IRON HACK

CAROLINE ARAUJO & DANI SIAJ

Ironhack Payment EDA

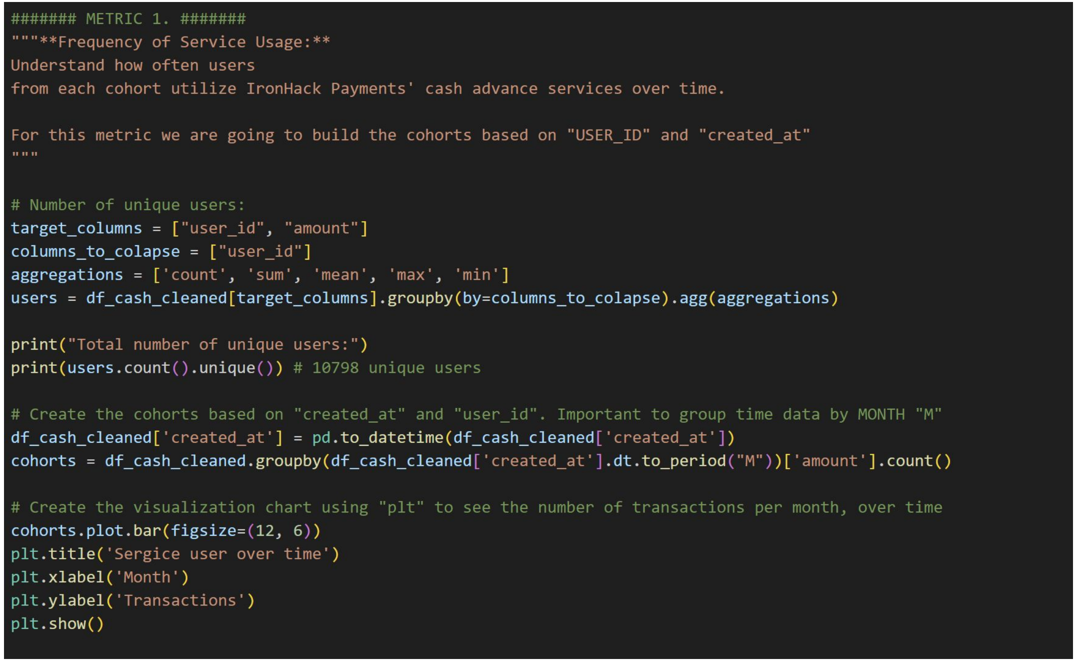
Table of Contents

- Metadata of Databases
- The code
- Tableau Dashboard

Metadata of the Databases

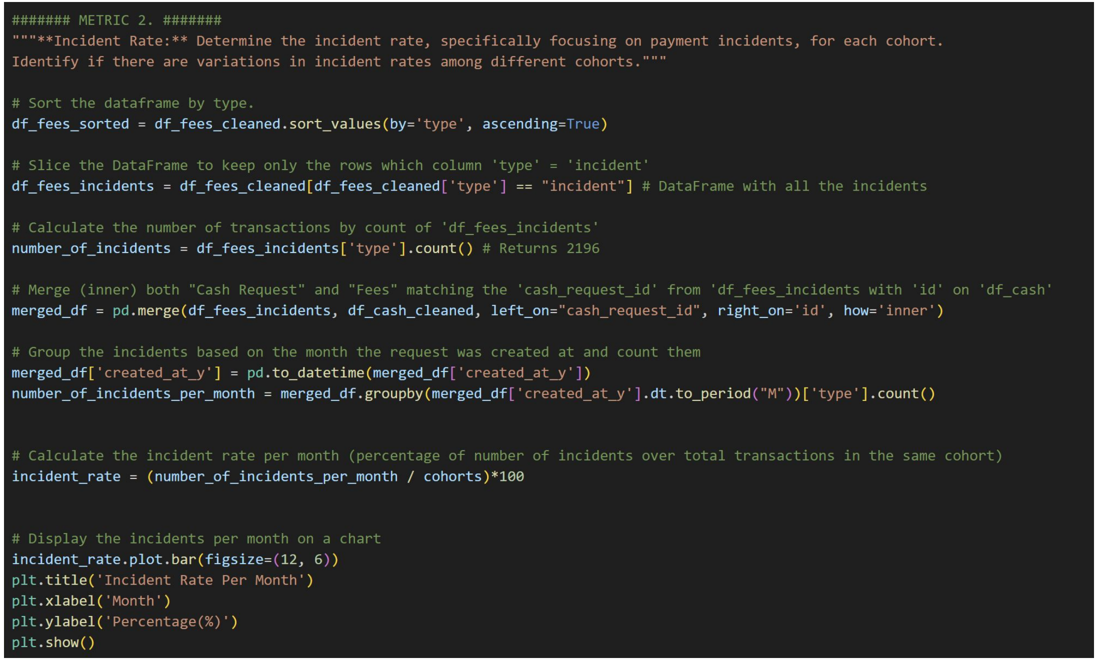
Cash Request Database (.csv)	Shape: 23970 columns, 16 rows
Fees Database (.csv)	Shape: 21061 columns, 16 rows
Lexique (.xlsx)	Shape: 16375 columns, 2 rows

The code



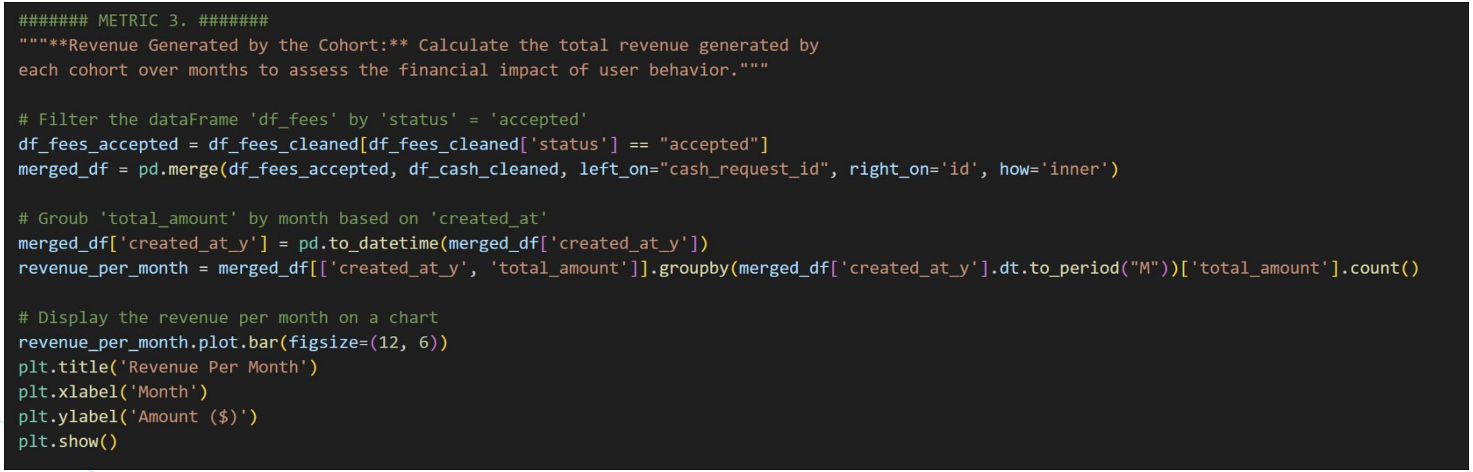
```
####### METRIC 1. #######
"""**Frequency of Service Usage:**
Understand how often users
from each cohort utilize IronHack Payments' cash advance services over time.

For this metric we are going to build the cohorts based on "USER_ID" and "created_at"
"""

# Number of unique users:
target_columns = ["user_id", "amount"]
columns_to_colapse = ["user_id"]
aggregations = ['count', 'sum', 'mean', 'max', 'min']
users = df_cash_cleaned[target_columns].groupby(by=columns_to_colapse).agg(aggregations)

print("Total number of unique users:")
print(users.count().unique()) # 10798 unique users

# Create the cohorts based on "created_at" and "user_id". Important to group time data by MONTH "M"
df_cash_cleaned['created_at'] = pd.to_datetime(df_cash_cleaned['created_at'])
cohorts = df_cash_cleaned.groupby(df_cash_cleaned['created_at'].dt.to_period("M"))['amount'].count()

# Create the visualization chart using "plt" to see the number of transactions per month, over time
cohorts.plot.bar(figsize=(12, 6))
plt.title('Sergice user over time')
plt.xlabel('Month')
plt.ylabel('Transactions')
plt.show()
```

The code



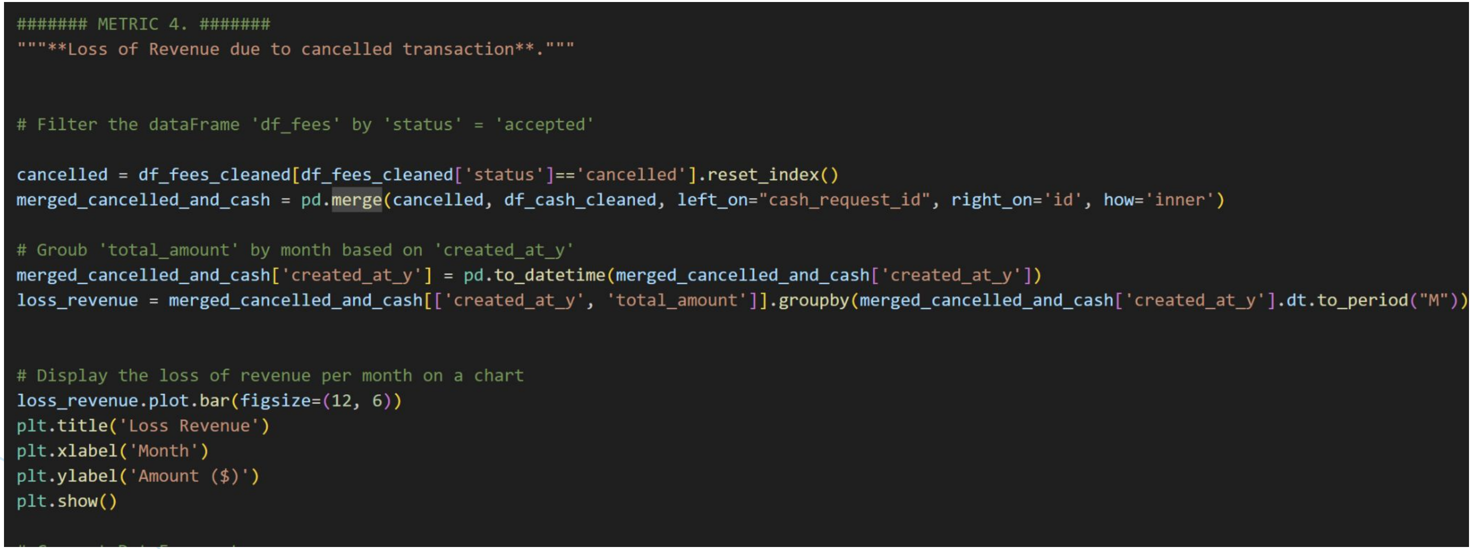
```
####### METRIC 2. #######
"""**Incident Rate:** Determine the incident rate, specifically focusing on payment incidents, for each cohort.
Identify if there are variations in incident rates among different cohorts."""

# Sort the dataframe by type.
df_fees_sorted = df_fees_cleaned.sort_values(by='type', ascending=True)

# Slice the DataFrame to keep only the rows which column 'type' = 'incident'
df_fees_incidents = df_fees_cleaned[df_fees_cleaned['type'] == "incident"] # DataFrame with all the incidents

# Calculate the number of transactions by count of 'df_fees_incidents'
number_of_incidents = df_fees_incidents['type'].count() # Returns 2196

# Merge (inner) both "Cash Request" and "Fees" matching the 'cash_request_id' from 'df_fees_incidents with 'id' on 'df_cash'
merged_df = pd.merge(df_fees_incidents, df_cash_cleaned, left_on="cash_request_id", right_on='id', how='inner')

# Group the incidents based on the month the request was created at and count them
merged_df['created_at_y'] = pd.to_datetime(merged_df['created_at_y'])
number_of_incidents_per_month = merged_df.groupby(merged_df['created_at_y'].dt.to_period("M"))['type'].count()


# Calculate the incident rate per month (percentage of number of incidents over total transactions in the same cohort)
incident_rate = (number_of_incidents_per_month / cohorts)*100


# Display the incidents per month on a chart
incident_rate.plot.bar(figsize=(12, 6))
plt.title('Incident Rate Per Month')
plt.xlabel('Month')
plt.ylabel('Percentage(%)')
plt.show()
```

The code



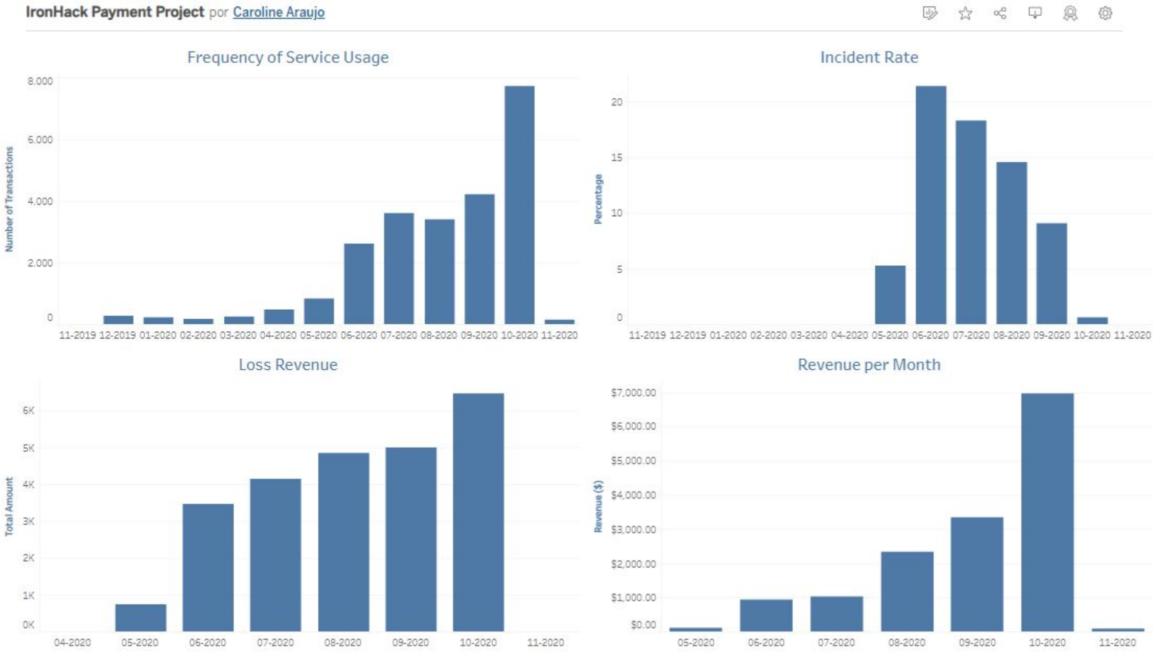
```
####### METRIC 3. #######
"""**Revenue Generated by the Cohort:** Calculate the total revenue generated by
each cohort over months to assess the financial impact of user behavior."""

# Filter the dataFrame 'df_fees' by 'status' = 'accepted'
df_fees_accepted = df_fees_cleaned[df_fees_cleaned['status'] == "accepted"]
merged_df = pd.merge(df_fees_accepted, df_cash_cleaned, left_on="cash_request_id", right_on='id', how='inner')

# Groub 'total_amount' by month based on 'created_at'
merged_df['created_at_y'] = pd.to_datetime(merged_df['created_at_y'])
revenue_per_month = merged_df[['created_at_y', 'total_amount']].groupby(merged_df['created_at_y'].dt.to_period("M"))['total_amount'].count()

# Display the revenue per month on a chart
revenue_per_month.plot.bar(figsize=(12, 6))
plt.title('Revenue Per Month')
plt.xlabel('Month')
plt.ylabel('Amount ($)')
plt.show()
```

The code



```
####### METRIC 4. #######
"""**Loss of Revenue due to cancelled transaction**."""


# Filter the dataFrame 'df_fees' by 'status' = 'accepted'

cancelled = df_fees_cleaned[df_fees_cleaned['status']=='cancelled'].reset_index()
merged_cancelled_and_cash = pd.merge(cancelled, df_cash_cleaned, left_on="cash_request_id", right_on='id', how='inner')

# Groub 'total_amount' by month based on 'created_at_y'
merged_cancelled_and_cash['created_at_y'] = pd.to_datetime(merged_cancelled_and_cash['created_at_y'])
loss_revenue = merged_cancelled_and_cash[['created_at_y', 'total_amount']].groupby(merged_cancelled_and_cash['created_at_y'].dt.to_period("M"))


# Display the loss of revenue per month on a chart
loss_revenue.plot.bar(figsize=(12, 6))
plt.title('Loss Revenue')
plt.xlabel('Month')
plt.ylabel('Amount ($)')
plt.show()
```

Tableau Dashboard



IronHack Payment Project por Caroline Araujo

IRON HACK

CAROLINE ARAUJO & DANI SIAJ

Thank you! Happy weekend!

🎉🥳🎊✨