

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 1
по дисциплине «Разработка нейросетевых систем»

Тема: «Создание истории о данных»

ИСПОЛНИТЕЛЬ:

группа ИУ5-24М

Кравцов А.Н.

ФИО

подпись

"26" апреля 2024 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

ФИО

подпись

"__" _____ 2024 г.

Москва – 2024

Задание

- Выбрать набор данных (датасет).
 - Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:
 1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
 2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
 3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
 4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
 - История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.
- Сформировать отчет и разместить его в своем репозитории на github.

Загрузка датасета

```
df = pd.read_csv("StudentsPerformance.csv")
print(df)
```

```
gender race/ethnicity parental level of education lunch \
0 female group B bachelor's degree standard
1 female group C some college standard
2 female group B master's degree standard
3 male group A associate's degree free/reduced
4 male group C some college standard
.. ...
995 female group E master's degree standard
996 male group C high school free/reduced
997 female group C high school free/reduced
998 female group D some college standard
999 female group D some college free/reduced

test preparation course math score reading score writing score
0 none 72 72 74
1 completed 69 90 88
2 none 90 95 93
3 none 47 57 44
4 none 76 78 75
.. ...
995 completed 88 99 95
996 none 62 55 55
997 completed 59 71 65
998 completed 68 78 77
999 none 77 86 86
```

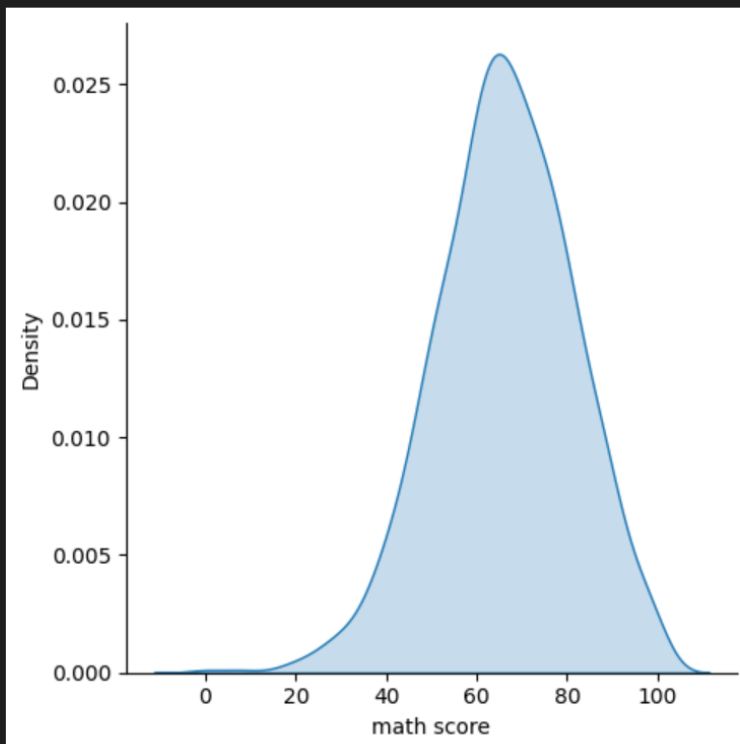
[1000 rows x 8 columns]

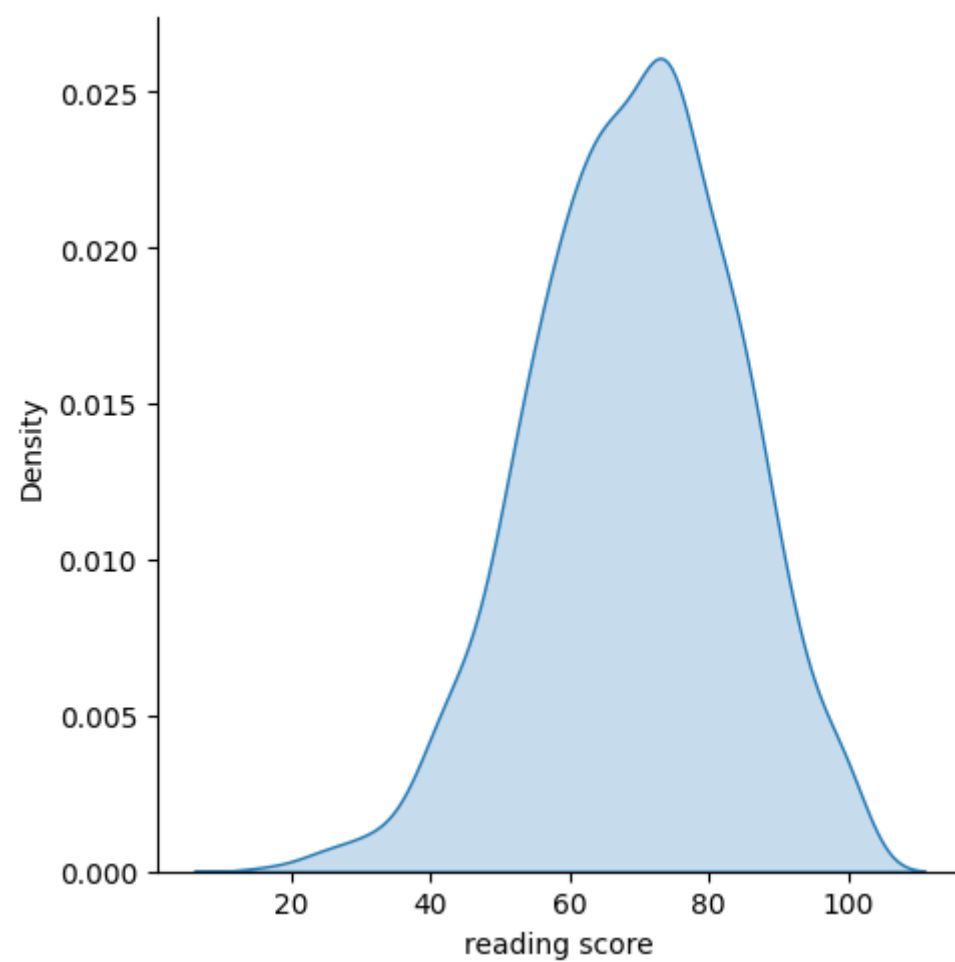
DENSITY

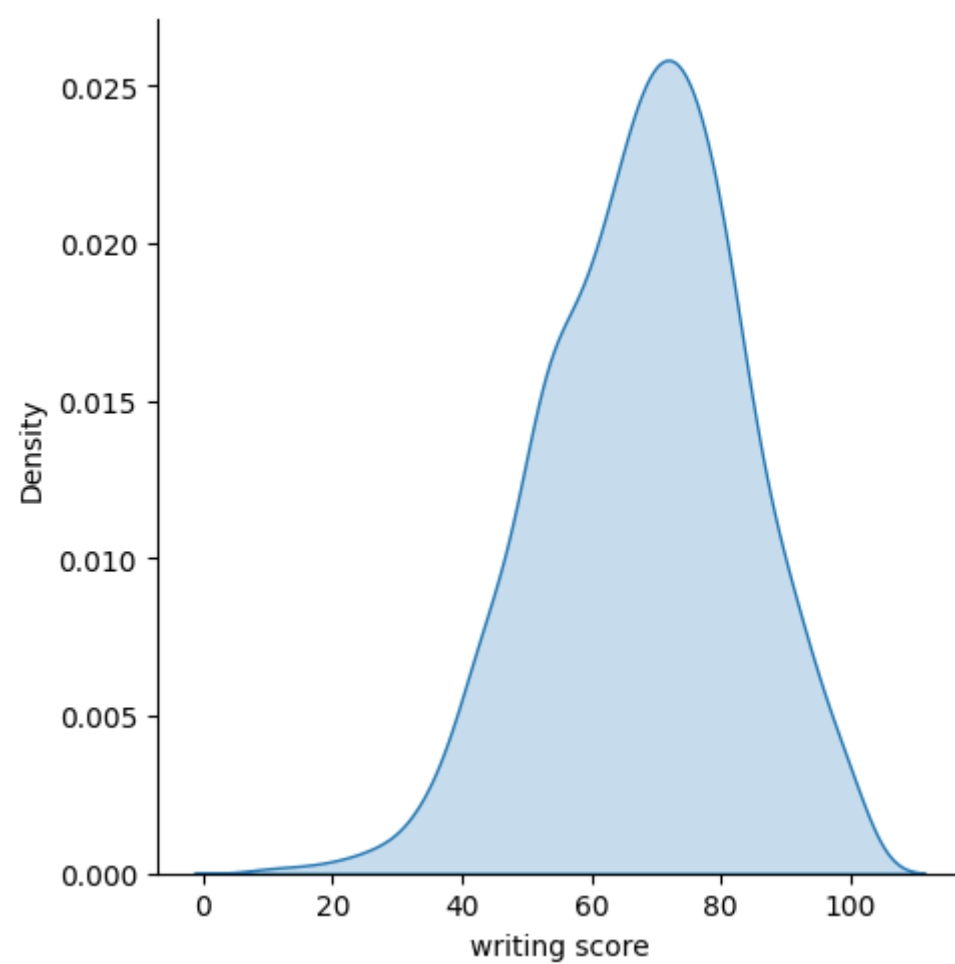
```
p = sns.displot(df, x="math score", kind="kde", fill=True)

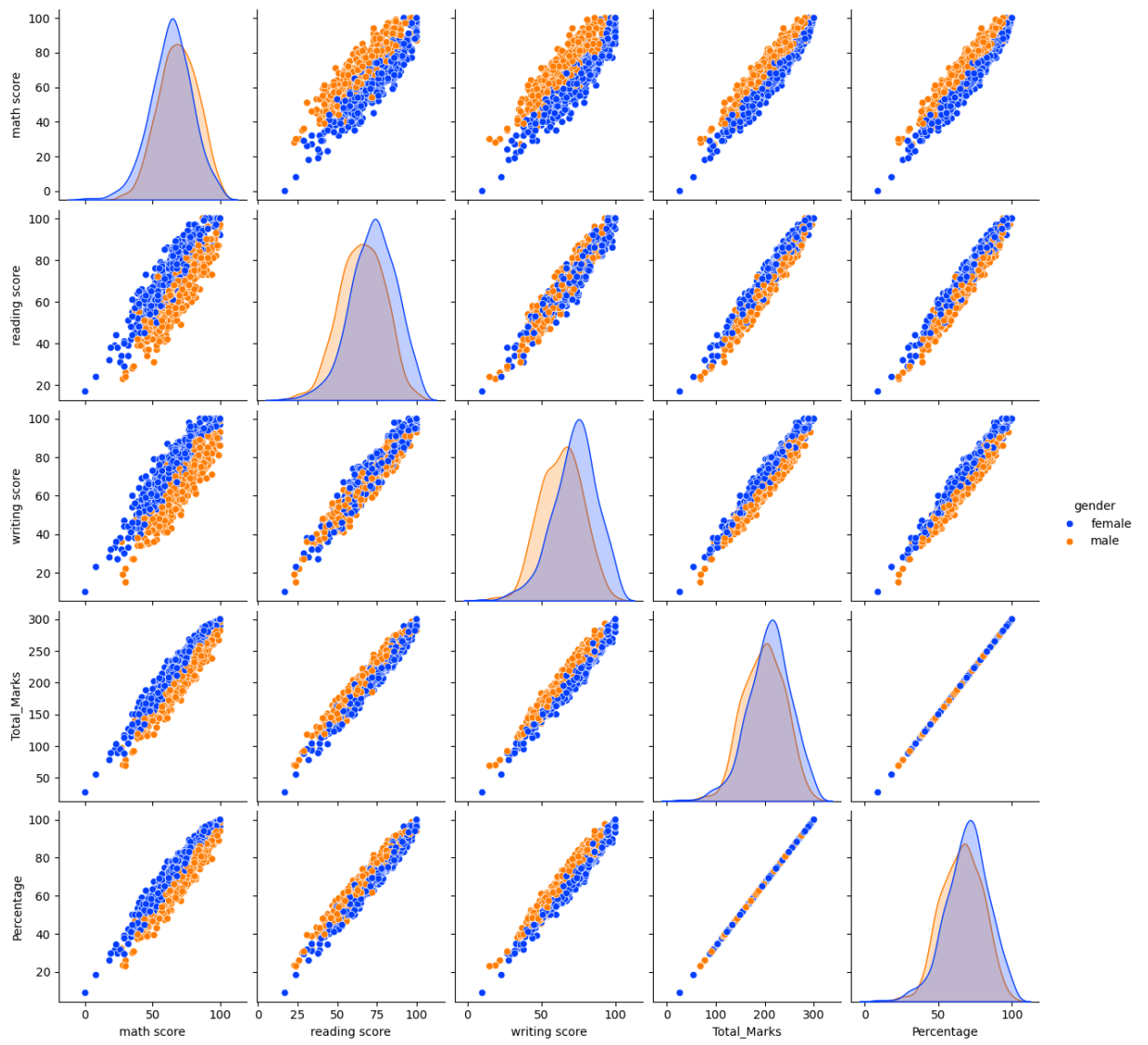
p = sns.displot(df, x="reading score", kind="kde", fill=True)

p = sns.displot(df, x="writing score", kind="kde", fill=True)
```







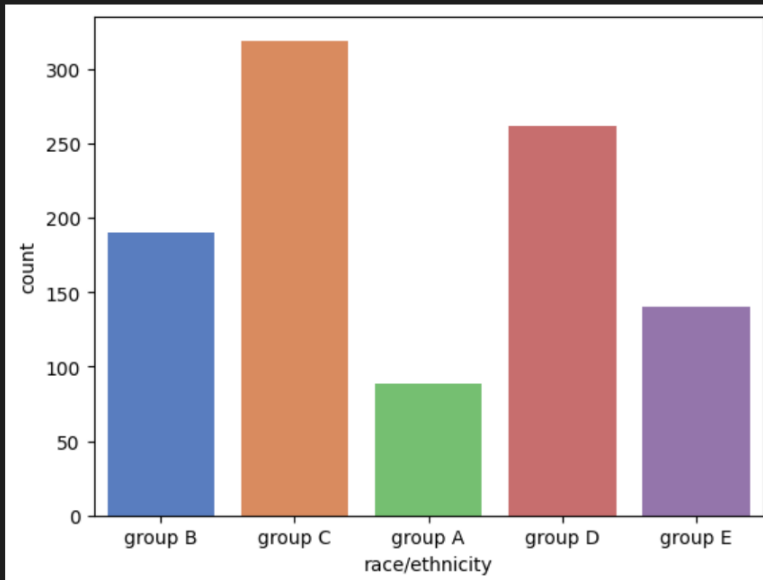


```
p = sns.countplot(x="race/ethnicity", data = df, palette="muted")
```

C:\Users\kravtandr\AppData\Local\Temp\ipykernel_10440\2662061580.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue`

```
p = sns.countplot(x="race/ethnicity", data = df, palette="muted")
```

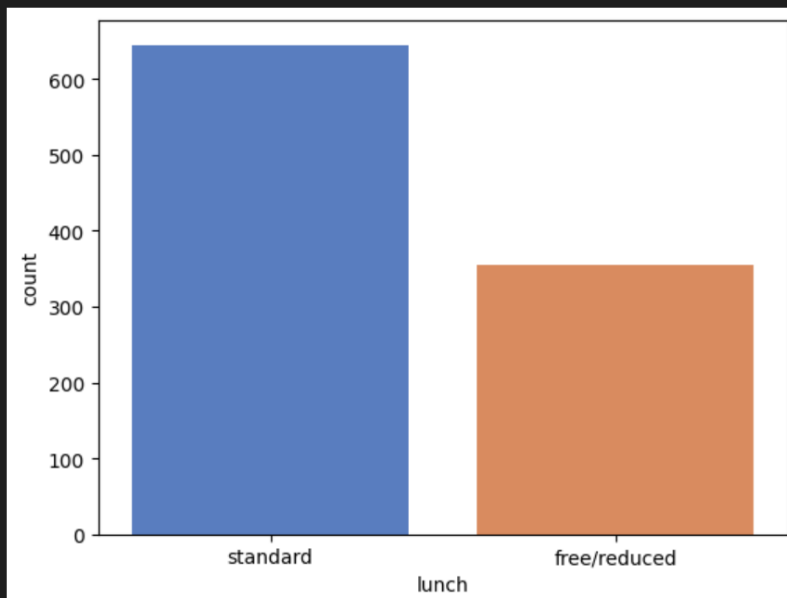


```
p = sns.countplot(x="lunch", data = df, palette="muted")
```

C:\Users\kravtandr\AppData\Local\Temp\ipykernel_10440\3781548851.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue`

```
p = sns.countplot(x="lunch", data = df, palette="muted")
```

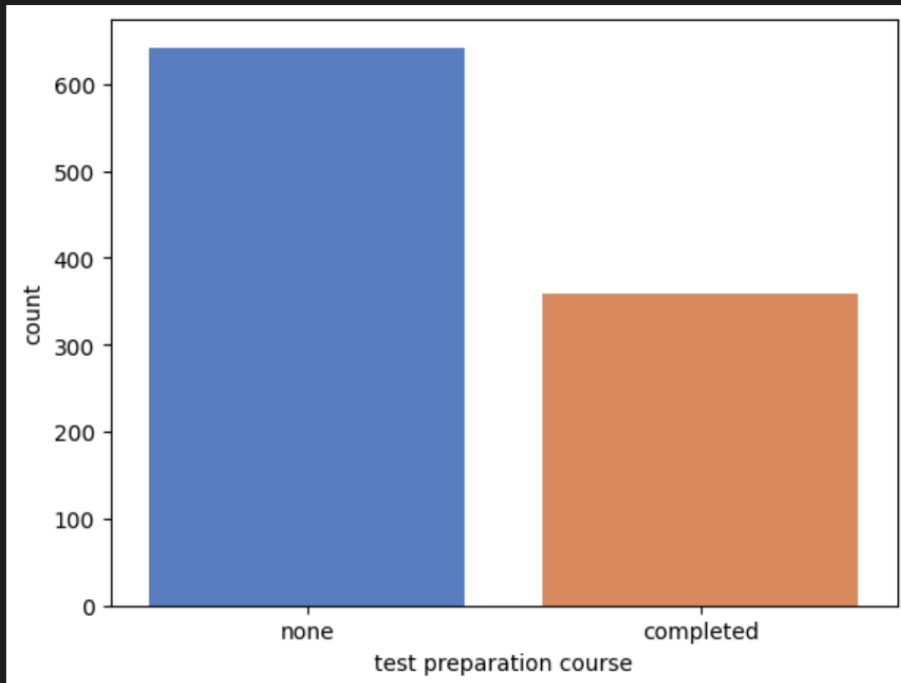


```
p = sns.countplot(x="test preparation course", data = df, palette="muted")
```

C:\Users\kravtandr\AppData\Local\Temp\ipykernel_10440\305923387.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

```
p = sns.countplot(x="test preparation course", data = df, palette="muted")
```

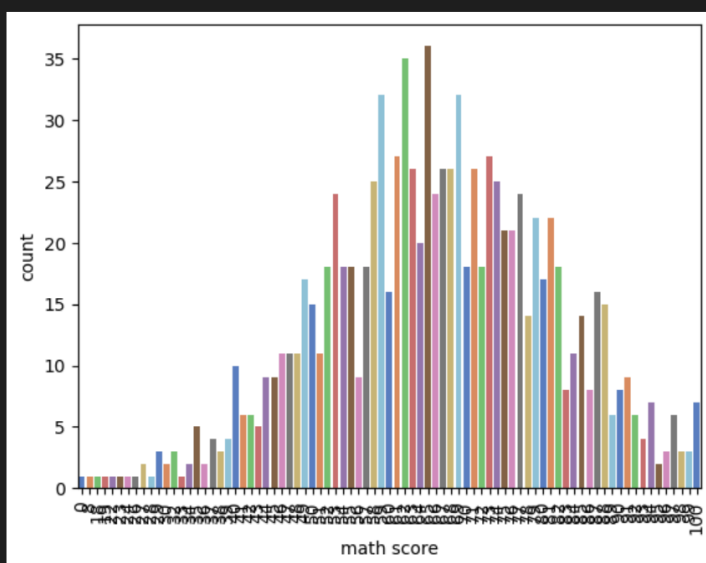


```
p = sns.countplot(x="math score", data = df, palette="muted")
```

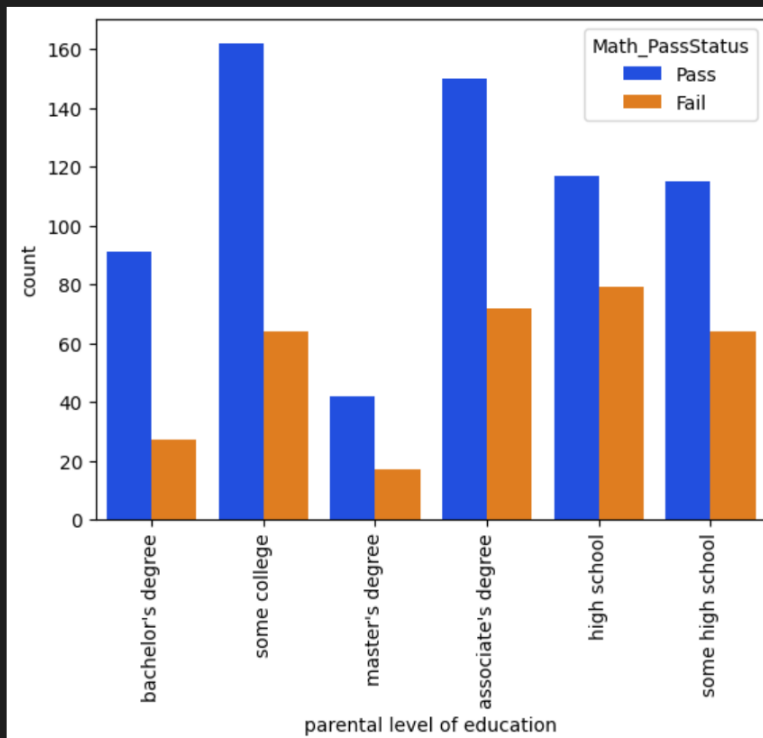
C:\Users\kravtandr\AppData\Local\Temp\ipykernel_10440\1023825716.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and s

```
p = sns.countplot(x="math score", data = df, palette="muted")
```




```
p = sns.countplot(x='parental level of education', data = df, hue='Math_PassStatus', palette='bright')
_ = plt.setp(p.get_xticklabels(), rotation=90)
```



```
df['Reading_PassStatus'] = np.where(df['reading score'] < passmark, 'Fail', 'Pass')
df['Writing_PassStatus'] = np.where(df['writing score'] < passmark, 'Fail', 'Pass')

df['OverAll_PassStatus'] = df.apply(lambda x : 'Fail' if x['Math_PassStatus'] == 'Fail' or
                                     x['Reading_PassStatus'] == 'Fail' or x['Writing_PassStatus'] == 'Fail' else 'Pass', axis = 1)

df.OverAll_PassStatus.value_counts()
```

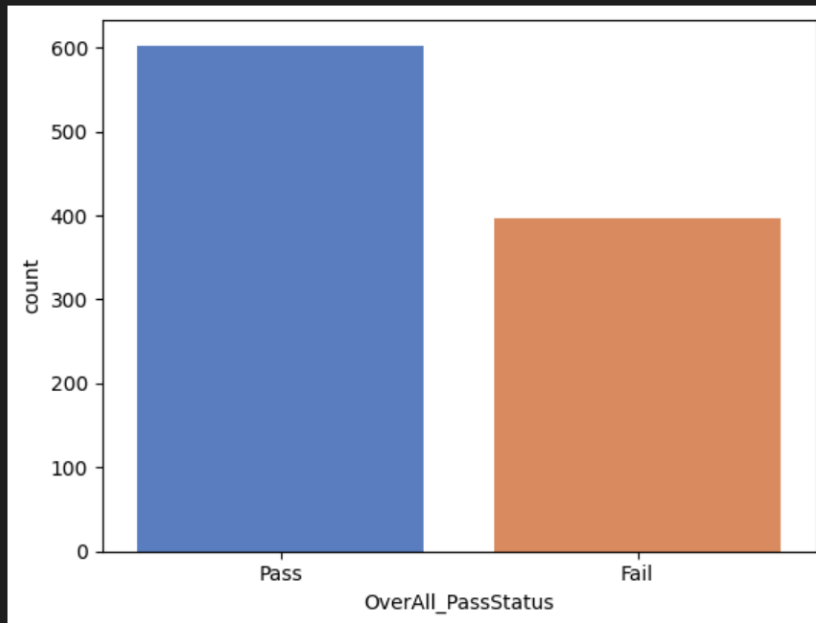
```
OverAll_PassStatus
Pass    603
Fail    397
Name: count, dtype: int64
```

```
sns.countplot(x="OverAll_PassStatus", data = df, order=['Pass','Fail'], palette="muted")
plt.show()
```

C:\Users\kravtandr\AppData\Local\Temp\ipykernel_10440\1239282655.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable

```
sns.countplot(x="OverAll_PassStatus", data = df, order=['Pass','Fail'], palette="muted")
```



```
def GetGrade(Percentage, OverAll_PassStatus):
    if ( OverAll_PassStatus == 'неуд'):
        return 'неуд'
    if ( Percentage >= 85 ):
        return 'отл'
    if ( Percentage >= 70 ):
        return 'хор'
    if ( Percentage >= 60 ):
        return 'удовл'
    else:
        return 'неуд'

df['Grade'] = df.apply(lambda x : GetGrade(x['Percentage'], x['OverAll_PassStatus']), axis=1)

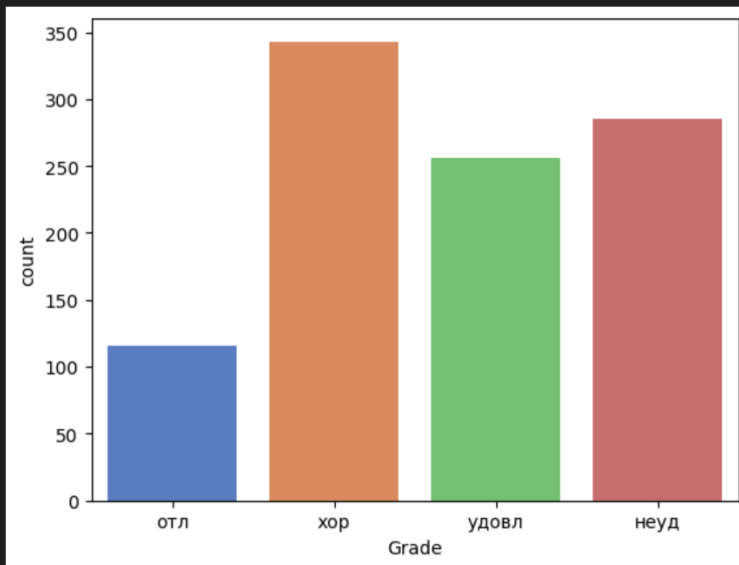
df.Grade.value_counts()
```

```
Grade
хор      343
неуд     285
удовл    256
отл      116
Name: count, dtype: int64
```

C:\Users\kravtandr\AppData\Local\Temp\ipykernel_10440\3962153730.py:1: FutureWarning:

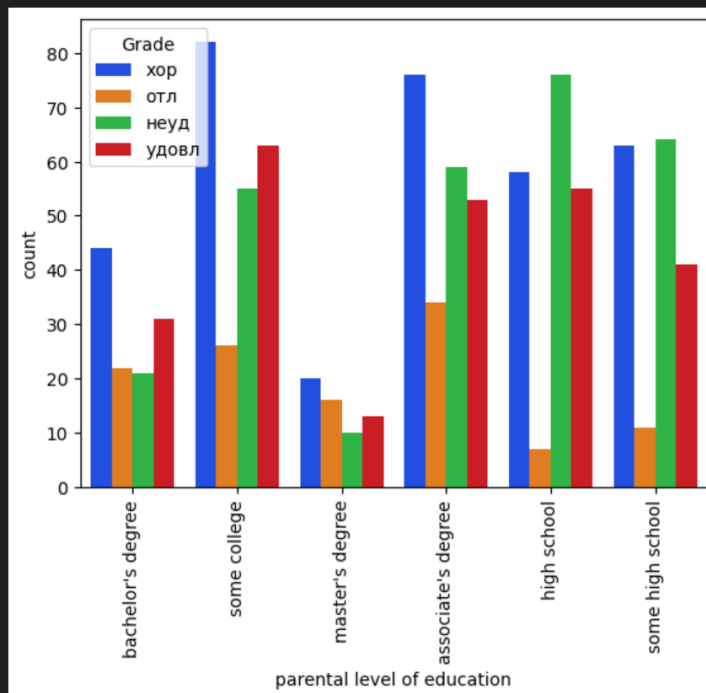
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue`

```
sns.countplot(x="Grade", data = df, order=['отл','хор','удовл','неуд'], palette="muted")
```

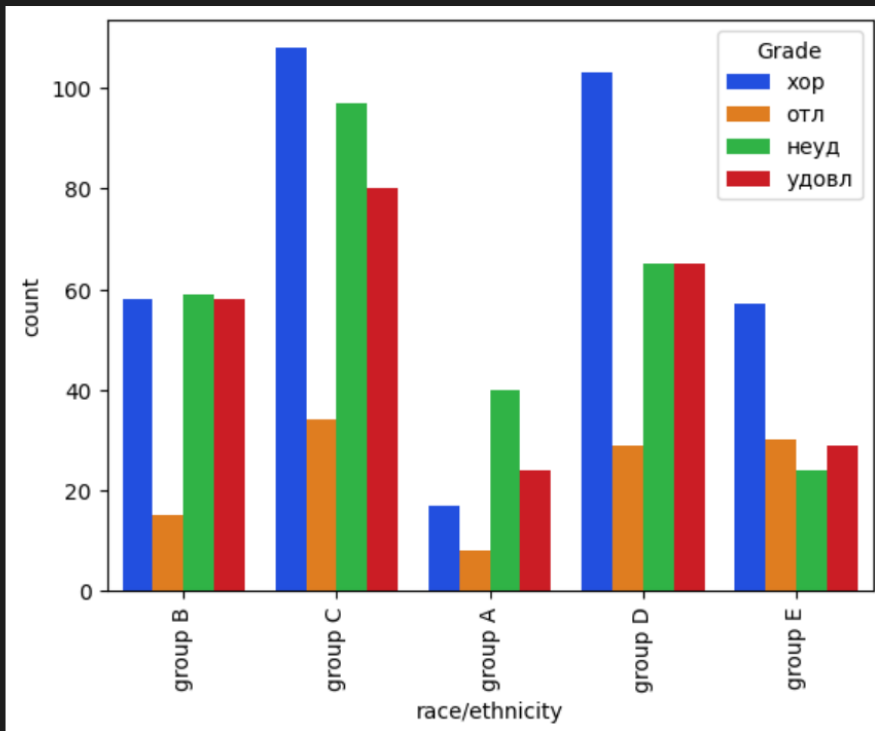


```
p = sns.countplot(x='parental level of education', data = df, hue='Grade', palette='bright')
_ = plt.setp(p.get_xticklabels(), rotation=90)
```

[27]



```
p = sns.countplot(x='race/ethnicity', data = df, hue='Grade', palette='bright')
_ = plt.setp(p.get_xticklabels(), rotation=90)
```



```
p = sns.countplot(x='lunch', data = df, hue='Grade', palette='bright')
_ = plt.setp(p.get_xticklabels(), rotation=90)
```

