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 №4, №5, №6
по курсу «Специальные технологии баз данных»

Выполнила
студентка группы С20-702
Нуритдинходжаева А.А.

Преподаватель: Манаенкова Т.А.

Москва 2023

Лабораторная работа №4
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 EmploymentField, EmploymentStatus, Gender, JobPref, JobWherePref, MaritalStatus, Income.

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. Исследуйте связи между парами переменных (используйте только наблюдения, где эти поля заполнены):

   a. Gender, JobPref;

   b. Gender, JobWherePref;

   c. JobWherePref, MaritalStatus;

   d. EmploymentField, JobWherePref;

   e. EmploymentStatus, JobWherePref.

Выполняя исследование, не используйте процедуру ANOVA. Для каждой пары постройте таблицу сопряжённости, таблицу ожидаемых значений. Обоснованно выберите один из методов: хи-квадрат Пирсона, хи-квадрат Пирсона с поправкой Йейтса, точный критерий Фишера (обычный или на основе приближения МонтеКарло), точный критерий Фримана-Холтона (обычный или на основе приближения Монте-Карло).

5. Для каждой пары интерпретируйте результаты.

6. Замените переменную Income на три уровня дохода: низкий, средний, высокий.

7. Исследуйте связи между парой переменных Gender, Income (в новом формате) аналогично заданию 4. Интерпретируйте результаты.

```
import pandas as pd
import scipy.stats as sst

df = pd.read_csv('2016-Data.csv', delimiter = ',', parse_dates = ['Part1EndTime', 'Part1StartTime', 'Part2EndTime', 'Part2StartTime'],
                 dtype = {'CodeEventOther' : str, 'JobRoleInterestOther' : str})
pd.to_numeric(df['Age'], errors = 'coerce')
pd.to_numeric(df['Income'], errors = 'coerce')
pd.set_option('display.max_columns', 2000) #для того, чтобы ввыводило все столбцы
pd.set_option('display.width', 20000) #максимальная ширина, чтобы ничего не переносилось
print(df)
"""
import pandas as pd
from sklearn.preprocessing import StandardScaler

numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()

scaler = StandardScaler()
df[numeric_columns] = scaler.fit_transform(df[numeric_columns])
print(df)"""

#2
df1 = df[['EmploymentField', 'EmploymentStatus', 'Gender', 'JobPref', 'JobWherePref', 'MaritalStatus', 'Income']]
print(df1)

#3
df2 = df1.dropna()
df2 = df2[((df2['Gender'] == 'male') | (df2['Gender'] == 'female'))].dropna()
#pd.set_option('display.max_rows', None)
print(df2)

missing_values = df2.isnull().any()

print("Пропущенные значения в каждом столбце:")
print(missing_values)
```

```
all_fields_filled = not missing_values.any()
print(f"Все ли поля заполнены в DataFrame? {all_fields_filled}")

#4
#a
#Gender, JobPref
print("\n", "a", "\n")
sopr = pd.crosstab(df2.Gender, df2.JobPref, margins = True)
print("ТАБЛИЦА СОПРЯЖЕННОСТИ:", '\n', '\n', sopr, '\n')

sorp_exp = pd.crosstab(df2.Gender, df2.JobPref, margins = False)
exp = sst.contingency.expected_freq(sorp_exp)
print("ОЖИДАЕМЫЕ ЗНАЧЕНИЯ:", '\n', '\n', exp, '\n')

#хи2 статистика Пирсона
print(sst.chi2_contingency(sopr, correction = False))

#b
#Gender, JobWherePref
print("\n", "b", "\n")
sopr = pd.crosstab(df2.Gender, df2.JobWherePref, margins = True)
print("ТАБЛИЦА СОПРЯЖЕННОСТИ:", '\n', '\n', sopr, '\n')

sorp_exp = pd.crosstab(df2.Gender, df2.JobWherePref, margins = False)
exp = sst.contingency.expected_freq(sorp_exp)
print("ОЖИДАЕМЫЕ ЗНАЧЕНИЯ:", '\n', '\n', exp, '\n')

#хи2 статистика
print(sst.chi2_contingency(sopr, correction = False))

#c
#JobWherePref, MaritalStatus
print("\n", "c", "\n")
sopr = pd.crosstab(df2.JobWherePref, df2.MaritalStatus, margins = True)
print("ТАБЛИЦА СОПРЯЖЕННОСТИ:", '\n', '\n', sopr, '\n')

sorp_exp = pd.crosstab(df2.JobWherePref, df2.MaritalStatus, margins = False)
exp = sst.contingency.expected_freq(sorp_exp)
print("ОЖИДАЕМЫЕ ЗНАЧЕНИЯ:", '\n', '\n', exp, '\n')

#d
#EmploymentField, JobWherePref
print("\n", "d", "\n")
sopr = pd.crosstab(df2.EmploymentField, df2.JobWherePref, margins = True)
```

```
print("ТАБЛИЦА СОПРЯЖЕННОСТИ:", '\n', '\n', sopr, '\n')

sorp_exp = pd.crosstab(df2.EmploymentField, df2.JobWherePref, margins =
False)
exp = sst.contingency.expected_freq(sorp_exp)
print("ОЖИДАЕМЫЕ ЗНАЧЕНИЯ:", '\n', '\n', exp, '\n')

#e
#EmploymentStatus, JobWherePref
print("\n", "e", "\n")
sopr = pd.crosstab(df2.EmploymentStatus, df2.JobWherePref, margins = True)
print("ТАБЛИЦА СОПРЯЖЕННОСТИ:", '\n', '\n', sopr, '\n')

sorp_exp = pd.crosstab(df2.EmploymentStatus, df2.JobWherePref, margins =
False)
exp = sst.contingency.expected_freq(sorp_exp)
print("ОЖИДАЕМЫЕ ЗНАЧЕНИЯ:", '\n', '\n', exp, '\n')

#хи2 статистика
print(sst.chi2_contingency(sopr, correction = False))

#6
income_bins = [0, 50000, 80000, float('inf')]
income_labels = ['Низкий', 'Средний', 'Высокий']
df2['Income_Level'] = pd.cut(df2['Income'], bins=income_bins,
labels=income_labels, right=False)
print(df2)

df2 = df1.dropna()
df2 = df2[((df2['Gender'] == 'male') | (df2['Gender'] == 'female'))].dropna()
#pd.set_option('display.max_rows', None)
print(df2)

#7
sop = pd.crosstab(df2.Gender, df2.Income_Level, margins = True)
print("ТАБЛИЦА СОПРЯЖЕННОСТИ:", '\n', '\n', sopr, '\n')

sop_exp = pd.crosstab(df2.Gender, df2.Income_Level, margins = False)
exp = sst.contingency.expected_freq(sop_exp)
print("ОЖИДАЕМЫЕ ЗНАЧЕНИЯ:", '\n', '\n', exp, '\n')

#хи2 статистика
print(sst.chi2_contingency(sop, correction = False))
```