Зозуля О.А. ИУ5-25М РК1

Вариант 6

Задача №6.

Для набора данных проведите устранение пропусков для одного (произвольного) числового признака с использованием метода заполнения средним значением.

Датасет: Бар

```
import matplotlib
from mpl_toolkits import mplot3d
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
import os
```

#Загрузка данных из файла

```
path=os.environ["userprofile"]+"\\"+".atom"+"\\"+"Transactions.csv"
print(path)
```

```
data = pd.read_csv(path)
print(data)
```

#Обнаружение пропущенных значений для массивовидных объектов.

```
data.isnull().sum()
```

```
Date_and_time_of_unloading        0
Product_code                      0
Amount                            0
Sale_amount                      51
Discount_amount               10202
Profit                           14
Percentage_markup              1939
Discount_percentage           10202
dtype: int64
```

#вычисляет и отображает сводную статистику для фрейма данных Python.

```
display(data[["Percentage_markup"]].describe())
```

8 rows × 1 columns

	Percentage_markup
count	48145.000000
mean	109.184511
std	1182.538753
min	-100.000000
25%	59.850000
50%	84.210000
75%	107.790000
max	79900.000000

```
from sklearn.impute import SimpleImputer
```

выполняет как подгонку, так и преобразование.

```
data["Percentage_markup"] = SimpleImputer(strategy =
"mean").fit_transform(data[["Percentage_markup"]])
```

#Вывод результатов

```
display(data.shape)
display(data[["Percentage_markup"]].isnull().sum())
display(data[["Percentage_markup"]].describe())
```

```
In 13  1  display(data.shape)
       2  display(data[["Percentage_markup"]].isnull().sum(
       3  display(data[["Percentage_markup"]].describe())

    (50084, 8)

Percentage_markup    0
dtype: int64

8 rows  8 rows × 1 columns

       Percentage_markup
count          50084.000000
mean             109.184511
std             1159.421366
min             -100.000000
25%               60.920000
50%               85.530000
75%              109.184511
max            79900.000000
```

#Построение парных диаграмм

```
sns.pairplot(data, vars=['Percentage_markup', 'Profit'], diag_kind='kde')
```

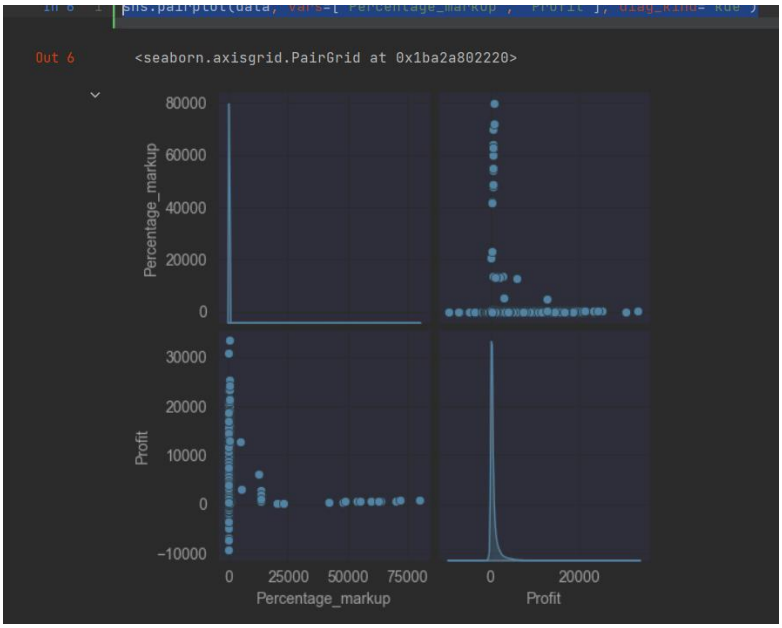

Рисунок 1. Парная диаграмма (Процентная отметка и Прибыль)

#Сохранение изменённых данных в файл

```
data.to_csv('file_with_filled_values.csv', index=False)
```

Вывод:

В этом примере мы загружаем данные из файла ‘Transactions.csv’, заменяем пропущенные значения в столбце 'Percentage_markup' средним значением, строим парные диаграммы для столбцов 'Percentage_markup' и ‘Profit’ с помощью метода pairplot из библиотеки seaborn и сохраняем измененные данные в файл 'file_with_filled_values.csv'.

Я использовала параметр diag_kind='kde' для построения графиков плотности распределения на диагонали парных диаграмм.

Задача №26.

Для набора данных для одного (произвольного) числового признака проведите обнаружение и замену (найденными верхними и нижними границами) выбросов на основе правила трех сигм.

Датасет: об убийствах

Вот пример кода на языке Python, который демонстрирует, как это можно сделать:

```
import matplotlib
from mpl_toolkits import mplot3d
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
import os
```

#Загрузка данных из файла

```
path=os.environ["userprofile"]+"\\"+".atom"+"\\"+"database.csv"
print(path)
```

```
data = pd.read_csv(path)
print(data)
```

#Вычисление среднего и стандартного отклонения

```
mean_value = data['Year']. mean ()
std_value = data['Year']. std()
```

#Вычисление верхней и нижней границы

```
K1 = 3
lower_boundary = data['Year'].mean() - (K1 * data['Year'].std())
upper_boundary = data['Year'].mean() + (K1 * data['Year'].std())
```

Флаги для удаления выбросов

```
outliers_temp = np.where(data['Year'] > upper_boundary, True, np.where(data['Year']<
lower_boundary, True, False))
```

Удаление данных на основе флага

```
data_trimmed = data['Year'].loc[~(outliers_temp),]
```

#Задаем имя столбцу

```
data_trimmed=data_trimmed
data_trimmed.name='Year'
print(data_trimmed)
```

#Строим график

```
sns.distplot(data_trimmed, hist=True, kde=False, rug=False )
```

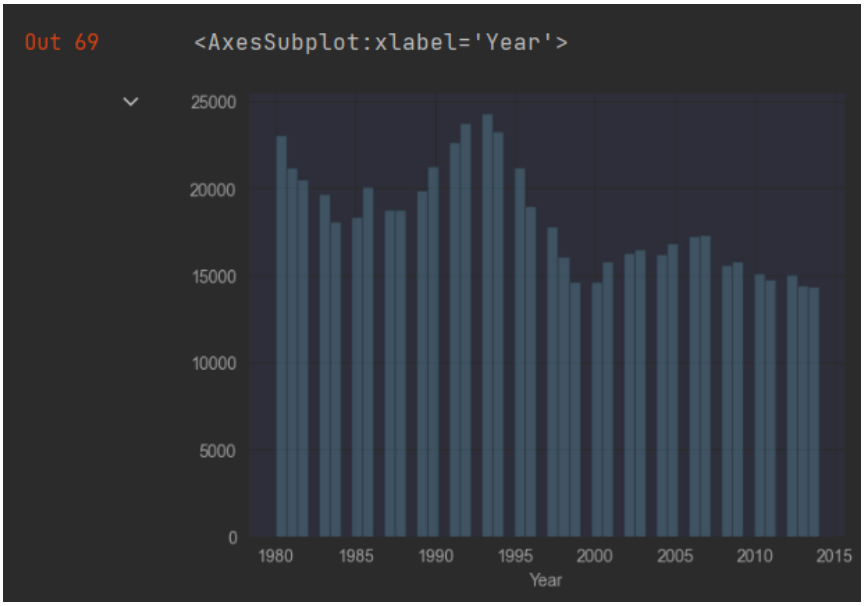

Рисунок 1

#Построение парных диаграмм

```
sns.pairplot(data, vars=['Year', 'Incident'][:500], diag_kind='kde')
```

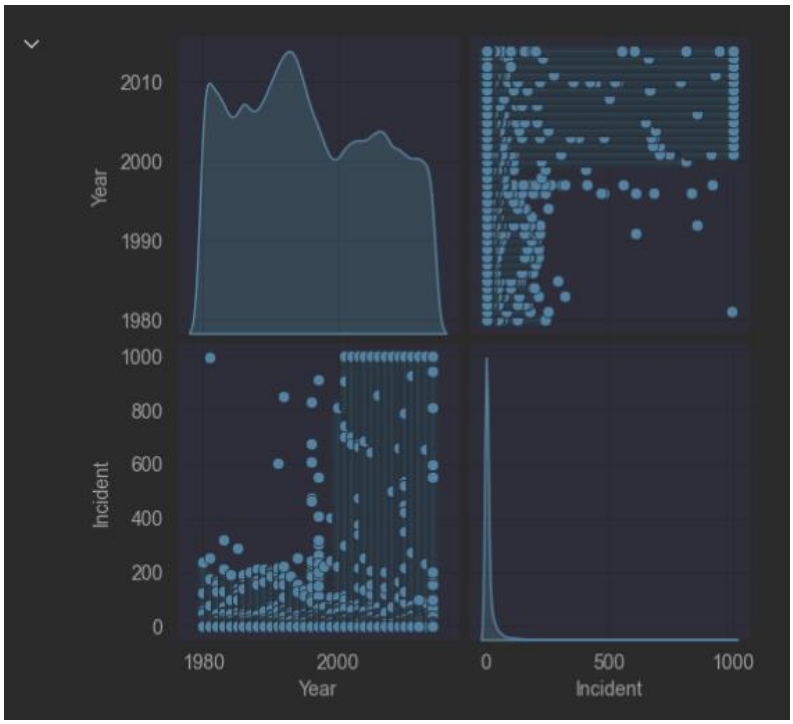

Рисунок 2. Парная диаграмма (Год и количество инцидентов)

Вывод:

В этом примере мы загружаем данные из файла ‘database.csv’, вычисляем среднее значение и стандартное отклонение для столбца ‘Year’, вычисляем верхнюю и нижнюю границы на основе правила трех сигм, заменяем выбросы на границы, строим парные диаграммы для столбцов ‘Year’ и ‘Incident’ с помощью метода pairplot из библиотеки seaborn и сохраняем измененные данные в файл 'file_with_replaced_outliers.csv'.

Я использовала параметр diag_kind='kde' для построения графиков плотности распределения на диагонали парных диаграмм.