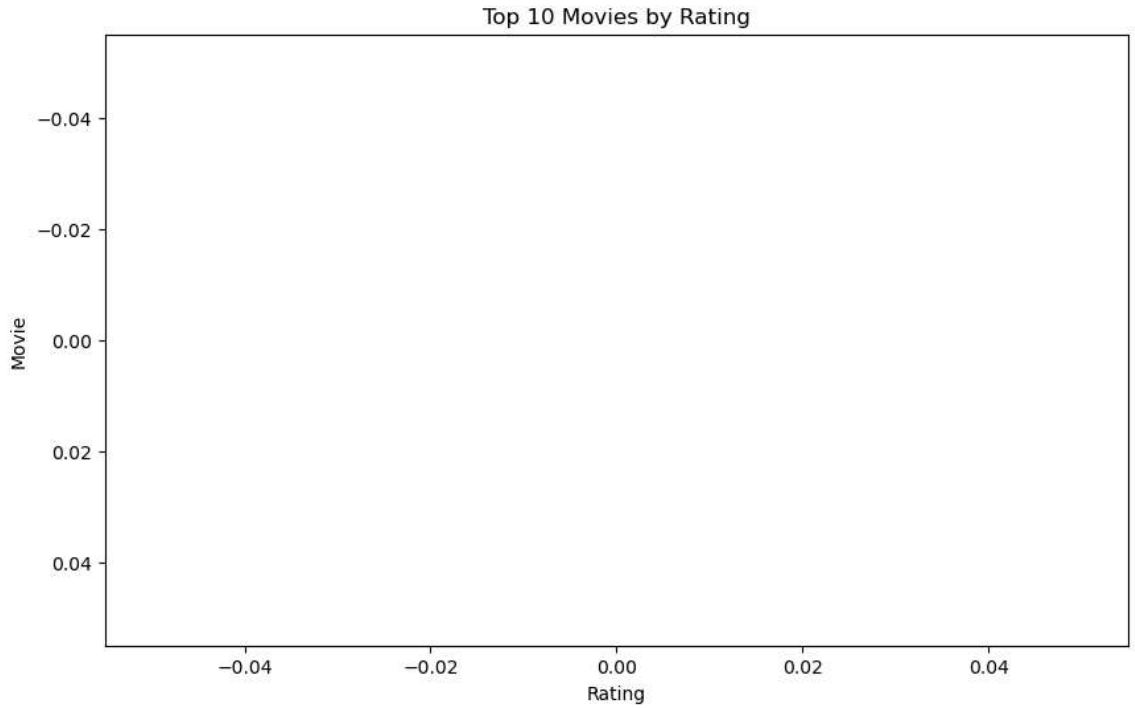
In [7]:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

# Scrape data
url = 'https://www.imdb.com/chart/top/'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

movies = []
ratings = []

for movie in soup.select('td.titleColumn'):
    movies.append(movie.find('a').text)
    ratings.append(float(movie.find('strong').text))

# Data cleaning and preparation
data = pd.DataFrame({'Movie': movies, 'Rating': ratings})

# Data analysis
# Summary statistics
summary_stats = data.describe()

# Top 10 movies by rating
top_10_movies = data.nlargest(10, 'Rating')

# Visualization
plt.figure(figsize=(10, 6))
bars = plt.barh(top_10_movies['Movie'], top_10_movies['Rating'], color='sky
plt.xlabel('Rating')
plt.ylabel('Movie')
plt.title('Top 10 Movies by Rating')
plt.gca().invert_yaxis()  # Invert y-axis to show highest-rated movie on to

# Add numerical values on bars
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height()/2, f'{bar.get_
             va='center', ha='left', fontsize=8, color='black')

plt.show()

# Display summary statistics
print("Summary Statistics:")
print(summary_stats)

# Display top 10 movies by rating
print("\nTop 10 Movies by Rating:")
print(top_10_movies)
```

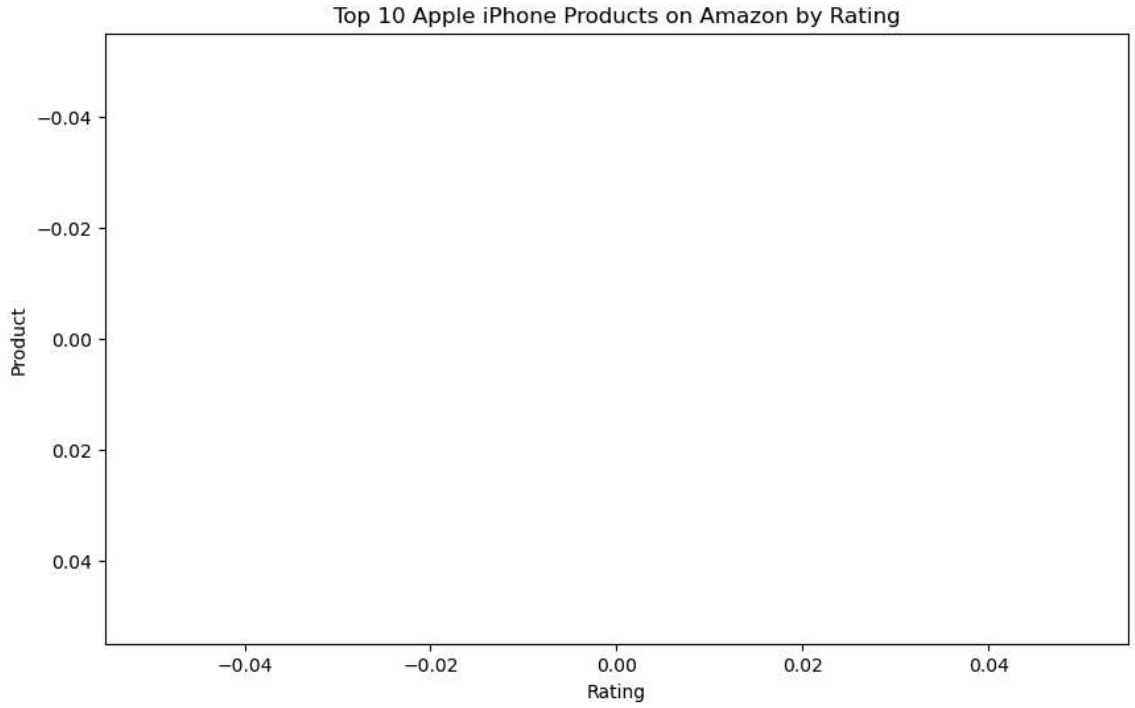



```
Summary Statistics:
       Movie  Rating
count    0.0     0.0
mean     NaN     NaN
std      NaN     NaN
min      NaN     NaN
25%      NaN     NaN
50%      NaN     NaN
75%      NaN     NaN
max      NaN     NaN

Top 10 Movies by Rating:
Empty DataFrame
Columns: [Movie, Rating]
Index: []
```

In [6]:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

# Scrape data
url = 'https://www.amazon.com/s?k=apple+iphone'
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/53
}
response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, 'html.parser')

products = []
prices = []
ratings = []

for product in soup.find_all('div', {'data-component-type': 's-search-resul
    name_elem = product.find('span', {'class': 'a-size-medium'})
    price_elem = product.find('span', {'class': 'a-price-whole'})
    rating_elem = product.find('span', {'class': 'a-icon-alt'})

    if name_elem and price_elem and rating_elem:
        try:
            name = name_elem.text.strip()
            price = float(price_elem.text.replace('$', '').replace(',', '')
            rating = float(rating_elem.text.split()[0])
            products.append(name)
            prices.append(price)
            ratings.append(rating)
        except ValueError:
            continue

# Data cleaning and preparation
data = pd.DataFrame({'Product': products, 'Price': prices, 'Rating': rating

# Data analysis
# Summary statistics
summary_stats = data.describe()

# Top 10 products by rating
top_10_products = data.nlargest(10, 'Rating')

# Visualization
plt.figure(figsize=(10, 6))
bars = plt.barh(top_10_products['Product'], top_10_products['Rating'], colo
plt.xlabel('Rating')
plt.ylabel('Product')
plt.title('Top 10 Apple iPhone Products on Amazon by Rating')
plt.gca().invert_yaxis()  # Invert y-axis to show highest-rated product on
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height()/2, f'{bar.get_
             va='center', ha='left', fontsize=8, color='black')
plt.show()

# Display summary statistics
print("Summary Statistics:")
```

```
print(summary_stats)

# Display top 10 products by rating
print("\nTop 10 Apple iPhone Products on Amazon by Rating:")
print(top_10_products)
```




```
Summary Statistics:
       Product  Price  Rating
count      0.0    0.0     0.0
mean       NaN    NaN     NaN
std        NaN    NaN     NaN
min        NaN    NaN     NaN
25%        NaN    NaN     NaN
50%        NaN    NaN     NaN
75%        NaN    NaN     NaN
max        NaN    NaN     NaN

Top 10 Apple iPhone Products on Amazon by Rating:
Empty DataFrame
Columns: [Product, Price, Rating]
Index: []
```