Data Scientist
A/B Testing & Statistical Experiment Presentation

By: Jeff Tu

Agenda

1. Objectives
2. Executive Summary
3. Background
4. Recommendation
5. Analyses
6. Further Explorations
7. Appendix

Objectives

Design Ride Cancellation Fee Policy to maximize **long term financial benefit & system health** while maintain healthy utilizations from **both riders and drivers**

Maximize

Long Term Ride Revenue from both **Successful** and **Cancelled** Rides

Minimize

Rider and Driver churn / bad experiences

Monitor

Check on potential issues on

1. Supply & Demand
2. Ride Matching algorithm
3. ETA calculation algorithm

Executive Summary for Cancellation Penalty Policy

Cancellation Fee = <u>Base Penalty Fee</u> x (1 + <u>Cancellation Fee Multiplier (CFM)</u>)

Part 1: **<u>Base Penalty Fee</u>** = $5.0

Part 2: **<u>CFM</u>** = c1*w1 + c2*w2 … + c7*w7

CFM Rider Cancellation Criteria (Yes = 1, No = 0)	Weights*
c1. Cancelled after matched?	w1: +2%
c2. Late night rides?	w2: +2%
c3. Peak hours rides?	w3: +2%
c4. By high cancellation rider?	w4: +2%
c5. Driver picky frequently?	w5: +2%
c6. Cancel mistake & rider with low mistaken cancel history?	w6: -2%
c7. Frequent rider?	w7: -2%

* Assuming equal weights, further investigation for assigning weights

<u>Business Impact:</u>

1. Increase average ride revenues per rider
2. Maintain good driver rider supply demand
3. Reduce troubleshooting cost of engineering and system issues

Data Background

Date: 2019-04-14 to 2019-05-26

Duration: 42 days

Location: LA, US

Number of unique riders = 529,084

Number of ride requested = 1,397,335

Group	Cancel Penalty	Rider Count
Control	$ 5.0	176,856
Treatment 2	$ 3.0	177,000
Treatment 1	$ 1.0	176,900

Ride Data

1. **ride_id -** Unique identifier for the ride request.
2. **rider_id -** Unique identifier for the rider who requested the ride.
3. **driver_id -** Unique identifier for the driver.
4. **ride_type** - Type of ride requested (shared, normal).
5. **upfront_fare** - Final fare quote provided to the Rider before the request was made. This is surfaced to the rider after they enter both an origin and destination in the Company app.
6. **rider_paid_amount** - Total amount of money the rider paid to the Company.
7. **eta_to_rider_pre_match** - ETA (estimated time to arrival) shown to the rider immediately before the ride request was made.
8. **eta_to_rider_post_match** - ETA shown to the rider immediately after the ride request was matched to a specific driver.
9. **requested_at_local** - Time when the ride was requested.
10. **accepted_at_local** - Time when the driver accepted the ride request.
11. **arrived_at_local** - Time when the driver arrived at the pickup location.
12. **picked_up_at_local** - Time when the rider was picked up from the pickup location.
13. **dropped_off_at_local** - Time when the rider was dropped off.
14. **actual_time_to_arrival** - Time (in seconds) for the driver to reach the designated pickup location after being matched with the ride request.
15. **cancellation_flag** - Boolean flag for whether the ride was canceled.
16. **rider_request_number** - Sequential count of ride requests for each rider.

Experiment Data

The company recently launched a randomized experiment to test the effect of charging riders cancellation fees, of varying amounts, if they cancel a ride request. Riders were assigned to each variant and informed that the new cancellation fee would apply to all future rides. This experiment was in effect for the entire duration of the Ride Request Dataset.

1. **rider_id** - unique identifier for a Rider.
2. **variant** - experiment group the Rider was in.
3. **cancel_penalty** - cancellation penalty fee for the variant.

Recommendation

Cancellation Fee = <u>Base Penalty Fee</u> x (1 + <u>Cancellation Fee Multiplier</u>)

1) Base Penalty Fee is a **<u>constant</u>** penalty fee to all rides

2) Cancellation Fee Multiplier (CFM) considers **<u>factors</u>** affecting ride cancellation in a **<u>long run</u>**

Recommendation Analysis - Part 1
Base Penalty Fee

Cancellation Fee = **Base Penalty Fee** x (1 + Cancellation Fee Multiplier)

Success Metrics

1. Rider Average Cancellation Rate (Lower, Better)

= cancellation count per rider / total ride count per rider

2. Average Ride Revenue per Rider (Higher, Better)

= total paid amount per rider / total ride count per rider

Guardrail Metrics

3. Average Request Count per Rider (Higher / Stable, Better)

= ride request count per rider / total unique rider count

Recommendation Analysis - Part 1
Base Penalty Fee

3 groups of Penalty Fees

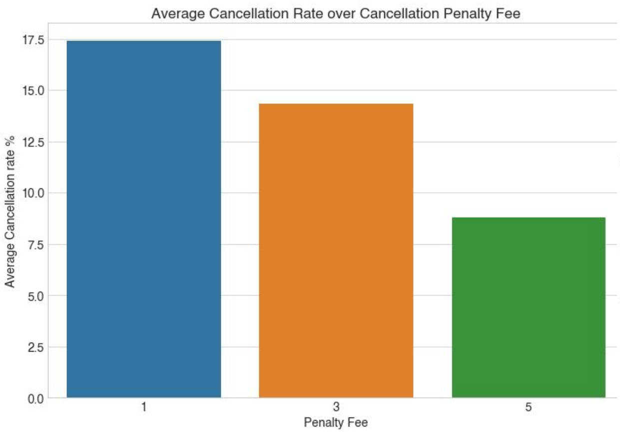
Group	Cancel Penalty
Control	$ 5.0
Treatment 2	$ 3.0
Treatment 1	$ 1.0

Compare over 3 Metrics

Metrics
Average Cancellation Rate per Rider
Average Ride Revenue per Rider
Average Request Count per Rider

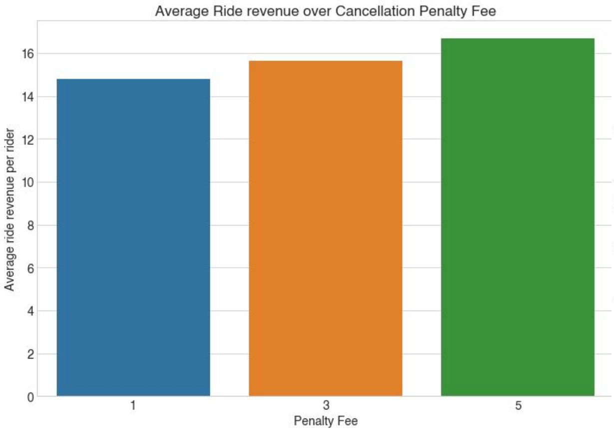
Exploratory Data Analysis over Metrics among Groups

Average Cancellation Rate



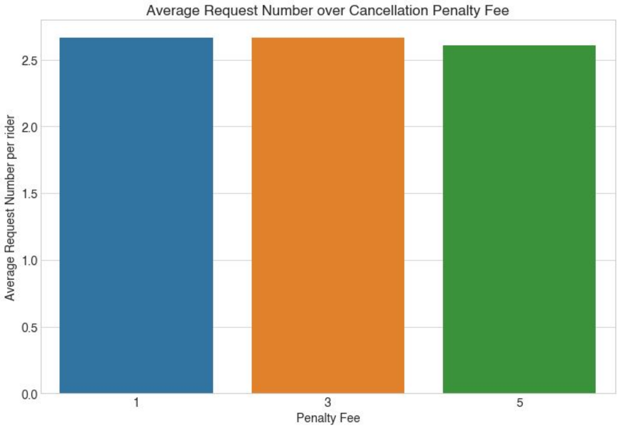
- **Negative correlation** between average rider cancellation rate when penalty fee increase.
- Almost double the drop from **17.5% to ~9%**

Average Ride Revenue per Rider



- **Slightly positive correlation** to Average Ride Revenue with penalty fee.
- Increase from **$15 to $17**

Average Rider Request Count



- **Little to no obvious strong correlation** between Average Ride request per rider with penalty fee.

Statistical Tests for 3 Independent groups

Goal:

- To statistically test the **means / medians difference** of the 3 metrics among 3 cancel penalty groups

Observation of data:

- Riders with different cancel penalties are **independent** to each other*
- **Normally Distributed** with large sample sizes^
- **Variances** among groups and metrics are **different** (Even after log transformation, see Appendix for Levene's Test results)

*Although risk of information spillover among groups

^Since sample size is huge, we can assume the sample means are normally distributed through Central Limit Theorem

Non-Parametric Statistical Tests

1. Kruskal-Wallis H-test (Alpha = 0.05)

H0 : There are no statistical significant difference in medians
Reject H0 when p-value < alpha

2. Pairwise Post Hoc Dunn’s Test (Alpha = 0.05, Bonferroni adjusted)

H0 : There are no statistical significant difference in medians
Reject H0 when p-value < alpha

Statistical Analysis - Kruskal-Wallis H-test (Alpha = 0.05)

```
Kruskal-Wallis H-test - Assume variables indepenent
Statistics=4238.919, p=0.000
cancellation_rate has different distributions (reject H0) among different penalty group
```

```
Statistics=3826.589, p=0.000
rider_paid_amount has different distributions (reject H0) among different penalty group
```

```
Statistics=4.636, p=0.098
rider_request_number has same distributions (fail to reject H0) among different penalty group
```

Kruskal-Wallis H-test, Alpha = 0.05

- H0 : There are no statistical significant difference
- Reject H0 when **p-value < alpha (0.05)**

Conclusion:

- There are **statistically significant difference** for **cancellation rate** and **average ride revenues**
- **No** statistically significant difference for **average ride request per rider**

Statistical Analysis - Post hoc pairwise Dunn’s Test

Dunn’s Test (Adjusted alpha = 0.05)
H0: means have no statistical significant difference

- P-values are adjusted by Bonferroni correction
- Reject H0 when p-value < alpha

Adjusted P-values are **below 0.05** for pairwise comparison shows **statistically significant difference** for both **cancellation rate** and **average ride revenue per rider** for **all penalty groups**

Adjusted P-values are **above 0.05** shows **no statistically significant difference** for **average ride request per rider** for **all penalty groups**

Cancellation Rate

Adjusted P-value < 0.05
cancellation rate

	1.0	3.0	5.0
1.0	False	True	True
3.0	True	False	True
5.0	True	True	False

Average Ride revenue per rider

Adjusted P-value < 0.05
rider paid amount

	1.0	3.0	5.0
1.0	False	True	True
3.0	True	False	True
5.0	True	True	False

Average Ride Request per rider

Adjusted P-value < 0.05
rider request number

	1.0	3.0	5.0
1.0	False	False	False
3.0	False	False	False
5.0	False	False	False

Conclusion - Base Penalty Fee

We will set the **base penalty fee to $5.0**

Business Implications:

Comparing all 3 penalty groups, **higher** Base Penalty Fee **($5.0)**:

1. Discouraged riders to cancel
2. Generated higher revenue on average
3. Didn’t cause significant drop in ride request count

Recommendation Analysis - Part 2
Cancellation Fee Multiplier (CFM)

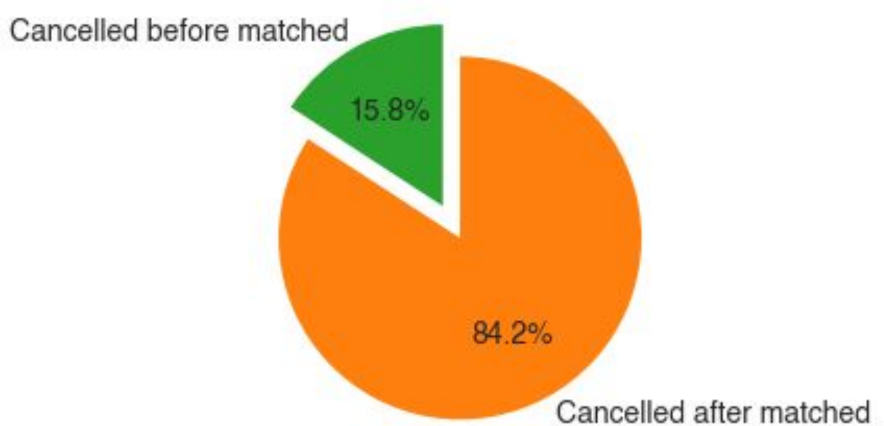
Cancellation Fee = Base Penalty Fee * (1 + Cancellation Fee Multiplier)

Cancellation Fee Multiplier (CFM) considers **factors** affecting ride cancellation in a **long run**

Goal: Penalize harmful scenarios, Benefit high LTV riders

CFM Rider Cancellation Criteria (Yes = 1, No = 0)
c1. Cancelled after matched?
c2. Late night rides?
c3. Peak hours rides?
c4. By high cancellation rider?
c5. Driver picky frequently?
c6. Cancel mistake & rider with low mistaken cancel history?
c7. Frequent rider?

Cancellation Stage - Before or after matched



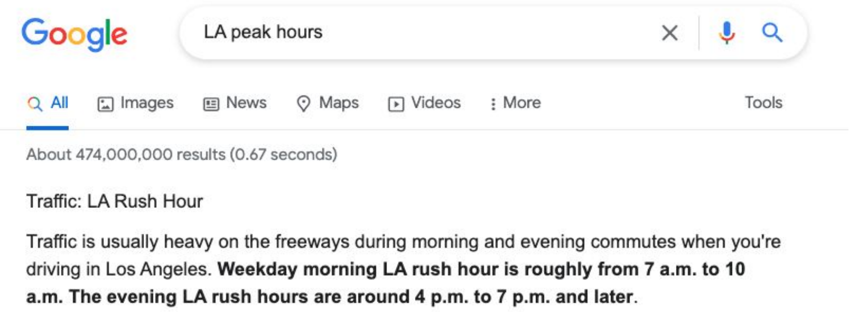
Cancellation by riders **after** matched with drivers are **more harmful**.

- **Negatively** Impact driver's experience
- It **reduces the available drivers** for **other potential riders** (when a driver accepted a ride, other potential rider has 1 less driver available to them)

Higher cancel penalty for cancellation **after** matched with drivers

Peak hour rides

- LA Peak hours = 7-10 AM & 4-7PM
- Peak hours have higher demand



Late night rides

- Late night rides = 11PM - 6AM
- Late nights have less driver supply

Higher cancel penalty during **peak hours or late night hours** to maintain good supply for riders who actually use a ride

High Cancellation Rate Rider

- Riders who have **50% or above** average cancellation rate are regarded as bad rider
- Only include riders with **at least 3 ride** requests (within 42 days experiment period)

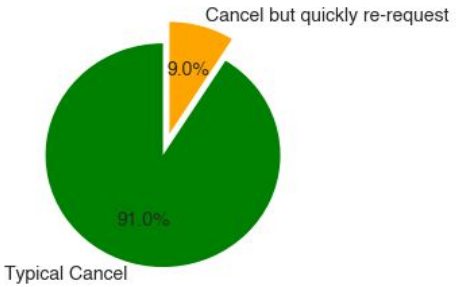
Further investigation:

- More data to find out rider historical behaviors

Higher penalty fee on **high cancellation rate** riders

Cancellation Mistake Or Driver Picky riders

Identify those that were **cancelled but quickly requested** the next actual ride **within 5 min**



Possibility 1:
Rides that were cancelled and re-requested shortly could be a **mistaken cancellation**

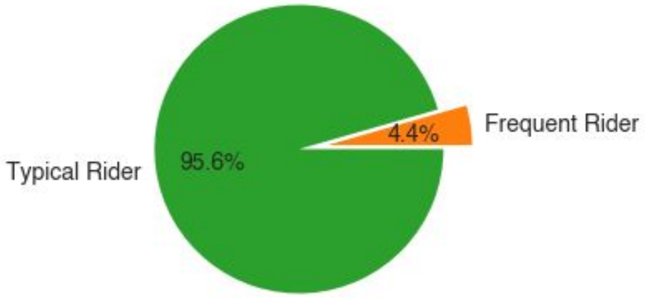
Possibility 2:
Those who've done that **consistently** could be too **"driver picky"***

- **Lower** cancel penalty for **mistaken cancellation**
- **Higher** cancel penalty for **frequent driver-picky rider**

*Assuming riders know some information about the driver including vehicle model and driver rating, so they intentionally chose one driver over another
Need to investigate driver data

Frequent Rider

Frequent riders are those who at least successfully **completed 10 rides** over 42 days AND **at least 2 successful** rides per **week**



Lower penalty fee to Frequent Rider

- Frequent riders have **high expected LTV**, we want to keep them

Conclusion - Cancellation Fee Multiplier (CFM)

CFM Rider Cancellation Criteria (Yes = 1, No = 0)	**Weights (%)***
c1. Cancelled after matched?	w1: +2%
c2. Late night rides?	w2: +2%
c3. Peak hours rides?	w3: +2%
c4. By high cancellation rider?	w4: +2%
c5. Driver picky frequently?	w5: +2%
c6. Cancel mistake & rider with low mistaken cancel history?	w6: -2%
c7. Frequent rider?	w7: -2%

$$CFM = c1*w1 + c2*w2 \ldots + c7*w7$$

* Assuming equal weights, further investigation for assigning weights

Monitoring metrics for Market & System Health Check

Ride Request and Available Driver ratio

- Monitor for **Oversupply or Undersupply of drivers**

Accepted and Requested Gap

- **Large time gap** between accepted at and requested at time
- **undersupply** or **matching algorithm** issues

Post-Pre matched ETA Gap

- **Large time gap** between the pre-matched and post-matched ETA gap **continuously**
- **Matching and ETA** calculating **algorithm** issues

Maintaining good Market and System Health could prevent potential cancellations

Recommendation

Cancellation Fee = $5 x (1 + Cancellation Fee Multiplier)

- Base Penalty Fee $5.0 is a **constant** penalty fee to **all** rides

- Cancellation Fee Multiplier (CFM) attempts to consider **long term cost and benefits** to adjust ride cancellation

  *CFM = c1*w1 + c2*w2 … + c7*w7*

- Monitor
  1. Ride Request & Available Driver ratio
  2. Accepted Requested gap
  3. Post & Pre-matched ETA gap
  for market and algorithm health check

CFM Rider Cancellation Checklist (Yes = 1, No = 0)	Weights*
c1. Cancelled after matched?	w1: +2%
c2. Late night rides?	w2: +2%
c3. Peak hours rides?	w3: +2%
c4. By high cancellation rider?	w4: +2%
c5. Driver picky frequently?	w5: +2%
c6. Cancel mistake & rider with low mistaken cancel history?	w6: -2%
c7. Frequent rider?	w7: -2%

* Assuming equal weights, further investigation for assigning weights

Further Investigation and Potential Improvement

Experiment Approach

- **Prevent Network Overspill and sharing supply** by **split in similar city** (e.g. New York)
- Have experiment of **$0 penalty fee as another group** and compare the 0 to $1 behaviors
- **Pre-Post data analysis** to compare the the metric difference **before and after** the experiment **within** penalty group

Rider Segments & CFM

- Utilize Clustering ML algorithms to find interesting segmentations
- **Adjust CFM weightings**
  - More historical data to analyse rider behaviors
  - Real-time or Batch Update
    - Engineering Cost and Benefit Tradeoff

Features & External factors

- Features e.g. **cancel fee notification** message pop up might **discourage cancellation**
- Analyze **driver data** to see a broader impact of cancellation fee to our supply
  - Driver Online / Offline Behaviors
  - Driver Ratings
- Longer Ride data period for **seasonality** and **special event** analysis
- Benchmark with **competitors** (not just Uber but other transportation means)

Further Investigation and Potential Improvement (cont.)

Other Business Priority

- **Reduce / Remove** the cancellation fee if the goal is to **maximize rider acquisition**
    - E.g. Entering a **new market**

Thank You

Appendix

1-Way ANOVA Test (Code)

```python
def anova_test(df, label, sampled=False):
    import scipy.stats as stats
    fvalue, pvalue = stats.f_oneway(
                                    df[label][df['cancel_penalty']==5],
                                    df[label][df['cancel_penalty'] ==3],
                                    df[label][df['cancel_penalty'] ==1]
                                    )
    if sampled:
        print(f"Sampled - Comparing {label} mean values among different cancel penalties")
    else:
        print(f"Comparing {label} mean values among different cancel penalties")
    print(f"f value: {fvalue}")
    print(f"p value: {pvalue}")
```

Hypothesis Test for 3 independent groups

```
Comparing cancellation_rate mean values among different cancel penalties
f value: 3625.225772240985
p value: 0.0
```

```
Comparing rider_paid_amount mean values among different cancel penalties
f value: 893.2333787450867
p value: 0.0
```

```
[redacted] Comparing rider_request_number mean values among different cancel penalties
f value: 8.13244905811076
p value: 0.00029457166677791677
```

One way - ANOVA test, Alpha = 0.05

- H0 : There are no statistical significant difference
- Reject H0 when p-value < alpha

All Cancellation Rates, Average Ride Revenue, and Ride request number have **statistically significant difference**

P-values < 0.05

Not exactly what we observed, we then test with non-parametric test

Statistical Analysis - Tukey's Test

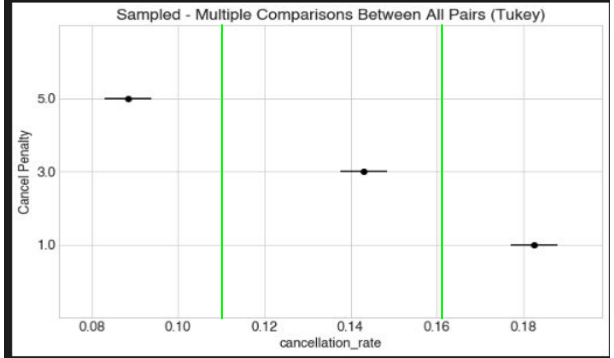
No overlapping of confidence interval (C.I.) among 3 penalty groups shows **statistically significant difference** for **both cancellation rate and average ride revenue per rider**

Some overlapping of confidence interval (C.I.) shows **no statistically significant difference for average ride request per rider**

Cancellation Rate

Multiple Comparison of Means - Tukey HSD, FWER=0.05

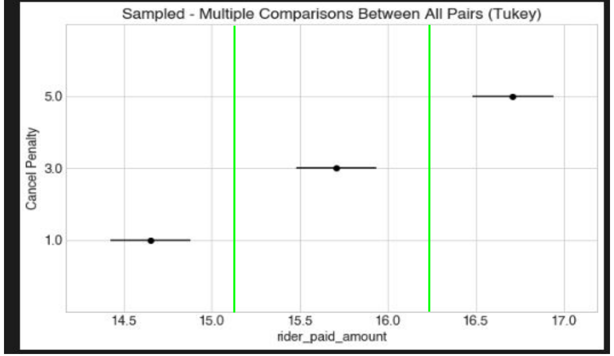
group1	group2	meandiff	p-adj	lower	upper	reject
1.0	3.0	-0.0395	0.001	-0.0503	-0.0287	True
1.0	5.0	-0.0941	0.001	-0.105	-0.0833	True
3.0	5.0	-0.0546	0.001	-0.0655	-0.0438	True



Average Ride revenue per rider

Multiple Comparison of Means - Tukey HSD, FWER=0.05

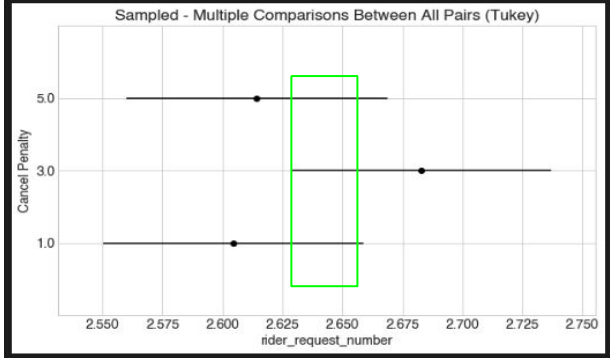
group1	group2	meandiff	p-adj	lower	upper	reject
1.0	3.0	1.0562	0.001	0.5988	1.5136	True
1.0	5.0	2.058	0.001	1.5991	2.5168	True
3.0	5.0	1.0018	0.001	0.5433	1.4603	True



Average Ride Request per rider

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
1.0	3.0	0.0782	0.2089	-0.0302	0.1867	False
1.0	5.0	0.0099	0.9	-0.0989	0.1187	False
3.0	5.0	-0.0683	0.3044	-0.177	0.0404	False



Tukey’s Test (Code)

```
def tukey_test(df, label, sampled=False):
    from statsmodels.stats.multicomp import pairwise_tukeyhsd
    tukey = pairwise_tukeyhsd(endog=df[label],
                              groups=df['cancel_penalty'],
                              alpha=0.05)
    print(tukey)
    tukey.plot_simultaneous(ylabel= "Cancel Penalty", xlabel= label)
    if sampled:
        plt.title("Sampled - Multiple Comparisons Between All Pairs (Tukey)")
```

Variance Test - Levene's Test (Code) (H0= same variances)

Levene's test without log transformation
all 3 metrics have p-value < 0.05
=> They don't have similar variances

```
# cancellation_rate
from scipy.stats import levene
label = 'cancellation_rate'

stat, p = levene(df_rider_1[label], df_rider_3[label], df_rider_5[label])
print(f"Levene's test for {label}'s p value = {p}")
```

✓ 0.7s

```
Levene's test for cancellation_rate's p value = 0.0
```

```
# rider_paid_amount

label = 'rider_paid_amount'

stat, p = levene(df_rider_1[label], df_rider_3[label], df_rider_5[label])
print(f"Levene's test for {label}'s p value = {p}")
```

✓ 0.1s

```
Levene's test for rider_paid_amount's p value = 5.165590426548579e-29
```

```
# rider_request_number

label = 'rider_request_number'

stat, p = levene(df_rider_1[label], df_rider_3[label], df_rider_5[label])
print(f"Levene's test for {label}'s p value = {p}")
```

✓ 0.4s

```
Levene's test for rider_request_number's p value = 2.1415124671703738e-08
```

Levene's test after log transformation
all 3 metrics have p-value < 0.05
=> They don't have similar variances

```
label = 'cancellation_rate'

stat, p = levene(np.log(df_rider_1[label]+1), np.log(df_rider_3[label]+1), np.log(df_rider_5[label]+1))
print(f"Levene's test for {label}'s p value = {p} (log tranformed)")
```

✓ 0.8s

```
Levene's test for cancellation_rate's p value = 0.0 (log tranformed)
```

```
label = 'rider_request_number'

stat, p = levene(np.log(df_rider_1[label]+1), np.log(df_rider_3[label]+1), np.log(df_rider_5[label]+1))
print(f"Levene's test for {label}'s p value = {p} (log tranformed)")
```

✓ 0.6s

```
Levene's test for rider_request_number's p value = 0.0032048132773742194 (log tranformed)
```

```
label = 'rider_paid_amount'

stat, p = levene(np.log(df_rider_1[label]+1), np.log(df_rider_3[label]+1), np.log(df_rider_5[label]+1))
print(f"Levene's test for {label}'s p value = {p} (log transformed)")
```

✓ 0.7s

```
Levene's test for rider_paid_amount's p value = 0.0 (log transformed)
```

Dunn’s Test p-values

Multiple Comparison for all pairs of penalty groups ($1, $3, $5)
Adjusted P-value < 0.05
cancellation rate

	1.0	3.0	5.0
1.0	1.000000e+00	7.732986e-111	0.0
3.0	7.732986e-111	1.000000e+00	0.0
5.0	0.000000e+00	0.000000e+00	1.0

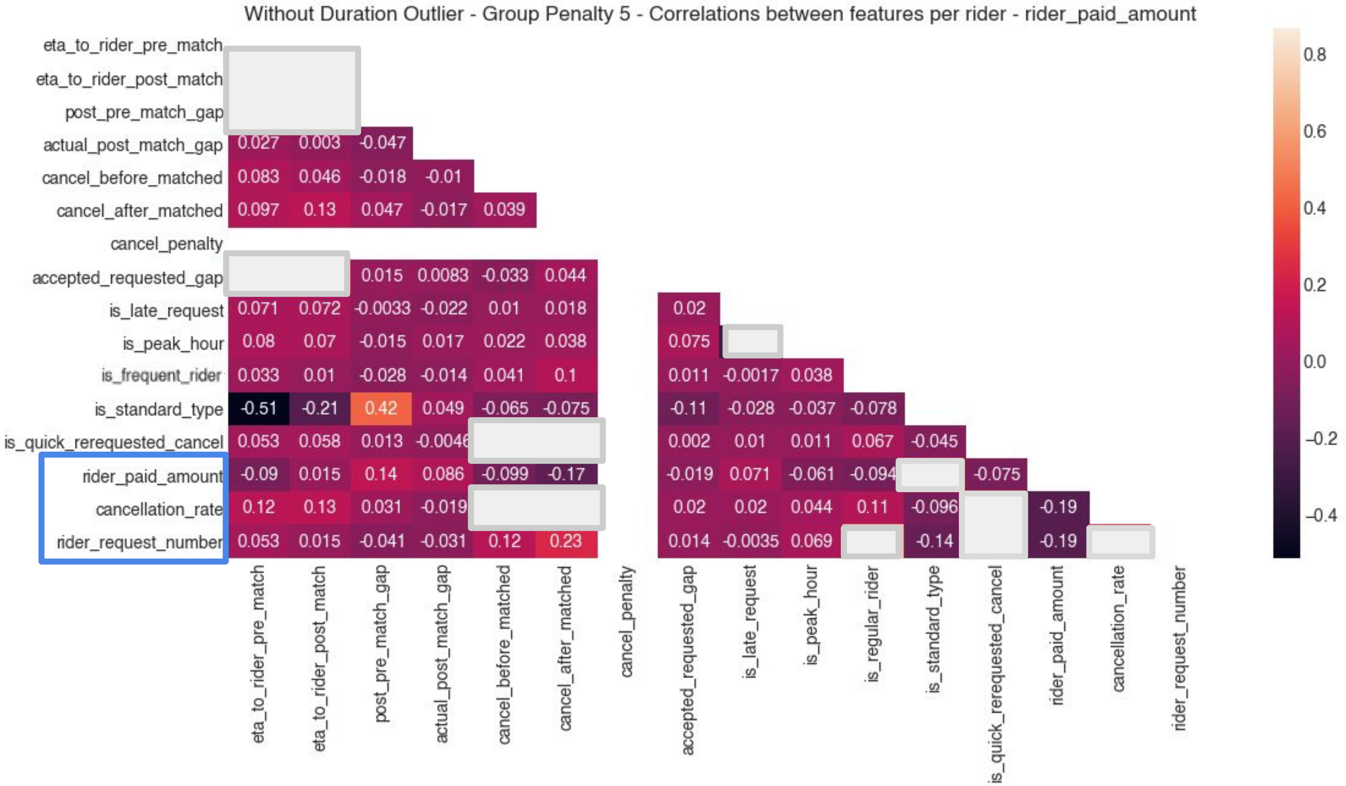
Multiple Comparison for all pairs of penalty groups ($1, $3, $5)
Adjusted P-value < 0.05
rider paid amount

	1.0	3.0	5.0
1.0	1.000000e+00	1.594119e-151	0.000000e+00
3.0	1.594119e-151	1.000000e+00	1.340138e-273
5.0	0.000000e+00	1.340138e-273	1.000000e+00

Multiple Comparison for all pairs of penalty groups ($1, $3, $5)
Adjusted P-value < 0.05
rider request number

	1.0	3.0	5.0
1.0	1.000000	1.000000	0.202117
3.0	1.000000	1.000000	0.172886
5.0	0.202117	0.172886	1.000000

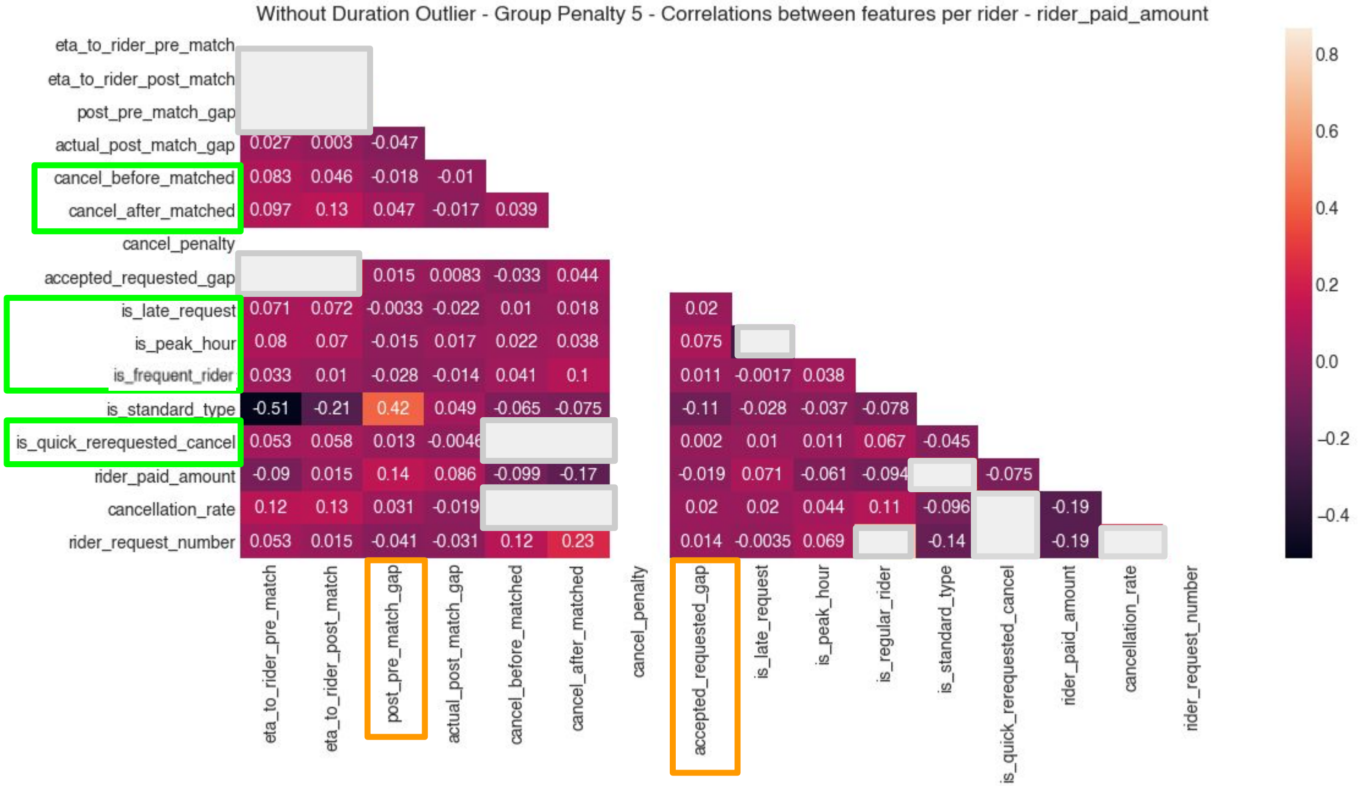
Cancellation Fee Multiplier (CFM) Rider segments analysis



From a correlation analysis among some engineered features

Other than the overfitted features (framed in grey), We didn’t see strong feature correlations with our target metrics (in blue)

Cancellation Fee Multiplier (CFM)
Rider segments analysis



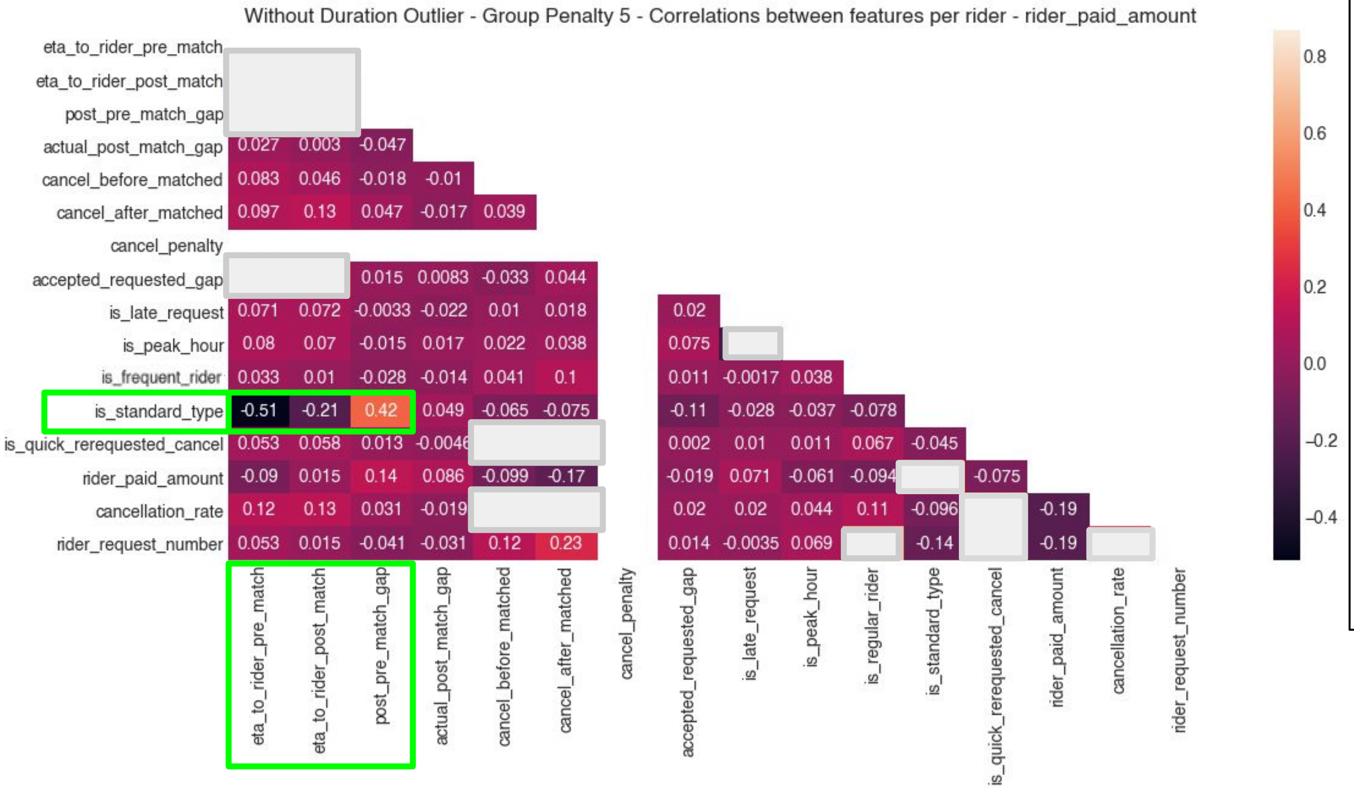
Green:

- Subtle / longer term effects, and consider them in the CFM.

Orange:

- Not included in CFM but monitor for health check

Cancellation Fee Multiplier (CFM)
Rider segments analysis



<u>Observation:</u>

- **<u>ETA</u>** seems to have correlations with whether the ride is **<u>standard or shared</u>**
- Shared type ride have a longer ETA and more variances of ETA for driver have to pick up multiple riders and thus have more potential issues along the way

Normality Test - Shapiro-Wilk test

Shapiro-Wilk test

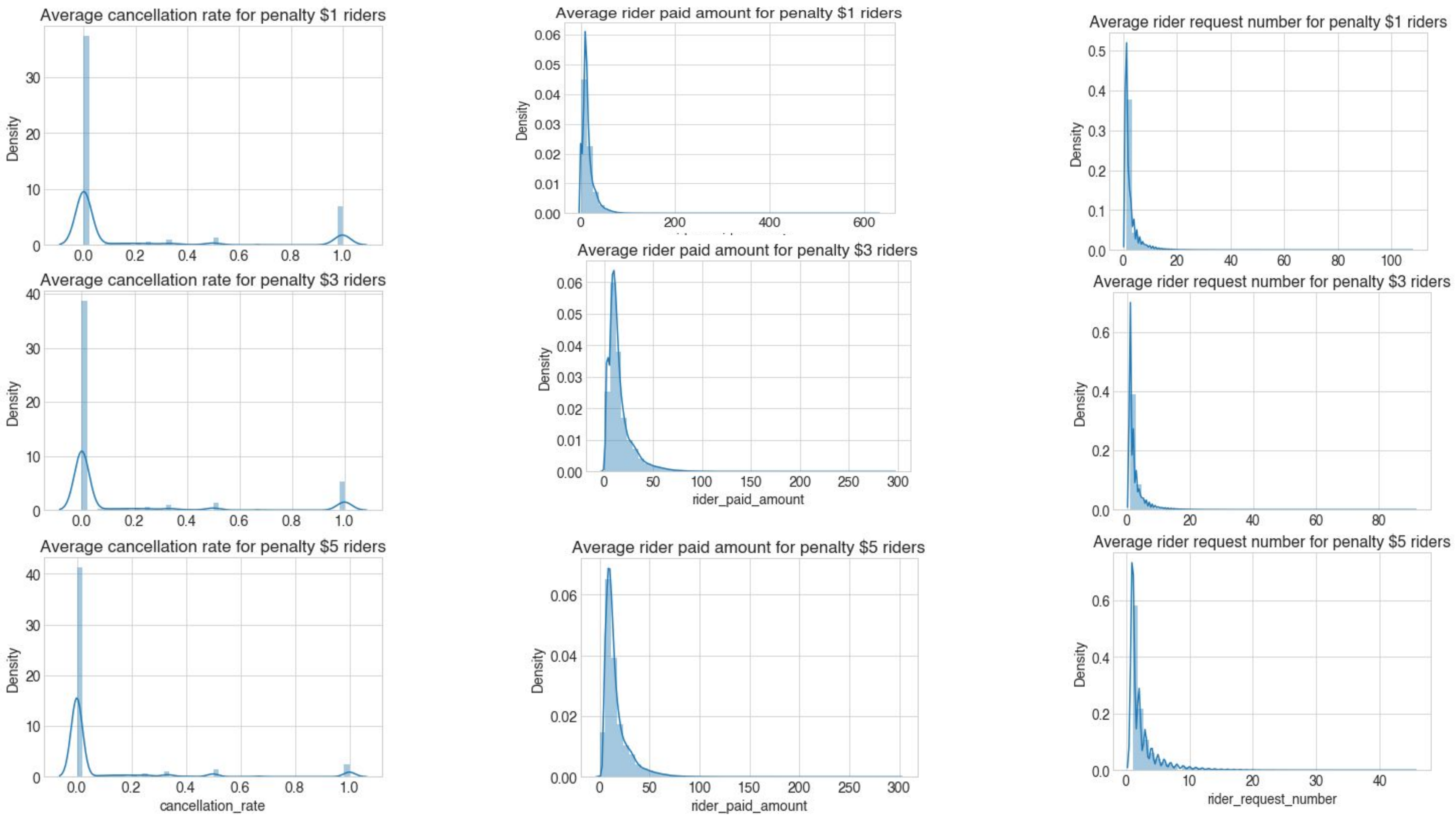
```python
import scipy.stats as stats

w_test_statistic, p_value = stats.shapiro(model.resid)
print(f"W Test Statistics = {w_test_statistic}\nP value = {p_value}")
```

[191]

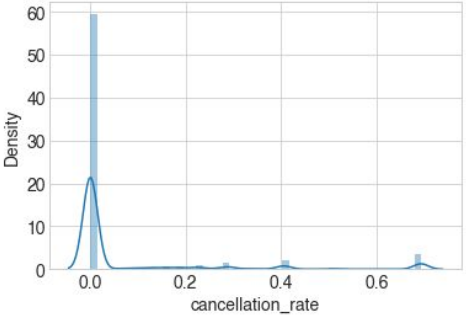
```
W Test Statistics = 0.5717940926551819
P value = 0.0
```

Distribution of metrics

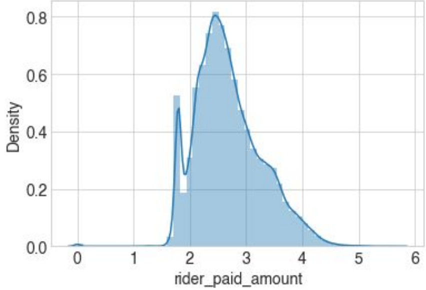


Distribution of metrics - Log Transformation

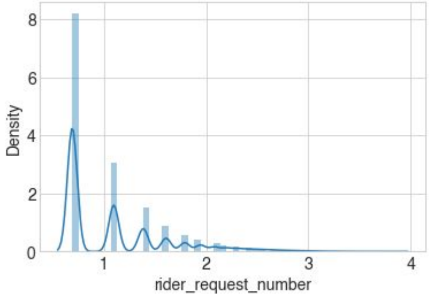
Average cancellation rate for penalty $5 riders - log transformed



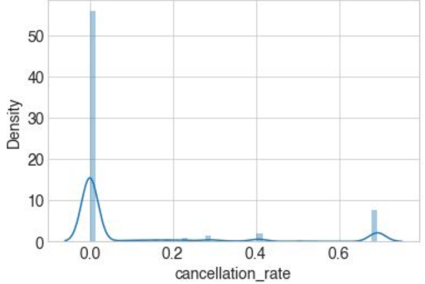
Average rider paid amount for penalty $5 riders - log transformed



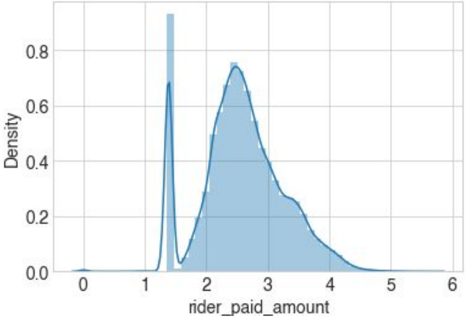
Average rider request number for penalty $5 riders - log transformed



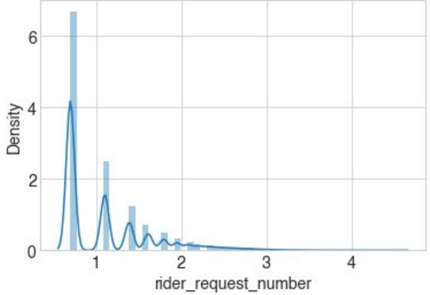
Average cancellation rate for penalty $3 riders - log transformed



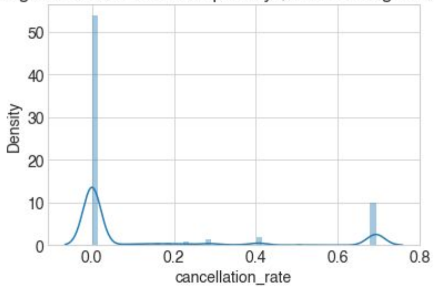
Average rider paid amount for penalty $3 riders - log transformed



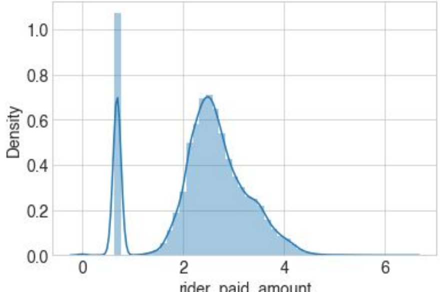
Average rider request number for penalty $3 riders - log transformed



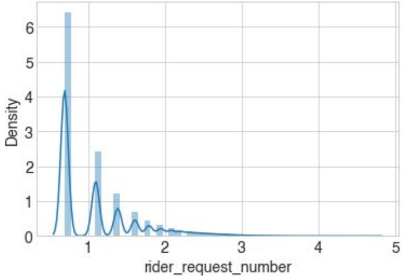
Average cancellation rate for penalty $1 riders - log transformed



Average rider paid amount for penalty $1 riders - log transformed



Average rider request number for penalty $1 riders - log transformed



Minimum size and minimum required sample size code

```
# get the required sample size for each group
effect_size = sms.proportion_effectsize(0.13, 0.15)    # Calculating effect size based on our expected rates

required_n = sms.NormalIndPower().solve_power(
    effect_size,
    power=0.8,
    alpha=0.05,
    ratio=1
    )                                                   # Calculating sample size needed

required_n = ceil(required_n)                          # Rounding up to next whole number

print(f"We need at least {required_n} samples for each group.")
```

```
We need at least 4720 samples for each group.
```