Code & Output

EX 7.3

Code:

```
# Julia Cuellar
# DSC 550
# Final project

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# Display pizza place data
def read_file():
    pizza = pd.read_csv('pizzaplace.csv')
    print('Original pizza data:\n', pizza)


# Display described, summarized, and length of pizza place data
def des_sum_len():
    pizza = pd.read_csv('pizzaplace.csv')
    print('Described pizza data:\n', pizza.describe())
    print('Summarized pizza data:\n', pizza.describe(include=['O']))
    print('Length of pizza data:\n', len(pizza))


# Display bar chart of pizza name
def showBar_Pname():
    pizza = pd.read_csv('pizzaplace.csv')
    pizza['name'].value_counts().plot(kind='barh').invert_yaxis()
    plt.title('Pizza name')
    plt.show()


# Display bar chart of pizza size
def showBar_Psize():
    pizza = pd.read_csv('pizzaplace.csv')
    pizza['size'].value_counts().plot(kind='barh')
    plt.title('Pizza size')
    plt.show()


# Display pie chart of pizza type
def showPie_Ptype():
    pizza = pd.read_csv('pizzaplace.csv')
    plt.pie(pizza['type'].value_counts(), autopct=lambda p: f'{p:.2f}%',
labels=['classic', 'supreme', 'veggie',

'chicken'])
    plt.title('Pizza type')
    plt.show()
```

```
# Display boxplot of pizza price
def showBoxplot_Pprice():
    pizza = pd.read_csv('pizzaplace.csv')
    sns.boxplot(pizza['price'])
    plt.title('Pizza price')
    plt.show()


# Check the nulls from pizza file
def check_null():
    pizza = pd.read_csv('pizzaplace.csv')
    print("Display pizza data with null:\n", pizza.isnull())
    print("Display counts of null from pizza data:\n", pizza.isnull().sum())


# Rename unname column then drop along with id and date
def rename_drop():
    pizza = pd.read_csv('pizzaplace.csv')
    pizza.rename(columns={'Unnamed: 0': 'num'}, inplace=True)
    pizza.drop(['num', 'id', 'date'], axis=1, inplace=True)
    print('Pizza data updated:\n', pizza.head(5))


# Check for outlier in pizza size column by counts
def size_count():
    pizza = pd.read_csv('pizzaplace.csv')
    pizza.rename(columns={'Unnamed: 0': 'num'}, inplace=True)
    pizza.drop(['num', 'id', 'date'], axis=1, inplace=True)
    print('Pizza size count:\n', pizza['size'].value_counts())


# Check for outlier in pizza price column by describe then remove and update
def price_out_r_up():
    pizza = pd.read_csv('pizzaplace.csv')
    pizza.rename(columns={'Unnamed: 0': 'num'}, inplace=True)
    pizza.drop(['num', 'id', 'date'], axis=1, inplace=True)
    print('Pizza price:\n', pizza['price'].describe())
    p_price = pizza[pizza['price'] >= 35].index
    pizza.drop(p_price, inplace=True)
    print('Described pizza price:\n', pizza['price'].describe())
    sns.boxplot(pizza['price'])
    plt.title('Pizza price updated')
    plt.show()


# Redisplay pizza place data with described, summarized, and length
def pizza_up():
    pizza = pd.read_csv('pizzaplace.csv')
    pizza.rename(columns={'Unnamed: 0': 'num'}, inplace=True)
    pizza.drop(['num', 'id', 'date'], axis=1, inplace=True)
    p_price = pizza[pizza['price'] >= 35].index
    pizza.drop(p_price, inplace=True)
    print('Pizza data updated:\n', pizza)
    print('Described pizza data updated:\n', pizza.describe())
    print('Summarized pizza data updated:\n', pizza.describe(include=['O']))
    print('Length of pizza data updated:\n', len(pizza))
```

```
if __name__ == "__main__":
    read_file()
    des_sum_len()
    showBar_Pname()
    showBar_Psize()
    showPie_Ptype()
    showBoxplot_Pprice()
    check_null()
    rename_drop()
    size_count()
    price_out_r_up()
    pizza_up()
```

Output:

Original pizza data:

```
      Unnamed: 0         id       date  ... size     type  price
0              1  2015-000001  2015-01-01  ...    M  classic  13.25
1              2  2015-000002  2015-01-01  ...    M  classic  16.00
2              3  2015-000002  2015-01-01  ...    M   veggie  16.00
3              4  2015-000002  2015-01-01  ...    L  chicken  20.75
4              5  2015-000002  2015-01-01  ...    L   veggie  18.50
...          ...         ...         ...  ... ...  ...     ...    ...
49569      49570  2015-021348  2015-12-31  ...    L   veggie  17.95
49570      49571  2015-021348  2015-12-31  ...    S  classic  12.00
49571      49572  2015-021348  2015-12-31  ...    M  chicken  16.75
49572      49573  2015-021349  2015-12-31  ...    L   veggie  20.25
49573      49574  2015-021350  2015-12-31  ...    S  chicken  12.75

[49574 rows x 8 columns]
```

Described pizza data:

```
         Unnamed: 0         price
count  49574.000000  49574.000000
mean   24787.500000     16.497762
std    14310.925459      3.621954
min        1.000000      9.750000
25%    12394.250000     12.750000
50%    24787.500000     16.500000
75%    37180.750000     20.250000
max    49574.000000     35.950000
```

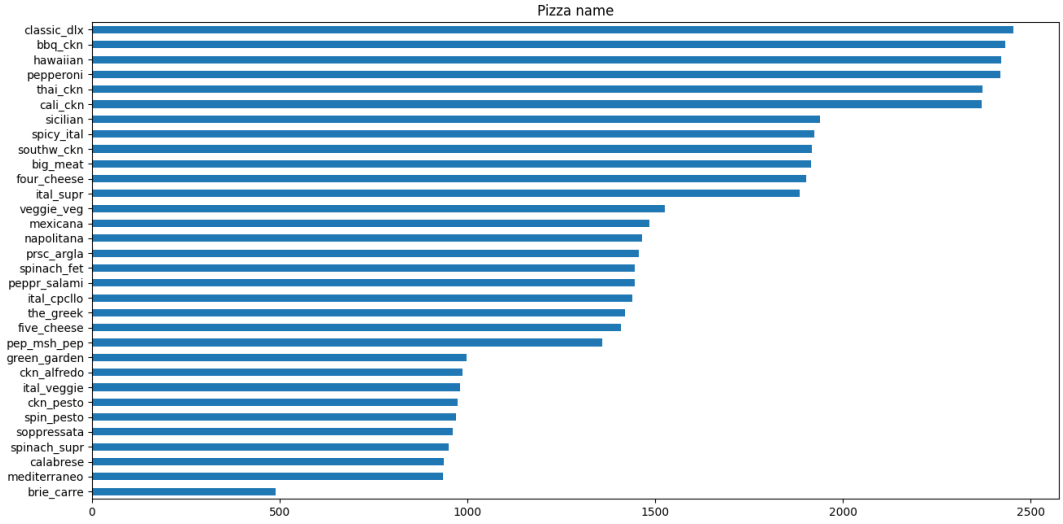
Summarized pizza data:

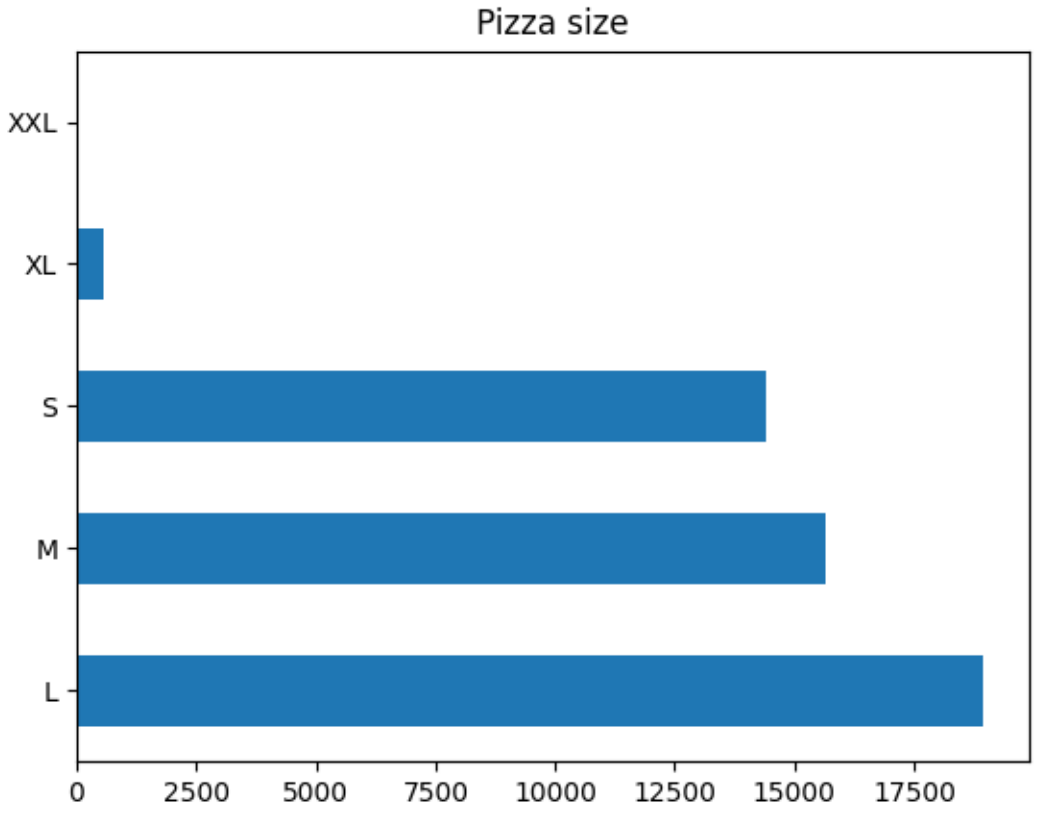
```
                id       date      time         name   size     type
```

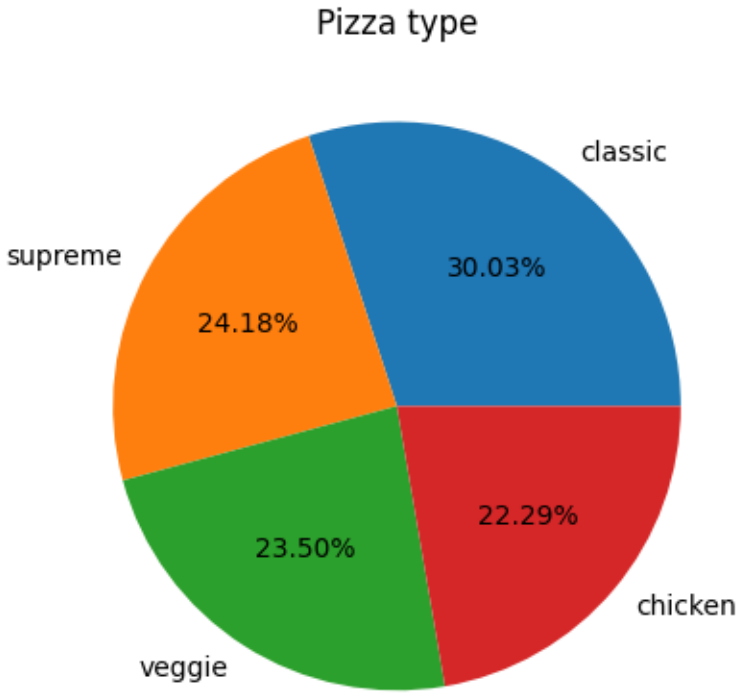
count	49574	49574	49574	49574	49574	49574
unique	21350	358	16382	32	5	4
top	2015-018845	2015-11-26	12:25:12	classic_dlx	L	classic
freq	28	266	28	2453	18956	14888

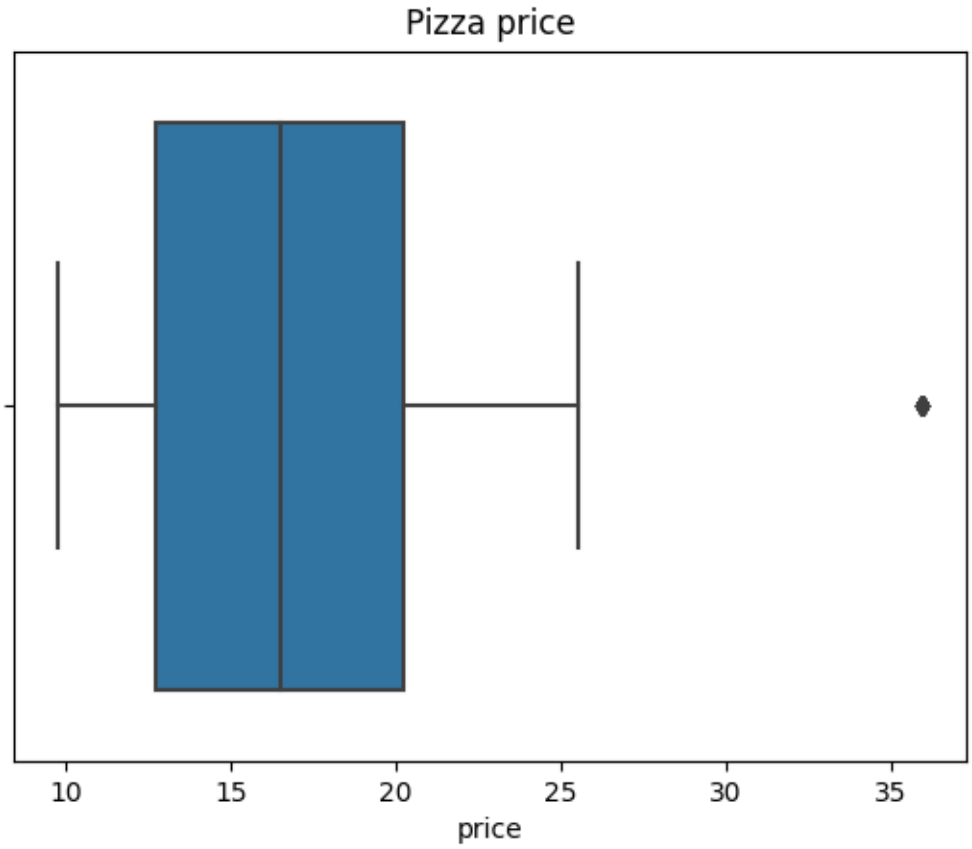
Length of pizza data:

49574




Pizza size
XXL
XL
S
M
L
0
2500
5000
7500
10000
12500
15000
17500


Pizza type
classic
supreme
30.03%
24.18%
22.29%
23.50%
chicken
veggie



Display pizza data with null:

	Unnamed: 0	id	date	time	name	size	type	price
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
49569	False	False	False	False	False	False	False	False
49570	False	False	False	False	False	False	False	False
49571	False	False	False	False	False	False	False	False
49572	False	False	False	False	False	False	False	False
49573	False	False	False	False	False	False	False	False

[49574 rows x 8 columns]

Display counts of null from pizza data:

```
 Unnamed: 0    0
id             0
date           0
time           0
name           0
size           0
type           0
price          0
dtype: int64
```

Pizza data updated:

	time	name	size	type	price
0	11:38:36	hawaiian	M	classic	13.25
1	11:57:40	classic_dlx	M	classic	16.00
2	11:57:40	mexicana	M	veggie	16.00
3	11:57:40	thai_ckn	L	chicken	20.75
4	11:57:40	five_cheese	L	veggie	18.50

Pizza size count:

```
 L      18956
M       15635
S       14403
XL        552
XXL        28
Name: size, dtype: int64
```

Pizza price:

```
 count    49574.000000
```

mean 16.497762

std 3.621954

min 9.750000

25% 12.750000

50% 16.500000

75% 20.250000

max 35.950000

Name: price, dtype: float64

Described pizza price:

count 49546.000000

mean 16.486769

std 3.593327

min 9.750000

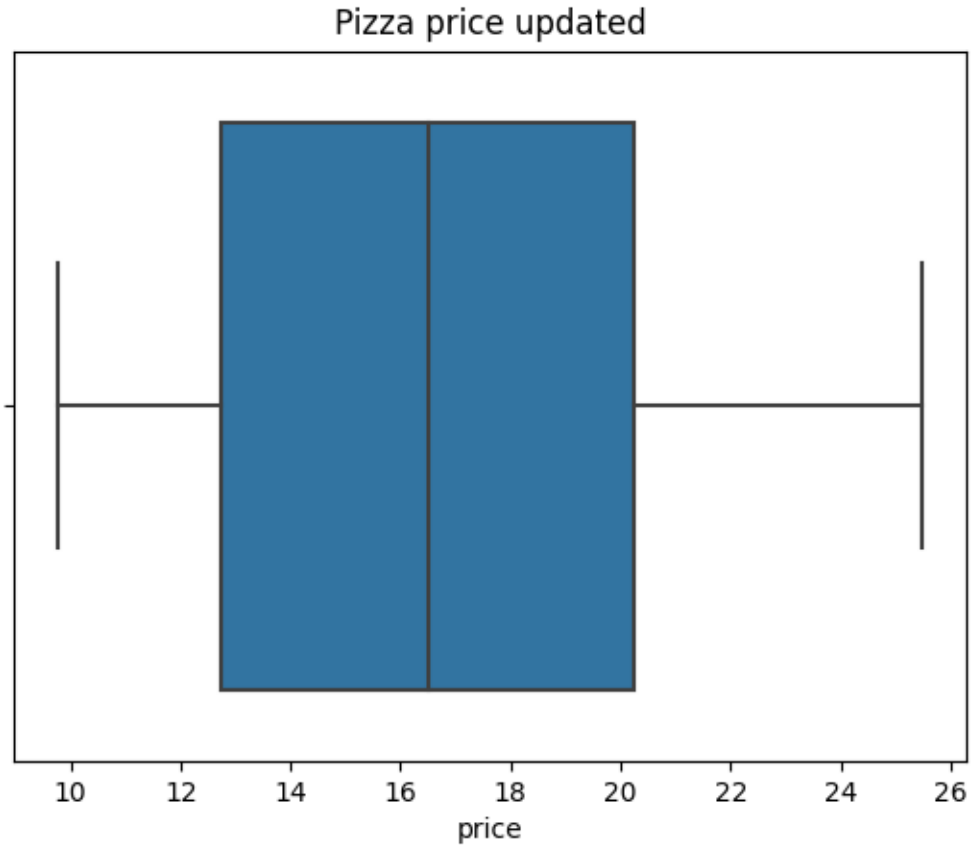
25% 12.750000

50% 16.500000

75% 20.250000

max 25.500000

Name: price, dtype: float64



Pizza data updated:

	time	name	size	type	price
0	11:38:36	hawaiian	M	classic	13.25
1	11:57:40	classic_dlx	M	classic	16.00
2	11:57:40	mexicana	M	veggie	16.00
3	11:57:40	thai_ckn	L	chicken	20.75
4	11:57:40	five_cheese	L	veggie	18.50
...	...	...	...	...	...
49569	21:23:10	four_cheese	L	veggie	17.95
49570	21:23:10	napolitana	S	classic	12.00
49571	21:23:10	ckn_alfredo	M	chicken	16.75
49572	22:09:54	mexicana	L	veggie	20.25
49573	23:02:05	bbq_ckn	S	chicken	12.75

[49546 rows x 5 columns]

Described pizza data updated:

	price
count	49546.000000
mean	16.486769
std	3.593327
min	9.750000
25%	12.750000
50%	16.500000
75%	20.250000
max	25.500000

Summarized pizza data updated:

	time	name	size	type
count	49546	49546	49546	49546
unique	16378	32	4	4
top	12:25:12	classic_dlx	L	classic
freq	28	2453	18956	14860

Length of pizza data updated:

49546