

Practical Exam Sample: Pet Supplies

PetMind is a retailer of products for pets. They are based in the United States.

PetMind sells products that are a mix of luxury items and everyday items. Luxury items include toys. Everyday items include food.

The company wants to increase sales by selling more products for some animals repeatedly.

They have been testing this approach for the last year.

They now want a report on how repeat purchases impact sales.

Data

The data is available in the table `pet_supplies`.

The dataset contains the sales records in the stores last year.

| Column Name | Criteria |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| product_id | Nominal. The unique identifier of the product. Missing values are not possible due to the database structure. |
| category | Nominal. The category of the product, one of 6 values (Housing, Food, Toys, Equipment, Medicine, Accessory). Missing values should be replaced with "Unknown". |
| animal | Nominal. The type of animal the product is for. One of Dog, Cat, Fish, Bird. Missing values should be replaced with "Unknown". |
| size | Ordinal. The size of animal the product is for. Small, Medium, Large. Missing values should be replaced with "Unknown". |
| price | Continuous. The price the product is sold at. Can be any positive value, round to 2 decimal places. Missing values should be replaced with the overall median price. |
| sales | Continuous. The value of all sales of the product in the last year. This can be any positive value, rounded to 2 decimal places. Missing values should be replaced with the overall median sales. |
| rating | Discrete. Customer rating of the product from 1 to 10. Missing values should be replaced with 0. |
| repeat_purchase | Nominal. Whether customers repeatedly buy the product (1) or not (0). Missing values should be removed. |

Unknown integration

DataFrame as df

SELECT * FROM pet_supplies

| ... | ↑↓ | pro... | ... | ↑↓ | cate... | ... | ↑↓ | ani... | ... | ↑↓ | size | ... | ↑↓ | price | ... | ↑↓ | sales | ... | ↑↓ | r... | ... | ↑↓ | repeat... | ... | ↑↓ | |
|-----|----|--------|-----|----|-----------|-----|----|--------|-----|----|--------|-----|----|-------|-----|----|---------|-----|----|------|-----|----|-----------|-----|----|--|
| | 0 | | | 1 | Food | | | Bird | | | large | | | 51.1 | | | 1860.62 | | | 7 | | | 1 | | | |
| | 1 | | | 2 | Housing | | | Bird | | | MEDIUM | | | 35.98 | | | 963.6 | | | 6 | | | 0 | | | |
| | 2 | | | 3 | Food | | | Dog | | | medium | | | 31.23 | | | 898.3 | | | 5 | | | 1 | | | |
| | 3 | | | 4 | Medicine | | | Cat | | | small | | | 24.95 | | | 982.15 | | | 6 | | | 1 | | | |
| | 4 | | | 5 | Housing | | | Cat | | | Small | | | 26.18 | | | 832.63 | | | 7 | | | 1 | | | |
| | 5 | | | 6 | Housing | | | Dog | | | Small | | | 30.77 | | | 874.58 | | | 7 | | | 0 | | | |
| | 6 | | | 7 | Housing | | | Dog | | | Small | | | 31.04 | | | 875.07 | | | 5 | | | 0 | | | |
| | 7 | | | 8 | Toys | | | Cat | | | medium | | | 28.9 | | | 1074.31 | | | 4 | | | 0 | | | |
| | 8 | | | 9 | Equipment | | | Fish | | | MEDIUM | | | 17.82 | | | 503.67 | | | 5 | | | 0 | | | |
| | 9 | | | 10 | Medicine | | | Dog | | | medium | | | 24.93 | | | 838.88 | | | 8 | | | 0 | | | |
| | 10 | | | 11 | Food | | | Dog | | | Large | | | 40.87 | | | 1457.22 | | | 7 | | | 1 | | | |
| | 11 | | | 12 | Medicine | | | Bird | | | MEDIUM | | | 34.96 | | | 1204.6 | | | 5 | | | 1 | | | |
| | 12 | | | 13 | Food | | | Dog | | | MEDIUM | | | 31.07 | | | 889.73 | | | 4 | | | 0 | | | |
| | 13 | | | 14 | Food | | | Dog | | | large | | | 40.8 | | | 1450.5 | | | 6 | | | 1 | | | |
| | 14 | | | 15 | Accessory | | | Bird | | | medium | | | 33.13 | | | 859.29 | | | 4 | | | 1 | | | |
| | 15 | | | 16 | Accessory | | | Bird | | | large | | | 43.09 | | | 1418.72 | | | 1 | | | 1 | | | |

Rows: 1,500

Expand

Task 1

From taking a quick look at the data, you are pretty certain it isn't quite as it should be. You need to make sure all of the data is clean before you start your analysis. The table below shows what the data should look like.

Write a query to return a table that matches the description provided.

Do not update the original table.

| Column Name | Criteria |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| product_id | Nominal. The unique identifier of the product. Missing values are not possible due to the database structure. |
| category | Nominal. The category of the product, one of 6 values (Housing, Food, Toys, Equipment, Medicine, Accessory). Missing values should be replaced with "Unknown". |
| animal | Nominal. The type of animal the product is for. One of Dog, Cat, Fish, Bird. Missing values should be replaced with "Unknown". |
| size | Ordinal. The size of animal the product is for. Small, Medium, Large. Missing values should be replaced with "Unknown". |
| price | Continuous. The price the product is sold at. Can be any positive value, round to 2 decimal places. Missing values should be replaced with 0. |
| sales | Continuous. The value of all sales of the product in the last year. This can be any positive value, rounded to 2 decimal places. Missing values should be replaced with the overall median sales. |
| rating | Discrete. Customer rating of the product from 1 to 10. Missing values should be replaced with 0. |
| repeat_purchase | Nominal. Whether customers repeatedly buy the product (1) or not (0). Missing values should be removed. |

Unknown integration

DataFrame as explore_category_data

SELECT DISTINCT category FROM pet_supplies;

| index | ... | ↑↓ | category | ... | ↑↓ | |
|-------|-----|----|-----------|-----|----|--|
| | | 0 | Medicine | | | |
| | | 1 | Food | | | |
| | | 2 | Equipment | | | |
| | | 3 | - | | | |
| | | 4 | Accessory | | | |
| | | 5 | Housing | | | |
| | | 6 | Toys | | | |

Rows: 7

Expand

Unknown integration

DataFrame as explore_animal_data

SELECT DISTINCT animal FROM pet_supplies;

| index | ... | ↑↓ | animal | ... | ↑↓ | |
|-------|-----|----|--------|-----|----|--|
| | | 0 | Fish | | | |
| | | 1 | Cat | | | |
| | | 2 | Bird | | | |
| | | 3 | Dog | | | |

Rows: 4

Expand

Unknown integration

DataFrame as explore_size_data

SELECT DISTINCT size FROM pet_supplies;

| index | ... | ↑↓ | size | ... | ↑↓ | |
|-------|-----|----|--------|-----|----|--|
| | | 0 | large | | | |
| | | 1 | medium | | | |
| | | 2 | Large | | | |
| | | 3 | Medium | | | |
| | | 4 | SMALL | | | |
| | | 5 | small | | | |
| | | 6 | LARGE | | | |
| | | 7 | MEDIUM | | | |
| | | 8 | Small | | | |

Rows: 9

Expand

Unknown integration

DataFrame as explore_rating_data

SELECT DISTINCT rating FROM pet_supplies;

| index | ... | ↑↓ | rating | ... | ↑↓ | |
|-------|-----|----|--------|-----|----|--|
| | | 0 | | | 8 | |
| | | 1 | | | 9 | |
| | | 2 | | | | |
| | | 3 | | | 7 | |
| | | 4 | | | 1 | |
| | | 5 | | | 5 | |
| | | 6 | | | 2 | |
| | | 7 | | | 4 | |
| | | 8 | | | 6 | |
| | | 9 | | | 3 | |

Rows: 10

Expand

Unknown integration DataFrame as clean_data

SELECT

```
product_id,
REPLACE(category, '-', 'Unknown') AS category,
COALESCE(animal, 'Unknown') AS animal,
COALESCE(INITCAP(size), 'Unknown') AS size,
TO_CHAR(CAST(REPLACE(price, 'unlisted', '0') AS DECIMAL(10, 2)), '999999999.99') AS price,
COALESCE(sales, CAST((SELECT PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY sales) FROM pet_supplies) AS DECIMAL(10, 2))) AS
sales,
COALESCE(rating, '0') AS rating,
repeat_purchase
FROM pet_supplies
WHERE product_id IS NOT NULL AND repeat_purchase IS NOT NULL;
```

| index | ... | ↑↓ | product_id | ... | ↑↓ | category | ... | ↑↓ | animal | ... | ↑↓ | size | ... | ↑↓ | price | ... | ↑↓ | sales | ... | ↑↓ |
|-------|-----|----|------------|-----|----|-----------|-----|----|--------|-----|----|--------|-----|----|-------|-----|----|---------|-----|----|
| | | 0 | | | 1 | Food | | | Bird | | | Large | | | 51.1 | | | 1860.62 | | |
| | | 1 | | | 2 | Housing | | | Bird | | | Medium | | | 35.98 | | | 963.6 | | |
| | | 2 | | | 3 | Food | | | Dog | | | Medium | | | 31.23 | | | 898.3 | | |
| | | 3 | | | 4 | Medicine | | | Cat | | | Small | | | 24.95 | | | 982.15 | | |
| | | 4 | | | 5 | Housing | | | Cat | | | Small | | | 26.18 | | | 832.63 | | |
| | | 5 | | | 6 | Housing | | | Dog | | | Small | | | 30.77 | | | 874.58 | | |
| | | 6 | | | 7 | Housing | | | Dog | | | Small | | | 31.04 | | | 875.07 | | |
| | | 7 | | | 8 | Toys | | | Cat | | | Medium | | | 28.9 | | | 1074.31 | | |
| | | 8 | | | 9 | Equipment | | | Fish | | | Medium | | | 17.82 | | | 503.67 | | |
| | | 9 | | | 10 | Medicine | | | Dog | | | Medium | | | 24.93 | | | 838.88 | | |
| | | 10 | | | 11 | Food | | | Dog | | | Large | | | 40.87 | | | 1457.22 | | |
| | | 11 | | | 12 | Medicine | | | Bird | | | Medium | | | 34.96 | | | 1204.6 | | |
| | | 12 | | | 13 | Food | | | Dog | | | Medium | | | 31.07 | | | 889.73 | | |
| | | 13 | | | 14 | Food | | | Dog | | | Large | | | 40.8 | | | 1450.8 | | |
| | | 14 | | | 15 | Accessory | | | Bird | | | Medium | | | 33.13 | | | 859.29 | | |
| | | 15 | | | 16 | Accessory | | | Bird | | | Large | | | 43.09 | | | 1418.72 | | |

Rows: 1,500

Expand

Task 2

You want to show whether sales are higher for repeat purchases for different animals. You also want to give a range for the sales.

Write a query to return the `animal`, `repeat_purchase` indicator and the `avg_sales`, along with the `min_sales` and `max_sales`. All values should be rounded to whole numbers.

You should use the original `pet_supplies` data for this task.

Unknown integration DataFrame as animal_sales

```
SELECT animal, repeat_purchase,
       AVG(sales) AS avg_sales,
       MIN(sales) AS min_sales,
       MAX(sales) AS max_sales
FROM pet_supplies
GROUP BY animal, repeat_purchase
ORDER BY animal, repeat_purchase;
```

| index | ... | ↑↓ | animal | ... | ↑↓ | repeat_purcha... | ... | ↑↓ | avg_sales | ... | ↑↓ | min_sales | ... | ↑↓ | max_sales | ... | ↑↓ |
|-------|-----|----|--------|-----|----|------------------|-----|----|-----------------|-----|----|-----------|-----|----|-----------|-----|----|
| | 0 | | Bird | | | | | 0 | 1380.2915116279 | | | 857.7 | | | 2254.96 | | |
| | 1 | | Bird | | | | | 1 | 1408.0623423423 | | | 852.94 | | | 2255.96 | | |
| | 2 | | Cat | | | | | 0 | 1035.2509865471 | | | 512.36 | | | 1729.76 | | |
| | 3 | | Cat | | | | | 1 | 998.1719767442 | | | 511.95 | | | 1724.15 | | |
| | 4 | | Dog | | | | | 0 | 1084.3280503145 | | | 574.02 | | | 1795.05 | | |
| | 5 | | Dog | | | | | 1 | 1038.4499519231 | | | 573.78 | | | 1797.05 | | |
| | 6 | | Fish | | | | | 0 | 705.3682539683 | | | 288.01 | | | 1307.35 | | |
| | 7 | | Fish | | | | | 1 | 692.9320987654 | | | 286.94 | | | 1301.35 | | |

Rows: 8

Expand

Task 3

The management team want to focus on efforts in the next year on the most popular pets - cats and dogs - for products that are bought repeatedly.

Write a query to return the `product_id`, `sales` and `rating` for the relevant products.

You should use the original `pet_supplies` data for this task.

Unknown integration DataFrame as popular_pet_products

```
SELECT product_id, sales, rating
FROM pet_supplies
WHERE animal IN ('Cat', 'Dog') AND repeat_purchase = '1'
ORDER BY sales DESC;
```

| index | ... | ↑↓ | product_id | ... | ↑↓ | sales | ... | ↑↓ | rating | ... | ↑↓ |
|-------|-----|----|------------|-----|------|-------|-----|---------|--------|-----|----|
| | 0 | | | | 518 | | | 1797.02 | | | 7 |
| | 1 | | | | 280 | | | 1795.77 | | | 5 |
| | 2 | | | | 728 | | | 1793.71 | | | 6 |
| | 3 | | | | 20 | | | 1792.63 | | | 7 |
| | 4 | | | | 946 | | | 1788.28 | | | 8 |
| | 5 | | | | 863 | | | 1724.15 | | | 7 |
| | 6 | | | | 1383 | | | 1723.87 | | | 8 |
| | 7 | | | | 272 | | | 1470.65 | | | 6 |
| | 8 | | | | 561 | | | 1469.55 | | | 5 |
| | 9 | | | | 285 | | | 1467.21 | | | 4 |
| | 10 | | | | 752 | | | 1466.78 | | | 6 |
| | 11 | | | | 332 | | | 1466.3 | | | 6 |
| | 12 | | | | 152 | | | 1464.8 | | | 4 |
| | 13 | | | | 1061 | | | 1463.58 | | | 7 |
| | 14 | | | | 88 | | | 1463.29 | | | 4 |
| | 15 | | | | 370 | | | 1462.67 | | | |

Rows: 552

Expand