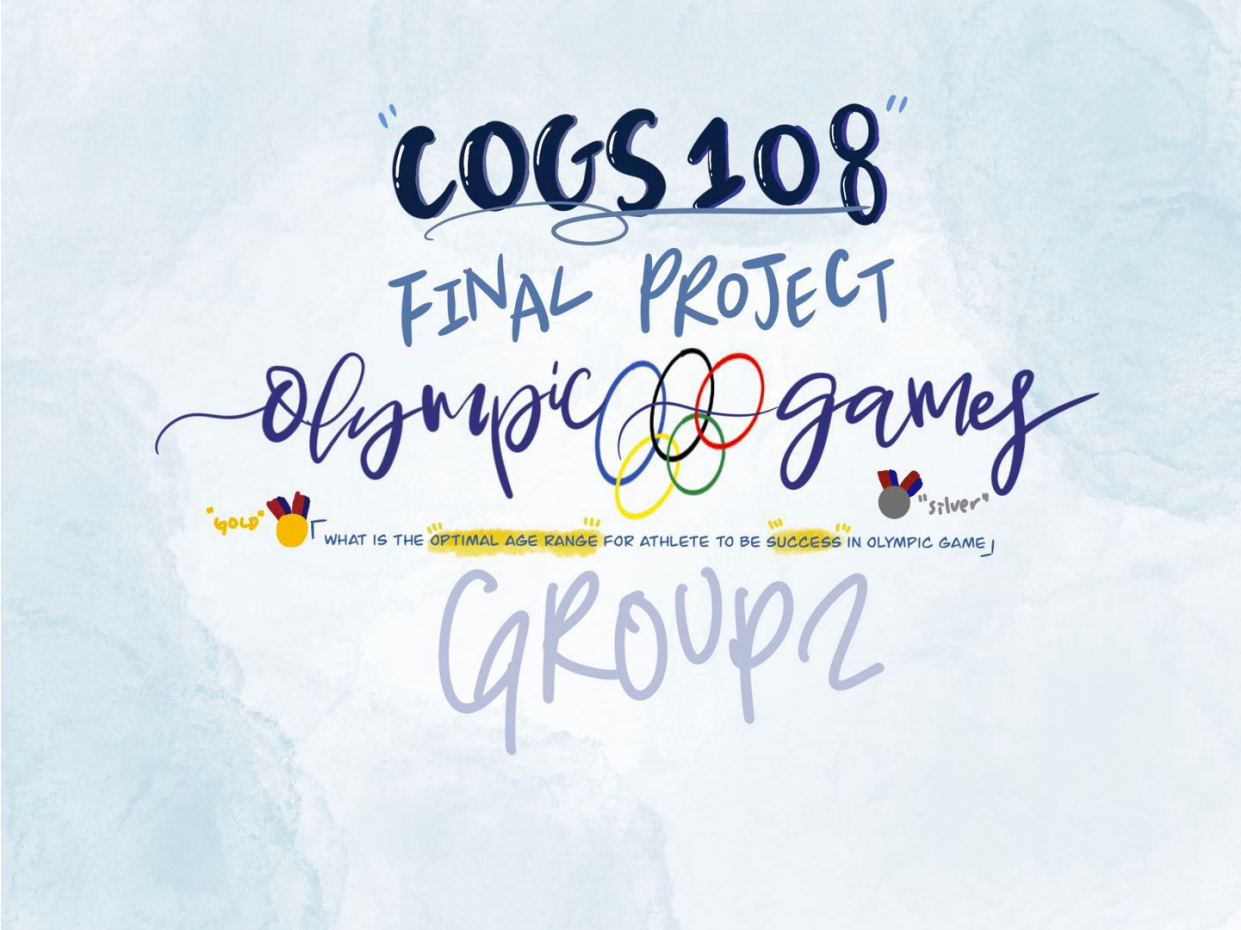
COGS 108 - Final Project ¶



Permissions

Place an X in the appropriate bracket below to specify if you would like your group's project to be made available to the public. (Note that PIDs will be scraped from the public submission, but student names will be included.)

- [x] YES - make available
- [ ] NO - keep private

Overview

Our project explores the athletes from Olympic Dataset from 1912 to 2016, where we attempt to identify the optical age range for athletes being sucessful. Specifically, we define being sucessful as whether or not an athlete won a medal. We split the original athletes dataset into two groups: athletes with medals and those without and subsequently plot data and perform statistical testing to try to determine the optimal age range. Next, we further define more specific optical age ranges for athletes with respect to sex and season using similar approach. Finally, we obtain multiple optical age ranges, where some are more general and others are more specific.

Names

- Ruoyu Liu
- Ting Yang Hung
- Yijun Liu
- Yiluo Qin
- Yu-chieh Chen

Group Members IDs

- A14857134
- A14899180
- A14476616
- A13997863
- A14388105

Research Question

What is the optimal age range for being successful, for example getting medals, at Olympic Games?

Background and Prior Work

Background

All the members on our team are genuinely interested in sports. For example, I am particularly a big fan of soccer, and my favourite soccer team is FC Barcelona, a spanish soccer team with super stars like Lionel Messi. Another team member was a promising swimmer, and his idol has always been Michael Phelps, who is considered to be one the greatest athletes in Olympic history. Therefore, we came together and decided we want to discover something interesting regarding sports. We then decided to explore Olympic datasets because the Olympics symbolizes the pinnacle of almost all kinds of sports events. The Olympics also is considered a global celebration for all the athletes from the globe to compete and to achieve higher standards. We were very excited to further explore this dataset and find interesting results.

Our hypothesis is that the optimal age to be successful at the Olympics is between 20 to 25 years old. Historically, according to many other popular sports such as basketball, soccer, swimming, and football. The blooming age of an athlete is around age 20. For example, most of the world records in swimming were set by swimmers in their early 20s. Additionally, most players who won their Ballon D'or(FIFA Best Player of the Year) were around 25 years old. As a result, we hypothesized that the most likely age range for an athlete to succeed at the Olympics is from 20 to 25 years old.

Testing our hypothesis is important because we could predict how long we possibly need to wait to see a given athlete to succeed at the Olympics. For those who have already won medals in the past Olympics, we could approximately predict how many years left for him/her to maintain at the highest competitive level. Furthermore, assuming our hypothesis is correct, we could even tell our friends to train harder if they want to compete at the highest sports events! In general, our hypothesis would serve as an indicator predicting a lot of useful information in many fields of sports.

Prior Work

From National Geographic Magazine[1], I noticed that although the article agrees that the majority of athletes won gold medals in their early and mid 20s, there were some great exceptions that simply prove age really is just a number. The article uses Michael Phelps as a great example to show that even people who retire from the Olympics can make a great comeback. As the article writes: “Thirty-one-year-old Michael Phelps won his 20th and 21st gold medals as part of this year's Olympics in Rio de Janeiro—impressive feats for a swimmer his age.” What really determines an athlete’s success is not age but really is his/her talent and how much commitment he/she put into the games and training.

According to TIME[2], there is “no specific age limit to compete in the Olympic Games.” Age restriction varies according to sports; however, we actually see a surge of young athletes compete in recent Olympics games. For example, American snowboarder, Chloe Kim took home a gold medalist in the 2018 Winter Olympics at age 17! Being a young athlete and has achieved such high performance, no doubt she would shine in the next decade. Definitely she has a bright future in front of her. Therefore, based on such statistics, it is actually possible the optimal age is getting younger and younger. Or we might witness a longer span of optimalgolden age. Instead of only 5 years like we hypothesize, the actual optimal age span might be larger.

With everything researched, we are very excited to delve into our dataset and find out the results ourselves.

References

- 1) Brady, Heather. “Meet the Olympians Who Prove Age Really Is Just a Number.” National Geographic, 11 Aug. 2016, www.nationalgeographic.com/news/2016/08/oldest-olympians-compete-games-2016-rio/#close (http://www.nationalgeographic.com/news/2016/08/oldest-olympians-compete-games-2016-rio/#close).
- 2) Quackenbush, Casey. “How Old Do You Have to Be to Compete in the Olympics?” Time, Time, 13 Feb. 2018, time.com/5154982/age-requirement-olympics-2018/.
- 3) Rgriffin. “120 Years of Olympic History: Athletes and Results.” Kaggle, 15 June 2018, www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results (http://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results).

Hypothesis

Under majority conditions, athletes get enough practice and have the physical conditions to achieve their peak. Therefore, we hypothesize the optimal age range is between 20 - 25 for athletes to be successful in Olympics games.

Dataset(s)

- Dataset Name: athelete_events
- Link to the dataset: https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results (https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results)
- Number of observations: 271116 observations, with a total of 15 attributes
- athlete_events dataset: This dataset includes a comprehensive documentation of all the atheletes' records in the competition. The dataset includes crucial attributes for answering our question such as ID, Name, Age, Event, and Medal. From this information, we are able to investigate the golden age for the atheletes.

- Metadata:

Column Name	Type	Description
ID	int	Unique number (id) for each athlete
Name	str	The name of the athlete's
Sex	str	M or F; The gender of the athlete
Age	int	The age of the athlete (depend on the participated year)
Height	float	The height of the athlete in centimeters
Weight	float	The weight of the athlete in kilograms
Team	str	Team name
NOC	str	National Olympic Committee in 3-letter code
Games	str	Year and season of the Olympics
Year	int	The year of the olympics
Season	str	Summer or Winter; Season type of the Olympics game
City	str	Host city
Sport	str	Sport type of the game that the athlete participated in
Event	str	The game event
Medal	str	Gold, Silver, Bronze, or NA; the prize that the athlete earned

Setup

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from matplotlib.patches import Polygon
```

Data Cleaning

Describe your data cleaning steps here.

In [2]:
```
# Read csv
athlete = pd.read_csv('athlete_events.csv')
athlete.head()
```

Out[2]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	E
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	A
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	

In [3]:
```
# replace medal to integer (Ordinal Encoding)
athlete['Medal'] = athlete['Medal'].replace({'Gold':3,'Silver':2,'Bror
```

In [4]:
```
# drop duplicate row
athlete = athlete[athlete.duplicated()==False]
```

In [5]:
```
# drop weight, height, games, team, city, id, and name columns
athlete.drop(columns=['Weight','Height','Games','Team','City','ID','Na
```

In [6]:
```
# add a column called got_medal ( 0 means the person doesn't get the m
athlete['got_medal']= athlete['Medal'].apply(lambda x: 0 if x == 0 els
```

In [7]:
```
# Get event title
athlete['Event'] = athlete.apply(lambda x: ''.join([i+' '
                                                    for i in x['Event'].split(' ')
```

In [8]:
```
# Check type
athlete.dtypes
```

Out[8]:
```
Sex           object
Age          float64
NOC           object
Year           int64
Season        object
Sport         object
Event         object
Medal          int64
got_medal      int64
dtype: object
```

In [9]:
```
# get the means of the age in different years that are null
athlete.assign(isNull = athlete['Age'].isna()).groupby('Year')['isNull
```

Out[9]:
```
Year
1896    0.428947
1900    0.405690
1904    0.210607
1906    0.428736
1908    0.207559
1912    0.038614
1920    0.196878
1924    0.198025
1928    0.176403
1932    0.108271
1936    0.027848
1948    0.157930
1952    0.029600
1956    0.099161
1960    0.023931
1964    0.005907
1968    0.011261
1972    0.008027
1976    0.004951
1980    0.020924
1984    0.018640
1988    0.007495
1992    0.002681
1994    0.000633
1996    0.000581
1998    0.000555
2000    0.000072
2002    0.000000
2004    0.000000
2006    0.000000
2008    0.000147
2010    0.000000
2012    0.000000
2014    0.000000
2016    0.000000
Name: isNull, dtype: float64
```

In [10]:
```
# only keeps the year that is after 1908
athlete = athlete[athlete['Year']> 1908]
```

In [11]:
```
# drop the columns that does not have age (age is np.NaN)
athlete.dropna(subset=['Age'],inplace=True)
```

In [12]:
```
athlete.head()
```

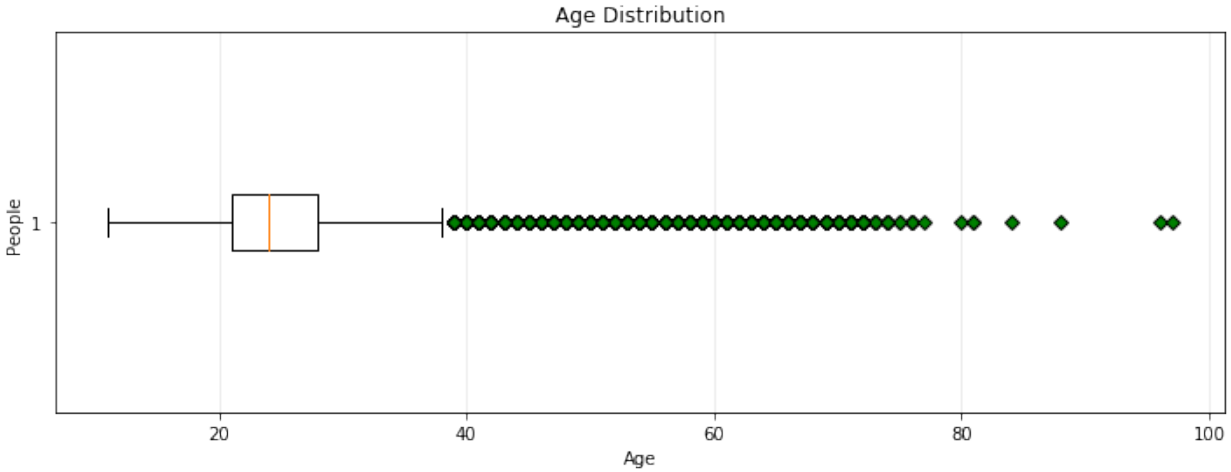
Out[12]:

	Sex	Age	NOC	Year	Season	Sport	Event	Medal	got_medal
0	M	24.0	CHN	1992	Summer	Basketball	Basketball	0	0
1	M	23.0	CHN	2012	Summer	Judo	Extra-Lightweight	0	0
2	M	24.0	DEN	1920	Summer	Football	Football	0	0
4	F	21.0	NED	1988	Winter	Speed Skating	500 metres	0	0
5	F	21.0	NED	1988	Winter	Speed Skating	1,000 metres	0	0

Optimal Age Boxplot

In [13]:
```
green_diamond = dict(markerfacecolor='g', marker='D')
fig3, ax1 = plt.subplots(figsize=(12,4))
ax1.boxplot(athlete['Age'], vert=False, flierprops=green_diamond)
ax1.xaxis.grid(True, linestyle='-', which='major', color='lightgrey',a
ax1.set_title('Age Distribution')
ax1.set_xlabel('Age')
ax1.set_ylabel('People')
```

Out[13]:
```
Text(0, 0.5, 'People')
```



In [14]:
```
print("Age 37.5 percentile:", np.percentile(athlete['Age'], 37.5, axis
print("Age 62.5 percentile:", np.percentile(athlete['Age'], 62.5, axis
```

```
Age 37.5 percentile: 23.0
Age 62.5 percentile: 26.0
```

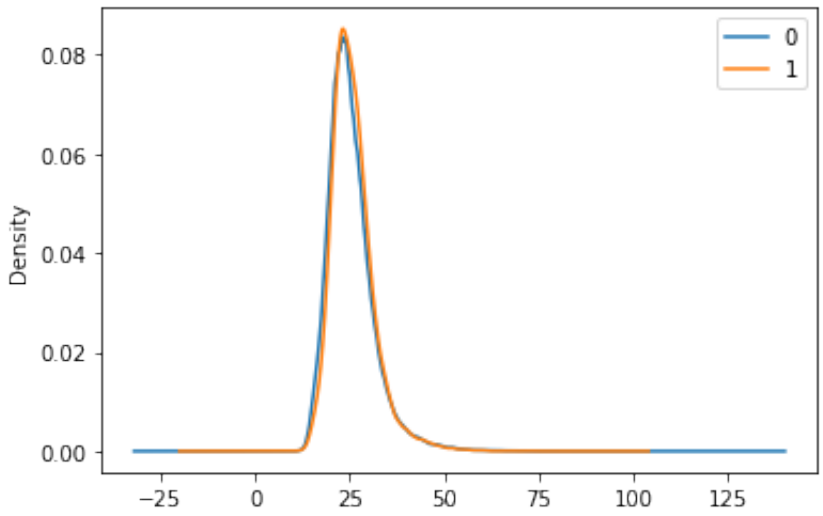
Majority of participants in Olympics have age range from 23 - 26.

Data Analysis & Results

Age

In [15]:
```
athlete.groupby('got_medal')['Age'].plot(kind='kde',legend=True)
```

Out[15]:
```
got_medal
0    AxesSubplot(0.125,0.125;0.775x0.755)
1    AxesSubplot(0.125,0.125;0.775x0.755)
Name: Age, dtype: object
```



In [16]:
```
got = athlete[athlete['got_medal']==1]['Age'].mean()
no_got = athlete[athlete['got_medal']==0]['Age'].mean()
got,no_got, got - no_got
```

Out[16]:
```
(25.84854547420785, 25.34127480081237, 0.5072706733954782)
```

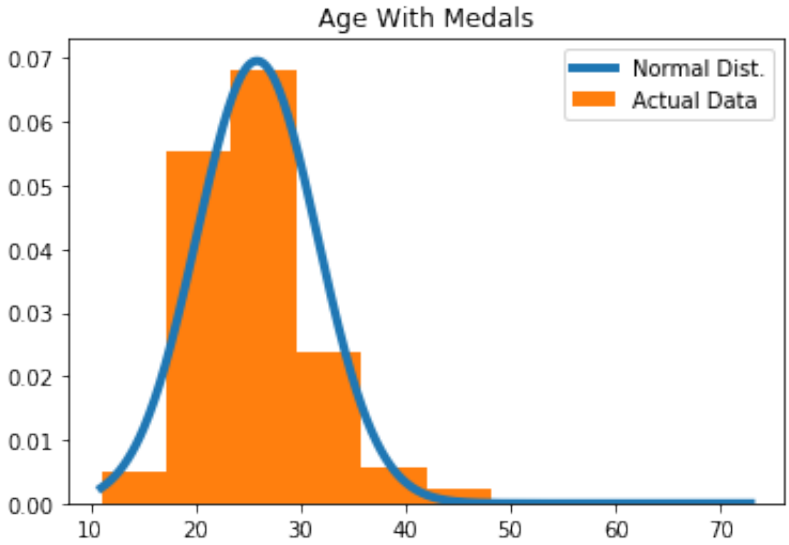
We find that the distributions of athletes who get medals and those don't get medals look very similar. However, the average age still have a difference about 0.5. Thus, we want to check if the 0.5 difference is a significant difference. We use t-test package and permutation test to double check if 0.5 is a significant difference.

Overview of Age Distribution between medals and no medals

Medal data: plot the comparison of the data and a normal distribution

This plots a histogram, with the hypothetical normal distribution (with same mean and variance)

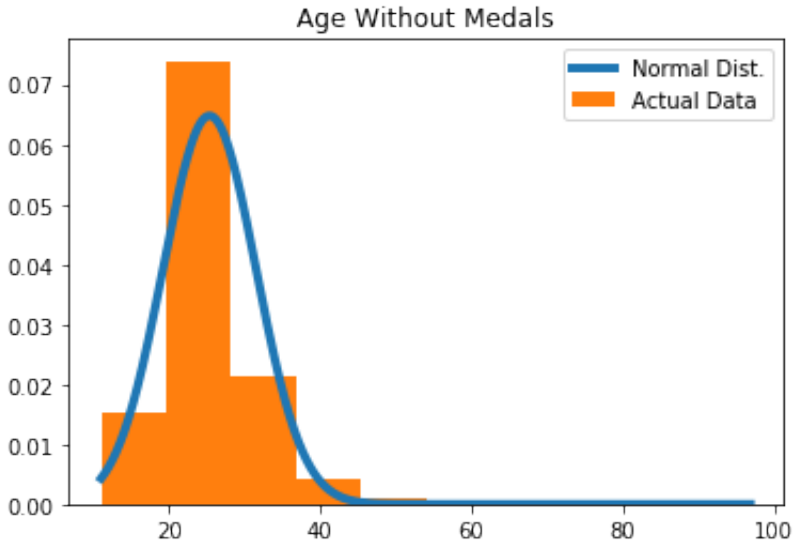
In [17]:
```
athlete_age_medals = athlete[athlete['got_medal']==1]['Age']
xs = np.arange(athlete_age_medals.min(), athlete_age_medals.max(), 0.1
fit = stats.norm.pdf(xs, np.mean(athlete_age_medals), np.std(athlete_a
plt.plot(xs, fit, label = 'Normal Dist.', lw = 4)
plt.hist(athlete_age_medals, density = True, label = 'Actual Data');
plt.title('Age With Medals')
plt.legend();
```



No medal data: plot the comparison of the data and a normal distribution

This plots a histogram, with the hypothetical normal distribution (with same mean and variance)

In [18]:
```
athlete_age_no_medals = athlete[athlete['got_medal']==0]['Age']
xs = np.arange(athlete_age_no_medals.min(), athlete_age_no_medals.max(
fit = stats.norm.pdf(xs, np.mean(athlete_age_no_medals), np.std(athlet
plt.plot(xs, fit, label = 'Normal Dist.', lw = 4)
plt.hist(athlete_age_no_medals, density = True, label = 'Actual Data')
plt.title('Age Without Medals')
plt.legend();
```



In [19]:
```
# Print out the average age for the two groups
print('Average age of people who get medal is \t\t\t {:2.2f} inches'.f
print('Average age of people who does not get medal is \t {:2.2f} inch
```

```
Average age of people who get medal is                  25.85 inches
Average age of people who does not get medal is         25.34 inches
```

From the above plots, we found that both athletes' ages with medals won and those with no medals won have a normal distribution. Therefore, for the later t-test, we can be assure that the resulting p value will deterministically tell us to either reject our null hypothesis. Later, we printed the means for both ditributions and figured out that there was a roughly 0.5 age difference. We will later perform the t-test to test if the age difference is significant or not.

Permutation Test

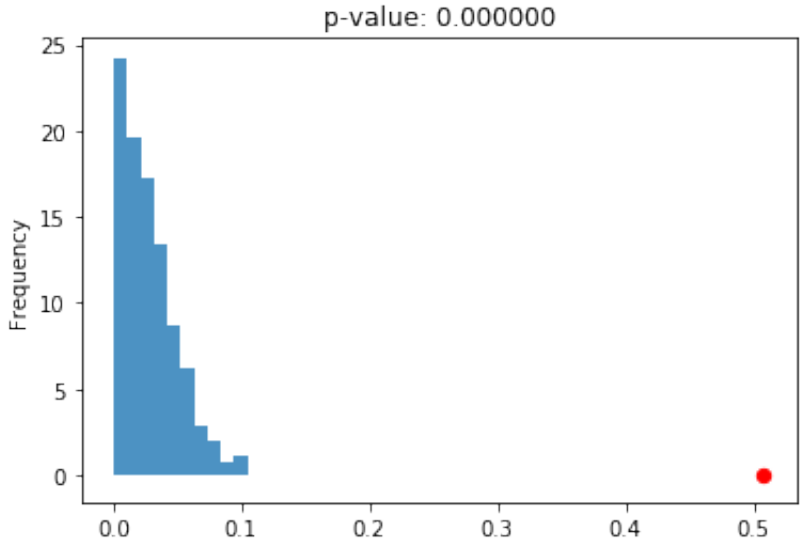
In [20]:
```
stats.ttest_ind(athlete_age_medals, athlete_age_no_medals)
```

Out[20]:
```
Ttest_indResult(statistic=14.789012546725141, pvalue=1.80831651697641
67e-49)
```

In addition, we did a t-test to see whether there is a difference in age distribution between athletes who won medals and those who did not. In this case, our null hypothesis stating that there is no difference in age distribution between athletes who won medals and athletes who did not. What we got for our p-value is close to zero, which suggests us to reject our null hypothesis and turns for our alternative hypothesis that there is a difference between the age distribution be between athletes who won medals and athletes who did not.

In [21]:
```
#### Hypothesis testingn for observed mean
n_repetitions = 500

means = []
for i in range(n_repetitions):
    # shuffle the gender column
    shuffled_col = (
        athlete['Age']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )

    # put them in a table
    shuffled = (
        athlete
        .assign(**{
            'Age': shuffled_col,
            'medal_not': athlete['Medal'] == 0
        })
    )

    # compute the differences in means
    mean = shuffled.groupby('medal_not')['Age'].mean().diff().abs().il

    means.append(mean)
```

In [22]:
```
obs = (athlete.loc[athlete.Medal!= 0].Age.mean() - athlete.loc[athlete
pval = np.mean(np.array(means) > obs)
pd.Series(means).plot(kind='hist', density=True, alpha=0.8, title='p-v
plt.scatter(obs, 0, color='red', s=40);
```



Permutation Test Conclusion

After we getting our data, we observed that there is an approximately 0.6 years old differences beween athletes who won medals and athletes who did not. Therefore, we want to discover whether there is a difference in age between these two groups, or more specifically, the 0.6 years old difference is meaningful.

- Null hypothesis -- there is no difference in age between athletes who won medals and athletes who did not.
- Alternative hypothesis -- there is a difference in age between athletes who won medals and athletes who did not.

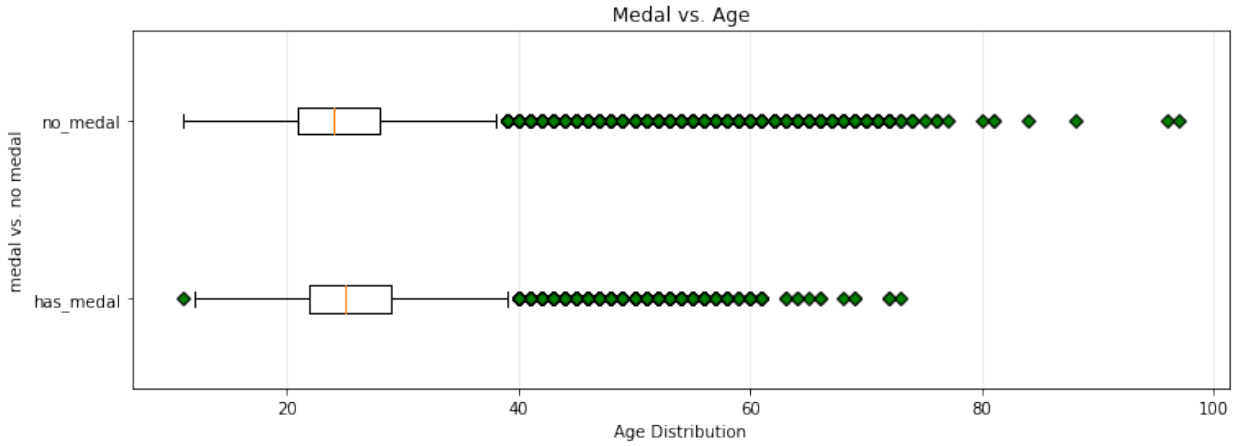
After deciding our Null Hypothesis and Alternative Hypothesis, we perform a permutation test, choosing a significance level = 0.05. This permutation test gives us a p-value = 0.0, which is smaller than our significance level, meaning we need to reject our null hypothesis, and concluding that there is a difference in ahe between athletes who won medals and athletes who did not.

Optimal Age Ranges for Medal vs No Medal

In [23]:
```
age_medal = athlete[athlete['got_medal'] == 1]['Age']
age_no_medal = athlete[athlete['got_medal'] == 0]['Age']
data = [age_medal, age_no_medal]
```

In [24]:
```
green_diamond = dict(markerfacecolor='g', marker='D')
fig3, ax1 = plt.subplots(figsize=(12,4))
ax1.boxplot(data, vert=False, flierprops=green_diamond)
ax1.xaxis.grid(True, linestyle='-', which='major', color='lightgrey',a
ax1.set_title('Medal vs. Age')
ax1.set_xlabel('Age Distribution')
ax1.set_ylabel('medal vs. no medal')
plt.yticks(np.arange(1,3), ('has_medal','no_medal'))
```

Out[24]:
```
([<matplotlib.axis.YTick at 0x1a2482f510>,
  <matplotlib.axis.YTick at 0x181ed7b750>],
 <a list of 2 Text yticklabel objects>)
```



In [25]:
```
print("medal's 37.5 percentile:", np.percentile(age_medal, 37.5, axis
print("medal's 62.5 percentile:", np.percentile(age_medal, 62.5, axis

print("no medal's 37.5 percentile:", np.percentile(age_no_medal, 37.5,
print("no medal's 62.5 percentile:", np.percentile(age_no_medal, 62.5,
```

```
medal's 37.5 percentile: 24.0
medal's 62.5 percentile: 27.0
no medal's 37.5 percentile: 23.0
no medal's 62.5 percentile: 26.0
```

The optimal age range for getting `medals` is 24 - 27.

Gender

In [26]:
```
gender = athlete[athlete['got_medal']==False].groupby(['Sex'])['Age'].
gender_medal = athlete[athlete['got_medal']==True].groupby(['Sex'])['A
print('athletes who do not get medals')
display(gender.set_index('Sex'))
print('athletes who get medals')
display(gender_medal.set_index('Sex'))
fig, axes = plt.subplots(1,2)
axes[0].bar(gender['Sex'],gender['Age'])
axes[1].bar(gender_medal['Sex'],gender_medal['Age'])
```

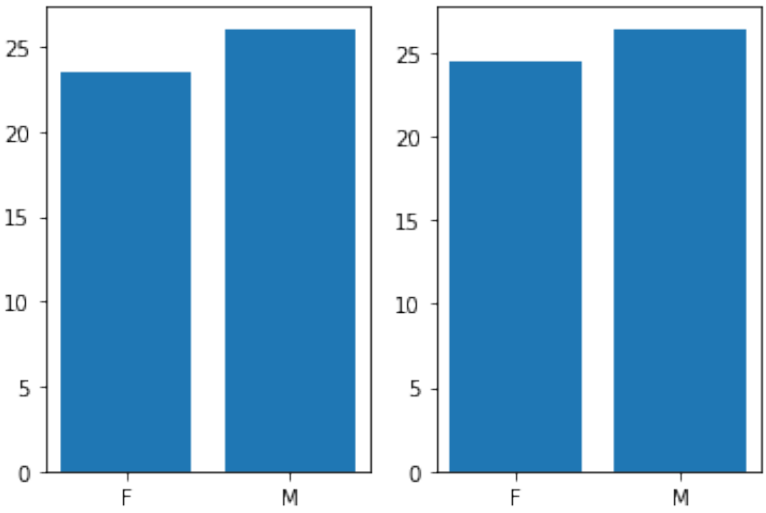
```
athletes who do not get medals
```

Sex	Age
F	23.545034
M	26.068144

```
athletes who get medals
```

Sex	Age
F	24.473167
M	26.445810

Out[26]:
```
<BarContainer object of 2 artists>
```



Here, the bar chart indicates that on average, within those two groups -- athletes with medals and athletes without, we found that males tend to have a higher average age than females.

In [27]:
```
## for with medals
with_medals = athlete[athlete['got_medal']==1]
stats.ttest_ind(with_medals.loc[with_medals['Sex'] == 'F']['Age'].drop
                with_medals.loc[with_medals['Sex'] == 'M']['Age'].drop
```

Out[27]:
```
Ttest_indResult(statistic=-30.791569409443806, pvalue=1.3531323792160
763e-205)
```

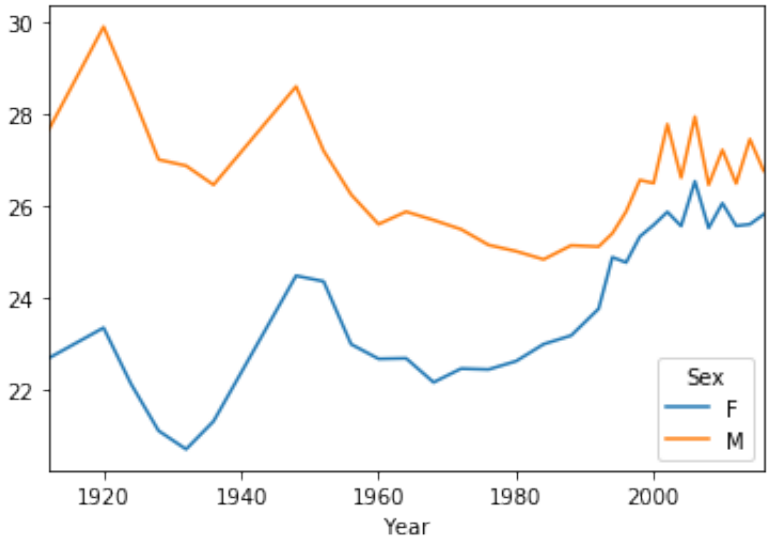
In [28]:
```
## for without medals
without_medals = athlete[athlete['got_medal']==0]
stats.ttest_ind(without_medals.loc[without_medals['Sex'] == 'F']['Age'
                without_medals.loc[without_medals['Sex'] == 'M']['Age'
```

Out[28]:
```
Ttest_indResult(statistic=-88.08056048165984, pvalue=0.0)
```

From the t-test, we notice that, no matter for athletes with or without medals, the age distribution between Sex has a difference. In this case, our null hypothesis states that there is no difference in age distribution between females and males. What we got for our p-value are both close to 0, no matter for athletes with medals or those without medals. Therefore, in both groups (athletes with medals or athletes without medals), we conclude that there is a differences in age distribution between females and males.

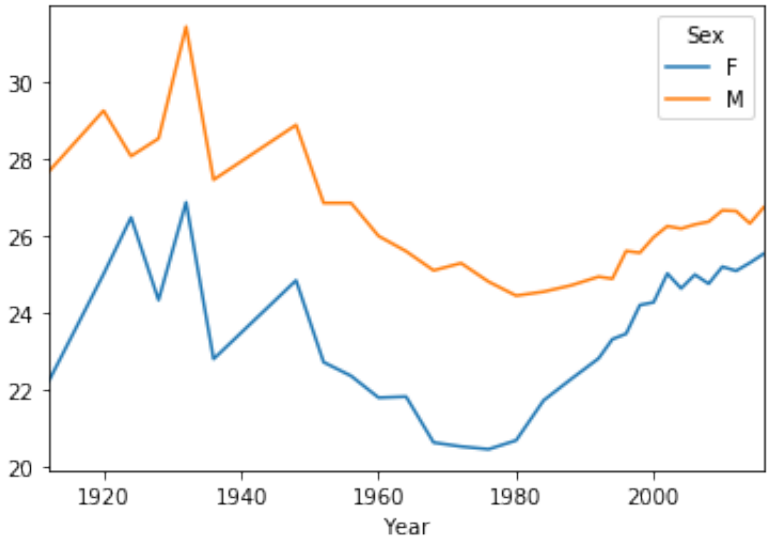
In [29]:
```
gender = athlete[athlete['got_medal']==True].pivot_table(columns = 'Se
gender.plot()
```

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1fb43350>



In [30]:
```
gender = athlete[athlete['got_medal'] == False].pivot_table(columns =
gender.plot()
```

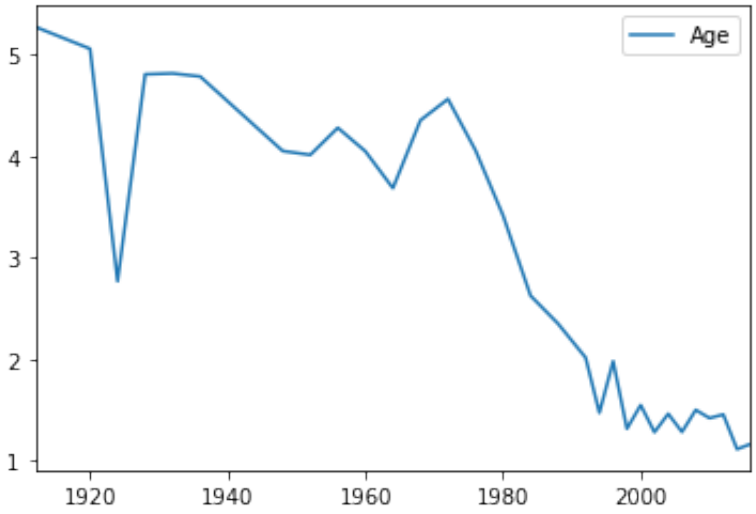
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x1a201c1190>



From the above plots, we observed that as the years pass by, the average age differences between female athletes and male athletes are shrinking, no matter for the athletes won medals or not.

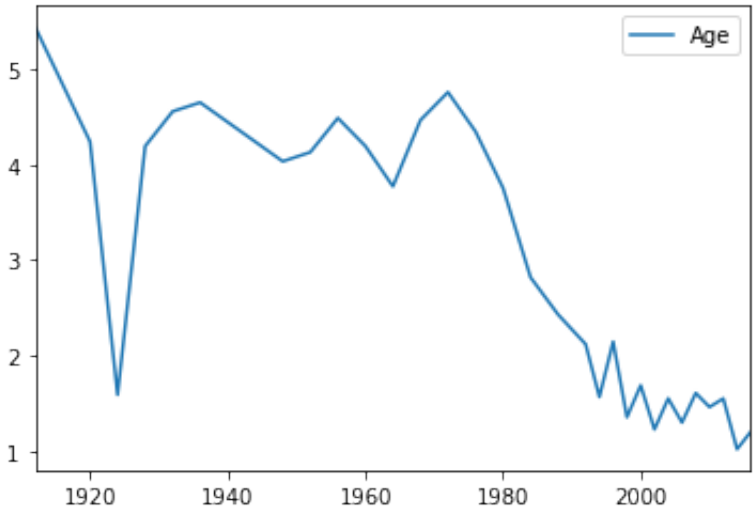
In [31]:
```
age_diff = athlete.groupby(['Year','Sex'])['Age'].mean().reset_index()
age_diff.groupby('Year')['Age'].diff().dropna().to_frame().set_index(a
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1ffeead0>



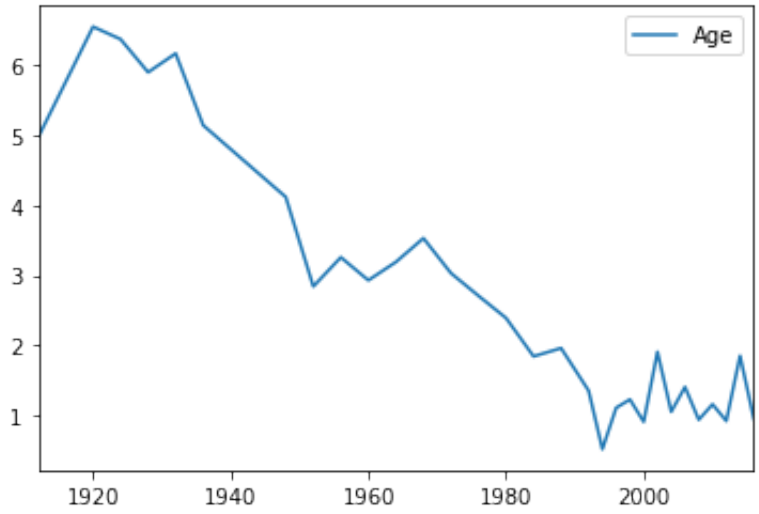
In [32]:
```
age_diff = athlete[athlete['got_medal']==False].groupby(['Year','Sex']
age_diff.groupby('Year')['Age'].diff().dropna().to_frame().set_index(a
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x1a202006d0>



In [33]:
```
age_diff = athlete[athlete['got_medal']==True].groupby(['Year','Sex'])
age_diff.groupby('Year')['Age'].diff().dropna().to_frame().set_index(a
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1a254b2310>



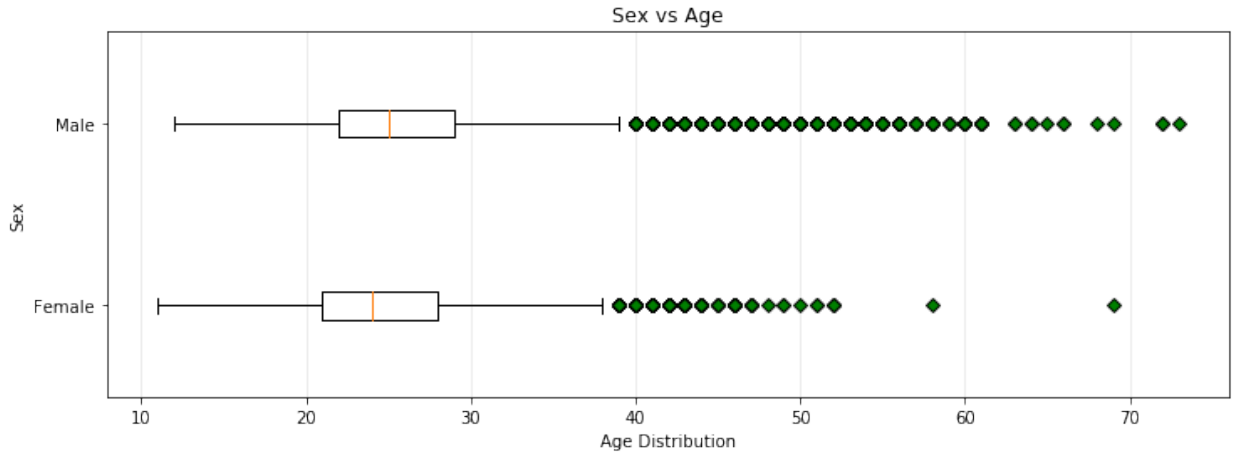
These two plots further justify the age difference between sex is decreasing in the 2 groups (athletes with medals and athletes without medals).

Optimal Age Ranges for Female vs Male

In [34]:
```
age_female = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']=='F')
age_male = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']=='M')]
data = [age_female, age_male]
```

In [35]:
```
green_diamond = dict(markerfacecolor='g', marker='D')
fig3, ax1 = plt.subplots(figsize=(12,4))
ax1.boxplot(data, vert=False, flierprops=green_diamond)
# Add a horizontal grid to the plot, but make it very light in color
# so we can use it for reading data values but not be distracting
ax1.xaxis.grid(True, linestyle='-', which='major', color='lightgrey',a
# Hide these grid behind plot objects
ax1.set_title('Sex vs Age')
ax1.set_xlabel('Age Distribution')
ax1.set_ylabel('Sex')
plt.yticks(np.arange(1,3), ('Female', 'Male'))
```

Out[35]:
```
([<matplotlib.axis.YTick at 0x1a24820510>,
  <matplotlib.axis.YTick at 0x1a2494aa50>],
 <a list of 2 Text yticklabel objects>)
```



In [36]:
```
print("Female's 37.5 percentile:", np.percentile(age_female, 37.5, axi
print("Female's 62.5 percentile:", np.percentile(age_female, 62.5, axi

print("Male's 37.5 percentile:", np.percentile(age_male, 37.5, axis =
print("Male's 62.5 percentile:", np.percentile(age_male, 62.5, axis =
```

```
Female's 37.5 percentile: 23.0
Female's 62.5 percentile: 26.0
Male's 37.5 percentile: 24.0
Male's 62.5 percentile: 27.0
```

Findings:

- **The optimal age for `female` getting medals is 23-26**
- **The optimal age for `male` getting medals is 24-27**

Season

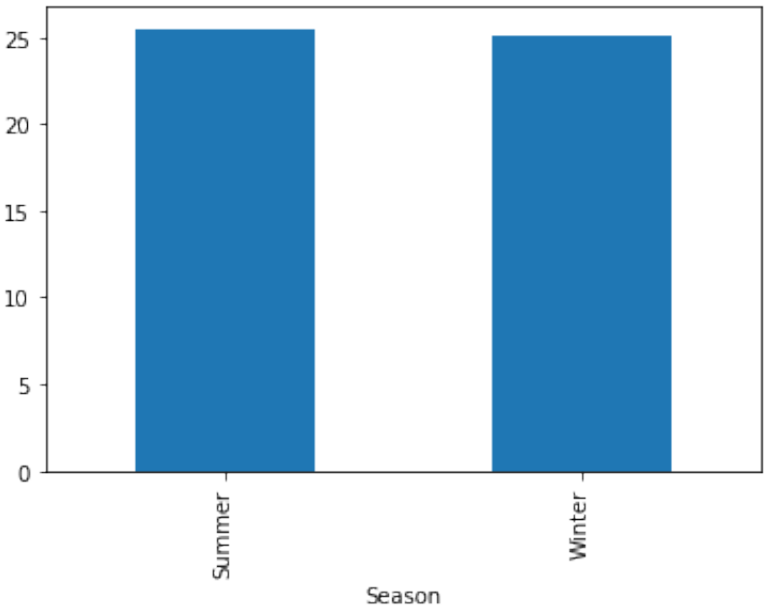
a) Age & Season

In this part, we try to discover the average age differences between Summer and Winter Olympics and try to test whether the age distributions between Summer and Winter Olympics have significant differences.

In [37]:
```
season_df = athlete.groupby('Season')['Age'].mean()
display(season_df)
season_df.plot(kind='bar')
```

```
Season
Summer    25.502896
Winter    25.039147
Name: Age, dtype: float64
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x1a24a0f790>



This bar chart suggests that the average ages for both summer and winter are very similar. There only exists an approximately 0.5 age difference between summer and winter groups, slightly suggesting there is a difference in age distribution between these two groups. To further discover whether there is a difference, we perform following t-test.

In [38]:
```
## for general age distribution between seasons
stats.ttest_ind(athlete.loc[athlete['Season'] == 'Summer']['Age'].drop
                athlete.loc[athlete['Season'] == 'Winter']['Age'].drop
```

Out[38]:
```
Ttest_indResult(statistic=15.040707692283027, pvalue=4.17925512263208
9e-51)
```

This t-test justifies that we *reject* our null hypothesis that there is no difference between age distributions in summer and winter Olympic seasons. We are in favor of our alternative hypothesis, which states that the age distribution between summer and winter Olympic seasons is different.

To dive deeper into athletes who won medals in Olympics with perspective to Summer and Winter Seasons, we did following exploration.

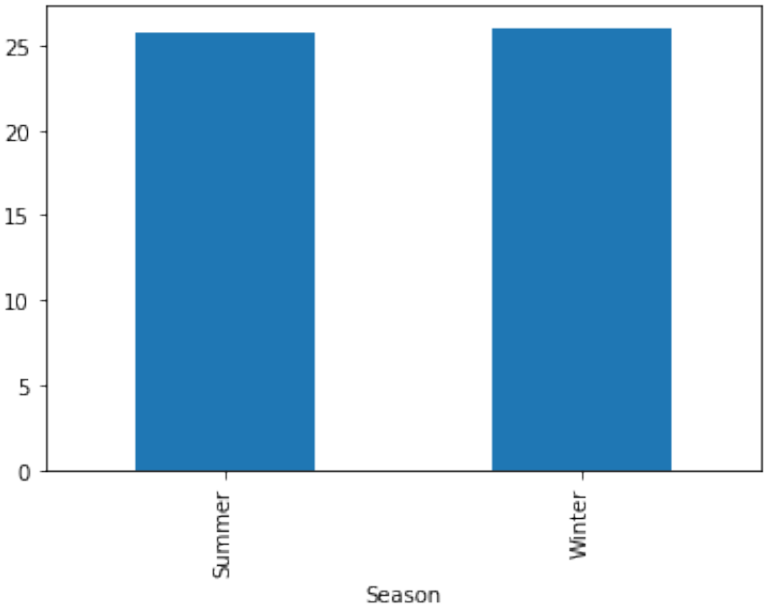
b) Age & Seasons for Atheltes with Medals

Under the group of athletes with Medals, we try to discover whether there is difference in age distribution:

In [39]:
```
season_df = athlete[athlete['got_medal']==1].groupby(['Season'])['Age'
display(season_df)
season_df.plot(kind='bar')
```

```
Season
Summer    25.809481
Winter    26.063688
Name: Age, dtype: float64
```

Out[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2abd7850>
```



This bar chart suggests that the differences in average age between Summer and Winter seasons is small, and approximately equal to 2 years old. To further discover whether there exists a differences in age distribution, we did following t-test:

In [40]:
```
## for summer, t-test for athletes got medal or not
stats.ttest_ind(athlete[athlete['got_medal']==1].loc[with_medals['Seas
                athlete[athlete['got_medal']==1].loc[with_medals['Seas
```

Out[40]:
```
Ttest_indResult(statistic=-3.075735419606377, pvalue=0.00210137255599
842)
```

In this case, our t-test gives back a pvalue = 0.002, which is smaller than the general significance level =0.05, therefore, we *reject* our Null hypothesis that there is no differences in age distribution between Summer and Winter Olympic seasons for athletes who won medals.

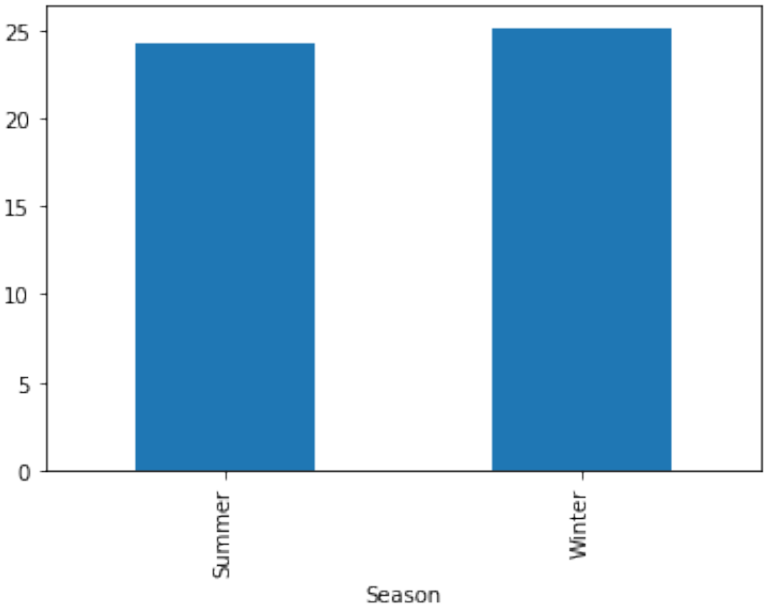
c) Age & Seasons & Sex for Atheltes with Medals

In this section, we are trying to see whether there is a difference between age distribution with perspective to `Sex` and `Season` for athletes with medals

In [41]:
```
print('Female')
season_df = athlete[(athlete['got_medal']==1) & (athlete['Sex']=='F')]
display(season_df)
season_df.plot(kind='bar')
```

```
Female

Season
Summer    24.335535
Winter    25.186637
Name: Age, dtype: float64
```

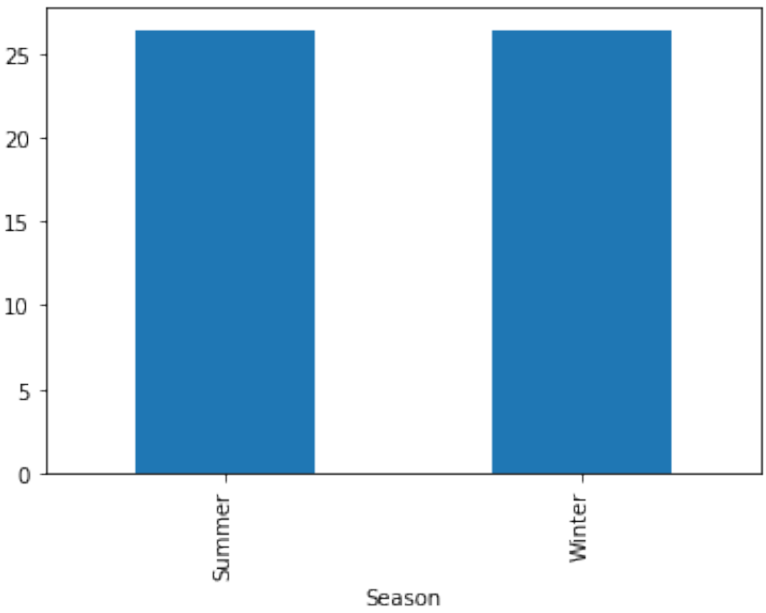
Out[41]: `<matplotlib.axes._subplots.AxesSubplot at 0x1a2aa6af10>`



In [42]:
```
print('Male')
season_df = athlete[(athlete['got_medal']==1) & (athlete['Sex']=='M')]
display(season_df)
season_df.plot(kind='bar')
```

```
Male

Season
Summer    26.440865
Winter    26.473793
Name: Age, dtype: float64
```

Out[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a27dac490>
```



The above 2 bar chart indicate that, for females, there exists an approximately 1 year difference in average age between Summer and Winter seasons for atheltes who won medals. For males, there exists an approximately 0.03 year difference. Therefore, we are curious about whether this specific difference suggests there are differences in age distributions for these two groups. Therefore, we perform following t-tests:

In [43]:
```
print('Female')
females_w_medals = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']
stats.ttest_ind(females_w_medals.loc[females_w_medals['Season'] == 'Su
                females_w_medals.loc[females_w_medals['Season'] == 'Wi
```

```
Female
```

Out[43]:
```
Ttest_indResult(statistic=-6.402427599851293, pvalue=1.59044660845551
13e-10)
```

In [44]:
```
print('Male')
males_w_medals = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']==
stats.ttest_ind(males_w_medals.loc[males_w_medals['Season'] == 'Summer
                males_w_medals.loc[males_w_medals['Season'] == 'Winter
```

```
Male
```

Out[44]:
```
Ttest_indResult(statistic=-0.32268241640130196, pvalue=0.746938386197
3562)
```

Interesting findings here:

- **For females, we *reject* our Null hypothesis and conclude that there has differences in age distribution with perspective to Summer or Winter seasons for female athletes who won medals, as the pvalue is approximately to 0;**
- **However, for males, there is no difference in age distribution with perspective to Summer or Winter seasons for athletes who won medals since the p-value we got is 0.74, which is far larger than the genral significance level 0.05. Therefore, in the male case, we *cannot reject* our Null hypothesis, and conclude that, there is no difference in age distribution with perspective to Summer and Winter seasons for male athletes who won medals.**

Such interesting findings lead us to wonder: why when combining genders, the t-tests show different result. So, we consider one specific sport, gymnastics, to discover whether the age for participating in sports, for example, gymnastics which only hold in summer, will result such differences.

In [45]:
```
athlete.loc[(athlete['got_medal'] == 1) & (athlete['Sport'] == 'Gymnas
```

Out[45]:
```
Season
Summer    1969
dtype: int64
```

In [46]:
```
athlete.loc[(athlete['got_medal'] == 1) & (athlete['Sport'] == 'Gymnas
```

Out[46]:
```
Sex
F    20.200000
M    25.022763
Name: Age, dtype: float64
```

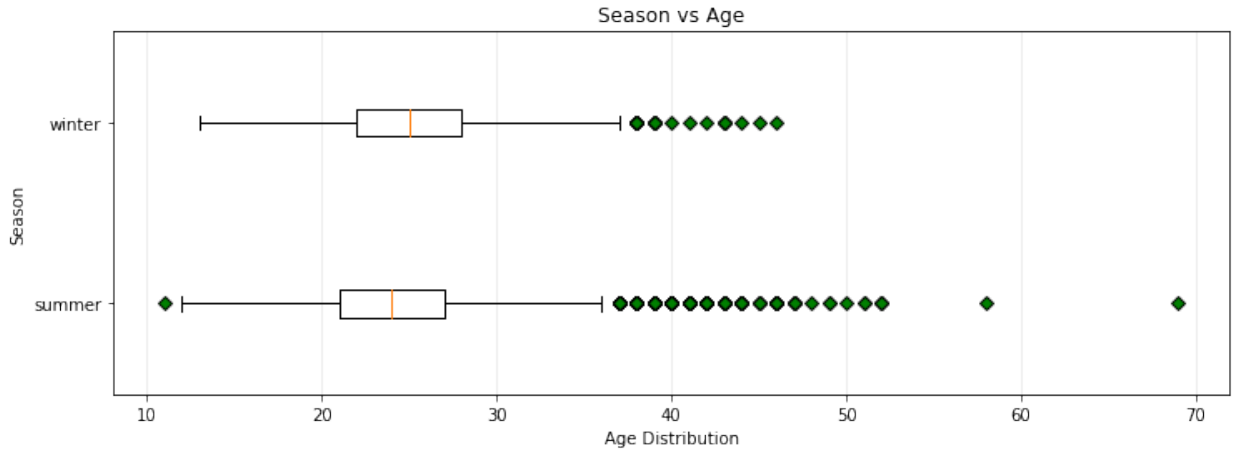
The above statistic indicates that, for gymnastics, female athletes who won medals tends to had a smaller age than male athletes who won medals. As the data shows, female athletes who won medals is around `20`, which is much smaller than the average age `24.4` for females who won medals. However, male athletes who won medals is around `25`, which is much closer to `26.4` than female's data. Thus, we understand that the reason why female get different optimal age ranges for different season is due to the fact that some sports like gymnastics which only hold in summer have a smaller average age than female overall.

Optimal Age Ranges for Winter vs Summer

In [47]:
```
summer_f = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']=='F')&(
winter_f = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']=='F')&(
data = [summer_f, winter_f]
```

In [48]:
```
green_diamond = dict(markerfacecolor='g', marker='D')
fig3, ax1 = plt.subplots(figsize=(12,4))
ax1.boxplot(data, vert=False, flierprops=green_diamond)
# Add a horizontal grid to the plot, but make it very light in color
# so we can use it for reading data values but not be distracting
ax1.xaxis.grid(True, linestyle='-', which='major', color='lightgrey',a
# Hide these grid behind plot objects
ax1.set_title('Season vs Age')
ax1.set_xlabel('Age Distribution')
ax1.set_ylabel('Season')
plt.yticks(np.arange(1,3), ('summer', 'winter'))
```

Out[48]:
```
([<matplotlib.axis.YTick at 0x1a2535c650>,
  <matplotlib.axis.YTick at 0x1a25357d90>],
 <a list of 2 Text yticklabel objects>)
```



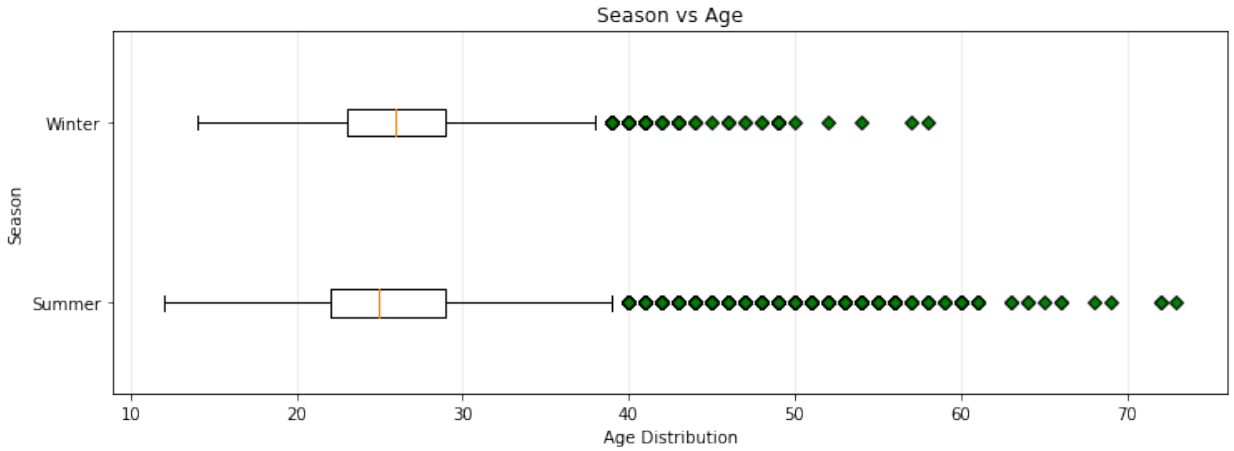
In [49]:
```
print("Summer female's 37.5 percentile:", np.percentile(summer_f, 37.5
print("Summer female's 62.5 percentile:", np.percentile(summer_f, 62.5

print("Winter female's 37.5 percentile:", np.percentile(winter_f, 37.5
print("Winter female's 62.5 percentile:", np.percentile(winter_f, 62.5
```

```
Summer female's 37.5 percentile: 22.0
Summer female's 62.5 percentile: 26.0
Winter female's 37.5 percentile: 23.0
Winter female's 62.5 percentile: 26.0
```

In [50]:
```
summer_m = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']=='M')&(
winter_m = athlete[(athlete['got_medal'] == 1)&(athlete['Sex']=='M')&(
data = [summer_m, winter_m]
```

In [51]:
```
green_diamond = dict(markerfacecolor='g', marker='D')
fig3, ax1 = plt.subplots(figsize=(12,4))
ax1.boxplot(data, vert=False, flierprops=green_diamond)
# Add a horizontal grid to the plot, but make it very light in color
# so we can use it for reading data values but not be distracting
ax1.xaxis.grid(True, linestyle='-', which='major', color='lightgrey',a
# Hide these grid behind plot objects
ax1.set_title('Season vs Age')
ax1.set_xlabel('Age Distribution')
ax1.set_ylabel('Season')
plt.yticks(np.arange(1,3), ('Summer', 'Winter'))
```

Out[51]:
```
([<matplotlib.axis.YTick at 0x1a27f0aad0>,
  <matplotlib.axis.YTick at 0x1a27f0a250>],
 <a list of 2 Text yticklabel objects>)
```



In [52]:
```
print("Summer male's 37.5 percentile:", np.percentile(summer_m, 37.5,
print("Summer male's 62.5 percentile:", np.percentile(summer_m, 62.5,

print("Winter male's 37.5 percentile:", np.percentile(winter_m, 37.5,
print("Winter male's 62.5 percentile:", np.percentile(winter_m, 62.5,
```

```
Summer male's 37.5 percentile: 24.0
Summer male's 62.5 percentile: 27.0
Winter male's 37.5 percentile: 24.0
Winter male's 62.5 percentile: 27.0
```

- The optimal age range for `Female` participating in `Summer` Olympic is 22-26
- The optimal age range for `Female` participating in `Winter` Olympic is 23-26
- The optimal age range for `Male` participating in `Summer` Olympic is 24-27
- The optimal age range for `Male` participating in `Winter` Olympic is 24-27

Ethics & Privacy

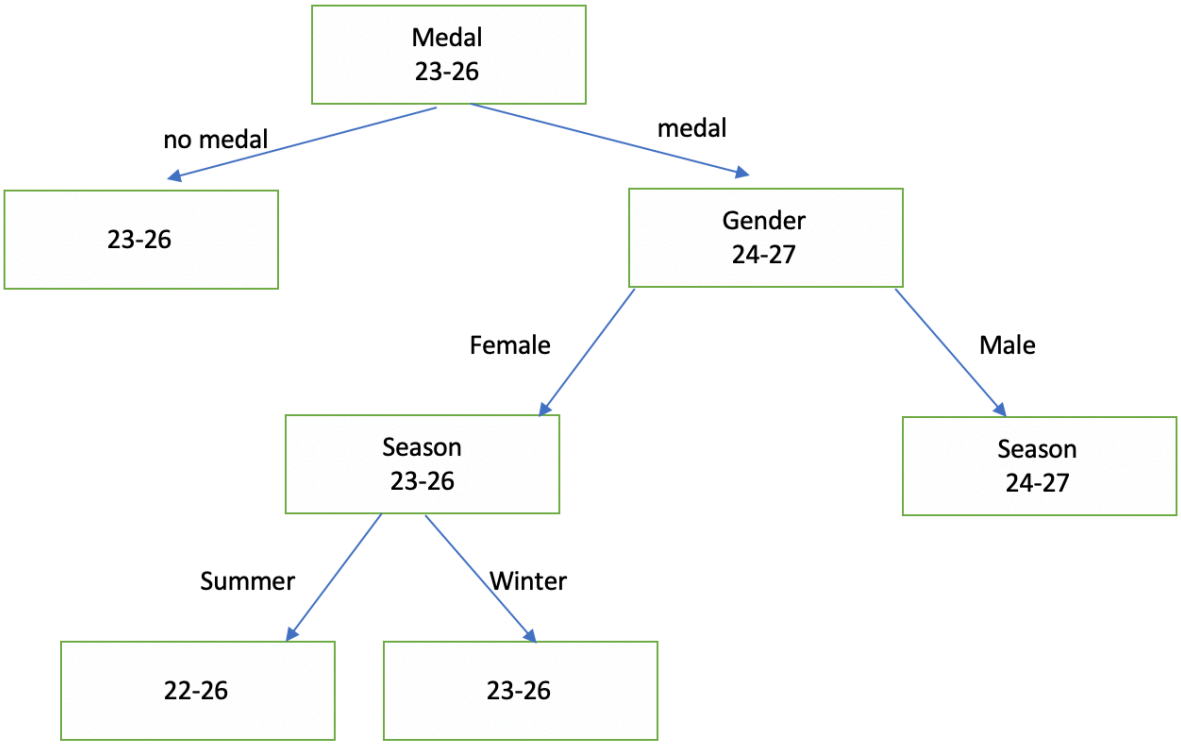
The data is from Kaggle, which is an open-source platform. Therefore, we have the permission to use this dataset and there is no privacy concern to deal with. Also, the Olympic Games are internationalized and the information of the athletes are shared to the public, there are no privacy issues about recording and analyzing these open data, including events they participated in, medals they earned, or the countries they represented.

However, since the date for this dataset varies from 1896 to 2016, the data for weight and height from the earlier years may not be accurate due to the malfunctioness of machines in those years. Also, the name for athletes might not be accurate and our goal has nothing to deal with names of athletes. So, we would like to drop `name`, `weight`, and `height` columns from our dataset.

What's more, the dataset might have a gender bias in earlier years. For instance, in 1896, women represented 0% of the dataset. Women began to appear in 1900. Until 1996, the percentage of women in those data entries reached about 50%. Thus, the dataset has a clear gender bias before 1996. Therefore, to be more accurate in analysis, we decide to separate females from males and to discover the “optimal age” for each group.

If our analysis justifies that athletes between age 20 and 25 could win the most medals, this will possibly lead to the discriminations on athletes who were not in this age range between age 20 and 25. Also, this research might cause talented athletes outside this age range being overlooked by sponsers and failed in their paths as athletes on Olympics Games.

Optimal Age Range Decision Tree



Conclusion & Discussion

Our project focuses on finding the optimal age of athletes in the Olympic games. We attempt to verify the correctness of our hypothesis: The optimal age is between 20 - 25 for athletes to get successful in the Olympic games. We find a dataset on Kaggle and use it as our dataset for this analysis. Our leading question is “What is the optimal age for athletes to be successful”. In our analysis, we clean the dataset by dropping the duplicate rows and columns that are not related to our hypothesis. In addition, we check the number of NaN values in the ‘Age’ column and decide to drop the years that have a high proportion of NaN values.

For data analysis, we first use the “Age” column to find out the optimal age range. We then separate the dataset into ‘medal’ and ‘no medal’ to define what we mean by successful. We find out a 0.5 age difference and then using a permutation test and t-test to prove that 0.5 is a significant difference. Thus, we decide that getting a medal means being successful. Later on, we develop data analysis in the ‘Gender’ and ‘Season’ category. We decide to use the middle 25% athletes as our optimal age range since there are many outliners. As a result, the overall optimal age is around 24 - 27. However, the female’s optimal age range is smaller than the male’s, which is 23 - 26 and 24 - 27 accordingly. In addition, we further calculate a more specific optical age ranges with respect to Sex and Season.

Our analysis has some limitations. First of all, it is hard to do a deeper analysis of the ‘Sport’ or ‘NOC’ column since they have too many unique values. Secondly, our data is not completed. We decide to drop the NaN in the ‘Age’ column, which reduces the size of our dataset. In addition, since most of the data are categorical data, it is not suitable to use the regression model like linear regression. Lastly, since the Olympics only happen once every four years, it is extremely hard to define the true optimal age range because of the gap . In the upcoming Tokyo Olympics game, our predicted optimal age range might be a helpful resource to determine the athletes who can be successful.

Team Contributions

- Rouyu Liu : She offers a significant amount of contributions on the conceptual framework. For instance, She helped to define ways of approaching our hypothesis through statistics.
- Ting-Yang Hung : He contributed on data visualizations and statistical testings (t-test, permutation-test) as well as explaining the outcome obtained from each testing in a simplified manner.
- Yijun Liu : She contributed mostly on visualization the data through plotting as well as explaining what each plot represents. In addition, she had written codes for most of the hypothesis/permutation tests.
- Yiluo Qin : He contributed mostly on plotting the data as well as offering suggestions on which plot option can best convey that ideas that we are trying to deliver across to the audiances.
- Yu-Chieh Chen : She did most of the data cleaning. For instance, she dropped the unncessary columns that are not related to the hypothesis and replace the null values to a for meaningful value(such as ordinal encoding). Also, she plotted many plots for data as well as polishing the plots.