A Linear Regression of Student Exam Takers

Introduction

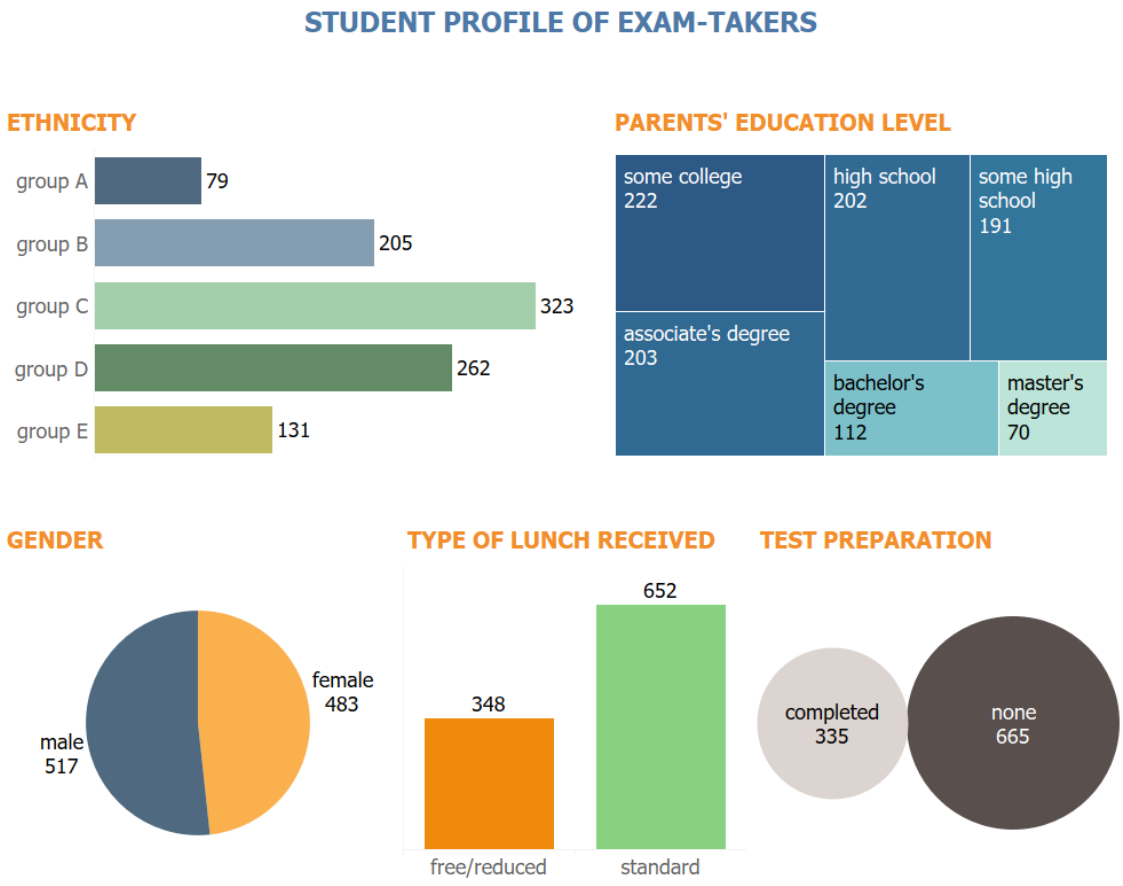
The following dataset explores 1000 students' scores in math, reading and writing, and gives information on their gender, ethnicity (given as group A-E), parental education level, type of lunch given (a standard or free/reduced lunch) and the type of test preparation completed.

In this project I will be exploring the data using Tableau, then answering 3 questions:

1. What factors contribute to receiving a high mark?
2. What is the effect of test preparation on students' marks?
3. What is the effect of gender on students' marks?

Exploratory Data Analysis

Profile of Student Exam-Takers



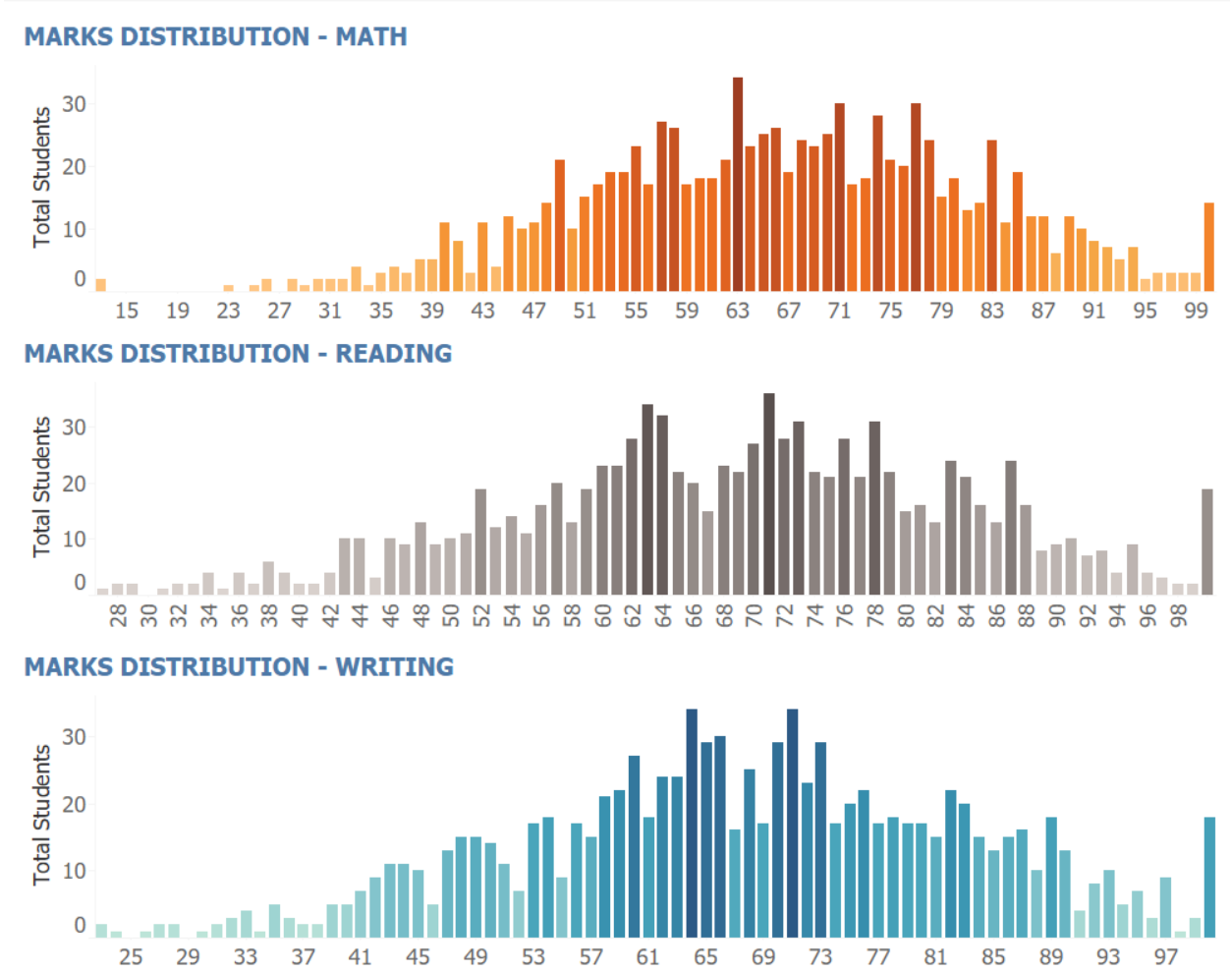
- Most students' ethnicities are group C
- A minority of students have finished higher education (Bachelor's or Master's degree)

- A slight majority of males to females (517 to 483) took the exam
- Around 35% of students had a free/reduced lunch
- Around ⅓ of students completed a test preparation course

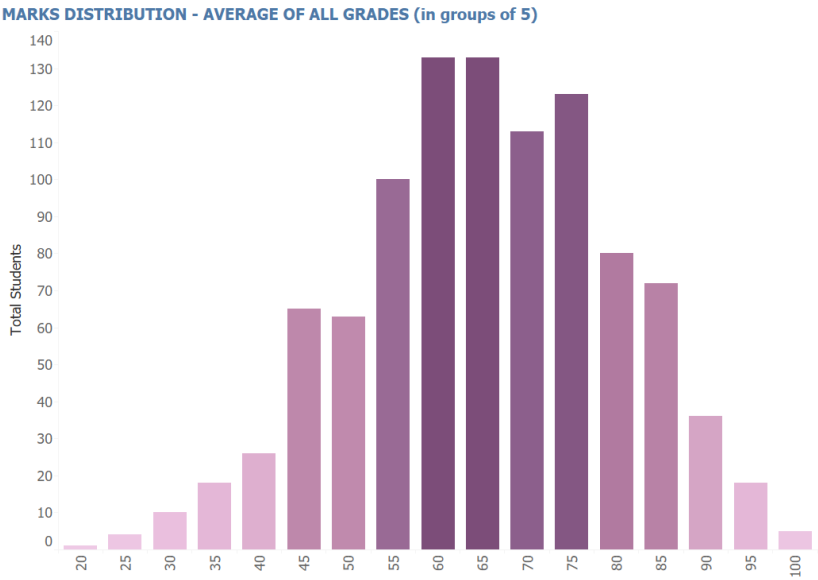
Distribution of Marks

The following histograms for the math, writing and reading exams show a distribution of grades and the number of students that received each grade. The distributions tend to follow a normal distribution. However, there are a few interesting observations:

- a surprising number of students received perfect 100 scores, which doesn't follow the normal distribution. Could it be because these students are smart, or could there be some cheating going on?
- There seems to be a dip in the number of students receiving marks between 60-70 for reading and writing. This may be just random variance, but it is interesting to notice.



The chart below shows a distribution of students' average grade, in groups of 5.



Data Modeling - Factors that influence exam scores

Libraries

The following is a list of libraries used in the model.

```
import statsmodels.api as sm
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import metrics
from sklearn.model_selection import KFold
```

Exploring the Data

We can see from the 'count' row that there are no columns with missing variables (each column has 1000 entries).

	id	math score	reading score	writing score
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	66.396000	69.002000	67.738000
std	288.819436	15.402871	14.737272	15.600985
min	1.000000	13.000000	27.000000	23.000000
25%	250.750000	56.000000	60.000000	58.000000
50%	500.500000	66.500000	70.000000	68.000000
75%	750.250000	77.000000	79.000000	79.000000
max	1000.000000	100.000000	100.000000	100.000000

Creating an "average_score" column

I created an average_score column to average out the math, writing and reading marks. This column will be used as my dependent variable.

```
# Create an "average score" column
df['average_score'] = round(((df['math score'] + df['reading score'] + df['writing score']) / 3), 1)
```

Creating Dummy Variables

As the independent variables to be observed are in string format, they were converted to dummy variables. For the boolean variables (gender, lunch, test preparation), I dropped 1 variable to reduce multicollinearity.

```
# Create dummy variables
X = pd.get_dummies(X, columns=['gender', 'lunch', 'test preparation course'])
del X['gender_male']
del X['lunch_free/reduced']
del X['test preparation course_none']

X = pd.get_dummies(X, columns=['race/ethnicity', 'parental level of education'])
```

Data Modeling - What factors contribute to receiving a high average mark?

An initial regression analysis shows an RMSE of 12.28. The variables 'parental level of education_some college' and 'parental level of education_high school' are not statistically significant (p-values > 0.05), and are removed from the model. The final model has the following variables:

```
gender_female
lunch_standard
test preparation course_completed
race/ethnicity_group A
race/ethnicity_group B
race/ethnicity_group C
race/ethnicity_group D
race/ethnicity_group E
parental level of education_associate's degree
parental level of education_bachelor's degree
parental level of education_master's degree
parental level of education_some high school
```

Using 8 k-folds, we get an average RMSE of 12.2 and an average RSQ of 0.3.

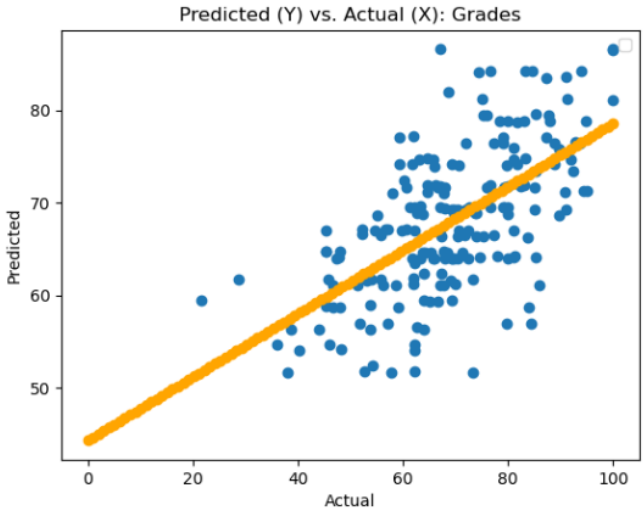
```
Scores for all folds:
*********************************************
RMSE Average :    12.277041783561902
RMSE SD:          0.7024664741761776
BIC Average :     6932.100694498401
BIC SD:           14.190268321855747
RSQ Average :     0.300406207187058
RSQ SD:           0.00754272505169838
```

Looking at the coefficients reveals some insights. According to the model:

- females expect to score around 2 points more than males on the exam
- eating a standard lunch scores around 10 points more than a free/reduced lunch
- completing a test prep course scores around 7.5 points more on the exam
- ethnicity groups D and E score more than twice as much as the other groups
- the higher the parents' education level, the higher the student's exam score. Additionally, one with parents' education level as "some high school" is expected to negatively affect one's exam score

	coef
const	46.6759
gender_female	2.3527
lunch_standard	10.1904
test preparation course_completed	7.4516
race/ethnicity_group A	6.9567
race/ethnicity_group B	5.9326
race/ethnicity_group C	6.4030
race/ethnicity_group D	12.4961
race/ethnicity_group E	14.8875
parental level of education_associate's degree	4.3144
parental level of education_bachelor's degree	5.9841
parental level of education_master's degree	9.2410
parental level of education_some high school	-4.1714

The scatter plot below shows how the predicted vs actual exam scores fit along the best fit line.

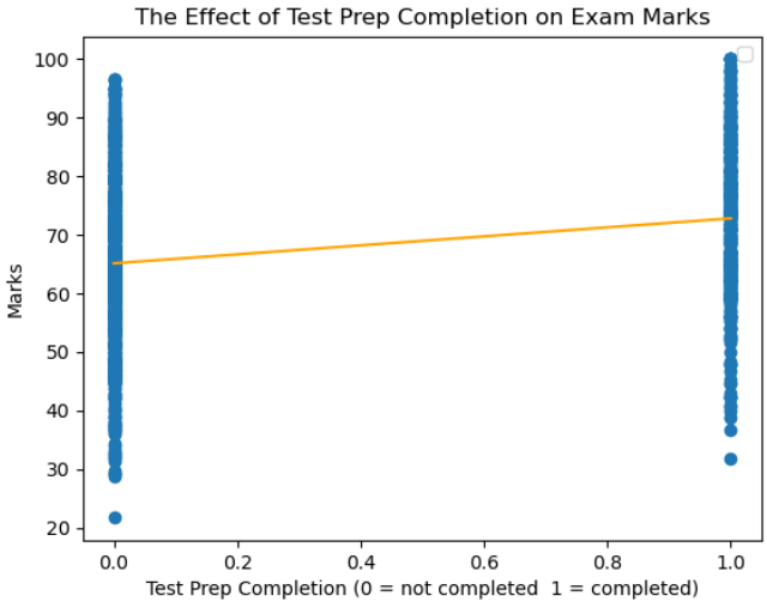


Data Modeling - Effect of Test Preparation on Exam Marks

If students complete a test preparation course before the exam, will their marks improve? We can verify this by performing another linear regression with "test preparation course_completed" as the independent variable and "average_score" as the dependent variable.

	coef	std err	t	P>|t|	[0.025	0.975]
const	65.2302	0.596	109.355	0.000	64.059	66.401
test preparation course_completed	7.5310	1.040	7.239	0.000	5.489	9.573

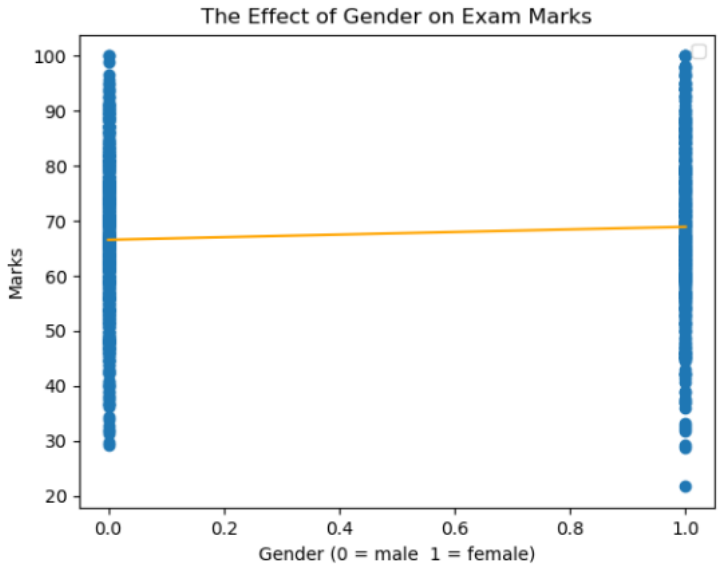
While the R2 is only ~6%, the model does show that completing a test prep course improves one's score by 7.5 points. The chart on the next page outlines this positive relationship.



Data Modeling - Effect of Gender on Exam Marks

Finally, a linear model using gender as the independent variable and student exam marks as the dependent variable shows a slight increase of ~2.5 marks if a student is female.

	coef	std err	t	P>|t|	[0.025	0.975]
const	66.6172	0.696	95.655	0.000	65.250	67.984
gender_female	2.5490	1.000	2.549	0.011	0.586	4.512



Appendix

Code - Factors that affect exam scores

```
import statsmodels.api as sm
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import metrics
from sklearn.model_selection import KFold

# Functions
def plotPredictionVsActual(title, y_test, predictions):
   plt.scatter(y_test, predictions)
   b, a = np.polyfit(y_test, predictions, deg=1)
   plt.legend(loc='best')
   plt.xlabel("Actual")
   plt.ylabel("Predicted")
   plt.title('Predicted (Y) vs. Actual (X): ' + title)
   xseq = np.linspace(0, 100, num=100)
   plt.plot(xseq, a + b*xseq, '-o', color='orange')
   plt.show()

FOLDER = "C:\\datasets\\"
FILE = 'exams.csv'

# Create DataFrame.
df = pd.read_csv(FOLDER + FILE)

# Show all columns.
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)

# Explore the data, check for missing values
print(df.describe())

# Create an "average score" column
df['average_score'] = round(((df['math score'] + df['reading score'] +
df['writing score']) / 3), 1)

# Create X and y
y = df['average_score']

X = df.copy()
del X['math score']
del X['reading score']
del X['writing score']
del X['average_score']
del X['id']

# Create dummy variables
X = pd.get_dummies(X, columns=['gender', 'lunch', 'test preparation course'])
del X['gender_male']
del X['lunch_free/reduced']
del X['test preparation course_none']
```

```
X = pd.get_dummies(X, columns=['race/ethnicity', 'parental level of
education'])

# Removed the following variables as they were not statistically significant
del X['parental level of education_some college']
del X['parental level of education_high school']

# Adding an intercept
X = sm.add_constant(X)

# prepare cross validation with three folds.
kfold = KFold(n_splits=8, shuffle=True)
rmseList = []
bicList = []
rsquareLst = []

count = 1

for train_index, test_index in kfold.split(X):
   X_train = X.loc[X.index.isin(train_index)]
   X_test = X.loc[X.index.isin(test_index)]
   y_train = y.loc[y.index.isin(train_index)]
   y_test = y.loc[y.index.isin(test_index)]

   # Perform linear regression.
   model = sm.OLS(y_train, X_train).fit()
   print(model.summary())

   y_pred = model.predict(X_test)  # make the predictions by the model
   mse = metrics.mean_squared_error(y_test, y_pred)
   rmse = np.sqrt(mse)
   rmseList.append(rmse)
   bic = model.bic
   bicList.append(bic)
   rsqr = model.rsquared
   rsquareLst.append(rsqr)

   print("\n***K-fold: " + str(count))
   print("RMSE:     " + str(rmse))
   print("BIC:      " + str(bic))
   print("R^2:      " + str(rsqr))

   count += 1

   plotPredictionVsActual("Grades", y_test, y_pred)

# Show averages of scores over multiple runs.
print("*******************************************")
print("\nScores for all folds:")
print("*******************************************")
print("RMSE Average :    " + str(np.mean(rmseList)))
print("RMSE SD:          " + str(np.std(rmseList)))
print("BIC Average :     " + str(np.mean(bicList)))
print("BIC SD:           " + str(np.std(bicList)))
print("RSQ Average :     " + str(np.mean(rsquareLst)))
```

```
print("RSQ SD:           " + str(np.std(rsquareLst)))
```

Code - Effect of Test Prep and Gender on Exam Marks

```
import statsmodels.api as sm
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
from sklearn import metrics

# Functions
def plotPredictionVsActual(title, xlabel, ylabel, y_test, predictions):
   plt.scatter(y_test, predictions)
   b, a = np.polyfit(y_test, predictions, deg=1)
   plt.legend(loc='best')
   plt.xlabel(xlabel)
   plt.ylabel(ylabel)
   plt.title(title)
   xseq = np.linspace(0, 1, num=100)
   plt.plot(xseq, a + b*xseq, '-', color='orange')
   plt.show()

FOLDER = "C:\\datasets\\"  # Windows
FILE = 'exams.csv'

# Create DataFrame.
df = pd.read_csv(FOLDER + FILE)

# Show all columns.
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)

# Create an "average score" column
df['average_score'] = round(((df['math score'] + df['reading score'] +
df['writing score']) / 3), 1)


# Create dummy variables
df = pd.get_dummies(df, columns=['gender', 'race/ethnicity', 'parental level
of education',
                                 'lunch', 'test preparation course'])

count = 0
while count < 2:
   if count == 0:
       X = df['test preparation course_completed']
   if count == 1:
       X = df['gender_female']
   y = df['average_score']
   x = X.copy()
```

```
    # Adding an intercept
    X = sm.add_constant(X)

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

    # Build and evaluate model.
    model = sm.OLS(y_train, X_train).fit()
    predictions = model.predict(X_test)  # make the predictions by the model
    print(model.summary())
    print('Root Mean Squared Error:',
          np.sqrt(metrics.mean_squared_error(y_test, predictions)))

    if count == 0:
        plotPredictionVsActual('The Effect of Test Prep Completion on Exam
Marks'
                               , 'Test Prep Completion (0 = not completed  1 =
completed)'
                               , 'Marks', x, y)
    if count == 1:
        plotPredictionVsActual('The Effect of Gender on Exam Marks'
                               , 'Gender (0 = male  1 = female)'
                               , 'Marks', x, y)
    count += 1
```