# Homework 1 - Data Preparation Phase

Name/Znumber: Jordan Small / Z23465928  Professor : Juan Yepes  Date : 31 JAN, 2024

Dataset: STUDENT PERFORMANCE DATASET

Description: This data is related to student achievement of a Portuguese school, obtained from using school reports and questionnaires. The classification goal is to predict student performance, particularly in a post-secondary math course.

https://archive.ics.uci.edu/dataset/320/student+performance

```python
import pandas as pd #Import the pandas module
```

## 1. Load/read the csv file and display its shape

```python
math = pd.read_csv('student-mat.csv', sep=';')

#Check the shape of the object/dataframe

math.shape
```

```
(395, 33)
```

## 2. Print the header and the last few rows

```python
#Print the header

math.head(n=5)
```

```
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob
Fjob  ...  \
0     GP   F   18       U     GT3       A     4     4  at_home
teacher  ...
1     GP   F   17       U     GT3       T     1     1  at_home
other  ...
2     GP   F   15       U     LE3       T     1     1  at_home
other  ...
3     GP   F   15       U     GT3       T     4     2   health
services  ...
4     GP   F   16       U     GT3       T     3     3    other
other  ...

  famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0      4        3      4     1     1      3        6   5   6   6
1      5        3      3     1     1      3        4   5   5   6
2      4        3      2     2     3      3       10   7   8  10
3      3        2      2     1     1      5        2  15  14  15
4      4        3      2     1     2      5        4   6  10  10
```

```
[5 rows x 33 columns]

#Print the last few rows

math.tail(n=5)

    school sex  age address famsize Pstatus  Medu  Fedu      Mjob
Fjob  \
390     MS   M   20        U     LE3         A     2     2  services
services
391     MS   M   17        U     LE3         T     3     1  services
services
392     MS   M   21        R     GT3         T     1     1     other
other
393     MS   M   18        R     LE3         T     3     2  services
other
394     MS   M   19        U     LE3         T     1     1     other
at_home

     ... famrel freetime  goout  Dalc  Walc health absences  G1  G2
G3
390  ...      5        5      4     4     5      4       11   9   9
9
391  ...      2        4      5     3     4      2        3  14  16
16
392  ...      5        5      3     3     3      3        3  10   8
7
393  ...      4        4      1     3     4      5        0  11  12
10
394  ...      3        2      3     3     3      5        5   8   9
9

[5 rows x 33 columns]
```

## 3. Print a summary of the dataset's statistical details

```
math.describe()

              age        Medu        Fedu  traveltime   studytime
failures  \
count  395.000000  395.000000  395.000000  395.000000  395.000000
395.000000
mean    16.696203    2.749367    2.521519    1.448101    2.035443
0.334177
std      1.276043    1.094735    1.088201    0.697505    0.839240
0.743651
min     15.000000    0.000000    0.000000    1.000000    1.000000
0.000000
25%     16.000000    2.000000    2.000000    1.000000    1.000000
```

```
0.000000
50%     17.000000    3.000000    2.000000    1.000000    2.000000
0.000000
75%     18.000000    4.000000    3.000000    2.000000    2.000000
0.000000
max     22.000000    4.000000    4.000000    4.000000    4.000000
3.000000

           famrel    freetime       goout        Dalc        Walc
health  \
count  395.000000  395.000000  395.000000  395.000000  395.000000
395.000000
mean     3.944304    3.235443    3.108861    1.481013    2.291139
3.554430
std      0.896659    0.998862    1.113278    0.890741    1.287897
1.390303
min      1.000000    1.000000    1.000000    1.000000    1.000000
1.000000
25%      4.000000    3.000000    2.000000    1.000000    1.000000
3.000000
50%      4.000000    3.000000    3.000000    1.000000    2.000000
4.000000
75%      5.000000    4.000000    4.000000    2.000000    3.000000
5.000000
max      5.000000    5.000000    5.000000    5.000000    5.000000
5.000000

         absences          G1          G2          G3
count  395.000000  395.000000  395.000000  395.000000
mean     5.708861   10.908861   10.713924   10.415190
std      8.003096    3.319195    3.761505    4.581443
min      0.000000    3.000000    0.000000    0.000000
25%      0.000000    8.000000    9.000000    8.000000
50%      4.000000   11.000000   11.000000   11.000000
75%      8.000000   13.000000   13.000000   14.000000
max     75.000000   19.000000   19.000000   20.000000
```

## 4. Display a concise summary of the dataframe

```
math.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64 
 7   Fedu        395 non-null    int64 
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64 
 13  studytime   395 non-null    int64 
 14  failures    395 non-null    int64 
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64 
 24  freetime    395 non-null    int64 
 25  goout       395 non-null    int64 
 26  Dalc        395 non-null    int64 
 27  Walc        395 non-null    int64 
 28  health      395 non-null    int64 
 29  absences    395 non-null    int64 
 30  G1          395 non-null    int64 
 31  G2          395 non-null    int64 
 32  G3          395 non-null    int64 
dtypes: int64(16), object(17)
memory usage: 102.0+ KB
```

## 5. Add an index column and display its new shape

```
math['index'] = pd.Series(range(0,395))
math.shape

(395, 34)
```

## 6. Choose a numerical field and display its unique values

```
print(math['absences'].unique())

[ 6  4 10  2  0 16 14  7  8 25 12 54 18 26 20 56 24 28  5 13 15 22  3
21
  1 75 30 19  9 11 38 40 23 17]
```

## 7. Replace an extreme set of values in the dataset with zero, -1, or np.NaN

```
#After searching through all numerical data columns, no extreme
outliers
#were found; instead, I replaced the absence value of 0, as seen
below.

import numpy as np
np.NaN #Not a number
```

```
nan
```

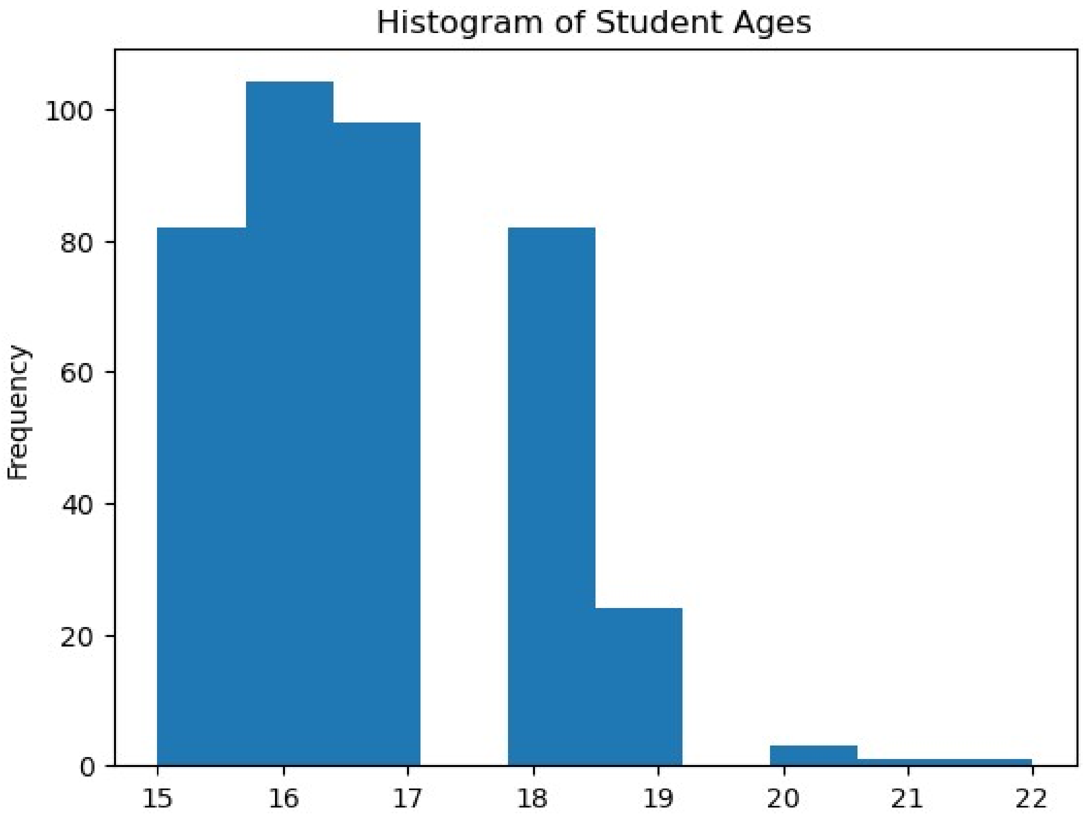
```
#Replace all absences = 0 for np.NaN

math['absences'] = math['absences'].replace({0: np.NaN})

print(math['absences'].unique())
```

```
[ 6.  4. 10.  2. nan 16. 14.  7.  8. 25. 12. 54. 18. 26. 20. 56. 24.
28.
  5. 13. 15. 22.  3. 21.  1. 75. 30. 19.  9. 11. 38. 40. 23. 17.]
```

## 8. Plot a histogram of a numerical value

```
math['age'].plot(kind = 'hist', title = 'Histogram of Student Ages')
```

```
<Axes: title={'center': 'Histogram of Student Ages'},
ylabel='Frequency'>
```



9. Choose a categorical field and represent it as a numerical field (create a new column)

```
math["Pstatus_numeric"] = math['Pstatus'] #Replicate the Pstatus
variable

print(math['Pstatus'].unique()) #Print/inspect unique values for
Pstatus

['A' 'T']

#Create a dictionary with the numeric value equivalencies

dict_Pstatus = {"Pstatus_numeric": {'A':0,'T':1}}
dict_Pstatus

{'Pstatus_numeric': {'A': 0, 'T': 1}}

#Make the replacement and display the last few rows

math.replace(dict_Pstatus, inplace=True)

math.tail(n=5)

    school sex  age address famsize Pstatus  Medu  Fedu      Mjob
Fjob  \
390      MS    M    20         U      LE3         A      2      2  services
services
391      MS    M    17         U      LE3         T      3      1  services
services
392      MS    M    21         R      GT3         T      1      1     other
other
393      MS    M    18         R      LE3         T      3      2  services
other
394      MS    M    19         U      LE3         T      1      1     other
at_home

     ... Dalc Walc  health  absences  G1  G2  G3 index Pstatus_numeric
\
390  ...    4    5       4        11   9   9   9   390               0

391  ...    3    4       2         3  14  16  16   391               1

392  ...    3    3       3         3  10   8   7   392               1

393  ...    3    4       5         0  11  12  10   393               1

394  ...    3    3       5         5   8   9   9   394               1


        age_z
390  2.592380
391  0.238380
392  3.377047
393  1.023046
394  1.807713

[5 rows x 36 columns]
```

## 10. Use the zscore function to standardize a numerical field (create a new column)

$z = \text{Standarized Value} = \frac{x - \bar{x}}{s} = \frac{\text{Data value - mean}}{\text{Standard deviation}}$

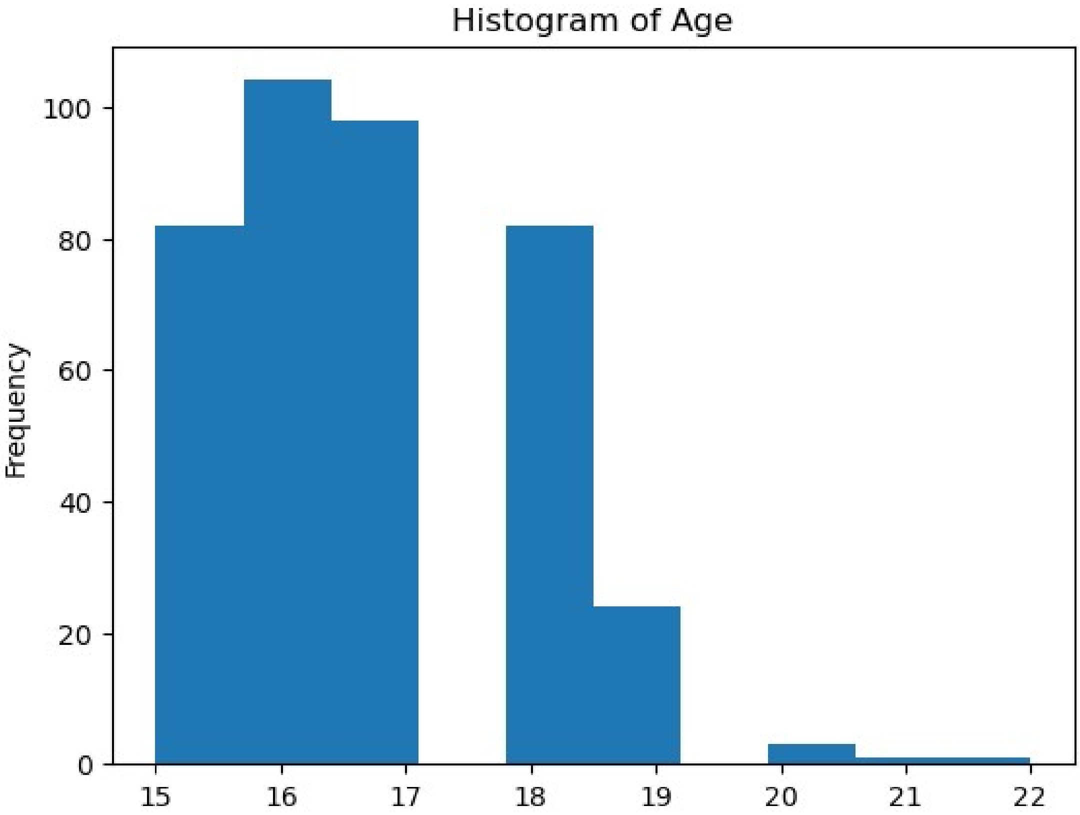
```python
#Import the scipy module

from scipy import stats

#Number of standard deviations above the mean

math["age_z"] = stats.zscore(math['age'])
print(math['age_z'].head(n=20))
```

```
0     1.023046
1     0.238380
```

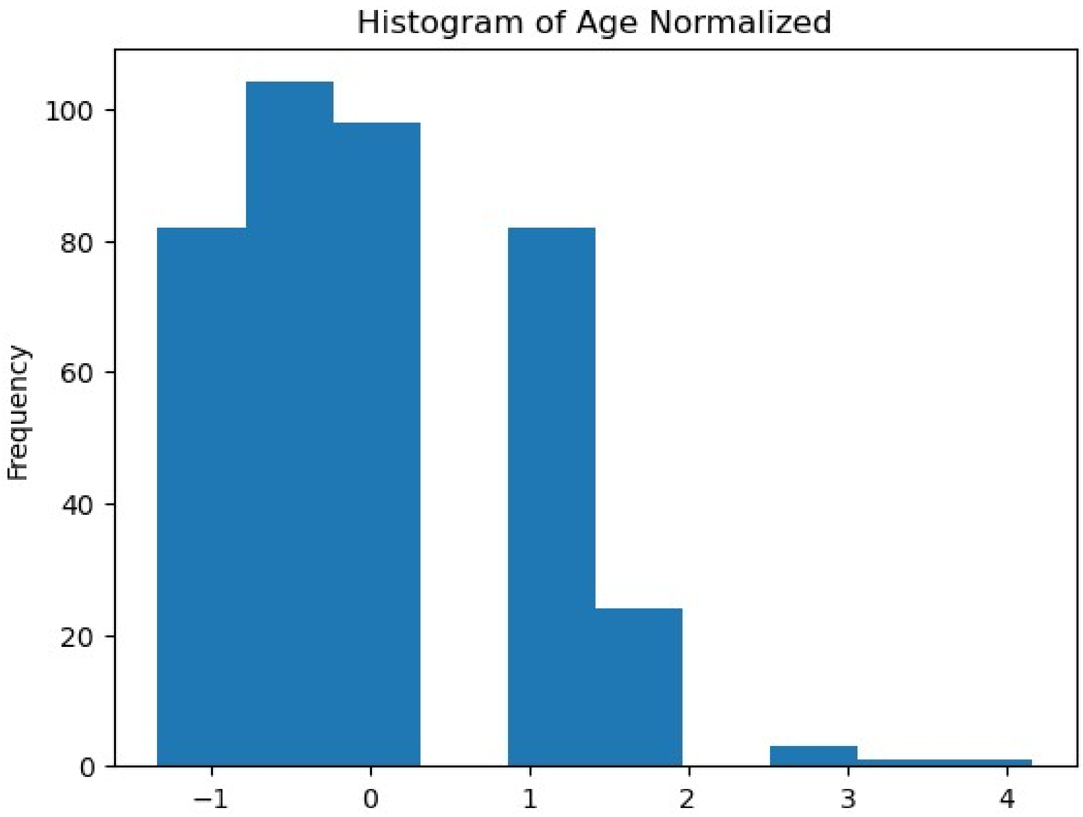
```
2    -1.330954
3    -1.330954
4    -0.546287
5    -0.546287
6    -0.546287
7     0.238380
8    -1.330954
9    -1.330954
10   -1.330954
11   -1.330954
12   -1.330954
13   -1.330954
14   -1.330954
15   -0.546287
16   -0.546287
17   -0.546287
18    0.238380
19   -0.546287
Name: age_z, dtype: float64
```

```
#Age Histogram (not normalized)

math['age'].plot(kind = 'hist', title = 'Histogram of Age')
```

```
<Axes: title={'center': 'Histogram of Age'}, ylabel='Frequency'>
```



```
#Age Histogram (Normalized)

math['age_z'].plot(kind = 'hist', title = 'Histogram of Age
Normalized')

<Axes: title={'center': 'Histogram of Age Normalized'},
ylabel='Frequency'>
```



11. Identify a field and use a criterion of your choosing to filter for outliers. Create a new dataset with the outliers

```
math.query('age_z > 3 | age_z < -3') #List the outliers

    school sex  age address famsize Pstatus  Medu  Fedu      Mjob
Fjob  \
247     GP   M   22       U     GT3       T     3     1  services
services
392     MS   M   21       R     GT3       T     1     1     other
other

     ... Dalc Walc  health  absences  G1 G2 G3 index Pstatus_numeric
age_z
247  ...    5    5       1        16   6  8  8   247               1
4.161713
392  ...    3    3       3         3  10  8  7   392               1
3.377047

[2 rows x 36 columns]

#Create a new dataset with the outlier values

math_outliers = math.query('age_z > 3 | age_z < -3')
```

```
#Display the first few rows of outliers
math_outliers.head()

    school sex  age address famsize Pstatus  Medu  Fedu      Mjob
Fjob  \
247     GP   M   22       U     GT3       T     3     1  services
services
392     MS   M   21       R     GT3       T     1     1     other
other

     ... Dalc Walc  health  absences  G1 G2 G3 index Pstatus_numeric
age_z
247  ...    5    5       1        16   6  8  8   247               1
4.161713
392  ...    3    3       3         3  10  8  7   392               1
3.377047

[2 rows x 36 columns]
```

## 12. Sort the dataset and display 15 interesting fields

```
#Sort the dataset

math_sort = math.sort_values(['age_z'], ascending = False)

#Display 15 interesting fields

print(math_sort[['age', 'failures']].head(n=15))

     age  failures
247   22         3
392   21         3
390   20         2
306   20         0
376   20         2
394   19         0
310   19         1
340   19         1
336   19         1
315   19         1
314   19         2
313   19         1
127   19         3
311   19         0
307   19         1
```