# Title: Descriptive analysis of the effects of Cultural cognition and racial background on the performance of a student

Samrawit Kebreab Tekie

4/26/2021

## Introduction

The dataset that is being used for this project depicts a sample taken from a student population of a university. It represents a section of freshman students, and the data was collected for a project that envisioned to assist in identifying the need for an early intervention for students after their first exam. The office of enrollment course registration data was used to collect the information.However, refining manipulation had been done by the analyst of this project before using it for this project. The prior manipulation of the data is not included in this project report.

The data set have different information and fields with major categories including demographic, academic plan, course workload, and academic history based on SAT scores. However, for this project the targeted fields are “race” (as a proxy for background, economic status, and cultural cognition),“Diagnostic Exam”, “Exam One”, “Module 1”, “Academic Plan”, and “total course load”. Here in this report the main focus of the study is to investigate the effects of cultural cognition and racial background on a students performance. We try to evaluate different relationships and try identify if there is a reasonable pattern that can describe the relationships between these social factors and academic achievement. To that end, this project uses exam results from one of the core courses in STEM field. Diagnostic Exam: a test given to students on the first day of class to check and caliber their preparation and prior background for the course material.

*Module 1:* a test administered after the first unit of the course material. Exam One: a test administered after three units of the course material.

*Hypothesis:* Null: The background and cultural cognition of a student affect student’s preparedness for college and success in college. Alternative: A student’s preparedness for college and success in college are not related to background and cultural cognition.

### Load all the necessary packages

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import seaborn as sns
import scipy.stats as stats
import matplotlib.pyplot as plt
import warnings
from statsmodels.formula.api import ols
%matplotlib inline
```

## Data Processing

The BIO101_project was uploaded to the dataset repository and got loaded into the jupyter note book using the *pd.read_excel()*

Project=pd.read_excel("../BIO539/BIO101_project.xlsx")

## Data Cleaning

As mentioned above the author had done a pre-clean up of the data in the excel before importing for this project. During that clean up the data was made unanimous and unnecessary columns were dropped. No further data clean using python was necessary to do the work needed for this project except changing long field titles for columns (eg. "Exam One" to "Exam1" and dropping NA values in "Exam1", "DigEx", and "Mod1" fields as renamed later (see below).

## Dropping Null values

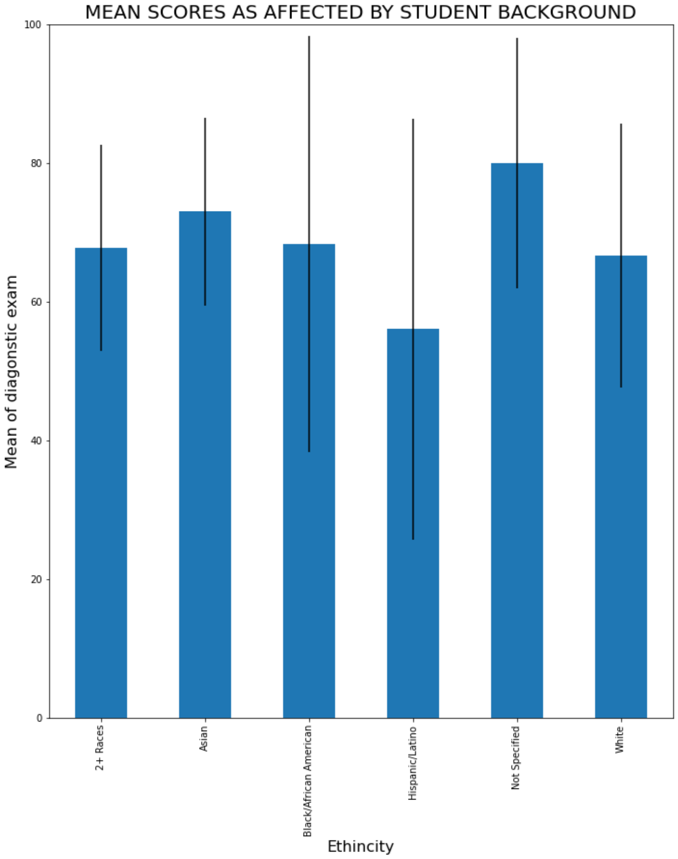
In order to make the analysis go smoothly null values were removed from the data set particularly from columns 'Exam1' and 'DigEx' using *pandas dropna* method.

## Effects of background on student performance

To investigate the relationship between students' background and their performance in the course the *groupedby* method was employed on the data 'Ethnicity' attribute of the data set. The *.agg* method was utilized on the grouped data to get statistical summaries mean and standard deviation of the column 'DigExam'.

## plot Graph

After aggregating mean and standard deviation, a bargraph drawn out of the evaluated data.This graph shows 'Effects of background on student educational progress. where x-axis represents Ethnicity and Y-axis Mean of diagnostic exam.



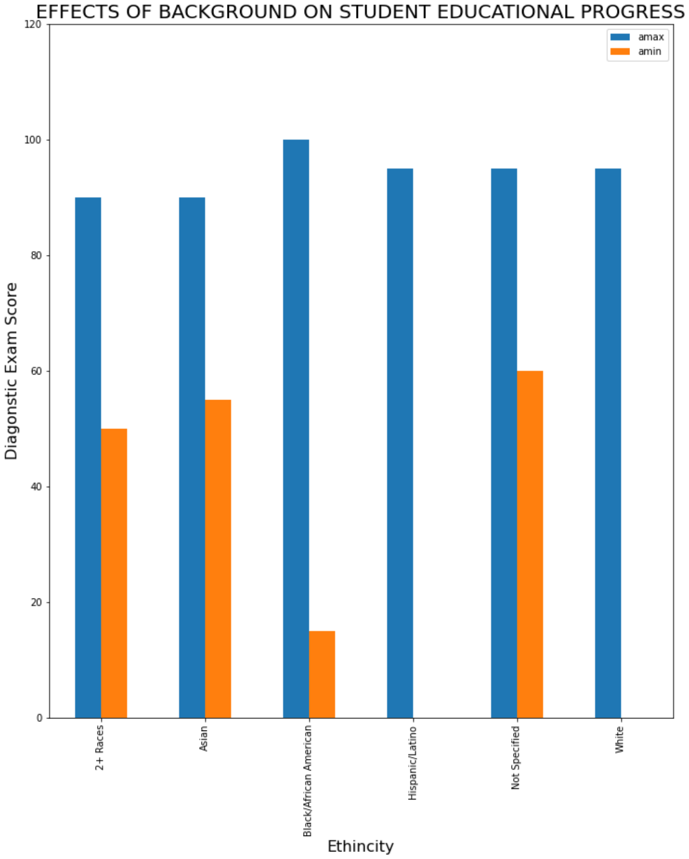
## Sorting the dataset

Here the researcher wants to view the maximum and minimum values of diagnostic exam results for each group. Therefore, first dataset sorted out grouping by Ethnicity, then connect it with the maximum and minimum values.

```
DigExMaxMin=Project.groupby('ETHNICITY').agg([np.max, np.min])['DigEx']
```

## Plot Graph

A bar graph represented to show the result of the evaluated data on the ‘Effects of background on student educational progress grouped by Ethnicity in connection with Maximum and Minimum value of diagnostic exam result’.



## A linear regression analysis

Setting a linear regression analysis to describe the results of exam one through the effects of Diagnostic exam, gender and first generation. Here a formula approach from the stats models is used, which uses C() to indicate a categorical variables.

- The formula assigns a dummy value to a categorical variable and uses it in the regression. This code was adopted from https://stackoverflow.com/questions/50733014/linear-regression-with-dummy-categorical-variables and modified to fit this project.

```
lm = ols(‘DigEx ~ C(GENDER) + C(FIRST_GEN)’, data=Project)
lm1= lm.fit()
lm1.summary()
```

### Printing the p-values of the above regression analysis

print(lm1.pvalues) output:

```
Intercept 6.833949e-38
C(GENDER)[T.M] 9.077231e-01
C(FIRST_GEN)[T.Y] 7.860944e-01
dtype: float64
```

## A liner regression analysis

Setting a linear regression analysis to describe the results of exam one through the effects of Diagnostic exam, gender and first generation. Here a formula approach from the stats models is used, which uses C() to indicate a categorical variables. This code was adopted from https://stackoverflow.com/questions/50733014/linear-regression-with-dummy-categorical-variables and modified to fit this project.

```
lm2 = ols(‘Exam1 ~ Mod1 + C(GENDER)+ C(ETHNICITY) + C(FIRST_GEN)’, data=Project)
lm3= lm2.fit()
lm3.summary()
```

### Printing p-values of the regression model

```
print(lm3.pvalues)
```

```
output:
Intercept 7.616668e-07
C(GENDER)[T.M] 8.217939e-01
C(ETHNICITY)[T.Asian] 4.502284e-01
C(ETHNICITY)[T.Black/African American] 5.108928e-01
C(ETHNICITY)[T.Hispanic/Latino] 4.438783e-01
C(ETHNICITY)[T.Not Specified] 1.048524e-01
C(ETHNICITY)[T.White] 9.829024e-02
C(FIRST_GEN)[T.Y] 6.126527e-01
```

```
Mod1 2.280686e-03
dtype: float64
```

**Using the residual test to measure the robustness (errors) of the prediction made using the linear regression testing residual by each predictor variable used in the regression analysis**

```
for x in (‘GENDER’, ‘ETHNICITY’, ‘FIRST_GEN’,‘Mod1’):
plt.figure(figsize=(10,2))
sns.scatterplot(Project[x], lm3.resid)
plt.title(“Residuals by %s” % x)
plt.ylim(-100, 100)
```

```
plt.axhline(np.mean(lm3.resid))
plt.show()
```

**testing residual of the predicted value**

```
plt.figure(figsize=(8,2))
sns.scatterplot(lm3.predict(), lm3.resid)
plt.title(“Residuals by Predicted Value”)
plt.ylim(-100, 100)
plt.axhline(np.mean(lm3.resid))
plt.show()
```

**printing the mean of the residual**

```
print(np.mean(lm3.resid))
warnings.simplefilter(action=‘ignore’, category=FutureWarning)
```

## Plot Graph

Plotting Exam1 against Primary program to see if there is a significant difference between students of different programs

```
sns.catplot(‘PRIMARY_PROGRAM’, ‘Exam1’, data=Project, kind=‘bar’)
plt.figure(figsize=(16,8))
plt.show()
```

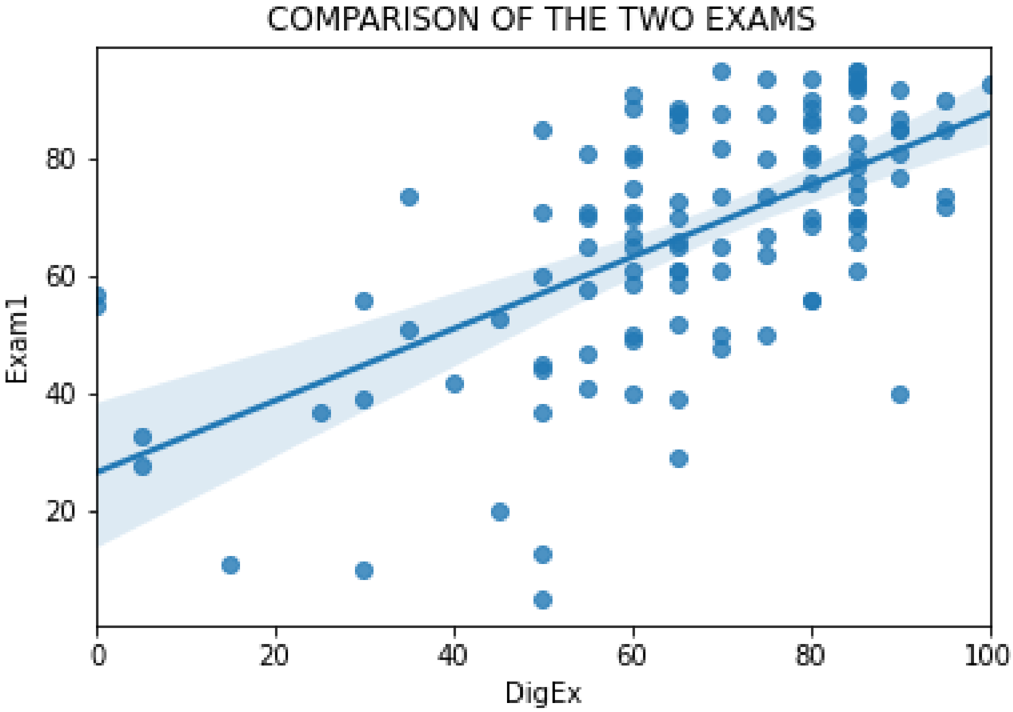
## Graph with Predictors

Adding other predictors such as ethnicity and first generation to the plot above to show how background of students can influence the Exam1 results.

```
sns.catplot(‘PRIMARY_PROGRAM’, ‘Exam1’, hue=‘ETHNICITY’, data=Project, kind=‘bar’)
plt.show
warnings.simplefilter(action=‘ignore’, category=FutureWarning)
```

## Plot Graph

```
ax = sns.regplot('DigEx', 'Exam1', data=Project).set_title('COMPARISON OF THE TWO EXAMS')
plt.show()
fig=ax.get_figure()
fig.savefig('compare.png')
warnings.simplefilter(action='ignore', category=FutureWarning)
```



## T-Test

independent t-test to test if the two fields Exam1 and DigEx results are statistically different from one another.

```
tstat, pval = stats.ttest_ind(Project.Exam1, Project.DigEx)
```

### printing the p-value of the t-test

```
print(pval)
```

## Linear regression model

linear regression model to describe Exam1 by DigEx as a predictor

```
lm4 = ols('Exam1 ~ DigEx', data=Project)
lm5= lm4.fit()
```

```
lm5.summary()
```

## printing the p-values of the regression for closer look

```
print(lm5.pvalues)
```

Output:

```
Intercept 1.406149e-06
DigEx 5.692343e-13
dtype: float64
```