jupyter Untitled12 Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[1]:
```
import pandas as pd
import numpy as np
import sqlite3 as sql
import matplotlib.pyplot as plt
from ydata_profiling import ProfileReport
```

[2]:
```
brands = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\brands.csv")
categories = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\categories.csv")
customers = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\customers.csv")
order_items = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\order_items.csv")
orders = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\orders.csv")
products = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\products.csv")
staffs = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\staffs.csv")
stocks = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\stocks.csv")
stores = pd.read_csv(r"C:\Users\20109\Downloads\Data_set\stores.csv")
```

[3]:
```
db = sql.connect("bike_store.db")
```

[4]:
```
print("Number of rows in brands: ", brands.to_sql("brands", db, if_exists = 'replace', index = 'False'))
print("Number of rows in categories: ", categories.to_sql("categories", db, if_exists = 'replace', index = 'False'))
print("Number of rows in customers: ", customers.to_sql("customers", db, if_exists = 'replace', index = 'False'))
print("Number of rows in order_items: ", order_items.to_sql("order_items", db, if_exists = 'replace', index = 'False'))
print("Number of rows in orders: ", orders.to_sql("orders", db, if_exists = 'replace', index = 'False'))
print("Number of rows in products: ", products.to_sql("products", db, if_exists = 'replace', index = 'False'))
print("Number of rows in staffs: ", staffs.to_sql("staffs", db, if_exists = 'replace', index = 'False'))
print("Number of rows in stocks: ", stocks.to_sql("stocks", db, if_exists = 'replace', index = 'False'))
print("Number of rows in stores: ", stores.to_sql("stores", db, if_exists = 'replace', index = 'False'))
```

```
Number of rows in customers:  1445
Number of rows in order_items:  4722
Number of rows in orders:  1615
Number of rows in products:  321
Number of rows in staffs:  10
Number of rows in stocks:  939
Number of rows in stores:  3
```

[5]:
```
q1 = """
select * from brands
"""
pd.read_sql(q1, db)
```

[5]:

	False	brand_id	brand_name
0	0	1	Electra
1	1	2	Haro
2	2	3	Heller
3	3	4	Pure Cycles
4	4	5	Ritchey
5	5	6	Strider
6	6	7	Sun Bicycles
8	8	9	Trek

[6]:
```
brands.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   brand_id    9 non-null      int64 
 1   brand_name  9 non-null      object
dtypes: int64(1), object(1)
memory usage: 276.0+ bytes
```

[7]:
```
brands.describe().round(2)
```

[7]:

	brand_id
count	9.00
mean	5.00
std	2.74
min	1.00
25%	3.00
50%	5.00
max	9.00

[8]:
```
brands['brand_name'].nunique()
```

[8]:
```
9
```

[9]:
```
brands['brand_name'].unique()
```

[9]:
```
array(['Electra', 'Haro', 'Heller', 'Pure Cycles', 'Ritchey', 'Strider',
       'Sun Bicycles', 'Surly', 'Trek'], dtype=object)
```

[10]:
```
brands.isna().sum()
```

[10]:
```
brand_id      0
```

```
brand_name    0
dtype: int64
```

[11]:
```
brands['brand_name'].value_counts()
```

[11]:
```
brand_name
Electra         1
Haro            1
Heller          1
Pure Cycles     1
Ritchey         1
Strider         1
Trek            1
Name: count, dtype: int64
```

[12]:
```
len(brands['brand_name'].value_counts())
```

[12]:
```
9
```

[13]:
```
brands.duplicated().sum()
```

[13]:
```
0
```

[14]:
```
brands.shape
```

[14]:
```
(9, 2)
```

[15]:
```
q1 = """
select * from categories
"""
pd.read_sql(q1, db)
```

[15]:

	False	category_id	category_name
0	0	1	Children Bicycles
1	1	2	Comfort Bicycles
2	2	3	Cruisers Bicycles
4	4	5	Electric Bikes
5	5	6	Mountain Bikes
6	6	7	Road Bikes

[16]:
```
categories.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   category_id    7 non-null      int64
 1   category_name  7 non-null      object
dtypes: int64(1), object(1)
memory usage: 244.0+ bytes
```

[17]:
```
categories.describe().round(2)
```

[17]:

	category_id
count	7.00
mean	4.00
std	2.16
25%	2.50
50%	4.00
75%	5.50
max	7.00

[18]:
```
categories['category_name'].nunique()
```

[18]:
```
7
```

[19]:
```
categories['category_name'].unique()
```

[19]:
```
array(['Children Bicycles', 'Comfort Bicycles', 'Cruisers Bicycles',
       'Cyclocross Bicycles', 'Electric Bikes', 'Mountain Bikes',
       'Road Bikes'], dtype=object)
```

[20]:
```
categories.isna().sum()
```

[20]:
```
category_id      0
category_name    0
dtype: int64
```

[21]:
```
categories['category_name'].value_counts()
```

[21]:
```
category_name
Children Bicycles       1
Comfort Bicycles        1
Cruisers Bicycles       1
Cyclocross Bicycles     1
Electric Bikes          1
Mountain Bikes          1
Road Bikes              1
Name: count, dtype: int64
```

[22]:
```
len(categories['category_name'].value_counts())
```

[22]:
```
7
```

[23]:
```
categories.duplicated().sum()
```

[23]:
```
0
```

[24]:
```
categories.shape
```

[24]:
```
categories.shape
```

[24]: (7, 2)

[25]:
```
q1 = """
select * from customers
"""
pd.read_sql(q1, db)
```

[25]:

	False	customer_id	first_name	last_name	phone	email	street	city	state	zip_code
?	?	2	Kasha	Todd	None	kasha.todd@yahoo.com	[illegible]	Campbell	CA	95008
2	2	3	Tameka	Fisher	None	tameka.fisher@aol.com	769C Honey Creek St.	Redondo Beach	CA	90278
3	3	4	Daryl	Spence	None	daryl.spence@aol.com	988 Pearl Lane	Uniondale	NY	11553
4	4	5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.	Sacramento	CA	95820
...	...	...	...	...	...	...	...	...	...	...
1440	1440	1441	Jamaal	Morrison	None	jamaal.morrison@msn.com	796 SE. Nut Swamp St.	Staten Island	NY	10301
1441	1441	1442	Cassie	Cline	None	cassie.cline@gmail.com	947 Lafayette Drive	Brooklyn	NY	11201
1442	1442	1443	Lezlie	Lamb	None	lezlie.lamb@gmail.com	401 Brandywine Street	Central Islip	NY	11722
1443	1443	1444	Ivette	Estes	None	ivette.estes@gmail.com	88 N. Canterbury Ave.	Canandaigua	NY	14424
1444	1444	1445	Ester	Acevedo	None	ester.acevedo@gmail.com	671 Miles Court	San Lorenzo	CA	94580

1445 rows × 10 columns

[26]:
```
customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1445 entries, 0 to 1444
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 2   last_name    1445 non-null   object
 3   phone        178 non-null    object
 4   email        1445 non-null   object
 5   street       1445 non-null   object
 6   city         1445 non-null   object
 7   state        1445 non-null   object
 8   zip_code     1445 non-null   int64 
dtypes: int64(2), object(7)
memory usage: 101.7+ KB
```

[27]:
```
customers['phone']=customers['phone'].fillna('No Data')
```

[28]:
```
customers['phone'].sample(10)
```

[28]:
```
1148           No Data
1026           No Data
1227    (510) 565-8496
584            No Data
1183           No Data
165            No Data
501            No Data
1081    (562) 264-3998
1260           No Data
34             No Data
Name: phone, dtype: object
```

[29]:
```
customers.describe().round(2)
```

count	1445.00	1445.00
mean	723.00	34200.02
std	417.28	34733.93
min	1.00	10002.00
25%	362.00	11419.00
50%	723.00	11967.00
75%	1084.00	76110.00
max	1445.00	95993.00

[30]:
```
customers[['city','state']].value_counts()
```

[30]:
```
city           state
Mount Vernon   NY       20
Ballston Spa   NY       17
Scarsdale      NY       17
Canandaigua    NY       14
Ossining       NY       13
                        ..
Far Rockaway   NY        2
Tonawanda      NY        1
Middle Village NY        1
Name: count, Length: 195, dtype: int64
```

[31]:
```
# Generate a profile report for the DataFrame df
profile = ProfileReport(customers)

# Display the report in the notebook
profile
```

Error displaying widget

Error displaying widget

Error displaying widget

Pandas Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview

Brought to you by YData

Overview Alerts 9 Reproduction

Dataset statistics

Number of variables	9
Number of observations	1445
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	101.7 KiB
Average record size in memory	72.1 B

Variable types

Numeric	2
Text	6
Categorical	1

Variables

[31]:

[32]:
```
customers.shape
```

[32]:
```
(1445, 9)
```

[33]:
```
q1 = """
select * from order_items
"""
pd.read_sql(q1, db)
```

[33]:

	False	order_id	item_id	product_id	quantity	list_price	discount
0	0	1	1	20	1	599.99	0.20
1	1	1	2	8	2	1799.99	0.07
2	2	1	3	10	2	1549.00	0.05
3	3	1	4	16	2	599.99	0.05
4	4	1	5	4	1	2899.99	0.20
...	...	...	...	...	...	...	...
4717	4717	1614	2	159	2	2299.99	0.07
4719	4719	1615	1	237	2	2259.99	0.20
4720	4720	1615	2	214	1	899.99	0.07
4721	4721	1615	3	182	1	2499.99	0.20

4722 rows × 7 columns

[34]:
```
order_items.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4722 entries, 0 to 4721
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   order_id    4722 non-null   int64  
 1   item_id     4722 non-null   int64  
 2   product_id  4722 non-null   int64  
 3   quantity    4722 non-null   int64  
 4   list_price  4722 non-null   float64
 5   discount    4722 non-null   float64
dtypes: float64(2), int64(4)
memory usage: 221.5 KB
```

[35]:
```
order_items.describe().round(2)
```

[35]:

	order_id	item_id	product_id	quantity	list_price	discount
mean	821.27	2.26	59.37	1.5	1212.71	0.11
std	465.15	1.20	67.31	0.5	1352.80	0.06
min	1.00	1.00	2.00	1.0	89.99	0.05
25%	423.25	1.00	14.00	1.0	429.00	0.05
50%	828.50	2.00	28.00	1.0	599.99	0.08
75%	1226.00	3.00	84.00	2.0	1549.00	0.20
max	1615.00	5.00	315.00	2.0	11999.99	0.20

[36]:
```
def data_summary(df):
    # Calculate metrics
```

```
    unique_values = df.nunique()
    missing_values = df.isnull().sum()
    missing_percentage = (df.isnull().sum() / len(df)) * 100
    duplicate_rows = df.duplicated().sum()
    data_types = df.dtypes

    # Create summary DataFrame
    summary_df = pd.DataFrame({
        'Unique Values': unique_values,
        'Missing Values': missing_values,
        'Missing Values (%)': missing_percentage,
    })

    return summary_df
summary = data_summary(order_items)
print(summary)
```

```
            Unique Values  Missing Values  Missing Values (%)  Duplicate Rows  \
order_id             1615               0                 0.0               0
item_id                 5               0                 0.0               0
product_id            307               0                 0.0               0
quantity                2               0                 0.0               0
list_price            104               0                 0.0               0
discount                4               0                 0.0               0

           Data Type
order_id       int64
item_id        int64
product_id     int64
quantity       int64
list_price   float64
discount     float64
```

[37]:
```
order_items.shape
```

[37]:
```
(4722, 6)
```

[38]:
```
q1 = """
pd.read_sql(q1, db)
```

[38]:

	False	order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
0	0	1	259	4	2016-01-01	2016-01-03	2016-01-03	1	2
1	1	2	1212	4	2016-01-01	2016-01-04	2016-01-03	2	6
2	2	3	523	4	2016-01-02	2016-01-05	2016-01-03	2	7
3	3	4	175	4	2016-01-03	2016-01-04	2016-01-05	1	3
4	4	5	1324	4	2016-01-03	2016-01-06	2016-01-06	2	6
...	...	...	...	...	...	...	...	...	...
1610	1610	1611	6	3	2018-09-06	2018-09-06	None	2	7
1611	1611	1612	3	3	2018-10-21	2018-10-21	None	1	3
1612	1612	1613	1	3	2018-11-18	2018-11-18	None	2	6
1613	1613	1614	135	3	2018-11-28	2018-11-28	None	3	8
1614	1614	1615	136	3	2018-12-28	2018-12-28	None	3	8

1615 rows × 9 columns

[39]:
```
orders.info()
```

```
RangeIndex: 1615 entries, 0 to 1614
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   order_id       1615 non-null   int64
 1   customer_id    1615 non-null   int64
 2   order_status   1615 non-null   int64
 3   order_date     1615 non-null   object
 4   required_date  1615 non-null   object
 5   shipped_date   1445 non-null   object
 6   store_id       1615 non-null   int64
 7   staff_id       1615 non-null   int64
dtypes: int64(5), object(3)
memory usage: 101.1+ KB
```

[40]:
```
orders['shipped_date'] = orders['shipped_date'].fillna('No Date')
```

[41]:
```
orders['shipped_date'].sample(10)
```

[41]:
```
774     2017-03-13
1346    2018-01-18
400     2016-09-01
297     2016-07-07
1261    2017-11-26
913     2017-05-31
1190    2017-10-18
834     2017-04-14
1360    2018-01-22
Name: shipped_date, dtype: object
```

[42]:
```
orders.describe().round(2)
```

[42]:

	order_id	customer_id	order_status	store_id	staff_id
count	1615.00	1615.00	1615.00	1615.00	1615.00
mean	808.00	654.17	3.78	1.89	5.86
std	466.35	443.23	0.69	0.56	1.91
min	1.00	1.00	1.00	1.00	2.00
25%	404.50	237.00	4.00	2.00	6.00
50%	808.00	638.00	4.00	2.00	6.00

75%	1211.50	1041.50	4.00	2.00	7.00
max	1615.00	1445.00	4.00	3.00	9.00

[43]:
```
summary = data_summary(orders)
print(summary)
```

```
               Unique Values  Missing Values  Missing Values (%)  \
order_id                1615               0                 0.0
customer_id             1445               0                 0.0
order_status               4               0                 0.0
shipped_date             676               0                 0.0
store_id                   3               0                 0.0
staff_id                   6               0                 0.0

               Duplicate Rows Data Type
order_id                    0     int64
customer_id                 0     int64
order_status                0     int64
order_date                  0    object
required_date               0    object
shipped_date                0    object
store_id                    0     int64
staff_id                    0     int64
```

[44]:
```
orders.shape
```

[44]:
```
(1615, 8)
```

[45]:
```
q1 = """
select * from products
"""
pd.read_sql(q1, db)
```

[45]:

	False	product_id	product_name	brand_id	category_id	model_year	list_price
0	0	1	Trek 820 - 2016	9	6	2016	379.99
2	2	3	[illegible] Frameset - 2016	9	6	2016	999.99
3	3	4	Trek Fuel EX 8 29 - 2016	9	6	2016	2899.99
4	4	5	Heller Shagamaw Frame - 2016	3	6	2016	1320.99
...	...	...	...	...	...	...	...
316	316	317	Trek Checkpoint ALR 5 - 2019	9	7	2019	1999.99
317	317	318	Trek Checkpoint ALR 5 Women's - 2019	9	7	2019	1999.99
318	318	319	Trek Checkpoint SL 5 Women's - 2019	9	7	2019	2799.99
319	319	320	Trek Checkpoint SL 6 - 2019	9	7	2019	3799.99
320	320	321	Trek Checkpoint ALR Frameset - 2019	9	7	2019	3199.99

321 rows × 7 columns

[46]:
```
products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 321 entries, 0 to 320
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   product_id    321 non-null    int64
 1   product_name  321 non-null    object
 4   model_year    321 non-null    int64
 5   list_price    321 non-null    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 15.2+ KB
```

[47]:
```
products.describe().round(2)
```

[47]:

	product_id	brand_id	category_id	model_year	list_price
count	321.00	321.00	321.00	321.00	321.00
mean	161.00	5.48	4.03	2017.59	1520.59
std	92.81	3.71	2.19	0.67	1612.15
min	1.00	1.00	1.00	2016.00	89.99
25%	81.00	1.00	2.00	2017.00	439.99
50%	161.00	7.00	3.00	2018.00	761.99
75%	241.00	9.00	6.00	2018.00	2299.99
max	321.00	9.00	7.00	2019.00	11999.99

[48]:
```
products['product_name'].value_counts()
```

[48]:
```
product_name
Electra Townie Original 7D - 2017            2
Electra Townie Commute 27D Ladies - 2018     2
Electra Townie Commute 8D Ladies' - 2018     2
                                            ..
Trek Ticket S Frame - 2018                   1
Trek X-Caliber 8 - 2018                      1
Trek Kids' Neko - 2018                       1
Trek Fuel EX 7 29 - 2018                     1
Trek Checkpoint ALR Frameset - 2019          1
Name: count, Length: 291, dtype: int64
```

[49]:
```
summary = data_summary(products)
print(summary)
```

```
              Unique Values  Missing Values  Missing Values (%)  \
product_id              321               0                 0.0
```

```
product_name            291              0               0.0
brand_id                  9              0               0.0
category_id               7              0               0.0
model_year                4              0               0.0
list_price              106              0               0.0

             Duplicate Rows Data Type
product_id                0     int64
product_name              0    object
brand_id                  0     int64
category_id               0     int64
model_year                0     int64
```

[50]:
```
q1 = """
select * from staffs
"""
pd.read_sql(q1, db)
```

[50]:

	False	staff_id	first_name	last_name	email	phone	active	store_id	manager_id
0	0	1	Fabiola	Jackson	fabiola.jackson@bikes.shop	(831) 555-5554	1	1	NaN
1	1	2	Mireya	Copeland	mireya.copeland@bikes.shop	(831) 555-5555	1	1	1.0
2	2	3	Genna	Serrano	genna.serrano@bikes.shop	(831) 555-5556	1	1	2.0
3	3	4	Virgie	Wiggins	virgie.wiggins@bikes.shop	(831) 555-5557	1	1	2.0
4	4	5	Jannette	David	jannette.david@bikes.shop	(516) 379-4444	1	2	1.0
5	5	6	Marcelene	Boyer	marcelene.boyer@bikes.shop	(516) 379-4445	1	2	5.0
6	6	7	Venita	Daniel	venita.daniel@bikes.shop	(516) 379-4446	1	2	5.0
7	7	8	Kali	Vargas	kali.vargas@bikes.shop	(972) 530-5555	1	3	1.0
8	8	9	Layla	Terrell	layla.terrell@bikes.shop	(972) 530-5556	1	3	7.0
9	9	10	Bernardine	Houston	bernardine.houston@bikes.shop	(972) 530-5557	1	3	7.0

[51]:
```
staffs.info()
```
```
RangeIndex: 10 entries, 0 to 9
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   staff_id    10 non-null     int64
 1   first_name  10 non-null     object
 2   last_name   10 non-null     object
 3   email       10 non-null     object
 4   phone       10 non-null     object
 5   active      10 non-null     int64
 6   store_id    10 non-null     int64
 7   manager_id  9 non-null      float64
dtypes: float64(1), int64(3), object(4)
memory usage: 772.0+ bytes
```

[52]:
```
staffs['manager_id'] = staffs['manager_id'].fillna('0')
```

[53]:
```
staffs.describe().round(2)
```

[53]:

	staff_id	active	store_id
count	10.00	10.0	10.00
mean	5.50	1.0	1.90
std	3.03	0.0	0.88
min	1.00	1.0	1.00
50%	5.50	1.0	2.00
75%	7.75	1.0	2.75
max	10.00	1.0	3.00

[54]:
```
summary = data_summary(staffs)
print(summary)
```
```
            Unique Values  Missing Values  Missing Values (%)  Duplicate Rows  \
staff_id               10               0                 0.0               0
first_name             10               0                 0.0               0
last_name              10               0                 0.0               0
email                  10               0                 0.0               0
phone                  10               0                 0.0               0
active                  1               0                 0.0               0
store_id                3               0                 0.0               0
manager_id              5               0                 0.0               0

           Data Type
staff_id       int64
first_name    object
last_name     object
email         object
phone         object
active         int64
store_id       int64
manager_id    object
```

[55]:
```
(10, 8)
```

[56]:
```
q1 = """
select * from stocks
"""
pd.read_sql(q1, db)
```

[56]:

	False	store_id	product_id	quantity
0	0	1	1	27
1	1	1	2	5
2	2	1	3	6

3	3	1	4	23
4	4	1	5	22
...	...	...	...	...
934	934	3	309	30
935	935	3	310	8
936	936	3	311	23
937	937	3	312	18

939 rows × 4 columns

[57]:
```
stocks.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 939 entries, 0 to 938
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   store_id    939 non-null    int64
 1   product_id  939 non-null    int64
 2   quantity    939 non-null    int64
dtypes: int64(3)
memory usage: 22.1 KB
```

[58]:
```
stocks.describe().round(2)
```

[58]:

	store_id	product_id	quantity
count	939.00	939.0	939.00
mean	2.00	157.0	14.39
std	0.82	90.4	8.83
min	1.00	1.0	0.00
50%	2.00	157.0	14.00
75%	3.00	235.0	22.00
max	3.00	313.0	30.00

[59]:
```
summary = data_summary(stocks)
print(summary)
```

```
            Unique Values  Missing Values  Missing Values (%)  Duplicate Rows  \
store_id                3               0                 0.0               0
product_id            313               0                 0.0               0
quantity               31               0                 0.0               0

           Data Type
store_id       int64
product_id     int64
quantity       int64
```

[60]:
```
stocks.shape
```

[60]:
```
(939, 3)
```

[61]:
```
q1 = """
select * from stores
"""
pd.read_sql(q1, db)
```

[61]:

	index	store_id	store_name	phone	email	street	city	state	zip_code
0	0	1	Santa Cruz Bikes	(831) 476-4321	santacruz@bikes.shop	3700 Portola Drive	Santa Cruz	CA	95060
1	1	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
2	2	3	Rowlett Bikes	(972) 530-5555	rowlett@bikes.shop	8000 Fairway Avenue	Rowlett	TX	75088

[62]:
```
stores.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   store_id    3 non-null      int64
 1   store_name  3 non-null      object
 2   phone       3 non-null      object
 3   email       3 non-null      object
 4   street      3 non-null      object
 5   city        3 non-null      object
 6   state       3 non-null      object
 7   zip_code    3 non-null      int64
dtypes: int64(2), object(6)
memory usage: 324.0+ bytes
```

[63]:
```
stores.describe().round(2)
```

count	3.0	3.00
mean	2.0	60526.67
std	1.0	43674.19
min	1.0	11432.00
25%	1.5	43260.00
50%	2.0	75088.00
75%	2.5	85074.00
max	3.0	95060.00

[64]:
```
stores['store_name'].value_counts()
```

[64]:
```
store_name
Santa Cruz Bikes    1
Baldwin Bikes       1
Rowlett Bikes       1
Name: count, dtype: int64
```

[65]:
```
# Generate a profile report for the DataFrame df
profile = ProfileReport(stores)

# Display the report in the notebook
```

Error displaying widget

Error displaying widget

Error displaying widget

Pandas Profiling Report — Overview | Variables | Correlations | Missing values | Sample

Overview

Brought to you by YData

Overview | Alerts 12 | Reproduction

Dataset statistics

Number of variables	8
Number of observations	3
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	324.0 B
Average record size in memory	108.0 B

Variable types

Categorical	2
Text	6

Variables

Select Columns

[65]:

[66]:
```
stores.shape
```

[66]:
```
(3, 8)
```

[68]:
```
categories.to_csv(r"C:\Users\20109\Downloads\Data_set\categories-Mif.csv")
```

[69]:
```
customers.to_csv(r"C:\Users\20109\Downloads\Data_set\customers-Mif.csv")
```

[70]:
```
order_items.to_csv(r"C:\Users\20109\Downloads\Data_set\order_items-Mif.csv")
```

[76]:
```
orders.to_csv(r"C:\Users\20109\Downloads\Data_set\orders-Mif.csv")
```

[78]:
```
products.to_csv(r"C:\Users\20109\Downloads\Data_set\products-Mif.csv")
```

[80]:
```
staffs.to_csv(r"C:\Users\20109\Downloads\Data_set\staffs-Mif.csv")
```

[82]:
```
stocks.to_csv(r"C:\Users\20109\Downloads\Data_set\stocks-Mif.csv")
```

[84]:
```
stores.to_csv(r"C:\Users\20109\Downloads\Data_set\stores-Mif.csv")
```

[ ]: