Homework 2 by Jeb Besecker

3/25/2025

1. Load and explore the Student Performance Dataset.

In [499…

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as mno
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.impute import SimpleImputer
```

In [500…

```
# Load and explore dataset
df = pd.read_csv("../../Homework/Data/Student Performance Dataset.csv")
df.head()
```

Out[500…

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [501…

```
# Checking null values
df.info()
```

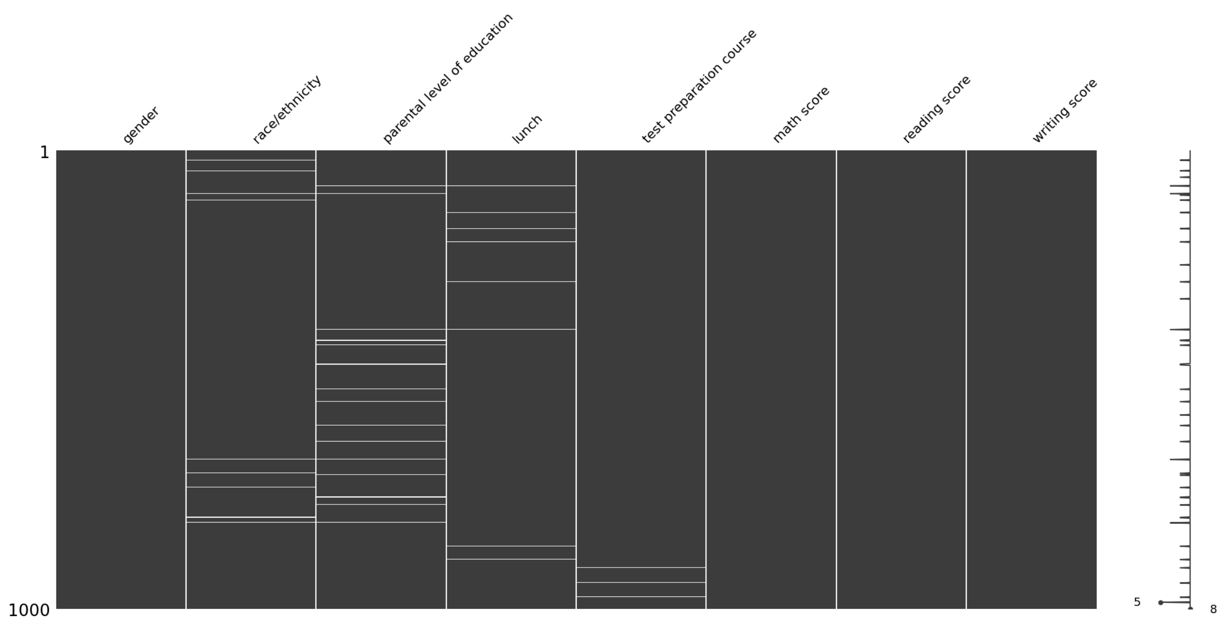
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race/ethnicity               989 non-null    object
 2   parental level of education  979 non-null    object
 3   lunch                        988 non-null    object
 4   test preparation course      996 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
```

In [502…

```python
# Explore MissingNo
mno.matrix(df)
```

Out[502…

```
<Axes: >
```



2. Create a table outlining the issues with the original data and how you treated it.

In [503…

```python
# Changelog to store changes
changelog = []
```

In [504…

```python
# Filter rows that contain any nulls
df_with_nulls = df[df.isnull().any(axis=1)]
print(df_with_nulls)
```

```
     gender race/ethnicity parental level of education         lunch  \
21   female            NaN                some college  free/reduced
44   female            NaN          associate's degree  free/reduced
58     male        group D                         NaN      standard
77     male        group A                         NaN           NaN
94   female            NaN                         NaN      standard
97   female        group E                some college           NaN
108  female            NaN          associate's degree  free/reduced
135    male        group C           bachelor's degree           NaN
170    male        group A                 high school           NaN
199  female        group B           bachelor's degree           NaN
249    male        group C                 high school           NaN
286    male        group E          associate's degree           NaN
323  female        group C            some high school           NaN
390    male        group E                         NaN           NaN
413    male        group B                         NaN      standard
414  female        group C                         NaN  free/reduced
424    male        group B                         NaN  free/reduced
465  female        group C                         NaN      standard
466  female        group D                         NaN  free/reduced
467    male        group A                         NaN  free/reduced
520    male        group D                         NaN      standard
547    male        group C                         NaN      standard
576    male        group A                         NaN      standard
599  female        group D                         NaN      standard
634    male        group D                         NaN      standard
673  female            NaN                         NaN      standard
703  female            NaN                some college      standard
707    male        group C                         NaN      standard
734  female            NaN                some college  free/reduced
755  female        group E                         NaN      standard
756    male        group D                         NaN      standard
772  female        group B                         NaN  free/reduced
799  female            NaN          associate's degree      standard
800    male            NaN            some high school      standard
811    male            NaN                         NaN  free/reduced
862    male        group D           bachelor's degree           NaN
891  female        group E          associate's degree           NaN
909    male        group E           bachelor's degree      standard
942    male        group C                 high school      standard
973  female        group D                some college  free/reduced
984  female            NaN            some high school           NaN

    test preparation course  math score  reading score  writing score
21                completed          65             75             70
44                     none          50             56             54
58                completed          58             59             58
77                completed          80             78             81
94                     none          79             86             92
97                completed          63             72             70
108                    none          52             76             70
135                    none          58             55             48
170               completed          72             73             74
199                    none          78             79             76
249                    none          68             60             53
286               completed          97             82             88
```

323	none	43	53	53
390	completed	73	67	59
413	completed	63	67	67
414	completed	51	72	79
424	none	41	39	34
465	none	84	87	91
466	none	26	31	38
467	completed	72	67	65
520	none	71	49	52
547	completed	72	67	64
576	completed	61	51	52
599	none	65	82	81
634	none	84	84	80
673	completed	65	84	84
703	none	63	64	67
707	none	66	59	52
734	none	53	58	57
755	none	84	95	92
756	none	55	58	52
772	completed	52	67	72
799	none	52	55	57
800	completed	67	73	68
811	none	45	47	49
862	completed	39	42	38
891	none	85	92	85
909	NaN	70	64	70
942	NaN	81	66	64
973	NaN	49	65	61
984	NaN	74	75	82

In [505…

```
# Categorical Variables imputation
si = SimpleImputer(strategy='most_frequent')
df[['race/ethnicity']] = si.fit_transform(df[['race/ethnicity']])
# Changelog addition explaining the imputation
changelog.append({
    'column': 'race/ethnicity',
    'change': 'Imputed missing values using mode',
    'rationale': 'Categorical variable with missing values uses mode',
})

df[['lunch']] = si.fit_transform(df[['lunch']])

# Changelog addition explaining the imputation
changelog.append({
    'column': 'lunch',
    'change': 'Imputed missing values using mode',
    'rationale': 'Categorical variable with missing values uses mode',
})
df[['test preparation course']] = si.fit_transform(df[['test preparation course']])

# Changelog addition explaining the imputation
changelog.append({
    'column': 'test preparation course',
    'change': 'Imputed missing values using mode',
    'rationale': 'Categorical variable with missing values uses mode',
})
```

```
# Parent did not attend highschool
si_str = SimpleImputer(strategy='constant', fill_value='did not attend highschool')
df[['parental level of education']] = si_str.fit_transform(df[['parental level of e

# Changelog addition explaining the imputation
changelog.append({
    'column': 'parental level of education',
    'change': 'Imputed missing values using constant: "did not attend highschool"',
    'rationale': 'Missing parent education could mean no education at all and I wan
})

# Check for missing values again
df.isnull().sum()
```

Out[505…

```
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```

In [506…

```
# Check dataframe info after imputation
df
```

Out[506…

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

In [507…

```
# Check Numerical Variables
numericalColumns = df.select_dtypes(include=['float64', 'int64'])
numericalColumns.describe()
```

Out[507…

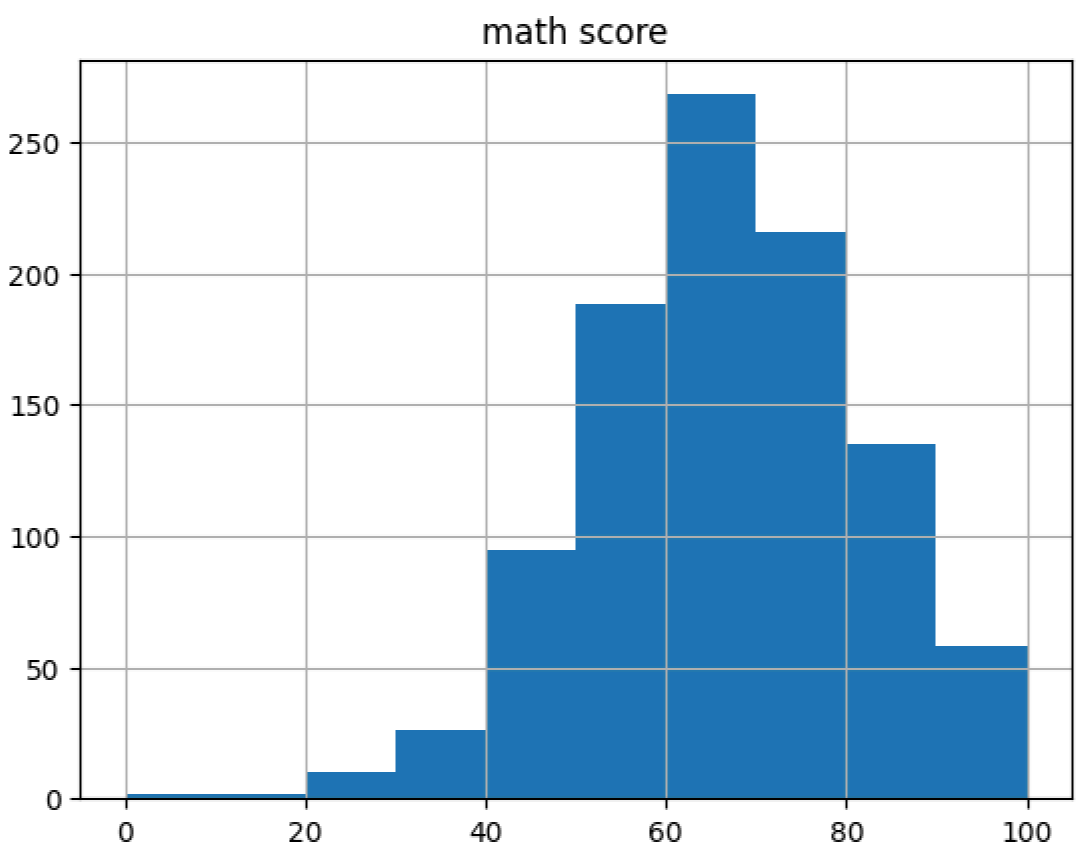
	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

In [508…

```
# Check for outliers
df.hist(column='math score')
```

Out[508…

```
array([[<Axes: title={'center': 'math score'}>]], dtype=object)
```



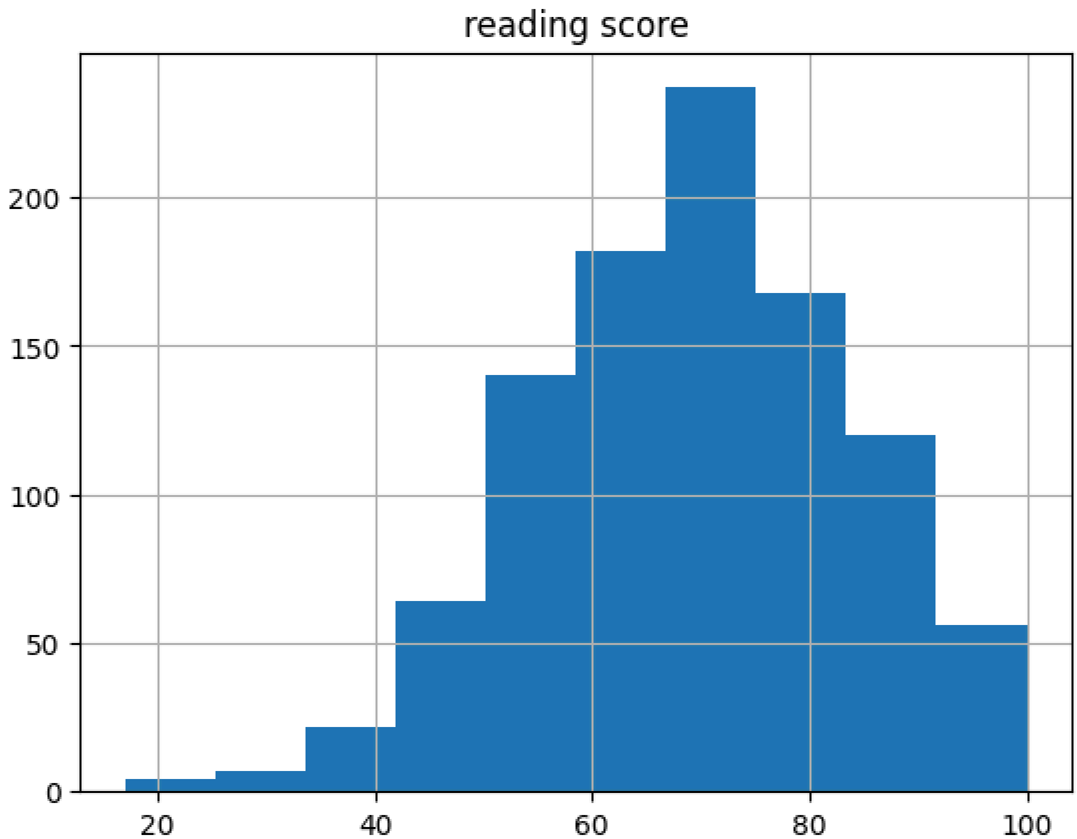
- No distinguishable outliers. A math score of 0 is possible.

In [509…

```
df.hist(column='reading score')
```

Out[509…

```
array([[<Axes: title={'center': 'reading score'}>]], dtype=object)
```



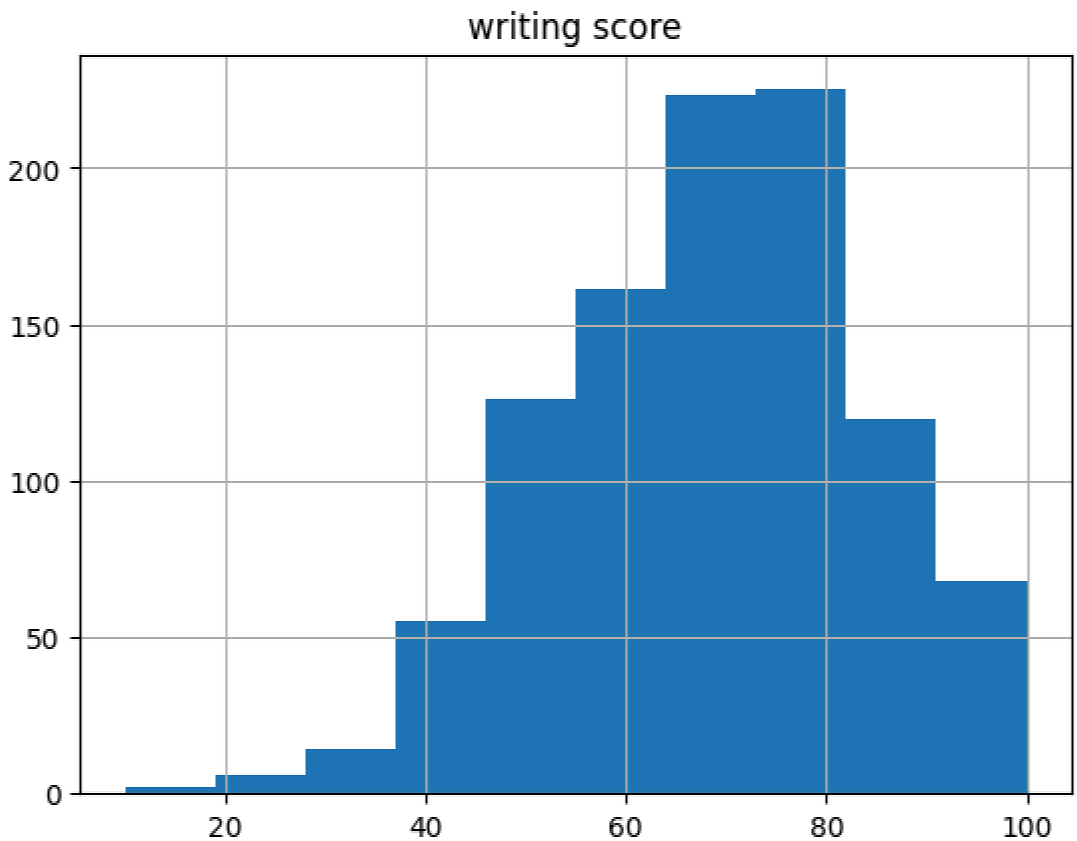
- No distinguishable outliers.

In [510…

```
df.hist(column='writing score')
```

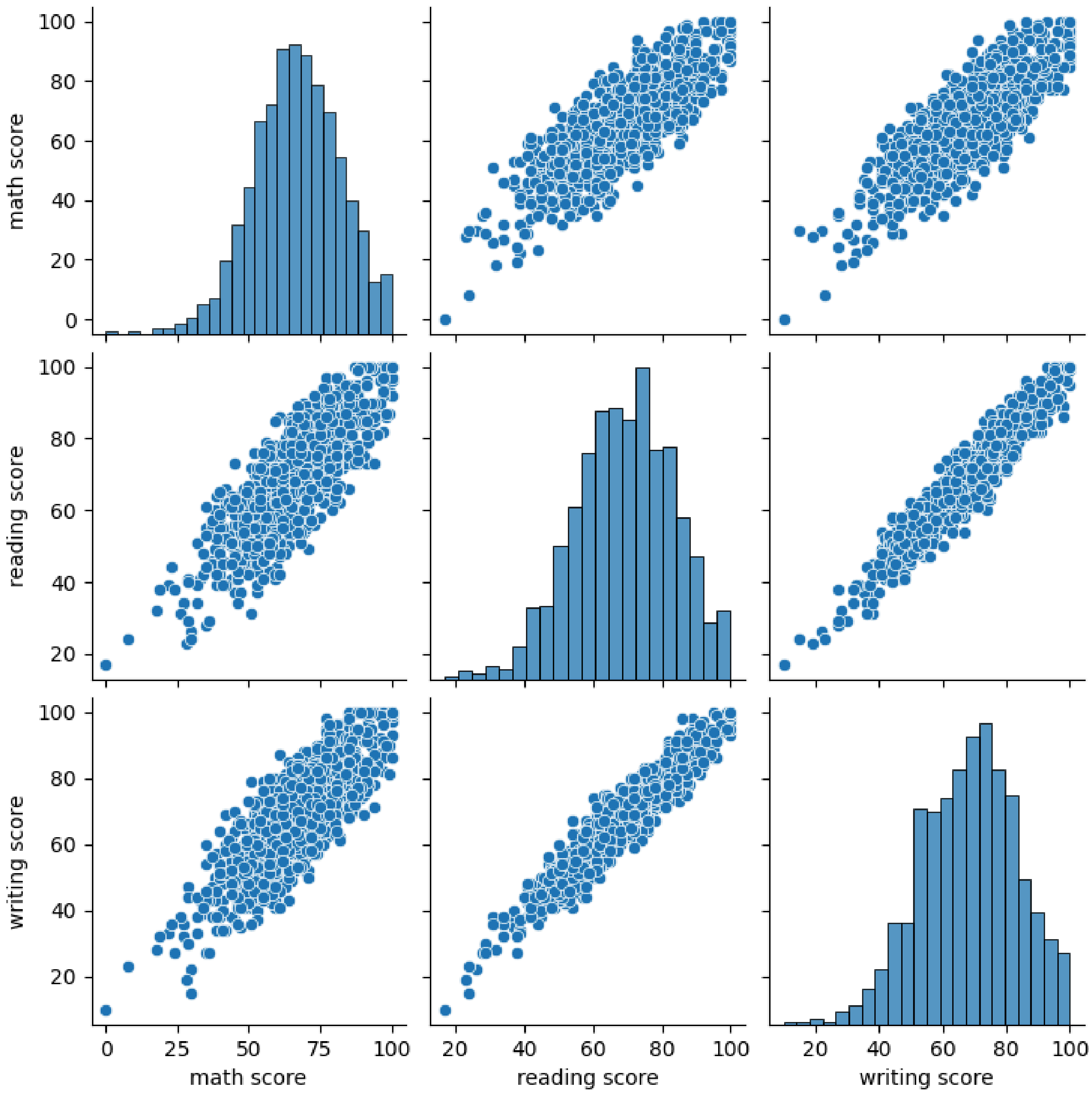
Out[510…

```
array([[<Axes: title={'center': 'writing score'}>]], dtype=object)
```



In [511…
```
sns.pairplot(df[numericalColumns.columns])
```

Out[511…
```
<seaborn.axisgrid.PairGrid at 0x1a18cbbecc0>
```



- No distinguishable outliers.

In [512…

```
# Pandas Get Dummies for categorical columns
categoricalColumns = df.select_dtypes(include=['object'])
categoricalColumns[:5]
```

Out[512...

	gender	race/ethnicity	parental level of education	lunch	test preparation course
0	female	group B	bachelor's degree	standard	none
1	female	group C	some college	standard	completed
2	female	group B	master's degree	standard	none
3	male	group A	associate's degree	free/reduced	none
4	male	group C	some college	standard	none

In [513...

```
df.columns
```

Out[513...

```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [514...

```
# Get dummies for categorical columns
df_categorical = pd.get_dummies(df, columns=categoricalColumns.columns, drop_first=
df_categorical.head()

# Changelog update
changelog.append({
    'column': 'Categorical Variables',
    'change': 'Created Dummy variables for all categorical variables',
    'rationale': 'Categorical variables were converted to dummy variables for model
})
```

In [515...

```
# Check the new dataframe with dummies
df_categorical
```

Out[515…

	math score	reading score	writing score	gender_male	race/ethnicity_group B	race/ethnicity_group C	race
0	72	72	74	False	True	False	
1	69	90	88	False	False	True	
2	90	95	93	False	True	False	
3	47	57	44	True	False	False	
4	76	78	75	True	False	True	
...	...	...	...	...	...	...	
995	88	99	95	False	False	False	
996	62	55	55	True	False	True	
997	59	71	65	False	False	True	
998	68	78	77	False	False	False	
999	77	86	86	False	False	False	

1000 rows × 16 columns

In [516…

```
# Change summary Dataframe
change_summary = pd.DataFrame(changelog)
pd.set_option('display.max_colwidth', None)
display(change_summary)
```

	column	change	rationale
0	race/ethnicity	Imputed missing values using mode	Categorical variable with missing values uses mode
1	lunch	Imputed missing values using mode	Categorical variable with missing values uses mode
2	test preparation course	Imputed missing values using mode	Categorical variable with missing values uses mode
3	parental level of education	Imputed missing values using constant: "did not attend highschool"	Missing parent education could mean no education at all and I wanted to fill it with a constant value as it could be a valid category
4	Categorical Variables	Created Dummy variables for all categorical variables	Categorical variables were converted to dummy variables for model training

3. Create two tables that provide descriptive statistics of the original data and pre-processed data. a. What differences do you notice?

In [517…

```
# Original DataFrame
orig_df = pd.read_csv("../../Homework/Data/Student Performance Dataset.csv")
```

In [518…

```
# Descriptive Statistics of Original Dataframe
orig_desc = orig_df.describe()
orig_desc
```

Out[518…

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

In [519…

```
# Descriptive Statistics of Cleaned Dataframe
desc = df_categorical.describe()
desc
```

Out[519…

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

a)

- Original Dataset: It contained clean variables that did not present any noticeable outliers. One student received a score of 0, but this is entirely plausible.
- Pre-Processed Data: After converting categorical variables into dummy variables, the number of numerical variables significantly increased, providing additional context for analysis.

- Key Differences: The descriptive statistics for scores (math, reading, writing) remain unchanged, but the addition of 13 new dummy variables enhances the available information for modeling.

```
b. Highlight the stats where the difference is significant (>20%),
moderate (10% - 20%), and
negligible (< 10%).
```

b)

- The original numerical variables (math, reading, writing scores) did not undergo any imputation or cleaning, thus, the descriptive statistics will remain the same.
- Incorporating categorical variables as dummy variables significantly impacts models, such as regression analyses. Previously, categorical data needed to be excluded or encoded manually, limiting insights. Including these variables as dummy variables provides context, potentially resulting in a better fitting model. However, to ensure model accuracy and interpretability, performing methods like stepwise regression is necessary to identify statistically significant predictors.

```
c. How might these differences impact your model performance?
```

- The addition of the categorical variables as dummy variables will drastically change a model like a regression model. As they now can be utilized in the models calculation whereas before they needed to be left out. More context could result in a better fitting model. However, we would need to perform a stepwise method to determine which variables to keep that are statistically significant.

4. Perform univariate and bivariate graphical analysis on the pre-processed. For each of the graphs generated, state your conclusion. You may want to analyze the data statistically as well to get a better picture (e.g., reviewing the descriptive stats, along with the graphical representation).

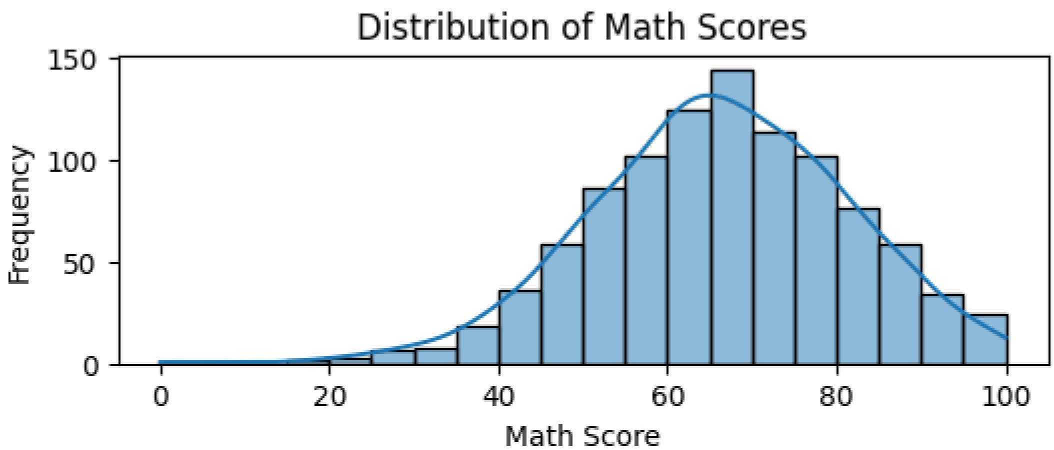
Univariate Analysis

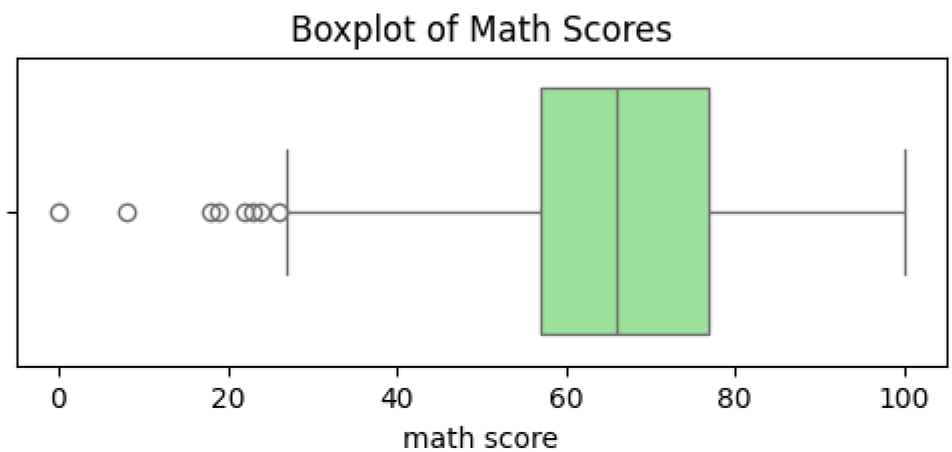
In [520…

```
# Univariate Analysis

# Math Scores
plt.figure(figsize=(6, 2))
sns.histplot(df_categorical['math score'], bins=20, kde=True)
plt.title('Distribution of Math Scores')
plt.xlabel('Math Score')
plt.ylabel('Frequency')
plt.show()
```

```
# Boxplot for math score
plt.figure(figsize=(6, 2))
sns.boxplot(x=df_categorical['math score'], color='lightgreen')
plt.title('Boxplot of Math Scores')
plt.show()
```



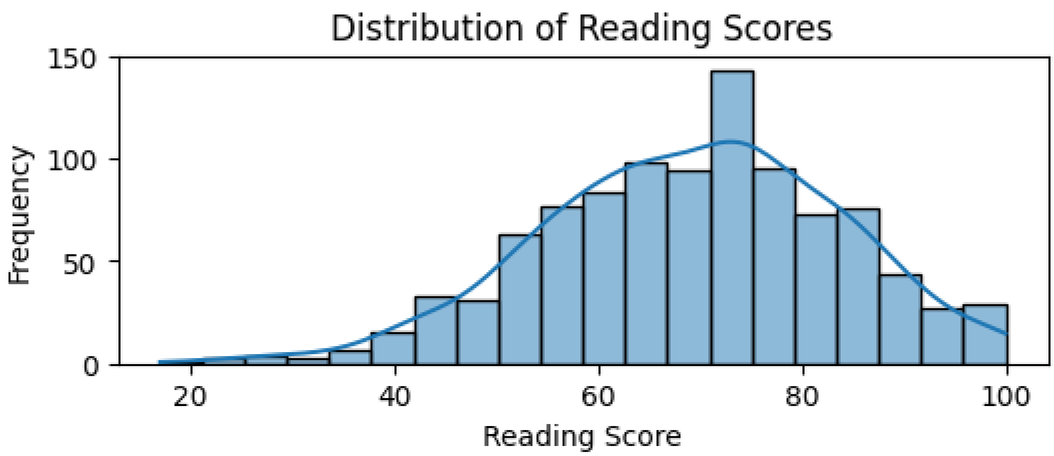


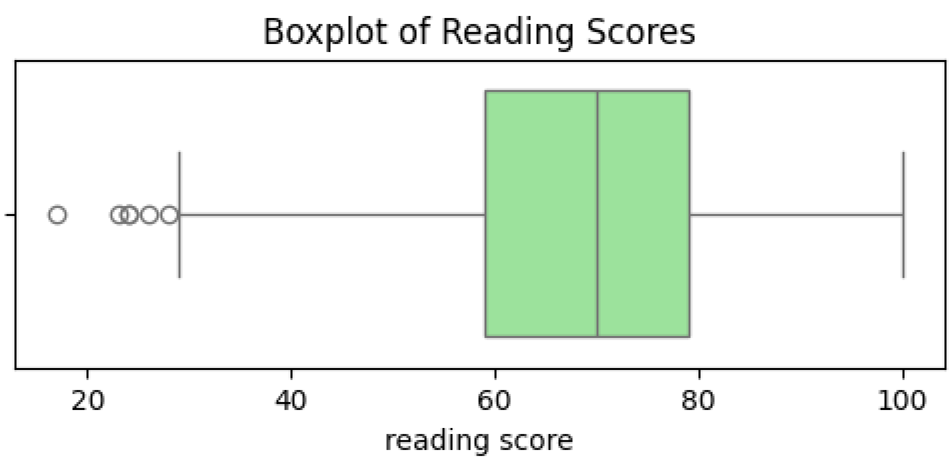
- Math scores show a roughly normal distribution with a slight left skew. The distribution peaks around scores of 65-70, representing the most common student performance. The left tail indicates several extremely low scores. Upon further investigation later seen, it appears students who scored low in math consistently scored lower across all subjects, highlighting a clear pattern of underperformance. This cluster likely shares common traits, such as not completing the test preparation course or having parents with lower education levels.
- The boxplot similarly highlights a group of lower-performing students, including a student who scored 0. The interquartile range (IQR) spans scores between 50 and 80, meaning half the students scored within this range.

In [521…
```
# Reading Scores
plt.figure(figsize=(6, 2))
sns.histplot(df_categorical['reading score'], bins=20, kde=True)
plt.title('Distribution of Reading Scores')
plt.xlabel('Reading Score')
```

```
plt.ylabel('Frequency')
plt.show()

# Boxplot for Reading score
plt.figure(figsize=(6, 2))
sns.boxplot(x=df_categorical['reading score'], color='lightgreen')
plt.title('Boxplot of Reading Scores')
plt.show()
```





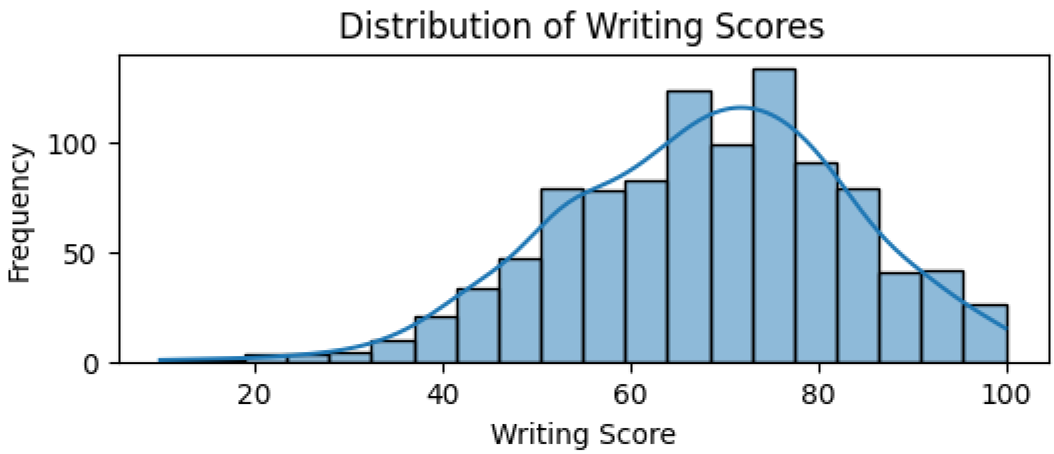
- Reading scores show a generally normal distribution with a mild left skew, peaking around scores of 70-75. A few notably low scores appear, again representing the lower performing group identified earlier.
- The IQR of 60-80 indicates that most students performed slightly better in reading than math.

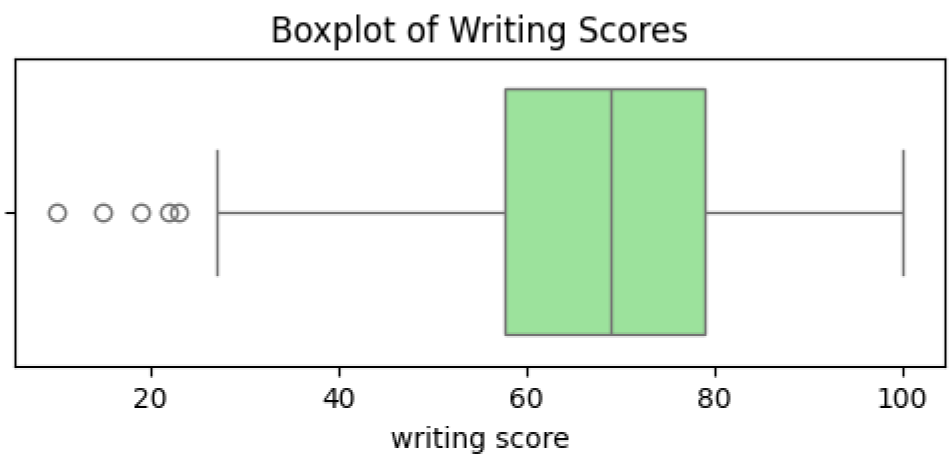
In [522…

```
# Writing Scores
plt.figure(figsize=(6, 2))
sns.histplot(df_categorical['writing score'], bins=20, kde=True)
plt.title('Distribution of Writing Scores')
plt.xlabel('Writing Score')
plt.ylabel('Frequency')
plt.show()

# Boxplot for Writing score
```

```
plt.figure(figsize=(6, 2))
sns.boxplot(x=df_categorical['writing score'], color='lightgreen')
plt.title('Boxplot of Writing Scores')
plt.show()
```



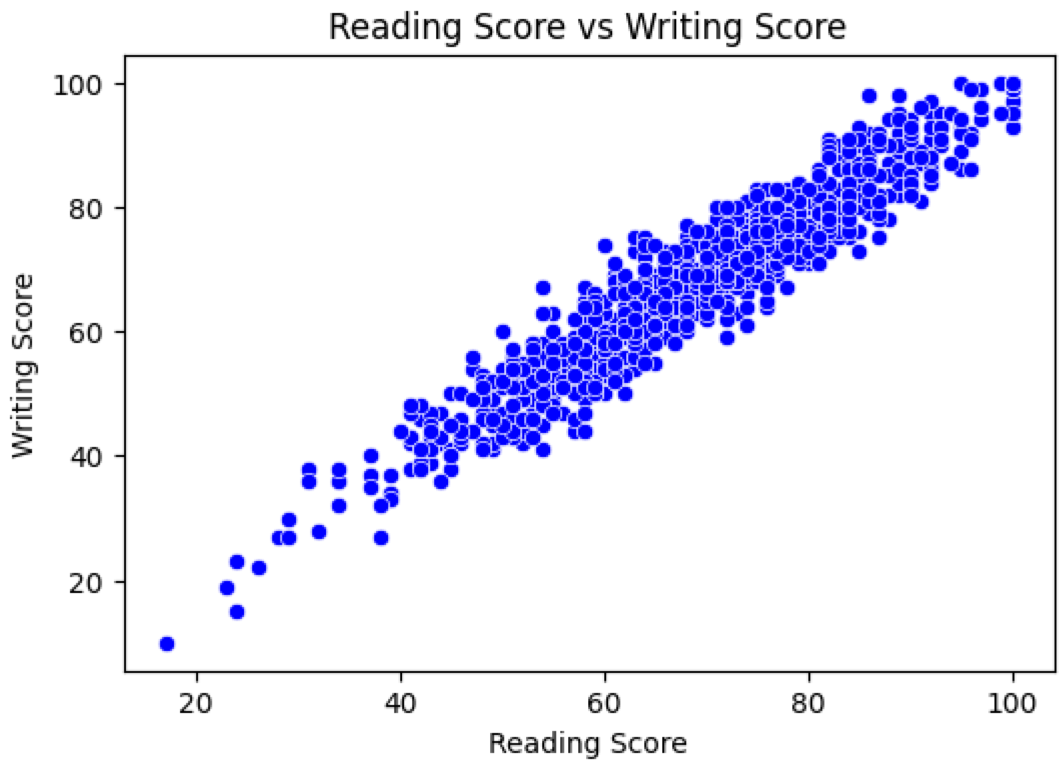


- Writing scores follow a generally normal distribution with a slight left skew, similar to reading scores. The distribution peaks at around 70-75. The pattern again highlights a consistent group of lower performing students.
- The IQR of 60-80 closely aligns with reading scores, reinforcing the relationship between reading and writing performance.

Bivariate Analysis

In [523…

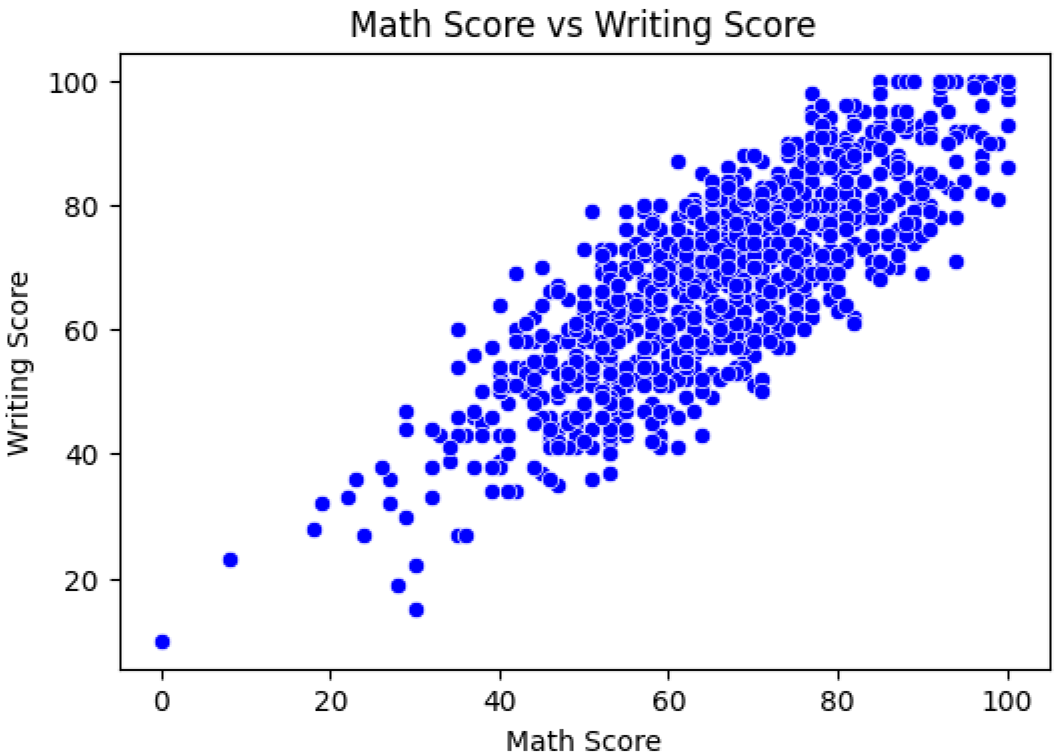
```
# Bivariate Analysis
# Scatterplot: Reading vs Writing score
plt.figure(figsize=(6, 4))
sns.scatterplot(data=df_categorical, x='reading score', y='writing score', color='b
plt.title('Reading Score vs Writing Score')
plt.xlabel('Reading Score')
plt.ylabel('Writing Score')
plt.show()
```



- A strong positive correlation exists, with students performing similarly in both areas. Students who struggled in reading also tended to struggle in writing, forming a distinct lower-performing group.

In [524…

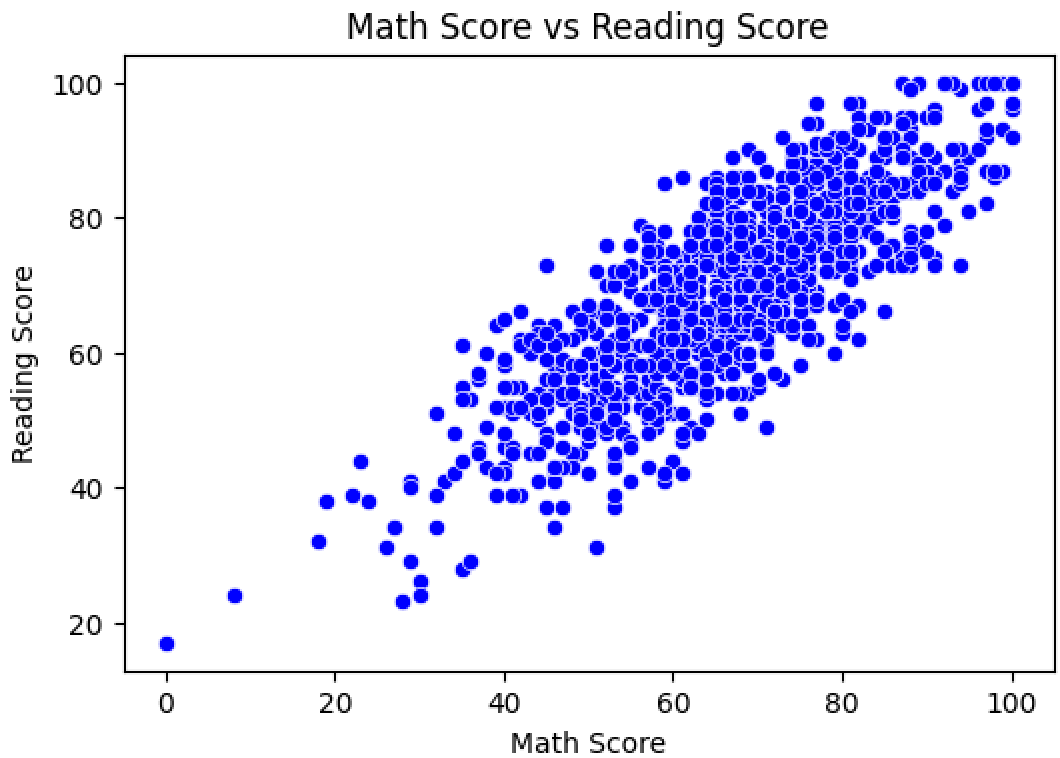
```
# Scatterplot: Math vs Writing score
plt.figure(figsize=(6, 4))
sns.scatterplot(data=df_categorical, x='math score', y='writing score', color='blue
plt.title('Math Score vs Writing Score')
plt.xlabel('Math Score')
plt.ylabel('Writing Score')
plt.show()
```



- A clear, moderate positive correlation is shown, though not as tight as between reading and writing. Higher scores in math tend to associate with higher scores in writing, suggesting shared underlying factors like general academic ability, study habits, or family support.

In [525…

```
# Scatterplot: Math vs Reading score
plt.figure(figsize=(6, 4))
sns.scatterplot(data=df_categorical, x='math score', y='reading score', color='blue
plt.title('Math Score vs Reading Score')
plt.xlabel('Math Score')
plt.ylabel('Reading Score')
plt.show()
```



- Again, a moderate positive correlation exists. Students with lower math scores consistently performed poorly in reading.
- As seen throughout all 3 graphs, the students performing negatively on the math scores usually also performed poorly on reading or writing scores. This indicates a struggling group of students in need of more assistance.