Final Project

Histogrmas

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

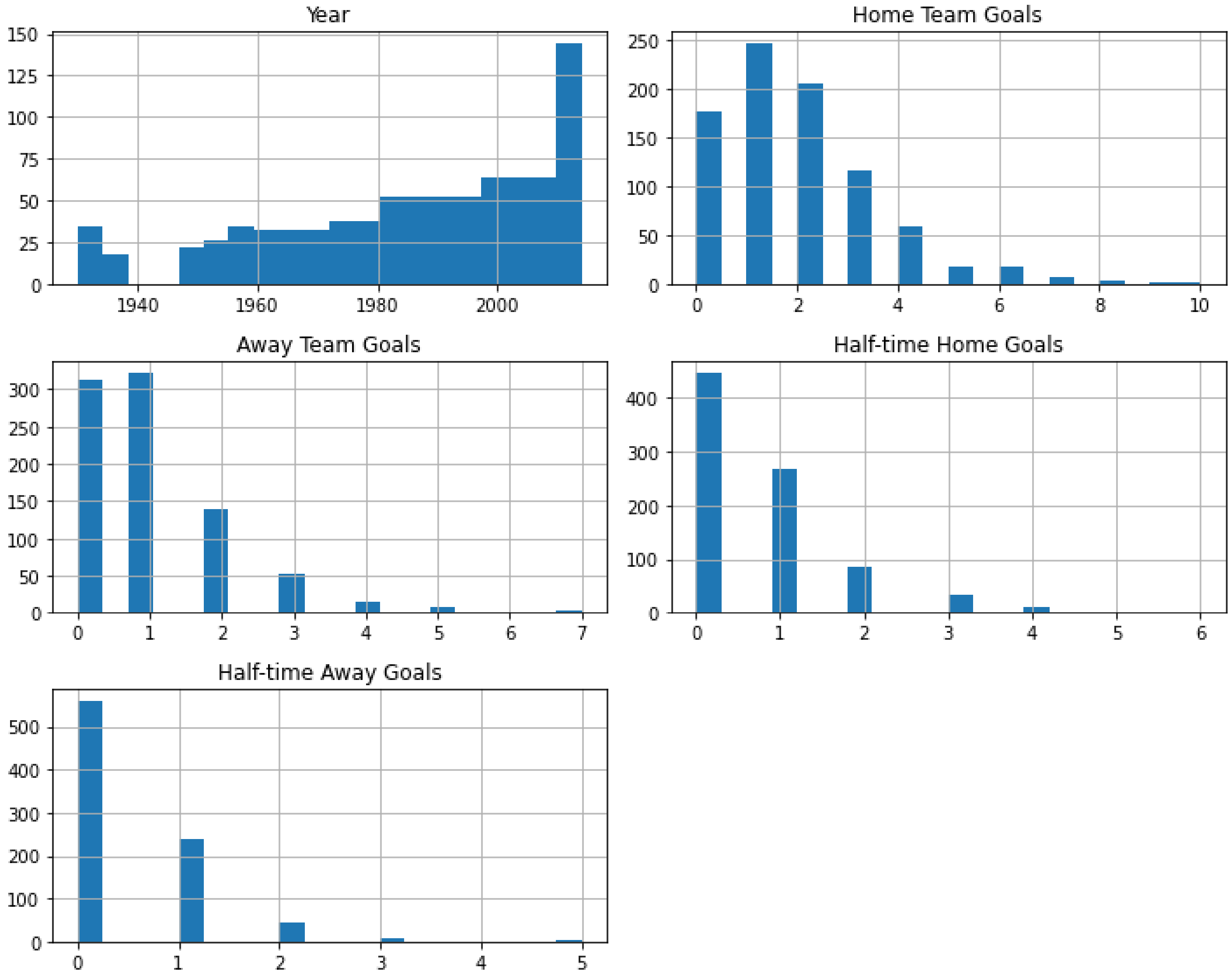
```
data = pd.read_csv('WorldCupMatches.csv')
```

In [3]:

```
selected_columns = ['Year', 'Home Team Goals', 'Away Team Goals', 'Half-time Home Goals',
df = data[selected_columns]
```

In [4]:

```
df.hist(bins=20, figsize=(10, 8))
plt.tight_layout()
plt.show()
```



Descriptive Characteristics

In [5]:

```
import statistics
```

In [6]:

```
for column in selected_columns:
    # Calculate mean
    mean = df[column].mean()

    # Calculate mode
    mode = statistics.mode(df[column])

    # Calculate range
    minimum = df[column].min()
    maximum = df[column].max()
    spread = maximum - minimum

    # Calculate tails
    q1 = df[column].quantile(0.25)
    q3 = df[column].quantile(0.75)
    iqr = q3 - q1
    lower_tail = q1 - 1.5 * iqr
    upper_tail = q3 + 1.5 * iqr

    print(f"Variable: {column}")
    print(f"Mean: {mean:.2f}")
    print(f"Mode: {mode}")
    print(f"Spread: {spread:.2f}")
    print(f"Lower Tail: {lower_tail:.2f}")
    print(f"Upper Tail: {upper_tail:.2f}")
    print("----------")
```

```
Variable: Year
Mean: 1985.09
Mode: 2014.0
Spread: 84.00
Lower Tail: 1922.00
Upper Tail: 2050.00
----------
Variable: Home Team Goals
Mean: 1.81
Mode: 1.0
Spread: 10.00
Lower Tail: -2.00
Upper Tail: 6.00
----------
Variable: Away Team Goals
Mean: 1.02
Mode: 1.0
Spread: 7.00
Lower Tail: -3.00
Upper Tail: 5.00
----------
Variable: Half-time Home Goals
Mean: 0.71
Mode: 0.0
Spread: 6.00
Lower Tail: -1.50
Upper Tail: 2.50
----------
Variable: Half-time Away Goals
Mean: 0.43
Mode: 0.0
Spread: 5.00
Lower Tail: -1.50
Upper Tail: 2.50
----------
```

Scenarios

In [38]:

```
import thinkstats2
import thinkplot
```

In [39]:

```
selected_variable = 'Home Team Goals'
```

In [43]:

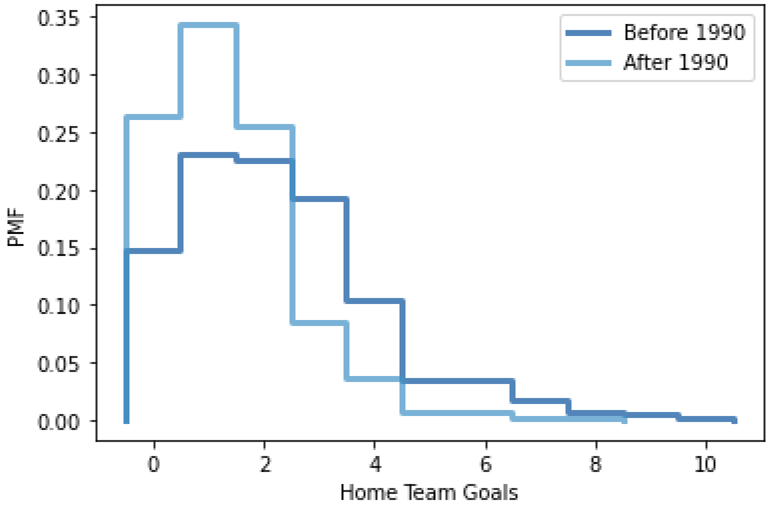
```
Before1990 = data[data['Year'] < 1990]
After1990 = data[data['Year'] >= 1990]
```

In [44]:

```
pmf1 = thinkstats2.Pmf(scenario1[selected_variable])
pmf2 = thinkstats2.Pmf(scenario2[selected_variable])
```

In [45]:

```
thinkplot.PrePlot(2)
thinkplot.Pmf(pmf1, label='Before 1990')
thinkplot.Pmf(pmf2, label='After 1990')
thinkplot.Config(xlabel=selected_variable, ylabel='PMF')
thinkplot.Show()
```



```
<Figure size 576x432 with 0 Axes>
```

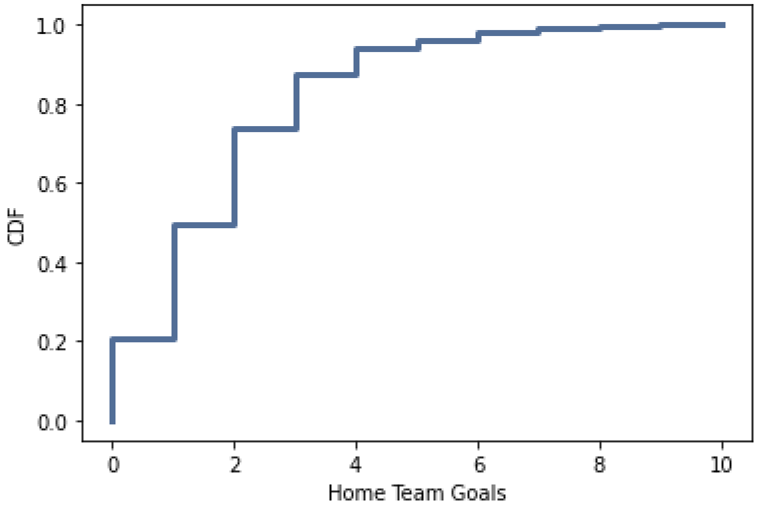
CDF

In [46]:

```
cdf = thinkstats2.Cdf(data[selected_variable])
```

In [47]:

```
thinkplot.Cdf(cdf)
thinkplot.Config(xlabel=selected_variable, ylabel='CDF')
thinkplot.Show()
```



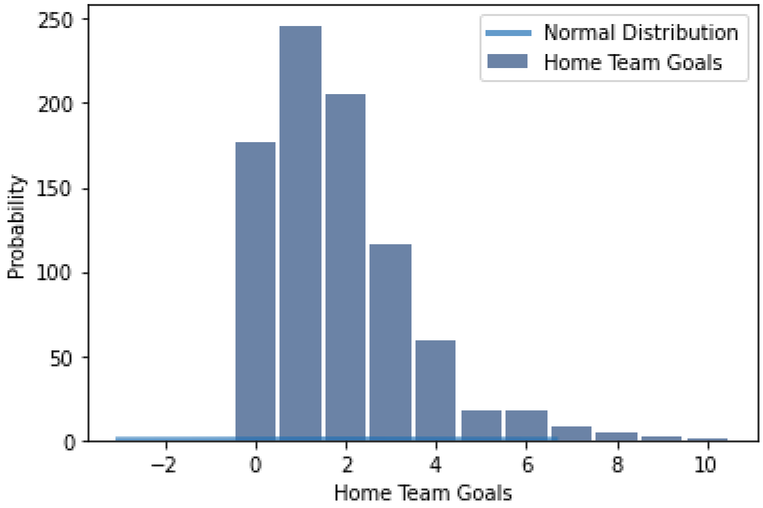
```
<Figure size 576x432 with 0 Axes>
```

Analytical Distribution

In [50]:
```
values = data[selected_variable].dropna()
```

In [51]:
```
dist = thinkstats2.NormalPdf(values.mean(), values.std())
```

In [52]:
```
thinkplot.Hist(thinkstats2.Hist(values, label=selected_variable))
thinkplot.Pdf(dist, label='Normal Distribution')
thinkplot.Config(xlabel=selected_variable, ylabel='Probability')
thinkplot.Show()
```



```
<Figure size 576x432 with 0 Axes>
```

Scatter Plots

In [56]:
```
import numpy as np
```

In [57]:
```
variable1 = 'Home Team Goals'
variable2 = 'Away Team Goals'
```

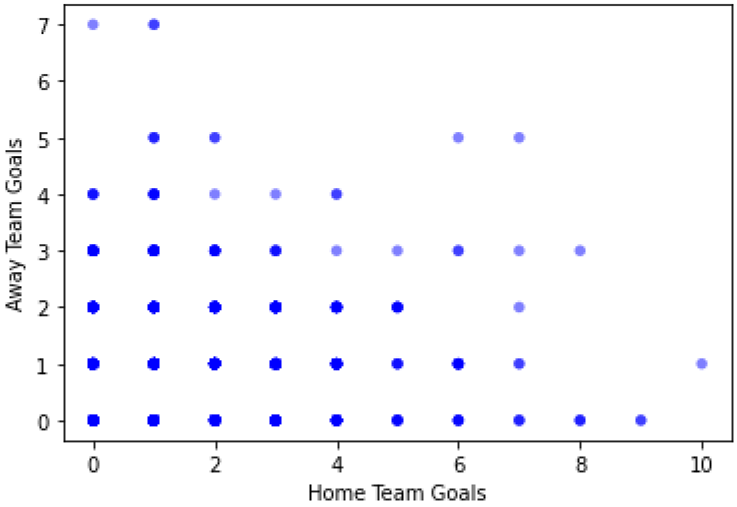
In [58]:
```
values1 = data[variable1].dropna()
values2 = data[variable2].dropna()
```

In [59]:

```
covariance = np.cov(values1, values2)
pearson_corr = np.corrcoef(values1, values2)[0, 1]
```

In [60]:

```
thinkplot.Scatter(values1, values2, alpha=0.5)
thinkplot.Config(xlabel=variable1, ylabel=variable2)
thinkplot.Show()
```

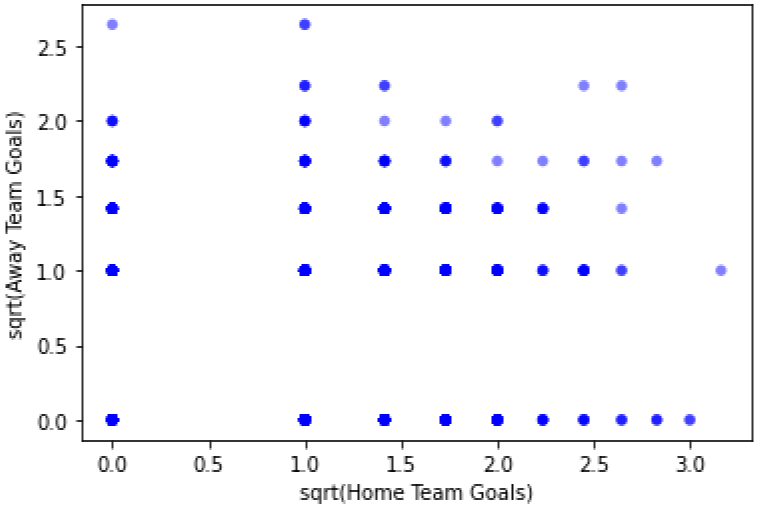


```
<Figure size 576x432 with 0 Axes>
```

In [61]:

```
transformed_values1 = np.sqrt(values1)
transformed_values2 = np.sqrt(values2)

thinkplot.Scatter(transformed_values1, transformed_values2, alpha=0.5)
thinkplot.Config(xlabel='sqrt(' + variable1 + ')', ylabel='sqrt(' + variable2 + ')')
thinkplot.Show()
```



```
<Figure size 576x432 with 0 Axes>
```

Regression Analysis

In [69]:

```
import statsmodels.api as sm
```

In [73]:

```
data_clean = data.dropna(subset=[variable1, variable2])
```

In [74]:

```
y = data_clean['Home Team Goals']
X = data_clean[['Away Team Goals', 'Half-time Home Goals', 'Half-time Away Goals']]
```

In [76]:

```
model = sm.OLS(y, X)
results = model.fit()
```

In [77]:

```
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Home Team Goals   R-squared:                       0.535
Model:                            OLS   Adj. R-squared:                  0.534
Method:                 Least Squares   F-statistic:                     325.5
Date:                Fri, 02 Jun 2023   Prob (F-statistic):          1.37e-140
Time:                        18:06:00   Log-Likelihood:                -1287.9
No. Observations:                 852   AIC:                             2584.
Df Residuals:                     848   BIC:                             2603.
Df Model:                           3
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                    0.8905      0.059     15.035      0.000       0.774       1.007
Away Team Goals          0.0998      0.048      2.073      0.038       0.005       0.194
Half-time Home Goals     1.2567      0.040     31.227      0.000       1.178       1.336
Half-time Away Goals    -0.1690      0.076     -2.230      0.026      -0.318      -0.020
==============================================================================
Omnibus:                      129.336   Durbin-Watson:                   1.799
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              203.386
Skew:                           1.001   Prob(JB):                     6.84e-45
Kurtosis:                       4.311   Cond. No.                         4.35
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifie
d.
```

In [ ]: