#slot 01

# DATA UNDERSTANDING

import pandas as pd

df = pd.read\_csv('dulieuxettuyendaihoc.csv',header=0,delimiter=',',encoding='utf-8')

df

df.head(10)

df.tail(10)

df.columns

df.rename(columns={'NGAYTHI':'NT','DINHHUONGNGHENGHIEP':'NGHE'}, inplace=True)

df.columns

df.dtypes

df.shape

df.index

df.info()

# DATA FRAME MANIPULATION

df['M1']

df[['M1']]

newdf = df[['M1','M2','KV','KT']]

newdf

df[2:97]

df.loc[92]

df.loc[4:10]

df.loc[5:9,['M1','KT']]

df.iloc[5]

df.iloc[6:9]

df.iloc[:5]

df.iloc[95:]

df.iloc[2:5,1:3]

df.loc[5:20,['M1','M2','M3']]

df.iloc[5:21,1:7]

df.sort\_values(by='M1', ascending=True)

df.sort\_values(by=['M1','M2'], ascending=[True,False])

df[df['M1'] >= 5]

df[(df['M1'] > 6) & (df['M2'] > 5)]

df.loc[df['KT'] == 'C']

# DATA SUMMARY

df.aggregate({'M1':['min','max'],'M2':['mean','sum']})

df.groupby(['KV'])['KV'].agg(['count'])

df.groupby(['KV'])['M1'].agg(['count','min','mean','std','max','sum'])

df.groupby(['KV'])[['M1','M2']].agg(['min','mean'])

df.groupby(['KV','KT'])[['M1','M2']].agg(['min','max'])

df\_gr = df.groupby(['KV'],as\_index=False).count()

df\_gr

pd.pivot\_table(df,values=['M1','M2'], columns='KV', aggfunc=['min','mean','max'])

pd.pivot\_table(df, values=['M1','M2'], columns=['KV','KT'], aggfunc=['min','mean','max'])

#slot 02

import pandas as pd

import seaborn as sns

df = pd.read\_csv('dulieuxettuyendaihoc.csv',

header=0,delimiter=','

,encoding='utf-8')

df = df[['GT','DT','KV','KT','M1',

'M2','M3',

'NGAYTHI','DINHHUONGNGHENGHIEP']]

#Phần 1: Xử lý dữ liệu cơ bản

df.info()

df.rename(columns={'NGAYTHI':'NT',

'DINHHUONGNGHENGHIEP':'NGHE'},

inplace=True)

df.head(5)

df.dropna(how='all', inplace=True)

df.drop\_duplicates(inplace=True)

import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))

sns.heatmap(df.isna().transpose(),cmap='YlGnBu',

cbar\_kws={'label':'Dữ liệu thiếu'})

plt.savefig('missingdata.png', dpi=100)

plt.show()

df['DT'].fillna('KINH', inplace=True)

df['M1'].fillna(df['M1'].mean(), inplace=True)

df['M2'].fillna(df['M2'].median(), inplace=True)

df['M3'].fillna(0, inplace=True)

# Phần 2: Kỹ thuật Feature Engineering

df['TBM'] = (df['M1']+df['M2']+df['M3'])/3

df

df.loc[df['TBM'] < 5.0, 'XL'] = 'FAIL'

df.loc[(df['TBM'] >= 5.0) &

(df['TBM'] < 7.0), 'XL'] = 'FAIR'

df.loc[(df['TBM'] >= 7.0) &

(df['TBM'] < 9.0), 'XL'] = 'GOOD'

df.loc[df['TBM'] >= 9.0, 'XL'] = 'EXCEL'

df

dict\_map = {

'A1':'G1', 'C':'G3', 'D1': 'G3', 'A': 'G1', 'B': 'G2'

}

df['NHOM'] = df['KT'].map(dict\_map)

df

def fplus(x,y):

if (x == 'G1' or x == 'G2') and (y >= 4.0):

return 1.0

else:

return 0.0

df['PLUS'] = list(map(fplus,df['NHOM'],df['TBM']))

df

# Sinh viên tự làm tạo cột MaxM = giá trị lớn nhất của (M1,M2,M3)

# Phần 3: Trực quan hóa dữ liệu

sns.countplot(x='GT', data=df)

plt.show()

'''

Sinh viên tự làm tương tự cho các cột DT

--> Dựa trên biểu đồ DT cho biết tại sao

ta không phân tích theo phân nhóm DT

Sinh viên tự làm tương tự cho các cột NHOM

--> Dựa trên biểu đồ NHOM

hãy cho biết nhóm nào có lượng thí sinh đông nhất

và thấp nhất

'''

sns.countplot(x='KT', hue='GT',data=df)

plt.show()

'''

Sinh viên tự làm tương tự cho các nhóm

biến định tính: (KV,KT)

-->Hãy cho biết khối A có sinh viên khu vực nào đăng ký nhiều nhất

'''

sns.barplot(x='GT', y='M1', data=df, errorbar=None)

plt.show()

'''

Hãy so sánh trung bình điểm M2 theo NHOM

-> Cho biết nhóm thi nào có điểm thi M2 cao nhất

'''

sns.barplot(x='GT', y='M1', hue='KT', data=df, errorbar=None)

plt.show()

sns.barplot(x='GT', y='M1', hue='KT', data=df)

plt.show()

sns.barplot(x='GT', y='M1', hue='KT', data=df,

errorbar=None, estimator=max)

plt.show()

import numpy as np

sns.barplot(x='KV', y='M1', hue='KT',

data=df, errorbar=None, estimator=np.max)

plt.show()

gb = df.groupby(['KT'])['KT'].agg(['count'])

labels = gb.index # labels = ['A','A1','B','C','D1']

data = list(gb['count']) # data= [49,6,9,14,22]

colors = sns.color\_palette('pastel') # Tạo bảng màu

plt.pie(data, labels=labels, colors=colors,

autopct='%1.1f%%', shadow=True)

plt.show()

'''

Sinh viên tự viết thống kê tỉ lệ phần trăm cho KV

'''

gb = df.groupby(['KV'])['PLUS'].agg(['sum'])

labels = gb.index

data = list(gb['sum'])

colors = sns.color\_palette('pastel') # Tạo bảng màu

plt.pie(data, labels=labels, colors=colors,

autopct='%1.1f%%', shadow=True)

plt.show()

sns.lineplot(x='NT', y='M1', data=df)

plt.show()

sns.lineplot(x='NT', y='M1', data=df, estimator=np.max)

plt.show()

sns.lineplot(x='NT', y='M1', hue='GT',

data=df, estimator=np.max)

plt.show()

# Phần 4: Phân tích mô tả định lượng - THỐNG KÊ MÔ TẢ

print(f'{df["M1"].min()}\n{df["M1"].max()}\n')

print(f'{df["M1"].mean()}\n{df["M1"].median()}\n')

print(f'{df["KT"].mode()}\n')

print(f'{df["M1"].quantile(0.25)}\n')

# Hãy tính phân vị thứ 9 trong thập phân vị (gọi là p9) cho M1 và M2. Gợi ý: phân vị thứ 9 có tỉ lệ là 0.9

df['M1'].max() - df['M1'].min()

df[['M1','M2']].max() - df[['M1','M2']].min()

# Bài tập

# Hãy tính khoảng range cho M1 và M2

df[['M1','M2']].max() - df[['M1','M2']].min()

df['M1'].quantile(0.75) - df['M1'].quantile(0.25)

# Hãy tính độ lớn khoảng IQR cho M1 và M2

# Bài tập

# Hãy tính độ lớn khoảng dao động của M2 (gợi ý: |range|)

# Hãy tính độ lớn của IQR của M2 là bao nhiêu (gợi ý: |IQR|)

# Hãy tính khoảng giá trị IQR của M2 (gợi ý: [Q1;Q3])

np.mean(np.abs(df['M1'] - np.mean(df['M1'])))

# Hãy tính độ lệch trung bình cho M1 và M2

df['M1'].var()

df['M1'].std()

print(df[['M1','M2']].mean())

print(df[['M1','M2']].std())

df[['M1','M2']].std()/df[['M1','M2']].mean()

df['M1'].describe()

df[['M1','M2','M3']].describe()

cv = df[['M1','M2','M3']].std() / df[['M1','M2','M3']].mean()

list(cv)

# Bài tập: Sinh viên viết code để tìm ra giá trị cv lớn nhất và nhỏ nhất của M1, M2, M3

# Hãy sắp xếp tăng dần danh sách các giá trị cv

# Phân tích đơn biến định lượng (univariate analysis)

df['M1'].describe()

df[['M1','M2','M3']].describe()

df.groupby('GT')['M1'].describe()

# Câu hỏi: Đánh giá xem nhóm giới tính nào thi M1 tốt hơn

df.groupby('GT')['M2'].std()/df.groupby('GT')['M2'].mean()

# Câu hỏi đặt ra:

# Sinh viên có giới tính nam thì thi môn nào tốt hơn so với nữ

# Lưu ý: Chỉ dùng 2 dòng code để thể hiện kết quả

# Trực quan dữ liệu phân tích univariate analysis

df['M1'].hist()

plt.show()

df['M1'].hist(bins=14)

plt.show()

sns.displot(df, x='M1', kind='kde')

plt.show()

sns.displot(data = df[['M1','M2','M3']], kind='kde')

plt.show()

# Hãy cho biết phân phối của biến số nào gần với phân phối chuẩn hơn

sns.displot(df, x='M1', hue='GT', kind='kde')

plt.show()

# Câu hỏi: Nhóm giới tính nào có phân phối điểm gần phân phối chuẩn

df['M1'].skew()

df[['M1','M2','M3']].skew()

df[['M1']].kurtosis()

# Câu hỏi: biến M1 có độ nhọn như thế nào

df[['M1','M2','M3']].kurtosis()

# Hãy cho biết dạng phân phối nào là thin, fat và fit

sns.displot(data = df[['M1','M2']], kind='kde')

plt.show()

'''

Câu hỏi: Nhìn biểu đồ hãy cho biết ý nghĩa của kutorsis

của 'M1','M2'

'''

from scipy import stats

stats.probplot(df['M1'],plot=sns.mpl.pyplot)

plt.show()

# Hãy cho biết phân phối M1 có gần như phân phối chuẩn hay không

sns.boxplot(data=df['M2'], orient="h")

plt.show()

# Với điểm M2

# Hãy tính khoảng giá trị nghi ngờ bất thường

# Hãy tính khoảng giá trị được cho là bất thường

# Hãy tính xem có bao nhiêu sinh viên có điểm thi là bất thường

sns.boxplot(data=df[['M1','M2','M3']], orient='h')

plt.show()

# Câu hỏi: Hãy cho biết điểm số môn nào không xảy ra bất thường

sns.boxplot(x='M2', y='KT',data=df, orient='h')

plt.show()

# Câu hỏi: Khối thi nào có lower bound trùng với tứ phân vị thứ 1 (tức là Q1)

sns.boxplot(x='GT',y='M2',data=df)

plt.show()

# 1. M2 trên giới tính nào có bất thường

# 2. Có dễ dàng dự báo M2 theo yếu tố giới tính (GT) hay không?

sns.boxplot(x='KT', y='M2',hue='GT',data=df)

plt.show()

# Câu hỏi: Xác định các biểu đồ có dữ liệu bất thường

# Phân tích đa biến định lượng (multivariate analysis)

df[['M1','M2']].cov()

# Bài tập: Xây dựng ma trận phương sai M1, M2, M3.

# So sánh sự tương quan (ảnh hưởng, tác động, liên hệ) giữa

# M1 so với M3

# M2 so với M3

df[['M1','M2']].corr()

sns.lmplot(data=df, x='M1', y='M2', fit\_reg=True)

plt.show()

# Câu hỏi: Vẽ biểu đồ khám phá tương quan tuyến tính giữa M2 và M3

df[['M1','M2','M3']].corr()

# Câu hỏi: Hãy viết code sắp xếp tăng dần mức độ tác động (tương quan tuyến tính) của M2, M3 đến M1

# Câu hỏi: Hãy viết code tìm cặp tương quan tuyến tính cao nhất

sns.heatmap(df[['M1','M2','M3']].corr(), vmax=1.0, square=False).xaxis.tick\_top()

sns.lmplot(data=df, x='M1', y='M2',hue='GT', fit\_reg=True)

plt.show()

# Câu hỏi: Tương quan tuyến tính (M1, M2) của nhóm giới tính nào cao hơn

sns.pairplot(df[['M1','M2','M3']],diag\_kind='kde', kind='reg')

plt.show()

#Slot 03

import pandas as pd

df = pd.read\_csv('dulieuxettuyendaihoc.csv',

header=0,delimiter=',',encoding='utf-8')

df.rename(columns={'NGAYTHI':'NT',

'DINHHUONGNGHENGHIEP':'NGHE'},

inplace=True)

df

dfKhoiC = df.loc[df['KT'] == 'C']

dfKhoiC['M1']

dfKhoiC['M1'].describe()

import scipy.stats as stats

# one sample t-test ( n < 30)

stats.ttest\_1samp(dfKhoiC['M1'], popmean=4.0)

# Bài tập: Sinh viên kiểm tra xem điểm trung bình của môn thi M2 của khối thi C có bằng 5.7 hay không

# Bài tập: Sinh viên về tự tìm hiểu cách thiết lập mức tin cậy hoặc sai lầm

# Bài tập: Hãy kiểm tra xem có phải điểm trung bình M3

# của thí sinh thi khối C có là 6.5 hay không

# với mức sai lầm là 10%

from statsmodels.stats.weightstats import ztest as ztest

# one sample z-test ( n >= 30)

ztest(df['M1'], value=8.0)

#Bài tập: Hãy kiểm tra xem trung bình M2 có bằng 6.7

# two sample t-test ( n < 30)

dfKhoiC = df.loc[df['KT'] == 'C']

stats.ttest\_ind(dfKhoiC['M1'], dfKhoiC['M2'], equal\_var=True)

# Bài tập:

# Suy diễn xem trung bình M1 có bằng trung bình M3 cho

# thí sinh thi khối C hay không với mức tin cậy 95%

# two sample Z-test ( n >= 30)

ztest(df['M1'], df['M2'], value=0)

# Bài tập: Kiểm tra trung bình M2 và M3 có bằng nhau không

# với mức tin cậy 95%

from scipy.stats.stats import pearsonr (dùng cho tương quan hay không)

pearsonr(df['M1'], df['M2'])

# Bài tập: Sinh viên làm tương tự cho M1 và M3 có tương quan hay không?

# Bài tập: Sinh viên làm tương tự cho M2 và M3 có tương quan hay không?

# Fisher Test : dùng cho 2 biến định tính mà ở đó giá trị của 2 biến dữ liệu là nhị phân

import scipy.stats as stats

crosdata = pd.crosstab(df['GT'], [df['NGHE']], rownames=['GT'], colnames=['NGHE'])

crosdata

odd\_ratio, p\_value = stats.fisher\_exact(crosdata)

print('odd ratio is : ' + str(odd\_ratio))

print('p\_value is : ' + str(p\_value))

from scipy.stats import chi2\_contingency

crosdata = pd.crosstab(df['KV'], [df['KT']], rownames=['KV'], colnames=['KT'])

crosdata

# Chsquare Test

import matplotlib.pyplot as plt

plt.pcolor(crosdata,cmap='RdBu')

plt.colorbar()

plt.show()

stat, p, dof, expected = chi2\_contingency(crosdata)

alpha = 0.05

print("p value is " + str(p))

if p <= alpha:

print('Dependent (reject H0)')

else:

print('Independent (H0 holds true)')

# one way anova test

# Bài tập: GT và KT có mối quan hệ / mối liên hệ hay không?

import statsmodels.api as sm

from statsmodels.formula.api import ols

model = ols('M1 ~ GT', data=df).fit()

aov\_table = sm.stats.anova\_lm(model, typ=1)

aov\_table

# Bài tập: Điểm M2 có phụ thuộc vào khu vực hay không

# Bài tập: Điểm M3 có phụ thuộc khối thi hay không

model = ols('M1 ~ GT + KV + GT:KV',data=df).fit()

result = sm.stats.anova\_lm(model, type=2)

print(result)

# Bài tập: Phân tích xem M2 có phụ thuộc trên nhóm KV theo từng loại KT hay không

import statsmodels.api as sm

X\_with\_constant = sm.add\_constant(df[["M1"]].values)

y = df[['M3']].values

result = sm.OLS(y, X\_with\_constant).fit()

print(result.summary())

# Bài tập: Giả sử thí sinh có điểm M1 = 6.7 thì hãy dự báo điểm M3 sẽ là bao nhiêu

# Bài tập: Phân tích sự ảnh hưởng của M2 đến điểm thi M3. Hãy cho biết M2 = 5.7 , hỏi M3 được dự báo là bao nhiêu

import statsmodels.api as sm

X\_with\_constant = sm.add\_constant(df[["M1","M2"]].values)

y = df[['M3']].values

result = sm.OLS(y, X\_with\_constant).fit()

print(result.summary())

# Bài tập: Hãy phân tích sự ảnh hưởng của điểm M1, M2 đến M3 trên những thí sinh thi khối A1

# Cho biết nếu M1 = 6.0 và M2 = 5.5

# thì dự đoán học sinh đi thi khối A1 sẽ được điểm M3 là nhiêu

X = df[["M1","M2"]].values

y = df[['M3']].values

result = sm.OLS(y, X).fit()

print(result.summary())