You're working as a sports journalist at a major online sports media company, specializing in soccer analysis and reporting. You've been watching both men's and women's international soccer matches for a number of years, and your gut instinct tells you that more goals are scored in women's international football matches than men's. This would make an interesting investigative article that your subscribers are bound to love, but you'll need to perform a valid statistical hypothesis test to be sure!

While scoping this project, you acknowledge that the sport has changed a lot over the years, and performances likely vary a lot depending on the tournament, so you decide to limit the data used in the analysis to only official `FIFA World Cup` matches (not including qualifiers) since `2002-01-01`.

You create two datasets containing the results of every official men's and women's international football match since the 19th century, which you scraped from a reliable online source. This data is stored in two CSV files: `women_results.csv` and `men_results.csv`.

The question you are trying to determine the answer to is:

> Are more goals scored in women's international soccer matches than men's?

You assume a **10% significance level**, and use the following null and alternative hypotheses:

H_0 : The mean number of goals scored in women's international soccer matches is the same as men's.

H_A : The mean number of goals scored in women's international soccer matches is greater than men's.

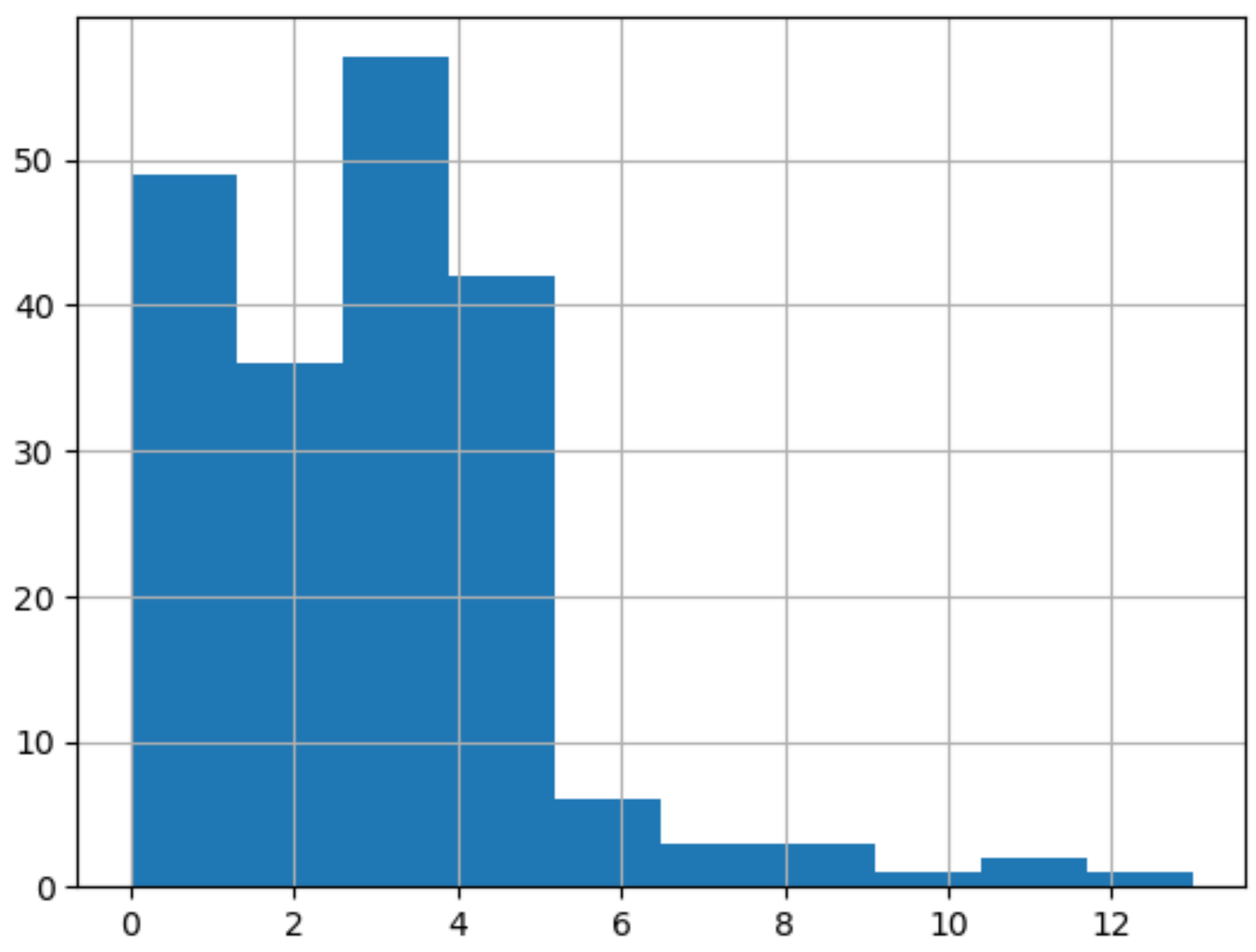
In [5]:

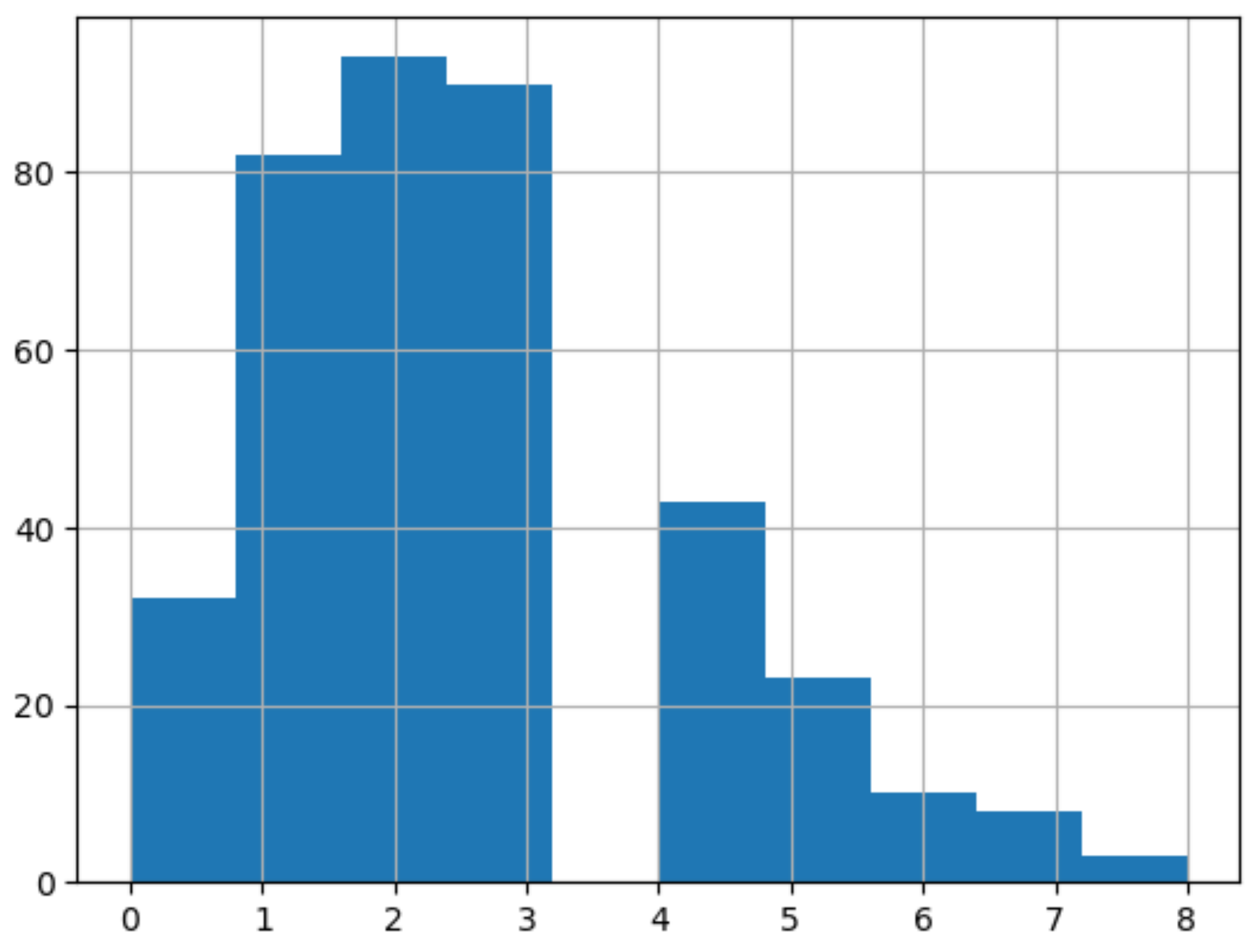
```
# Start your code here!
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

women_results = pd.read_csv("women_results.csv")
men_results = pd.read_csv("men_results.csv")

# to filter based on date, we need to convert the [date] column to time

women_results['date']= pd.to_datetime(women_results['date'])
men_results['date'] = pd.to_datetime(men_results['date'])

filter_w = (women_results['date'] >= '2002-01-01') & (women_results['tournament'] == 'FIFA World Cup')
filter_m = (men_results['date'] >= '2002-01-01') & (men_results['tournament'] == 'FIFA World Cup')

women = women_results[filter_w]
women['total_score'] = women['home_score'] + women['away_score']
women['group'] ='women'
men = men_results[filter_m]
men['total_score'] = men['home_score'] + men['away_score']
men['group'] = 'men'
```

```
/var/folders/yj/53qmn0w907qf_67npl86j5tr0000gn/T/ipykernel_44250/658476920.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  women['total_score'] = women['home_score'] + women['away_score']
/var/folders/yj/53qmn0w907qf_67npl86j5tr0000gn/T/ipykernel_44250/658476920.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  women['group'] ='women'
/var/folders/yj/53qmn0w907qf_67npl86j5tr0000gn/T/ipykernel_44250/658476920.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  men['total_score'] = men['home_score'] + men['away_score']
/var/folders/yj/53qmn0w907qf_67npl86j5tr0000gn/T/ipykernel_44250/658476920.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  men['group'] = 'men'
```

In [6]:

```
#Check normality of data
women['total_score'].hist()
plt.show()
men['total_score'].hist()
plt.show()
#Based on the plot, the distribution is not nromal, and we are comparing two group, so we have to use non-param test, and the data is not paired, hence we are going to use Wilcoxon-Mann_Whitney Tes
```





In [7]:

```
#Bring in pingouin package
import pingouin
#first we need to concat the data together for men and women
FIFA = pd.concat([men, women], axis = 0, ignore_index= True)
#Get only the columns that we need
FIFA_test = FIFA[['group', 'total_score']]
```

In [8]:

```
#Step 1: Cover the data from long to wide
group_vs_score_wide = FIFA_test.pivot(columns = 'group', values = 'total_score')

#Step 2: use pingouin.mwu to perform test and get p-value
test_result = pingouin.mwu(x = group_vs_score_wide['women'],
            y = group_vs_score_wide['men'],
            alternative='greater')
#step 3
p_val = test_result['p-val'].values[0]
alpha = 0.1
if p_val > alpha:
    result = 'fail to reject'
else:
    result = 'reject'
```

In [9]:

```
result_dict = {'p_val': p_val, 'result' : result}
display(result_dict)
```

```
{'p_val': 0.005106609825443641, 'result': 'reject'}
```