```
# Names have been change to protect the identity of individuals and the client.
```

```
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
     Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd
import numpy as np

# all employees information with the exception of William P.
file_path = '/content/drive/MyDrive/Client_Final_Recipient.xlsx'

# William P. information
file_path2 = '/content/drive/MyDrive/RecipientsExport_Client employee.xlsx'

# importing the demographics without William P. information

file_path3 = '/content/drive/MyDrive/Client_Demographic_Survey.xlsx'

# William P. demographic information

file_path4 ='/content/drive/MyDrive/Client employee info.xlsx'
```

```
#viewing [insert name]. information

wlp = pd.read_excel(file_path4)

# named the dataframe demographics
demographic = pd.read_excel(file_path3)

#named the dataframe demographics
Recepient = pd.read_excel(file_path)

Recepient.head()

#loading Williams separate file
wp = pd.read_excel(file_path2)

wp
```

Show hidden output

```
Recepient.head()
```

Show hidden output

```
# This is where I am dropping William from the original dataset
drop_will = Recepient[(Recepient["First Name"] == 'William') & (Recepient["Last Name"] == 'Packer')].index

Recepient = Recepient.drop(drop_will)

Recepient
```

Show hidden output

```
N_Recepient = pd.merge(Recepient, wp, on=['Email', 'First Name', 'Last Name', 'Sent', 'Responded', 'Custom 1', 'Custom 2', 'Custom 3', 'Custom 4', 'Custom 5', 'Custom 6'], how='ou

N_Recepient
```

Show hidden output

```
# This is where I am dropping Nancy from the original dataset
drop_nancy = N_Recepient[(N_Recepient["First Name"] == 'Nancy') & (N_Recepient["Last Name"] == 'Martinez')].index

N_Recepient = N_Recepient.drop(drop_nancy)

N_Recepient
```

Show hidden output

```
# This is where I am dropping Betty from the original dataset
drop_betty = N_Recepient[(N_Recepient["First Name"] == 'Betty') & (N_Recepient["Last Name"] == 'Rodriguez')].index

N_Recepient = N_Recepient.drop(drop_betty)

N_Recepient
```

Show hidden output

```
N_Recepient['Leader'] = np.where(N_Recepient['Custom 1']== 'President', 'Yes','No')
N_Recepient
```

Show hidden output

```
# Created columns for leader, board, Client board 2, Client Board 1, senior staff, and staff members
N_Recepient['Board'] = np.where(N_Recepient['Custom 1'].isin(['Board', 'Client Board 2']), 'Yes', 'No')
N_Recepient['Senior Staff'] = np.where(N_Recepient['Custom 1'].isin(['Leadership Team', 'President']), 'Yes', 'No')
N_Recepient['Staff'] = np.where(N_Recepient['Custom 1'].isin(['Leadership Team', 'President', 'Other Staff']), 'Yes', 'No')
N_Recepient['Client Board 2'] = np.where(N_Recepient['Custom 1'].isin(['Client Board 2']), 'Yes', 'No')
N_Recepient['Client Board 1'] = np.where(N_Recepient['Custom 1'].isin(['Board']), 'Yes', 'No')

N_Recepient.head()
```

Show hidden output

```
#Created a list of columns that I want to drop
drop_columns = ['Custom 2', 'Custom 3','Custom 4','Custom 5','Custom 6']


#Dropped the unneccessary columns
N_Recepient = N_Recepient.drop(columns=drop_columns)


# there are alot of columns to merge on so I created a list of columns and put it in a df.
merge_columns = ['Email Address', 'First Name', 'Last Name','Custom Data 1', 'Race & Ethnicity How do you publicly self-identify?',
                  'Asian/Asian American IdentityHow do you publicly self-identify?',
                  'Gender IdentityHow do you publicly self-identify?',
                 'Transgender IdentityHow do you publicly self-identify?',
                 'Sexual OrientationHow do you publicly self-identify?',
                 'Disability StatusHow do you publicly self-identify?',
                 'Survey FeedbackThank you for completing this survey. If you have any feedback on your experience completing this survey, please write it in the space below. If y
```

```
# merging the demographics with William P. Information
n_demographic = pd.merge(demographic, wlp, on=merge_columns, how='outer')

n_demographic
```

Show hidden output

```
n_demographic = n_demographic.drop(0)
n_demographic
```

Show hidden output

```
# This is where I am dropping Betty from the original dataset
drop_betty2 = n_demographic[(n_demographic["First Name"] == 'Betty') & (n_demographic["Last Name"] == 'Rodriguez')].index

n_demographic = n_demographic.drop(drop_betty2)

n_demographic
```

Show hidden output

```
# creating a list of unneccessary columns to drop
dropit = ['Respondent ID_x', 'Collector ID_x','Start Date_x',   'End Date_x',   'IP Address_x','Respondent ID_y','Collector ID_y',  'Start Date_y', 'End Date_y',   'IP Address_y'


# I am going to drop the unneccessary columns which is IP address, collector id, respondent id, etc.

n_demographic = n_demographic.drop(columns=dropit)

n_demographic
```

Show hidden output

```
# I am going to rename some of the columns so it can be easier for me to code

n_demographic = n_demographic.rename(columns={'Race & Ethnicity How do you publicly self-identify?': 'Race/Ethnicity','Asian/Asian American IdentityHow do you publicly self-identi

n_demographic.head()
```

Show hidden output

```
N_Recepient.head()
```

Show hidden output

```
# Look over and edit the code down below


# Merging both the Recepient and demographic df into one.
Client = pd.merge(N_Recepient, n_demographic, how='outer', on=('First Name', 'Last Name'))

Client.head()
```

Show hidden output

```
Client.columns.tolist()


# Drop the specified columns from 'Client'
N_Client = Client.drop(columns=['Email Address', 'Custom Data 1'])

N_Client.head()
```

Show hidden output

```
# Drop duplicate rows
N_Client_deduplicated = N_Client.drop_duplicates()

# Output the DataFrame after dropping duplicates
print(N_Client_deduplicated)
```

Show hidden output

```
# Check for duplicates based on specific columns
duplicates = N_Client.duplicated(subset=['First Name', 'Last Name'])

# Output the rows with duplicates
print(N_Client[duplicates])
```

Show hidden output

```
Final_Client = N_Client.drop_duplicates(subset=['First Name', 'Last Name'])

print(Final_Client)
```

Show hidden output

```
#Below is the demographic breakdown of the leaders at Client
```

```
#follow this code pattern to get demographic breakdown.
Client_Leader = Final_Client[Final_Client['Leader'] == 'Yes']
Client_Leader_counts_race = Client_Leader['Race/Ethnicity'].value_counts()
Client_Leaderpct_race = Client_Leader['Race/Ethnicity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Leader_counts_race)
print(Client_Leaderpct_race)
```

Show hidden output

```
Client_Leader_counts_trans = Client_Leader['Transgender'].value_counts()
Client_Leaderpct_trans = Client_Leader['Transgender'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Leader_counts_trans)
print(Client_Leaderpct_trans)
```

Show hidden output

```
Client_Leader_counts_ds = Client_Leader['Disability'].value_counts()
Client_Leaderpct_ds = Client_Leader['Disability'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Leader_counts_ds)
print(Client_Leaderpct_ds)
```

Show hidden output

```
Client_Leader_counts_so = Client_Leader['Sexual Orientation'].value_counts()
Client_Leaderpct_so = Client_Leader['Sexual Orientation'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Leader_counts_so)
print(Client_Leaderpct_so)
```

Show hidden output

```
Client_Leader_counts_gender = Client_Leader['Gender Identity'].value_counts()
Client_Leaderpct_gender = Client_Leader['Gender Identity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Leader_counts_gender)
print(Client_Leaderpct_gender)
```

Show hidden output

```
# Below breakdown of Board members demographic this includes both the Client Board 1 and Client Board 2
```

```
Final_Client[Final_Client['Board'] == 'Yes'].count()
```

Show hidden output

```
Final_Client.count()
```

Show hidden output

```
All_Board = Final_Client[Final_Client['Board'] == 'Yes']
All_Board_counts_race = All_Board['Race/Ethnicity'].value_counts()
All_Boardpct_race = All_Board['Race/Ethnicity'].value_counts(normalize=True) * 100


# Display the total count
print(All_Board_counts_race)
print(All_Boardpct_race)
```

Show hidden output

```
All_Board_counts_trans = All_Board['Transgender'].value_counts()
All_Boardpct_trans = All_Board['Transgender'].value_counts(normalize=True) * 100


# Display the total count
print(All_Board_counts_trans)
print(All_Boardpct_trans)
```

Show hidden output

```
All_Board_counts_ds = All_Board['Disability'].value_counts()
All_Boardpct_ds = All_Board['Disability'].value_counts(normalize=True) * 100


# Display the total count
print(All_Board_counts_ds)
print(All_Boardpct_ds)
```

Show hidden output

```
All_Board_counts_so = All_Board['Sexual Orientation'].value_counts()
All_Boardpct_so = All_Board['Sexual Orientation'].value_counts(normalize=True) * 100


# Display the total count
print(All_Board_counts_so)
print(All_Boardpct_so)
```

Show hidden output

```
All_Board_counts_gender = All_Board['Gender Identity'].value_counts()
All_Boardpct_gender = All_Board['Gender Identity'].value_counts(normalize=True) * 100


# Display the total count
print(All_Board_counts_gender)
print(All_Boardpct_gender)
```

Show hidden output

```
# Breakdown of Client Board 1 only excluding Client Board 2


Client_Board1 = Final_Client[Final_Client['Client Board 1'] == 'Yes']
Client_Board1_counts_race = Client_Board1['Race/Ethnicity'].value_counts()
Client_Board1pct_race = Client_Board1['Race/Ethnicity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board1_counts_race)
print(Client_Board1pct_race)
```

Show hidden output

```
Client_Board1_counts_trans = Client_Board1['Transgender'].value_counts()
Client_Board1pct_trans = Client_Board1['Transgender'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board1_counts_trans)
print(Client_Board1pct_trans)
```

Show hidden output

```
Client_Board1_counts_ds = Client_Board1['Disability'].value_counts()
Client_Board1pct_ds = Client_Board1['Disability'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board1_counts_ds)
print(Client_Board1pct_ds)
```

Show hidden output

```
Client_Board1_counts_so = Client_Board1['Sexual Orientation'].value_counts()
Client_Board1pct_so = Client_Board1['Sexual Orientation'].value_counts(normalize=True) * 100

# Display the total count
print(Client_Board1_counts_so)
print(Client_Board1pct_so)
```

Show hidden output

```
Client_Board1_counts_gender = Client_Board1['Gender Identity'].value_counts()
Client_Board1pct_gender = Client_Board1['Gender Identity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board1_counts_gender)
print(Client_Board1pct_gender)
```

Show hidden output

```
#Breakdown of the Client Board 2 excluding the Client Board 1
```

```
Client_Board2 = Final_Client[Final_Client['Client Board 2'] == 'Yes']
Client_Board2_counts_race = Client_Board2['Race/Ethnicity'].value_counts()
Client_Board2pct_race = Client_Board2['Race/Ethnicity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board2_counts_race)
print(Client_Board2pct_race)
```

Show hidden output

```
Client_Board2_counts_trans = Client_Board2['Transgender'].value_counts()
Client_Board2pct_trans = Client_Board2['Transgender'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board2_counts_trans)
print(Client_Board2pct_trans)
```

Show hidden output

```
Client_Board2_counts_ds = Client_Board2['Disability'].value_counts()
Client_Board2pct_ds = Client_Board2['Disability'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board2_counts_ds)
print(Client_Board2pct_ds)
```

Show hidden output

```
Client_Board2_counts_so = Client_Board2['Sexual Orientation'].value_counts()
Client_Board2pct_so = Client_Board2['Sexual Orientation'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board2_counts_so)
print(Client_Board2pct_so)
```

Show hidden output

```
Client_Board2_counts_gender = Client_Board2['Gender Identity'].value_counts()
Client_Board2pct_gender = Client_Board2['Gender Identity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board2_counts_gender)
print(Client_Board2pct_gender)
```

Show hidden output

```
#Breakdown of senior staff demographic


Client_srstaff = Final_Client[Final_Client['Senior Staff'] == 'Yes']
Client_srstaff_counts_race = Client_srstaff['Race/Ethnicity'].value_counts()
Client_srstaffpct_race = Client_srstaff['Race/Ethnicity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_srstaff_counts_race)
print(Client_srstaffpct_race)
```

Show hidden output

```
Client_srstaff['Sexual Orientation'] = Client_srstaff['Sexual Orientation'].fillna('Decline to state')

Client_srstaff.head()
```

Show hidden output

```
Client_srstaff['Gender Identity'] = Client_srstaff['Gender Identity'].fillna('Decline to state')

Client_srstaff.head()
```

Show hidden output

```
Client_srstaff
```

Show hidden output

```
Client_srstaff_counts_trans = Client_srstaff['Transgender'].value_counts()
Client_srstaffpct_trans = Client_srstaff['Transgender'].value_counts(normalize=True) * 100


# Display the total count
print(Client_srstaff_counts_trans)
print(Client_srstaffpct_trans)
```

Show hidden output

```
Client_srstaff_counts_so = Client_srstaff['Sexual Orientation'].value_counts()
Client_srstaffpct_so = Client_srstaff['Sexual Orientation'].value_counts(normalize=True) * 100


# Display the total count
print(Client_srstaff_counts_so)
print(Client_srstaffpct_so)
```

Show hidden output

```
Client_srstaff_counts_gender = Client_srstaff['Gender Identity'].value_counts()
Client_srstaffpct_gender = Client_srstaff['Gender Identity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_srstaff_counts_gender)
print(Client_srstaffpct_gender)
```

Show hidden output

```
Client_srstaff_counts_ds = Client_srstaff['Disability'].value_counts()
Client_srstaffpct_ds = Client_srstaff['Disability'].value_counts(normalize=True) * 100


# Display the total count
print(Client_srstaff_counts_ds)
print(Client_srstaffpct_ds)
```

Show hidden output

```
#Breakdown of staff demographics


Client_staff = Final_Client[Final_Client['Staff'] == 'Yes']
Client_staff_counts_race = Client_staff['Race/Ethnicity'].value_counts()
Client_staffpct_race = Client_staff['Race/Ethnicity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_staff_counts_race)
print(Client_staffpct_race)
```

Show hidden output

```
Client_staff_counts_gender = Client_staff['Gender Identity'].value_counts()
Client_staffpct_gender = Client_staff['Gender Identity'].value_counts(normalize=True) * 100


# Display the total count
print(Client_staff_counts_gender)
print(Client_staffpct_gender)
```

Show hidden output

```
Client_staff_counts_trans = Client_staff['Transgender'].value_counts()
Client_staffpct_trans = Client_staff['Transgender'].value_counts(normalize=True) * 100


# Display the total count
print(Client_staff_counts_trans)
print(Client_staffpct_trans)
```

Show hidden output

```
Client_staff_counts_ds = Client_staff['Disability'].value_counts()
Client_staffpct_ds = Client_staff['Disability'].value_counts(normalize=True) * 100


# Display the total count
print(Client_staff_counts_ds)
print(Client_staffpct_ds)
```

Show hidden output

```
Client_staff['Sexual Orientation'] = Client_staff['Sexual Orientation'].fillna('Decline to state')

Client_staff.head()
```

Show hidden output

```
Client_staff_counts_so = Client_staff['Sexual Orientation'].value_counts()
Client_staffpct_so = Client_staff['Sexual Orientation'].value_counts(normalize=True) * 100


# Display the total count
print(Client_staff_counts_so)
print(Client_staffpct_so)
```

Show hidden output

```
#Response rate breakdown by position


Client_Leader_counts_responded = Client_Leader['Responded'].value_counts()
Client_Leaderpct_responded = Client_Leader['Responded'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Leader_counts_responded)
print(Client_Leaderpct_responded)
```

Show hidden output

```
# All Board response rate including both Client Board 1 and Client Board 2
All_Board_counts_responded = All_Board['Responded'].value_counts()
All_Boardpct_responded = All_Board['Responded'].value_counts(normalize=True) * 100


# Display the total count
print(All_Board_counts_responded)
print(All_Boardpct_responded)
```

Show hidden output

```
# Client Board 1 response rate
Client_Board1_counts_responded = Client_Board1['Responded'].value_counts()
Client_Board1pct_responded = Client_Board1['Responded'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board1_counts_responded)
print(Client_Board1pct_responded)
```

Show hidden output

```
# AFF Board response rate
Client_Board2_counts_responded = Client_Board2['Responded'].value_counts()
Client_Board2pct_responded = Client_Board2['Responded'].value_counts(normalize=True) * 100


# Display the total count
print(Client_Board2_counts_responded)
print(Client_Board2pct_responded)
```

Show hidden output

```
# Staff response rate
Client_staff_counts_responded = Client_staff['Responded'].value_counts()
Client_staffpct_responded = Client_staff['Responded'].value_counts(normalize=True) * 100


# Display the total count
print(Client_staff_counts_responded)
print(Client_staffpct_responded)
```

Show hidden output

```
#Senior Staff Response Rate
Client_srstaff_counts_responded = Client_srstaff['Responded'].value_counts()
Client_srstaffpct_responded = Client_srstaff['Responded'].value_counts(normalize=True) * 100


# Display the total count
```