

P2_Filtering_Sorting

January 31, 2026

Python Tools, Getting to know your Data, Filtration, and Visualization

0.0.1 1. Import the necessary libraries

```
[27]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

0.0.2 2. Import the chipotle dataset

```
[28]: path = 'chipotle.tsv'

chipo = pd.read_csv(path, sep = '\t')
```

0.0.3 3. Load the dataset and display the first 5 rows.

```
[29]: chipo.head()
```

```
[29]:
```

	order_id	quantity	item_name \
0	1	1	Chips and Fresh Tomato Salsa
1	1	1	Izze
2	1	1	Nantucket Nectar
3	1	1	Chips and Tomatillo-Green Chili Salsa
4	2	2	Chicken Bowl

	choice_description	item_price
0	NaN	\$2.39
1	[Clementine]	\$3.39
2	[Apple]	\$3.39
3	NaN	\$2.39
4	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	\$16.98

0.0.4 4. How many rows and columns does the dataset have?

```
[30]: chipo.shape
```

```
[30]: (4622, 5)
```

0.0.5 5. What are the column names in the dataset?

```
[31]: chipo.columns
```

```
[31]: Index(['order_id', 'quantity', 'item_name', 'choice_description',  
         'item_price'],  
        dtype='object')
```

0.0.6 6. How is the dataset indexed?

```
[32]: chipo.index
```

```
[32]: RangeIndex(start=0, stop=4622, step=1)
```

0.0.7 7. Which columns are categorical and which are numerical?

```
[33]: categorical_cols = chipo.select_dtypes(include='object').columns  
      numerical_cols = chipo.select_dtypes(exclude='object').columns  
  
      print(categorical_cols)  
      print(numerical_cols)
```

```
Index(['item_name', 'choice_description', 'item_price'], dtype='object')  
Index(['order_id', 'quantity'], dtype='object')
```

```
[34]: #OR  
      chipo.dtypes
```

```
[34]: order_id          int64  
      quantity         int64  
      item_name        object  
      choice_description object  
      item_price        object  
      dtype: object
```

0.0.8 8. Why is item_price not considered numerical?

Let us investigate that

```
[35]: chipo['item_price'].head()
```

```
[35]: 0    $2.39  
      1    $3.39  
      2    $3.39  
      3    $2.39  
      4   $16.98  
      Name: item_price, dtype: object
```

0.0.9 9. Convert item_price to a numerical (float) column

```
[36]: chipo['item_price'] = chipo['item_price'].str.replace('$', '').astype(float)
```

```
[37]: #check the data type of 'item_price'  
chipo['item_price'].dtype
```

```
[37]: dtype('float64')
```

0.0.10 10. Show basic statistics for numerical columns

```
[38]: chipo.describe()
```

```
[38]:
```

	order_id	quantity	item_price
count	4622.000000	4622.000000	4622.000000
mean	927.254868	1.075725	7.464336
std	528.890796	0.410186	4.245557
min	1.000000	1.000000	1.090000
25%	477.250000	1.000000	3.390000
50%	926.000000	1.000000	8.750000
75%	1393.000000	1.000000	9.250000
max	1834.000000	15.000000	44.250000

0.1 Sorting and Filtering Data

0.1.1 11. How many products cost more than \$10.00?

```
[39]: chipo[chipo.item_price > 10.00]['order_id'].count()
```

```
[39]: np.int64(1130)
```

```
[40]: # OR you can use size  
chipo[chipo.item_price > 10.00]['order_id'].size
```

```
[40]: 1130
```

```
[41]: print(chipo[chipo.item_price > 10.00]['order_id'].count())
```

```
1130
```

Important Notes The difference between `size` and `count()` in Pandas: - `size` measures total entries, while `count()` measures valid (non-null) entries. - `size` returns a single integer (total elements), while `count()` returns a series (per-column counts in a DataFrame) or a single integer (for a Series).

0.1.2 12. What is the price of each item?

A simple way to do that is to get the data frame with only two columns, `item_name` and `item_price`

```
[42]: prices = chipo[['item_name', 'item_price']]
prices
```

```
[42]:
```

	item_name	item_price
0	Chips and Fresh Tomato Salsa	2.39
1	Izze	3.39
2	Nantucket Nectar	3.39
3	Chips and Tomatillo-Green Chili Salsa	2.39
4	Chicken Bowl	16.98
...
4617	Steak Burrito	11.75
4618	Steak Burrito	11.75
4619	Chicken Salad Bowl	11.25
4620	Chicken Salad Bowl	8.75
4621	Chicken Salad Bowl	8.75

[4622 rows x 2 columns]

0.1.3 13. Sort the dataset by the item name

```
[43]: chipo.item_name.sort_values()
```

```
[43]: 3389    6 Pack Soft Drink
341      6 Pack Soft Drink
1849    6 Pack Soft Drink
1860    6 Pack Soft Drink
2713    6 Pack Soft Drink
...
2384    Veggie Soft Tacos
781      Veggie Soft Tacos
2851    Veggie Soft Tacos
1699    Veggie Soft Tacos
1395    Veggie Soft Tacos
Name: item_name, Length: 4622, dtype: object
```

```
[44]: # OR
chipo.sort_values(by = "item_name")
```

```
[44]:
```

	order_id	quantity	item_name \
3389	1360	2	6 Pack Soft Drink
341	148	1	6 Pack Soft Drink
1849	749	1	6 Pack Soft Drink
1860	754	1	6 Pack Soft Drink
2713	1076	1	6 Pack Soft Drink
...
2384	948	1	Veggie Soft Tacos
781	322	1	Veggie Soft Tacos

2851	1132	1	Veggie Soft Tacos
1699	688	1	Veggie Soft Tacos
1395	567	1	Veggie Soft Tacos

	choice_description	item_price
3389	[Diet Coke]	12.98
341	[Diet Coke]	6.49
1849	[Coke]	6.49
1860	[Diet Coke]	6.49
2713	[Coke]	6.49
...
2384	[Roasted Chili Corn Salsa, [Fajita Vegetables, ...]	8.75
781	[Fresh Tomato Salsa, [Black Beans, Cheese, Sou...]	8.75
2851	[Roasted Chili Corn Salsa (Medium), [Black Bea...]	8.49
1699	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, ...]	11.25
1395	[Fresh Tomato Salsa (Mild), [Pinto Beans, Rice, ...]	8.49

[4622 rows x 5 columns]

0.1.4 14. What is the quantity of the most expensive item ordered?

```
[45]: chipo.sort_values(by = "item_price", ascending = False)['quantity'].head(1)
```

```
[45]: 3598    15
      Name: quantity, dtype: int64
```

```
[46]: # OR
      chipo[chipo['item_price'] == max(chipo['item_price'])]['quantity']
```

```
[46]: 3598    15
      Name: quantity, dtype: int64
```

As you saw above, the above way returns a series with both the index and the value. If you want just the value without the index, you can extract it in a few different ways:

```
[47]: chipo.sort_values(by="item_price", ascending=False)['quantity'].head(1).
      ↪values[0]
```

```
[47]: np.int64(15)
```

```
[48]: chipo.sort_values(by="item_price", ascending=False)['quantity'].iloc[0]
```

```
[48]: np.int64(15)
```

```
[49]: chipo.sort_values(by="item_price", ascending=False)['quantity'].head(1).item()
```

```
[49]: 15
```

0.1.5 15. How many times was a Veggie Salad Bowl ordered?

```
[50]: print(chipo[chipo.item_name == "Veggie Salad Bowl"]['quantity'].sum())
```

18

0.1.6 16. How many times did someone order more than one Canned Soda?

```
[51]: condition = (chipo.item_name == "Canned Soda") & (chipo.quantity > 1)
```

```
[52]: chipo[condition]['quantity'].count()
```

```
[52]: np.int64(20)
```

0.1.7 17. How many different products are sold?

```
[53]: unique_products = chipo['item_name'].nunique()

print(unique_products)
```

50

0.1.8 18. What is the total revenue?

```
[54]: total_revenue = (chipo['item_price'] * chipo['quantity']).sum()

print(round(total_revenue))
```

39237

0.1.9 19. What is the average price of items?

```
[55]: average_price = chipo['item_price'].mean()

print(round(average_price, 2))
```

7.46

0.1.10 20. How many orders were made in total?

```
[56]: total_orders = chipo['order_id'].nunique()

print(total_orders)
```

1834

0.1.11 21. What is the total quantity of items ordered?

```
[57]: total_quantity = chipo['quantity'].sum()

print(total_quantity)
```

4972

0.1.12 22. Which item has the highest average price?

```
[58]: chipo.groupby('item_name')['item_price'].mean().idxmax()
```

[58]: 'Bowl'

0.1.13 23. How many items include “Chicken” in their name?

```
[59]: chipo[chipo['item_name'].str.contains('Chicken']]['item_name'].count()
```

[59]: np.int64(1560)

```
[60]: chicken_items = chipo[chipo['item_name'].str.contains('Chicken']]['item_name'].
      ↪count()

print(chicken_items)
```

1560

0.1.14 24. Which item was the most-ordered item?

```
[61]: chipo.groupby('item_name')['quantity'].sum().idxmax()
```

[61]: 'Chicken Bowl'

```
[62]: #Option 2
c = chipo.groupby('item_name')
c = c.sum()
c = c.sort_values(['quantity'], ascending=False)
c.reset_index().item_name.head(1)
```

[62]: 0 Chicken Bowl
 Name: item_name, dtype: object

0.1.15 25. Group the dataset by item_name and count how many times each product appears.

```
[63]: chipo.groupby('item_name').size().sort_values(ascending=False)
```

[63]: item_name
 Chicken Bowl 726

Chicken Burrito	553
Chips and Guacamole	479
Steak Burrito	368
Canned Soft Drink	301
Chips	211
Steak Bowl	211
Bottled Water	162
Chicken Soft Tacos	115
Chicken Salad Bowl	110
Chips and Fresh Tomato Salsa	110
Canned Soda	104
Side of Chips	101
Veggie Burrito	95
Barbacoa Burrito	91
Veggie Bowl	85
Carnitas Bowl	68
Barbacoa Bowl	66
Carnitas Burrito	59
Steak Soft Tacos	55
6 Pack Soft Drink	54
Chips and Tomatillo Red Chili Salsa	48
Chicken Crispy Tacos	47
Chips and Tomatillo Green Chili Salsa	43
Carnitas Soft Tacos	40
Steak Crispy Tacos	35
Chips and Tomatillo-Green Chili Salsa	31
Steak Salad Bowl	29
Nantucket Nectar	27
Barbacoa Soft Tacos	25
Chips and Roasted Chili Corn Salsa	22
Chips and Tomatillo-Red Chili Salsa	20
Izze	20
Veggie Salad Bowl	18
Chips and Roasted Chili-Corn Salsa	18
Barbacoa Crispy Tacos	11
Barbacoa Salad Bowl	10
Chicken Salad	9
Carnitas Crispy Tacos	7
Veggie Soft Tacos	7
Burrito	6
Carnitas Salad Bowl	6
Veggie Salad	6
Steak Salad	4
Salad	2
Bowl	2
Crispy Tacos	2
Carnitas Salad	1

Chips and Mild Fresh Tomato Salsa	1
Veggie Crispy Tacos	1

dtype: int64

0.1.16 26. calculate the total quantity sold for each product.

```
[64]: item_quantity = chipo.groupby('item_name')['quantity'].sum()
      item_quantity
```

```
[64]: item_name
      6 Pack Soft Drink      55
      Barbacoa Bowl        66
      Barbacoa Burrito     91
      Barbacoa Crispy Tacos 12
      Barbacoa Salad Bowl  10
      Barbacoa Soft Tacos   25
      Bottled Water       211
      Bowl                 4
      Burrito              6
      Canned Soda         126
      Canned Soft Drink   351
      Carnitas Bowl       71
      Carnitas Burrito    60
      Carnitas Crispy Tacos 8
      Carnitas Salad       1
      Carnitas Salad Bowl  6
      Carnitas Soft Tacos  40
      Chicken Bowl       761
      Chicken Burrito    591
      Chicken Crispy Tacos 50
      Chicken Salad       9
      Chicken Salad Bowl 123
      Chicken Soft Tacos  120
      Chips              230
      Chips and Fresh Tomato Salsa 130
      Chips and Guacamole 506
      Chips and Mild Fresh Tomato Salsa 1
      Chips and Roasted Chili Corn Salsa 23
      Chips and Roasted Chili-Corn Salsa 18
      Chips and Tomatillo Green Chili Salsa 45
      Chips and Tomatillo Red Chili Salsa 50
      Chips and Tomatillo-Green Chili Salsa 33
      Chips and Tomatillo-Red Chili Salsa 25
      Crispy Tacos        2
      Izze                20
      Nantucket Nectar    29
      Salad               2
```

Side of Chips	110
Steak Bowl	221
Steak Burrito	386
Steak Crispy Tacos	36
Steak Salad	4
Steak Salad Bowl	31
Steak Soft Tacos	56
Veggie Bowl	87
Veggie Burrito	97
Veggie Crispy Tacos	1
Veggie Salad	6
Veggie Salad Bowl	18
Veggie Soft Tacos	8

Name: quantity, dtype: int64

0.1.17 27. What is the average quantity of items per order?

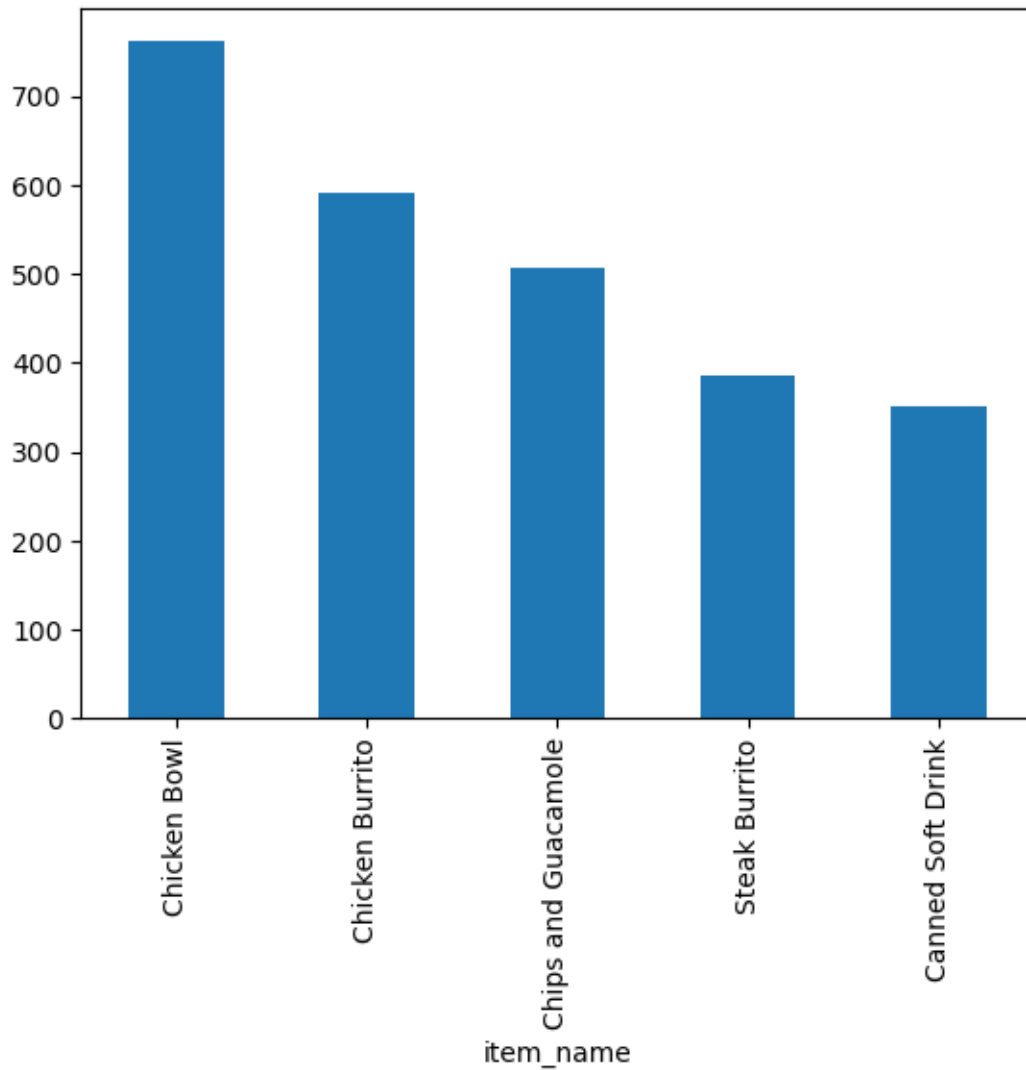
```
[65]: avg_items_per_order = chipo.groupby('order_id')['quantity'].mean()
      avg_items_per_order
```

```
[65]: order_id
      1      1.0
      2      2.0
      3      1.0
      4      1.0
      5      1.0
      ...
    1830      1.0
    1831      1.0
    1832      1.0
    1833      1.0
    1834      1.0
      Name: quantity, Length: 1834, dtype: float64
```

0.2 Visualization

0.2.1 28. Plot the top 5 most ordered items.

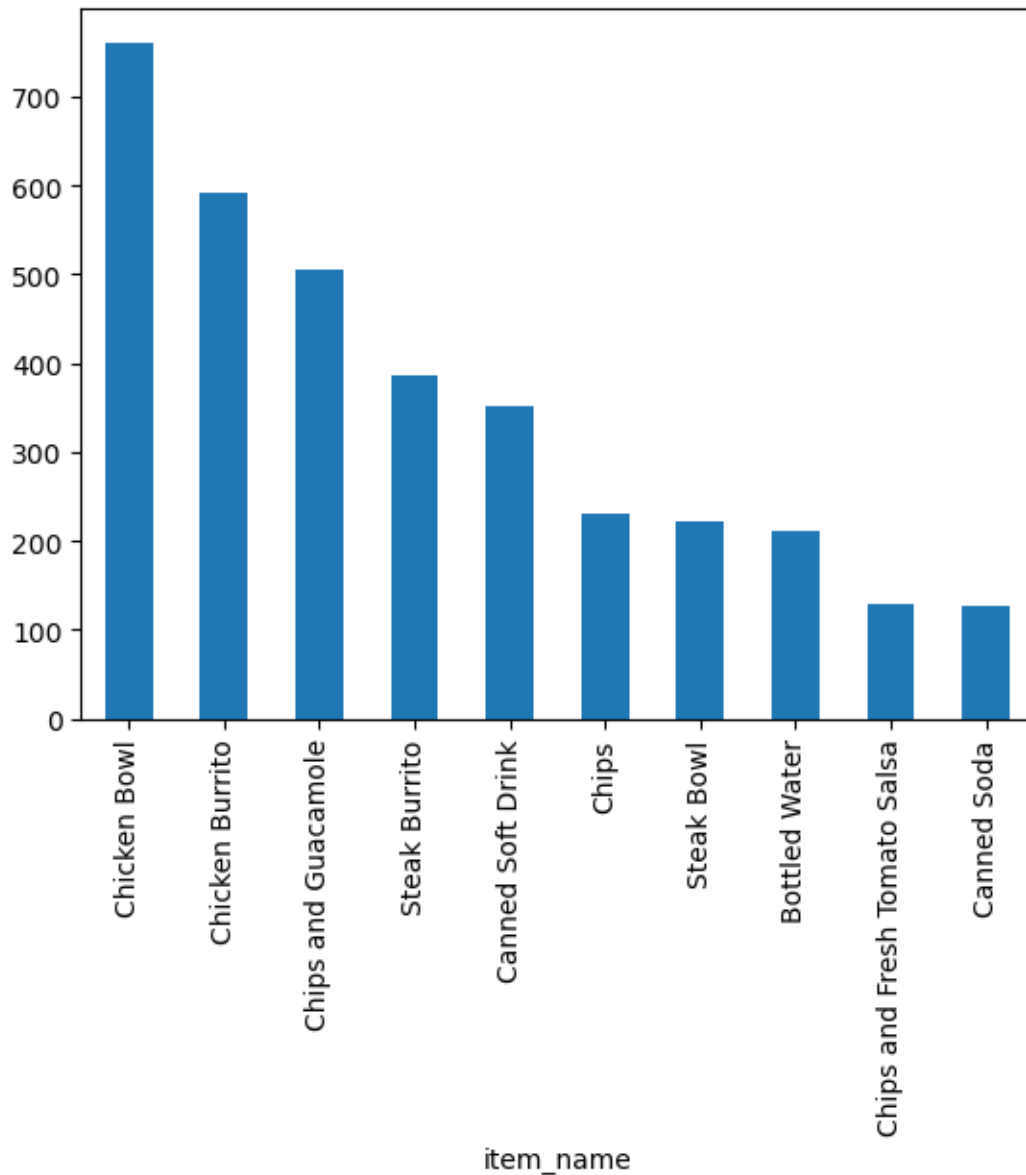
```
[66]: top5 = chipo.groupby('item_name')['quantity'].sum().
      ↪sort_values(ascending=False).head(5)
      top5.plot(kind='bar');
```



0.2.2 29. Plot the total quantity sold for the top 10 items.

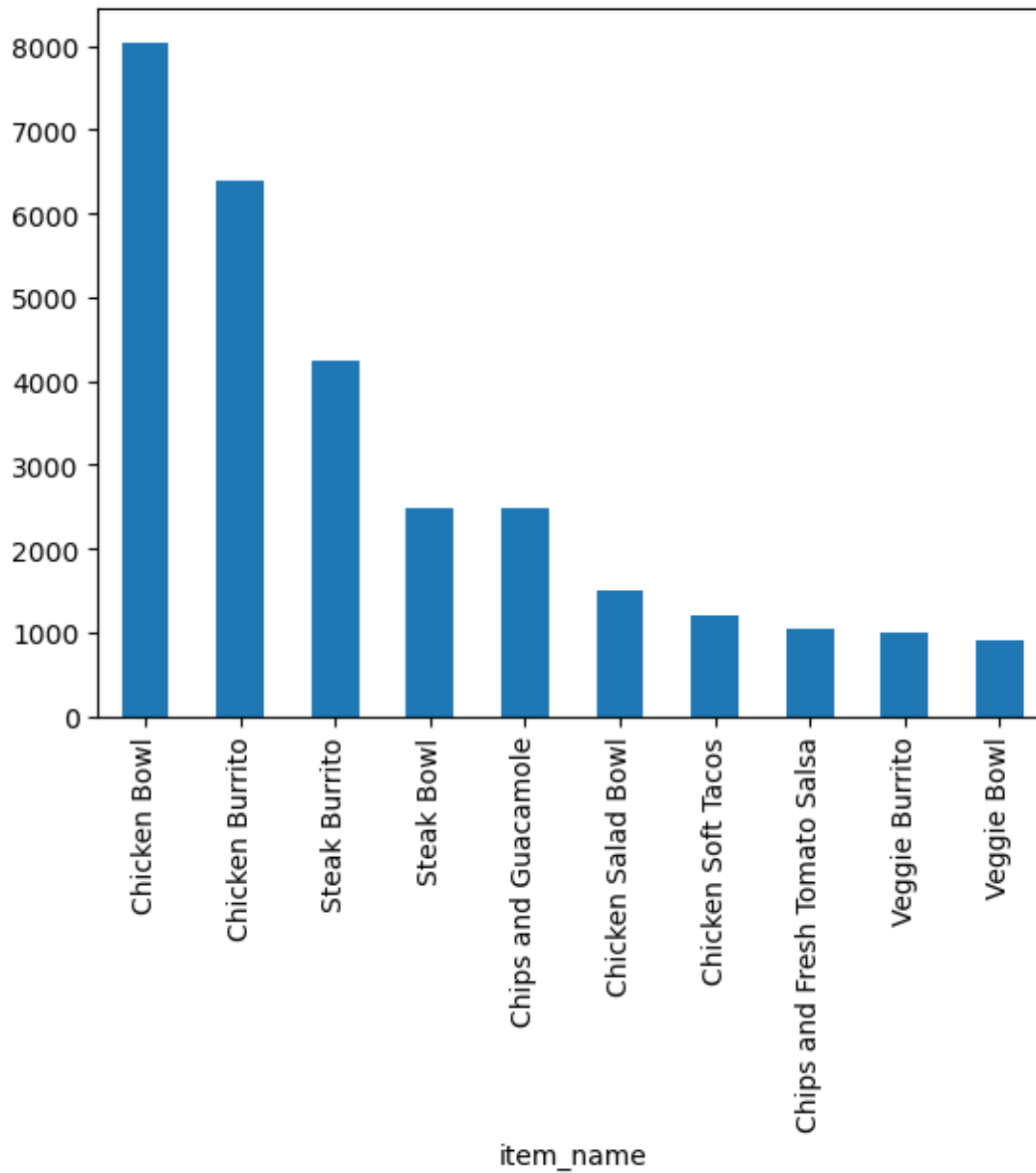
Same as Q28, but with different wording.

```
[67]: top10 = chipo.groupby('item_name')['quantity'].sum().  
      ↪ sort_values(ascending=False).head(10)  
      top10.plot(kind='bar');
```



0.2.3 30. Calculate the total revenue generated by each item and plot the top 10 highest-revenue items.

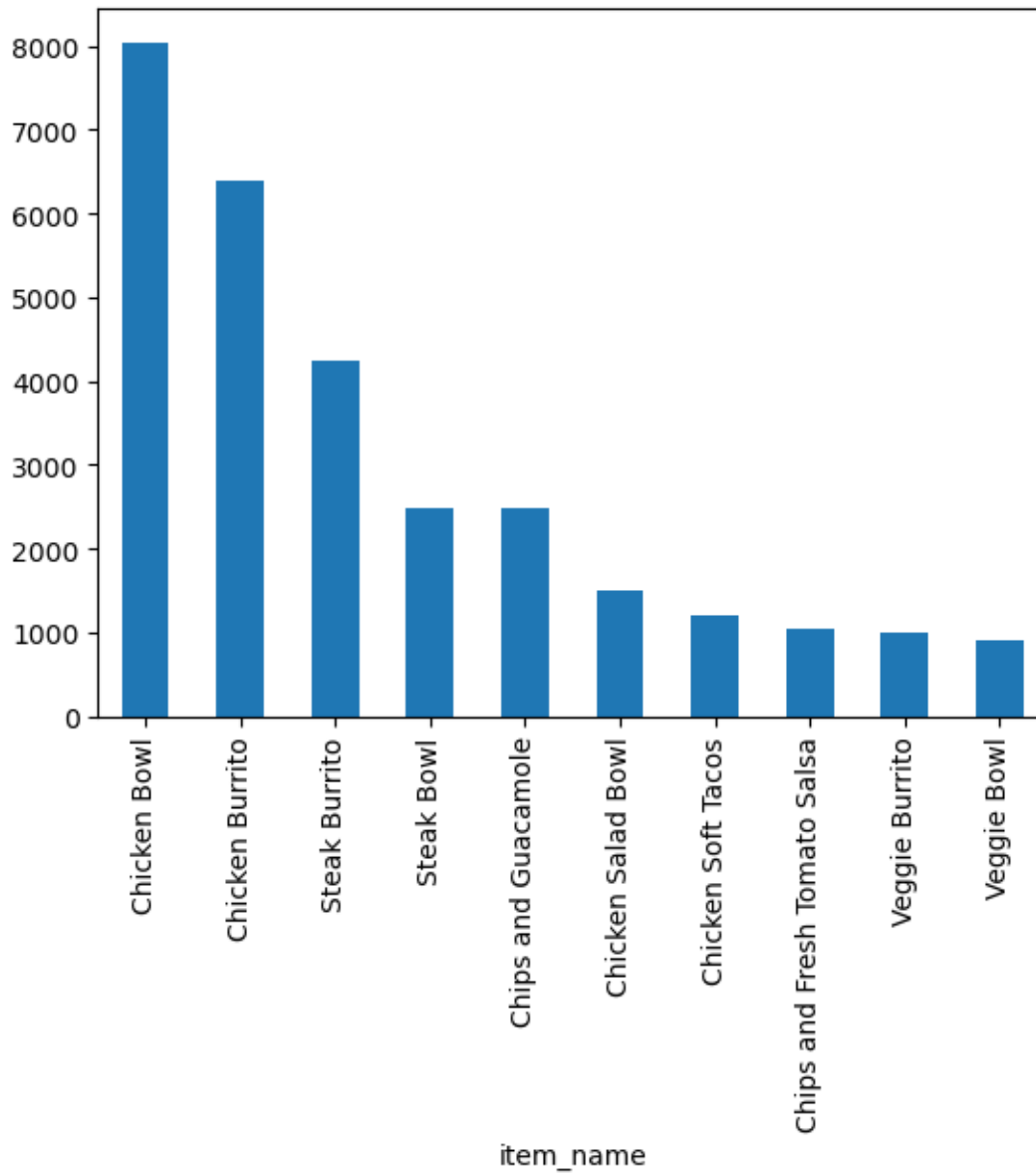
```
[68]: # Option 1
      #Create a revenue column
      chipo['revenue'] = chipo['item_price']*chipo['quantity']
      #groupby revenue and sum, and then plot
      chipo.groupby('item_name')['revenue'].sum().sort_values(ascending=False).
        ↪head(10).plot(kind='bar');
```



```
[69]: #Option 2
# Calculate revenue per item
revenue = (chipo['item_price'] * chipo['quantity']).groupby(chipo['item_name']).
    ↪sum()

# Sort and get top 10 items
top10_revenue = revenue.sort_values(ascending=False).head(10)

# Plot
top10_revenue.plot(kind='bar');
```



0.3 Challenge yourself

0.3.1 31. What was the most ordered item in the choice_description column?

[]:

0.3.2 32. What is the average revenue amount per order?

[]:

0.3.3 33. Which product has the highest total quantity sold?

[]:

0.3.4 34. calculate the average price of each product.

[]:

0.3.5 35. Which choice appears most often?

[]: