In [1]:
```
# DSC530-T302
# Stephen Smitshoek
# Final Project
# Climate Change
```

In [2]:
```
import numpy as np
import pandas
from datetime import datetime
import thinkstats2
import thinkplot
import scipy
import statsmodels.formula.api as smf
```

In [3]:
```
def import_data():
    global_temp = pandas.read_csv('GlobalTemperatures.csv')
    co2_atmo = pandas.read_csv('co2_atmo.csv')
    co2_emission = pandas.read_csv('co2_emission.csv')
    sea_levels = pandas.read_csv('sea_levels_2015.csv')

    return {'global_temp': global_temp, 'co2_atmo': co2_atmo, 'co2_emission': co2_emission, 'sea_levels': sea_levels}
```

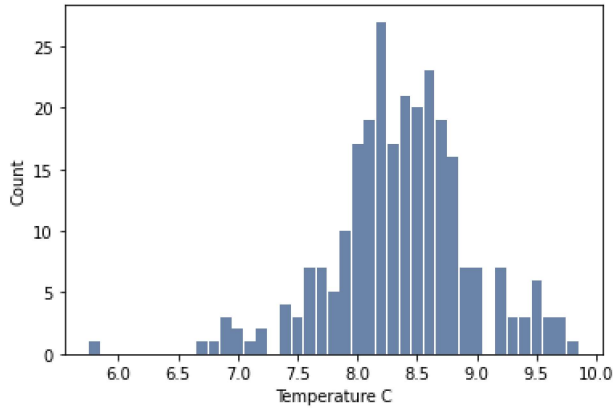
In [4]:
```
def clean_up(climate_dict):
    # Keep only the desired columns from the global_temp dataframe
    climate_dict['global_temp'] = climate_dict['global_temp'][['dt', 'LandAverageTemperature']]
    # Convert date into datetime object
    climate_dict['global_temp']['dt'] = pandas.to_datetime(climate_dict['global_temp']['dt'], format='%Y-%m-%d')
    # Average the temperature for the year
    climate_dict['global_temp'] = climate_dict['global_temp'].groupby(climate_dict['global_temp'].dt.dt.year) \
        ['LandAverageTemperature'].mean()

    # Create single column containing date in the co2_atmo dataframe
    climate_dict['co2_atmo']['dt'] = climate_dict['co2_atmo']['Year'].astype(str) + '-' + \
                                     climate_dict['co2_atmo']['Month'].astype(str) + '-' + \
                                     '1'
    # Convert date into datetime object
    climate_dict['co2_atmo']['dt'] = pandas.to_datetime(climate_dict['co2_atmo']['dt'], format='%Y-%m-%d')
    # Keep only the desired columns from the global_temp dataframe
    climate_dict['co2_atmo'] = climate_dict['co2_atmo'][['dt', 'Seasonally Adjusted CO2 (ppm)']]
    # Average the CO2 ppm for the year
    climate_dict['co2_atmo'] = climate_dict['co2_atmo'].groupby(climate_dict['co2_atmo'].dt.dt.year) \
        ['Seasonally Adjusted CO2 (ppm)'].mean()

    climate_dict['co2_emission'] = climate_dict['co2_emission'].groupby(["Year"])["Annual CO₂ emissions (tonnes )"].sum()

    climate_dict['sea_levels']['dt'] = pandas.to_datetime(climate_dict['sea_levels']['Time'], format='%Y-%m-%d')
    climate_dict['sea_levels'] = climate_dict['sea_levels'].groupby(climate_dict['sea_levels'].dt.dt.year)['GMSL'].mean()
```

In [5]:
```
def combine_data(climate_dict):
    global_temp = climate_dict['global_temp'].to_frame()
    co2_atmo = climate_dict['co2_atmo'].to_frame()
    co2_emission = climate_dict['co2_emission'].to_frame()
    co2_emission = co2_emission.rename_axis('dt')
    sea_levels = climate_dict['sea_levels'].to_frame()

    climate_df = pandas.concat([global_temp, co2_atmo, co2_emission, sea_levels], axis=1)

    return climate_df
```

In [6]:
```
class CorrelationPermute(thinkstats2.HypothesisTest):
    def TestStatistic(self, data):
        xs, ys = data
        test_stat = abs(thinkstats2.Corr(xs, ys))
        return test_stat

    def RunModel(self):
        xs, ys = self.data
        xs = np.random.permutation(xs)
        return xs, ys
```

In [7]:
```
climate_dict = import_data()
clean_up(climate_dict)
climate_df = combine_data(climate_dict)
```

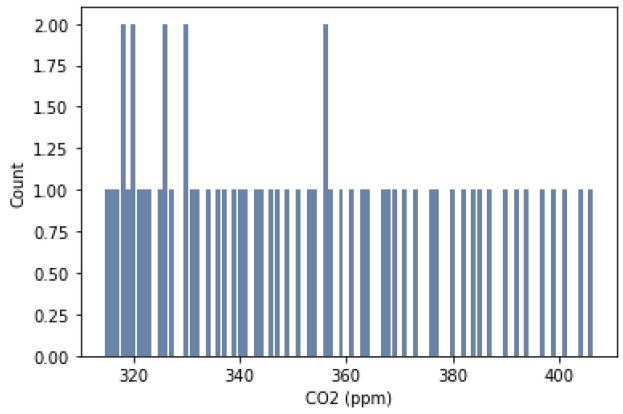
In [8]:
```
temp_hist = thinkstats2.Hist(round(climate_df['LandAverageTemperature'], 1))
thinkplot.Hist(temp_hist)
thinkplot.Config(xlabel='Temperature C', ylabel='Count')
```



In [9]:
```
temp_mean = thinkstats2.Mean(climate_df['LandAverageTemperature'].dropna())
temp_median = thinkstats2.Median(climate_df['LandAverageTemperature'].dropna())
temp_std = thinkstats2.Std(climate_df['LandAverageTemperature'].dropna())

print(f'The mean of the average land temperature is {temp_mean:.2f}')
print(f'The median of the average land temperature is {temp_median:.2f}')
print(f'The standard deviation of the average land temperature is {temp_std:.2f}')
```

```
The mean of the average land temperature is 8.37
The median of the average land temperature is 8.37
The standard deviation of the average land temperature is 0.58
```

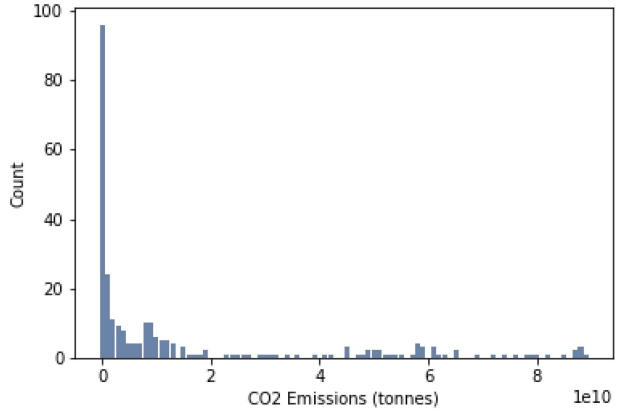
In [10]:
```
co2_atmo_hist = thinkstats2.Hist(round(climate_df['Seasonally Adjusted CO2 (ppm)'], 0))
thinkplot.Hist(co2_atmo_hist)
thinkplot.Config(xlabel='CO2 (ppm)', ylabel='Count')
```



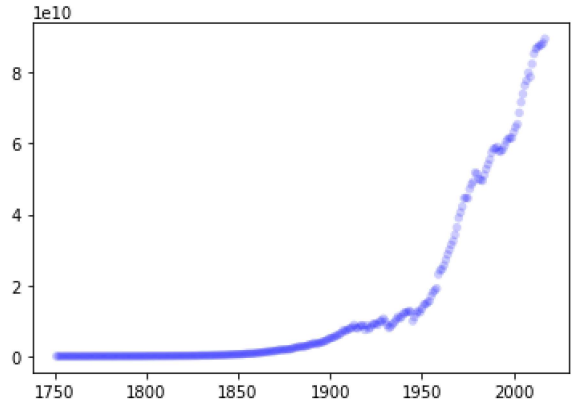
In [11]:
```
co2_mean = thinkstats2.Mean(climate_df['Seasonally Adjusted CO2 (ppm)'].dropna())
co2_median = thinkstats2.Median(climate_df['Seasonally Adjusted CO2 (ppm)'].dropna())
co2_std = thinkstats2.Std(climate_df['Seasonally Adjusted CO2 (ppm)'].dropna())

print(f'The mean of the yearly average CO2 (ppm) is {co2_mean:.2f}')
print(f'The median of the yearly average CO2 (ppm) is {co2_median:.2f}')
print(f'The standard deviation of the yearly average CO2 (ppm) is {co2_std:.2f}')
```

```
The mean of the yearly average CO2 (ppm) is 352.78
The median of the yearly average CO2 (ppm) is 348.93
The standard deviation of the yearly average CO2 (ppm) is 26.82
```

In [12]:
```
emissions_hist = thinkstats2.Hist(round(climate_df['Annual CO₂ emissions (tonnes )']/1000000000, 0)*1000000000)

thinkplot.Hist(emissions_hist)
thinkplot.Config(xlabel='CO2 Emissions (tonnes)', ylabel='Count')
```



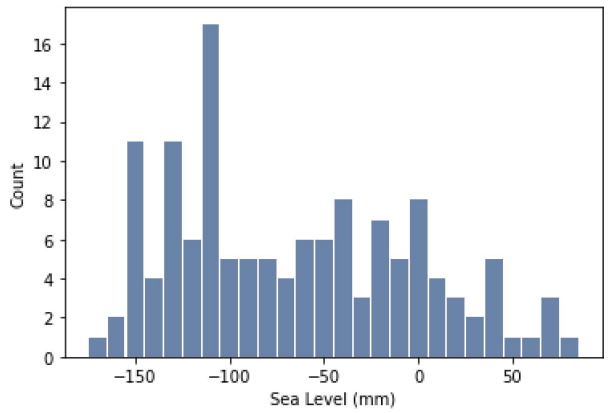
In [13]:
```
thinkplot.Scatter(climate_df['Annual CO₂ emissions (tonnes )'].dropna())
```



In [14]:
```
emissions_mean = thinkstats2.Mean(climate_df['Annual CO₂ emissions (tonnes )'].dropna())
emissions_median = thinkstats2.Median(climate_df['Annual CO₂ emissions (tonnes )'].dropna())
emissions_std = thinkstats2.Std(climate_df['Annual CO₂ emissions (tonnes )'].dropna())

print(f'The mean of the global yearly emissions is {emissions_mean:.0f}')
print(f'The median of the global yearly emissions is {emissions_median:.0f}')
print(f'The standard deviation of the global yearly emissions is {emissions_std:.0f}')
```

```
The mean of the global yearly emissions is 15077558446
The median of the global yearly emissions is 2724293920
The standard deviation of the global yearly emissions is 23887325178
```

In [15]:
```
gmsl_hist = thinkstats2.Hist(round(climate_df['GMSL']/10, 0)*10)
thinkplot.Hist(gmsl_hist)
thinkplot.Config(xlabel='Sea Level (mm)', ylabel='Count')
```

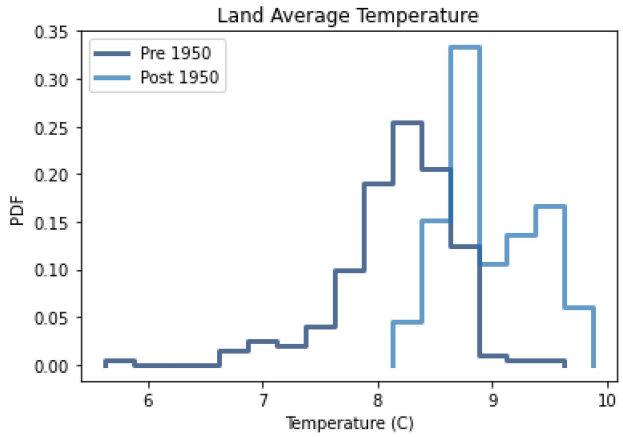


In [16]:
```
gmsl_mean = thinkstats2.Mean(climate_df['GMSL'].dropna())
gmsl_median = thinkstats2.Median(climate_df['GMSL'].dropna())
gmsl_std = thinkstats2.Std(climate_df['GMSL'].dropna())

print(f'The mean of the global mean sea level is {gmsl_mean:.2f}')
print(f'The median of the global mean sea level is {gmsl_median:.2f}')
print(f'The standard deviation of the global mean sea level is {gmsl_std:.2f}')
```

```
The mean of the global mean sea level is -66.08
The median of the global mean sea level is -76.11
The standard deviation of the global mean sea level is 62.70
```

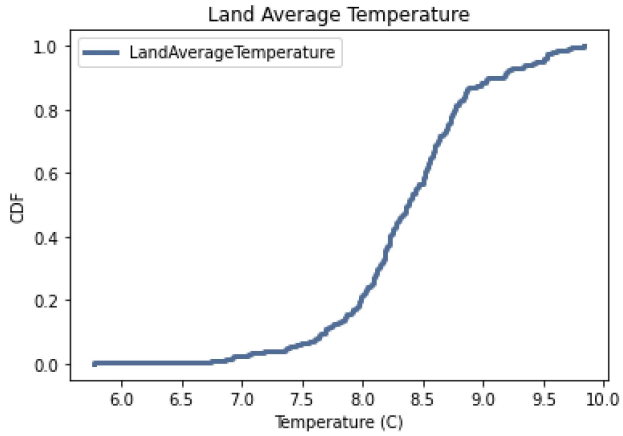
In [17]:
```
# Compare two scenarios in your data using a PMF
temp_df = climate_df.LandAverageTemperature.dropna().to_frame()
temp_df = temp_df.rename_axis('year').reset_index()
temp_df['rounded_temp'] = .25 * round(temp_df['LandAverageTemperature']/.25)
temp_pre_1950 = temp_df.loc[temp_df['year'] < 1950]
temp_post_1950 = temp_df.loc[temp_df['year'] >= 1950]
```

In [18]:
```
pre_1950_pmf = thinkstats2.Pmf(temp_pre_1950['rounded_temp'], label="Pre 1950")
post_1950_pmf = thinkstats2.Pmf(temp_post_1950['rounded_temp'], label="Post 1950")
thinkplot.Pmfs([pre_1950_pmf, post_1950_pmf])
thinkplot.Show(xlabel='Temperature (C)', ylabel='PDF', title='Land Average Temperature')
```




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

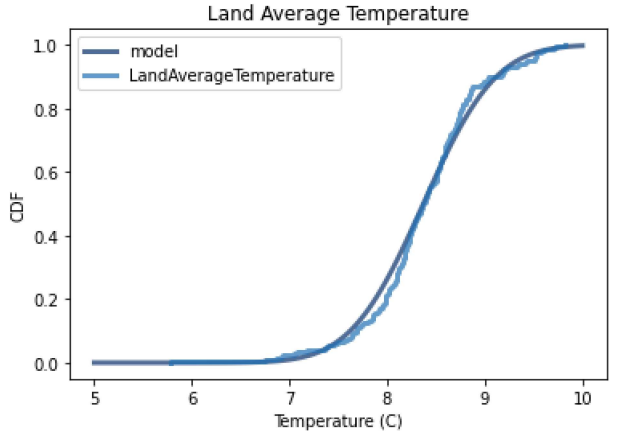
In [19]:
```
temp_cdf = thinkstats2.Cdf(temp_df.LandAverageTemperature, label='LandAverageTemperature')
thinkplot.Cdf(temp_cdf)
thinkplot.Show(xlabel='Temperature (C)', ylabel='CDF', title='Land Average Temperature')
```




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

In [20]:
```
mean = climate_df.LandAverageTemperature.dropna().mean()
std = climate_df.LandAverageTemperature.dropna().std()
x = climate_df.LandAverageTemperature.dropna()
xs, ps = thinkstats2.RenderNormalCdf(mean, std, low=5, high=10)
```

In [21]:
```
thinkplot.Plot(xs, ps, label='model')
thinkplot.Cdf(temp_cdf)
thinkplot.Show(xlabel='Temperature (C)', ylabel='CDF', title='Land Average Temperature')
```



<Figure size 576x432 with 0 Axes>

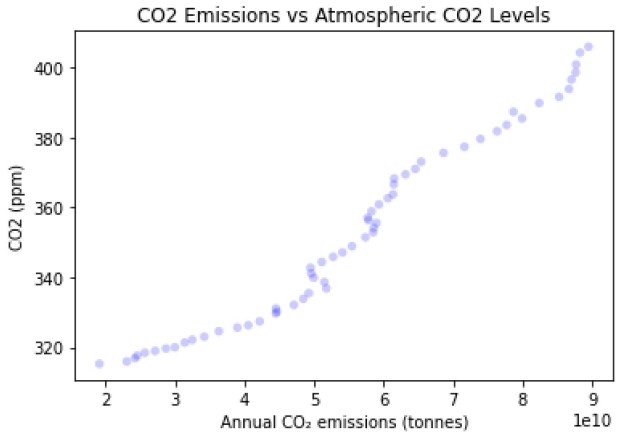
In [22]:
```
emission_vs_ppm = climate_df[['Annual CO₂ emissions (tonnes )', 'Seasonally Adjusted CO2 (ppm)']].dropna()
corr = thinkstats2.Corr(emission_vs_ppm['Annual CO₂ emissions (tonnes )'], \
                        emission_vs_ppm['Seasonally Adjusted CO2 (ppm)'])
sp_corr = thinkstats2.SpearmanCorr(emission_vs_ppm['Annual CO₂ emissions (tonnes )'], \
                                   emission_vs_ppm['Seasonally Adjusted CO2 (ppm)'])
print(f'The correlation factor between CO2 levels in the atmosphere and Anuual CO2 emissions is {corr:.2f} '\
      f'and the Spearman correlation \nfactor is {sp_corr:.2f}')
```

```
The correlation factor between CO2 levels in the atmosphere and Anuual CO2 emissions is 0.98 and the Spearman correlation
factor is 1.00
```

In [23]:
```
cor_test = CorrelationPermute((emission_vs_ppm['Annual CO₂ emissions (tonnes )'], \
                               emission_vs_ppm['Seasonally Adjusted CO2 (ppm)']))
cor_test.PValue()
```

Out[23]:
```
0.0
```

In [24]:
```
thinkplot.Scatter(emission_vs_ppm['Annual CO₂ emissions (tonnes )'], \
                  emission_vs_ppm['Seasonally Adjusted CO2 (ppm)'])
thinkplot.Config(xlabel='Annual CO₂ emissions (tonnes)',
                 ylabel='CO2 (ppm)',
                 title='CO2 Emissions vs Atmospheric CO2 Levels')
```



In [25]:
```
temp_vs_emission = climate_df[['LandAverageTemperature', 'Annual CO₂ emissions (tonnes )']]
temp_vs_emission = temp_vs_emission.dropna()
```

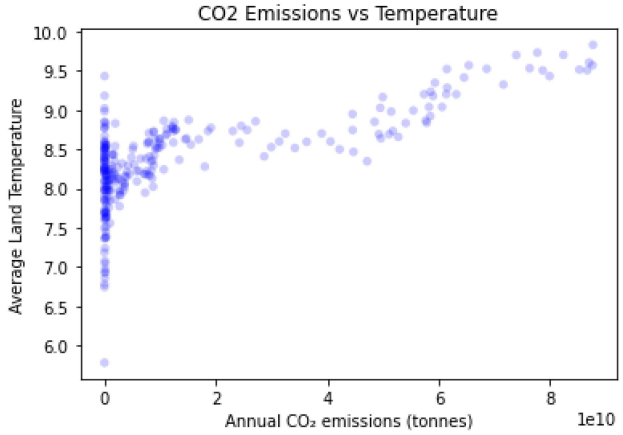
In [26]:
```
corr = thinkstats2.Corr(temp_vs_emission['Annual CO₂ emissions (tonnes )'], \
                        temp_vs_emission['LandAverageTemperature'])
sp_corr = thinkstats2.SpearmanCorr(temp_vs_emission['Annual CO₂ emissions (tonnes )'], \
                                   temp_vs_emission['LandAverageTemperature'])
print(f'The correlation factor between average land temperature and Anuual CO2 emissions is {corr:.2f}'\
      f'and the Spearman correlation factor is {sp_corr:.2f}')
```

```
The correlation factor between average land temperature and Anuual CO2 emissions is 0.70and the Spearman correlation factor
is 0.62
```

In [27]:
```
cor_test = CorrelationPermute((temp_vs_emission['Annual CO₂ emissions (tonnes )'], \
                               temp_vs_emission['LandAverageTemperature']))
cor_test.PValue()
```

Out[27]:
```
0.0
```

In [28]:
```
thinkplot.Scatter(temp_vs_emission['Annual CO₂ emissions (tonnes )'], temp_vs_emission['LandAverageTemperature'])
thinkplot.Config(xlabel='Annual CO₂ emissions (tonnes)',
                 ylabel='Average Land Temperature',
                 title='CO2 Emissions vs Temperature')
```

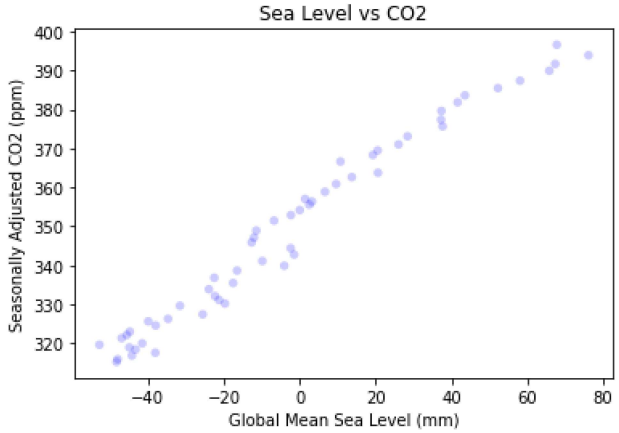


In [29]:
```
gmsl_vs_ppm = climate_df[['GMSL', 'Seasonally Adjusted CO2 (ppm)']]
gmsl_vs_ppm = gmsl_vs_ppm.dropna()
```

In [30]:
```
corr = thinkstats2.Corr(gmsl_vs_ppm['GMSL'], gmsl_vs_ppm['Seasonally Adjusted CO2 (ppm)'])
sp_corr = thinkstats2.SpearmanCorr(gmsl_vs_ppm['GMSL'], gmsl_vs_ppm['Seasonally Adjusted CO2 (ppm)'])
print(f'The correlation factor between global mean sea level and CO2 ppm is {corr:.2f} and the Spearman correlation '\
      f'factor is {sp_corr:.2f}')
```

```
The correlation factor between global mean sea level and CO2 ppm is 0.99 and the Spearman correlation factor is 0.99
```

In [31]:
```
thinkplot.Scatter(gmsl_vs_ppm['GMSL'], gmsl_vs_ppm['Seasonally Adjusted CO2 (ppm)'])
thinkplot.Config(xlabel='Global Mean Sea Level (mm)',
                 ylabel='Seasonally Adjusted CO2 (ppm)',
                 title='Sea Level vs CO2')
```



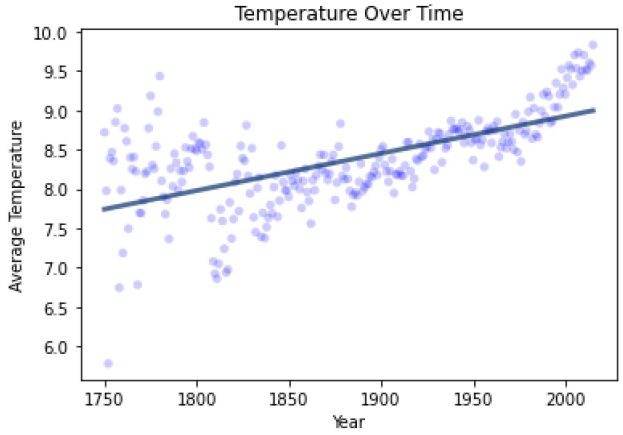
In [32]:
```
cor_test = CorrelationPermute((gmsl_vs_ppm['GMSL'], gmsl_vs_ppm['Seasonally Adjusted CO2 (ppm)']))
cor_test.PValue()
```

Out[32]:
```
0.0
```

In [33]:
```
climate_df = climate_df.rename_axis('year').reset_index()
```

In [34]:
```
model = smf.ols('LandAverageTemperature ~ year', climate_df)
results = model.fit()
years = model.exog[:,1]
values = model.endog
```

In [35]:
```
thinkplot.Scatter(years, values)
thinkplot.Plot(years, results.fittedvalues)
thinkplot.Config(xlabel='Year', ylabel='Average Temperature', title='Temperature Over Time')
```


Temperature Over Time
Average Temperature
Year
10.0
9.5
9.0
8.5
8.0
7.5
7.0
6.5
6.0
1750
1800
1850
1900
1950
2000