```
import pandas as pd
import numpy as np
import seaborn as sns
import  matplotlib.pyplot as plt
```

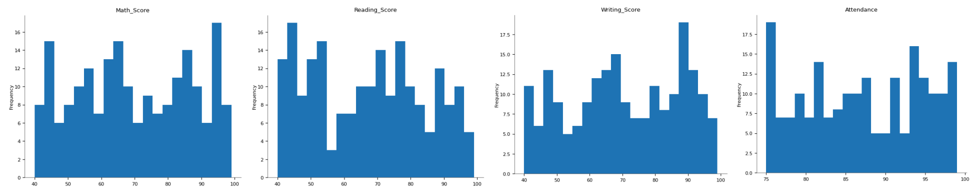
```
dt = pd.read_csv('/content/student_data.csv')
```

```
dt
```

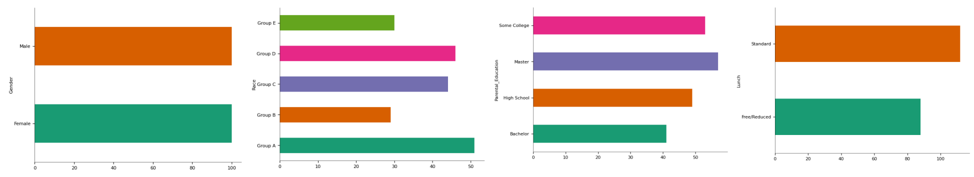
3	Male	Group D	Bachelor	Free/Reduced	NaN	66	70	75	82
4	Male	Group E	Some College	Standard	Completed	90	79	75	76
...	...	...	...	...	...	...	...	...	...
195	Female	Group C	Bachelor	Standard	NaN	69	43	93	97
196	Female	Group B	Bachelor	Free/Reduced	NaN	70	51	96	79
197	Female	Group B	High School	Free/Reduced	NaN	63	84	86	96
198	Male	Group B	Master	Standard	NaN	94	41	76	90
199	Male	Group A	Bachelor	Standard	NaN	48	66	42	80

200 rows × 11 columns

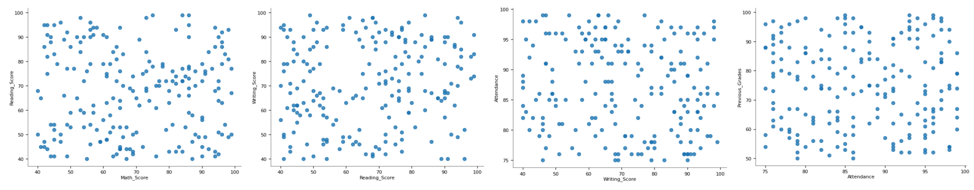
Distributions



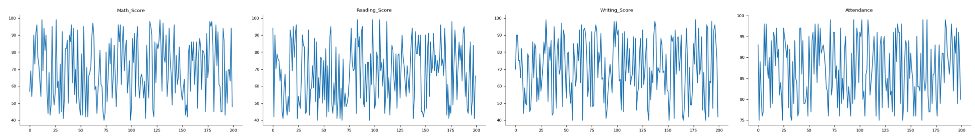
Categorical distributions



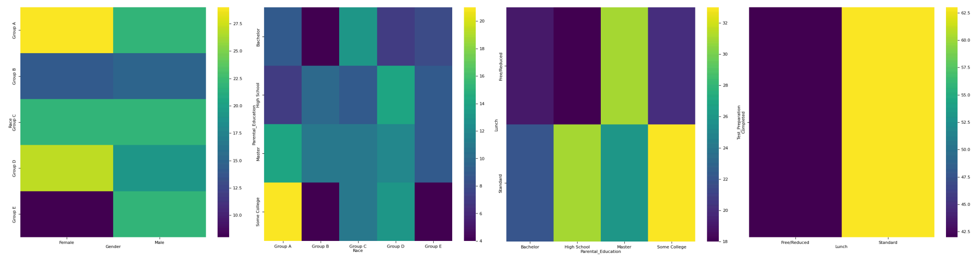
2-d distributions



Values



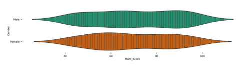
2-d categorical distributions



Faceted distributions

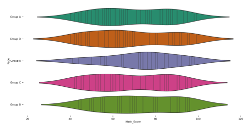
```
<string>:5: FutureWarning:
```

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



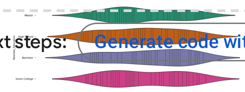
```
<string>:5: FutureWarning:

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



```
<string>:5: FutureWarning:

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



```
<string>:5: FutureWarning:
```

Next steps: Generate code with dt | View recommended plots | New interactive sheet

```
dt.describe()
```

	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Grades
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	69.910000	67.320000	70.635000	87.095000	74.825000
std	17.482278	17.426013	17.266803	7.370727	14.576927
min	40.000000	40.000000	40.000000	75.000000	50.000000
25%	56.000000	51.000000	57.750000	80.750000	62.000000
50%	69.000000	68.500000	70.000000	87.000000	74.000000
75%	86.000000	80.250000	86.250000	94.000000	88.000000
max	99.000000	99.000000	99.000000	99.000000	99.000000

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Gender              200 non-null    object
 1   Race                200 non-null    object
 2   Parental_Education  200 non-null    object
 3   Lunch               200 non-null    object
 4   Test_Preparation    105 non-null    object
 5   Math_Score          200 non-null    int64 
 6   Reading_Score       200 non-null    int64 
 7   Writing_Score       200 non-null    int64 
 8   Attendance          200 non-null    int64 
 9   Previous_Grades     200 non-null    int64 
 10  Performance         200 non-null    object
dtypes: int64(5), object(6)
memory usage: 17.3+ KB
```

```
dt['Gender'].unique()
```

```
array(['Male', 'Female'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l = LabelEncoder()
```

```
dt['Gender'] = l.fit_transform(dt['Gender'])
```

```
dt['Gender'].unique()
```

```
array([1, 0])
```

```
dt
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Gr
0	1	Group A	Master	Standard	NaN	57	94	70	93	
1	0	Group D	High School	Standard	Completed	69	42	90	75	
2	1	Group E	High School	Free/Reduced	Completed	54	90	90	89	
3	1	Group D	Bachelor	Free/Reduced	NaN	66	70	75	82	
4	1	Group E	Some College	Standard	Completed	90	79	75	76	
...	...	...	...	...	...	...	...	...	...	
195	0	Group C	Bachelor	Standard	NaN	69	43	93	97	
196	0	Group B	Bachelor	Free/Reduced	NaN	70	51	96	79	

Next steps: Generate code with dt View recommended plots New interactive sheet

```
dt['Race'].unique()
```

```
array(['Group A', 'Group D', 'Group E', 'Group C', 'Group B'],
      dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l = LabelEncoder()
```

```
dt['Race'] = l.fit_transform(dt['Race'])
```

```
dt['Race'].unique()
```

```
array([0, 3, 4, 2, 1])
```

```
dt
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Gra
0	1	0	Master	Standard	NaN	57	94	70	93	
1	0	3	High School	Standard	Completed	69	42	90	75	
2	1	4	High School	Free/Reduced	Completed	54	90	90	89	
3	1	3	Bachelor	Free/Reduced	NaN	66	70	75	82	
4	1	4	Some College	Standard	Completed	90	79	75	76	
...	...	...	...	...	...	...	...	...	...	
195	0	2	Bachelor	Standard	NaN	69	43	93	97	
196	0	1	Bachelor	Free/Reduced	NaN	70	51	96	79	
197	0	1	High School	Free/Reduced	NaN	63	84	86	96	
198	1	1	Master	Standard	NaN	94	41	76	90	
199	1	0	Bachelor	Standard	NaN	48	66	42	80	

200 rows × 11 columns

Next steps: Generate code with dt View recommended plots New interactive sheet

```
dt['Parental_Education'].unique()
```

```
array(['Master', 'High School', 'Bachelor', 'Some College'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder

l = LabelEncoder()

dt['Parental_Education'] = l.fit_transform(dt['Parental_Education'])

dt['Parental_Education'].unique()
```

```
array([2, 1, 0, 3])
```

```
dt
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Gra
0	1	0	2	Standard	NaN	57	94	70	93	
1	0	3	1	Standard	Completed	69	42	90	75	
2	1	4	1	Free/Reduced	Completed	54	90	90	89	
3	1	3	0	Free/Reduced	NaN	66	70	75	82	
4	1	4	3	Standard	Completed	90	79	75	76	
...	...	...	...	...	...	...	...	...	...	
195	0	2	0	Standard	NaN	69	43	93	97	
196	0	1	0	Free/Reduced	NaN	70	51	96	79	
197	0	1	1	Free/Reduced	NaN	63	84	86	96	
198	1	1	2	Standard	NaN	94	41	76	90	
199	1	0	0	Standard	NaN	48	66	42	80	

200 rows × 11 columns

Next steps: Generate code with dt View recommended plots New interactive sheet

```
dt['Lunch'].unique()
```

```
array(['Standard', 'Free/Reduced'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder

l = LabelEncoder()

dt['Lunch'] = l.fit_transform(dt['Lunch'])

dt['Lunch'].unique()
```

```
array([1, 0])
```

```
dt
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Grades	Pe
0	1	0	2	1	NaN	57	94	70	93	57	
1	0	3	1	1	Completed	69	42	90	75	74	
2	1	4	1	0	Completed	54	90	90	89	67	
3	1	3	0	0	NaN	66	70	75	82	74	
4	1	4	3	1	Completed	90	79	75	76	61	
...	...	...	...	...	...	...	...	...	...	...	
195	0	2	0	1	NaN	69	43	93	97	86	
196	0	1	0	0	NaN	70	51	96	79	97	
197	0	1	1	0	NaN	63	84	86	96	62	
198	1	1	2	1	NaN	94	41	76	90	77	
199	1	0	0	1	NaN	48	66	42	80	54	

200 rows × 11 columns

Next steps: Generate code with dt View recommended plots New interactive sheet

```
dt['Test_Preparation'].unique()
```

```
array([nan, 'Completed'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l = LabelEncoder()
```

```
dt['Test_Preparation'] = l.fit_transform(dt['Test_Preparation'])
```

```
dt['Test_Preparation'].unique()
```

```
array([1, 0])
```

```
dt
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Grades	Pe
0	1	0	2	1	1	57	94	70	93	57	
1	0	3	1	1	0	69	42	90	75	74	
2	1	4	1	0	0	54	90	90	89	67	
3	1	3	0	0	1	66	70	75	82	74	
4	1	4	3	1	0	90	79	75	76	61	
...	...	...	...	...	...	...	...	...	...	...	
195	0	2	0	1	1	69	43	93	97	86	
196	0	1	0	0	1	70	51	96	79	97	
197	0	1	1	0	1	63	84	86	96	62	
198	1	1	2	1	1	94	41	76	90	77	
199	1	0	0	1	1	48	66	42	80	54	

200 rows × 11 columns

Next steps: Generate code with dt View recommended plots New interactive sheet

```
dt['Performance'].unique()
```

```
array(['High', 'Low'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l = LabelEncoder()
```

```
dt['Performance'] = l.fit_transform(dt['Performance'])
```

```
dt['Performance'].unique()
```

```
array([0, 1])
```

```
dt
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Grades	Pe
0	1	0	2	1	1	57	94	70	93	57	
1	0	3	1	1	0	69	42	90	75	74	
2	1	4	1	0	0	54	90	90	89	67	
3	1	3	0	0	1	66	70	75	82	74	
4	1	4	3	1	0	90	79	75	76	61	
...	...	...	...	...	...	...	...	...	...	...	
195	0	2	0	1	1	69	43	93	97	86	
196	0	1	0	0	1	70	51	96	79	97	
197	0	1	1	0	1	63	84	86	96	62	
198	1	1	2	1	1	94	41	76	90	77	
199	1	0	0	1	1	48	66	42	80	54	

200 rows × 11 columns

Next steps: Generate code with dt View recommended plots New interactive sheet

```
from sklearn.model_selection import train_test_split
```

```
x = dt.drop(['Performance'],axis=1)
y = dt['Performance']
```

```
x
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Grades
0	1	0	2	1	1	57	94	70	93	57
1	0	3	1	1	0	69	42	90	75	74
2	1	4	1	0	0	54	90	90	89	67
3	1	3	0	0	1	66	70	75	82	74
4	1	4	3	1	0	90	79	75	76	61
...	...	...	...	...	...	...	...	...	...	...
195	0	2	0	1	1	69	43	93	97	86
196	0	1	0	0	1	70	51	96	79	97
197	0	1	1	0	1	63	84	86	96	62
198	1	1	2	1	1	94	41	76	90	77
199	1	0	0	1	1	48	66	42	80	54

200 rows × 10 columns

Next steps: Generate code with x View recommended plots New interactive sheet

```
dt.corr()
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attenda
Gender	1.000000	0.117279	-0.110567	0.080582	-0.030038	0.040714	0.101827	-0.023514	0.042
Race	0.117279	1.000000	-0.146536	-0.078755	0.027577	0.020532	0.093640	-0.020866	-0.081
Parental_Education	-0.110567	-0.146536	1.000000	0.015592	-0.017990	-0.027745	-0.030225	0.108471	-0.073
Lunch	0.080582	-0.078755	0.015592	1.000000	-0.084717	-0.039232	-0.133768	0.142627	-0.031
Test_Preparation	-0.030038	0.027577	-0.017990	-0.084717	1.000000	0.041081	0.015322	-0.080412	0.009
Math_Score	0.040714	0.020532	-0.027745	-0.039232	0.041081	1.000000	-0.015757	-0.008866	-0.068
Reading_Score	0.101827	0.093640	-0.030225	-0.133768	0.015322	-0.015757	1.000000	0.113538	-0.018
Writing_Score	-0.023514	-0.020866	0.108471	0.142627	-0.080412	-0.008866	0.113538	1.000000	-0.131
Attendance	0.042844	-0.081026	-0.073682	-0.031017	0.009499	-0.068413	-0.018156	-0.131880	1.000
Previous_Grades	-0.035418	-0.063395	0.053023	0.048215	0.001119	-0.079608	-0.108503	0.061716	-0.012
Performance	-0.010005	0.013333	-0.053742	0.006449	-0.058600	-0.370456	-0.510493	-0.588647	0.147

```
xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size=0.50)
```

```
xtest
```

	Gender	Race	Parental_Education	Lunch	Test_Preparation	Math_Score	Reading_Score	Writing_Score	Attendance	Previous_Grades
153	1	2	1	0	1	43	91	90	76	67
102	0	1	1	0	1	74	80	63	99	67
183	1	3	2	0	0	96	86	89	84	54
64	1	0	2	1	0	58	52	62	93	94
97	0	0	0	1	0	65	61	98	97	94
...	...	...	...	...	...	...	...	...	...	...
134	1	1	1	1	0	90	56	69	76	92
27	0	0	2	1	1	59	47	72	93	99
96	0	1	3	0	0	56	79	82	86	52
66	0	1	0	1	1	44	41	40	88	93
108	0	3	2	1	0	53	60	63	99	64

100 rows × 10 columns

Next steps: Generate code with `xtest` View recommended plots New interactive sheet

```
sns.pairplot(dt, hue='Performance')
```