**Installing Python and Running a Data Set Exploration in PyCharm EDU Program**

Ali Bruckner

CSU Global

Dr. Dwight Davis

April 19, 2020

1. **Code for Option #1: Installing Python and Running a Data Set Exploration in PyCharm EDU Program**

*# BEGIN  
  
#############################################################  
"""Python data exploration using Pandas library  
No inputs are required  
Exploration statistics will be outputted"""  
###############################################################  
  
# Import Pandas library --comment added by Ali Bruckner 4/16/2020***import** pandas **as** pd  
  
*# Get the group by objects from Pandas  
# from pandas.core.groupby import DataFrameGroupBy  
  
# Create data\_frame of array values*df = pd.DataFrame({  
 **'name'**: [**'Matt'**, **'Lisa'**, **'Richard'**, **'John'**, **'Julia'**, **'Jane'**, **'Marlon'**],  
 **'age'**: [23, 78, 22, 19, 45, 33, 20],  
 **'gender'**: [**'M'**, **'F'**, **'M'**, **'M'**, **'M'**, **'F'**, **'M'**],  
 **'state'**: [**'DC'**, **'CO'**, **'DE'**, **'VA'**, **'MD'**, **'DE'**, **'NY'**],  
 **'years\_of\_service'**: [10, 0, 2, 0, 2, 1, 5],  
 **'iq'**: [300, 100, 110, 200, 300, 10, 40]  
})  
  
*# Display DataFrame results including empty line after result  
# Empty lines will be included with each result set going forward  
# This will be useful in validating calculated results generated below*print(**'DataFrame Table'**)  
print(df)  
print()  
  
*# Generate 25% sample of data*rows = df.sample(frac=.25)

*# Validate the sample result is 25% of the number of records in DF  
# Result is rounded to nearest whole number  
# Shows the actual result and rounded result in order to the sample generated  
# Outputs the results of calculating 25% of the DF and sample results if they match  
# Error message displayed for non matches"""*print(**'Validate sample result'**)  
**if** round(.25 \* (len(df))) == len(rows):  
 print(**'DF = '**, .25 \* len(df), **' DF(rounded) = '**, round(.25 \* (len(df))))  
 print(**'Sample result = '**, len(rows))  
 print(**'Excellent! It matches!'**)  
 print(  
 )  
**else**:  
 print(**'Sample does not equate to 25%'**)  
  
*# Display the resulting samples and details*print(**'sample of 25%'**)  
print(rows)  
print()  
  
*# Display the average IQ by gender  
# Utilizes the group by objects within Pandas to group by gender*groupby\_gender = df.groupby(**'gender'**)  
**for** gender, value **in** groupby\_gender[**'iq'**]:  
 print((gender, value.mean()))  
 print()  
  
*# Display the Summation of all ages in the data*SumofAge = df[**'age'**].sum()  
print(**'Sum of Ages'**, SumofAge)  
print()  
  
*# Display the mean of all ages in the data rounded to 2 decimal points*MeanAge = round(df[**'age'**].mean(), 2)  
print(**'Average Ages'**, MeanAge)  
print()  
  
*# Display the mean of all float type columns (age, years of service, IQ)*print(**'Means of each column'**)  
print(round(df.mean(axis=0), 2))  
print()

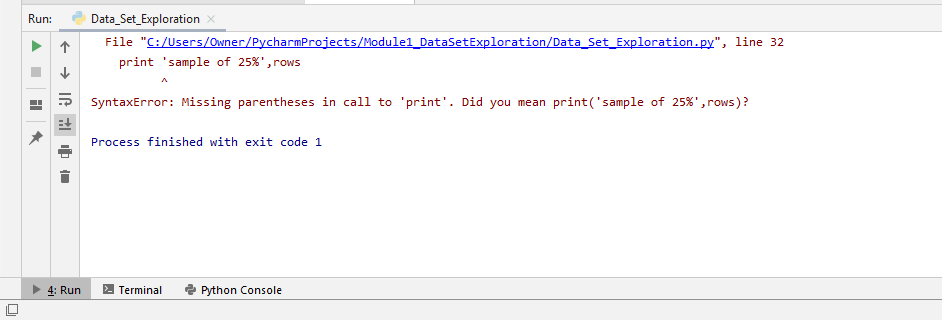
*# Display the data descriptions for all float type columns (age, years of service, IQ)  
# Includes: count, mean, standard deviation(std), minimum value(min),  
# data distribution (25th, 50th & 75th percentiles) and maximum value (max)  
  
# Data description for age*print(**'Data Description - Age'**)  
print(df[**'age'**].describe())  
print()  
  
*# Data description for years of service*print(**'Data Description - Years of Service'**)  
print(df[**'years\_of\_service'**].describe())  
print()  
  
*# Data description for IQ*print(**'Data Description - IQ'**)  
print(df[**'iq'**].describe())  
print()  
  
*# END*

1. **Code Execution**

The errors in Figures 1 through 4 were generated running the original code prior to any changes and updates.

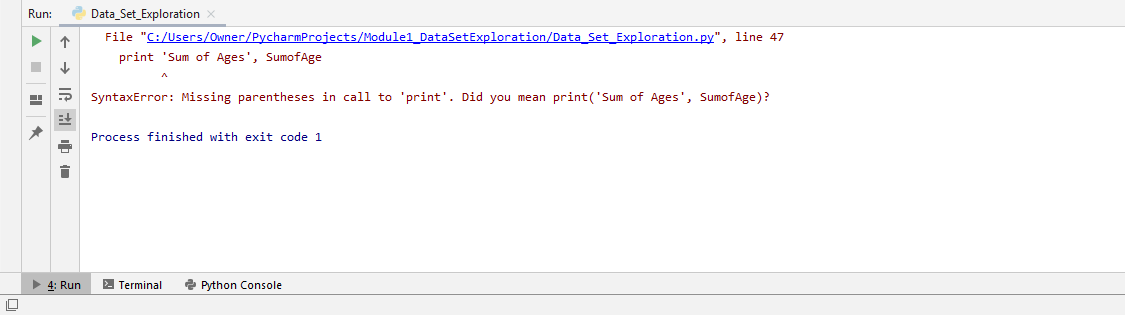
**Figure 1**

*Missing parentheses in call to ‘print’; Line 32*



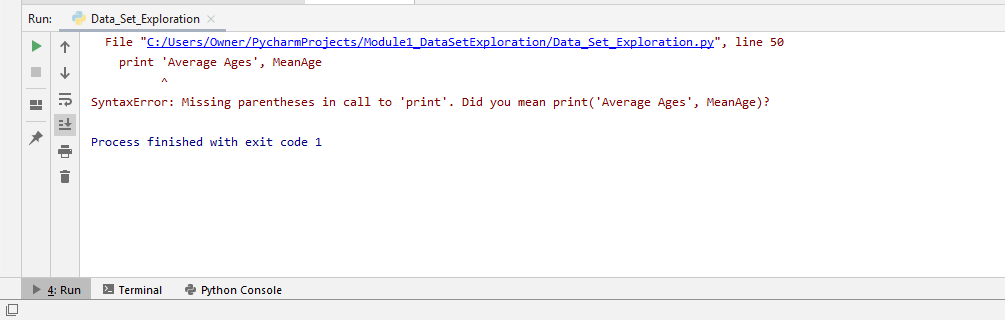
**Figure 2**

*Missing parentheses in call to ‘print’; Line 47*



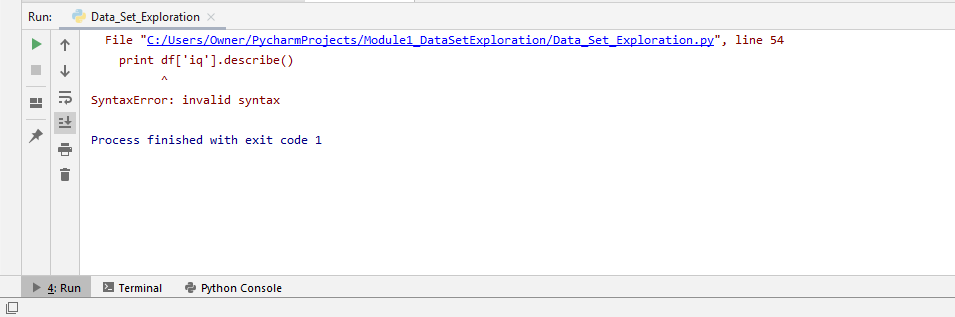
**Figure 3**

*Missing parentheses in call to ‘print’; Line 50*



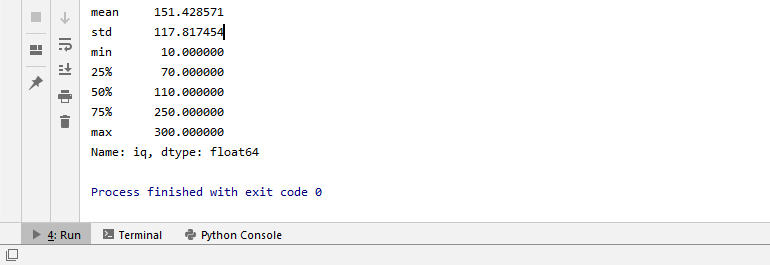
**Figure 4**

*Missing parentheses in call to ‘print’; Line 54*



**Figure 5**

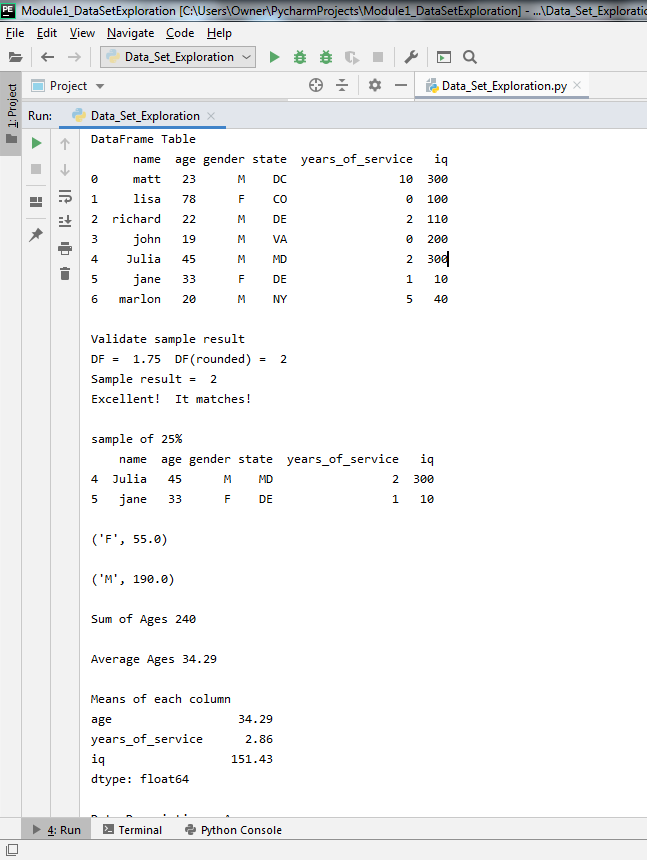
*Successful Execution*



1. **Output of the Program**

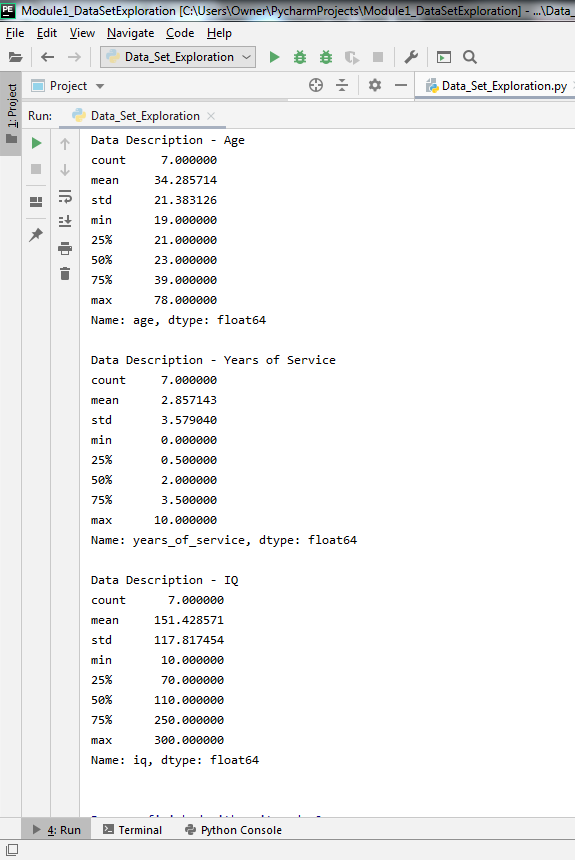
**Figure 6**

*Output of the program – View #1*



**Figure 7**

*Output of the program – View #2*



**Figure 8**

*Output of the program – View #3*

