a5-Manual-PDF

● Graded

22 Hours, 28 Minutes Late

Group

Feven Tefera
Mihret Kemal
Michael Ngala
...and 1 more
View or edit group

Total Points

31.25 / 33 pts

Question 1

Question 1 — **31.25** / 32 pts

1.1 **Q1(b)** — **7.5** / 8 pts

✓ **– 0.5 pts** unnecessary markup, color, or addition to the plot

1.2 **Q1(c)** — **8** / 8 pts

✓ **– 0 pts** Correct

1.3 **Q1(d)** — **7.75** / 8 pts

✓ **– 0.25 pts** too many bins

1.4 **Q1(f)** — **8** / 8 pts

✓ **– 0 pts** Correct

Question 2

General — **0** / 1 pt

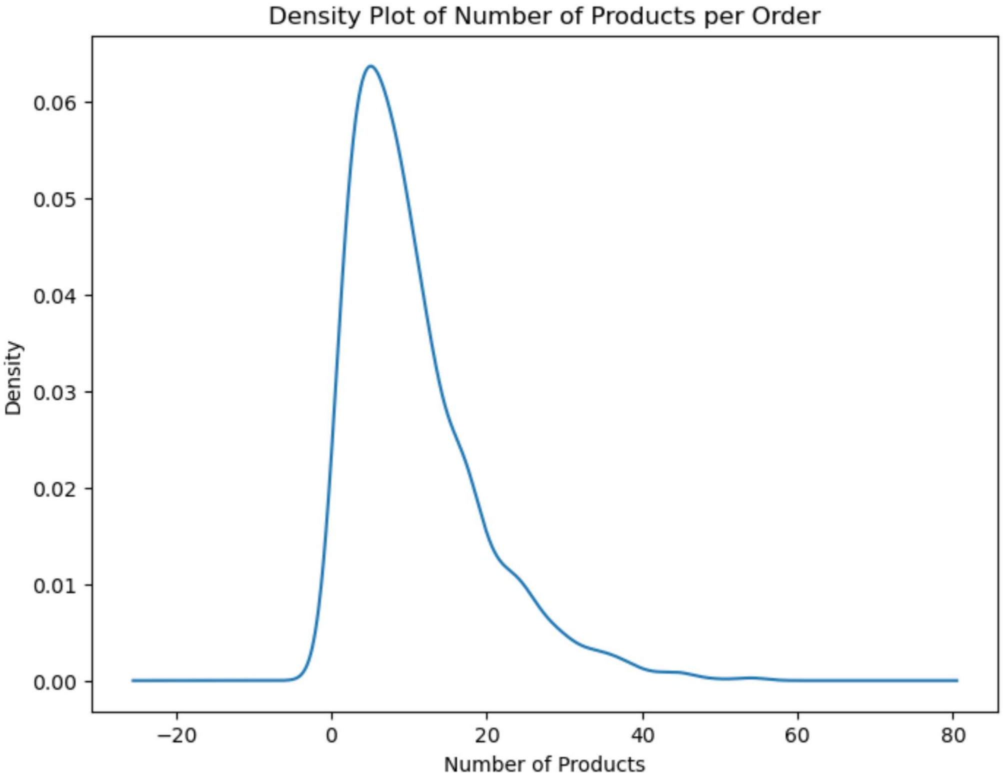
✓ **– 3 pts** Stylistically, you should break your commands that are running over 80 characters long into separate lines. Future assignments will lose points

0.1 Q1(b) - Explore the Data

Create a density plot showing the number of products per order using the `orders` data set.

```
In [98]: # Calculate the total number of products per order
         num_products_per_order = orders.sum(axis=1)

         # Plot the number of products per order
         plt.figure(figsize=(8, 6))
         num_products_per_order.plot(kind='density')
         plt.title('Density Plot of Number of Products per Order')
         plt.xlabel('Number of Products')
         plt.ylabel('Density')
         plt.show()
```



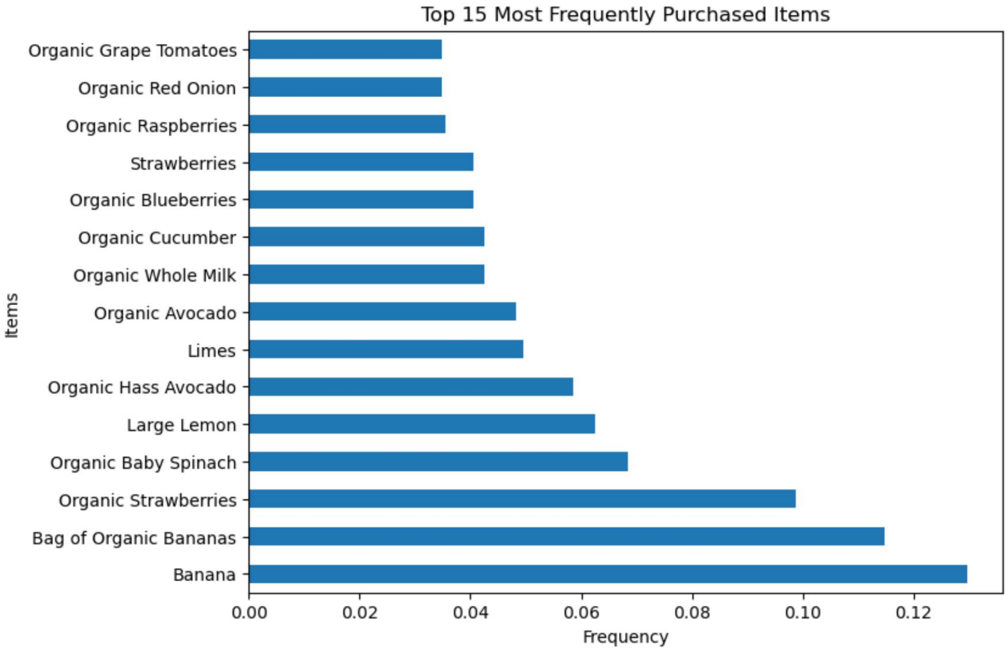
Questions assigned to the following page: 1.2 and 2

0.2 Q1(c) - Explore the Data, part 2

For the `orders` dataset, create an top 15 item frequency plot, that is plot the top 15 most frequently purchased items. This should be a bar plot with items vs. frequency (relative support).

```
In [99]: # Calculate the mean frequency of each item and select the top 15 most frequently purchased it
         top_15_items = orders.mean().sort_values(ascending = False).head(15)

         # Plot top 15 most frequently purchased product (by relative support)
         plt.figure(figsize=(8, 6))
         top_15_items.plot(kind = 'barh')
         plt.title('Top 15 Most Frequently Purchased Items')
         plt.xlabel('Frequency')
         plt.ylabel('Items')
         plt.show()
```

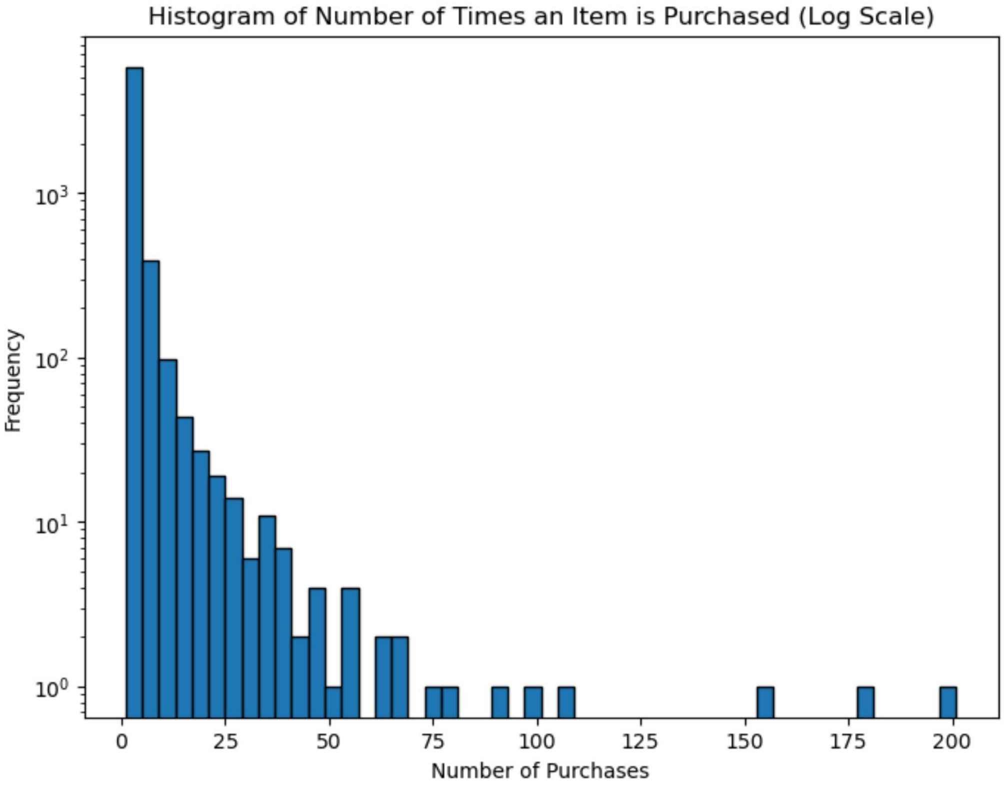


0.3 Q1(d) - Explore the Data, part 3

For the `orders` dataset, create an histogram of the number of times an item is purchased. You may want to consider using log scaling to view the data distribution more easily.

```
In [100]: # Calculate the total number of times each item is purchased
          num_item_purchased = orders.sum(axis=0)

          # Plot histogram of number of purchases per item.
          plt.figure(figsize=(8, 6))
          plt.hist(num_item_purchased, bins = 50, edgecolor = 'black', log = True)
          plt.title('Histogram of Number of Times an Item is Purchased (Log Scale)')
          plt.xlabel('Number of Purchases')
          plt.ylabel('Frequency')
          plt.show()
```

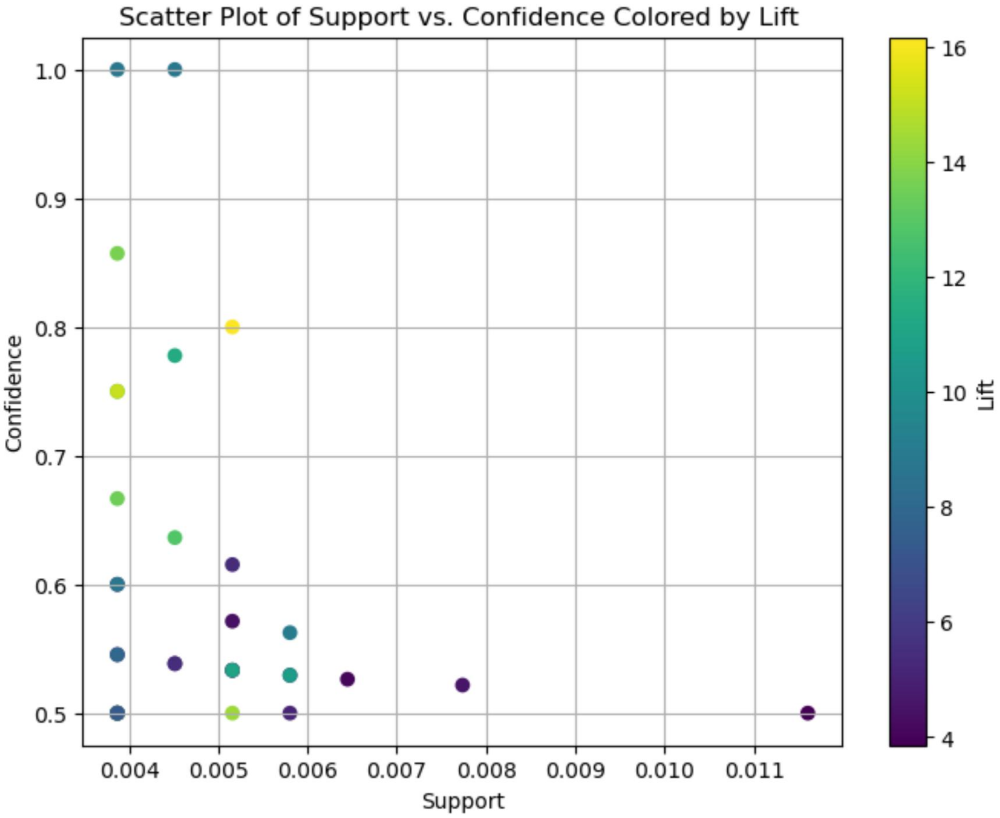


Questions assigned to the following page: 2 and 1.4

0.4 Q1(f) - Apriori, part 2

Create a scatterplot of the rules, plotting support vs. confidence colored by lift value.

```
In [103]: # Plot the results of Apriori
          plt.figure(figsize=(8, 6))
          plt.scatter(rules['support'], rules['confidence'], c = rules['lift'], cmap = 'viridis')
          plt.colorbar(label = 'Lift')
          plt.title('Scatter Plot of Support vs. Confidence Colored by Lift ')
          plt.xlabel('Support')
          plt.ylabel('Confidence')
          plt.grid(True)
          plt.show()
```


Scatter Plot of Support vs. Confidence Colored by Lift
Confidence
Support
Lift