Using Colab to Analyze Data

Georgia Martin

Doing It Yourself: Setup

Setting Up Google Colab

Open google colab and open a new notebook

Add this code into a different block and click run:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

This will import different “libraries” which adds tons of different things you can do (like build graphs, edit data sets, etc.) that colab can’t do on its own

Setting Up Google Colab

All of these calculations were used via google colab. If you would like to do your own calculations on this data set, follow the instructions below:

Download the data set to your google account here and make a folder named “College Info” and put the data set in it

Copy & paste this and click “Run”, you should see the first few rows of the data set

```
from google.colab import drive
drive.mount('/content/drive')
wage_info = pd.read_csv('/content/drive/My Drive/College Info/July_Union_Roster.csv')
wage_info['Postal Code'] = wage_info['Postal Code'].astype(str)
wage_info.head()
```

Sign into your google account and allow all permissions

If you have any trouble, here is a video that is relatively use friendly that explains how to do this

Wage Gaps

Key Hourly Wage Gaps

$4.30, 81.19%

GENDER

There is a $4.30 gap between average hourly wages of male and females, which is about a 81.19% of the male average hourly wage.

$4.73, 80.34%

ETHNICITY

There is a $4.37 gap between hourly wages of white people and all other races, which is about 80.34% of the average hourly wages of white people.

$3.41, 83.71%

WORK STATUS

There is a $3.41 gap between hourly wages full and part time workers, which is about 83.71% of full time workers average hourly wage.

More Specific Hourly Wage Gaps

$0.88, 95.43%

RACE & GENDER - FEMALE

There is a $0.88 gap between average hourly wages of white females and black females, which is about a 95.43% of the white female average hourly wage.

$5.70, 77.96%

RACE & GENDER - MALE

There is a $3.41 gap between hourly wages full and part time workers, which is about 83.71% of full time workers average hourly wage.

Doing It Yourself: Wage Gaps

To Find More Gaps - Copy & Paste

Single Variable

```
def wage_av_calc(Title, Specific):
  total = 0.0
  num = 0
  for i in range(len(wage_info.index)):
    if wage_info[Title][i] == Specific:
      total = total + (wage_info['Hourly Rate'][i])
      num = num + 1
  average = total / num
  return average
def wage_gap_calc(Title,Specific1,Specific2):
  av1 = wage_av_calc(Title,Specific1)
  av2 = wage_av_calc(Title,Specific2)
  gap = av1 - av2
  print("the", Specific1,"group makes",gap,"more
than", Specific2)
  print(Specific1,"group average:", av1)
  print(Specific2,"group average:", av2)
```

Two Variables

```
def
wage_av_calc_two(Title1,Specific1,Title2,Specific2)
:
  total = 0.0
  num = 0
  for i in range(len(wage_info.index)):
    if wage_info[Title1][i] == Specific1 and
wage_info[Title2][i] == Specific2:
      total = total + (wage_info['Hourly Rate'][i])
      num = num + 1
  average = total / num
  return average
def
wage_gap_calc_mult(Title1,Specific1,Title2,Specific
2,Title1A,Specific1A,Title2A,Specific2A):
  av1 =
wage_av_calc_two(Title1,Specific1,Title2,Specific2)
  av2 =
wage_av_calc_two(Title1A,Specific1A,Title2A,Specifi
```

To Find More Gaps - Test

Single Variable:

In a new code block under after running the code on the previous slide, type:

```
wage_gap_calc("Full/Part Time","Full time","Part time")
```

If the output is:

```
the Full time group makes 3.413607594936572 more than Part time Full time group average: 20.951107594936573 Part time group average: 17.5375
```

You have set it up correctly

Double Variable:

In a new code block under after running the code on the previous slide, type:

```
wage_gap_calc_mult("Person Ethnicity","Black or African American","Person Gender","Female","Person Ethnicity","White","Person Gender","Female")
```

If the output is:

```
Black or African American and Female group average: 18.362500000000054 White and Female group average: 19.24067796610169 the Black or African American and Female group makes -0.8781779661016351 more than White and Female group
```

You have set it up correctly

To Find More Gaps - Use

Single Variable:

In the parentheses, there are three phrases in quotations. The first one will give the category of the desired gap. The next two phrases give the two groups you would like to compare.

You must spell the groups exactly how they are inserted in the dataset or it will not work (see the document)

Examples:
Gender wage gap: wage_gap_calc("Person Gender","Female","Male")

Racial wage gap: wage_gap_calc("Person Ethnicity","Black of African American","White")

Double Variable:

There are eight phrases this time (a lot!), but it can make things very specific. The first four pertain to the first group. The order goes: Category, Label- and for each person you give two different categories and labels.

For example:
Group A are female senior cooks
Group B are male custodians

Code:
wage_gap_calc_mult("Job","Senior Cook","Person Gender","Female","Job","Custodian","Person Gender","Male")

Important Graphs

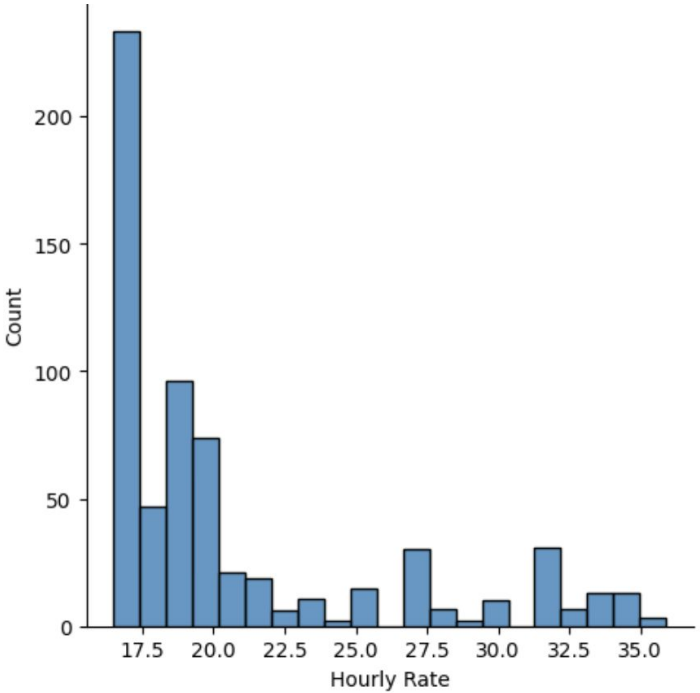
Graph of all Hourly Rates

Most Common Pay

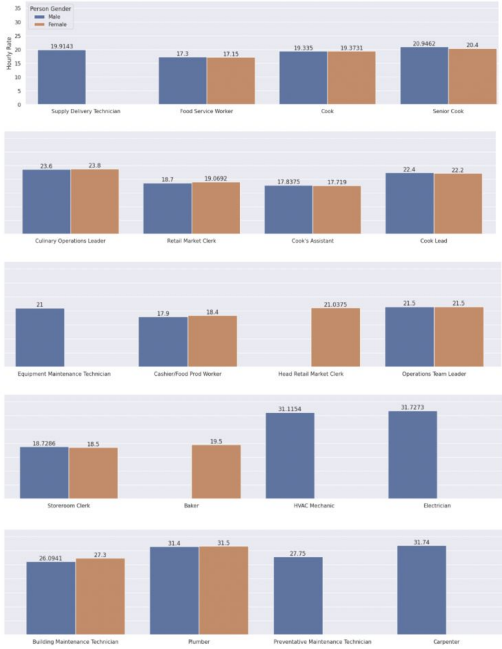
As seen by the graph, the most common hourly wage is less than $17.5

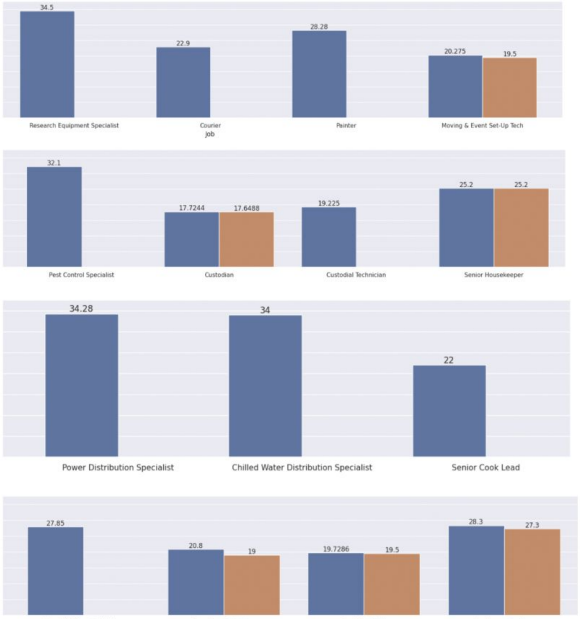
There will not be a DIY on how to make this graph, given that this is the only true use of it. However, the code for it is below if you would like to try for yourself.

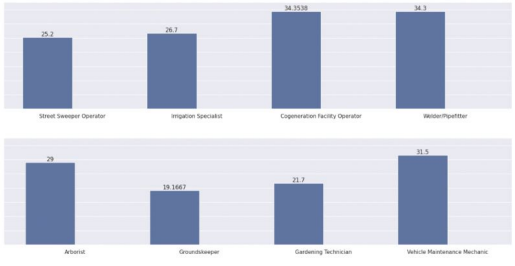
```
sns.displot(data=wage_info,
x="Hourly_Rate")
```

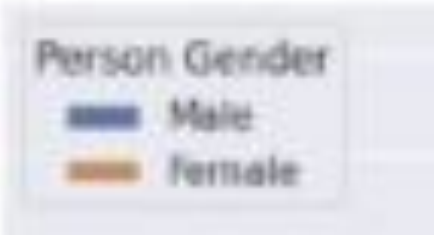


Job Hourly Wages Separated by Gender

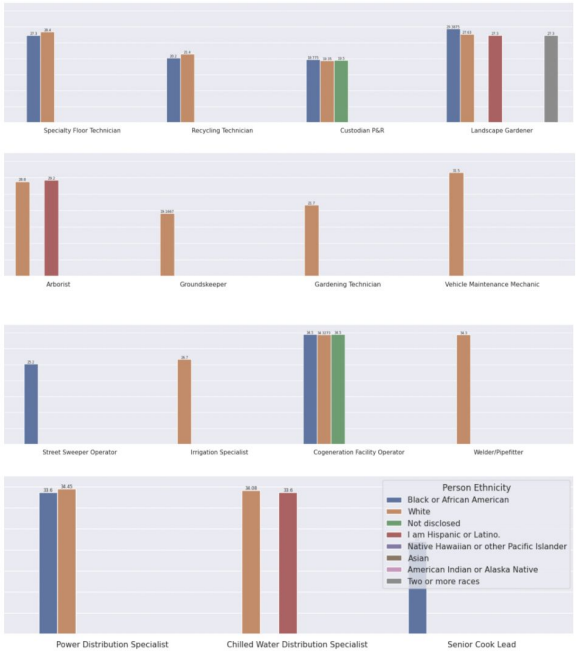


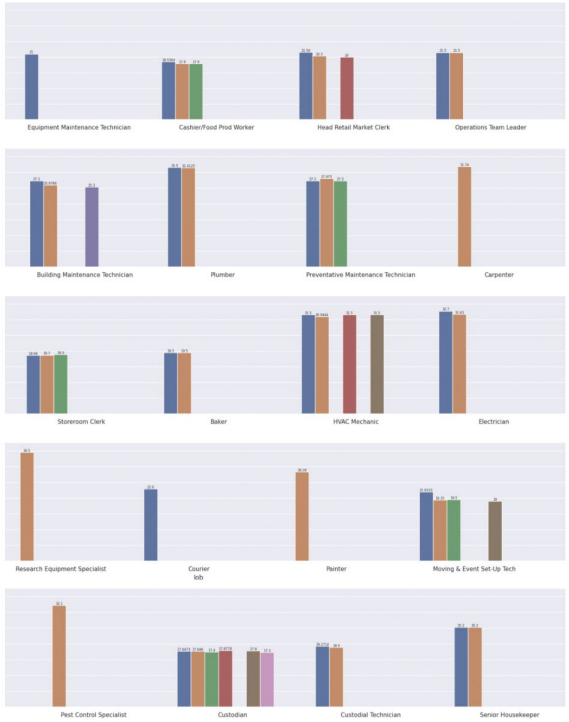


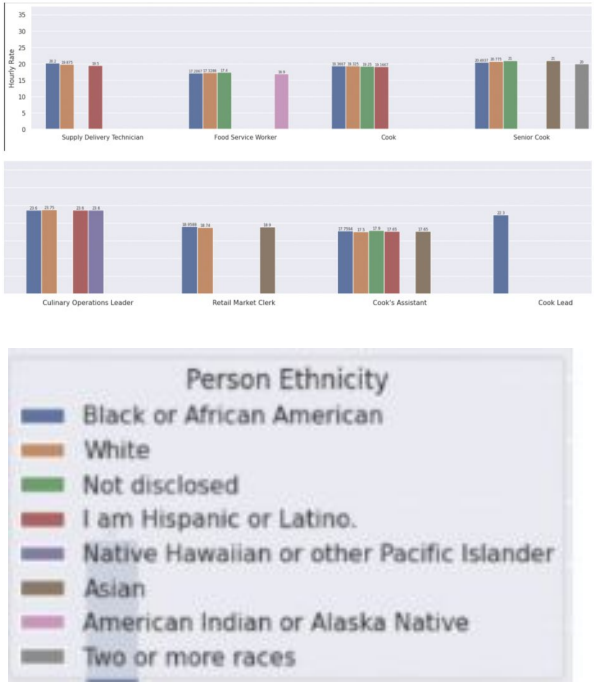




Job Hourly Wages Seperated by Race







DIY: Comparison Graphs

Graphing

Start by pasting this into a different code block and running it (run BEFORE trying to make any graphs):

```
sns.set(rc={"figure.figsize":(200, 4)})
```

If your graphs look squished together, come back to this and mess around with the 200 and the 4, which are x and y limitations to the graphs appearance. But first try clicking the graph because it might expand.

Here is an example, everything in uppercase is to be filled in with a desired field

```
NAME OF GRAPH = sns.barplot(data=wage_info, x="COLUMN OF COMPARISON ON X AXIS", y="COLUMN OF COMPARISON ON Y AXIS",hue="FACTOR TO DIVVY UP EACH X-VALUE", errwidth = 0)
for i in NAME OF GRAPH.containers:
  NAME OF GRAPH.bar_label(i,)
plt.show()
```

Note: The name of your graph can be anything, but do not use spaces or quotations!!
The “hue” variable is completely optional if you want a more simple graph.

Graphing

Here is the code from the two graphs seen previously that you can use as a guide as well:

Gender

```
gender = sns.barplot(data=wage_info, x="Job", y="Hourly Rate", hue="Person Gender", errwidth = 0)
for i in gender.containers:
gender.bar_label(i,)
plt.show()
```

Race

```
ethnicity = sns.barplot(data=wage_info, x="Job", y="Hourly Rate", hue="Person Ethnicity",errwidth = 0)
for i in ethnicity.containers:
ethnicity.bar_label(i,)
plt.show()
```

What if Gender is Not Disclosed on a Future Dataset?

My Algorithm

Here is a link to pre-written code. There are instructions on how to change the code to fit the desired dataset.

This dataset uses BERT (an advanced NLP tool) to determine gender. It will be patent pending soon.

How Accurate is it?

After testing several datasets, the accuracy was above 80% for all of them. For example, the accuracy for this dataset was 87.57%.

Though the accuracy is not 100%, this tool can be used to show indications of gender pay disparity.

Column & Names for Use, Closer Look at Graphs

This document contains the exact names of columns and their values that could be used for analysis.

This document contains the graphs shown in this slideshow but zoomed in.


slidesgo