Untitled

January 29, 2021

1 FINAL Assignment

Import necessary library

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(color_codes=True)
```

1.0.1 Read data

[4]:
```
data = pd.read_csv("clean_data.csv", index_col = 0)
data.drop_duplicates(subset=None, inplace=True)
data.head()
```

[4]:
```
                             tên  dd  mm    yy  toán  ngữ văn  khxh  khtn  \
sbd
2000001      Phạm Hoàng Hương Ái   4  11  2002   6.6     6.25  6.67 -1.00
2000002       Đặng Huỳnh Vĩnh An  13  12  2002   8.2     7.75  7.58 -1.00
2000003  Lâm Nguyễn Mộng Thùy An   6   4  2001   6.8     6.75  6.92 -1.00
2000004         Lê Tiêu Hoàng An  18  11  2002   7.8     6.25 -1.00  6.25
2000005              Lư Thuận An  14   1  2002   6.4     6.50 -1.00  6.17

         lịch sử  địa lí  gdcd  sinh học  vật lí  hóa học  tiếng anh
sbd
2000001     5.75    7.00  7.25      -1.0   -1.00    -1.00        5.2
2000002     7.00    7.25  8.50      -1.0   -1.00    -1.00        7.0
2000003     4.75    7.75  8.25      -1.0   -1.00    -1.00        6.0
2000004    -1.00   -1.00 -1.00       7.0    5.50     6.25        5.6
2000005    -1.00   -1.00 -1.00       5.5    6.75     6.25        8.2
```

So, you can see we have 16 columns; the first is just the ID of students, the next 4 are students's name, day , month and year of birth. The rest are the scores of every subjects.

[4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 74444 entries, 2000001 to 2074718
```

```
Data columns (total 15 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   tên        74444 non-null  object 
 1   dd         74444 non-null  int64  
 2   mm         74444 non-null  int64  
 3   yy         74444 non-null  int64  
 4   toán       74444 non-null  float64
 5   ngữ văn    74444 non-null  float64
 6   khxh       74444 non-null  float64
 7   khtn       74444 non-null  float64
 8   lịch sử    74444 non-null  float64
 9   địa lí     74444 non-null  float64
 10  gdcd       74444 non-null  float64
 11  sinh học   74444 non-null  float64
 12  vật lí     74444 non-null  float64
 13  hóa học    74444 non-null  float64
 14  tiếng anh  74444 non-null  float64
dtypes: float64(11), int64(3), object(1)
memory usage: 9.1+ MB
```

Check number of rows and non-null object in each columns

```
[5]: data.describe()
```

```
[5]:               dd            mm            yy          toán       ngữ văn  \
count  74444.000000  74444.000000  74444.000000  74444.000000  74444.000000   
mean      15.596529      6.830906   2001.730401      7.332647      6.556288   
std        8.794005      3.480114      1.232871      1.389498      1.480246   
min        0.000000      0.000000   1963.000000     -1.000000     -1.000000   
25%        8.000000      4.000000   2002.000000      6.600000      6.250000   
50%       16.000000      7.000000   2002.000000      7.600000      6.750000   
75%       23.000000     10.000000   2002.000000      8.200000      7.250000   
max       31.000000     12.000000   2003.000000      9.800000      9.250000   

               khxh          khtn       lịch sử        địa lí          gdcd  \
count  74444.000000  74444.000000  74444.000000  74444.000000  74444.000000   
mean       1.674875      3.099591      1.599514      2.128442      2.140172   
std        3.773566      3.686329      3.278103      3.904282      4.433405   
min       -1.000000     -1.000000     -1.000000     -1.000000     -1.000000   
25%       -1.000000     -1.000000     -1.000000     -1.000000     -1.000000   
50%       -1.000000      5.000000     -1.000000     -1.000000     -1.000000   
75%        6.330000      6.500000      4.750000      6.500000      7.750000   
max        9.580000      9.250000      9.750000      9.750000      9.750000   

           sinh học        vật lí       hóa học     tiếng anh  
count  74444.000000  74444.000000  74444.000000  74444.000000  
mean       2.807473      3.363972      3.408226      5.049084  
```

```
std        3.429285     3.887831     3.916852     2.757938
min       -1.000000    -1.000000    -1.000000    -1.000000
25%       -1.000000    -1.000000    -1.000000     4.000000
50%        4.250000     5.000000     5.000000     5.400000
75%        5.750000     7.000000     7.000000     7.000000
max        9.750000     9.750000     9.750000     9.800000
```

Getting understand more about data set

[6]:

```
a = data.corr()
a
```

[6]:

```
                 dd        mm        yy      toán   ngữ văn      khxh  \
dd         1.000000  0.014751  0.007692  0.000671 -0.002525 -0.000482
mm         0.014751  1.000000  0.006247  0.001299  0.010008 -0.000331
yy         0.007692  0.006247  1.000000  0.227143  0.372715  0.107668
toán       0.000671  0.001299  0.227143  1.000000  0.171862 -0.203280
ngữ văn   -0.002525  0.010008  0.372715  0.171862  1.000000  0.167407
khxh      -0.000482 -0.000331  0.107668 -0.203280  0.167407  1.000000
khtn       0.003667  0.002699  0.189316  0.486808  0.141675 -0.788314
lịch sử   -0.003692  0.002649 -0.060215 -0.338173  0.119328  0.859222
địa lí    -0.002154  0.004201  0.003510 -0.348444  0.117505  0.888177
gdcd      -0.000690  0.000111  0.106583 -0.216069  0.159683  0.988797
sinh học   0.003416 -0.003213  0.137134  0.475638  0.030630 -0.787020
vật lí     0.003819  0.002652  0.138205  0.490812  0.033660 -0.795661
hóa học    0.003636 -0.000544  0.110571  0.496083 -0.029338 -0.797774
tiếng anh -0.003800 -0.003994  0.317795  0.475481  0.350649  0.071560

               khtn   lịch sử    địa lí      gdcd  sinh học    vật lí  \
dd         0.003667 -0.003692 -0.002154 -0.000690  0.003416  0.003819
mm         0.002699  0.002649  0.004201  0.000111 -0.003213  0.002652
yy         0.189316 -0.060215  0.003510  0.106583  0.137134  0.138205
toán       0.486808 -0.338173 -0.348444 -0.216069  0.475638  0.490812
ngữ văn    0.141675  0.119328  0.117505  0.159683  0.030630  0.033660
khxh      -0.788314  0.859222  0.888177  0.988797 -0.787020 -0.795661
khtn       1.000000 -0.881898 -0.891118 -0.787706  0.953841  0.959826
lịch sử   -0.881898  1.000000  0.945938  0.836699 -0.880450 -0.890116
địa lí    -0.891118  0.945938  1.000000  0.877299 -0.889655 -0.899422
gdcd      -0.787706  0.836699  0.877299  1.000000 -0.786413 -0.795047
sinh học   0.953841 -0.880450 -0.889655 -0.786413  1.000000  0.891884
vật lí     0.959826 -0.890116 -0.899422 -0.795047  0.891884  1.000000
hóa học    0.949798 -0.892481 -0.901812 -0.797159  0.943269  0.920703
tiếng anh  0.325259 -0.187650 -0.172473  0.060921  0.276653  0.317403

            hóa học  tiếng anh
dd         0.003636  -0.003800
mm        -0.000544  -0.003994
```

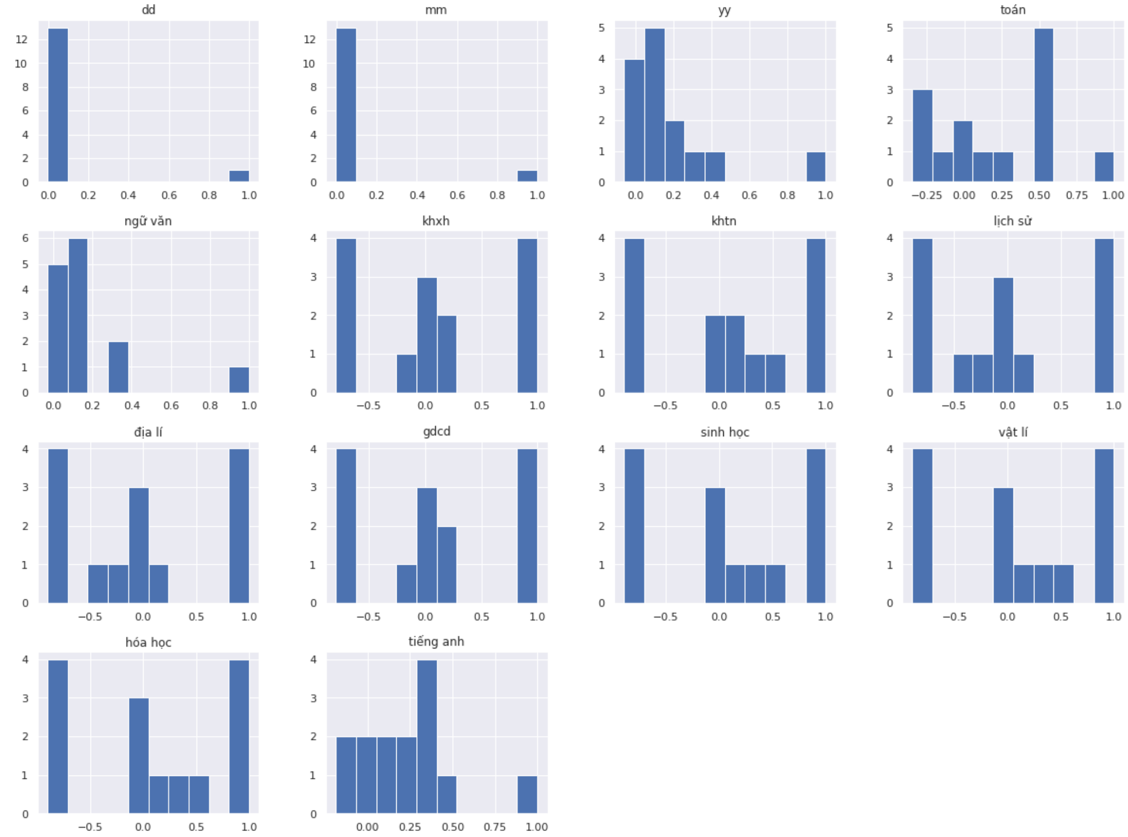
```
yy          0.110571   0.317795
toán        0.496083   0.475481
ngữ văn    -0.029338   0.350649
khxh       -0.797774   0.071560
khtn        0.949798   0.325259
lịch sử    -0.892481  -0.187650
địa lí     -0.901812  -0.172473
gdcd       -0.797159   0.060921
sinh học    0.943269   0.276653
vật lí      0.920703   0.317403
hóa học     1.000000   0.249967
tiếng anh   0.249967   1.000000
```

```
[40]: a["dd"].sort_values(ascending = False)
```

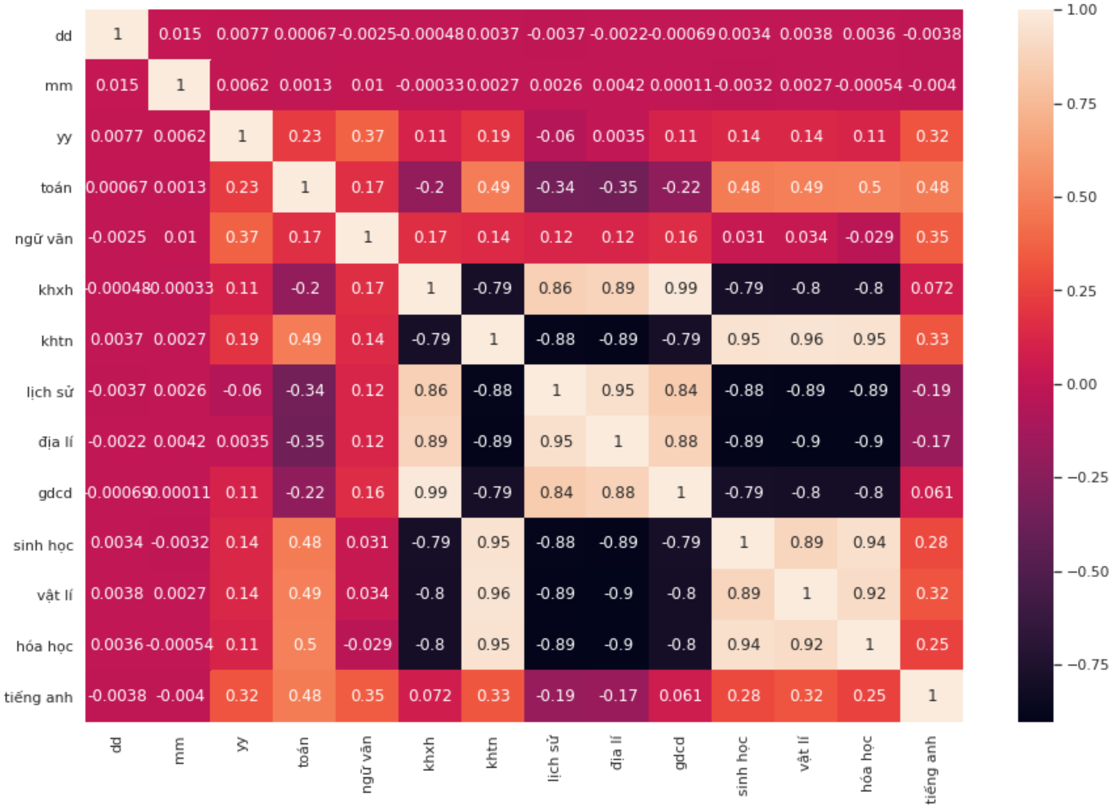
```
[40]: dd           1.000000
      mm           0.014751
      yy           0.007692
      vật lí       0.003819
      khtn         0.003667
      hóa học      0.003636
      sinh học     0.003416
      toán         0.000671
      khxh        -0.000482
      gdcd        -0.000690
      địa lí      -0.002154
      ngữ văn     -0.002525
      lịch sử     -0.003692
      tiếng anh   -0.003800
      Name: dd, dtype: float64
```

```
[7]: a.hist(figsize = (20,15))
```

```
[7]: array([[<AxesSubplot:title={'center':'dd'}>,
             <AxesSubplot:title={'center':'mm'}>,
             <AxesSubplot:title={'center':'yy'}>,
             <AxesSubplot:title={'center':'toán'}>],
            [<AxesSubplot:title={'center':'ngữ văn'}>,
             <AxesSubplot:title={'center':'khxh'}>,
             <AxesSubplot:title={'center':'khtn'}>,
             <AxesSubplot:title={'center':'lịch sử'}>],
            [<AxesSubplot:title={'center':'địa lí'}>,
             <AxesSubplot:title={'center':'gdcd'}>,
             <AxesSubplot:title={'center':'sinh học'}>,
             <AxesSubplot:title={'center':'vật lí'}>],
            [<AxesSubplot:title={'center':'hóa học'}>,
             <AxesSubplot:title={'center':'tiếng anh'}>, <AxesSubplot:>,
<AxesSubplot:>]], dtype=object)
```



[8]:
```
plt.figure(figsize = (15,10))
sns.heatmap(data.corr(),annot=True)
plt.show()
```

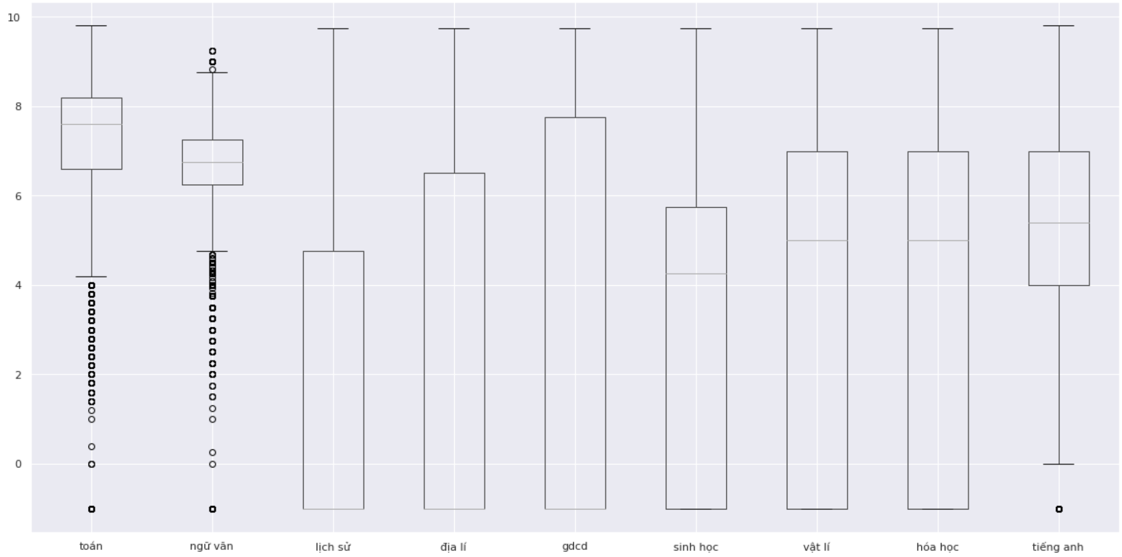


You can see clearly how important that attribute with dd

```
[9]: new_data = data[["toán","ngữ văn","lịch sử","địa lí","gdcd","sinh học","vật␣
     ↪lí","hóa học","tiếng anh"]]
```

```
[10]: plt.figure(figsize = (20, 10))
      new_data.boxplot()
```

```
[10]: <AxesSubplot:>
```



From the boxplot above alone, we can see that each subject is clearly

[44]:

```
new_data1 = data[["vật lí","toán","khtn"]]
new_data1.drop(new_data1.index[new_data1['vật lí'] == -1], inplace = True)
new_data1.drop(new_data1.index[new_data1['toán'] == -1], inplace = True)
new_data1.drop(new_data1.index[new_data1['khtn'] == -1], inplace = True)
```

```
/home/long/anaconda3/envs/data/lib/python3.7/site-
packages/pandas/core/frame.py:4174: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
/home/long/anaconda3/envs/data/lib/python3.7/site-
packages/pandas/core/frame.py:4174: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
/home/long/anaconda3/envs/data/lib/python3.7/site-
packages/pandas/core/frame.py:4174: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

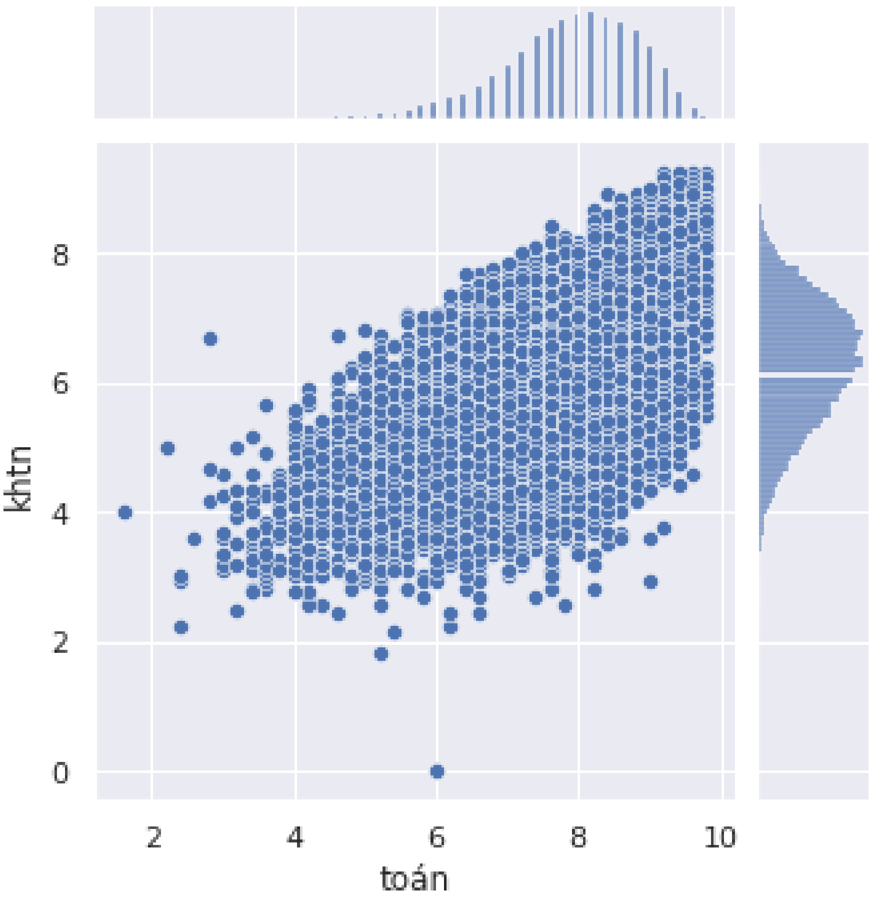
new_data1.plot(kind=“scatter”, x=“toán”, y=“khtn”)

In the graph above, , you can see there seems to be a split between portions of the data.The scores are mostly concentrated between 6 and 10, the rest are less than and above 4 are few. You can see some outlier on the left and below.In math, as you can see clearly a few students can not pass the example because have scores that under 2 which can not pass exams. Also, in the below maybe that students not take part in that combination.

[29]:
```
sns.jointplot(x="toán", y="khtn", data=new_data1, size=5)
```

```
/home/long/anaconda3/envs/data/lib/python3.7/site-
packages/seaborn/axisgrid.py:2073: UserWarning: The `size` parameter has been
renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

[29]:
```
<seaborn.axisgrid.JointGrid at 0x7fb58092c2b0>
```

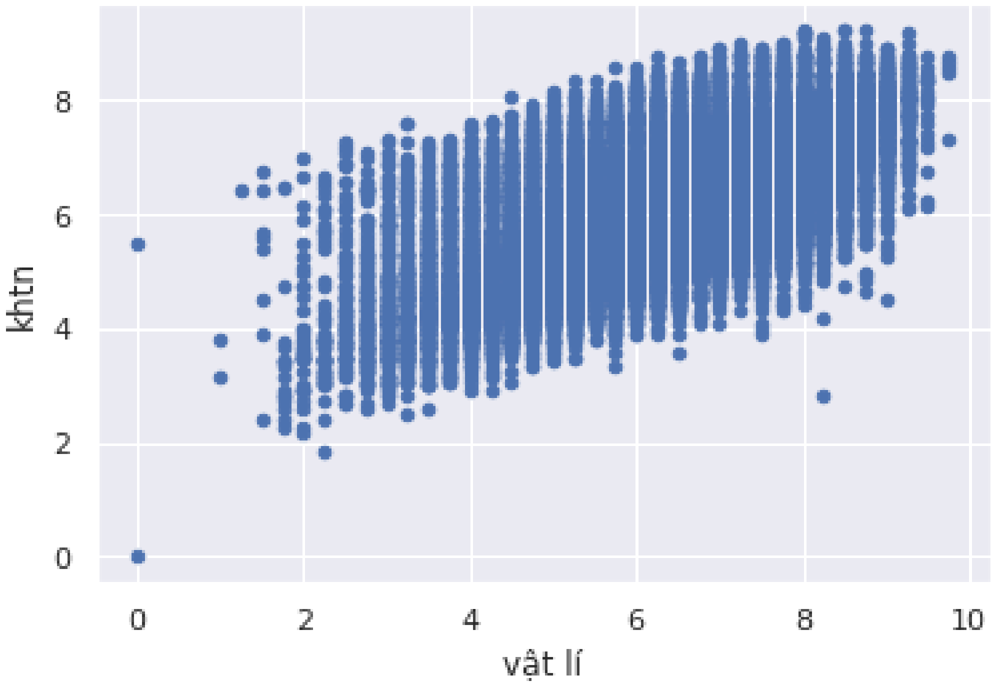


The additional detail provided by the histograms shows us that the students who have high score in Math also have high score in combination.

[30]:
```
new_data1.plot(kind="scatter", x="vật lí", y="khtn")
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
```

[30]:
```
<AxesSubplot:xlabel='vật lí', ylabel='khtn'>
```

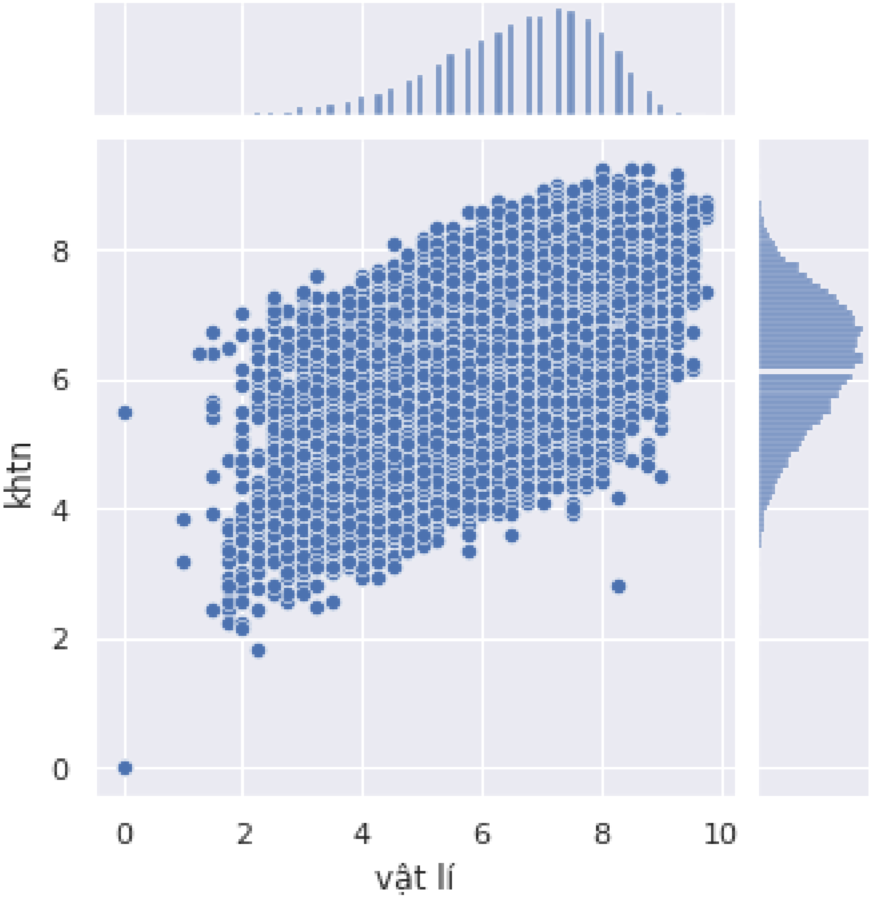


In that graph, Physical is one of 3 subjects in combination but it is not a compulsory subject so students take part in that subject and combination have scores more evenly than in Math. The concentration ratio is from 2 scores to more than 9 . We can easily see that do not have students have maximun scores in Math and Physical. In this graph has some special that also have student can not pass exam but have 2 points have 0 in Physical so it means that students when finish 1 in 3 combination and they know they can not pass exam so cancel it.

[31]:
```
sns.jointplot(x="vật lí", y="khtn", data=new_data1, size=5)
```

```
/home/long/anaconda3/envs/data/lib/python3.7/site-
packages/seaborn/axisgrid.py:2073: UserWarning: The `size` parameter has been
renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

[31]:
```
<seaborn.axisgrid.JointGrid at 0x7fb5807b5400>
```



```
[51]: with open("clean_data.csv", encoding="utf8") as file:
          data = file.read().split("\n")
```

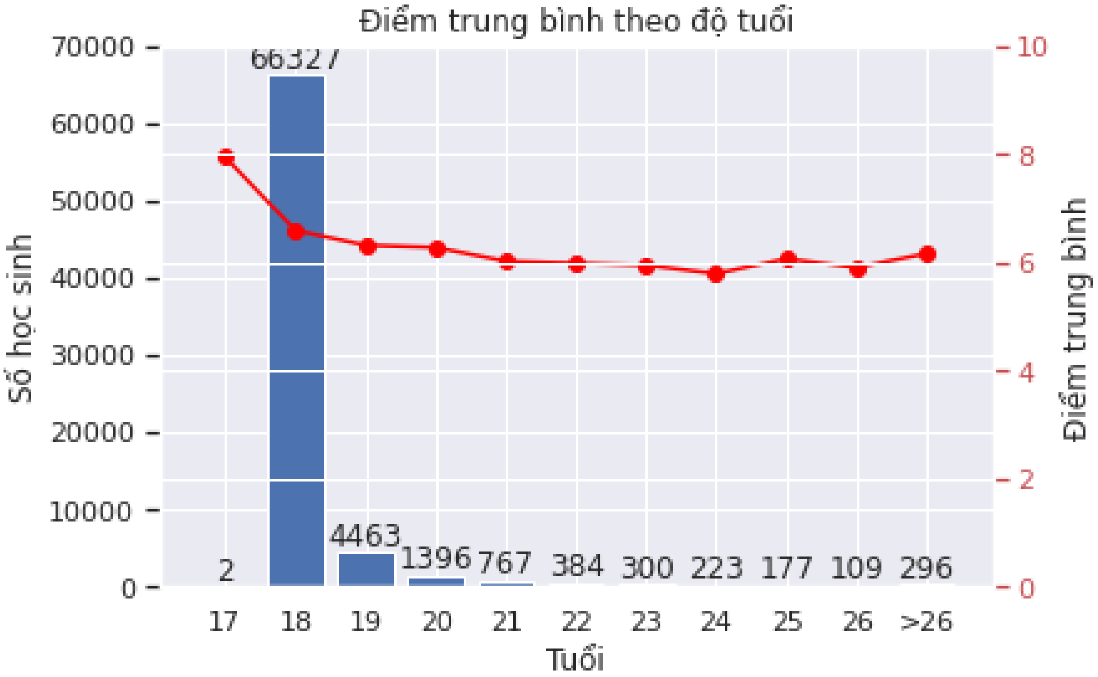
```
[52]: header = data[0]
      students = data[1:]
```

```
[53]: total_student = len(students)

      header = header.split(",")
      subjects = header[5:]

      for i in range(len(students)):
          students[i] = students[i].split(",")

      # remove last student (empty student)
      students.pop()
```

```
num_of_student_per_age_group = [0,0,0,0,0,0,0,0,0,0,0]
average_of_student_per_age_group = [0,0,0,0,0,0,0,0,0,0,0]

for s in students:
    age = 2020 - int(s[4])
    if age >= 27:
        age = 27
    num_of_student_per_age_group[age - 17] += 1

    sum_score = 0 # Tổng điểm
    count_score = 0 # Số môn thi
    for i in range(11):
        if s[i+5] != "-1":
            count_score += 1
            sum_score += float(s[i+5])

    average = sum_score/count_score
    average_of_student_per_age_group[age-17] += average

for i in range(len(average_of_student_per_age_group)):
        average_of_student_per_age_group[i] =␣
 ↪average_of_student_per_age_group[i]/num_of_student_per_age_group[i]

for i in range(len(average_of_student_per_age_group)):
        average_of_student_per_age_group[i] =␣
 ↪average_of_student_per_age_group[i] * 7000

print(num_of_student_per_age_group)
print(average_of_student_per_age_group)

# Draw barchart
# https://matplotlib.org/3.1.0/gallery/ticks_and_spines/custom_ticker1.
 ↪html#sphx-glr-gallery-ticks-and-spines-custom-ticker1-py


age_label = [17,18,19,20,21,22,23,24,25,26,">26"]
x = np.arange(11)
y = np.arange(11)

fig, axis = plt.subplots()
plt.bar(x, num_of_student_per_age_group)
plt.plot(x, average_of_student_per_age_group, color='red', marker='o')
# set limit
axis.set_ylim(0,70000)

# label for column x
plt.xticks(x, age_label)
```

```
axis.set_ylabel('Số học sinh')
axis.set_xlabel("Tuổi")

# right side ticks
ax2 = axis.twinx()
ax2.tick_params('y', colors='r')
ax2.set_ylabel("Điểm trung bình")
ax2.set_ylim(0,10)

rects = axis.patches
# Label for barchart
# https://stackoverflow.com/questions/28931224/
 ↪adding-value-labels-on-a-matplotlib-bar-chart
labels = [2, 66327, 4463, 1396, 767, 384, 300, 223, 177, 109, 296]
for rect, label in zip(rects, labels):
    height = rect.get_height()
    axis.text(rect.get_x() + rect.get_width() / 2, height + 2, label,
            ha='center', va='bottom')

plt.title('Điểm trung bình theo độ tuổi')

plt.show()
```

```
[2, 66327, 4463, 1396, 767, 384, 300, 223, 177, 109, 296]
[55674.99999999999, 46142.82397816897, 44245.86389098245, 43956.57103629422,
42182.226857887865, 41931.009114583336, 41655.41388888891, 40588.5754857997,
42488.064971751395, 41378.675840978576, 43154.051801801805]
```

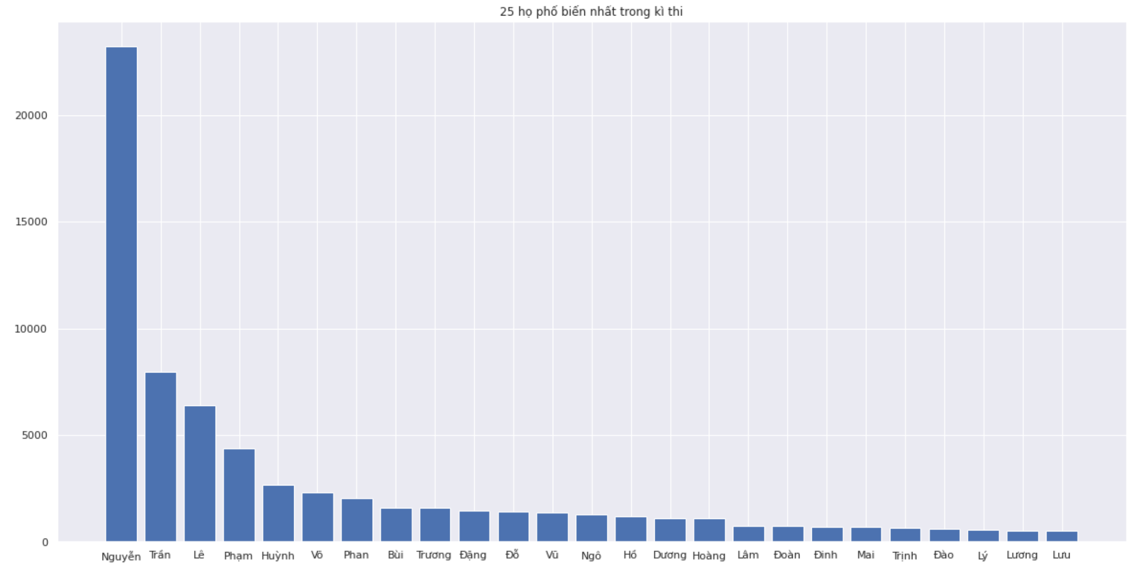


We can see the students in group 18 years old accounts more because it is a graduation exam but the average scores are about more than 6. The students in 17 years old group only have 2 but have the average nearly 8 which is the highest. The rest are students that retest and the average scores are about 6.

[46]:
```
with open("clean_data.csv", encoding="utf8") as file:
    data = file.read().split("\n")

header = data[0]
students = data[1:]

total_student = len(students)

header = header.split(",")
subjects = header[5:]

for i in range(len(students)):
    students[i] = students[i].split(",")

# remove last student (empty student)
students.pop()

name = [] # Danh sách các họ
name_count = [] # Số lần lặp của họ
```

```
for s in students:
    s_name = s[1].split(" ")
    lastname = s_name[0]
    if lastname not in name:
        name.append(lastname)
        name_count.append(0)
        name_count[name.index(lastname)] += 1
    else:
        name_count[name.index(lastname)] += 1



counted_max_num = [] # Số lần lặp lại các họ từ lớn đến bé
sort_index = [] # Danh sách vị trí sau khi đã sắp xếp

# Tạo counted_max_num, danh sách số lần lặp các họ lớn nhất
for i in range(len(name)):
    max_number = 0
    for j in range(len(name)):
        if name_count[j] > max_number and name_count[j] not in counted_max_num:
            max_number = name_count[j]
    counted_max_num.append(max_number)

# Tạo sort_index, vị trí bằng cách tìm vị trí của các con số lớn nhất từ counted_max_num
for max_num in counted_max_num:
        for i in range(len(name)):
                if name_count[i] == max_num and i not in sort_index:
                        sort_index.append(i)

name_sorted = [] # Danh sách họ đã sắp xếp
name_count_sorted = [] # Danh sách số lần lặp mỗi họ đã sắp xếp

# Dùng sort_index để sắp xếp lại họ và số lần lặp
for index in sort_index:
        name_sorted.append(name[index])
        name_count_sorted.append(name_count[index])

# print(name_sorted)
# print(name_count_sorted)

# Vẽ biểu đồ
# https://matplotlib.org/3.1.0/gallery/ticks_and_spines/custom_ticker1.html#sphx-glr-gallery-ticks-and-spines-custom-ticker1-py
import matplotlib.pyplot as plt
import numpy as np
```

```python
num = 25 # Số họ được vẽ

x = np.arange(num)
y = np.arange(num)

plt.figure(figsize = (20, 10))
plt.bar(x, name_count_sorted[0:num])

# label for column x
plt.xticks(x, name_sorted[0:num])
axis.set_ylabel('Số học sinh')
rects = axis.patches

# Make some labels.
# https://stackoverflow.com/questions/28931224/adding-value-labels-on-a-matplotlib-bar-chart
labels = name_count_sorted[0:num]
for rect, label in zip(rects, labels):
    height = rect.get_height()
    axis.text(rect.get_x() + rect.get_width() / 2, height + 2, label, ha='center', va='bottom')

plt.title(str(num) + ' họ phố biến nhất trong kì thi')

plt.show()
```



In Viet Nam, the most population last name is “Nguyễn”.

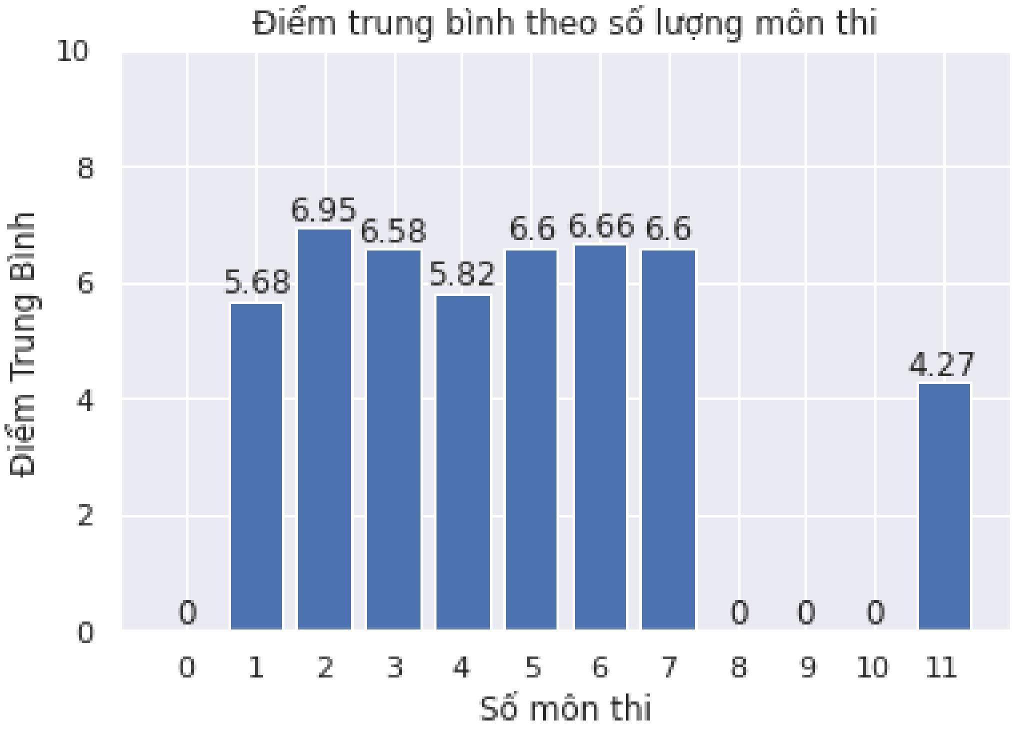
[74]:

```
# read file
with open("clean_data.csv", encoding="utf8") as file:
    data = file.read().split("\n")

header = data[0]
students = data[1:]

total_student = len(students)

# split header
header = header.split(",")
subjects = header[5:]

# turn each student to a list
for i in range(len(students)):
    students[i] = students[i].split(",")

students.pop()
not_take_exam = [0,0,0,0,0,0,0,0,0,0,0]
# number of students who took 0,1,2,3,... subjects
num_of_exam_taken = [0,0,0,0,0,0,0,0,0,0,0,0]
average = [0,0,0,0,0,0,0,0,0,0,0,0]

for s in students:

    count = 0
    total = 0
    for i in range(11):
        if s[i+5] != "-1":
            total += float(s[i+5])
            count += 1
    if count == 11 :
        print(s)

    num_of_exam_taken[count] += 1
    average[count] += total/count

for i in range(12):
    if num_of_exam_taken[i] != 0:
        average[i] = round(average[i]/num_of_exam_taken[i], 2)

# print(num_of_exam_taken)
# print(average)


x = np.arange(12)
y = np.arange(12)
```

```
plt.figure(figsize = (20, 10))

fig, axis = plt.subplots()
plt.bar(x, average)

# set limit
axis.set_ylim(0,10)

# label for column x
plt.xticks(x, y)

axis.set_ylabel('Điểm Trung Bình')
axis.set_xlabel('Số môn thi')

rects = axis.patches

# Make some labels.
# https://stackoverflow.com/questions/28931224/
 ↪adding-value-labels-on-a-matplotlib-bar-chart
labels = average
for rect, label in zip(rects, labels):
    height = rect.get_height()
    axis.text(rect.get_x() + rect.get_width() / 2, height, label,ha='center',␣
 ↪va='bottom')

plt.title('Điểm trung bình theo số lượng môn thi')

plt.show()
print(average)
print(num_of_exam_taken)
```

```
['02050326', 'Trần Ngọc Minh Châu', '28', '4', '2001', '6.00', '7.00', '7.08',
'0', '5.50', '7.75', '8.00', '0.00', '0.00', '0.00', '5.60']
```

```
<Figure size 1440x720 with 0 Axes>
```



```
[0, 5.68, 6.95, 6.58, 5.82, 6.6, 6.66, 6.6, 0, 0, 0, 4.27]
[0, 80, 122, 2598, 4334, 318, 2730, 64261, 0, 0, 0, 1]
```

We can see from the graph that students focus on range from 5 to 7 subjects that is compulsory. From the list I prints, we can know that after took exams in subject 1 and 2 (Math and Literature) there is not much difference skip exam.But after 2 subjects, 2598 students countinue to skip exam but the scores is not much difference. So I think that not because of low scores is the reason that they skip exam maybe it depends on many reasons. And from the graph and list we can see it has one student that take part in 11 subjects which is a excited infromation. We can see the name of student that take 11 subject above.

[55]:
```
with open("clean_data.csv", encoding="utf8") as file:
    data = file.read().split("\n")

header = data[0]
students = data[1:]

total_student = len(students)

# split header
header = header.split(",")
subjects = header[5:]

# turn each student to a list
```

```python
for i in range(len(students)):
    students[i] = students[i].split(",")

# remove empty list (end of file)
students.pop()

max_name_length = 0
index = 0
for i in range(len(students)):

    if len(students[i][1]) >= max_name_length:
        max_name_length = len(students[i][1])
        index = i

# In số báo danh
print(students[index][0])
# In tên
print(students[index][1])
```

```
02033237
Đoàn Huỳnh Nguyễn Châu Thanh Tú
```

It is a student has the longest name in the exams.