DATA1002 Project Stage 2-CensusSchool
SID: 480556587

Part 1: Analysis and Results

Domain Situation

This report investigates the correlation in a plethora of question on students' surveys across three different countries. In addition to this, a variety of statistics to indicate student's height has been calculated for each country in an attempt to reveal which variable has the strongest correlation in our data frame. There are 3 countries are included in this analysis, which includes Canada, New Zealand, the United State of America. The data also included some subgroups, such as gender has subgroup (Male and Female), handedness (right-hand or left-hand), and travel method to school (Several vehicles). For example, we could investigate handedness independent across three datasets. However, our great interest to the analysis is student's height thereby we want to use our variables from above to determine which one have greatest influence on our student's height. We also need to think about our confounder variable since some variables have subgroup which might be misleading our result. Overall, our students' height will then be astutely compared with our strongest correlated variable thereby identifying the relationship between them.

Origin of the data

Our datasets for this project were amalgamated from three main datasets, 'CENSUS AT SCHOOL New Zealand program' [1], as sourced from Stats NZ, CENSUS AT SCHOOL CANADA program [2] and 'CENSUS AT SCHOOL US program' [3], sourced in The Canadian or American Statistical Association. All three datasets are provided solely for the purposes of teaching and learning. This data cannot be reproduced in any way, other than by teachers or students for school curriculum and assessment activities. However, these datasets were come from national statistical societies. The clear documentation of each individual creator, as well as the reputability of national statistical societies as well as the original sources of data, such as Stats NZ, reinforces the reliability and validity of the dataset, and it is suitable for analysis.

Research Questions

In exploring our dataset, we have our two research questions:

1. Which variables have greatest impact in our students' height? And how to use them to estimate student's height?
2. Is handedness independent of gender in different age group?

   Before making any conclusions, we devised several summary statistics that align with generally used students' data, including:

   - **Year:** student's age in the school.

- **Gender:** students' sexuality
- **Handedness:** left right or right hand or both
- **Height:** recorded it in centimeter.
- **Feet_length:** recorded it in centimeter.
- **Travel method to school:** the types of method for student to go to school

This data is summarized in Figure 1,2,3, ordered in descending order by students' height, and showing only the top few students' data.

	Year	Gender	Ageyears	Country	Languages_spoken	Handedness	Height	Foot_Length	Travel method to school
29	13	male	17.0	Australia	1.0	left	198.0	31.0	bike
384	10	female	14.0	New Zealand	1.0	right	198.0	28.0	train
109	12	male	16.0	New Zealand	1.0	right	195.0	28.0	motor
218	12	male	18.0	New Zealand	1.0	right	195.0	32.0	motor
615	13	male	17.0	United States of America	2.0	ambi	190.0	33.0	bus
64	12	male	16.0	New Zealand	1.0	right	189.0	28.0	walk
8	13	male	17.0	New Zealand	1.0	right	188.0	28.0	motor
895	10	male	15.0	New Zealand	1.0	ambi	186.0	29.0	bus
83	10	male	14.0	New Zealand	2.0	left	185.0	26.0	walk
264	11	male	15.0	New Zealand	1.0	right	185.0	28.0	motor

	Country	ClassGrade	Gender	Ageyears	Handed	Height	Footlength_cm	Armspan_cm	Languages_spoken
139	USA	11	Male	17	Right-Handed	208	30	215.0	1.0
167	USA	11	Male	16	Right-Handed	204	32	202.0	1.0
135	USA	12	Male	18	Right-Handed	194	29	190.0	2.0
107	USA	12	Male	17	Right-Handed	193	29	191.0	2.0
245	USA	11	Male	18	Right-Handed	192	25	174.0	2.0
53	USA	12	Male	17	Right-Handed	191	34	184.0	2.0
114	USA	9	Male	15	Right-Handed	191	33	187.0	2.0
141	USA	12	Male	17	Right-Handed	190	27	179.0	1.0
204	USA	11	Male	15	Right-Handed	190	28	189.0	1.0
176	USA	11	Male	16	Right-Handed	188	27	190.0	1.0

	Country	Gender	Ageyears	Handed	Height	Foot_Length	Arm_Span	Languages_spoken	Travel_to_School
176	CA	Male	19	Right-Handed	197	26	189	3	Walk
554	CA	Male	18	Right-Handed	196	26	188	1	Car
987	CA	Female	19	Right-Handed	196	22	176	3	Walk
341	CA	Male	18	Right-Handed	194	21	194	1	Car
235	CA	Male	20	Left-Handed	193	32	189	2	Walk
632	CA	Male	18	Right-Handed	193	26	200	1	Car
892	CA	Male	18	Right-Handed	193	30	200	1	Bus
467	CA	Male	17	Right-Handed	192	28	189	1	Walk
523	CA	Male	17	Right-Handed	192	31	201	2	Car
884	CA	Male	14	Right-Handed	192	30	187	2	Walk

Figure: Summary Student's Data Table for Top 10 height Rankings

Based on our figures from 3 cleaned students' dataset, we have identified that most people are the right hand and also, we have determined that our students' height has a strong correlation to a length of arm spin, foot length also their gender and ages.

Now given a general snapshot of students' height across three datasets, we continued on to analyze how these variables may affect these results. It was expected that student with a higher height would have a larger feet length and arm span length as the growth of the body's bones is related to the growth of height. When the bones of the legs grow, the bones of the feet and the bones of the fingers will grow to vary degrees. Therefore, in the case of a long body, the limbs, the soles of the feet, and the palms will "grow up."

Figure 4,5,6: Pivot Table of students' height across different gender in three datasets

Gender	mean Height	len Height	median Height
Female	152.617284	162	160
Male	165.719178	146	173

Gender	mean Height	len Height	median Height
Female	157.029979	467	159
Male	160.596226	530	161

Gender	mean Height	len Height	median Height
female	155.514970	501.0	158.0
male	157.659155	355.0	157.0

Above is a pivot table of counts of students' height for each gender. As a result, we determined that even we split all samples into two different gender group, boys are slightly higher girls in height.

Figure 7,8,9: Pivot Table of student handedness across different year or age group in three datasets

Year	mean Height	len Height	median Height
4	129.500000	6.0	134.0
5	139.682692	104.0	138.5
6	145.094118	85.0	145.0
7	149.459016	122.0	150.5
8	156.787402	127.0	157.0
9	162.361582	177.0	163.0
10	165.145985	137.0	165.0
11	168.724138	58.0	169.0
12	175.916667	24.0	176.0
13	170.500000	16.0	172.0

ClassGrade	mean Height	len Height	median Height
4	144.714286	7	142.0
5	147.888889	9	148.0
6	144.366667	30	150.5
7	138.210526	19	162.0
8	160.909091	22	165.0
9	137.125000	16	159.0
10	168.157895	19	165.0
11	165.560976	41	168.0
12	164.834483	145	168.0

Ageyears	mean Height	len Height	median Height
5	134.000000	1	134.0
6	150.250000	4	152.5
8	141.333333	3	137.0
9	139.285714	28	137.5
10	140.534483	58	141.0
11	147.308511	94	147.0
12	153.700000	170	154.0
13	159.326271	236	160.0
14	164.562500	160	164.0
15	165.098592	71	166.0
16	166.028571	35	167.0
17	172.950617	81	174.0
18	173.555556	45	173.0
19	178.750000	8	179.0
20	183.500000	2	183.5
21	183.000000	1	183.0

Above is a pivot table of counts of students' height for each age/year. As a result, we determined that even we split all samples into several different age/year group, the change for height for the majority of student are strongly correlated to their ages.

Figure 10: Pivot Table of student handedness across different gender in three datasets

Handedness	Gender	
ambi	female	25.0
	male	27.0
left	female	47.0
	male	50.0
right	female	429.0
	male	278.0

Handed	Gender	
Ambidextrous	Female	9
	Male	7
Left-Handed	Female	15
	Male	13
Right-Handed	Female	138
	Male	126

Handed	Gender	
B	Female	21
	Male	36
Left-Handed	Female	36
	Male	50
Right-Handed	Female	410
	Male	444

Above is a pivot table of counts of handedness for each gender. As a result, we determined that even we split all samples into two different gender group, the majority of student are right-handed.

Limitations and Further Questions

Three data sets used in this report has two main limitations.

First, the data are stratified by school year, and the sample includes 20 students each year. This allows us to come up with effective claims, for example, about the relative proportion of students in different academic years. However, this makes it difficult to claims for a whole student population. As students drop out of school as they get older, the total number of students enrolled each year tends to be less and less. However, our analysis necessarily considers the proportion of students per year equal, thus overestimating the impact of statistics on students in grades 13 and underestimating the impact of statistics on fourth-grade students. However, the sample roughly represents demographic data for students and our analysis is still valid.

Second, the survey was completed by volunteer students whose teachers participated in the Census AtSchool program and was able to provide electronic devices connected to the

Internet for all children in their class to complete the survey. This may mean that our data set is inherently biased towards wealthier schools because funding and free time are required to complete the survey. Having said that, there seems to be no strong demographic bias.

Conclusion

In conclusion, we found the following results for our sample of Students from the CensusAtSchool database:

Analysis of the pivot table has shown that Students height has a strong correlation to Ages, Years, Foot length, Gender and Length of Arm span. Also, by using a pivot table, we identify that the majority of students are right-handed no matter their gender. Our analysis also found that handedness was independent of gender. The CensusAtSchool program gives us a wonderful insight into the lives and opinions of students and allows us to practice our data analysis skills on a real-world dataset.

Part 2: Methodology of Analysis

Use the correlation matrix to determine the relationship to our students' height

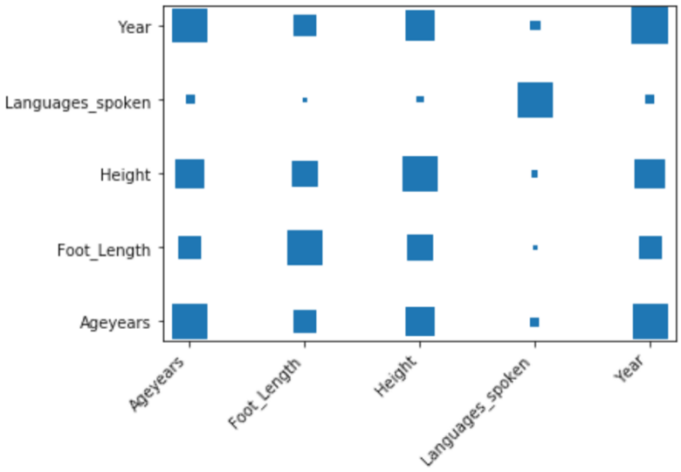
To begin, in order to effectively make comparisons between these variables, we have to use the correlation matrix to determine the relationship in them. A correlation matrix is introduced to evaluate to what extent each variable is related to students' height.

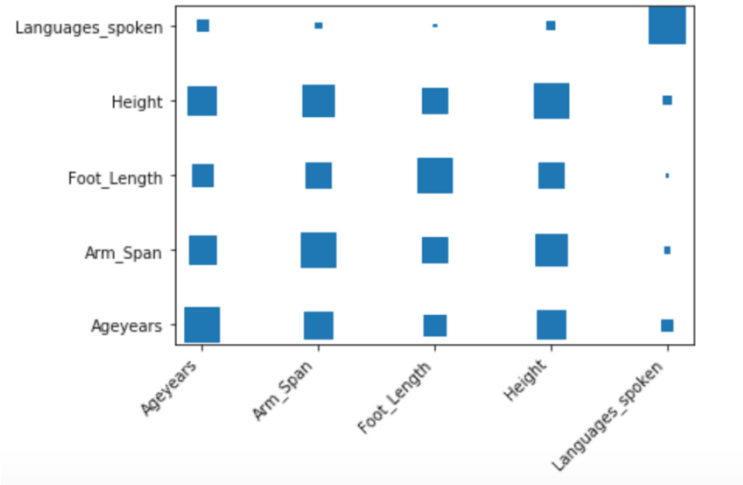
Correlation Matrix Formula:

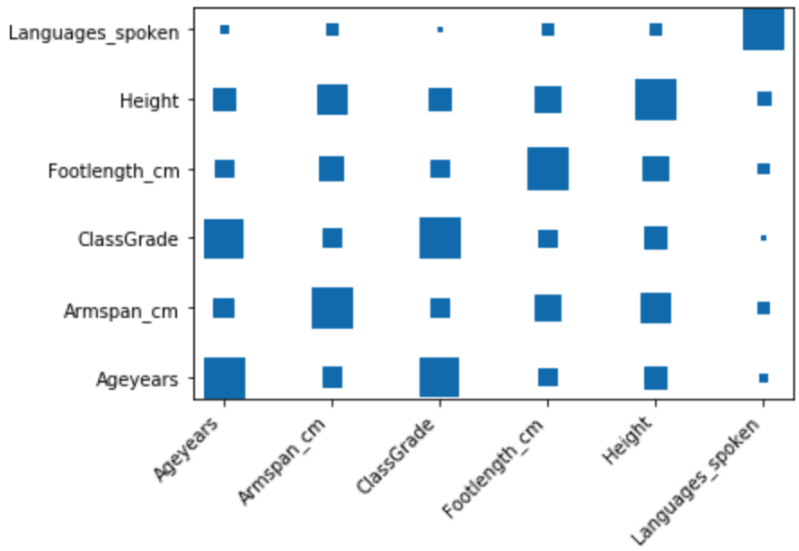
$$\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathrm{E}\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y}$$

```
# Step 1 - Make a scatter plot with square markers, set column names as labels

def heatmap(x, y, size):
    fig, ax = plt.subplots()

    # Mapping from column names to integer coordinates
    x_labels = [v for v in sorted(x.unique())]
    y_labels = [v for v in sorted(y.unique())]
    x_to_num = {p[1]:p[0] for p in enumerate(x_labels)}
    y_to_num = {p[1]:p[0] for p in enumerate(y_labels)}

    size_scale = 500
    ax.scatter(
        x=x.map(x_to_num), # Use mapping for x
        y=y.map(y_to_num), # Use mapping for y
        s=size * size_scale, # Vector of square sizes, proportional to size parameter
        marker='s' # Use square as scatterplot marker
    )

    # Show column labels on the axes
    ax.set_xticks([x_to_num[v] for v in x_labels])
    ax.set_xticklabels(x_labels, rotation=45, horizontalalignment='right')
    ax.set_yticks([y_to_num[v] for v in y_labels])
    ax.set_yticklabels(y_labels)

data = pd.read_csv("/Users/linenzheng/Desktop/cleaned data/clean1.csv")
columns = ['Year', 'Gender','Ageyears','Country','Languages_spoken','Handedness','Height', 'Foot_Length','Travel metho
corr = data[columns].corr()
corr = pd.melt(corr.reset_index(), id_vars='index') # Unpivot the dataframe, so we can get pair of arrays for x and y
corr.columns = ['x', 'y', 'value']
heatmap(
    x=corr['x'],
    y=corr['y'],
    size=corr['value'].abs()
)
```

It’s just a scatter plot， so we want to plot elements on a grid made by two categorical axes, we can use a scatter plot. Firstly, we make a scatter plot with square markers, set column names as labels, then mapping from column names to integer coordinates, like using mapping for x or y. Since the scatterplot requires x and y to be numeric arrays, we need to map our column names to numbers. And since we want our axis ticks to show column names instead of those numbers, we need to set custom ticks and tick labels. Finally, there’s code that loads the dataset and calculates all the correlations for three datasets as below.







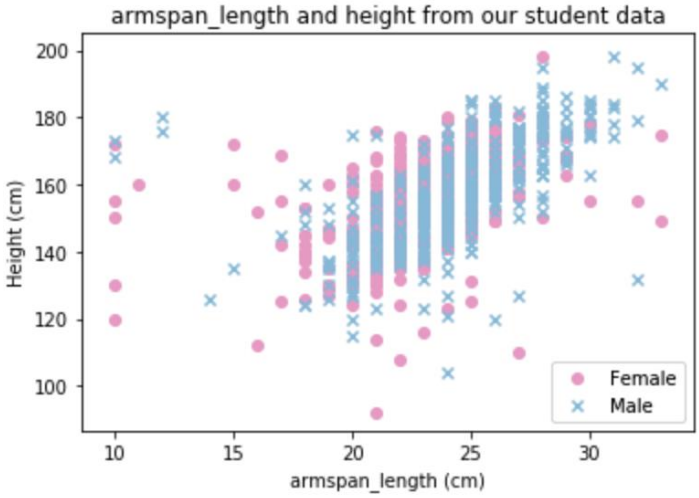
It is clear to say that age/foot length and length of arm span has the largest correlation coefficient to students' height, we may therefore conclude that these variables have greatest impact to our students’ height. On the other hand, other variables are less related to mark according to the graph. Different gender actually has a very different height distribution, but we do not consider this variable in the correlation matrix. Since it is not a variable, but we already know that from part 1, since it would be a very correlate to our students’ height even though didn’t perform a one-hot encoding to gender.

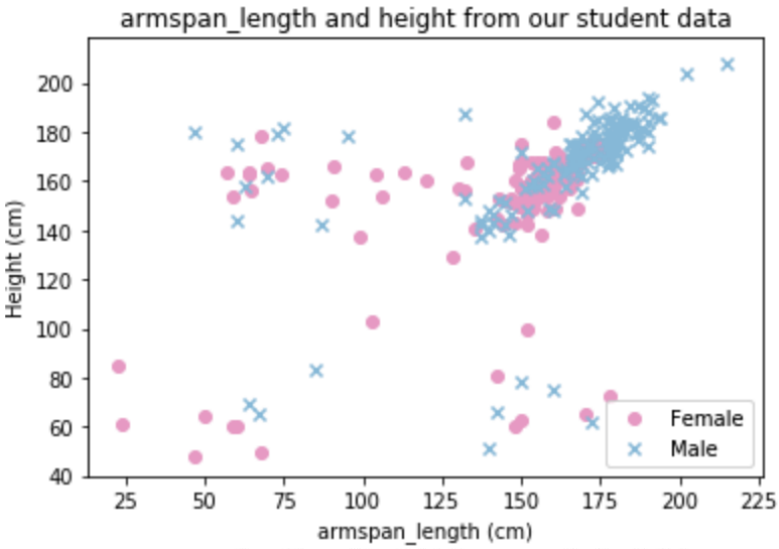
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('/Users/linenzheng/Desktop/cleaned data/clean1.csv')
females = df[df['Gender'] == 'female']
males = df[df['Gender'] == 'male']
f_armspan_length = females.values[:, 7]
f_heights = females.values[:, 6]
m_armspan_length = males.values[:, 7]
m_heights = males.values[:, 6]

plt.scatter(f_armspan_length, f_heights, color='#e9a3c9', marker='o', label='Female')
plt.scatter(m_armspan_length, m_heights, color='#91bfdb', marker='x', label='Male')
plt.title('armspan_length and height from our student data')
plt.xlabel('armspan_length (cm)')
plt.ylabel('Height (cm)')
plt.legend(loc='lower right')
plt.show()
```

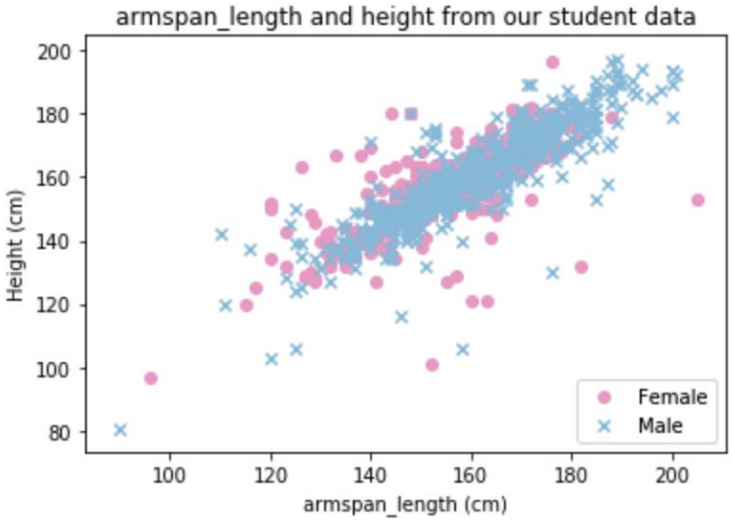
Firstly, we want to distinguish between females and males in our plot, we need to first slice the data. Our program slices the data based on gender with these two lines of code, then to plot the markers on the axes, we need to provide the x and y values. The first needed parameter is a series or a list of values that will be used to position the markers along the x-axis. For our plot, we supply f_armspan_length to place weight as the x-axis. Likewise, the second parameter is a series or a list of values that will be used to position the markers along the y-axis. In our case, we supply f_heights to place height as the y-axis. Finally, after plotting all the markers, our program then adds title and labels to the chart.

Using Matplotlib to create scatter plots

As we have identified that some attributes have a strong correlation to our students' height by using correlation matrix from above. Now, we want to create a scatter plot thereby visualized the relationship between our chosen variables.



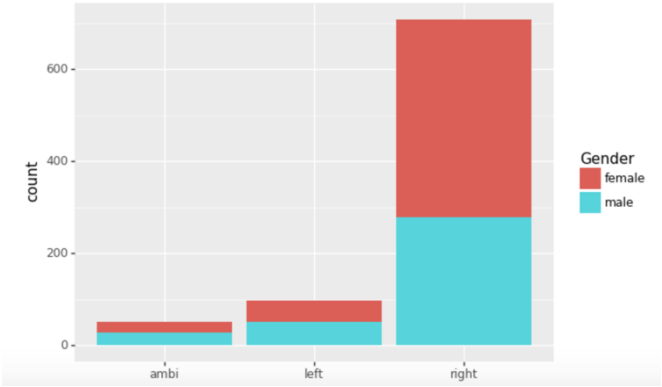


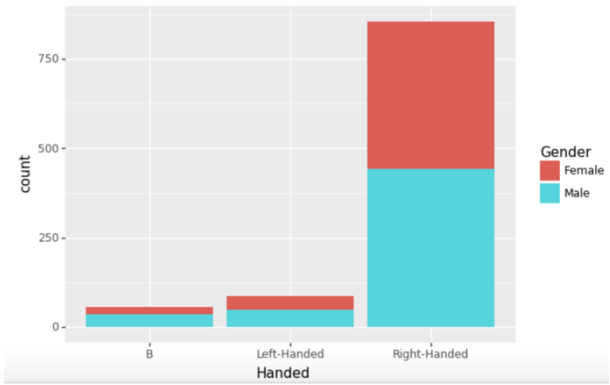


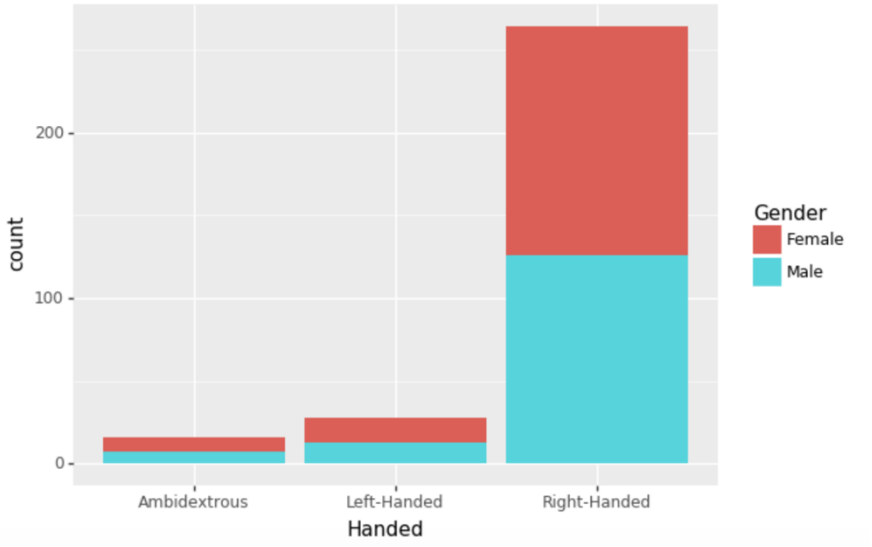
Using Matplotlib to create bar plots to determine the handedness

In Matplotlib, we applied the ggplot theme, then using the bar plot method to plot our variables for handedness, all variables extracted from the data frame.

```
import numpy as np
import pandas as pd
from plotnine import *

%matplotlib inline
df = pd.read_csv("/Users/linenzheng/Desktop/cleaned data/clean1.csv").dropna()
ggplot(df, aes(x='Handedness', fill = 'Gender')) + \
    geom_bar(stat = 'count')
```



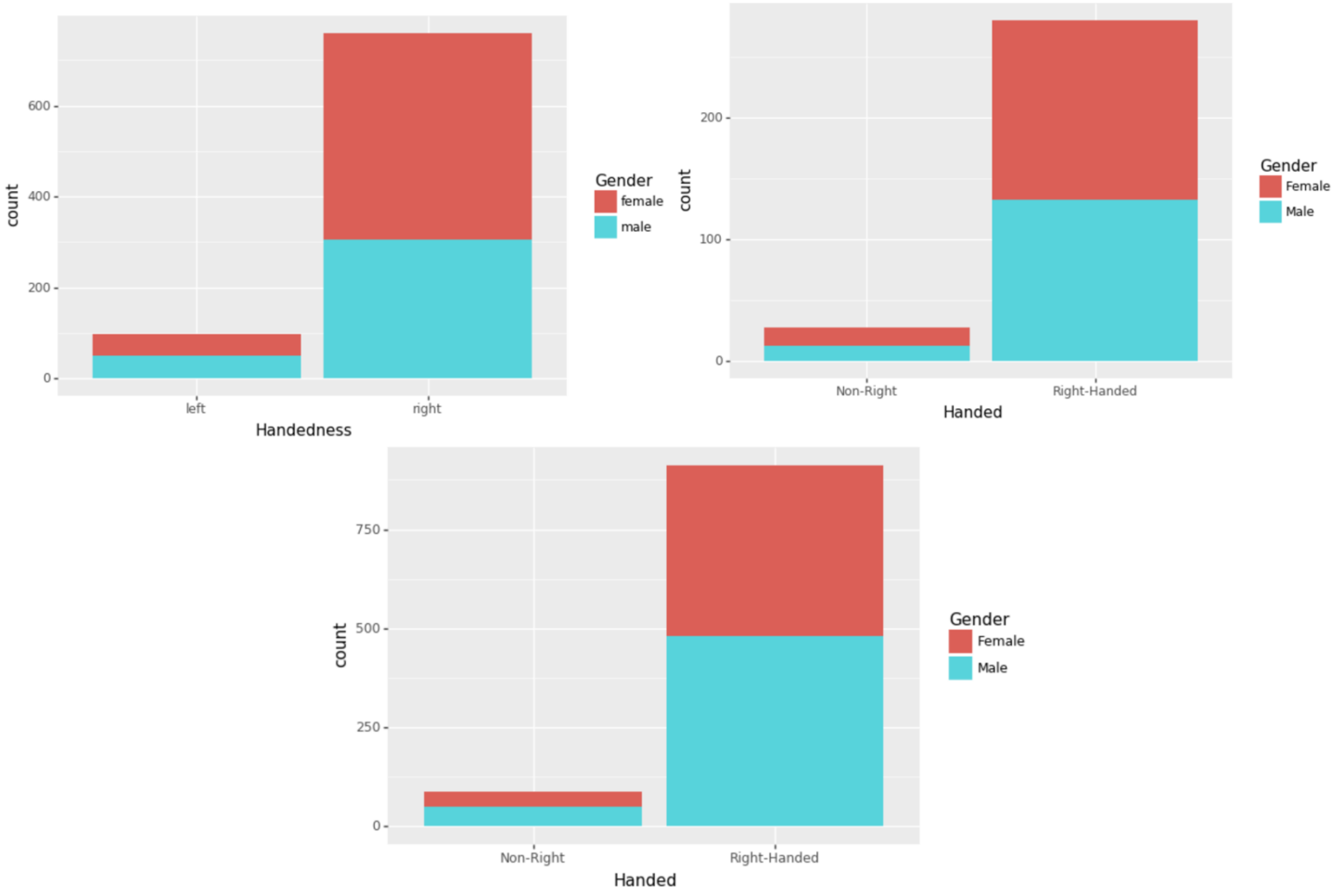




Note that in this test, we lump people who are ambidextrous into the group of people who are right handed.

We begin by removing missing values and recoding our data into right handers, and not right-handers.

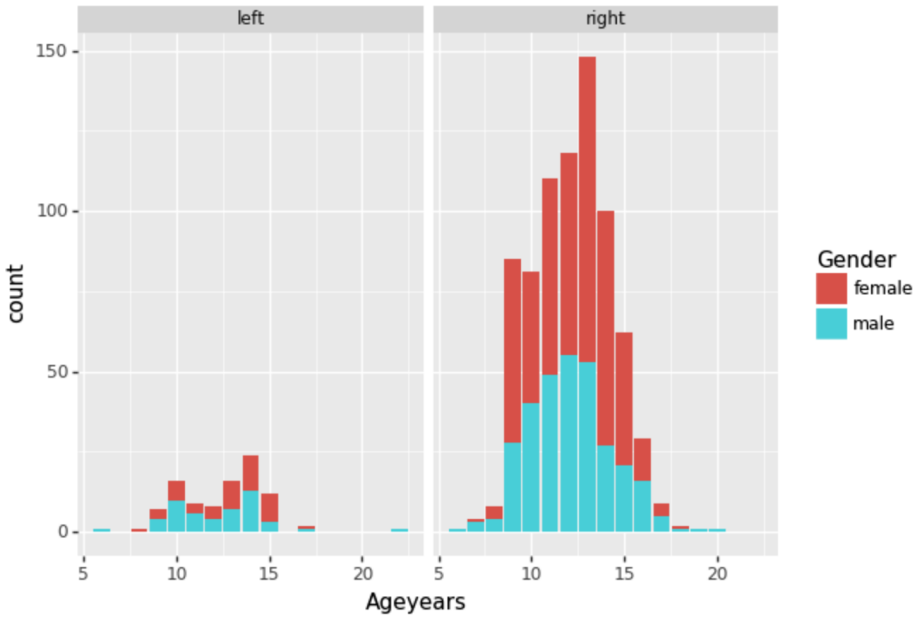
```
df['Handedness']=df['Handedness'].replace({'Left-Handed': 'Non-Right', 'ambi':'right'})
```

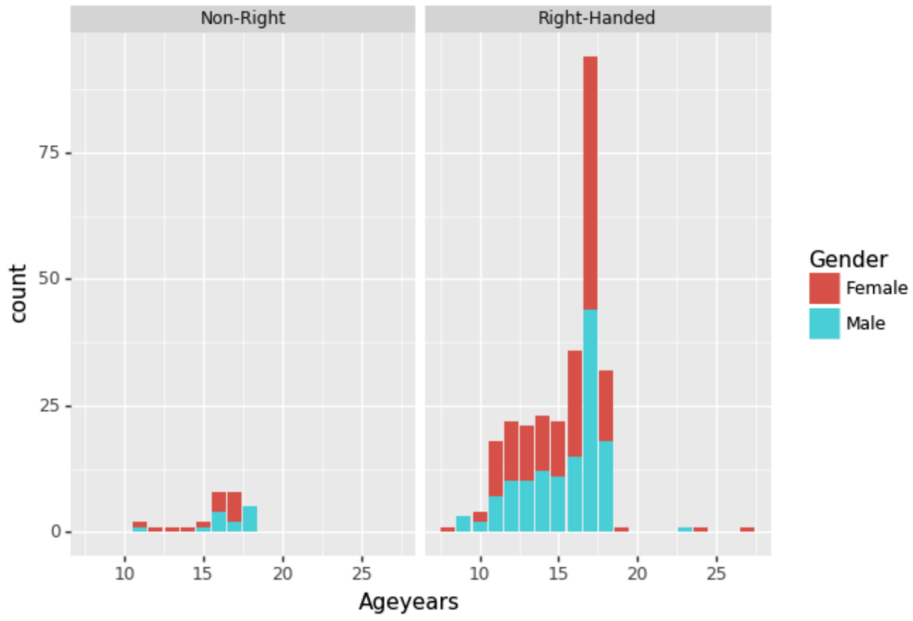


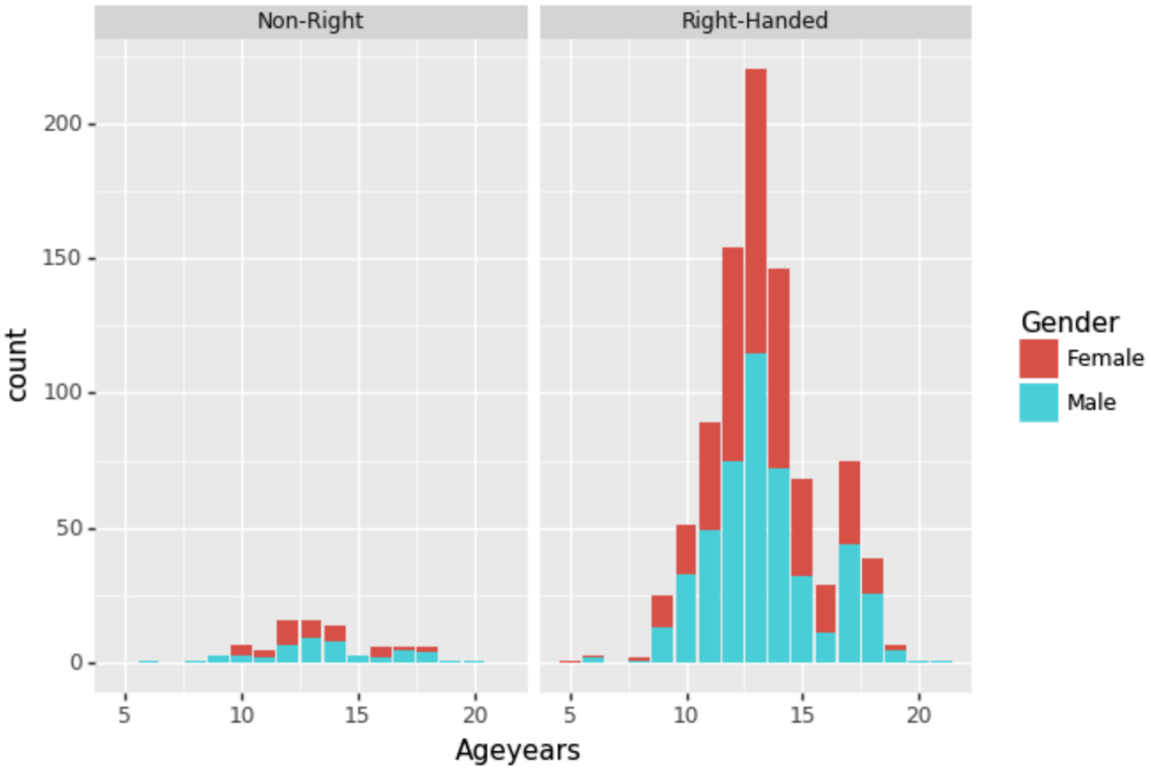
Continue using Matplotlib to create bar plots to determine the handedness independent of gender in different age group

ggplot has a special technique called faceting that allows splitting one plot into multiple plots based on a factor included in the dataset. We will use it to make one plot for a time series for each age groups.

```
ggplot(df, aes(x='Ageyears', fill = 'Gender')) + \
    geom_bar(stat = 'count') + \
    facet_wrap(['Handed'])
```







As can be seen from the figure, handedness does not depend on the age group and gender. However, when the code was finally implemented, we had an interactive expressive plot. We use multiple libraries to deepen our collective understanding of the Python mapping common process and the complexity, advantages and disadvantages of the four modules we experimented with. In the end, we assert that ggplot is the best module and is the easiest to use as both a creator and a user. In addition to these modules, this is not only immediately useful for the careful preparation of this report, but also for our future experience as a data scientist in troubleshooting and fixing code.

Part 3: Building predictive models

Predictive models that attempted to predict students' height based on their students' armlengths or feet length or age group, were built using 3 separate supervised machine learning techniques:

1. Multiple Linear Regression

2. K Nearest Neighbours

3. Neural networks

In general, none of these models were very accurate, because students' height might be influenced by their gender since males are slightly higher than females.

Initially, we created a simple multiple linear regression predictive model which was built using ages year, feet length and length of arm span columns as inputs; and students' height columns as outputs.

```
import pandas as pd
from math import sqrt
from sklearn import linear_model
from sklearn import metrics
from sklearn.model_selection import train_test_split
df = pd.read_csv("/Users/linenzheng/Desktop/cleaned data/clean2.csv").dropna()
X = df.values[:, 4:7]      # slice dataFrame for input variables
y = df.values[:, 7]        # slice dataFrame for target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
regr = linear_model.LinearRegression().fit(X_train, y_train)
# Let's create one sample and predict the height
sample = [18, 35, 180]        # a sample with Ageyears: 18 Footlength_cm: 35 Armspan_cm: 180
print('----- Sample case -----')
for column, value in zip(list(df)[4:7], sample):
    print(column + ': ' + str(value))
sample_pred = regr.predict([sample])
print('Predicted students height:', int(sample_pred))
print('-----------------------')

# The coefficients
print('Coefficients:')
print(regr.coef_)
# Use the model to predict y from X_test
y_pred = regr.predict(X_test)
# Root mean squared error
mse = metrics.mean_squared_error(y_test, y_pred)
print('Root mean squared error (RMSE):', sqrt(mse))
# R-squared score: 1 is perfect prediction
print('R-squared score:', metrics.r2_score(y_test, y_pred))
```

In general, the code for the model was placed under a loop that first considered the height column for y, evaluated the code, produced output then considered the height column for y. Within the loop, the code first sliced the above data frame for the input variables armlengths/ages/feet length. Then 10% of the training data was set aside for testing, and a Linear Regression model fitted. The sample case of values of armlengths/ages/feet length was inputted into the model and the y value (height) was outputted. Finally, the R-squared and RMSE statistics were calculated. This general process was also used for the other machine learning models.

```
----- Sample case -----
Ageyears: 18
Footlength_cm: 35
Armspan_cm: 180
Predicted students height: 185
-----------------------
Coefficients:
[1.88144929 1.0527201  0.33158278]
Root mean squared error (RMSE): 33.773323752874866
R-squared score: 0.19733771502925535
```

```
----- Sample case -----
Year: 12
Ageyears: 18
Foot_Length: 35
Predicted students height: 191
----------------------
Coefficients:
[3.50431697 0.47758476 1.6581211 ]
Root mean squared error (RMSE): 10.05200015213131
R-squared score: 0.47845176582117566
 ----- Sample case -----
 Ageyears: 18
 Foot_Length: 35
 Arm_Span: 180
 Predicted students height: 183
 ----------------------
 Coefficients:
 [1.19262383 0.5293005  0.63223615]
 Root mean squared error (RMSE): 6.008497765784784
 R-squared score: 0.7611217945059753
```

In the multiple linear regression model, we used armlengths/ages/feet length as input variables (X) and we used height as the target variable (y). This would predict the student's height across three data sets, and predicted height is in centimeter. In this case, the multiple linear regression predicted the student height would be roughly around 185 across three datasets.

The coefficients of the multiple linear regression model were [1.19262383 for Ageyears, 0.5293005 for Foot_length, 0.63223615 for Arm_span], with the equation taking the form of home team goals = 1.19262383*Ageyear+0.5293005*Foot_length+0.63223615*Arm_span+c (for last figure 3 from above, similar method in figure 1 and 2) for some constant c which sci-kit learn was unable to find.

The accuracy of the model was then tested using the 10% leftover data that was set aside earlier. The root means squared error (RMSE) was reasonable bigger, given that the range of our variables in the dataset was fairly large. This indicates that the model's predicted values were fairly close to the true value. However, the R-squared score was extremely low across three datasets which means that the data did not fit the model well at all. However, the model was probably much better than a straight line. This could be explained by the lack of correlation between the input and target variables, as we put bigger value in our input, our output is getting bigger too. While all of these are weaknesses for the multiple linear regression model, a strength is that multiple linear regression is much easier to analyze and

understand compared to other methods, as the model is essentially a rather simple algebraic equation. In our experience, it was also the fastest model to compute.

K-nearest Neighbours

Since it was discovered that the above model may not be appropriate, given the low R-squared values and the hypothesized presence of many outliers in large clusters, it was decided to try a K-nearest neighbors approach. This approach would attempt to cluster data points that are similar enough to each other, by finding the k closest points for each input in the training set, then averaging the y-values for those points. In our case, fork = 4, the algorithm would try to find the 4 closest Neighbours for our inputs and average the target variables for each.

```
import pandas as pd
from math import sqrt
from sklearn import metrics
from sklearn import neighbors
from sklearn.model_selection import train_test_split
df = pd.read_csv("/Users/linenzheng/Desktop/cleaned data/clean3.csv").dropna()
X = df.values[:, 3:6]      # slice dataFrame for input variables
y = df.values[:, 6]        # slice dataFrame for target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
neigh = neighbors.KNeighborsRegressor(n_neighbors=4).fit(X_train, y_train)
# Let's create one sample and predict the height
sample = [18, 35, 180]        # a sample with Ageyears: 18 Foot_Length: 35 Arm_Span: 180
print('----- Sample case -----')
for column, value in zip(list(df)[3:6], sample):
 print(column + ': ' + str(value))
sample_pred = neigh.predict([sample])
print('Predicted height:', int(sample_pred))
print('-----------------------')

# Use the model to predict X_test
y_pred = neigh.predict(X_test)
# Root mean squared error
mse = metrics.mean_squared_error(y_test, y_pred)
print('Root mean squared error (RMSE):', sqrt(mse))
# R-squared score: 1 is perfect prediction
print('R-squared score:', metrics.r2_score(y_test, y_pred))
```

```
----- Sample case -----
Ageyears: 18
Footlength_cm: 35
Armspan_cm: 180
Predicted height: 178
-----------------------
Root mean squared error (RMSE): 33.11118853879475
R-squared score: 0.22850202395057007
----- Sample case -----
Year: 12
Ageyears: 18
Foot_Length: 35
Predicted height: 189
-----------------------
Root mean squared error (RMSE): 11.283767152985332
R-squared score: 0.3427997232132389

----- Sample case -----
Ageyears: 18
Foot_Length: 35
Arm_Span: 180
Predicted height: 182
-----------------------
Root mean squared error (RMSE): 8.23976638018336
R-squared score: 0.5507640361115742
```

Again, similar to before a model was generated to predict home team scores. Both the root mean squared error and the R-squared score for three data sets appeared to suggest that this model was actually better than a straightforward multiple linear regression, with a less root mean squared error in both cases. In particular, the small R-squared score suggested the model was better than a straight line (i.e. where a constant value of y was predicted regardless of input variables). In this case, the actual predicted height was to the result of the linear regression module from above. A benefit of this model though was that it was both more accurate than polynomial regression and faster to run. This allowed the model to be run multiple times with different settings for k, as explored below. This revealed another benefit: that for sufficiently large values of k, the RMSE and thus accuracy of the model improves and eventually becomes slightly better than the multiple linear regression model.

Neural Networks

Since none of the other algorithms were successful, it was decided to use a neural network regression method that allowed the machine to determine the best algorithm for the prediction model. Broadly speaking, the neural network receives input, performs matrix multiplication using a "hidden layer", and then outputs the result [9]. The algorithm will

randomly select the input multiplied by the weight. This multiplication process of random selection weights is designed to mimic "neurons" in the mammalian brain. We can use the hidden_layer_sizes option below to choose the size of the hidden layer.

We are using the solver 'lbfgs'

We then fit the MLP Regressor model, which was chosen when predicting values rather than classifying them as usual. This is done using the following settings: Solver = 'lbfgs', according to sci-kit learning documentation [10], which is an optimization program that converges faster and performs better than many other options. Hidden_layer_sizes = (2,2) will change the size of the hidden layer that we are multiply the input value; alpha = 1x10-5, which is the L2 penalty parameter (this will make the data more regular by reducing the likelihood of overfitting); And random state, which makes our randomness consistent with different runs.

```
import pandas as pd
from math import sqrt
from sklearn.neural_network import MLPRegressor
from sklearn import metrics
from sklearn import neighbors
from sklearn.model_selection import train_test_split
df = pd.read_csv("/Users/linenzheng/Desktop/cleaned data/clean1.csv").dropna()
X = df.values[:, 2:5]      # slice dataFrame for input variables
y = df.values[:, 5]        # slice dataFrame for target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
clf = MLPRegressor(solver='lbfgs',alpha=1e-5,hidden_layer_sizes=(2,2),random_state=1)
clf.fit(X_train, y_train)
# Let's create one sample and predict the height
sample = [12, 18, 35]        # a sample with Year: 12 Ageyears: 18 Foot_Length: 35
print('----- Sample case -----')
for column, value in zip(list(df)[2:5], sample):
    print(column + ': ' + str(value))
sample_pred = clf.predict([sample])
print('Predicted height:', int(sample_pred))
print('-----------------------')

# Use the model to predict X_test
y_pred = clf.predict(X_test)
# Root mean squared error
mse = metrics.mean_squared_error(y_test, y_pred)
print('Root mean squared error (RMSE):', sqrt(mse))
# R-squared score: 1 is perfect prediction
print('R-squared score:', metrics.r2_score(y_test, y_pred))
```

```
----- Sample case -----
Ageyears: 12
Footlength_cm: 30
Armspan_cm: 185
Predicted height: 160
-----------------------
Root mean squared error (RMSE): 41.43754704874461
R-squared score: -0.20829597385095622
```

```
----- Sample case -----
Year: 12
Ageyears: 18
Foot_Length: 35
Predicted height: 156
----------------------
Root mean squared error (RMSE): 13.974998213697376
R-squared score: -0.008075427001686286
----- Sample case -----
Ageyears: 18
Foot_Length: 35
Arm_Span: 180
Predicted height: 158
----------------------
Root mean squared error (RMSE): 12.297908127734427
R-squared score: -0.0007076238073076002
```

This time, the algorithm is getting more worse than before a model was generated to predict height. Both the root mean squared error and the R-squared score for three data sets appeared to suggest that this model was actually worse than a straightforward multiple linear regression, with a higher root mean squared error in both cases. In particular, the negative R-squared score suggested the model was worse than a straight line. Previous two algorithm have similar predicted height, but this one is not.

Conclusions

The multiple linear regression model seemed to perform the best, followed by k-nearest neighbours and neural networks is the worst one.

The apparent success of multiple linear regression, often considered a very simple and inaccurate model, was most likely due to outlier in the age/feet length/arm span column, with the data being very heavily skewed towards lower or larger numbers.
The first two and last one models gave a very low or negative R-squared value, meaning that a straight-line predictor that ignored all input and just gave a consistent output would have performed very similarly to our models.

Overall, the first two modules gave us a reasonable predict height, so in practice, multiple linear regression model and k-nearest neighbours are the best way to predict our student height and we also determined that age/year/arm span/feet length have strongest impact to our students' height. Moreover, back to section 2, we also find age group/gender is independent to students' handedness.