МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

ИФТЭБ *ИНСТИТУТ ФИНАНСОВЫХ ТЕХНОЛОГИЙ И ЭКОНОМИЧЕСКОЙ БЕЗОПАСНОСТИ*

КАФЕДРА 75 «ФИНАНСОВЫЙ МОНИТОРИНГ»

Отчет по лабораторным работам №5
по курсу «Специальные технологии баз данных»

Выполнила
студентка группы С20-702
Нуритдинходжаева А.А.

Преподаватель: Манаенкова Т.А.

Москва 2023

Лабораторная работа №5
Вариант 1
Задание 1

Зарегистрируйтесь на сайте https://www.kaggle.com/datasets и загрузите с него набор

статистических данных, посвящённый опросам людей

https://www.kaggle.com/freecodecamp/2016-new-coder-survey-/version/1

1. На основе загруженного CSV-файла создайте Pandas DataFrame, подобрав правильные типы данных столбцов.
2. Создайте новый Pandas DataFrame, выбрав только переменные CityPopulation EmploymentStatus Gender HasDebt JobPref JobWherePref MaritalStatus Income SchoolDegree.
3. Удалите все наблюдения, содержащие либо значения поля пол (Gender), отличные от male или female, либо значения NA (нет ответа) в каких-либо из полей.
4. С помощью однофакторного дисперсионного анализа проверьте, как доход зависит от SchoolDegree. При этом проверьте:

   a. Нормальность распределения дохода (методы Жака (Харке)-Бера, Шапиро-Уилка, Андерсона-Дарлинга, Колмогорова-Смирнова):

      i. Если нормальность не выполняется, выполните лог-трансформацию дохода и проверьте заново.

      ii. Если нормальность не выполняется, ограничьте выборку 100 первыми записями.

   b. Отсутствие автокорреляции (тест Дарбина — Уотсона);

   c. Гомоскедастичность (Omnibus Test);

   d. Отсутствие мультиколлинеарности (Cond. Number).

5. Дайте интерпретацию результатам.
6. Проанализируйте уровни с помощью теста Тьюки.
7. С помощью многофакторного дисперсионного анализа проверьте, как доход зависит от остальных переменных, включите в проверку комбинацию Gender и MaritalStatus. При этом проверьте:

a. Нормальность распределения дохода (методы Жака (Харке)-Бера, Шапиро-Уилка, Андерсона-Дарлинга, Колмогорова-Смирнова):

i. Если нормальность не выполняется, выполните лог-трансформацию дохода и проверьте заново.

ii. Если нормальность не выполняется, попробуйте применить какие либо-методы, описанные в [1] (стр. 27) (Просто можно о них знать, какие существуют).

iii. Если нормальность не выполняется, ограничьте выборку 100 первыми записями.

b. Отсутствие автокорреляции (тест Дарбина — Уотсона);

c. Гомоскедастичность (Omnibus Test);

d. Отсутствие мультиколлинеарности (Cond. Number).

8. Дайте интерпретацию результатам.

```
import pandas as pd
import scipy.stats as sst
import numpy as np
import statsmodels.api as sm
from statsmodels.formula.api import ols
import matplotlib.pyplot as plt
import seaborn as sns

#1
df = pd.read_csv('2016-FCC-Data.csv', delimiter = ',', parse_dates =
['Part1EndTime', 'Part1StartTime', 'Part2EndTime', 'Part2StartTime'],
                dtype = {'CodeEventOther' : str, 'JobRoleInterestOther' : str})
#df = pd.read_csv('2016-FCC-New-Coders-Survey-Data.csv', delimiter = ',',
parse_dates = ['Part1EndTime', 'Part1StartTime', 'Part2EndTime',
'Part2StartTime'],
  #              dtype = {'CodeEventOther' : str, 'JobRoleInterestOther' : str})
pd.to_numeric(df['Income'], errors = 'coerce')

pd.set_option('display.max_columns', 2000)
pd.set_option('display.width', 20000)
print(df)

#2
```

```
df1 = df[['CityPopulation', 'EmploymentStatus', 'Gender', 'HasDebt', 'JobPref',
'JobWherePref', 'MaritalStatus', 'Income', 'SchoolDegree']]
print(df1)

#3
df1 = df1.dropna().sample(100, random_state=1234567)
df1 = df1[((df1['Gender'] == 'male') | (df1['Gender'] == 'female'))]
print(df1)

#4
  print(df1['SchoolDegree'].unique())

  g1 = df1[df1['SchoolDegree'] == "master's degree (non-professional)"]['Income']
  g2 = df1[df1['SchoolDegree'] == 'high school diploma or equivalent
(GED)']['Income']
  g3 = df1[df1['SchoolDegree'] == 'some college credit, no degree']['Income']
  g4 = df1[df1['SchoolDegree'] == "bachelor's degree"]['Income']
  g5 = df1[df1['SchoolDegree'] == 'professional degree (MBA, MD, JD,
etc.)']['Income']
  g6 = df1[df1['SchoolDegree'] == 'trade, technical, or vocational training'
]['Income']
  g7 = df1[df1['SchoolDegree'] == "associate's degree"]['Income']
  g8 = df1[df1['SchoolDegree'] == 'Ph.D.']['Income']
  g9 = df1[df1['SchoolDegree'] == 'some high school']['Income']
  g10 = df1[df1['SchoolDegree'] == 'no high school (secondary school)']['Income']

  print(sst.f_oneway(g1, g2, g3, g4, g5, g6, g7, g8, g9, g10), '\n')

#a
#Жака-Бера
print('Жака-Бера', '\n', '\n', sst.jarque_bera(df1['Income']), '\n') #отвергается

#Шапиро-Уилка
print('Шапиро-Уилка', '\n', '\n', sst.shapiro(df1['Income']), '\n') #отвергается

#Андерсона-Дарлинга
print('Андерсона-Дарлинга', '\n', '\n', sst.anderson(list(df1['Income'])),
'\n')#отвергаетс

#Колмогорова-Смирнова
print('Колмогорова-Смирнова', '\n', '\n', sst.kstest(list(df1['Income']), 'norm'), '\n')
#отвергается

# распределение не нормальное 0,05 > 2,44e-28
```

```
model = ols('Income ~ SchoolDegree', df1).fit()
print(model.summary())

#лог-трансформация
df1['Income'] = df1[['Income']].applymap(lambda x: np.log(x + 1))
model = ols('Income ~ SchoolDegree', df1).fit()
print(model.summary())

# зависимость лог-транформированного среднего дохода от образования
есть, 6,05e-06<0.05

# нормальность есть 0,54>0.05

#автокорреляция
df2 = pd.get_dummies(df1, columns=['SchoolDegree'], drop_first=True)
x = df2[['SchoolDegree_associate\'s degree','SchoolDegree_bachelor\'s
degree','SchoolDegree_high school diploma or equivalent
(GED)','SchoolDegree_master\'s degree (non-professional)','SchoolDegree_no high
school (secondary school)','SchoolDegree_professional degree (MBA, MD, JD,
etc.)','SchoolDegree_some college credit, no degree','SchoolDegree_some high
school','SchoolDegree_trade, technical, or vocational training']]
y = df2['Income']
x_ =sm.add_constant(x)
model = sm.OLS(y,x_).fit()
print(sm.stats.durbin_watson(model.resid))

#с
#гомоскедостичность
g1 = df1[df1['SchoolDegree'] == "master's degree (non-professional)"]['Income']
g2 = df1[df1['SchoolDegree'] == 'high school diploma or equivalent
(GED)']['Income']
g3 = df1[df1['SchoolDegree'] == 'some college credit, no degree']['Income']
g4 = df1[df1['SchoolDegree'] == "bachelor's degree"]['Income']
g5 = df1[df1['SchoolDegree'] == 'professional degree (MBA, MD, JD,
etc.)']['Income']
g6 = df1[df1['SchoolDegree'] == 'trade, technical, or vocational training'
]['Income']
g7 = df1[df1['SchoolDegree'] == "associate's degree"]['Income']
g8 = df1[df1['SchoolDegree'] == 'Ph.D.']['Income']
g9 = df1[df1['SchoolDegree'] == 'some high school']['Income']
g10 = df1[df1['SchoolDegree'] == 'no high school (secondary school)']['Income']

print(sst.levene(g1, g2, g3, g4, g5, g6, g7, g8, g9, g10))

g1 = df1[df1['SchoolDegree'] == "master's degree (non-professional)"]
```

```
g2 = df1[df1['SchoolDegree'] == 'high school diploma or equivalent (GED)']
g3 = df1[df1['SchoolDegree'] == 'some college credit, no degree']
g4 = df1[df1['SchoolDegree'] == "bachelor's degree"]
g5 = df1[df1['SchoolDegree'] == 'professional degree (MBA, MD, JD, etc.)']
g6 = df1[df1['SchoolDegree'] == 'trade, technical, or vocational training' ]
g7 = df1[df1['SchoolDegree'] == "associate's degree"]
g8 = df1[df1['SchoolDegree'] == 'Ph.D.']
g9 = df1[df1['SchoolDegree'] == 'some high school']
g10 = df1[df1['SchoolDegree'] == 'no high school (secondary school)']
plt.ylim(7, 15)
plt.boxplot((g1['Income'], g2['Income'], g3['Income'], g4['Income'], g5['Income'],
g6['Income'], g7['Income'], g8['Income'],
          g9['Income'], g10['Income']), labels = ["master's degree (non-
professional)", 'high school diploma or equivalent (GED)',
                                    'some college credit, no degree', "bachelor's
degree",
                                    'professional degree (MBA, MD, JD, etc.)',
                                    'trade, technical, or vocational training',
"associate's degree", 'Ph.D.',
                                    'some high school', 'no high school (secondary
school)'])
plt.show()

import seaborn as sns
import matplotlib.pyplot as plt

sns.distplot(g1['Income'], label="master's degree (non-professional)")
sns.distplot(g2['Income'], label='high school diploma or equivalent (GED)')
sns.distplot(g3['Income'], label='some college credit, no degree')
sns.distplot(g4['Income'], label="bachelor's degree")
sns.distplot(g5['Income'], label='professional degree (MBA, MD, JD, etc.)')
sns.distplot(g6['Income'], label='trade, technical, or vocational training')
sns.distplot(g7['Income'], label="associate's degree")
sns.distplot(g8['Income'], label='Ph.D.')
sns.distplot(g9['Income'], label='some high school')
sns.distplot(g10['Income'], label='no high school (secondary school)')

plt.xlim(5, 15)
#plt.legend()
plt.show()

#критерий Тьюки
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison
```

```
mc= MultiComparison(df1['Income'],df1['SchoolDegree'])
mc_results=mc.tukeyhsd()
print(mc_results)

#многофакторный анализ
model = ols('Income ~ C(Gender)*C(MaritalStatus)', df1).fit()
print(model.summary())

an_1 = sm.stats.anova_lm(model, typ = 1)
an_2 = sm.stats.anova_lm(model, typ = 2)
an_3 = sm.stats.anova_lm(model, typ = 3)
print(an_1)
print(an_2)
print(an_3)

model = ols('Income ~ C(Gender)+C(MaritalStatus)', df1).fit()
print(model.summary())

an_1 = sm.stats.anova_lm(model, typ = 1)
an_2 = sm.stats.anova_lm(model, typ = 2)
an_3 = sm.stats.anova_lm(model, typ = 3)
print(an_1)
print(an_2)
print(an_3)

mc= MultiComparison(df1['Income'],df1['Gender'])
mc_results=mc.tukeyhsd()
print(mc_results)

mc= MultiComparison(df1['Income'],df1['MaritalStatus'])
mc_results=mc.tukeyhsd()
print(mc_results)
```