My Case Study: Instacart Analysis

by Jaco Du Toit

GitHub

Project Overview

Objective

Instacart is an online grocery store that operates through an app. They already had very good sales, but wanted to uncover more about their sales patterns. My task was to perform an initial data and exploratory analysis of some of their data to show them how to improve their sales. This task took me just over a week to complete.

Data

- Open source data sets downloaded directly from Instacart.
- Al data sets include some kind of common identifier.

Skills

- Python
- Data wrangling
- Data merging
- Delivering variables
- Grouping data
- Aggregating data
- Reporting in Excel
- Population flows

Project / Tools

- Final Python Scripts
- Final Report
- Project Brief
- Data Sets
- Python
- Jupyter Notebooks
- Visual Studio Code 2

Initial Steps

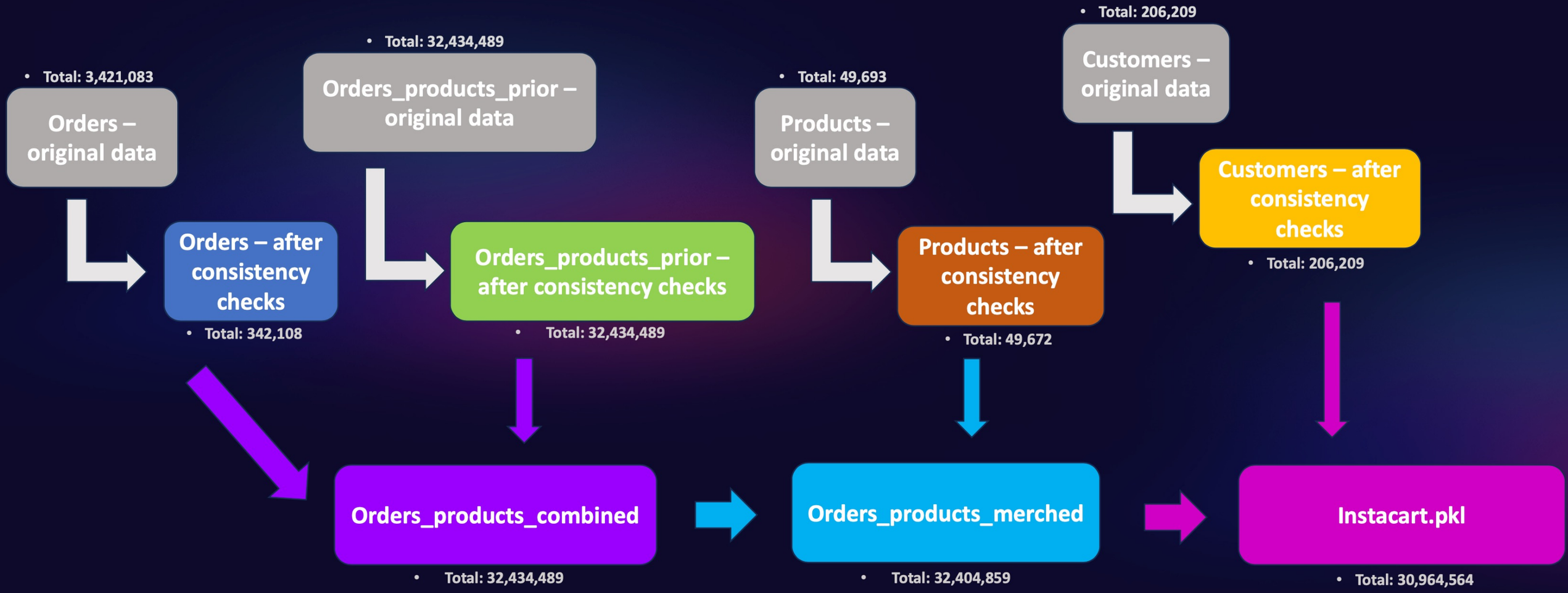
Consistency Checks, Data Wrangling and Merging

I was given four dataframes to work with: ’orders’, ‘products’, ‘departments’ and ’customers’. I then went through all the usual steps to get these dataframes ready to answer some questions. First I checked the consistency of the data and addressed all mixed type variables, missing values and all duplicates.

With that out of the way, I put on my cowboy hat, strapped on my boots and got my wrangling on. Here I dropped a couple of unnecessary columns, renamed a few and changed a couple of variables’ data types.

Finally I merged the four dataframes, into one, big, glorious dataframe, ready to tell me what I needed to know.

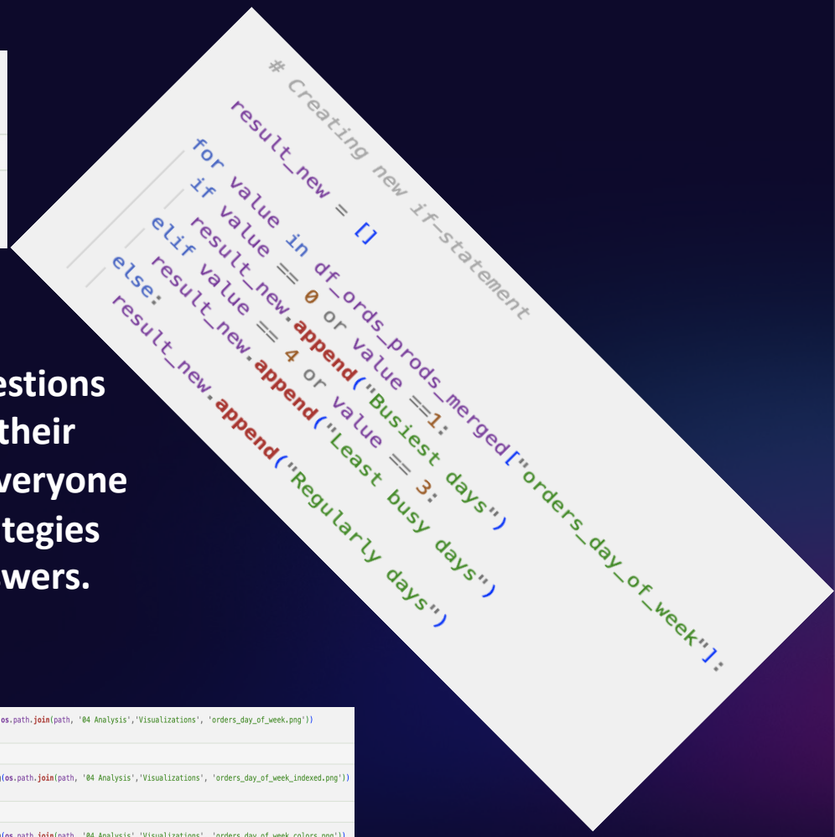
Key Analysis
Population flow



Exploratory Analysis and Visualizations


```
# Exporting the newly merged data set in pickle format
df_merged_large.to_pickle(os.path.join(path, '02 Data', 'Prepared Data', 'orders_products_combined.pkl'))
```

```
# Exporting the newly merged data set in .csv format. This is just for myself.
df_merged_large.to_csv(os.path.join(path, '02 Data', 'Prepared Data', 'orders_products_combined.csv'))
```



The stakeholders and sales team from Instacart had some questions that needed answering. They wanted to find out more about their customers' purchasing behaviors, since they couldn't target everyone the same way. They needed to create different marketing strategies for different groups of people and so they came to me for answers.

```
df_ords_prods_merged['orders_day_of_week'].value_counts(dropna = False)
```

```
orders_day_of_week
0    6204182
1    5660230
6    4496490
2    4213830
5    4205791
3    3840534
4    3783802
Name: count, dtype: int64
```

```
result = []

for value in df_ords_prods_merged["orders_day_of_week"]:
  if value == 0:
    result.append("Busiest day")
  elif value == 4:
    result.append("Least busy")
  else:
    result.append("Regularly busy")
```

```
# Define a function

def price_label(row):

  if row['prices'] <= 5:
    return 'Low-range product'
  elif (row['prices'] > 5) and (row['prices'] <= 15):
    return 'Mid-range product'
  elif row['prices'] > 15:
    return 'High range'
  else: return 'Not enough data'


# Apply the function
df['price_range'] = df.apply(price_label, axis = 1)
```

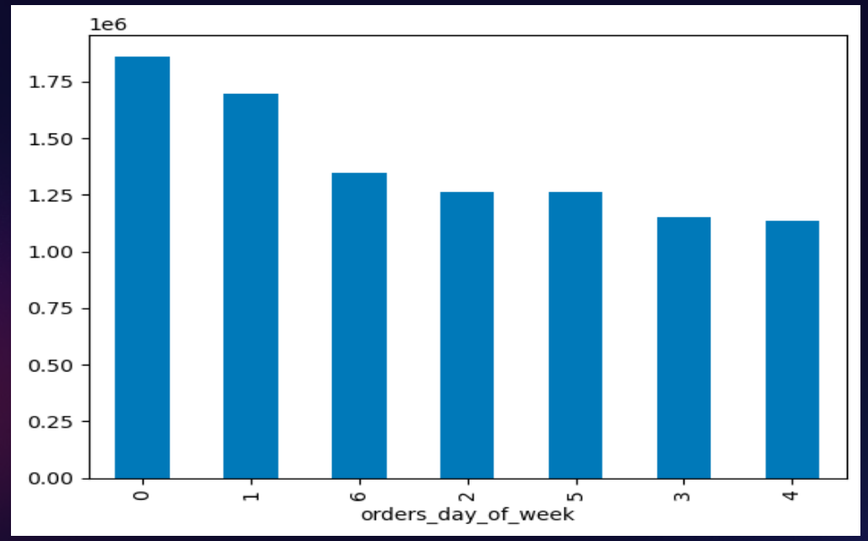
```
bar_chart.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'orders_day_of_week.png'))

bar_chart2.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'orders_day_of_week_indexed.png'))

bar_chart3.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'orders_day_of_week_colors.png'))

hist2.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'prices.png'))

hist.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'prices_more_bins.png'))

line.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'orders_day_of_week_line.png'))
```

Questions that Needed Answering

```
bar_chart = instacart_sub['orders_day_of_week'].value_counts().plot.bar()
```

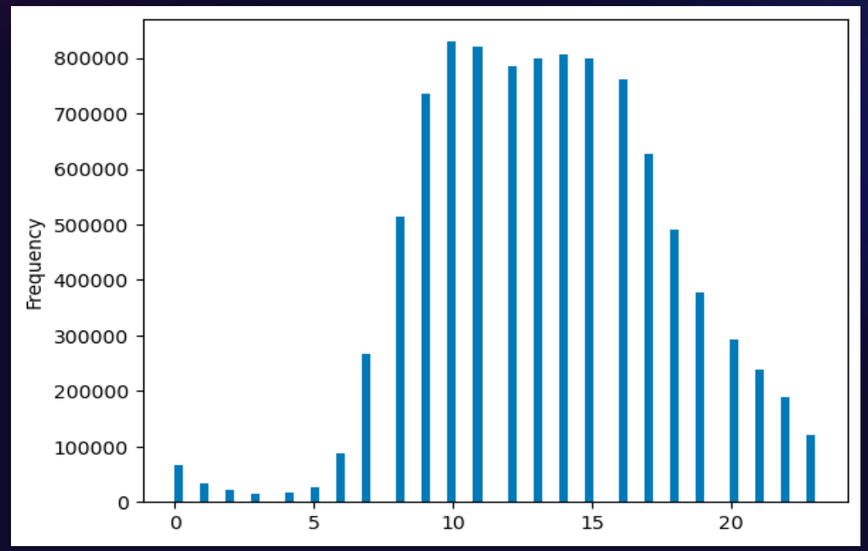
The sales team needed to know what the busiest days of the week and hours of the day were in order to schedule ads at times when there were fewer orders. The code to get those answers were pretty straightforward.

```
hist3 = instacart_sub['order_hour_of_day'].plot.hist(bins = 75)
```



Days of the week

0 – Saturday
1 – Sunday
2 – Monday
3 – Tuesday
4 – Wednesday
5 – Thursday
6 – Friday



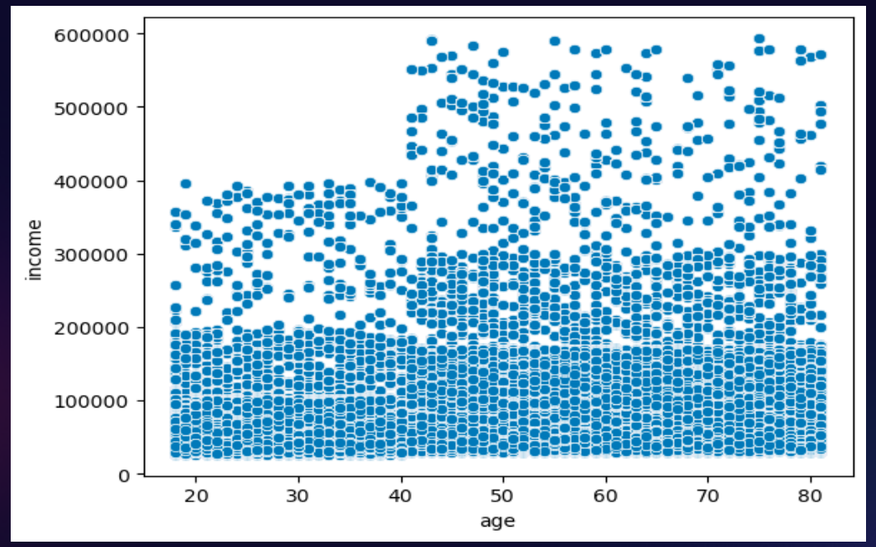
Hours of the day

Questions that Needed Answering

Who were the top buyers?

The older people got, the more spending power they had.

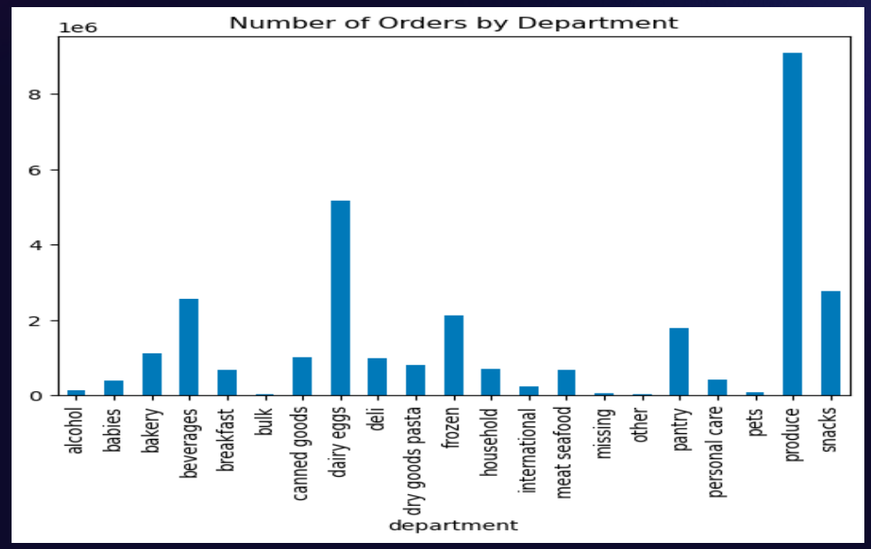
```
scatter = sns.scatterplot(x = 'age', y = 'income',data = instacart_sub)
```



And what products were the top sellers?

Produce, followed by dairy/eggs and snacks.

```
departments_frequency = df_merged['department'].value_counts().plot.barh()
# Title and labels
plt.title('Number of Orders by Department')
plt.ylabel('department')
```



I proposed the following:

Marketing

- Run targeted ads during peak hours when most people are shopping, thus ensuring most visibility.
- To attract more customers for the off-peak times, consider running ads that offer discounts and incentives for those periods.

Products

- Produce is the best-selling department among all ages and family groups. Expand the department even further. Offer bigger varieties and more options and increase advertising.

Customers

- Offer discounts and incentives to the people in the ‘lower income’ bracket as they spend almost no money compared to the rest. Vouchers sent directly to their house will give them a reason to shop.
- Target the older people with products specifically made for them as they have the most spending power.

Retrospective

What went well

I really enjoyed this analysis. Working with multiple data sets consisting of 32.5 million rows that needed cleaning, wrangling and merging, was a great opportunity to practice and showcase my analytical skills. In the end, I managed to answer all of the stakeholders' questions successfully.

What didn't go well

The dataframes had some constraints, which took a bit of time to overcome, but overall, I think this was a pretty straightforward analysis.

What's next

I would like to further explore and answer any additional questions Instacart might have. There is always room for improvement.

Final thoughts

Overall, I'm really happy with the entire project and how it all went. I ran into some issues, which my tutor helped me with and that actually gave me a chance to practice troubleshooting my code.