Geena Davis Institute on Gender in Media

If she can see it, she can be it.™

Final Project Report

04/05/2022

—

Sponsors

Meredith Conroy

Romeo Perez

Project Manager

Grace Yoon

Team Members

Dayong Wu (Team Leader)

Emmanouil Kritharakis

Yan Tong

Junfei Huang

Yuanli Wang

Abstract

Background

During this semester, our team worked on the **See Jane | Research Project Data Normalization** project. Our clients are Meredith Conroy and Romeo Perez,, who are members of the See Jane organization. This project aims to help the See Jane organization normalize, merge, and analyze their datasets related to the entertainment industry.

Both technical and non-technical prerequisites are needed to handle this project.

- Prior knowledge to the entertainment industry
- Sense of social responsibility of gender balance, race equality, diversity and inclusion, etc
- Comprehensive understanding of the codebooks
- Python Pandas, Numpy, Matplotlib (etc), and Excel skills are prerequisites

Motivation

- Create gender balance, foster inclusion and reduce negative stereotyping in family entertainment media.

Goal

- Normalize the datasets & refine the codebooks
- Analyze industry differences in representation in advertisements
- Summarize the change in representation in advertisements
- Detect the trends of representations in advertisements

Exploration

Before performing analysis on the datasets, our team has done a lot of data exploration work, which equips us with all necessary understandings about the datasets.

Through data exploration, we have developed a holistic comprehension of the meaning of each column and its values, different categories of the columns, the size of the dataset, etc. Below are our findings of some overarching columns:

- **Age**. Enter your best estimate of the character's age. If the character has multiple ages in the commercial, select "other" and provide details.
- **Sex**. Enter your best assessment of the character's "sex."
- **Gender**. Enter your best assessment of the character's gender performance, regardless of their sex. Masculinity refers to a set of stereotypical male traits and behavior, including assertiveness, being in control, aggression, an emphasis on physical strength, and sexual promiscuity. Femininity refers to a set of stereotypical female traits and behaviors, including passivity, an emphasis on being pleasing, gentleness, dependence, and an emphasis on caring and empathy. Hyper-masculinity and hyper-femininity are exaggerations of these gender performances (think of the typical roles played by Arnold Schwartzenegger and Marilyn Monroe, respectively). Gender queer and gender non-conforming describes for characters who do not fit conventional gender distinctions, rather, they identify with neither or a combination of masculinity and femininity. Code the character as "feminine" or "masculine" unless cued otherwise.
- **LGBTQ**. A character's sexuality is determined by his/her apparent enduring attraction (emotional, sexual, romantic) to men, women, or both sexes. Code the character as "heterosexual" unless cued otherwise.

Analysis

Result for Question 1: Are there industry differences in representation in advertisements?

First we have an observation for the entire dataset.

```
Segment's feature distribution as follow':
Segment feature has 4 different values
Confectionary    729
Petcare          346
Wrigley          119
Food              93
Name: Segment, dtype: int64
**************************
Age's feature distribution as follow':
Age feature has 8 different values
3      428
4      283
5      152
2      140
999    117
6       68
1       56
7       43
Name: Age, dtype: int64
**************************
Gender's feature distribution as follow':
Gender feature has 3 different values
1      760
2      524
888      3
Name: Gender, dtype: int64
**************************
Race's feature distribution as follow':
Race feature has 9 different values
1      705
3      216
999    113
2      107
4       86
6       28
888     21
8        6
5        5
Name: Race, dtype: int64
```

- For the Segment column, we can find the “Confectionary” accounts for half of the total data volume of the Segment column.
- For Age column, most of the values are 3, which is the age of 20-29 year olds.
- For Gender column, number of male is more than female, and there is a little people don't tell their sex.
- For the Race column, we find that value "1"("White") is the largest, the second one and third one are "3"("Asian/Asian American") and "999"("Not Applicable").
- We can find that the number of whites is greater than the sum of the numbers of all other races.

Since “industry” refers to the “Segment” column, and “representation” refers to all the question columns. We find the industry differences among Age, Gender Race columns, since they are common statistical variables.

Observation of industry differences for Age:

- We can find that among the Age=1, Petcare is the most.
- With the increasing age, more people are in the "Confectionary" industry, but when Age = 7 the Petcare is greater than Confectionary.
- We observe that the largest amount of data is in the "Confectionary" industry.
- Some don't tell or are unwilling to disclose the age information of age.

Observation of industry differences for Gender:

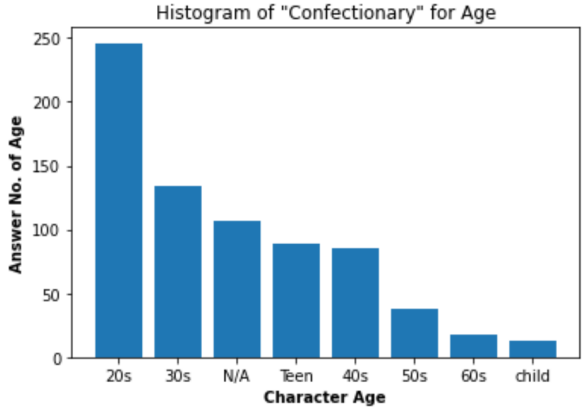
- We can find that among the Gender = 1 which is "Man", the "Confectionary" is the most, "Food" is the smallest.
- We can find that among the Gender = 2 which is "Woman", the "Confectionary" is the most, "Food" is the smallest.
- We can find that among the Gender = 888 which is "Can't tell", there is only Petcare.
- We observe that the largest amount of data is in the "Confectionary" industry.
- Some people can not tell their gender.

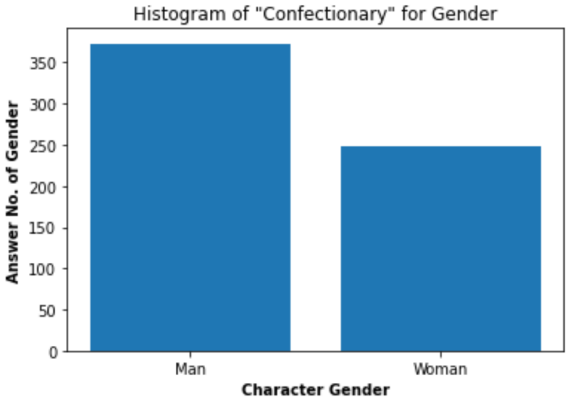
Observation of industry differences for Race:

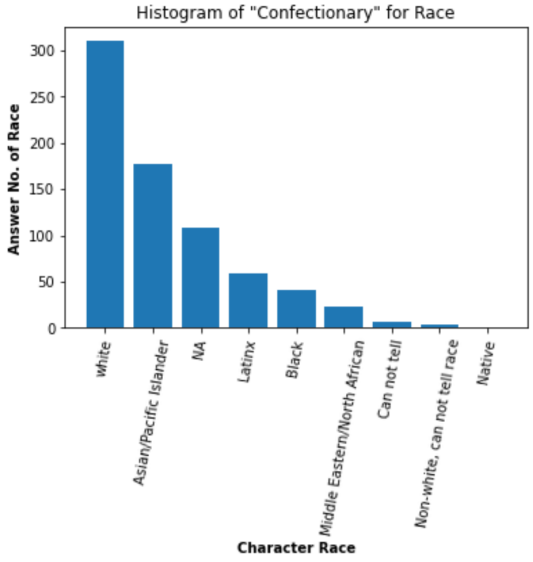
- We can find that among the Race = 1("White"), the number of "Confectionary" is 310 and "Petcare " is 263 which are far more than "Wrigley" and "Food" industry. Majority of "White" work in these two industries.
- We can find that among the Race = 2("Black"), the "Confectionary" is also higher than other industries, each industry has "Black". The portion of "Petcare" for "Black" is the highest compared with other ethnicities.
- We can find that among the Race = 3("Asian/Asian American"), there are a very high number of people working in the "Confectionary" industry.
- We can find that among the Race = 4("Latinx"), the "Confectionary" is much higher than other industries.
- We can find that among the Race = 5("Native"), there is a small amount of data.
- We can find that among the Race = 6("Middle Eastern"), no one is in the "Food" industry. The ratio of "Confectionary" among industries for "Asian" is the highest compared with other ethnicities.
- We can find that among the Race = 7("Multi-racial (only if you know for certain)"), there is no data.
- We can find that among the Race = 8("Non-white, but cannot tell specific race"), there are small amount of data, the number of "Confectionary" is 3, for "Petcare" is 2, for "Wrigley" is 1, no one is in "Food" industry.
- We can find that among the Race = 888("Can’t tell"), the number of data is relatively small, the number of people for "Confectionary" and "Food" is the same.

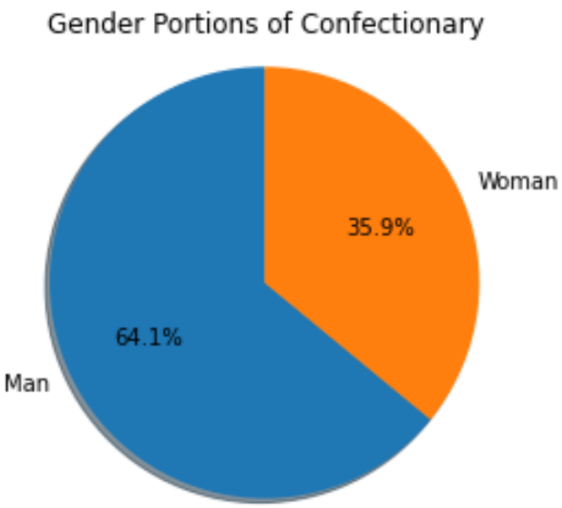
- We can find that among the Race = 999("Not Applicable"), the "Confectionary" is higher than other industries, people are only in the "Confectionary" and "Food" industry.
- We observe that the largest amount of data whatever their race is in the "Confectionary" industry.
- Many people are "Not Applicable" of Race in "Confectionary" and "Food".

Observation of industry differences for confectionary dataframe:





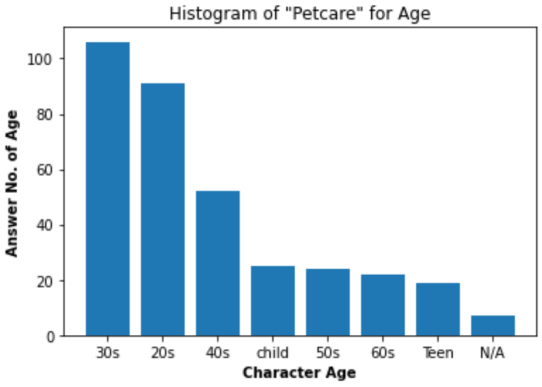


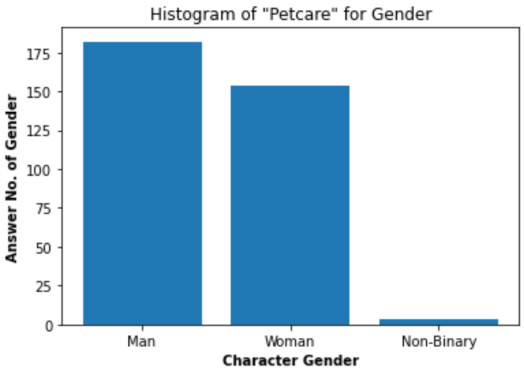


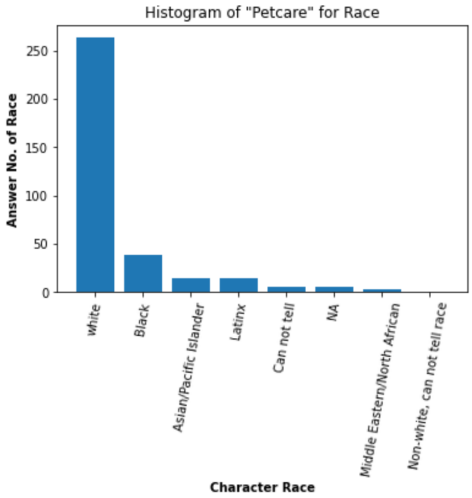
- We can find that for the “Confectionary” industry, the 20s has the largest proportion, the second one is 30s.
- Through the Gender portion graph, we can find that the number of male is definitely greater than females.
- From the histogram, we observe that "White" has the largest number of any race in the "Confectionary" industry, and there are large numbers of "Asian/Pacific Islander" and "Not Applicable".
- We can find that for the “Confectionary” industry, the amount of "Man(1)" is 64.1%, which is more than "Woman(2)" (35.9%).

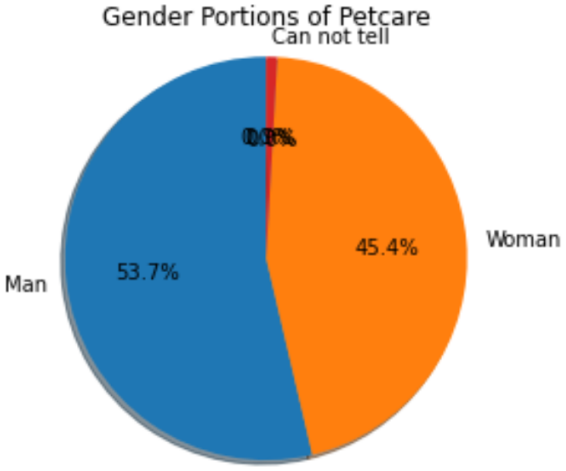
- From the four bar plots for gender portions of different industries, we can conclude that the ‘Confectionary’ has the largest difference in the number of men and women.
- We exclude the value of '999' and '888', and we can get the average value of ‘Age’ for the ‘Confectionary’ industry is 3.6,the std value is 1.27.

Observation of industry differences for Petcare dataframe:



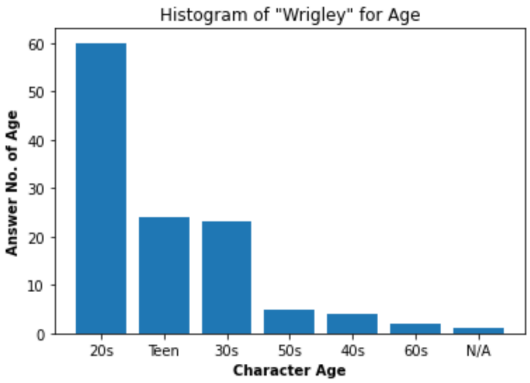


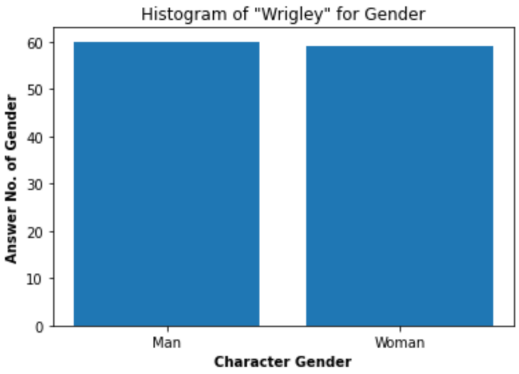


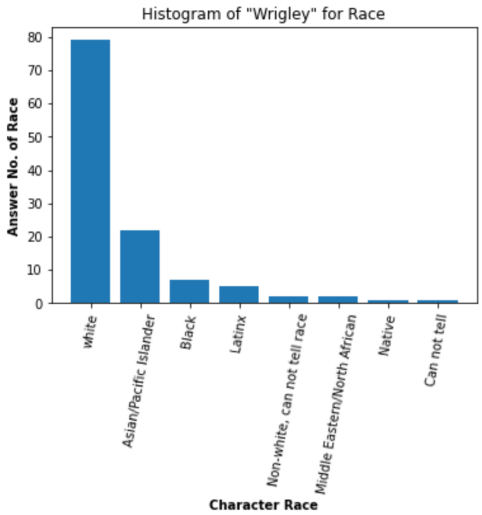


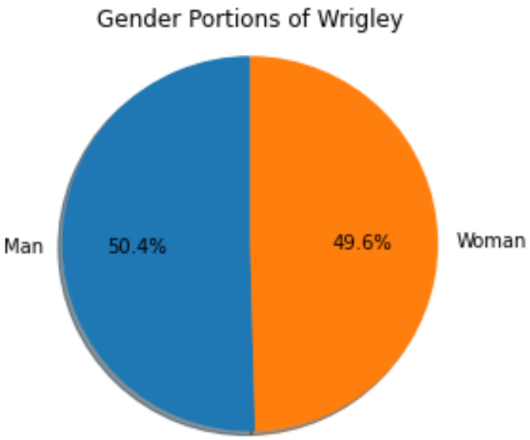
- We can find that for "Petcare", the 30s has the largest proportion, the second one is 20s.
- For "Percare", compared with other races, "White" is the largest.
- We can find that for Petcare, the amount of "Man(1)" is 53.8% which is the largest one, the amount of "Woman(2)" is 45.4% and the amount of "Can't tell(888)" is 0.9%.
- We can get the average value of Age for the Petcare industry is 3.89, which is higher than ‘Confectionary’ industry, the std value is 1.47.

Observation of industry differences for Wrigley dataframe:



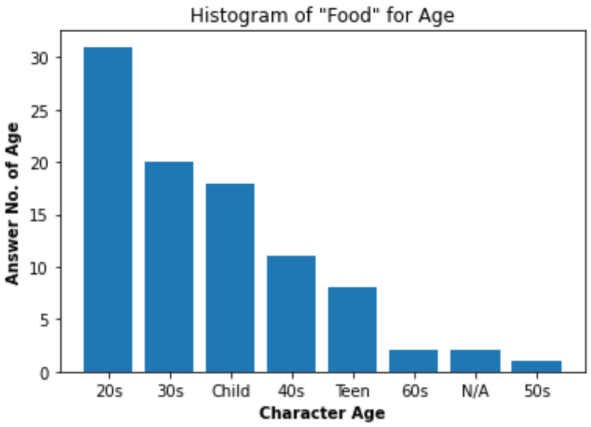


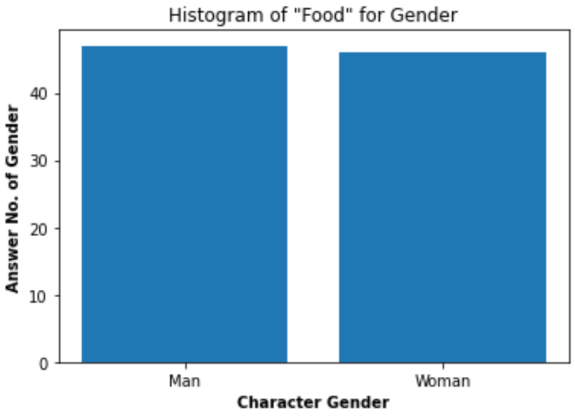


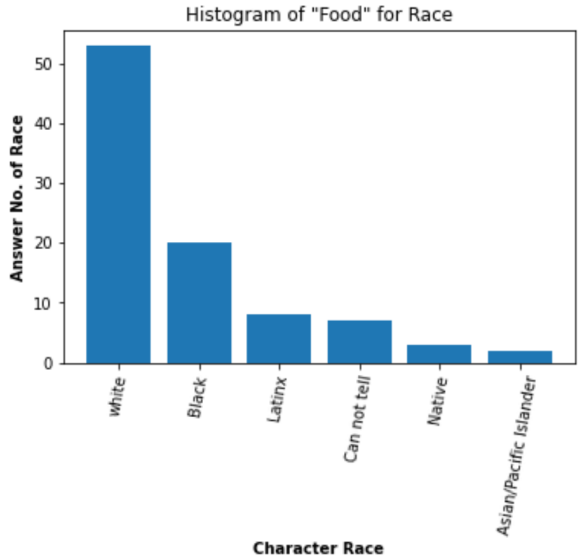


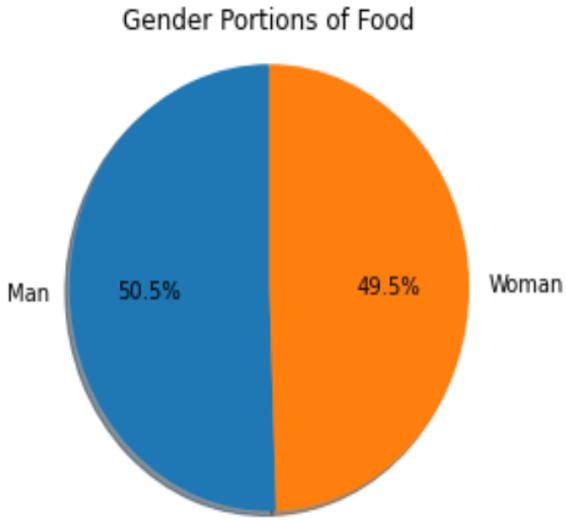
- We can find that for “Petcare”, the 20s has the largest proportion, the second one is Teen, but number of 30s is close to Teen
- For "Wrigley", compared with other races, "White" is the largest.
- We can find that for the Confectionary segment, the amount of men is 50.4% which is more than women 49.6%.
- The number of "men(1)" and "women(2)" is basically the same.
- From the four barplot for gender portions of different industries, we can conclude that “Wrigley” has the smallest difference in the number of men and women.
- We can get the average value of Age for the “Wrigley” industry is 3.25(around 20 years old), which is smaller than “Confectionary” and “Petcare” and bigger than “Food”, the std value is 1.06, which means the age distribution is relatively concentrated.

Observation of industry differences for Food dataframe:









- We can find that for the “Petcare” industry, the 20s has the largest proportion, the second one is 30s, and the number of child is close to 30s.
- For "Food", compared with other races, "White" is the largest
- We can find that for the “Confectionary” industry, the amount of "Man(1)" is 50.5% which is more than "Women(2)" 49.5%.
- The number of "Man(1)" and "Woman(2)" is basically the same.
- We can get the average value of Age for the “Food” industry is 3.1, which is smallest among these four industries, the std value is 1.43, which means the age distribution is not relatively concentrated.
- We can conclude that the average age of the “Food” industry is the youngest.

Result for Question 2: Is there a change in representation in advertisements over time?

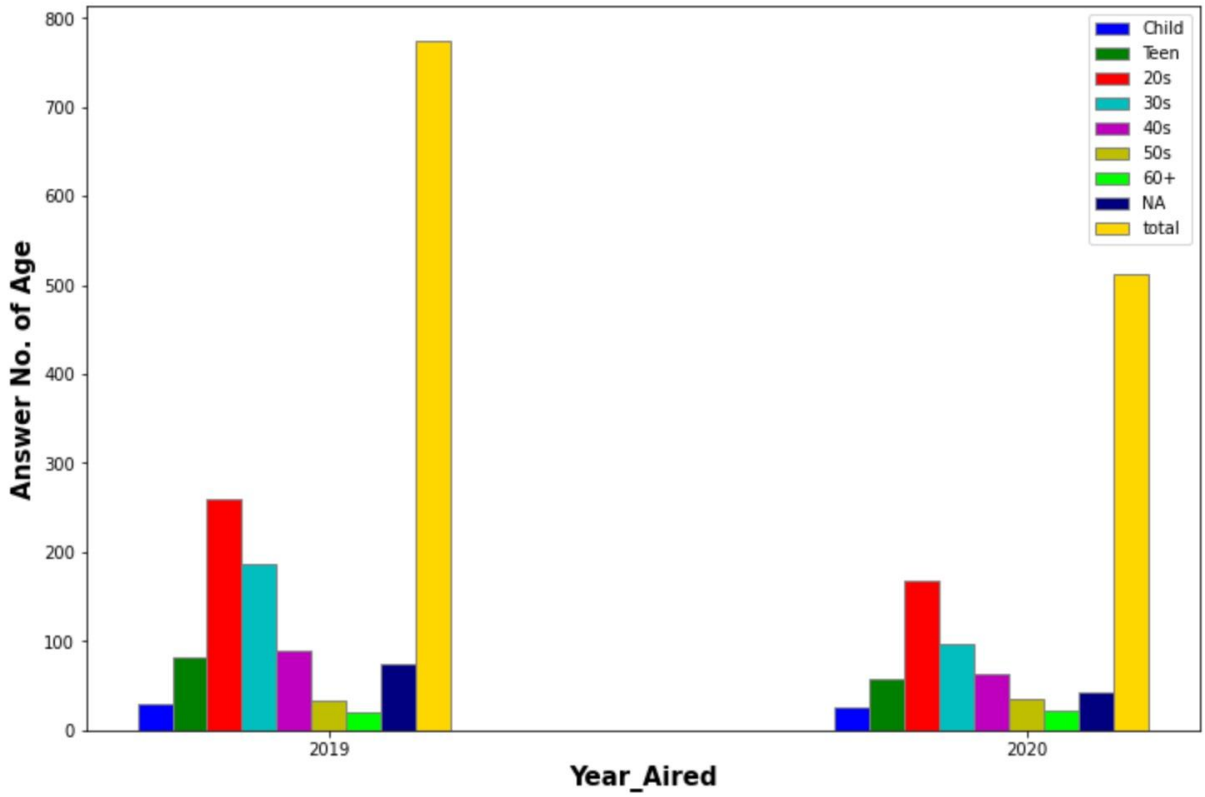
First we have an observation for the entire dataset:

- 1. For the Year_Aired column, there are only 2 unique values: 2019 and 2020.
- 2. For the Age column, most of values are Age = 4, which is the age of 20-29 year olds.
- 3. For the Gender column, number of male is more than female, and there is a little people don't tell their sex.
- 4. For the Race column, we find that value 1 is the largest, which means white people.

Since “time” refers to the “Year_Aired” column, and “representation” refers to all the question columns. We find the differences among 'Age', 'Gender' and ‘Race’ columns over ‘Year_Aired’, since they are common statistical variables

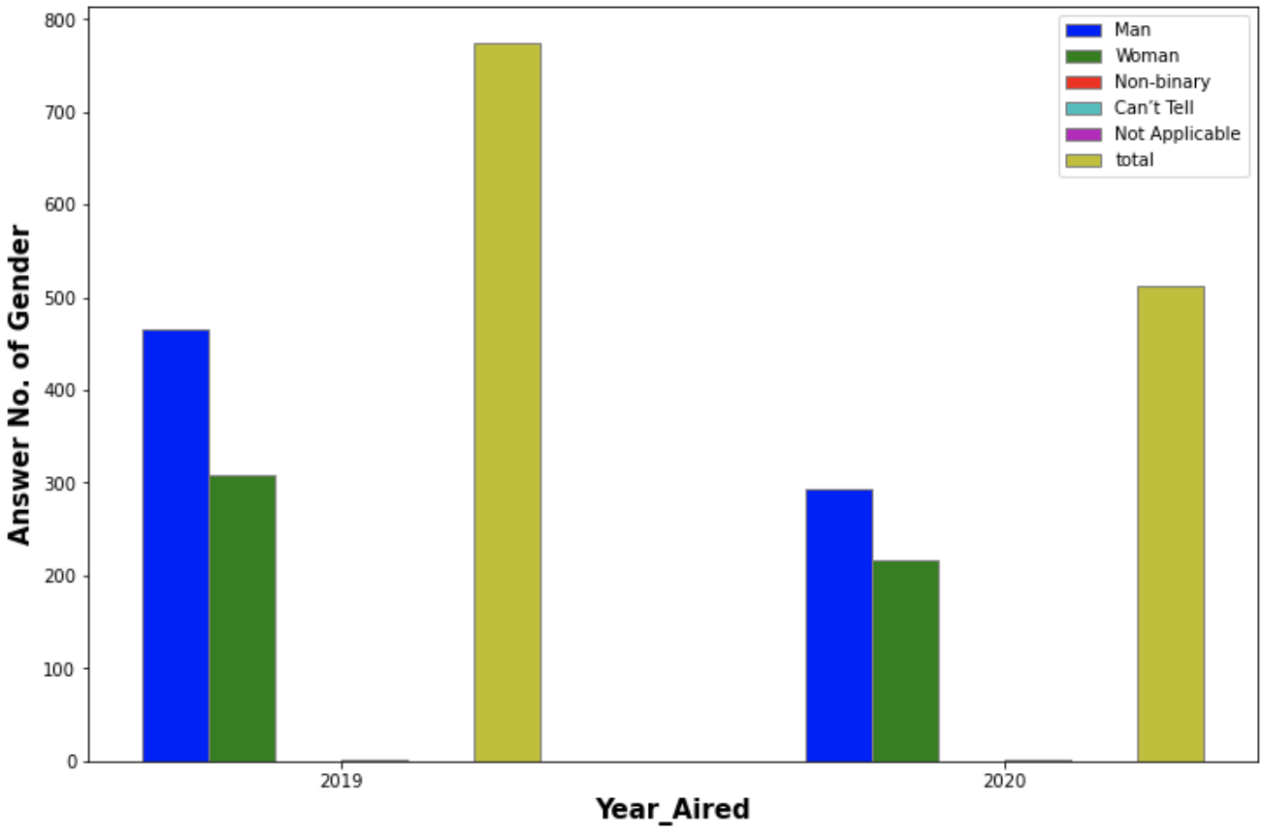
Observation of differences for Age over time:

- The total number decreases from 2019 to 2020.
- “20s” is the largest value between two years.
- Someone don't tell or are unwilling to disclose the information of age.
- The distributions of ages between two years are almost the same.



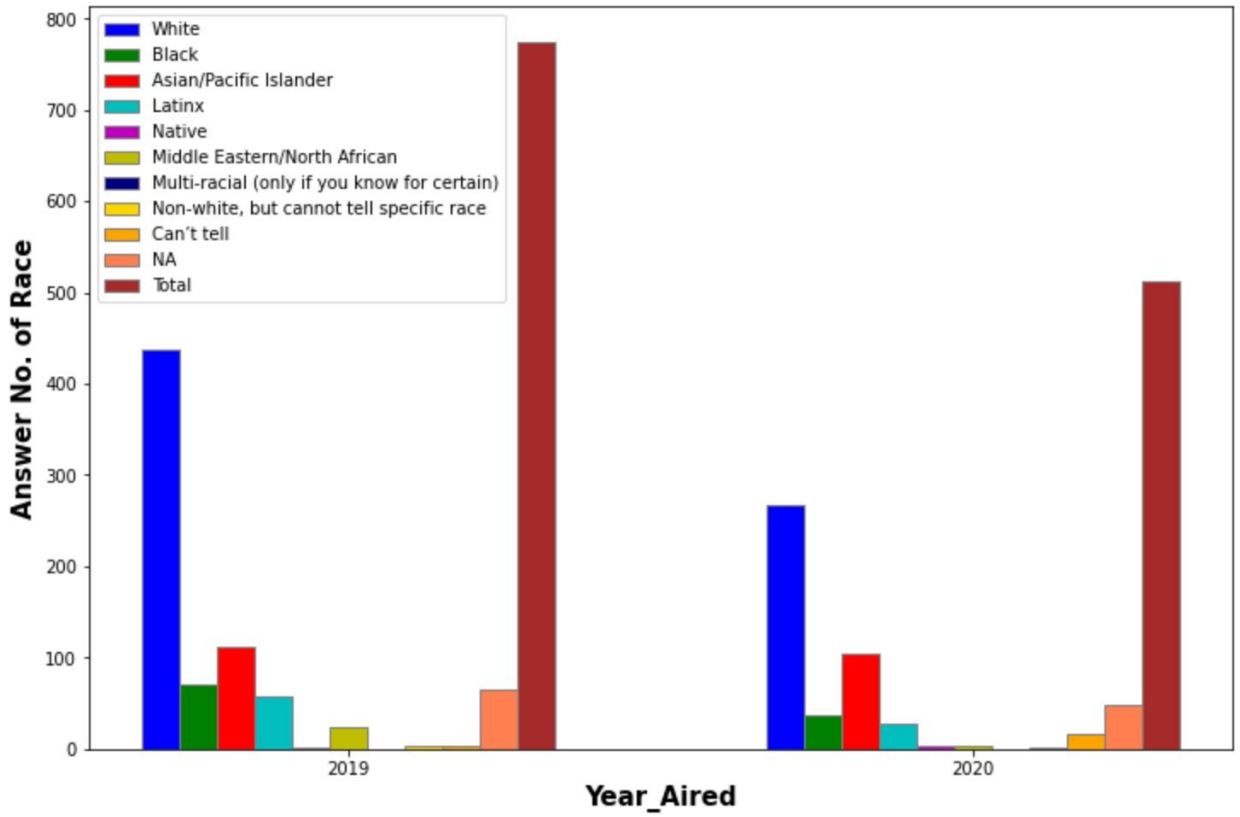
Observation of industry differences for Gender:

- We find that the total number decreases from 2019 to 2020.
- Male numbers are larger than female numbers in both two years 3. And some of people can not tell their sex.



Observation of industry differences for Race:

- We find that the total number of Race decreases from 2019 to 2020.
- Among all different answers, option 1, which refers to "White", is the largest in both two years.
- The total number of rest options is almost the same as the number of option 1.

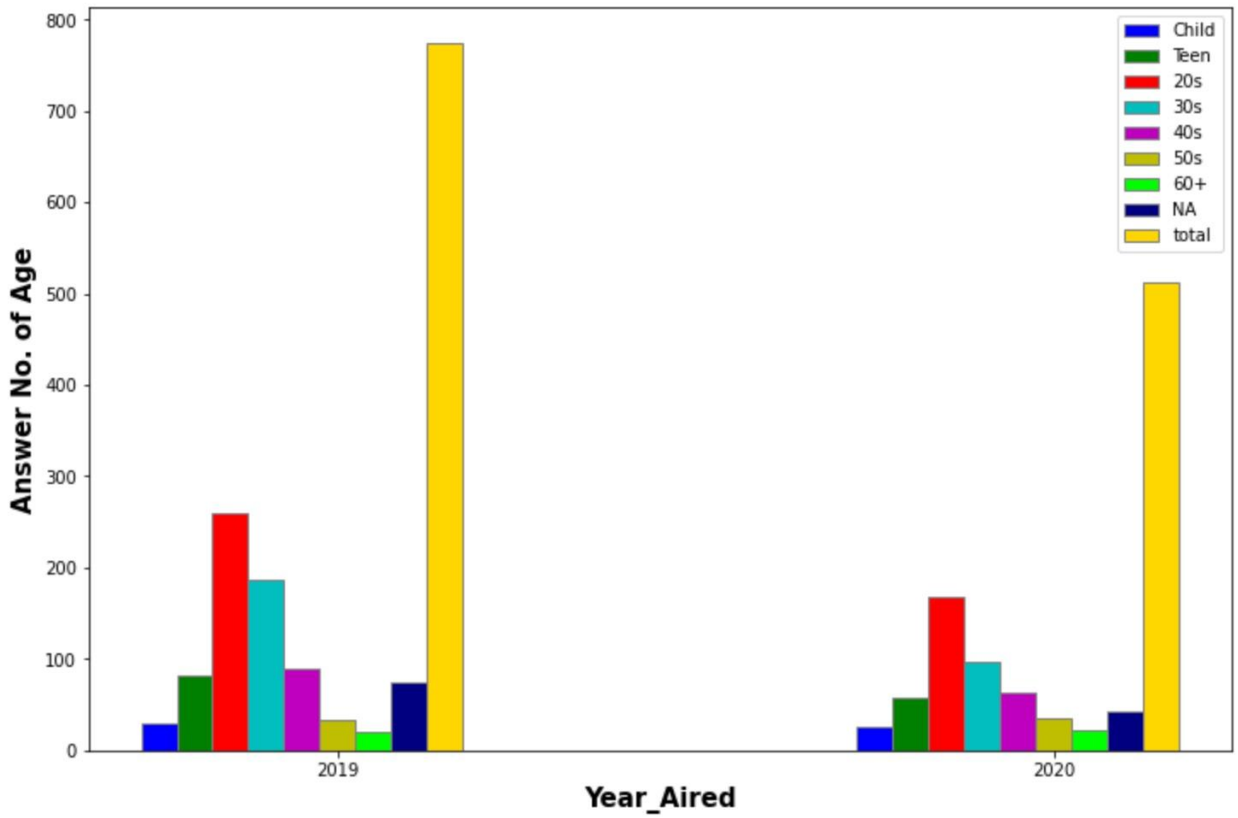


Result for Question 3: What are the trends?

Our group tackled this question based on our previous answer to question "Is there a change in representation in advertisements over time?" Our interpretation of "trends" is: the data moving tendency in terms of the columns (e.g. Age, Gender, Race) over time (i.e. Year_Aired).

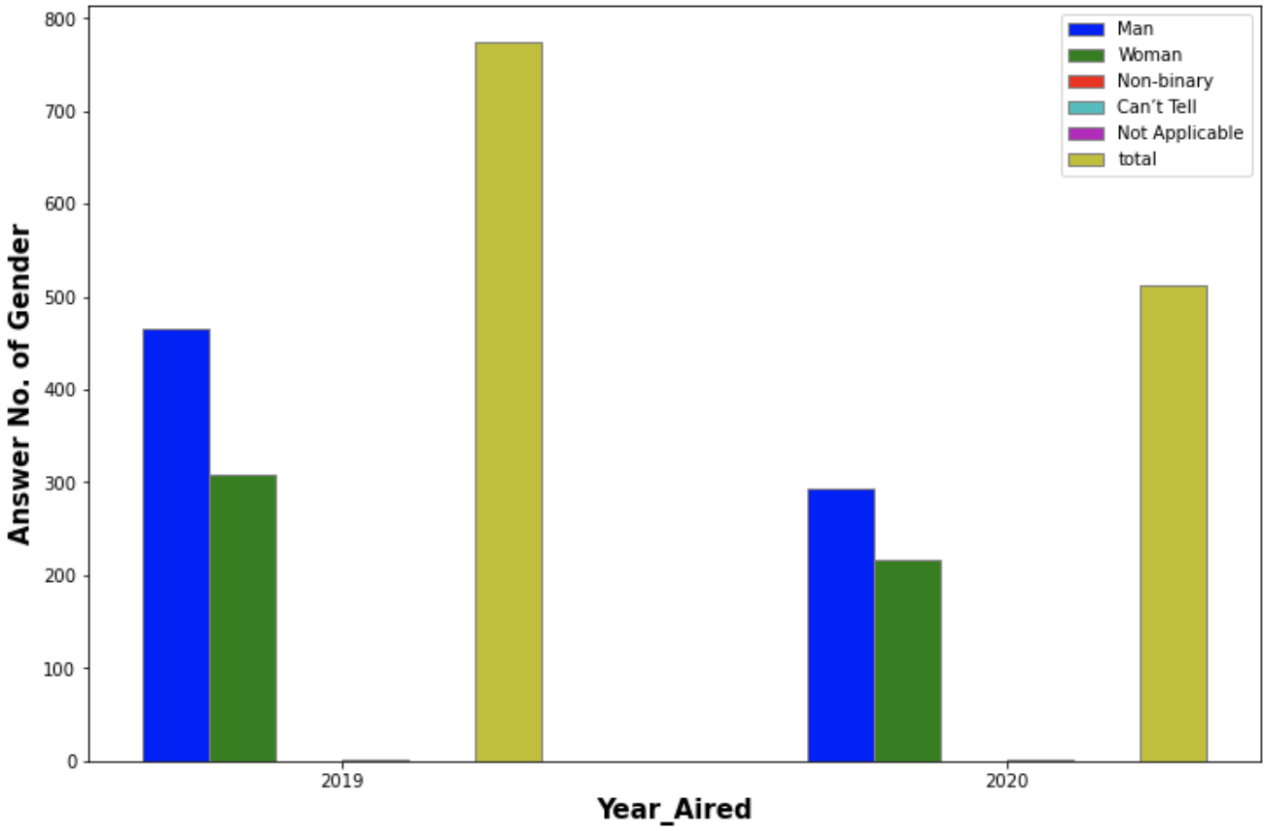
Trends of Age:

- The total number decreases from 2019 to 2020.
- “20s” is the largest value between two years.
- Someone doesn't tell or are unwilling to disclose the information of age 4. The distributions of ages between two years are almost the same.



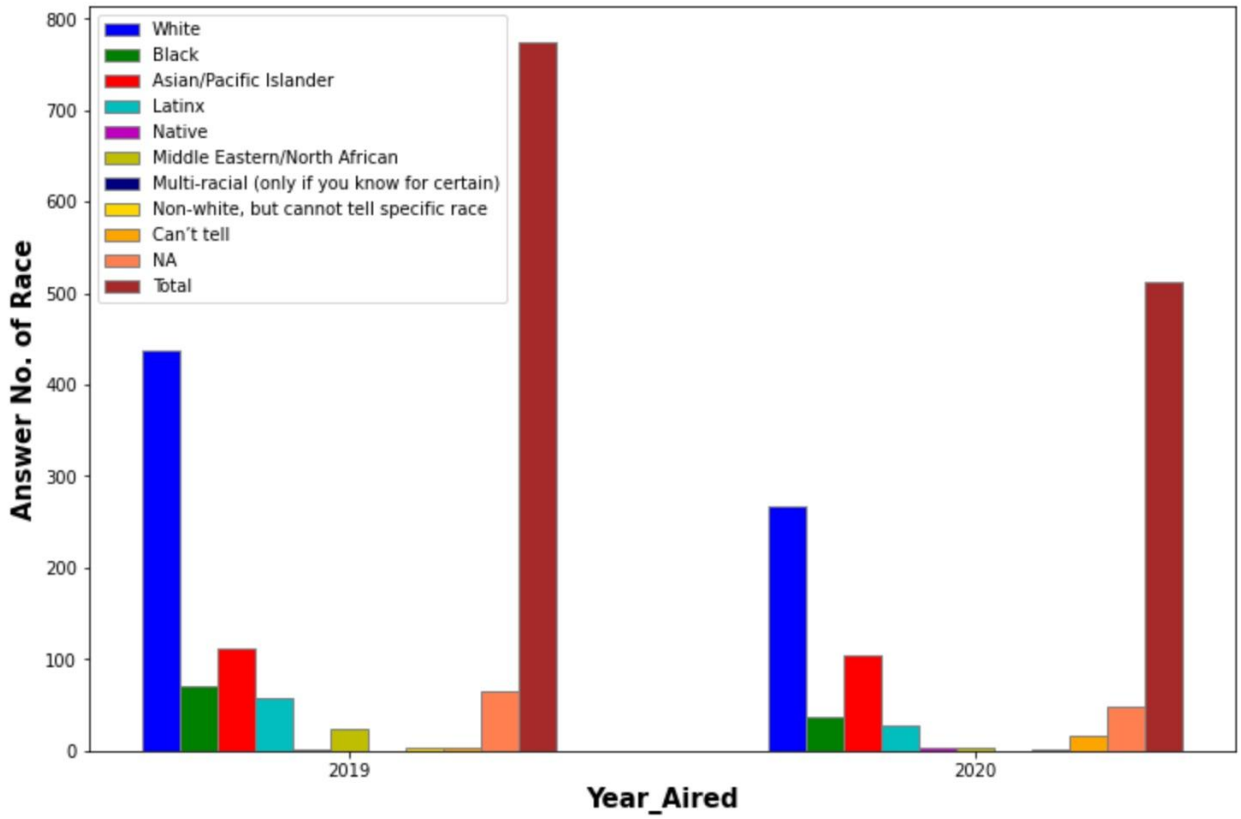
Trends of Gender:

- We find that the total number decreases from 2019 to 2020.
- Male numbers are larger than female numbers in both two years 3. And some of people can not tell their sex.



Trends of Race:

- We find that the total number of Race decreases from 2019 to 2020.
- Among all different answers, option 1, which refers to "White", is the largest in both two years.
- The total number of rest options is almost the same as the number of option 1.


White
Black
Asian/Pacific Islander
Latinx
Native
Middle Eastern/North African
Multi-racial (only if you know for certain)
Non-white, but cannot tell specific race
Can't tell
NA
Total
Answer No. of Race
0
100
200
300
400
500
600
700
800
2019
2020
Year_Aired

Appendix

This section will demonstrate all the code that our group has written to analyze the datasets.

Q1:

Analysis whole data set

In [4]:
```
df = pd.read_csv("Merged_Preprocessed_Mars2020_2021.csv")
```

Show all columns

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Coder', 'Asset_Name', 'Brand', 'Lead_Country', 'Year_Produced',
       'Year_Aired', 'Segment', 'Agency', 'Character_Name',
       'Character_Description', 'Prominence', 'Animated', 'Animated Specify',
       'Gender', 'Trans', 'Race', 'Race Other/Specify', 'API', 'Skin tone',
       'Sexual Orientation', 'Queer', 'Age', 'Disabled', 'Disability Specify',
       'Body Type', 'Shopping', 'Driving', 'Cleaning', 'Cooking', 'Working',
       'Socializing', 'Nothing', 'EatingDrinking', 'Exercising',
       'Other Activity', 'Activity Other Specify', 'Kitchen', 'Office', 'Car',
       'Store', 'Outdoors', 'Living Room', 'Restaurant/Bar', 'Gym', 'Bedroom',
       'Bathroom', 'Sporting Event', 'Classroom', 'Setting Other',
       'Other Setting Specify', 'Revealing Clothing', 'Nudity',
       'Visually Objectified', 'Verbally Objectified', 'Intelligent', 'Funny',
       'Occupation', 'Leader', 'Authority', 'Q27a_Disordered_Eating',
       'Q27b_Selfy_injury', 'Q27c_NegativeTalk', 'Q27d_Body_Modification',
       'Q28a_Visual_Shame', 'Q28b_Verbal_Shame', 'Q28c_Sizeist_Slurs',
       'Q28d_Punchline', 'Q28e_Denied_Personal_Opportunity',
       'Q28f_Denied_Professional_Opportunity', 'Q28g_Other_Prejudice',
       'Q28g_Prejudice_Other_Specify', 'Q29a_Lazy', 'Q29b_Physically_Slow',
       'Q29c_Stupid', 'Q29d_Loser', 'Q29e_Inactive', 'Q29f_Poorly_Dressed',
       'Q29g_Funny', 'Q29h_Jolly', 'Q29i_Clumsy', 'Q29j_Alone',
       'Q30a_Comic_Relief', 'Q30b_Sidekick', 'Q30c_Mamma_Hen', 'Q30d_Nympho',
       'Q31_Fat_to_Fit', 'Q32_Inspo_Porn', 'Q31_NOTES', 'Notes on Dwelling'],
      dtype='object')
```

Seperate to four dataframe of different columns

In [43]:
```
segment_col = df[['Segment']]

age_col = df[['Age','Segment']]

sex_col = df[['Gender','Segment']]
```

category feature nunique distribution

In [45]:
```
# classify feature
cat_fea =['Segment','Age','Gender','Race']
# category feature nunique distribution
for fea in cat_fea:
    print('****************************')
    print(fea + "'s feature distribution as follow': ")
    print("{} feature has {} different values".format(fea,df[fea].nunique()))
    print(df[fea].value_counts())
```

```
****************************
Segment's feature distribution as follow':
Segment feature has 4 different values
Confectionary    729
Petcare          346
Wrigley          119
Food              93
Name: Segment, dtype: int64
****************************
Age's feature distribution as follow':
Age feature has 8 different values
3      428
4      283
5      152
2      140
999    117
6       68
1       56
7       43
Name: Age, dtype: int64
****************************
Gender's feature distribution as follow':
Gender feature has 3 different values
1      760
2      524
888      3
Name: Gender, dtype: int64
****************************
Race's feature distribution as follow':
Race feature has 9 different values
1      705
3      216
999    113
2      107
4       86
6       28
888     21
8        6
5        5
Name: Race, dtype: int64
```

Observation for Q2_Age:

We can find that among the Q2_Age=1, the Petcare is the most

With the increasing of the age, more people are in the "Confectionary" industry, but when Q2_Age = 8 the Petcare is greater than Confectionary

And by the statistics, we observe that the largest amount of data are in the "Confectionary" industry

And someone don't tell or are unwilling to disclose the information of age

Industry differences of the Q3_Sex column

In [12]:
```
sex_list = [1,2,9]
for i in sex_list:
    df_sex = sex_col.loc[sex_col['Q3_Sex'] == i]
    print('Q3_Sex =',i)
    print(df_sex['Segment'].value_counts())
    print()
```

```
Q3_Sex = 1
Confectionary    262
Petcare          157
Wrigley           59
Food              46
Name: Segment, dtype: int64

Q3_Sex = 2
Confectionary    467
Petcare          186
Wrigley           60
Food              47
Name: Segment, dtype: int64

Q3_Sex = 9
Petcare    3
Name: Segment, dtype: int64
```

Industry differences of the Gender column

In [50]:
```
sex_list = [1,2,3,888,999]
for i in sex_list:
    df_sex = sex_col.loc[sex_col['Gender'] == i]
    print('Gender =',i)
    print(df_sex['Segment'].value_counts())
    print()
```

```
Gender = 1
Confectionary    467
Petcare          186
Wrigley           60
Food              47
Name: Segment, dtype: int64

Gender = 2
Confectionary    262
Petcare          157
Wrigley           59
Food              46
Name: Segment, dtype: int64

Gender = 3
Series([], Name: Segment, dtype: int64)

Gender = 888
Petcare    3
Name: Segment, dtype: int64

Gender = 999
Series([], Name: Segment, dtype: int64)
```

Deliverable 2 - Industry differences of the Race column

In [51]:
```
race_list = [1,2,3,4,5,6,7,8,888,999]
for i in race_list:
    df_race = race_col.loc[race_col['Race'] == i]
    print('Race =',i)
    print(df_race['Segment'].value_counts())
    print()
```

```
Race = 1
Confectionary    310
Petcare          263
Wrigley           79
Food              53
Name: Segment, dtype: int64

Race = 2
Confectionary    41
Petcare          39
Food             20
Wrigley           7
Name: Segment, dtype: int64

Race = 3
Confectionary    177
Wrigley           22
Petcare           15
Food               2
Name: Segment, dtype: int64

Race = 4
Confectionary    59
Petcare          14
Food              8
Wrigley           5
Name: Segment, dtype: int64
```

Create confectionary dataframe

In [52]:
```
df_confectionary = df.loc[df['Segment'] == 'Confectionary']
df_confectionary = df_confectionary.loc[:,['Age','Gender', 'Race']]
df_confectionary.reset_index(inplace = True)
df_confectionary.drop('index', axis = 1, inplace = True)
df_confectionary.head()
```

Out[52]:

	Age	Gender	Race
0	4	1	3
1	3	1	3
2	7	1	3
3	2	1	3
4	2	2	3

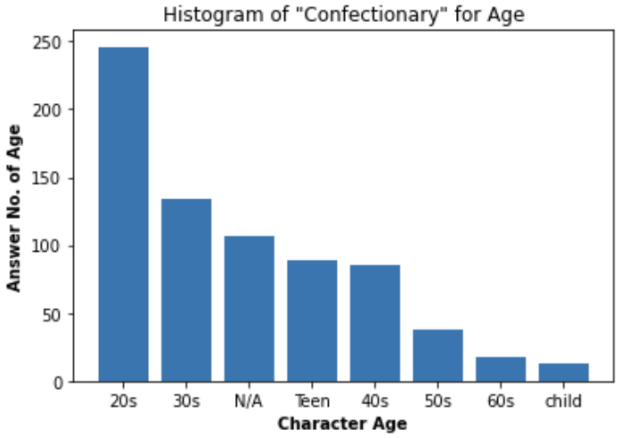
category feature nunique distribution

In [53]:
```
cat_fea_confectionary =['Age','Gender','Race']
# category feature nunique distribution
for fea in cat_fea_confectionary:
    print('*****************************')
    print(fea + "'s feature distribution as follow': ")
    print("{} feature has {} different values".format(fea,df_confectionary[fea].nunique()))
    print(df_confectionary[fea].value_counts())
```

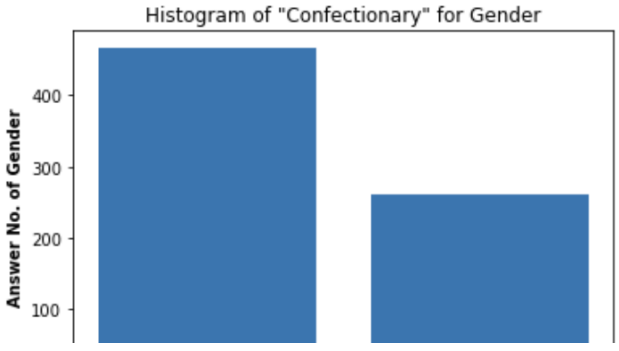
```
*****************************
Age's feature distribution as follow':
Age feature has 8 different values
3      246
4      134
999    107
2       89
5       85
6       38
7       17
1       13
Name: Age, dtype: int64
*****************************
Gender's feature distribution as follow':
Gender feature has 2 different values
1    467
2    262
```

Plot Histogram for different columns

In [148]:
```
# Age
age = ['20s','30s','N/A','Teen','40s','50s','60s','child']
v = df_confectionary['Age'].value_counts().nlargest(15)
plt.bar(age,v.values)
plt.subplot(1,1,1)
plt.xticks(rotation=0)
plt.xlabel('Character Age',fontweight ='bold', fontsize = 10)
plt.ylabel('Answer No. of Age',fontweight ='bold',  fontsize = 10)
plt.title('Histogram of "Confectionary" for '+'Age')
plt.show()
```

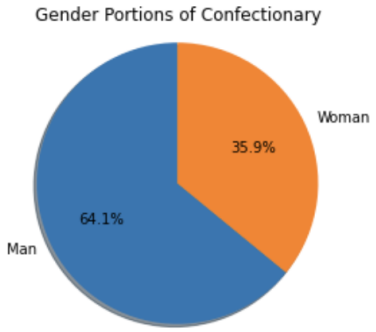


In [149]:
```
# Gender
gender = ['Man','Woman']
v = df_confectionary['Gender'].value_counts().nlargest(15)
plt.bar(gender,v.values)
plt.subplot(1,1,1)
plt.xticks(rotation=0)
plt.xlabel('Character Gender',fontweight ='bold', fontsize = 10)
plt.ylabel('Answer No. of Gender',fontweight ='bold',  fontsize = 10)
plt.title('Histogram of "Confectionary" for '+'Gender')
plt.show()
```



Barplot for gender portions of confectionary

In [151]:
```
Sex1 = df_confectionary[df_confectionary['Gender'] == 1]['Gender'].value_counts()
Sex2 = df_confectionary[df_confectionary['Gender'] == 2]['Gender'].value_counts()

labels = 'Man','Woman'
sizes = [Sex1.values.item(), Sex2.values.item()]
explode = (0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')

fig, ax = plt.subplots()
ax.pie(sizes, explode = explode, labels = labels, autopct='%1.1f%%',
        shadow = True, startangle = 90)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Gender Portions of Confectionary')
plt.show()
```



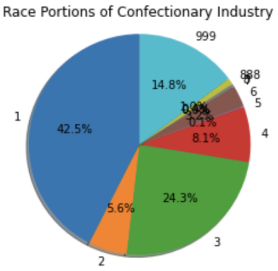
We can find that for Confectionary segment, the amount of "Man(1)" is 64.1%, which is more than "Woman(2)" (35.9

From the four barplot for gender portions of different industry, we can conclude that the Confectionary has the large man and woman

Barplot for Race portions of confectionary

In [152]:
```
Sex1 = df_confectionary[df_confectionary['Race'] == 1]['Race'].value_counts()
Sex2 = df_confectionary[df_confectionary['Race'] == 2]['Race'].value_counts()
Sex3 = df_confectionary[df_confectionary['Race'] == 3]['Race'].value_counts()
Sex4 = df_confectionary[df_confectionary['Race'] == 4]['Race'].value_counts()
Sex5 = df_confectionary[df_confectionary['Race'] == 5]['Race'].value_counts()
Sex6 = df_confectionary[df_confectionary['Race'] == 6]['Race'].value_counts()
Sex7 = df_confectionary[df_confectionary['Race'] == 7]['Race'].value_counts()
Sex8 = df_confectionary[df_confectionary['Race'] == 8]['Race'].value_counts()
Sex888 = df_confectionary[df_confectionary['Race'] == 888]['Race'].value_counts()
Sex999 = df_confectionary[df_confectionary['Race'] == 999]['Race'].value_counts()


labels = '1','2','3','4','5','6','7','8','888','999'
sizes = [Sex1.values.item(), Sex2.values.item(), Sex3.values.item(), Sex4.values.item(),
```

```
Sex888 = df_confectionary[df_confectionary['Race'] == 888]['Race'].value_counts()
Sex999 = df_confectionary[df_confectionary['Race'] == 999]['Race'].value_counts()


labels = '1','2','3','4','5','6','7','8','888','999'
sizes = [Sex1.values.item(), Sex2.values.item(), Sex3.values.item(), Sex4.values.item(),
         Sex5.values.item(), Sex6.values.item(), 0, Sex8.values.item(), Sex888.values.item(), Sex999.values.item()]
explode = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')

fig, ax = plt.subplots()
ax.pie(sizes, explode = explode, labels = labels, autopct='%1.1f%%',
        shadow = True, startangle = 90)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Race Portions of Confectionary Industry')
plt.show()
```



With barplot we can observe the result obviously

Compute Mean and std of Age for Confectionery

In [153]:
```
df_confectionary = df_confectionary.loc[df_confectionary['Age'] < 10]
age_mean_confectionary = df_confectionary['Age'].mean()
age_mean_confectionary = round(age_mean_confectionary, 2)
age_std_confectionary = df_confectionary['Age'].std()
age_std_confectionary = round(age_std_confectionary, 2)
print('Mean of Age is', age_mean_confectionary, 'Standard deviation of Age is',age_std_confectionary)
```

```
Mean of Age is 3.6 Standard deviation of Age is 1.27
```

We exclude the value of '999' and '888', and we can get the average value of Age for the Confectionary industry is 3.6,the std value is 1.27

The analysis for ‘Petcare’, ‘Wrigley’ and ‘Food’ industries are similar to the ‘Confectionary’ industry.

Q2 & Q3

> **Question: Is there a change in representation in advertisements over time?**

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
df = pd.read_csv('Merged_Preprocessed_Mars2020_2021.csv')
df.head()
```

Out[2]:

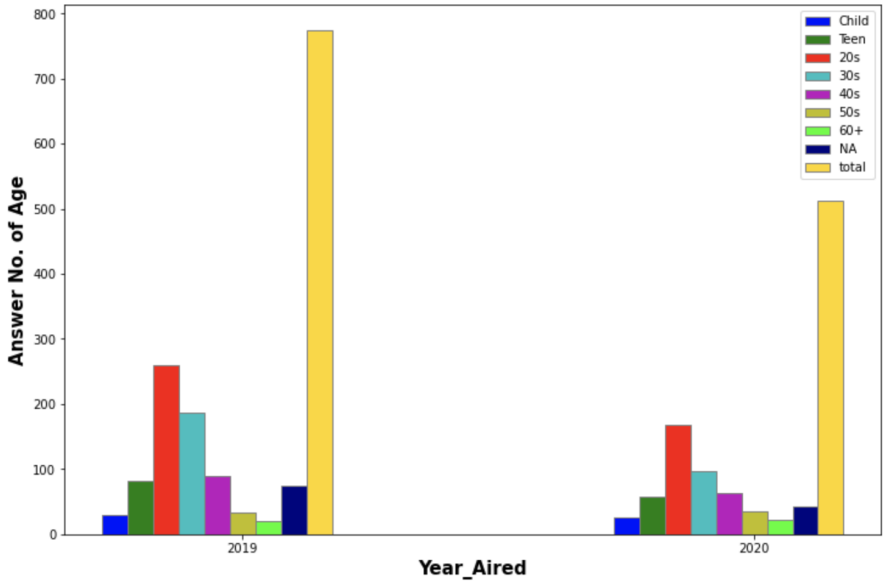
	Coder	Asset_Name	Brand	Lead_Country	Year_Produced	Year_Aired	Segment	Agency	Character_Name	Character_Description	...	Q29i_Clumsy
0	Emma	Airwaves 2020 Intense Mint launch campaign video	Airwaves	NORTH ASIA	2020	2020	Confectionary	DDB	Man	at party	...	1(
1	Pamela	BOOMER BICEP	BOOMER	India	2020	2020	Confectionary	DDB	Boomer Man	Shirtless, dark hair, chewing gum	...	1(
2	CEspinoza	BOOMER RAMAYAN	BOOMER	India	2020	2020	Confectionary	DDB	Older man watching TV	Older man watching TV	...	1(
3	CEspinoza	BOOMER RAMAYAN	BOOMER	India	2020	2020	Confectionary	DDB	Boy watching TV	Boy watching TV	...	1(
4	CAckel	BOOMER FISH BOWL	BOOMER	India	2020	2020	Confectionary	DDB	girl	sad	...	1(

5 rows × 89 columns

Age

In [4]:
```
# "Q2_Age", "Q3_Sex", and "Q7_Race_Ethnicity".
uniq_age = np.sort(df['Age'].unique()).tolist()
# to get age distribution of Year_Aired = 2020
age_total = [[0, 0] for _ in range(len(uniq_age) + 1)]
# age_total[i][0] means the value counts of i-th answer of Q2_Age in Year_Aired = 2019
# age_total[i][1] means the value counts of i-th answer of Q2_Age in Year_Aired = 2020
# age_total[-1][0/1] means the total value counts of Q2_Age in Year_Aired = 2019/2020

year_2020_counts = df[df['Year_Aired'] == 2020]['Age'].value_counts()
year_2019_counts = df[df['Year_Aired'] == 2019]['Age'].value_counts()
sum_of_2019 = 0
sum_of_2020 = 0
for i in range(len(age_total) - 1):
    age_idx = uniq_age[i]
    if age_idx in year_2020_counts.index:
        age_total[i][1] = year_2020_counts[age_idx]
        sum_of_2020 += age_total[i][1]
    else:
        age_total[i][1] = 0

    if age_idx in year_2019_counts.index:
        age_total[i][0] = year_2019_counts[age_idx]
        sum_of_2019 += age_total[i][0]
    else:
        age_total[i][0] = 0
uniq_age.append('total')
age_total[-1][0] = sum_of_2019
age_total[-1][1] = sum_of_2020
# age_total
```

In [5]:
```
# plot
# set width of bar
barWidth = 0.05
fig = plt.subplots(figsize =(12, 8))
colors = ['b','g','r','c','m','y', 'lime', 'navy', 'gold', 'orange', 'coral', 'brown']

# Set position of bar on X axis
br = []
br_tmp = np.arange(len(age_total[0]))
br.append(br_tmp)

for i in range(1, len(age_total)):
    br_tmp = [x + barWidth for x in br[i-1]]
    br.append(br_tmp)

uniq_age_desc = [
    'Child',
    'Teen',
    '20s',
    '30s',
    '40s',
    '50s',
    '60+',
    'NA',
    'total'
]
# Make the plot
for i in range(len(br)):
    plt.bar(br[i], age_total[i], color =colors[i], width = barWidth, edgecolor ='grey', label = str(uniq_age_desc[i]))

# Adding Xticks
plt.xlabel('Year_Aired', fontweight ='bold', fontsize = 15)
plt.ylabel('Answer No. of Age', fontweight ='bold', fontsize = 15)
plt.xticks([r + 5 * barWidth for r in range(len(br[0]))],
        ['2019', '2020'])

plt.legend()
plt.show()
```



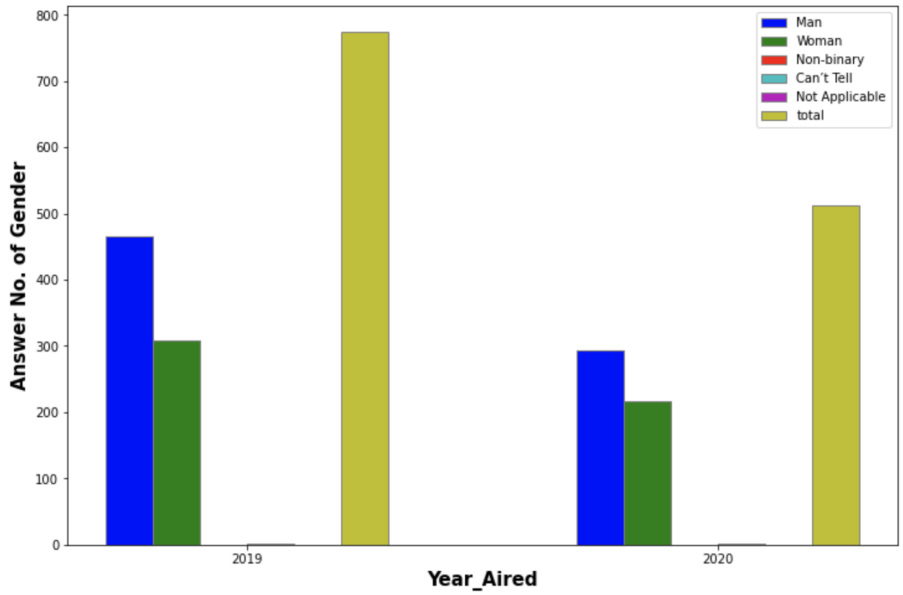
Conclusion

From the bar chart above, we can see that the distribution of two answers are almost the same; both figure out that option on those who in their 20s, have the most people. But comparing to 2019, the total number of answers in 2020 decreased.

Gender

In [6]:
```
# uniq_sex = np.sort(df['Q3_Sex'].unique()).tolist()
uniq_sex = [1,2,3,888,999]
uniq_sex_desc = ['Man','Woman','Non-binary','Can't Tell','Not Applicable', 'total']
# to get age distribution of Year_Aired = 2020
sex_total = [[0, 0] for _ in range(len(uniq_sex) + 1)]
# sex_total[i][0] means the value counts of i-th answer of Q3_Sex in Year_Aired = 2019
# sex_total[i][1] means the value counts of i-th answer of Q3_Sex in Year_Aired = 2020
# sex_total[-1][0/1] means the total value counts of Q3_Sex in Year_Aired = 2019/2020

year_2020_counts = df[df['Year_Aired'] == 2020]['Gender'].value_counts()
year_2019_counts = df[df['Year_Aired'] == 2019]['Gender'].value_counts()
sum_of_2019 = 0
sum_of_2020 = 0
for i in range(len(sex_total) - 1):
    sex_idx = uniq_sex[i]
    if sex_idx in year_2020_counts.index:
        sex_total[i][1] = year_2020_counts[sex_idx]
        sum_of_2020 += sex_total[i][1]
    else:
        sex_total[i][1] = 0

    if sex_idx in year_2019_counts.index:
        sex_total[i][0] = year_2019_counts[sex_idx]
        sum_of_2019 += sex_total[i][0]
    else:
        sex_total[i][0] = 0
uniq_sex.append('total')
sex_total[-1][0] = sum_of_2019
sex_total[-1][1] = sum_of_2020
# sex_total
```

In [7]:
```
# plot
# set width of bar
barWidth = 0.1
fig = plt.subplots(figsize =(12, 8))
colors = ['b','g','r','c','m','y', 'lime', 'navy', 'gold', 'orange', 'coral', 'brown']

# Set position of bar on X axis
br = []
br_tmp = np.arange(len(sex_total[0]))
br.append(br_tmp)

for i in range(1, len(sex_total)):
    br_tmp = [x + barWidth for x in br[i-1]]
    br.append(br_tmp)

# Make the plot
for i in range(len(br)):
    plt.bar(br[i], sex_total[i], color =colors[i], width = barWidth, edgecolor ='grey', label = str(uniq_sex_desc[i]))

# Adding Xticks
plt.xlabel('Year_Aired', fontweight ='bold', fontsize = 15)
plt.ylabel('Answer No. of Gender', fontweight ='bold', fontsize = 15)
plt.xticks([r + 2.5 * barWidth for r in range(len(br[0]))],
        ['2019', '2020'])

plt.legend()
plt.show()
```



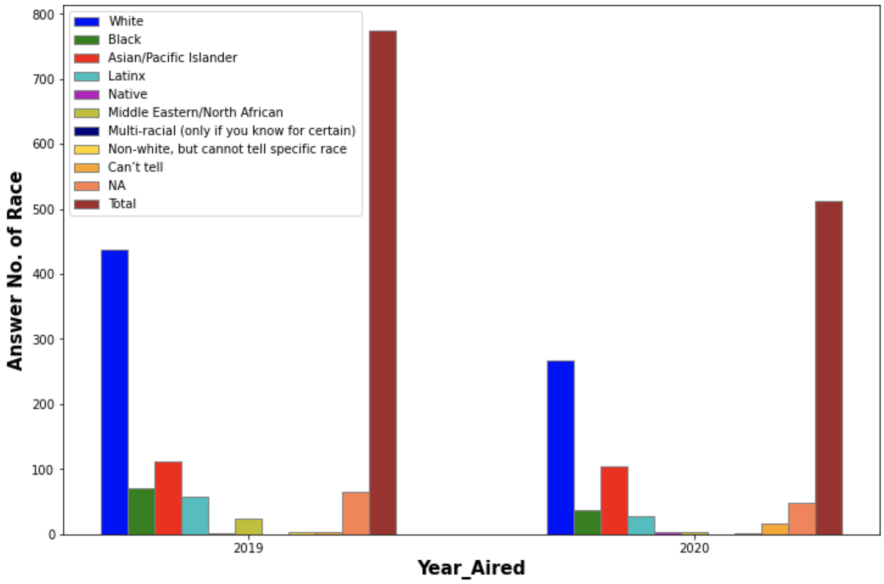
Conclusion

From the bar chart of the distribution of Gender answers, we can find that the distribution of two different years are almost the same; both of them have more answers of option 2 -- which is 'male'. But comparing to 2019, the total number of answers in 2020 decreased.

Race

In [8]:
```
# uniq_race = np.sort(df['Q7_Race_Ethnicity'].unique()).tolist()
# uniq_race = [i for i in range(1, 12)]
uniq_race = [1,2,3,4,5,6,7,8,888,999]
uniq_race_desc = ['White',
             'Black',
             'Asian/Pacific Islander',
             'Latinx',
             'Native',
             'Middle Eastern/North African',
             'Multi-racial (only if you know for certain)',
             'Non-white, but cannot tell specific race',
             'Can't tell',
             'NA',
             'Total'
]
# to get age distribution of Year_Aired = 2020
race_total = [[0, 0] for _ in range(len(uniq_race) + 1)]
# race_total[i][0] means the value counts of i-th answer of Q7_Race_Ethnicity in Year_Aired = 2019
# race_total[i][1] means the value counts of i-th answer of Q7_Race_Ethnicity in Year_Aired = 2020
# race_total[-1][0/1] means the total value counts of Q7_Race_Ethnicity in Year_Aired = 2019/2020

year_2020_counts = df[df['Year_Aired'] == 2020]['Race'].value_counts()
year_2019_counts = df[df['Year_Aired'] == 2019]['Race'].value_counts()
sum_of_2019 = 0
sum_of_2020 = 0
for i in range(len(race_total) - 1):
    race_idx = uniq_race[i]
    if race_idx in year_2020_counts.index:
        race_total[i][1] = year_2020_counts[race_idx]
        sum_of_2020 += race_total[i][1]
    else:
        race_total[i][1] = 0

    if race_idx in year_2019_counts.index:
        race_total[i][0] = year_2019_counts[race_idx]
        sum_of_2019 += race_total[i][0]
    else:
        race_total[i][0] = 0
uniq_race.append('total')
race_total[-1][0] = sum_of_2019
race_total[-1][1] = sum_of_2020
# race_total
```

In [9]:
```
# plot
# set width of bar
barWidth = 0.06
fig = plt.subplots(figsize =(12, 8))
colors = ['b','g','r','c','m','y', 'navy', 'gold', 'orange', 'coral', 'brown']

# Set position of bar on X axis
br = []
br_tmp = np.arange(len(race_total[0]))
br.append(br_tmp)

for i in range(1, len(race_total)):
    br_tmp = [x + barWidth for x in br[i-1]]
    br.append(br_tmp)

# Make the plot
for i in range(len(br)):
    plt.bar(br[i], race_total[i], color =colors[i], width = barWidth, edgecolor ='grey', label = str(uniq_race_desc[i]))

# Adding Xticks
plt.xlabel('Year_Aired', fontweight ='bold', fontsize = 15)
plt.ylabel('Answer No. of Race', fontweight ='bold', fontsize = 15)
plt.xticks([r + 5 * barWidth for r in range(len(br[0]))],
        ['2019', '2020'])

plt.legend()
plt.show()
```



Conclusion ¶

From the bar chart above, we can see that the distributions of answers between two years do not change much. White people (Option 1) are the majority of the answer set. But comparing to 2019, the total number of answers in 2020 decreased.