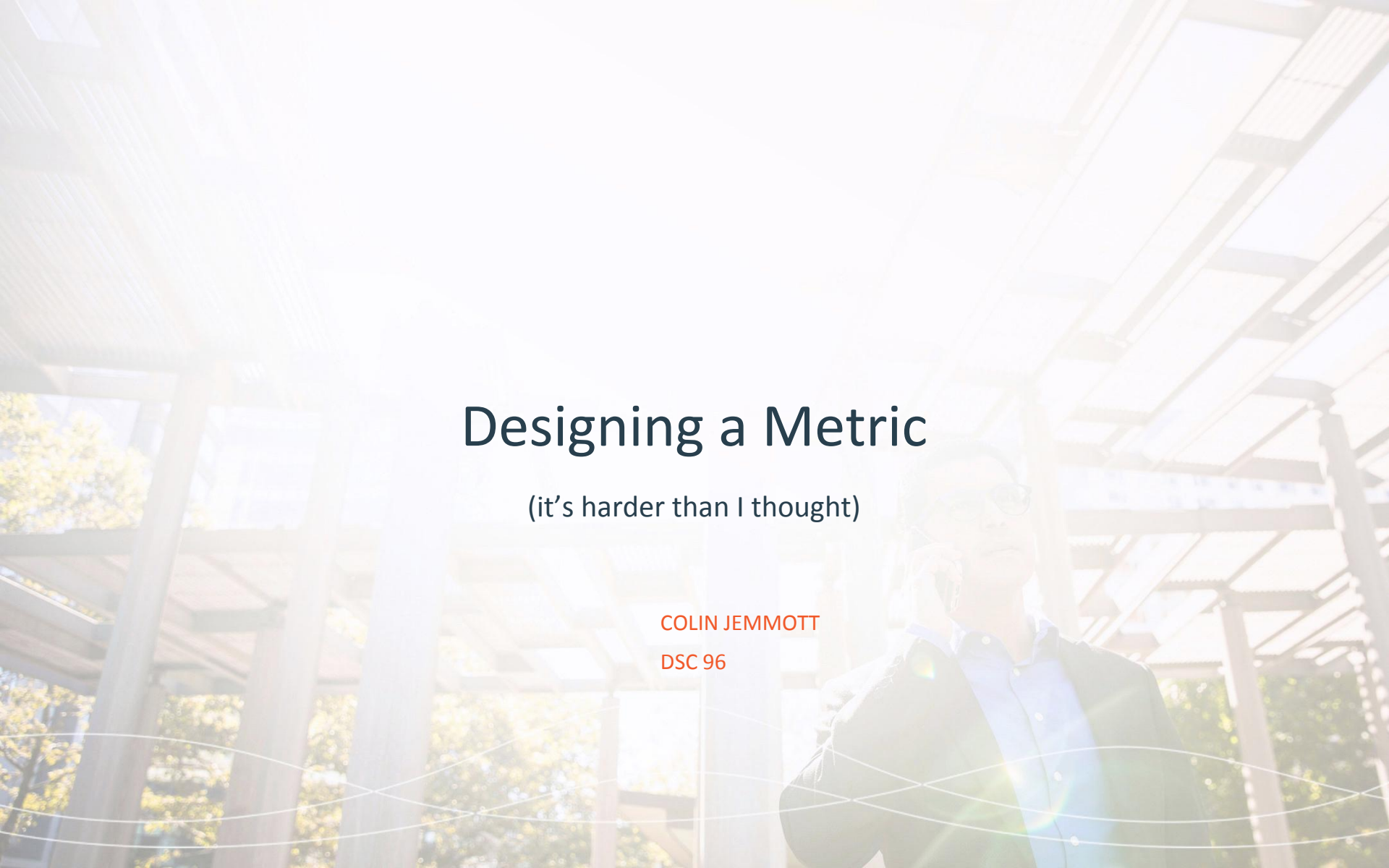


Designing a Metric

(it's harder than I thought)

COLIN JEMMOTT

DSC 96

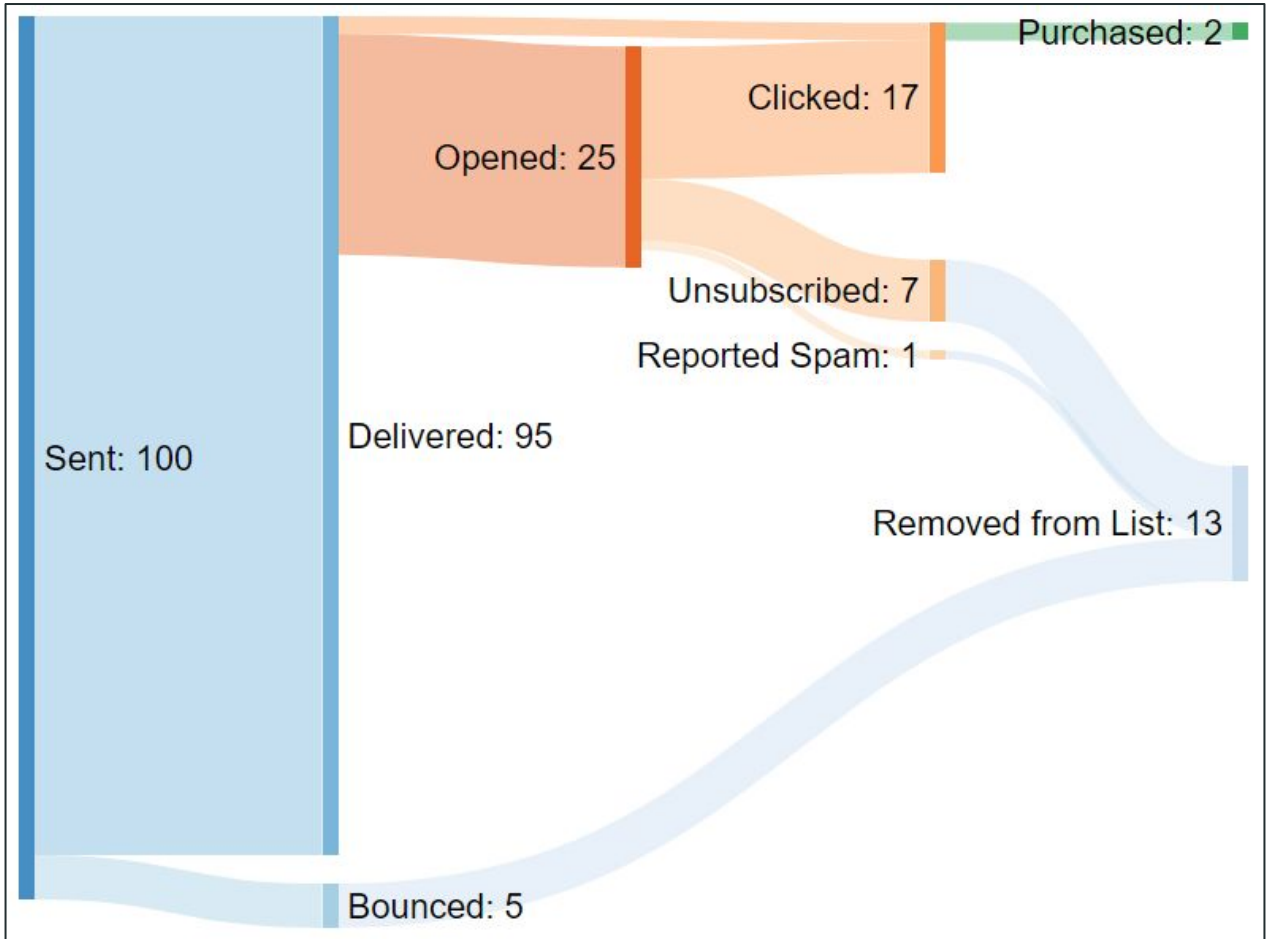
Scenario

You are a data scientist at a company that does email marketing.

The marketing team wants to know whenever “people stop engaging” with their email program.

Your task is to design a metric that can be used to send automatically notify the marketing team.

Email Flow


Purchased: 2
Clicked: 17
Opened: 25
Unsubscribed: 7
Reported Spam: 1
Sent: 100
Delivered: 95
Removed from List: 13
Bounced: 5

Data

You have access to a clean and real-time event flow!


```
{
  "actionType": "View",
  "email": "fred@llamas4u.com",
  "id": "5b86ed79-0c35-64b8-07ba-f28053e4207c",
  "campaignId": "eaeab5ae-119e-4875-bf90-35482c185fa6",
  "correlationId": "84c7fc5f-d28c-4de1-8b28-02e82cc97bf4",
  "sessionDuration": 111,
  "sessionId": "6f6aa3a8-d9fa-408f-9488-879608aeaf50",
  "sessionStartedAt": 1523927348,
  "sessionEndedAt": 1523927459,
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7
  "ipAddress": "7.6.1.4",
  "latitude": 38.5839,
  "longitude": -5.7455,
  "occuredAt": 1523927459,
  "writtenAt": 1523938893
}
```

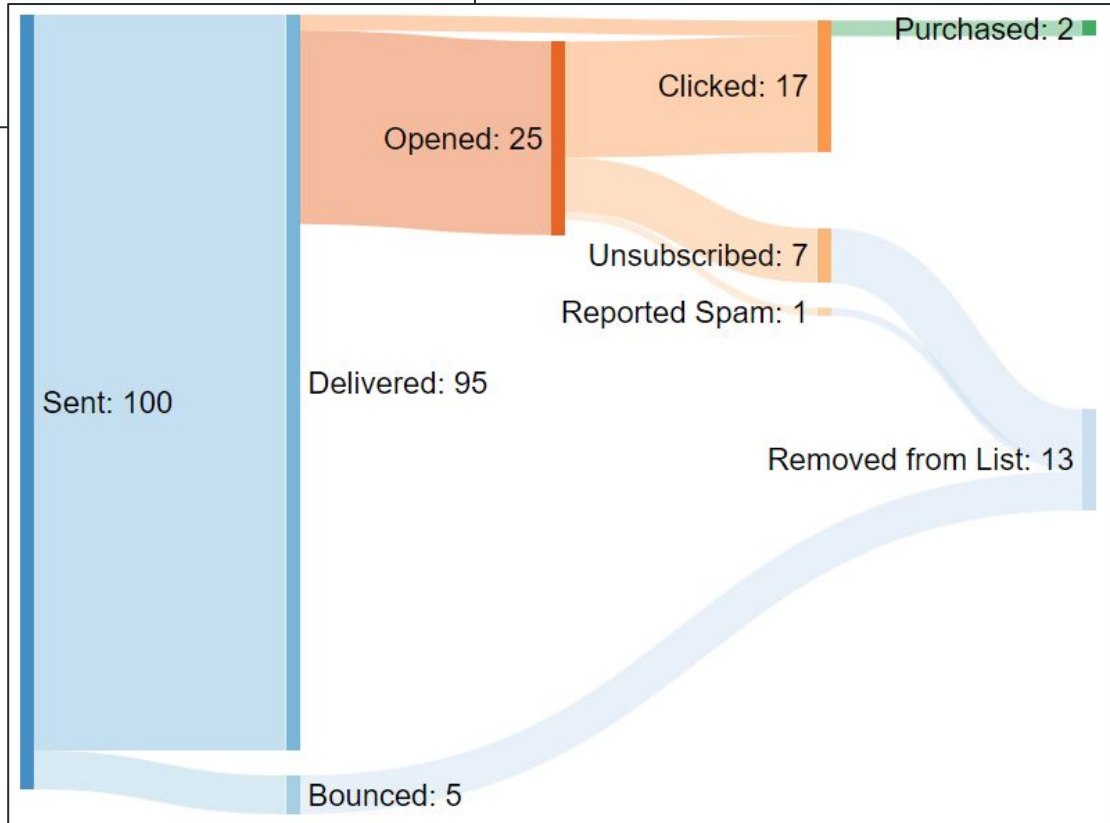
DESIGN EXERCISE

Task


```
{
  "actionType": "View",
  "email": "fred@llamas4u.com",
  "id": "5b86ed79-0c35-64b8-07ba-f28053e4207c",
  "campaignId": "eaeab5ae-119e-4875-bf90-35482c185fa6",
  "correlationId": "84c7fc5f-d28c-4de1-8b28-02e82cc97bf4",
  "sessionDuration": 111,
  "sessionId": "6f6aa3a8-d9fa-408f-9488-879608aeaf50",
  "sessionStartedAt": 1523927348,
  "sessionEndedAt": 1523927459,
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7
  "ipAddress": "7.6.1.4",
  "latitude": 38.5839,
  "longitude": -5.7455,
  "occuredAt": 1523927459,
  "writtenAt": 1523938893
}
```


Take five minutes to describe a metric that can be used to automatically notify the marketing team that user engagement is down.

Math, pseudo-code, flowchart, etc. are all ok.

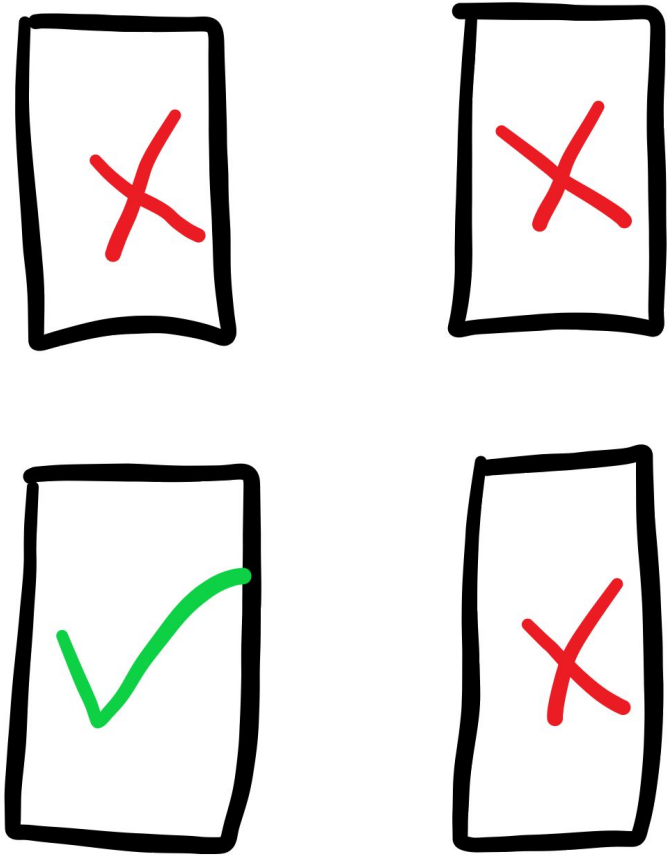


My Terrible Solution

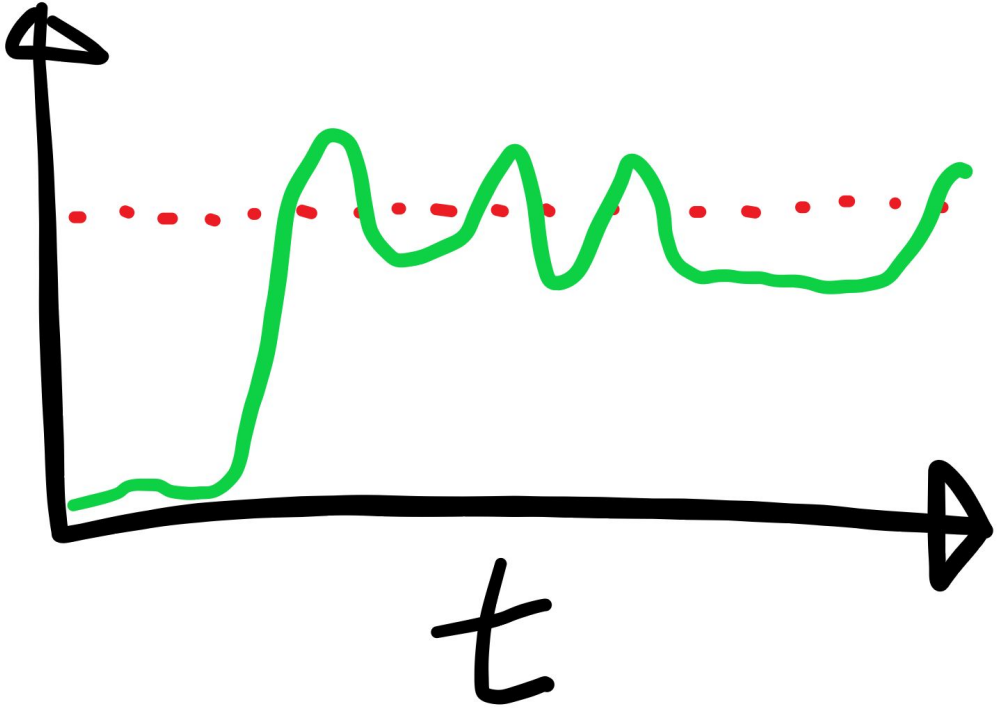
Run hourly:

```
# Count emails sent in last 24 hours
emailCount = COUNT(event) WHERE
          event["actionType"] == "Sent" AND
          event["occuredAt"] < ( TODAY()-1 )

# Count click events in the last 24 hours
clickCount = COUNT(event) WHERE
          event["actionType"] == "Click" AND
          event["occuredAt"] < ( TODAY()-1 )

# Calculate the click rate
clickRate = clickCount / emailCount

# Compare to threshold
threshold = 0.17
IF clickRate > threshold:
     alertState = True
```

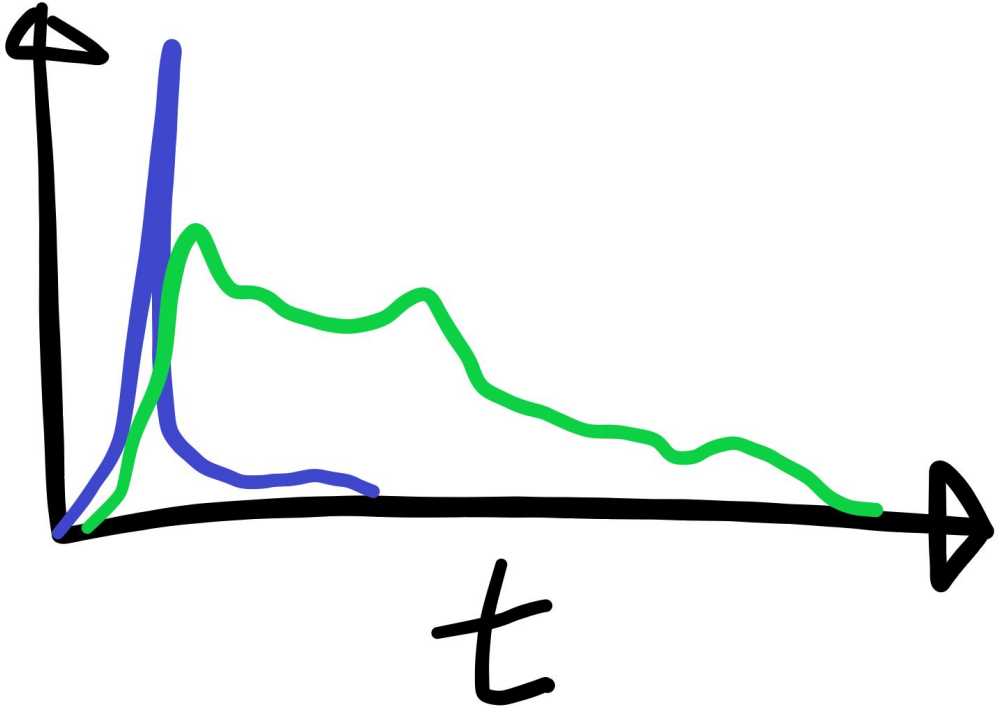
Dividing by Small Numbers is Noisy



Frequent Threshold Crossing



Clicks are Delayed



Start with the question

Business question:

-identify common scenarios (not rare cases) that have significant business impact

Math question:

-Robust, explainable, aligns with intuition

Even “simple” metrics are hard

click rate != clicks / emails

-Assign clicks to send time

-Unique versus total clicks

-Denominator is sends, deliveries, or opens?

-What time window? (hourly buckets allow flexibility)

-“expected final” click rate versus current

What percentage of delivered emails sent during the 12:00 hour today do we expect to be clicked at least once?

From Metrics to Scores

Wait, why not notify from our metric? That's what matters, right?

A score gives you:

-Normalization

-Robustness

-Anomaly detection

Good scoring answers the question "how surprising is this?“

Personal preference: 1 is “typical”, higher is “bad”

My first score was bad too

Score: 30 day average click rate / 24 hour click rate

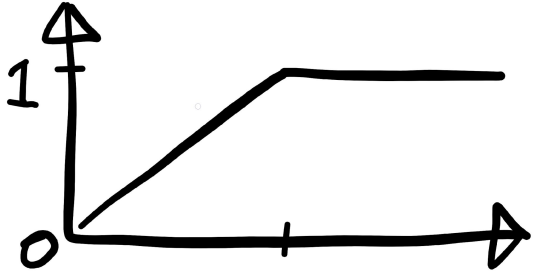
Why did this fail?

A Better Score

~~**Score**: 30 day average click rate / 24 hour click rate~~

Why did this fail?

Score: volume penalty * 30 day average click rate / 24 hour click rate



Volume penalty = max(today's volume / typical volume, 1)

Conclusion

Consumers of data science products are making data-driven decisions.

If a data consumer is mislead:

- They may make important business or life decisions that are based on falsehoods
- They may quickly lose trust that you may not be able to recover

To maintain this:

- Never knowingly ship bad data or analysis
- Acknowledge and quickly fix mistakes that are reported
- Check in with users to make sure they actually understand what is being presented

IN-CLASS EXERCISE

SDPD Collision Data

Import pd_collisions_datasd.csv

Explore the question “ is the population that is crashing cars different from the population getting pulled over?”