CSC398 Design and Tech Industry Analysis

July 30, 2020

CSC398 Report: Participation in industry-based Design and Tech Conferences

Submitted by: Akshit Goyal (1005095068)

0.0.1 Introduction

The purpose of this report is to analyse the participation of people in a particular chosen industry-based Design and Tech Conference i.e. The Future of Innovation, Technology, and Creativity(FITC) Toronto.

Participation in technology is changing over the years. It is important to analyse the trends and address the issues of under-representation of certain ethnic groups and genders. This report is an attempt to get an insight into how the community is represented and how its diversity changes over the years in the area of Design and Tech. To be specific, we will be looking at the years 2002-2019.

To achieve this, we will be testing multiple hypotheses. We will be testing the change in the population, representation of genders, and various ethnic groups.

Data Collection: In order to test these hypothesis, we had to collect relevant data. This was achieved by a web scrapper tool called Scrapy to extract names from the conference websites. Further information on names like gender and race-ethnicaity was obtained using a paid API i.e. NamSor.

0.0.2 Hypothesis: Number of people attending the Design and Tech conference increases each year.

The number of participants varies each year and the new technologies coming up rapidly and also new fields being introduced, it would interesting to test this hypothesis to measure the impact on Design and Tech conferences.

To test this, we will be using Linear Regression. First, we will set our Null Hypothesis.

Null Hypothesis: Population remains the same over the years. This implies that our coefficient of x i.e. B(Beta Value) in a linear regression model($y = Bx + c$) must be 0.

In the analysis, we will be using various python libraries for testing this.

[25]:
```
import csv
import numpy as np
import pandas as pd
```

```python
import matplotlib.pyplot as plt
import statsmodels.api as sm

# Read population data from the CSV file.
data = pd.read_csv("population_data.csv")
data
# Year 1 corresponds to '2002' and similarly other index values map to␣
↪consective years untill 2019 i.e. 17.
```

[25]:

	Year	Population
0	1	25
1	2	29
2	3	69
3	4	89
4	5	54
5	6	90
6	7	86
7	8	81
8	9	83
9	10	73
10	11	69
11	12	73
12	13	93
13	14	69
14	15	81
15	16	72
16	17	70
17	18	75

[26]:
```python
data.describe()
```

[26]:

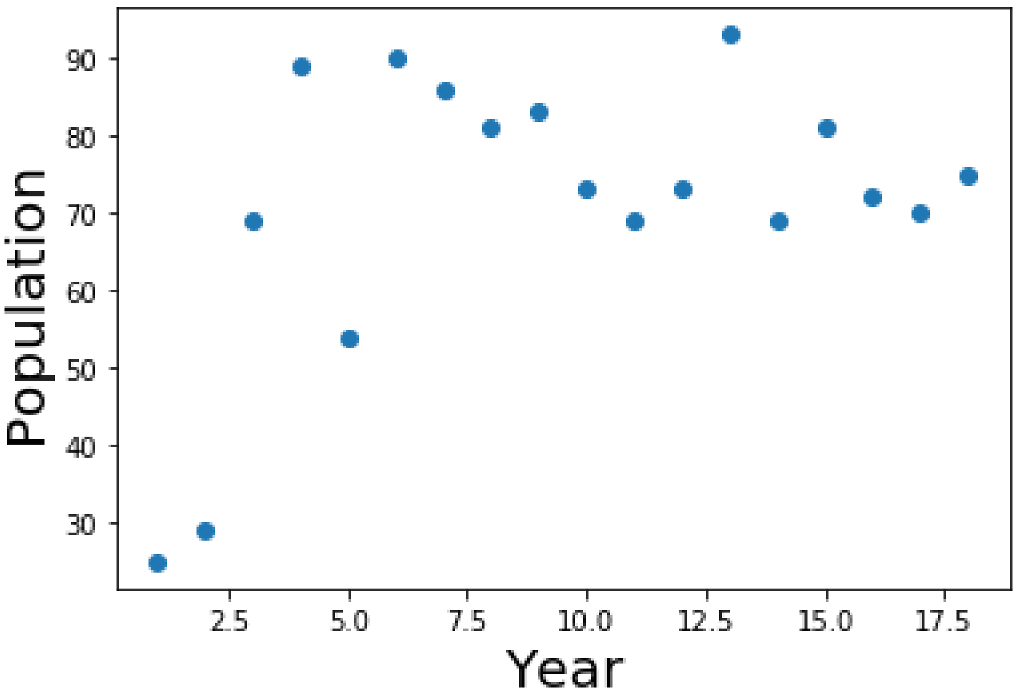
	Year	Population
count	18.000000	18.000000
mean	9.500000	71.166667
std	5.338539	18.699638
min	1.000000	25.000000
25%	5.250000	69.000000
50%	9.500000	73.000000
75%	13.750000	82.500000
max	18.000000	93.000000

Define the dependent(y) and the independent variable(x1)

[3]:
```python
y = data['Population']
x1 = data['Year']
```

Explore the Data

[4]:
```
plt.scatter(x1,y)
plt.xlabel('Year', fontsize = 20)
plt.ylabel('Population', fontsize = 20)
plt.show()
```



[5]:
```
x = sm.add_constant(x1)
results = sm.OLS(y,x).fit()
results.summary()
```

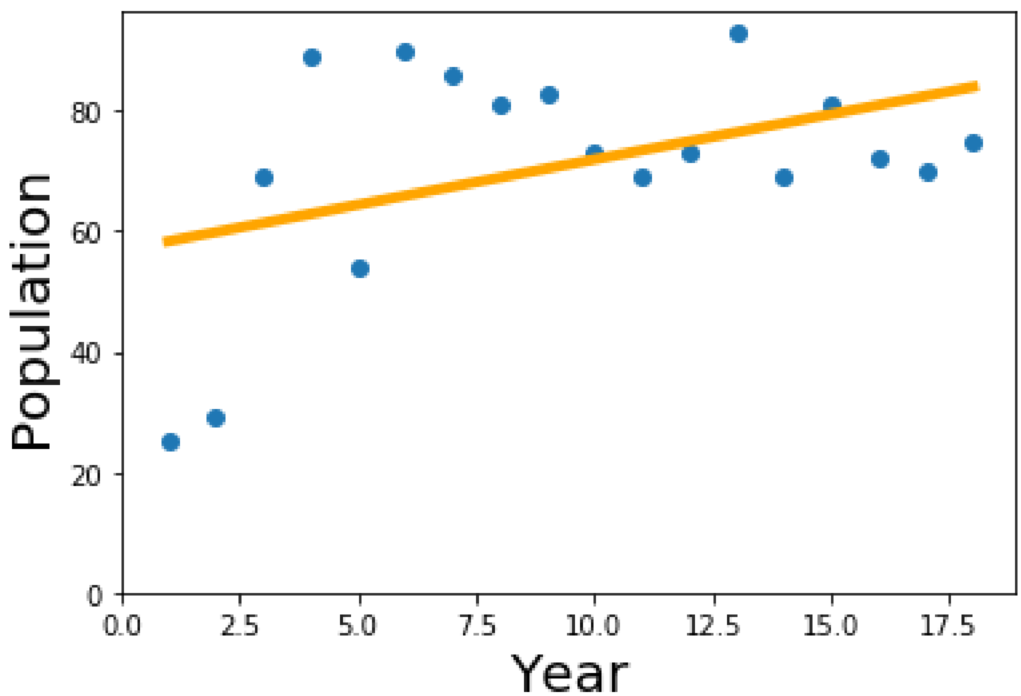
```
/home/akki/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1535:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=18
  "anyway, n=%i" % int(n))
```

[5]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:             Population   R-squared:                       0.185
Model:                            OLS   Adj. R-squared:                  0.134
Method:                 Least Squares   F-statistic:                     3.639
Date:                Sun, 26 Jul 2020   Prob (F-statistic):             0.0746
Time:                        21:30:03   Log-Likelihood:                -75.895
No. Observations:                  18   AIC:                             155.8
```

```
Df Residuals:                      16   BIC:                             157.6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         56.8431      8.556      6.644      0.000      38.706      74.980
Year           1.5077      0.790      1.908      0.075      -0.168       3.183
==============================================================================
Omnibus:                        0.451   Durbin-Watson:                   1.204
Prob(Omnibus):                  0.798   Jarque-Bera (JB):                0.484
Skew:                          -0.312   Prob(JB):                        0.785
Kurtosis:                       2.493   Cond. No.                         22.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

[22]:

```
plt.scatter(x1,y)
yhat = 1.5077*x1 + 56.8431 #These values are from Regression Table. 27.700 is␣
↪Year coef and 335.1 is constant.
fig = plt.plot(x1,yhat, lw=4, c='orange', label = 'regression line')
plt.xlabel('Year', fontsize = 20)
plt.ylabel('Population', fontsize = 20)
plt.xlim(0)
plt.ylim(0)
plt.show()
```



Observations

The first two dots in the graph above represents the initial years of the conference. Therefore, since the conference was new, it makes sense to have low attendance. These two years of data should be treated as outliers and be removed from the analysis. In further analysis, we will look at the sudden jump in population after 2 initial years.

Data Analysis after removing the outliers

[20]:
```
import matplotlib.pyplot as plt
data1 = pd.read_csv("population_data_modified.csv")
y1 = data1['Population']
x2 = data1['Year']
x = sm.add_constant(x2)
results = sm.OLS(y1,x).fit()
results.summary()
```

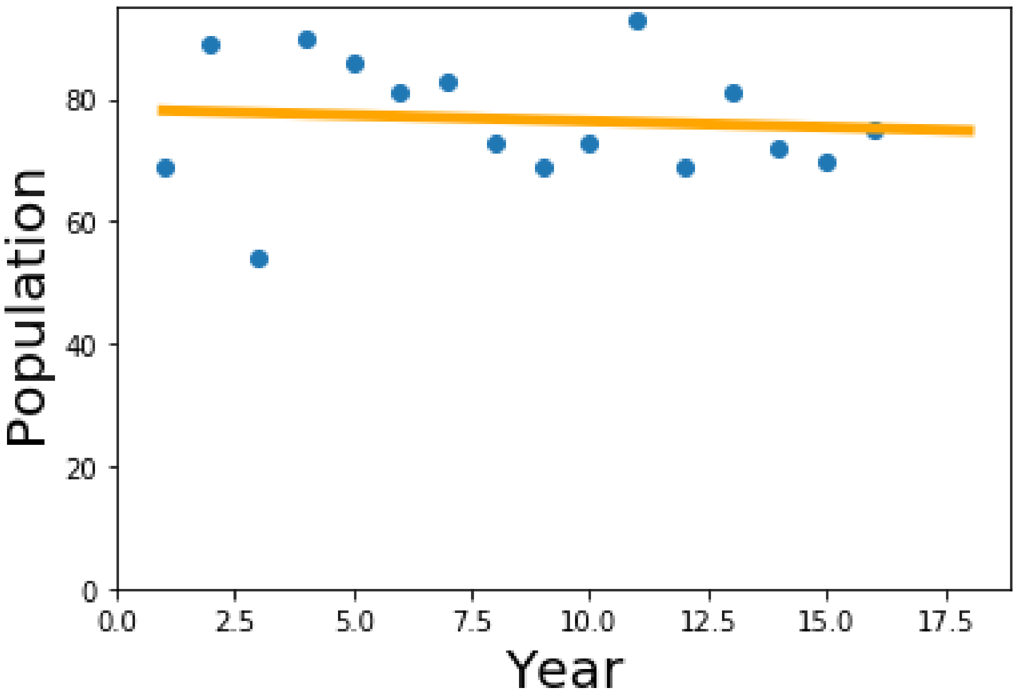
```
/home/akki/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1535:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=16
  "anyway, n=%i" % int(n))
```

[20]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:            Population   R-squared:                       0.008
Model:                           OLS   Adj. R-squared:                 -0.062
Method:                Least Squares   F-statistic:                    0.1187
Date:               Sun, 26 Jul 2020   Prob (F-statistic):              0.736
Time:                       23:01:30   Log-Likelihood:                -59.210
No. Observations:                 16   AIC:                             122.4
Df Residuals:                     14   BIC:                             124.0
Df Model:                          1
Covariance Type:           nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         78.3500      5.490     14.271      0.000      66.575      90.125
Year          -0.1956      0.568     -0.344      0.736      -1.413       1.022
==============================================================================
Omnibus:                       1.363   Durbin-Watson:                   2.822
Prob(Omnibus):                 0.506   Jarque-Bera (JB):                0.464
Skew:                         -0.413   Prob(JB):                        0.793
Kurtosis:                      3.121   Cond. No.                         20.5
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

[21]:

```
plt.scatter(x2,y1)
plt.xlabel('Year', fontsize = 20)
plt.ylabel('Population', fontsize = 20)
yhat = -0.1956*x1 + 78.35 #These values are from Regression Table. -0.1956 is
 ↪Year coef and 78.35 is constant.
fig = plt.plot(x1,yhat, lw=4, c='orange', label = 'regression line')
plt.xlim(0)
plt.ylim(0)
plt.show()
```



Analysis

The P-value for year = 0.736 (> 0.05) suggests that there is no significant impact of change in time over the population of the conference. Also, the P-value = 0.000 for population suggests that this result is definitely not by chance. The B-value(co-effecient of x) is -0.1956. This is incositent with our Null Hypothesis i.e. B = 0. However, it is important to note that the difference between 0 and -0.1956 is very small and it is difficult to make a strong conclusion about the Null Hypothesis.

Evidence: The yellow line in the graph suggests that the population remained similar over the years. To support this claim, an additional research was performed. Acording to the observations, it was noticed that the location of the conference had a strong influence over the population of the conference. In the initial years, the conference was held at different locations. However, after initial years, the conference has been held at the same location from past several years. This explains the low attendance in the initial years and constant after that. This also suggests that the organisers invited a limited number of people as per their capacity and the data trends of this conference cannot be used to generalise the trend in the industry as a whole. Therefore, we fail to reject the Null Hypothesis.

Furthermore, we cannot make any strong conclusions about our initial hypothesis i.e. the population of the conference increases over the years.

Hypothesis: The gender composition is changing over the years.

I decided to test this hypothesis because it is very crucial as women in tech are usually under-represented. There is an uneven split between number of males and females attending the conferences. It is important to analyse and study these data, so the participation of under-represented gender can be encouraged more. Even though more and more women are being encouraged and welcomed in the tech field, I still feel they are under-represented. In order to determine the gender of the participants, I used NamSor tool which tells me about the likely gender of each participant.

Exploring the Data

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# data to plot
# Read gender data from the CSV file.
gender_data = pd.read_csv("gender_data.csv")
gender_data
# Year 1 corresponds to '2002' and similarly other index values map to␣
 ↪consective years untill 2019 i.e. 18.
```

[1]:

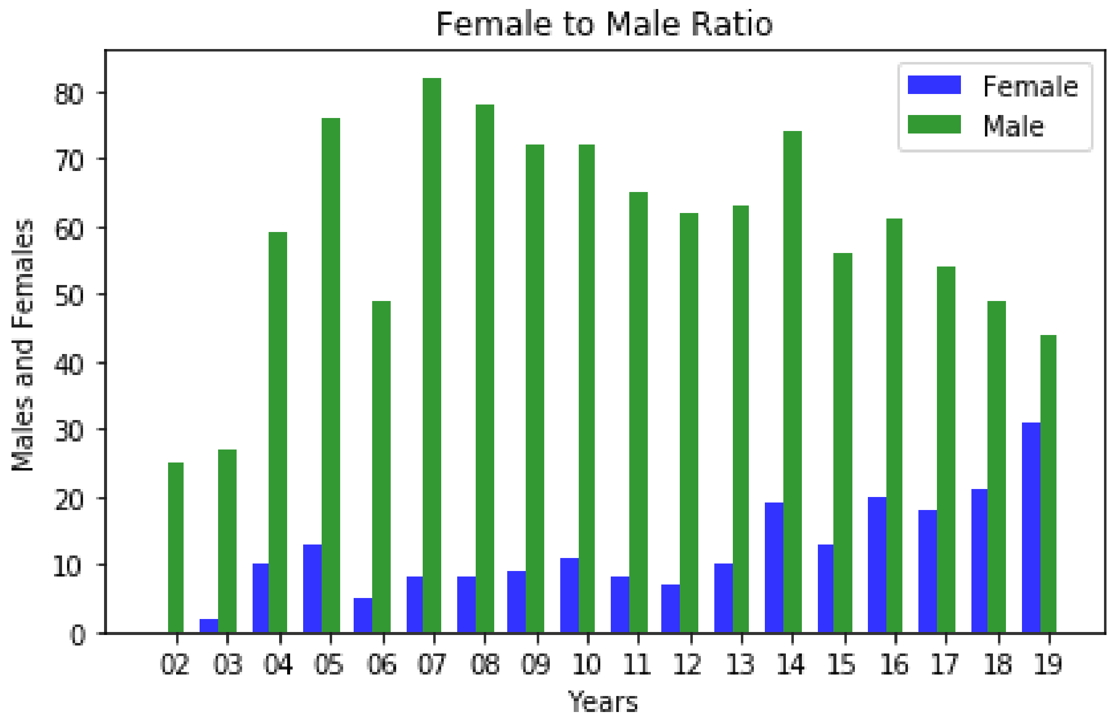
	Year	Male	Female	Ratio
0	1	25	0	0.000000
1	2	27	2	0.068966
2	3	59	10	0.144928
3	4	76	13	0.146067
4	5	49	5	0.092593
5	6	82	8	0.088889
6	7	78	8	0.093023
7	8	72	9	0.111111
8	9	72	11	0.132530
9	10	65	8	0.109589
10	11	62	7	0.101449
11	12	63	10	0.136986
12	13	74	19	0.204301
13	14	56	13	0.188406
14	15	61	20	0.246914
15	16	54	18	0.250000
16	17	49	21	0.300000
17	18	44	31	0.413333

[3]:
```python
gender_data.describe()
```

[3]:

	Year	Male	Female	Ratio
count	18.000000	18.000000	18.000000	18.000000
mean	9.500000	59.333333	11.833333	0.157171
std	5.338539	16.168233	7.578996	0.097043
min	1.000000	25.000000	0.000000	0.000000
25%	5.250000	50.250000	8.000000	0.095130
50%	9.500000	61.500000	10.000000	0.134758
75%	13.750000	72.000000	16.750000	0.200327
max	18.000000	82.000000	31.000000	0.413333

[4]:
```
x = gender_data["Year"]
y1 = gender_data["Male"]
y2 = gender_data["Female"]

# create a plot
fig, ax = plt.subplots()
index = np.arange(18)
bar_width = 0.35
opacity = 0.8

rects1 = plt.bar(index, y2, bar_width,
alpha=opacity,
color='b',
label='Female')

rects2 = plt.bar(index + bar_width, y1, bar_width,
alpha=opacity,
color='g',
label='Male')

plt.xlabel('Years')
plt.ylabel('Males and Females')
plt.title('Female to Male Ratio')
plt.xticks(index + bar_width,
           ('02', '03', '04',␣
 ↪'05','06','07','08','09','10','11','12','13','14','15','16','17','18','19'))
plt.legend()

plt.tight_layout()
plt.show()
```



Observations It is important to note that I judged the accuracy of the results of NamSor tool based on two evidences. First, I performed a manual analysis of the known names whose genders I am aware of. It matched all the actual genders. Secondly, most of the names have a confidence probability of over 0.89 which is very high. However, some of the names were 'pseudo-names' like 'GMUNK' which might be more famous in the industry. I highly doubt the accuracy of NamSor API on such names. However, the number of such names is very small(3-5). Therefore, it won't have a significant impact on the data.

Also, the NamSor tool only segregates a name by either male or female. It does not account for the people who identify themselves as LGBTQ communities. The conference is based in Toronto. According to Statistics Canada(https://www.statcan.gc.ca/eng/dai/smr08/2015/smr08_203_2015#a3), less than 2% Canadian population identified themselves belonging to the LGBTQ community. Therefore, it is less likely that this bias in data will have a significant impact on the overall trend of the data.

Another important observation is that, in the year 2002, there were no women at the conference. The representation was very uneven.

Looking at the bar graph, we can clearly see that there is an uneven split in the men-women representation in this conference. However, we need to verify our claim statistically.

To verify my claim, I will be using Linear Regression analysis. To model the data, we will be taking the ratio of females to the total population.

Therefore, we need a Null Hypothesis and the Alternative Hypothesis.

Null Hypothesis: The female to total ratio remains the same over the years, i.e. B(beta) value which is co-efficient of x in linear regression remains 0.

Alternative Hypothesis: The female to total ratio increases over the years i.e. B > 0

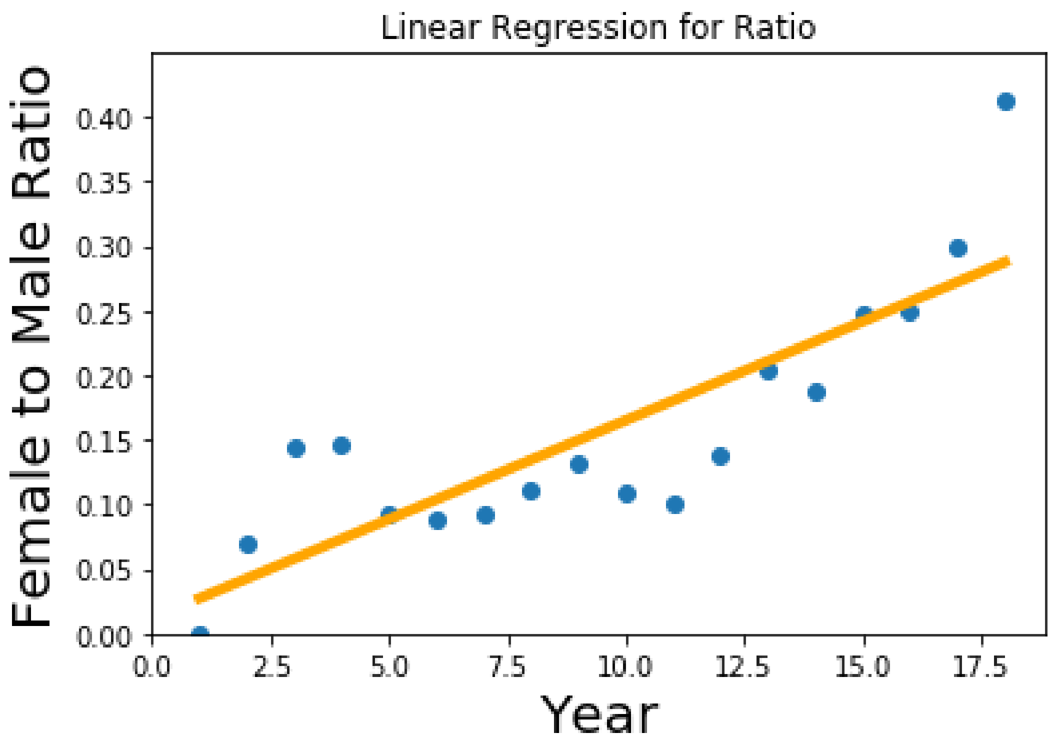
Linear Regression

[37]:
```
import statsmodels.api as sm
gender_data = pd.read_csv("gender_data.csv")
y3 = gender_data['Ratio']
x1 = sm.add_constant(x)
results = sm.OLS(y3,x1).fit()
results.summary()
```

```
/home/akki/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1535:
UserWarning: kurtosistest only valid for n>=20 … continuing anyway, n=18
  "anyway, n=%i" % int(n))
```

[37]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Ratio   R-squared:                       0.707
Model:                            OLS   Adj. R-squared:                  0.688
Method:                 Least Squares   F-statistic:                     38.51
Date:                Mon, 27 Jul 2020   Prob (F-statistic):           1.26e-05
Time:                        21:21:07   Log-Likelihood:                 27.993
No. Observations:                  18   AIC:                            -51.99
Df Residuals:                      16   BIC:                            -50.21
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0120      0.027      0.451      0.658      -0.044       0.069
Year           0.0153      0.002      6.206      0.000       0.010       0.020
==============================================================================
Omnibus:                        4.180   Durbin-Watson:                   0.659
Prob(Omnibus):                  0.124   Jarque-Bera (JB):                2.443
Skew:                           0.891   Prob(JB):                        0.295
Kurtosis:                       3.292   Cond. No.                         22.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

[39]:
```
plt.scatter(x,y3)
yhat = +0.0153*x + 0.0120 #These values are from Regression Table. 0.0153 is␣
↪Year coef and 0.0120 is constant.
fig = plt.plot(x,yhat, lw=4, c='orange', label = 'regression line')
plt.xlabel('Year', fontsize = 20)
plt.ylabel('Female to Total Ratio', fontsize = 20)
plt.xlim(0)
plt.ylim(0)
plt.title('Linear Regression')
plt.show()
```



Result and Summary We can see from the regression table and our graph that the B value(coefficient of x) is non-zero. Also, the p-value is 0.000 which suggests it is statistically significant and there is very weak evidence for the null hypothesis. It also suggests that this result is not by-chance. Therefore, our Null-Hypothesis is not true. This implies that the female to total ratio is not constant over the years.

Therefore, our Alternative Hypothesis is correct. Actually, the value of $B = 0.153 > 0$. This is really encouraging as it indicates that the participation of women is increasing over the years and heading towards the even split in gender ratios. Therefore, the regression analysis supports our original hypothesis that female to total population ratio has improved over the years.

0.0.3 Hypothesis 3: The race ethnicity is unevenly distributed in the Big Data conferences.

In any conference, there are people who come from different cultural backgrounds, different countries, and different races of ethnicity. This is an important factor such as diversity that leads to different perspectives at a conference. A dominant group also tends to lead the direction in which the future of the field goes. Therefore, I chose this hypothesis to further analyse the participation of various groups in the conference.

We used the NameSor tool to determine the race-ethnicity of the participants. The data is based on the race-ethnicity of the US. The US is full of people and immigrants from various cultures and race should be a good data set to determine the likely ethnicity of the participants.

Data Representation

[5]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# data to plot
# Read ethnicity data from the CSV file.
ethnicity_data = pd.read_csv("ethnicity_data.csv")
ethnicity_data
```

[5]:

	Year	Asian	Black, Non Latino	Hispano Latino	White, Non Latino
0	1	4	4	0	17
1	2	1	7	0	21
2	3	8	12	2	47
3	4	9	14	6	60
4	5	6	10	2	36
5	6	11	10	5	63
6	7	8	14	8	56
7	8	14	4	6	57
8	9	10	11	5	57
9	10	13	11	3	46
10	11	11	13	6	39
11	12	15	6	8	44
12	13	13	16	7	57
13	14	15	10	5	39
14	15	16	14	6	45
15	16	16	9	6	41
16	17	15	7	5	43
17	18	18	11	1	45

[8]:
```python
ethnicity_data.describe()
```

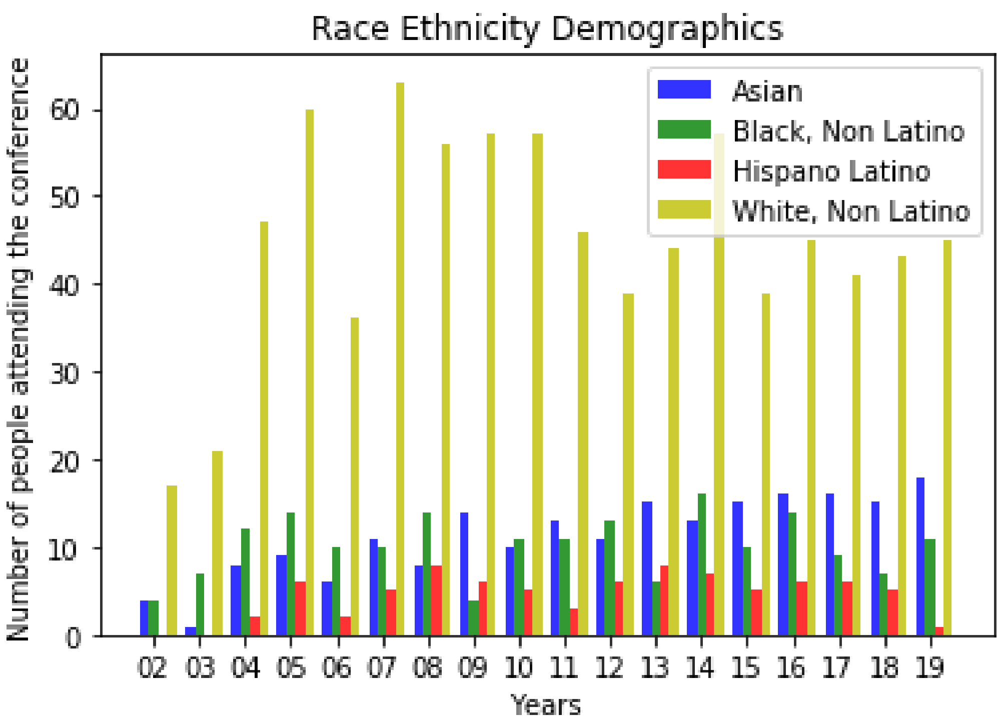
[8]:

	Year	Asian	Black, Non Latino	Hispano Latino	\
count	18.000000	18.000000	18.000000	18.00000	
mean	9.500000	11.277778	10.166667	4.50000	

```
std      5.338539   4.599304            3.485263          2.54951
min      1.000000   1.000000            4.000000          0.00000
25%      5.250000   8.250000            7.500000          2.25000
50%      9.500000  12.000000           10.500000          5.00000
75%     13.750000  15.000000           12.750000          6.00000
max     18.000000  18.000000           16.000000          8.00000

       White, Non Latino
count          18.000000
mean           45.166667
std            12.462886
min            17.000000
25%            39.500000
50%            45.000000
75%            56.750000
max            63.000000
```

[12]:
```
# Year 1 corresponds to '2002' and similarly other index values map to␣
 ↪consective years untill 2019 i.e. 18.
x = ethnicity_data["Year"]
y1 = ethnicity_data["Asian"]
y2 = ethnicity_data["Black, Non Latino"]
y3 = ethnicity_data["Hispano Latino"]
y4 = ethnicity_data["White, Non Latino"]


# create a plot
fig, ax = plt.subplots()
index = np.arange(18)
bar_width = 0.2
opacity = 0.8

rects1 = plt.bar(index, y1, bar_width,
alpha=opacity,
color='b',
label='Asian')

rects2 = plt.bar(index + bar_width, y2, bar_width,
alpha=opacity,
color='g',
label='Black, Non Latino')

rects3 = plt.bar(index + 2*bar_width, y3, bar_width,
alpha=opacity,
color='r',
label='Hispano Latino')
```

```
rects2 = plt.bar(index + 3*bar_width, y4, bar_width,
alpha=opacity,
color='y',
label='White, Non Latino')

plt.xlabel('Years')
plt.ylabel('Number of people attending the conference')
plt.title('Race Ethnicity Demographics')
plt.xticks(index + bar_width,
           ('02', '03', '04',␣
 ↪'05','06','07','08','09','10','11','12','13','14','15','16','17','18','19'))
plt.legend()

# plt.tight_layout()
plt.show()
```



Looking at the bar, graph we can clearly see the race-ethnicity distribution is not equal in this conference. However, we need to verify our claim statisticly.

To verify my claim statistically, I will be using ANOVA F-test (Analysis of Variance).

Therefore, we need a Null Hypothesis and Alternative Hypothesis.

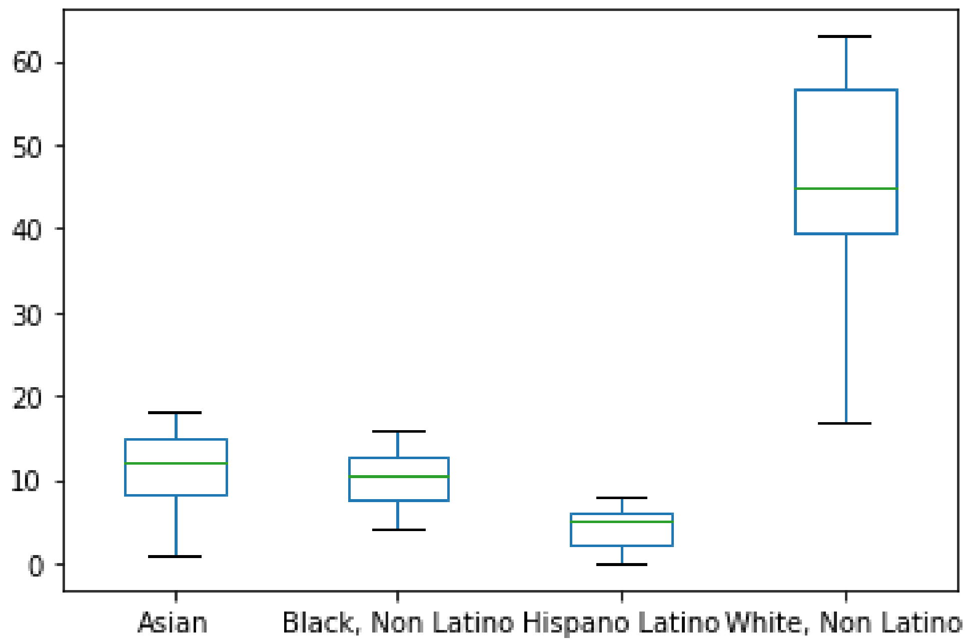
Null Hypothesis: The race ethnicity distribution is equal i.e. mean of each race group is equal.

Alternative Hypothesis: Atleast one race group's mean is different.

ANOVA F-Test

[13]:
```
# generate a boxplot to see the data distribution by race. Using boxplot, we␣
↪can easily detect the differences
# between different race ethnicity.
ethnicity_data.boxplot(column=['Asian', 'Black, Non Latino', 'Hispano Latino',␣
↪'White, Non Latino'], grid=False)
```

[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc5e6b47410>
```



[10]:
```
# load packages
import scipy.stats as stats
# stats f_oneway functions takes the groups as input and returns F and P-value
fvalue, pvalue = stats.f_oneway(ethnicity_data['Asian'], ethnicity_data['Black,␣
↪Non Latino'],
                                 ethnicity_data['Hispano Latino'],␣
↪ethnicity_data['White, Non Latino'])
print(fvalue, pvalue)
```

```
126.27400013398547 9.748241731748067e-28
```

Observations and Results

Here we can clearly see that the result of pvalue = 9.748241731748067e-28 ($P < 0.05$). This implies that there is significant difference in the means of the all the race groups. According to the box-plot graph, we can see that highest representation is by White, Non Latino community. The second highest is Asians, followed by Black and Hispano Latinos. This representation of ethnic groups has been relatively consistent over past 17 years. Since this conference is held in Toronto, I compared the above data with ethnic demographics of Toronto. (https://www.thestar.com/news/gta/2018/09/30/toronto-is-segregated-by-race-and-income-and-the-numbers-are-ugly.html). The article displays that Toronto has similar ethnic diversity when compared with this conference.

Therefore, our null hypothesis is rejected. Our alternative hypothesis that atleast one group has different mean is accepted. Observing the graphs and other articles, I can clearly see White community dominating this space. The same observation was supported by our analysis. Therefore, our original hypothesis that race ethnicity is unevenly distributed as been verified.

Summary In our report, we tested various hypotheses to determine the diversity in the participation of people in the Design and Tech industry conference. One of the interesting points to note in this analysis is that participation in the conference is highly affected by the location i.e. Toronto. A similar trend was seen in all the hypotheses. The number of people invited is limited and often the same people are invited over the years. The representation of both genders is moving towards the desired goal of equal split every year. The ethnic demographics are unevenly distributed. I feel that the conference is limited to local speakers. This also explains the unpopularity of the conference internationally. Even though this analysis displays many interesting trends, the data presented cannot be used to generalize similar participation in the overall field of Design and Technology.