

# Аналіз Даних

## Завдання#1; ПМ-4

Виконав Крошин Олександр

```
In [1]: import numpy as np
import pandas as pd
import os
import warnings

import calculator as magic #own implementation
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats

sns.set(style='whitegrid',palette='colorblind')
pd.set_option('max_columns',200)
pd.set_option('max_rows',200)
```

```
In [2]: file = os.listdir('data/')[1]
data = pd.read_excel('data/'+file,parse_dates=[ 'Timestamp' ])
data = data[data["Крыс"]==4]
data.reset_index(drop=True,inplace=True)
```

## Data Preprocessing and Tests evaluation

```
In [3]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

cols = data.columns
test_1_indices = (7, 26)
test_2_indices = (27,87)

test_1_cols = cols[test_1_indices[0]:test_1_indices[1]+1]
test_2_cols = cols[test_2_indices[0]:test_2_indices[1]]
test_3_cols = cols[87:]
```

```
In [4]: for col in cols[test_1_indices[0]:test_2_indices[1]]:
        data[col] = le.fit_transform(data[col])
        data.head()
```

Out[4]:

	Timestamp	Курс	Стать (бінарна)	ТІМС	Англійська або українська	Назва предмету, який вважаєш проф- орієнтованим	Оцінка з цього предмету:	1. Коли починаю будь-яку роботу, як правило, сподіваюся на успіх	2. Під час діяльності виявляю активність	схи- ви- ніч
0	2020-01-28 12:38:45.716	4	Жінка	62	93	Лінійна алгебра	100	1	1	
1	2020-01-28 12:44:22.881	4	Жінка	60	81	Математичний аналіз	75	1	1	
2	2020-01-28 13:23:32.746	4	Жінка	81	87	Технології прикладного програмування	100	1	1	
3	2020-01-28 13:24:42.728	4	Жінка	78	94	ОШІ	100	0	0	
4	2020-01-28 13:38:27.653	4	Жінка	81	87	Програмування	90	1	1	

```
In [5]: print(f'Всього маємо {data.shape[0]} відповідей від 25 приматів 4го курсу')
Всього маємо 19 відповідей від 25 приматів 4го курсу
```

**Test#1 evaluation**

Обчислимо оцінки для тестів

```

In [6]: test_1_questions = np.arange(20)
test_1_positive_indices = np.array([1,2,3,6,8,10,11,12,14,16,18,19,20])-1

test_1_keys = {
    'positives' : test_1_positive_indices,
    'negatives' : np.array(list(set(test_1_questions)-set(test_1_positive_indices)))
}

def map_test_1_ans(x):
    if x>=14:
        return "Мотивація досягнення успіху"
    elif x>=8:
        return "Полюс не виражений"
    else:
        return 'Мотивація уникнення невдачі'

data['Test#1'] =data[test_1_cols].iloc[:,test_1_keys['positives']].sum(axis=1) + (data[test_1_cols].iloc[:,test_1_keys['negatives']]==0).sum(axis=1)
data['Test#1: інтерпретація'] = data['Test#1'].map(map_test_1_ans)

```

## Test#2 evaluation

```

In [7]: subj_1_cols = test_2_cols.take([i for i in range(len(test_2_cols)) if i%3==0])
subj_2_cols = test_2_cols.take([i for i in range(len(test_2_cols)) if i%3==1])
subj_3_cols = test_2_cols.take([i for i in range(len(test_2_cols)) if i%3==2])

test_2_questions = np.arange(20)
test_2_positive_indices = np.array([1,2,5,6,8,11,12,14,17,19])-1

test_2_keys = {
    'positives': test_2_positive_indices,
    'negatives': np.array(list(set(test_2_questions)-set(test_2_positive_indices)))
}

def map_test_2_ans(x):
    if x>=15:
        return "Високий рівень внутрішньої мотивації"
    elif x>=6:
        return "Середній рівень внутрішньої мотивації"
    else:
        return "Низький рівень внутрішньої мотивації"

def eval_test_2(column_name,subj_cols,column_name_shortened):
    data['Test#2: ' + column_name] = (data[subj_cols].iloc[:,test_2_keys['positives']]>1).sum(axis=1)+(data[subj_cols].iloc[:,test_2_keys['negatives']]<2).sum(axis=1)
    data['Test#2: ' + column_name_shortened + ' - інтерпретація'] = data['Test#2: ' + column_name].map(map_test_2_ans)

eval_test_2('Рівняння мат. фізики',subj_1_cols, 'Мат.Фіз.')
eval_test_2('Технології прикладного програмування', subj_2_cols,'ТПП')
eval_test_2('Аналіз даних', subj_3_cols,'АД')

```

## Test#3 evaluation

```
In [8]: scales_dict = {
    'Комунікативні мотиви': np.array([7,10,14,32])-1,
    'Мотиви уникнення': np.array([6,12,13,15,19])-1,
    'Мотиви престижу': np.array([8,9,29,30,34])-1,
    'Професійні мотиви': np.array([1,2,3,4,5,26])-1,
    'Мотиви творчої самореалізації': np.array([27,28])-1,
    'Навчально-пізнавальні мотиви': np.array([17,18,20,21,22,23,24])-1,
    'Соціальні мотиви': np.array([11,16,25,31,33])-1,
  }
  for key in scales_dict.keys():
    data[key] = data[test_3_cols].iloc[:,scales_dict[key]].mean(axis=1).round(4)

  data.head(3)
```

Out[8]:

	Timestamp	Курс	Стать (бінарна)	ТІМС	Англійська або українська	Назва предмету, який вважаєш проф- орієнтованим	Оцінка з цього предмету:	1. Коли починаю будь-яку роботу, як правило, сподіваюся на успіх	2. Під час діяльності виявляю активність	схи ви: ініц
0	2020-01-28 12:38:45.716	4	Жінка	62	93	Лінійна алгебра	100	1	1	
1	2020-01-28 12:44:22.881	4	Жінка	60	81	Математичний аналіз	75	1	1	
2	2020-01-28 13:23:32.746	4	Жінка	81	87	Технології прикладного програмування	100	1	1	

1. Скласти інтервальний статистичний ряд (таблицю частот) для кожного тесту. Кількість інтервалів групування для тесту 1 — три, для тесту 2— чотири, для наступних тестів -формула Стерджесса.

```

In [9]: def plot_cumulative_hist(ax,col,bins):
        #data[col].value_counts().plot(ax=ax,kind='hist',cumulative=True,bins=bins,title=col)
        with warnings.catch_warnings():
            warnings.filterwarnings("ignore")
            sns.distplot(1e.fit_transform(data[col]),ax=ax,bins=bins).set_title(col)
def sturges(n):
    return 1 + np.log2(n)

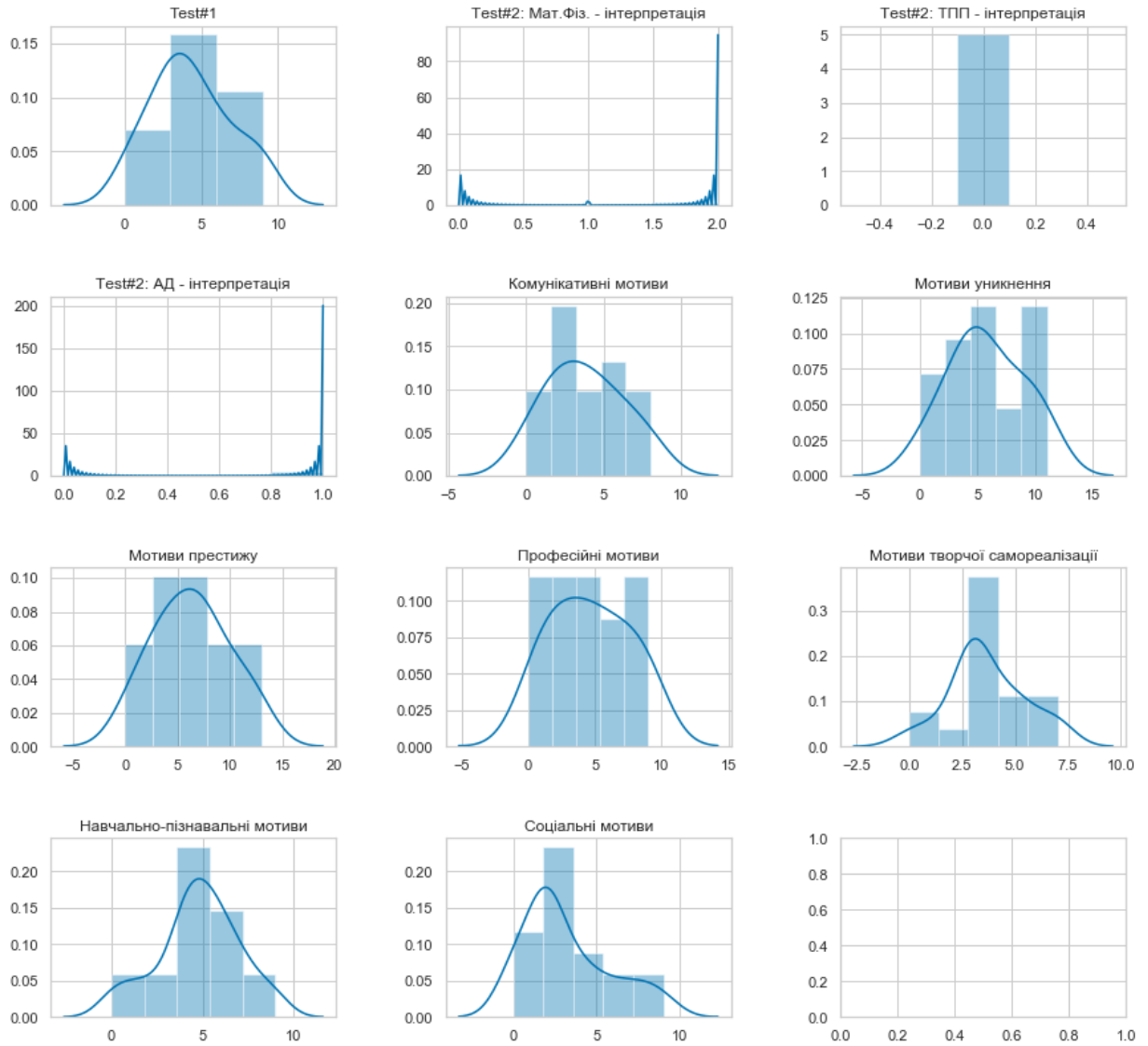
tests = ['Test#1','Test#2: Мат.Фіз. - інтерпретація','Test#2: ТПП - інтерпретація','Test
#2: АД - інтерпретація','Комунікативні мотиви','Мотиви уникнення','Мотиви престижу','Про
фесійні мотиви','Мотиви творчої самореалізації','Навчально-пізнавальні мотиви','Соціальн
і мотиви']
fig, axs = plt.subplots(4,3,figsize=(12,12))
fig.tight_layout(h_pad=5, w_pad=5)
fig.suptitle('Frequency tables',fontsize=16)

for i in range(len(tests)):
    i1,i2 = i//3,i%3
    if i==0:
        bins=3
    elif i==1:
        bins=4
    else:
        bins = int(sturges(data.shape[0]))

    plot_cumulative_hist(axs[i1,i2],tests[i],bins)
fig.subplots_adjust(top=0.9)

```

## Frequency tables



## Graphs:

1. Візуалізувати дані (для статті, оцінок, ознак з тестів 1,2,3). Побудувати полігон, гістограму, емпіричну функцію розподілу, кумулятивну криву, - ""відмітити на ній медіану та квартилі для оцінок, ознак з тестів 1, 2.

```

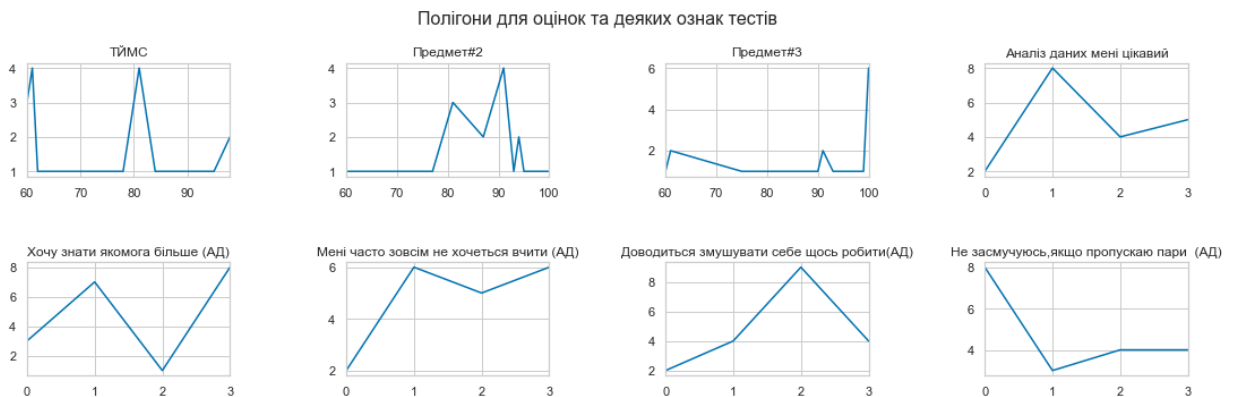
In [10]: data.rename(columns={'Англійська або українська':'Предмет#2','Оцінка з цього предмету':'Предмет#3'},inplace=True)
columns = ['ТІМС','Предмет#2','Предмет#3','#1. Вивчення цього предмета дає мені можливість дізнатися чимало важливого для себе, проявити свої здібності. [Аналіз даних]',
          '#2. Предмет, що вивчається, мені цікавий, і я хочу знати якомога більше. [Аналіз даних]',
          '#9. На заняттях з цього предмета у мене часто буває такий стан, коли «зовсім не хочеться вчитися». [Аналіз даних]',
          '#18. Цей предмет дається мені важко, доводиться змушувати себе виконувати і учбові завдання. [Аналіз даних]',
          '#19. Якщо через хворобу ( або з інших причин) я пропускаю пари з цього предмета, то мене це засмучує. [Аналіз даних]']
titles_dict = {
    '#1. Вивчення цього предмета дає мені можливість дізнатися чимало важливого для себе, проявити свої здібності. [Аналіз даних]':'Аналіз даних мені цікавий',
    '#2. Предмет, що вивчається, мені цікавий, і я хочу знати якомога більше. [Аналіз даних]':'Хочу знати якомога більше (АД)',
    '#9. На заняттях з цього предмета у мене часто буває такий стан, коли «зовсім не хочеться вчитися». [Аналіз даних]':'Мені часто зовсім не хочеться вчити (АД)',
    '#18. Цей предмет дається мені важко, доводиться змушувати себе виконувати учбові завдання. [Аналіз даних]':'Доводиться змушувати себе щось робити(АД)',
    '#19. Якщо через хворобу ( або з інших причин) я пропускаю пари з цього предмета, то мене це засмучує. [Аналіз даних]':'Не засмучуюсь,якщо пропускаю пари (АД)'
}

marks = data[['ТІМС','Предмет#2','Предмет#3']]

fig,axs = plt.subplots(2,4,figsize=(15,5))
fig.tight_layout(h_pad=6, w_pad=7)
fig.suptitle('Полігони для оцінок та деяких ознак тестів',fontsize=16)
fig.subplots_adjust(top=0.85)

for i,col in enumerate(columns):
    if col in titles_dict:
        title = titles_dict[col]
    else:
        title = col
    data[col].value_counts().sort_index().plot(ax=axs[i//4,i%4],title=title)

```

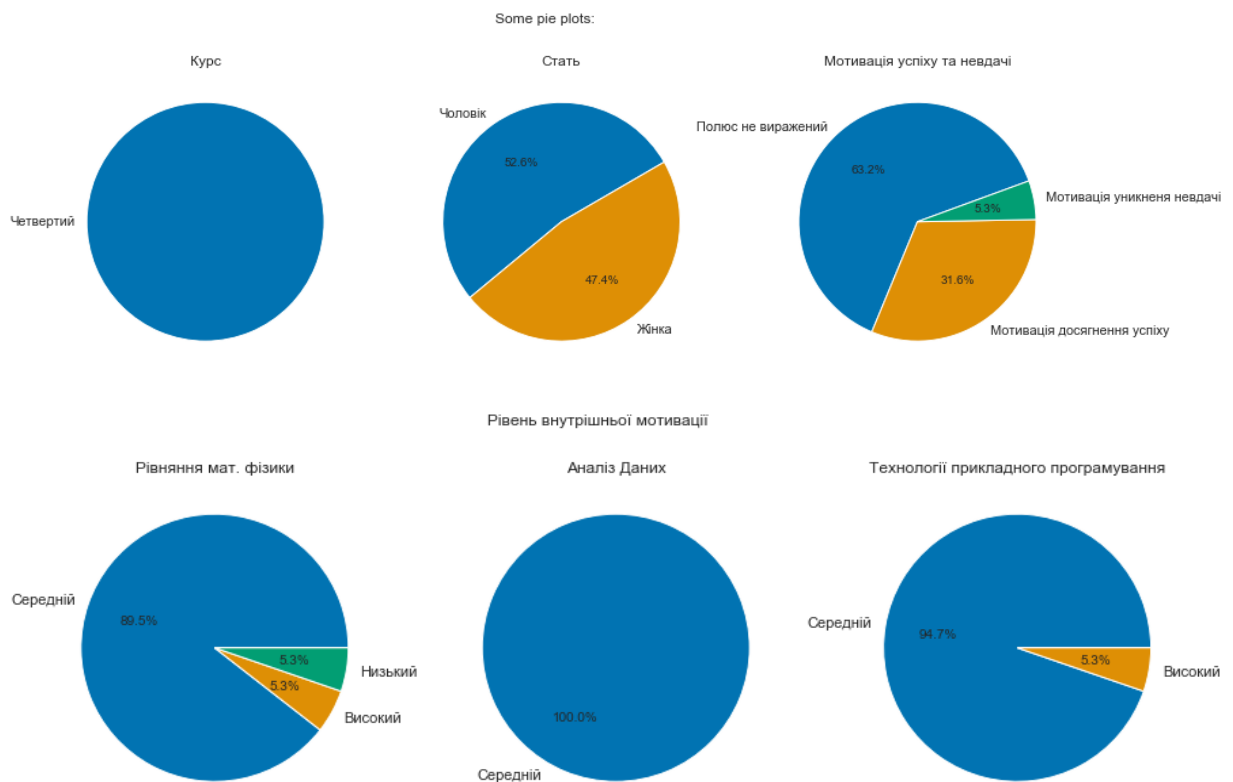


```
In [11]: def correct_labels(i):
    for j in range(3):
        ax[j].set_ylabel('')
        ax[j].set_title(suptitles[i][j])

suptitles = [['Курс', 'Стать', 'Мотивація успіху та невдачі'], ['Рівняння мат. фізики', 'Ан
аліз Даних', 'Технології прикладного програмування']]
test2_labels = ['Середній', 'Високий', 'Низький']

fig, ax = plt.subplots(1, 3, figsize=(15, 5))
plt.suptitle('Some pie plots:')
data['Курс'].value_counts().plot(ax=ax[0], kind='pie', labels=['Четвертий'])
data['Стать (бінарна)'].value_counts().plot(ax=ax[1], kind='pie', autopct='%1.1f%%', starta
ngle=30)
data['Test#1: інтерпретація'].value_counts().plot(ax=ax[2], kind='pie', autopct='%1.1f%%',
startangle=20)
correct_labels(0)

fig, ax = plt.subplots(1, 3, figsize=(15, 5))
plt.suptitle('Рівень внутрішньої мотивації')
data['Test#2: Мат.Фіз. - інтерпретація'].value_counts().plot(ax=ax[0], kind='pie', autopct
='%1.1f%%', startangle=0, labels=test2_labels)
data['Test#2: ТПП - інтерпретація'].value_counts().plot(ax=ax[1], kind='pie', autopct='%1.
1f%%', startangle=60, labels=test2_labels)
data['Test#2: АД - інтерпретація'].value_counts().plot(ax=ax[2], kind='pie', autopct='%1.1
f%%', startangle=0, labels=test2_labels)
correct_labels(1)
```



Більшість студентів мають середній рівень мотивації, як з конкретних предметів так і щодо мотивації успіху та невдач

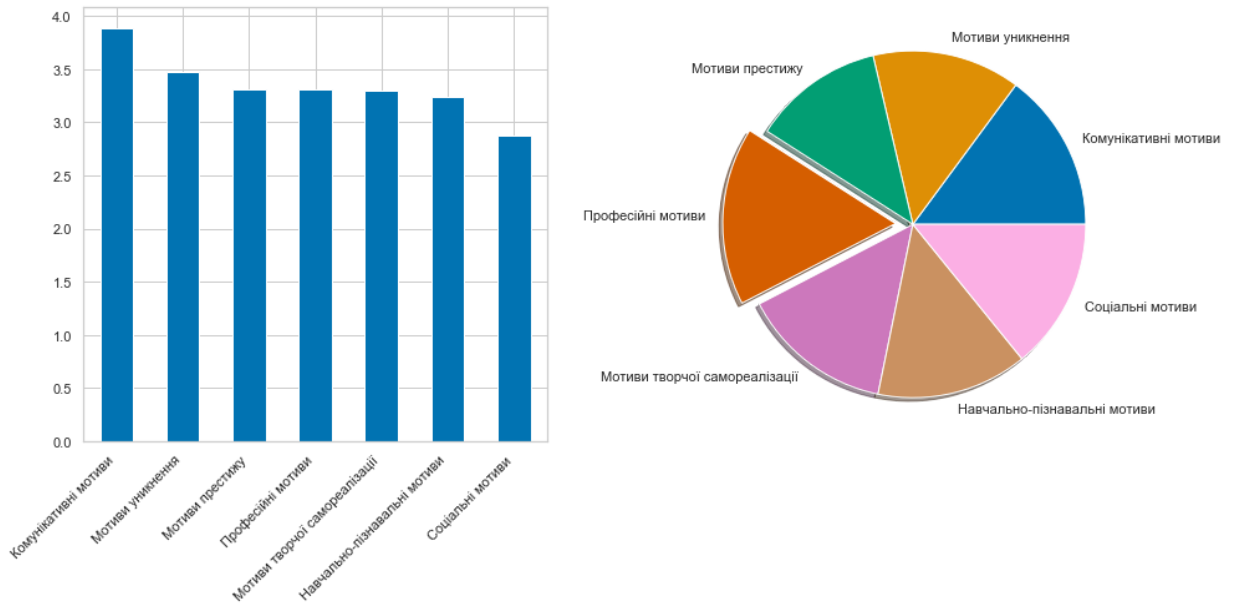


```
In [12]: fig,ax = plt.subplots(1,2,figsize=(14,7))

data.iloc[:,129:].mean().sort_values(ascending=False).plot(kind='bar',ax=ax[0])
ax[0].set_xticklabels(scales_dict.keys(),rotation=45,ha='right')

data.iloc[:,129:].mean().plot(kind='pie',ax=ax[1],explode=[0.,0.,0.,0.1,0.,0.,0.],shadow
=True)
ax[1].set_ylabel('')

fig.tight_layout()
```



Найголовнішим мотивом є професійний, найменш мотивуючими для студентів є мотиви уникнення та престижу

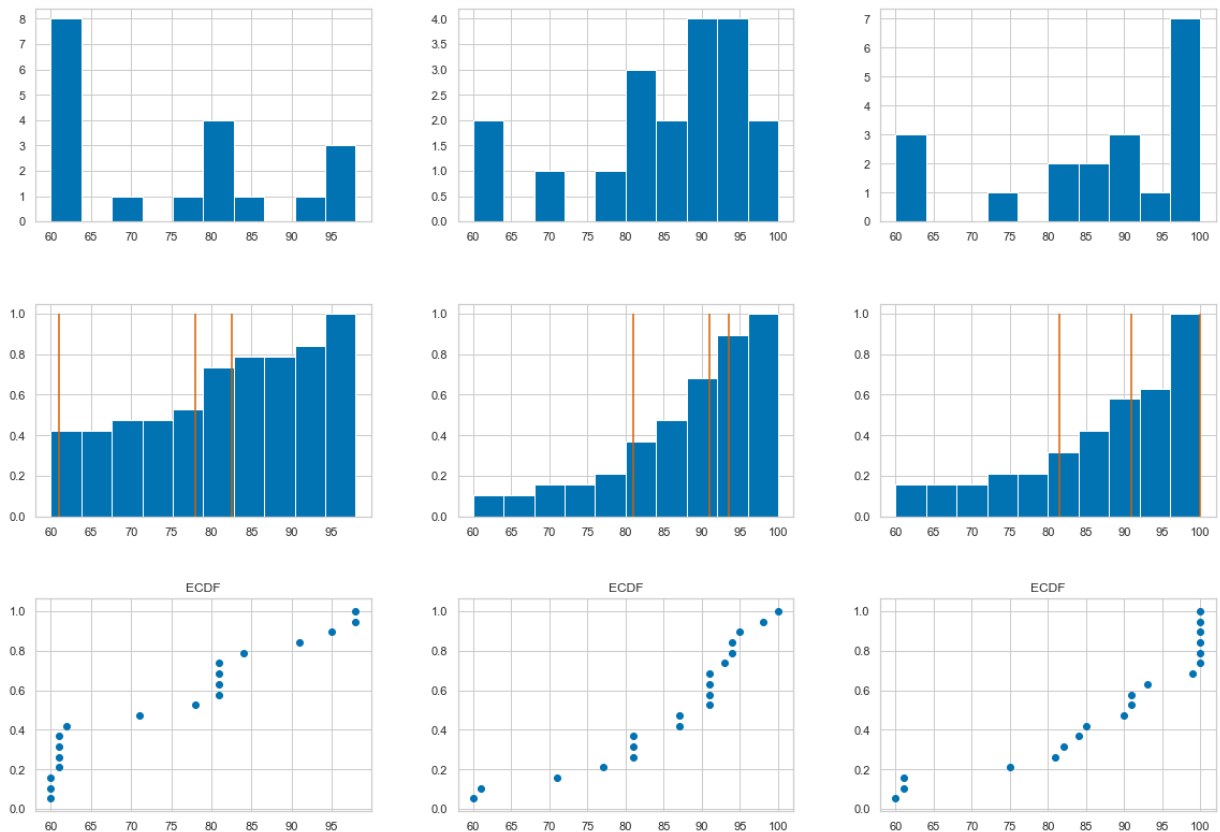
```
In [13]: fig, axs = plt.subplots(3,3,figsize = (16,12))
fig.suptitle("Marks' Histograms, cumulative histograms & EDF",fontsize=14)
for i,col in enumerate(marks.columns):
    data[col].hist(ax=axs[0,i])

    data[col].hist(cumulative=True,density=True,ax=axs[1,i])
    percentiles = np.percentile(data[col], [25, 50, 75])
    axs[1,i].vlines(percentiles, ymin=0, ymax=1, color='r')

    n =data[col].shape[0]
    axs[2,i].scatter(np.sort(data[col]), np.arange(1, n+1) / n)
    axs[2,i].set_title('ECDF')

fig.tight_layout(w_pad = 5,h_pad = 5)
fig.subplots_adjust(top = 0.9)
```

Marks' Histograms, cumulative histograms & EDF



1. (числові характеристики центральної тенденції та розкиду для оцінок та 2-х тестів — 2 б) Обчислити числові характеристики центральної тенденції та розкиду: вибіркове середнє, дисперсію, середньоквадратичне відхилення, моду, медіану, коефіцієнти асиметрії та ексцесу. Для обчислення застосувати табл.1 з прикладу 1.

```
In [14]: tests_cols = ['Test#1', 'Test#2: Аналіз даних']
quant_cols = list(marks.columns) + tests_cols

print('Характеристики центральної тенденції для оцінок та тестів. Зверніть увагу на дисперсію для оцінок з тестів')
pd.DataFrame([magic.mean(data[quant_cols]),
               magic.variance(data[quant_cols]),
               magic.stddev(data[quant_cols]),
               magic.mode(data[quant_cols]),
               magic.median(data[quant_cols]),
               magic.skewness(data[quant_cols]),
               magic.kurtosis(data[quant_cols]),
               magic.custom_percentile(data[quant_cols], 25),
               magic.custom_percentile(data[quant_cols], 50),
               magic.custom_percentile(data[quant_cols], 75)],
               index=['mean', 'var', 'std', 'mode', 'median', 'skewness', 'kurtosis', 'Q1', 'Q2', 'Q3']))
```

Характеристики центральної тенденції для оцінок та тестів. Зверніть увагу на дисперсію для оцінок з тестів

Out[14]:

	ТІМС	Предмет#2	Предмет#3	Test#1	Test#2: Аналіз даних
<b>mean</b>	75.000000	85.473684	87.000000	12.315789	10.000000
<b>var</b>	199.555556	132.040936	196.333333	10.005848	4.111111
<b>std</b>	14.126413	11.490907	14.011900	3.163202	2.027588
<b>mode</b>	61.000000	91.000000	100.000000	11.000000	9.000000
<b>median</b>	78.000000	91.000000	91.000000	12.000000	10.000000
<b>skewness</b>	0.654525	-2.223378	-1.902246	-0.155681	0.723372
<b>kurtosis</b>	-1.333915	0.111179	-0.471357	0.214273	0.587655
<b>Q1</b>	61.000000	81.000000	81.500000	11.000000	9.000000
<b>Q2</b>	78.000000	91.000000	91.000000	12.000000	10.000000
<b>Q3</b>	82.500000	93.500000	100.000000	14.000000	11.000000

Бачимо, що студенти мають середню мотивацію як до вивчення конкретних предметів, так і загалом. Оцінки з основного профільного предмету сильно (на 10+ балів) нижчі за оцінки з інших предметів. Зауважимо, що основним мотивом навчання студенти вважають професійні навички (Тест3)

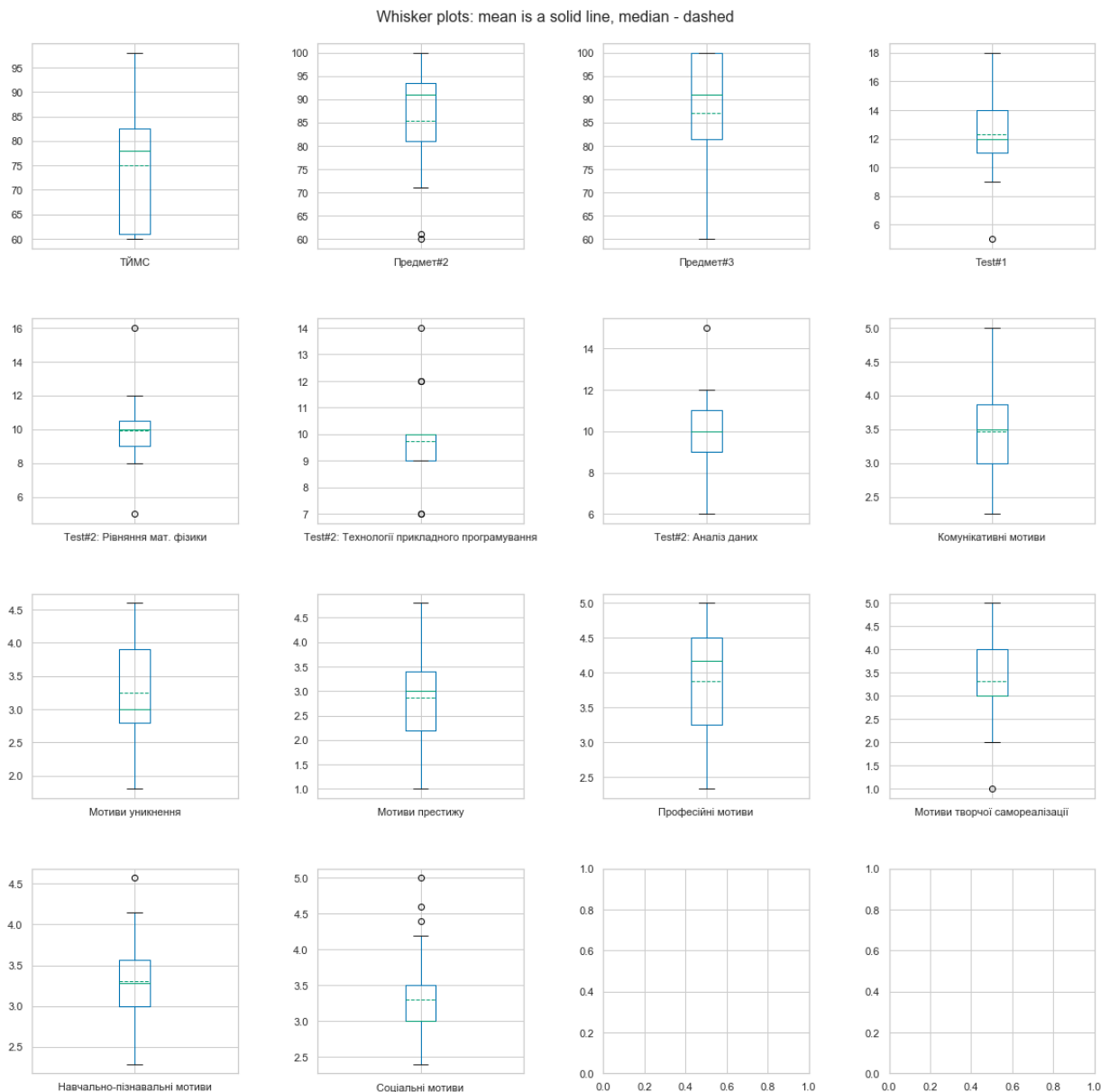
1. (коробки з вусами для оцінок та 2-х тестів — 2б) Побудувати коробки з вусами відносно середнього і відносно медіани.

```
In [15]: cols = ['ТІМС', 'Предмет#2', 'Предмет#3', 'Test#1', 'Test#2: Рівняння мат. фізики', 'Test#2: Технології прикладного програмування', 'Test#2: Аналіз даних'] + list(scales_dict.keys())

fig, axs = plt.subplots(4,4,figsize=(16,16))

fig.tight_layout(h_pad=5, w_pad=5)
fig.suptitle('Whisker plots: mean is a solid line, median - dashed', fontsize=16)
fig.subplots_adjust(top=0.95)

for i,col in enumerate(cols):
    i1,i2 = i//4,i%4
    data[col].plot(ax=axs[i1,i2],kind='box',meanline=True,showmeans=True)
```



Уточнюючи попередні висновки, коробки з вусами демонструють існування 1-2 високомотивованих студентів та 1-2 з вкрай низькою мотивацією. Медіана та середнє показані блакитною лінією та зеленим штрих-пунктиром відповідно

1. (статистичні похибки для оцінок та 2-х тестів -26) Визначити статистичну похибку у тесті дослідження середнього балу з кожної дисципліни і у тесті дослідження проценту жіночої статті (пропорції). Дослідити зв'язок з об'ємом вибірки. Визначити інтервальні оцінки.

```
In [16]: moe_columns = ["Sample size", 'Margin of error'] #moe: margin of error

moe = pd.DataFrame(index=quant_cols, columns=moe_columns)

error_rates = [5,5,5,0.5,0.5] # moe of 5 points for each mark, 5% for each test
for i,col in enumerate(quant_cols):
    margin_of_err = magic.margin_of_error_avg_score(data[col].shape[0],magic.stddev(data[col]))
    moe.loc[col]['Sample size'] = magic.sample_size_avg_score(magic.stddev(data[col]),margin_of_err)
    moe.loc[col]['Margin of error']= margin_of_err

moe
```

Out[16]:

	Sample size	Margin of error
ТЙМС	19	6.35201
Предмет#2	19	5.16694
Предмет#3	19	6.30052
Test#1	19	1.42235
Test#2: Аналіз даних	19	0.911715

Природньо, маємо велику похибку для такого низького об'єму вибірки

```
In [17]: moe_columns = ["Sample size", 'Margin of error'] #moe: margin of error

moe = pd.DataFrame(index=quant_cols, columns=moe_columns)

error_rates = [5,5,5,0.5,0.5] # moe of 5 points for each mark, 5% for each test
for i,col in enumerate(quant_cols):
    sample_size = magic.sample_size_avg_score(magic.stddev(data[col]),error_rates[i])

    moe.loc[col]['Margin of error']= magic.margin_of_error_avg_score(sample_size,magic.stddev(data[col]))
    moe.loc[col]['Sample size'] = sample_size

moe
```

Out[17]:

	Sample size	Margin of error
ТЙМС	30.6645	5
Предмет#2	20.2899	5
Предмет#3	30.1694	5
Test#1	153.754	0.5
Test#2: Аналіз даних	63.173	0.5

Бачимо, що нижча межа похибки вимагає більшого розміру вибірки

```
In [18]: gender_proportion = pd.DataFrame(index=['gender'], columns=moe_columns)
gender_proportion.loc['gender']['Sample size'] = magic.sample_size_proportion(0.05, 10/19)
gender_proportion.loc['gender']['Margin of error'] = magic.margin_of_error_proportion(383.096, 10/19)
gender_proportion
```

Out[18]:

	Sample size	Margin of error
gender	383.096	0.05

Більш низьке середнє квадратичне відхилення для тестів дає змогу мати меншу вибірку з тією ж похибкою. Більш натуральні дані з оцінок вказують на мінімальний розмір вибірки рівний 30. Для тесту на дослідження пропорції мінімальна похибка 0.05 вимагає мінімального розміру вибірки ~400

1. (метод моментів для оцінок -2б) Методом моментів знайти параметри розподілів. (нормальний, Стюдента), зробити графік, порівняти гістограму і графік розподілу.

```
In [19]: dist_params = pd.DataFrame(index=['ТІМС', "Предмет#2", "Предмет#3"], columns=['Mean', 'Variance', 'Degrees of freedom'])
for col in marks:
    dist_params.loc[col]['Mean'] = marks[col].mean()
    dist_params.loc[col]['Variance'] = marks[col].var()
    dist_params.loc[col]['Degrees of freedom'] = magic.ddof(marks[col].var()) #params of T-dist using variance
print('Параметри розподілів, отриманих методом моментів:')
dist_params
```

Параметри розподілів, отриманих методом моментів:

Out[19]:

	Mean	Variance	Degrees of freedom
ТІМС	75	199.556	2.01007
Предмет#2	85.4737	132.041	2.01526
Предмет#3	87	196.333	2.01024

```

In [20]: fig,axs = plt.subplots(2,3,figsize=(16,10))
plt.suptitle('Графіки відповідних розподілів',fontsize=15)

for i,col in enumerate(marks):
    mu,sigma,dof = marks[col].mean(),marks[col].std(),magic.ddof(marks[col].var())
    #data[col].hist(ax=axs[i//3,i%3],density=True)
    sns.distplot(marks[col],ax=axs[i//3,i%3],norm_hist=True,kde=False)
    lnspec = np.linspace(data[col].min()-30, data[col].max()+30, data[col].shape[0]*10)

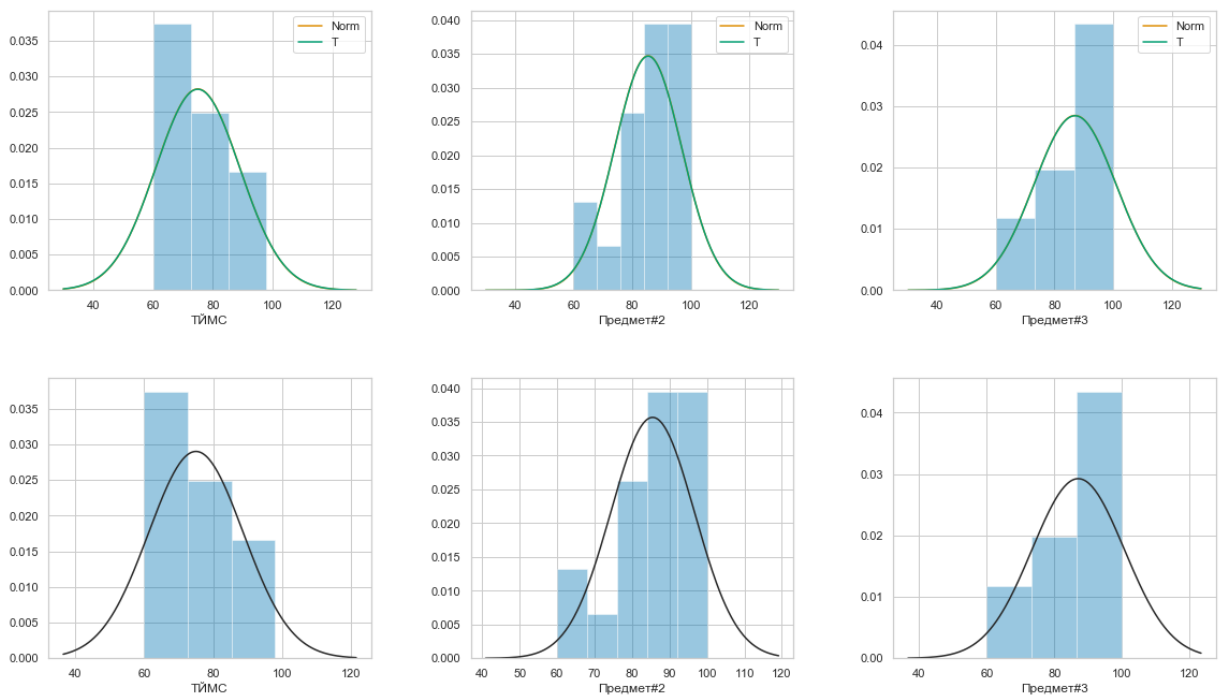
    pdf_n = stats.norm.pdf(lnspec,mu,sigma)
    pdf_t = stats.t.pdf(lnspec,len(lnspec)-dof,mu,sigma)
    axs[i//3,i%3].plot(lnspec, pdf_n, label="Norm")
    axs[i//3,i%3].plot(lnspec, pdf_t, label="T")
    axs[i//3,i%3].legend()

    sns.distplot(marks[col],fit=stats.norm,kde=False,norm_hist=True,ax=axs[1+i//3,i%3])

fig.tight_layout(h_pad=5, w_pad=5)
fig.subplots_adjust(top=0.9)

```

Графіки відповідних розподілів



Підібрані за методом моментів параметри нормального розподілу (Жовта крива на верхніх графіках) співпадають з обраним за MLE (Чорна крива на нижніх). Обчислюючи кількість ступенів свободи для t-розподілу за допомогою дисперсії, отримаємо t-розподіл дуже схожий на нормальний, якщо визначимо pdf від  $n - ddof$

Висновок: в ході дослідження були зібрані оцінки студентів ПМ-4 з трьох різних предметів та проведено три різні тести навчальної мотивації студентів. Отримані дані було візуалізовано, обчислено чисельні характеристики центральної тенденції та розкиду, побудовано коробки з вусами, визначено статистичну похибку та її зв'язок з об'ємом вибірки. Методом моментів були оцінені параметри нормального розподілу ( $\mu, \sigma$ ) та розподілу Стюдента ( $ddof$ ). Побудова графіків та порівняння результатами методу оцінки максимальної правдоподібності вказують на адекватність використання методу моментів в даній задачі.