In [1]:

```
import os
import pandas as pd
import numpy as np
```

Read Data

列：学生性别、种族、父母教育情况、午餐情况、考试准备课程完成情况、数学分数、阅读分数以及写作分数

In [2]:

```
data = pd.read_csv("exams.csv") # 读取数据
data.head(5) # 显示dataframe中的前5行数据
```

Out[2]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	male	group A	high school	standard	completed	67	67	63
1	female	group D	some high school	free/reduced	none	40	59	55
2	male	group E	some college	free/reduced	none	59	60	50
3	male	group B	high school	standard	none	77	78	68
4	male	group E	associate's degree	standard	completed	78	73	68

Tasks

In [3]:

```
df = data.copy()
```

NO.1 - Fill the missing data

使用恰当的方式填补缺失值，如有需要可对数据进行数据变换。

In [4]:

```
df.info() # 查看缺失值 —— 无缺失值
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64 
 6   reading score                1000 non-null   int64 
 7   writing score                1000 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

NO.2

- 筛选所有性别是female的女生，计算她们writing score的均值

In [5]:

```
df_female = df[df['gender']=='female'] # 筛选出所有性别为female的女生
print("Average writing score of girls: {}".format(
    np.mean(df_female["writing score"]))) # 计算均值
```

```
Average writing score of girls: 71.7080745341615
```

NO.3

- 父母教育程度是some college的学生有多少个，占所有学生的比例是多少

In [6]:

```
'''父母教育程度为some college的学生数量'''
len(df[df["parental level of education"]=="some college"])
```

Out[6]:

```
222
```

In [7]:

```
'''占比'''
len(df[df["parental level of education"]=="some college"]) / len(df)
```

Out[7]:

```
0.222
```

NO.4

- 请在所有学生中随机抽取十个，并计算他们math score的方差

In [8]:

```
# 随机取出10个学生的index (id)
rand_stu_index = np.random.randint(df.index.start, df.index.stop, size=10)
rand_ten_stu = df.iloc[rand_stu_index] # 从表中找到对应的学生
print("随机十个学生的math score方差: {}".format(np.var(
    rand_ten_stu["math score"])))
```

```
随机十个学生的math score方差: 255.08999999999997
```

NO.5

- 请使用循环语句打印前5个学生的种族，结果形如：学生1的种族为XXX

In [9]:

```
for i in range(5):
    print("学生{}的种族为{}".format(i, df.at[i, "race/ethnicity"]))
```

```
学生0的种族为group A
学生1的种族为group D
学生2的种族为group E
学生3的种族为group B
学生4的种族为group E
```

NO.6

- 为该数据增添新的一列，命名为‘新属性’。
- 计算所有学生的reading score均值，若一学生reading score大于该均值，‘新属性’取值为“high”；否则为“low”

In [10]:

```
# 计算所有学生的reading score均值
mean_reading_score = df["reading score"].mean()
df_new = df.copy()
df_new["新属性"] = df["reading score"].map(
    lambda x: "high" if x>mean_reading_score else "low") # 建新的列
```

In [11]:

```
df_new.head(5)
```

Out[11]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	新属性
0	male	group A	high school	standard	completed	67	67	63	low
1	female	group D	some high school	free/reduced	none	40	59	55	low
2	male	group E	some college	free/reduced	none	59	60	50	low
3	male	group B	high school	standard	none	77	78	68	high
4	male	group E	associate's degree	standard	completed	78	73	68	high

NO.7

- 任选角度，绘制不少于三个不同角度描述性统计图，尽量练习不同类型的图表。

In [12]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

Fig.1. 不同民族的学生人数柱状图

通过描述性统计可知：民族A的学生最少，民族C的学生最多；

In [13]:

```
# 民族列表 ['group A', 'group D', 'group E', 'group B', 'group C']
race_list = df["race/ethnicity"].unique()
# 各个民族学生的人数
race_num = [len(df[df["race/ethnicity"]==each]) for each in race_list]
plt.title("Different race student number")
plt.bar(race_list, race_num)
plt.show()
```

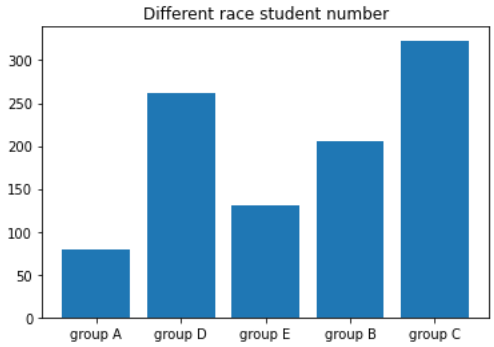


Fig.2. 各个考试分数的分布直方图

通过描述性统计分析可以得出：三门不同科目的考试成绩分布相近；成绩较低与较高的学生数量较少，成绩中等的学生数量较多。

In [14]:

```
f, [ax1, ax2, ax3] = plt.subplots(1,3, figsize=(15,5))
ax1.set_title('Math Score histgram')
ax1.hist(df["math score"])
ax2.set_title('Reading Score histgram')
ax2.hist(df["reading score"])
ax3.set_title('Writing Score histgram')
ax3.hist(df["writing score"])
plt.show()
```

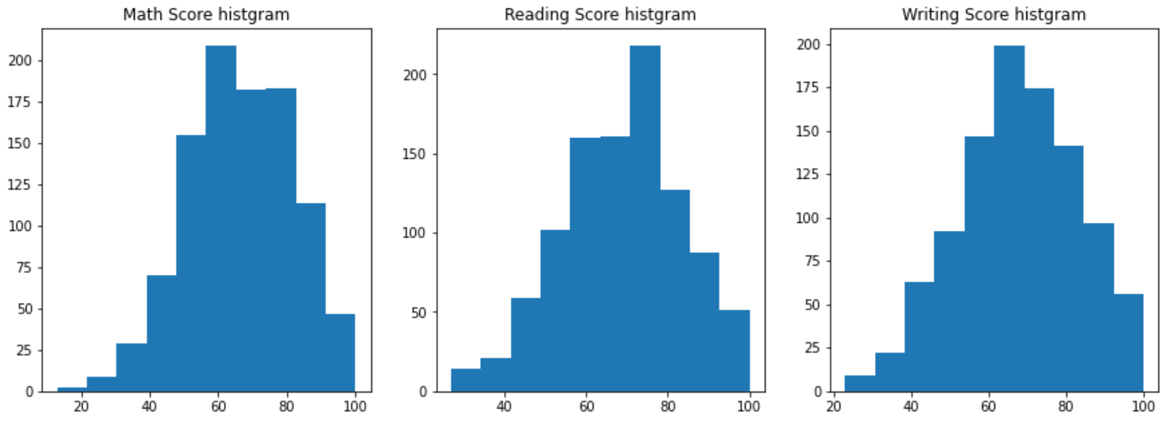


Fig.3 父母教育情况与孩子均分的关系

根据小提琴图和箱线图可以看出，父母受教育情况的不同，孩子的成绩分布也不同。

总体来看，学士学位、副学士学位和硕士学位家长的孩子，孩子平均成绩的中位数较高，成绩分布整体偏高。且平均成绩满分的学生均来自于本科及以上学历的家长。

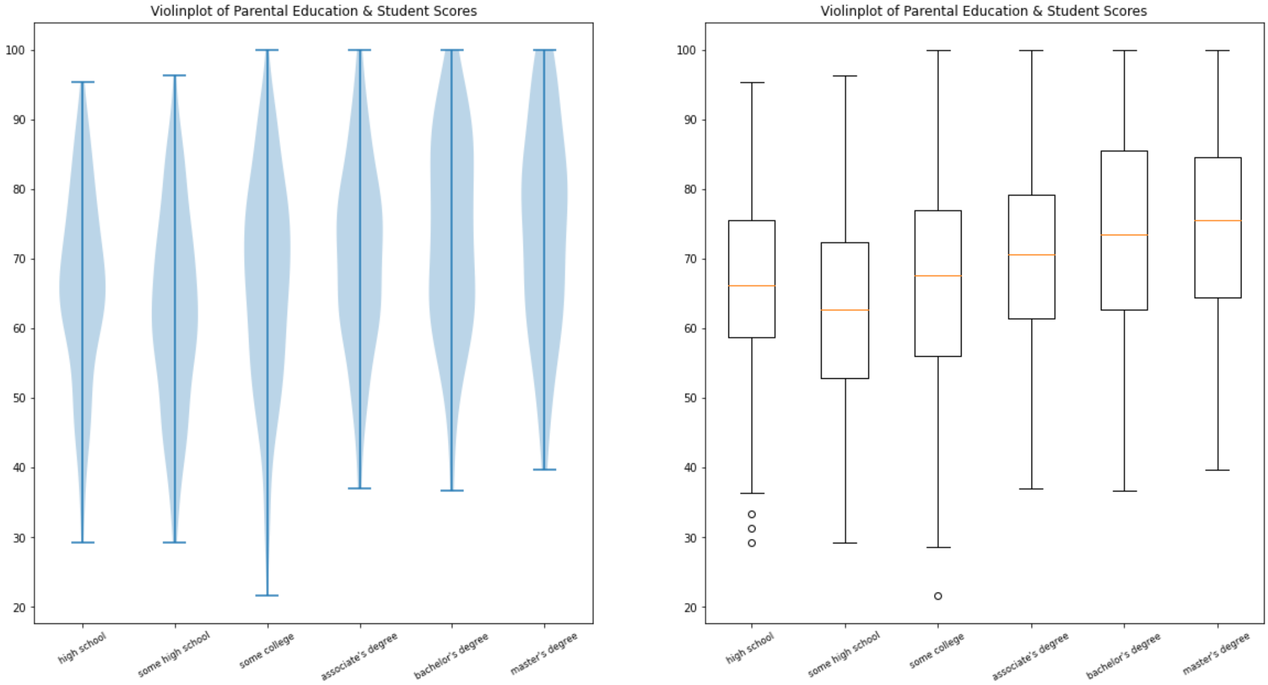
但家长的学历背景对孩子成绩的影响并不绝对，来自高中学历背景家庭的孩子，也有相当一部分人取得了均分90+的高分。

In [15]:

```
# 不同的家长教育背景
parent_edu = df["parental level of education"].unique()
# 不同家长教育背景的孩子，三门课程的平均分
diff_parent_edu_mean_stu = [df[df["parental level of education"]==each]
                            [["math score","reading score", "writing score"]]
                            .mean(axis=1)
                            for each in parent_edu]
```

In [16]:

```
f, [ax1, ax2] = plt.subplots(1,2, figsize=(20,10))
ax1.set_title("Violinplot of Parental Education & Student Scores")
ax1.violinplot(diff_parent_edu_mean_stu)
ax1.set_xticks([y + 1 for y in range(len(parent_edu))])
ax1.set_xticklabels(list(parent_edu),rotation = 30,fontsize = 'small')
ax2.set_title("Violinplot of Parental Education & Student Scores")
ax2.boxplot(diff_parent_edu_mean_stu)
ax2.set_xticks([y + 1 for y in range(len(parent_edu))])
ax2.set_xticklabels(list(parent_edu),rotation = 30,fontsize = 'small')
plt.show()
```


Violinplot of Parental Education & Student Scores
high school
some high school
some college
associate's degree
bachelor's degree
master's degree
Violinplot of Parental Education & Student Scores
high school
some high school
some college
associate's degree
bachelor's degree
master's degree