restart the kernel after installation

!pip install pandas-profiling --quiet

In [2]:
```
medical_charges_url = 'https://raw.githubusercontent.com/JovianML/opendatasets/master/
```

In [3]:
```
from urllib.request import urlretrieve
```

In [6]:
```
urlretrieve(medical_charges_url, 'medical-charges.csv')
```

Out[6]:
```
('medical-charges.csv', <http.client.HTTPMessage at 0x18c9ef25cd0>)
```

In [7]:
```
import pandas as pd
```

In [8]:
```
medical_df = pd.read_csv('medical.csv')
```

In [9]:
```
medical_df
```

Out[9]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [10]:
```
medical_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [11]:
```
medical_df.describe()
```

Out[11]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [15]:
```
!pip install plotly matplotlib seaborn --quiet
```

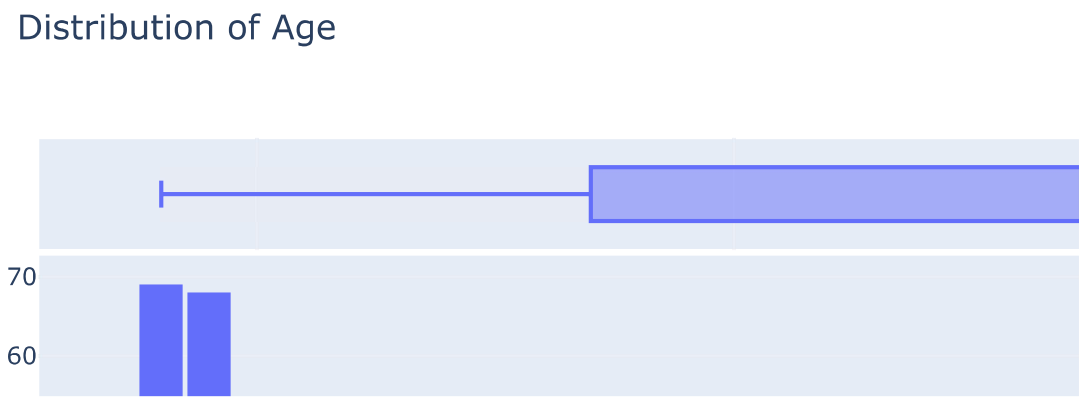
In [16]:
```
import plotly.express as px
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [17]:
```
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (10, 6)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

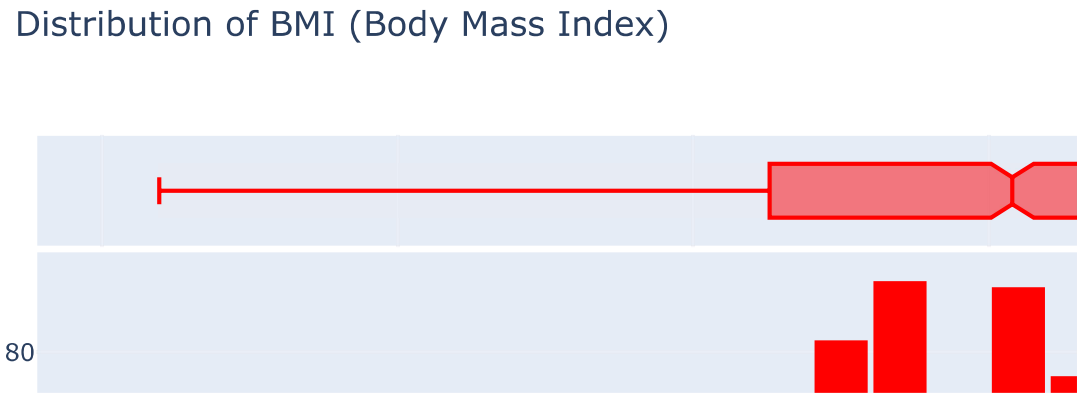
In [18]:
```
medical_df.age.describe()
```

Out[18]:
```
count    1338.000000
mean       39.207025
std        14.049960
min        18.000000
25%        27.000000
50%        39.000000
75%        51.000000
max        64.000000
Name: age, dtype: float64
```

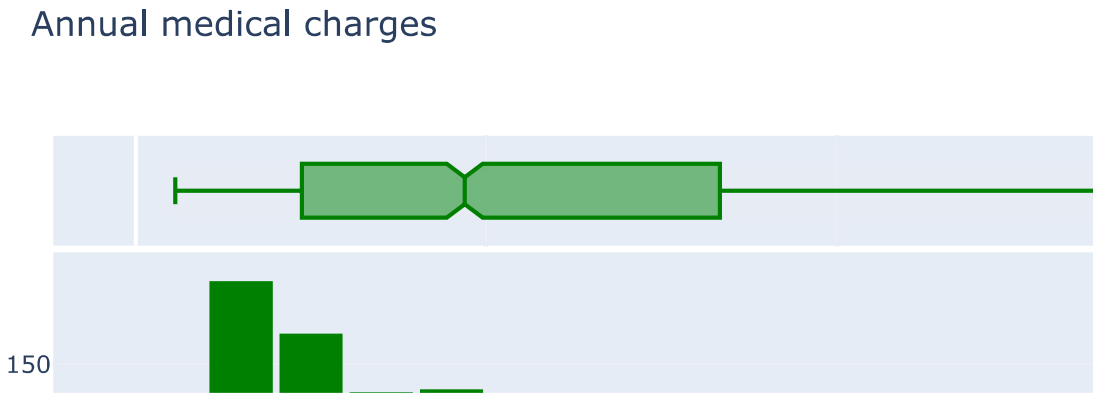
In [19]:
```
fig = px.histogram(medical_df,
                   x='age',
                   marginal='box',
                   nbins=47,
                   title='Distribution of Age')
fig.update_layout(bargap=0.1)
fig.show()
```



In [20]:
```
fig = px.histogram(medical_df,
                   x='bmi',
                   marginal='box',
                   color_discrete_sequence=['red'],
                   title='Distribution of BMI (Body Mass Index)')
fig.update_layout(bargap=0.1)
fig.show()
```



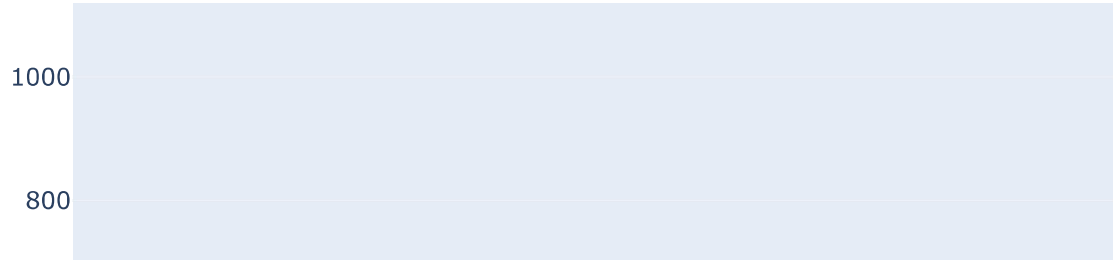
In [22]:
```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal='box',
                   color_discrete_sequence=['green','grey'],
                   title='Annual medical charges')
fig.update_layout(bargap=0.1)
fig.show()
```



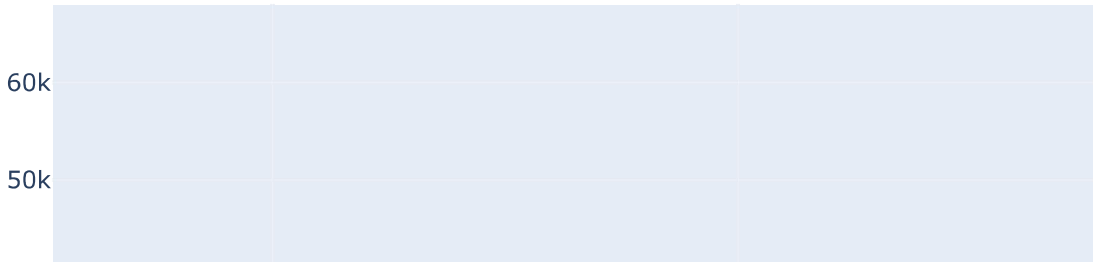
In [23]:
```
medical_df.smoker.value_counts()
```

Out[23]:
```
smoker
no     1064
yes     274
Name: count, dtype: int64
```

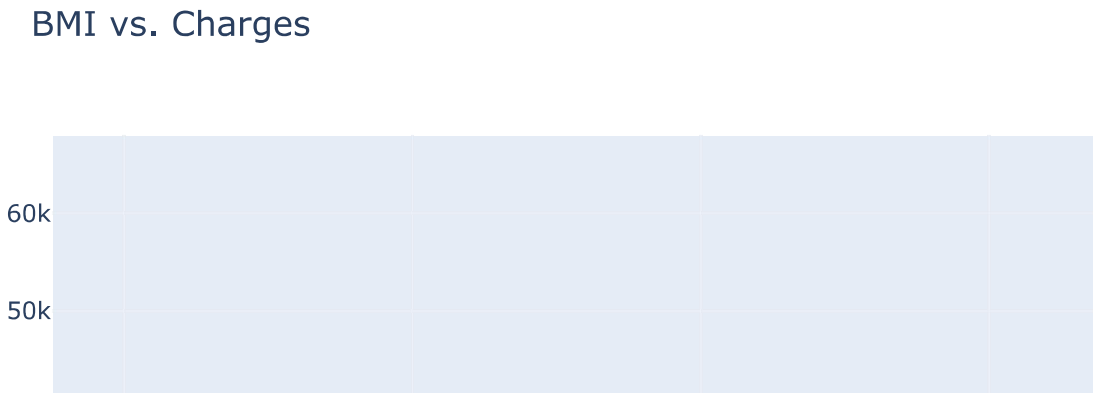
In [24]:
```
px.histogram(medical_df, x='smoker', color='sex', title='Smoker')
```



In [25]:
```
fig = px.scatter(medical_df,
                 x='age',
                 y='charges',
                 color='smoker',
                 opacity=0.8,
                 hover_data=['sex'],
                 title='Age vs. Charges')
fig.update_traces(marker_size=5)
fig.show()
```



In [26]:
```
fig = px.scatter(medical_df,
                 x='bmi',
                 y='charges',
                 color='smoker',
                 opacity=0.8,
                 hover_data=['sex'],
                 title='BMI vs. Charges')
fig.update_traces(marker_size=5)
fig.show()
```



In [27]:
```
medical_df.charges.corr(medical_df.age)
```

Out[27]:
```
0.29900819333064765
```

In [28]:
```
medical_df.charges.corr(medical_df.bmi)
```

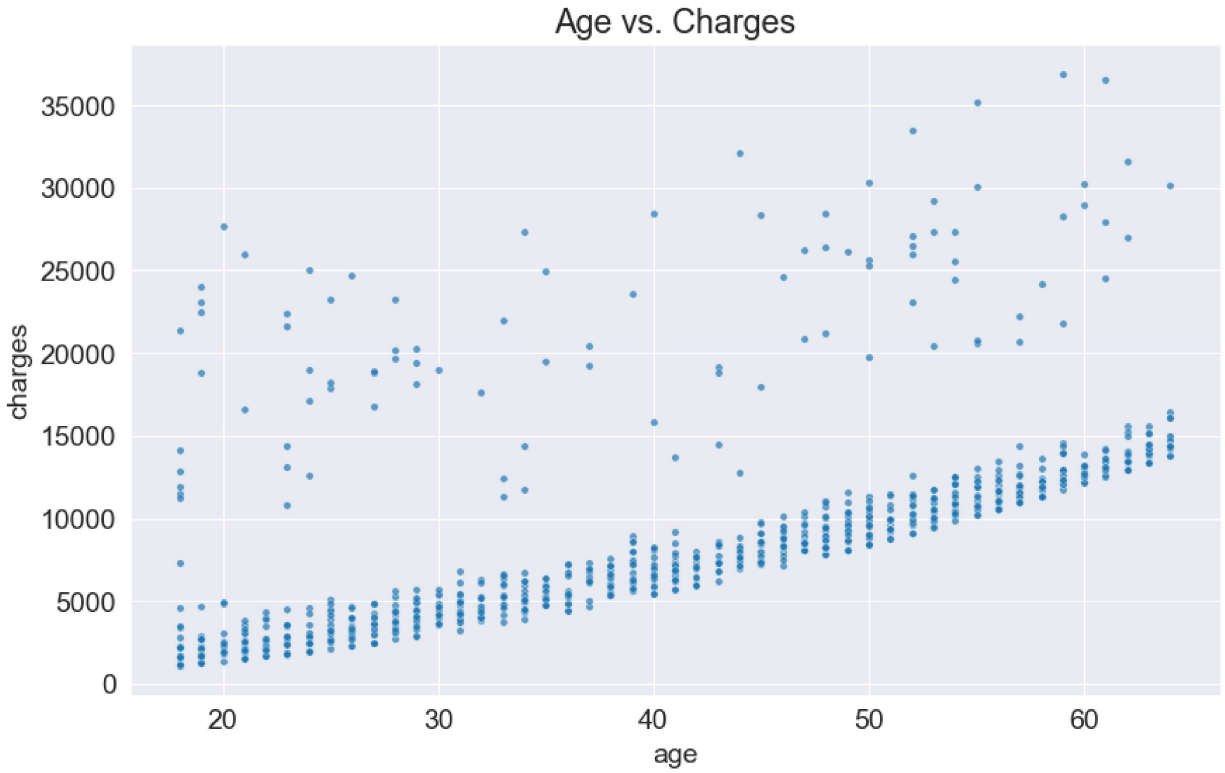
Out[28]:
```
0.19834096883362892
```

In [29]:
```
smoker_values = {'no': 0, 'yes': 1}
smoker_numeric = medical_df.smoker.map(smoker_values)
medical_df.charges.corr(smoker_numeric)
```

Out[29]:
```
0.7872514304984772
```

In [34]:
```
non_smoker_df = medical_df[medical_df.smoker == 'no']
```

In [35]:
```
plt.title('Age vs. Charges')
sns.scatterplot(data=non_smoker_df, x='age', y='charges', alpha=0.7, s=15);
```

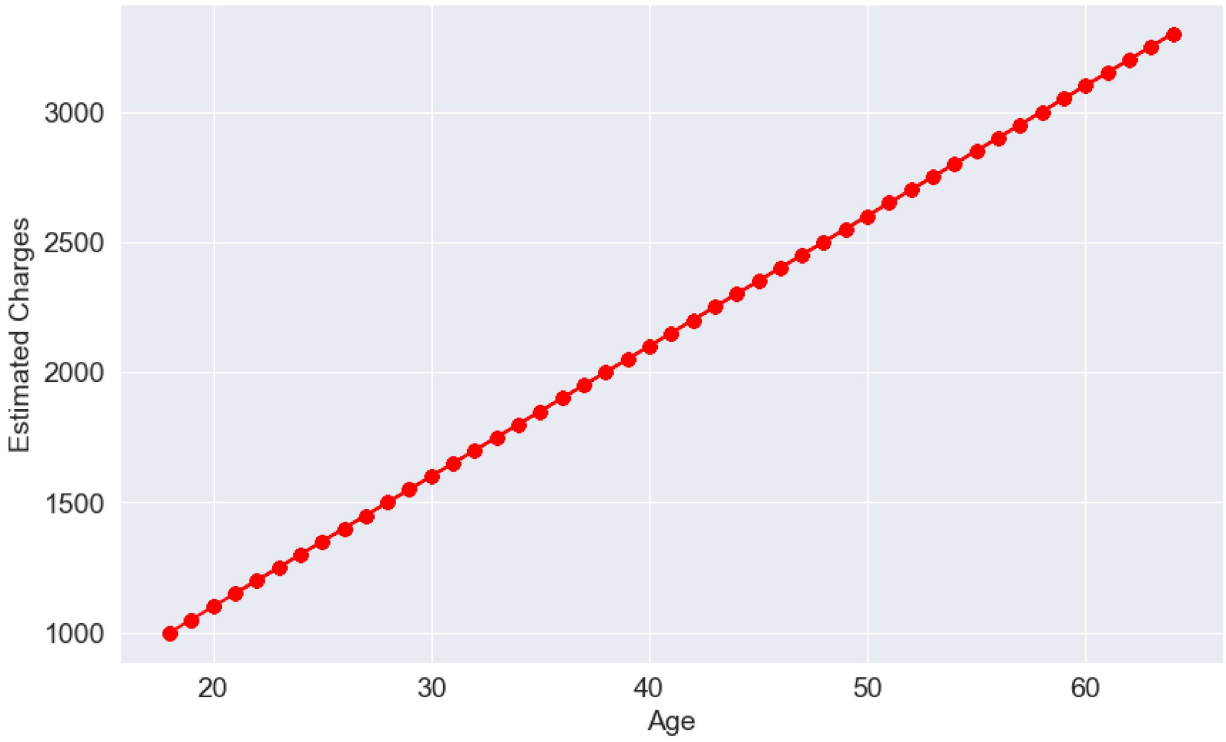


In [36]:
```
def estimate_charges(age, w, b):
    return w * age + b
```

In [37]:
```
w = 50
b = 100
```

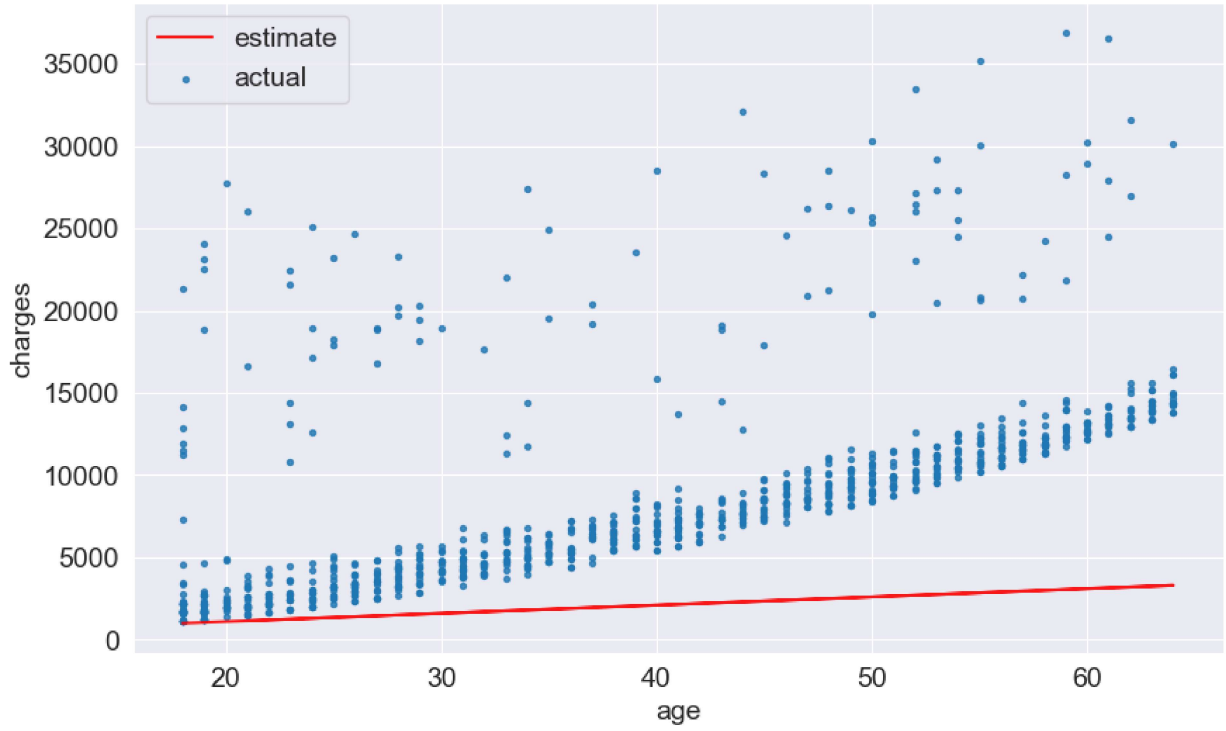
In [38]:
```
ages = non_smoker_df.age
estimated_charges = estimate_charges(ages, w, b)
```

In [39]:
```
plt.plot(ages, estimated_charges, 'r-o');
plt.xlabel('Age');
plt.ylabel('Estimated Charges');
```



In [40]:
```
target = non_smoker_df.charges

plt.plot(ages,estimated_charges, 'r', alpha=0.9);
plt.scatter(ages,target,s=8,alpha=0.8);
plt.xlabel('age');
plt.ylabel('charges')
plt.legend(['estimate','actual']);
```

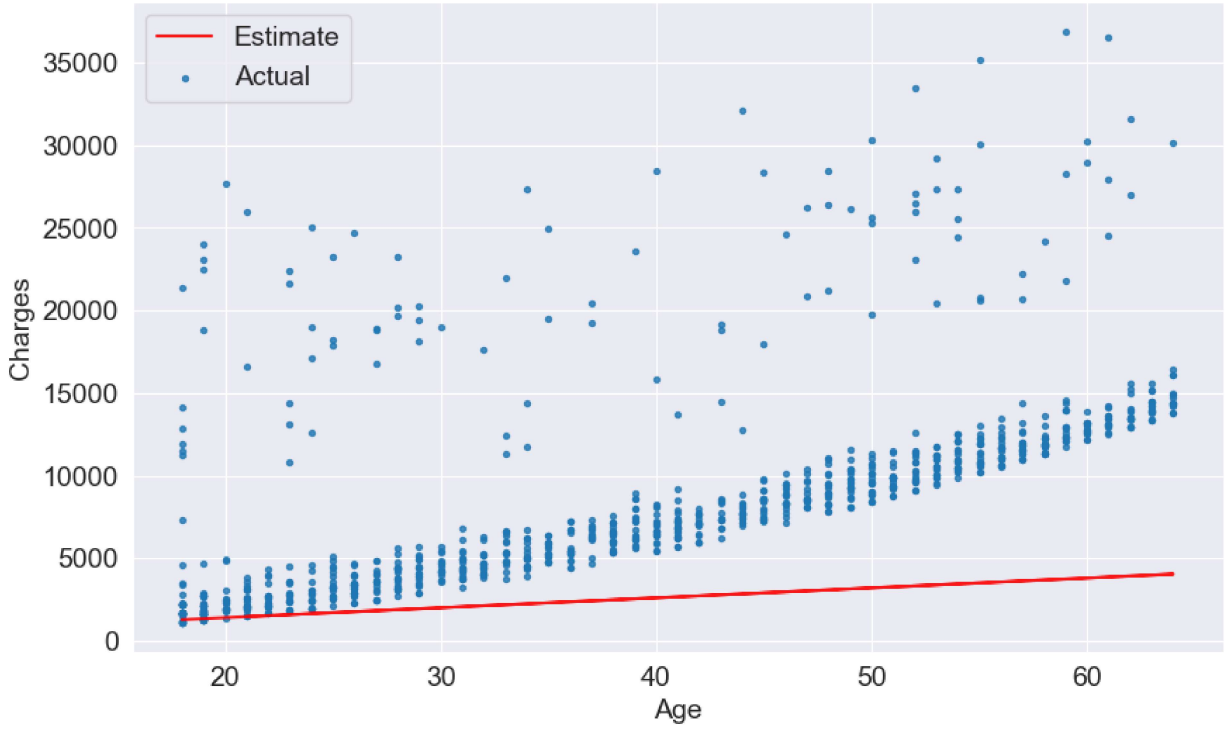


In [48]:
```
def try_parameters(w, b):
    ages = non_smoker_df.age
    target = non_smoker_df.charges
```

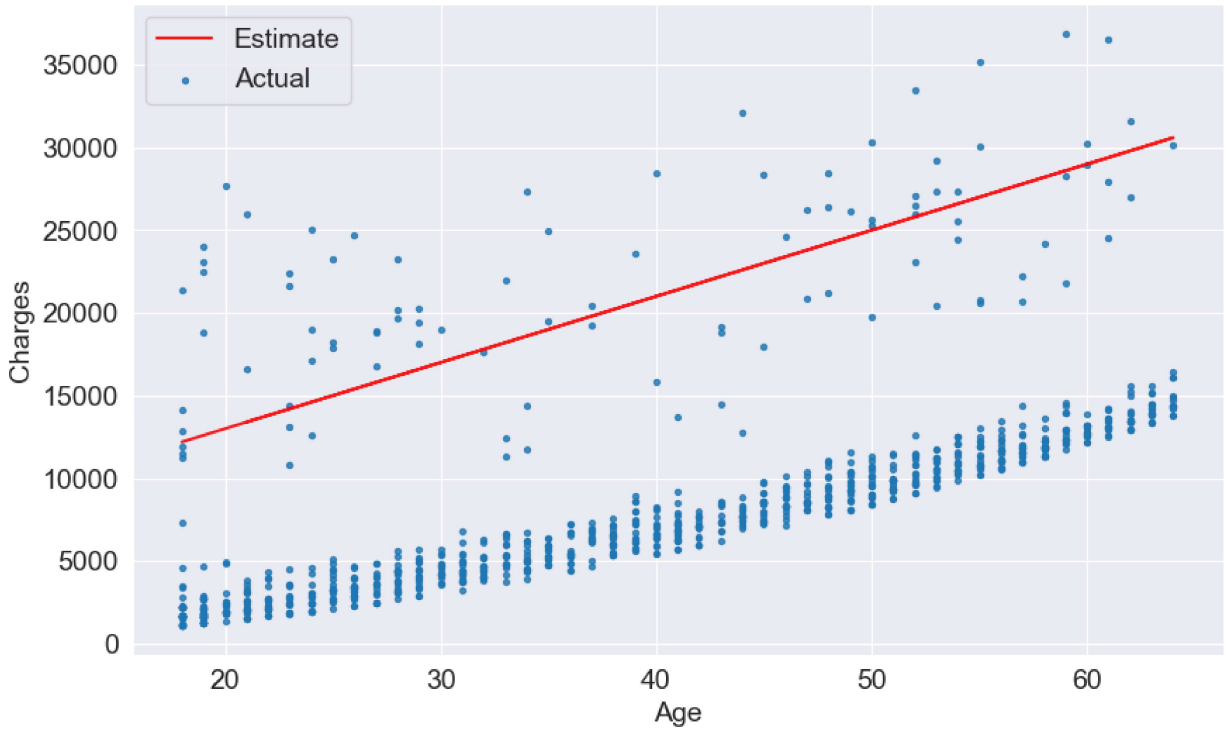
```
    estimated_charges = estimate_charges(ages, w, b)

    plt.plot(ages, estimated_charges, 'r', alpha=0.9);
    plt.scatter(ages, target, s=8,alpha=0.8);
    plt.xlabel('Age');
    plt.ylabel('Charges')
    plt.legend(['Estimate', 'Actual']);
```

In [49]:
```
try_parameters(60,200)
```



In [50]:
```
try_parameters(400,5000)
```



In [51]:
```
pip install numpy --quiet
```

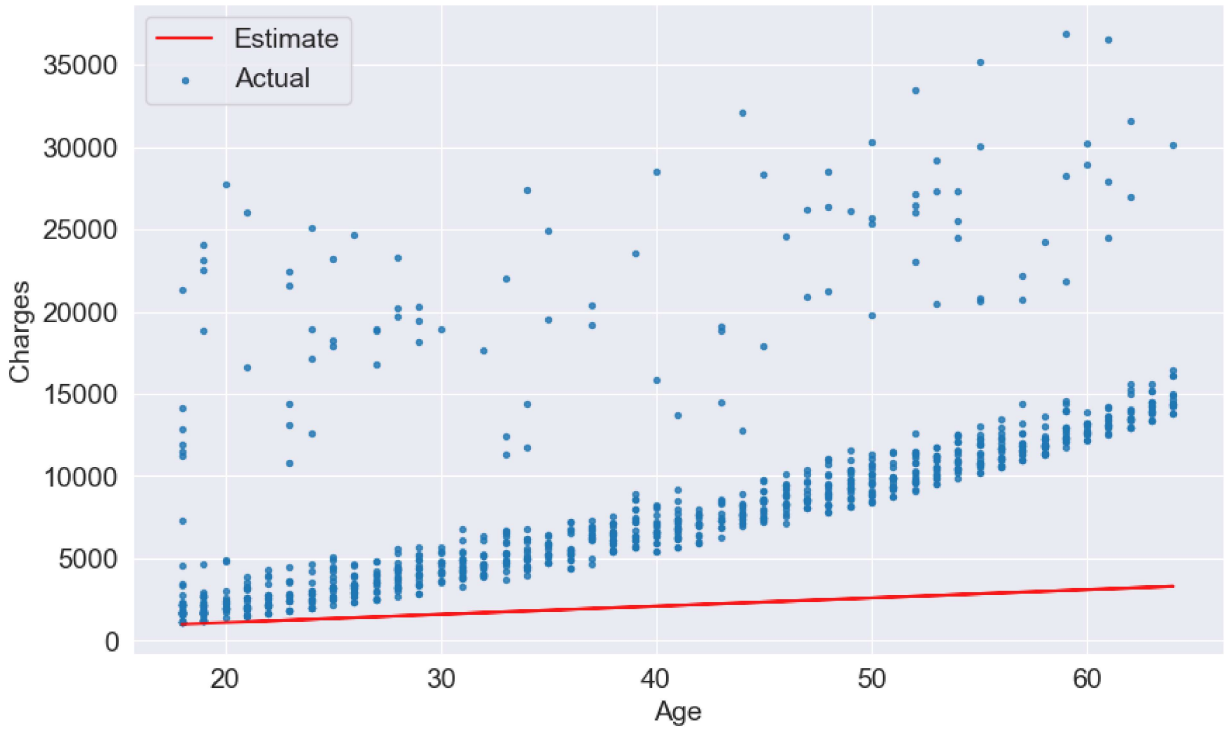
```
Note: you may need to restart the kernel to use updated packages.
```

In [52]:
```
import numpy as np
```

In [53]:
```
def rmse(targets, predictions):
    return np.sqrt(np.mean(np.square(targets - predictions)))
```

In [54]:
```
w = 50
b = 100
```

In [55]:
```
try_parameters(w, b)
```



In [56]:
```
targets = non_smoker_df['charges']
predicted = estimate_charges(non_smoker_df.age, w, b)
```

In [57]:
```
rmse(targets, predicted)
```

Out[57]:
```
8461.949562575493
```

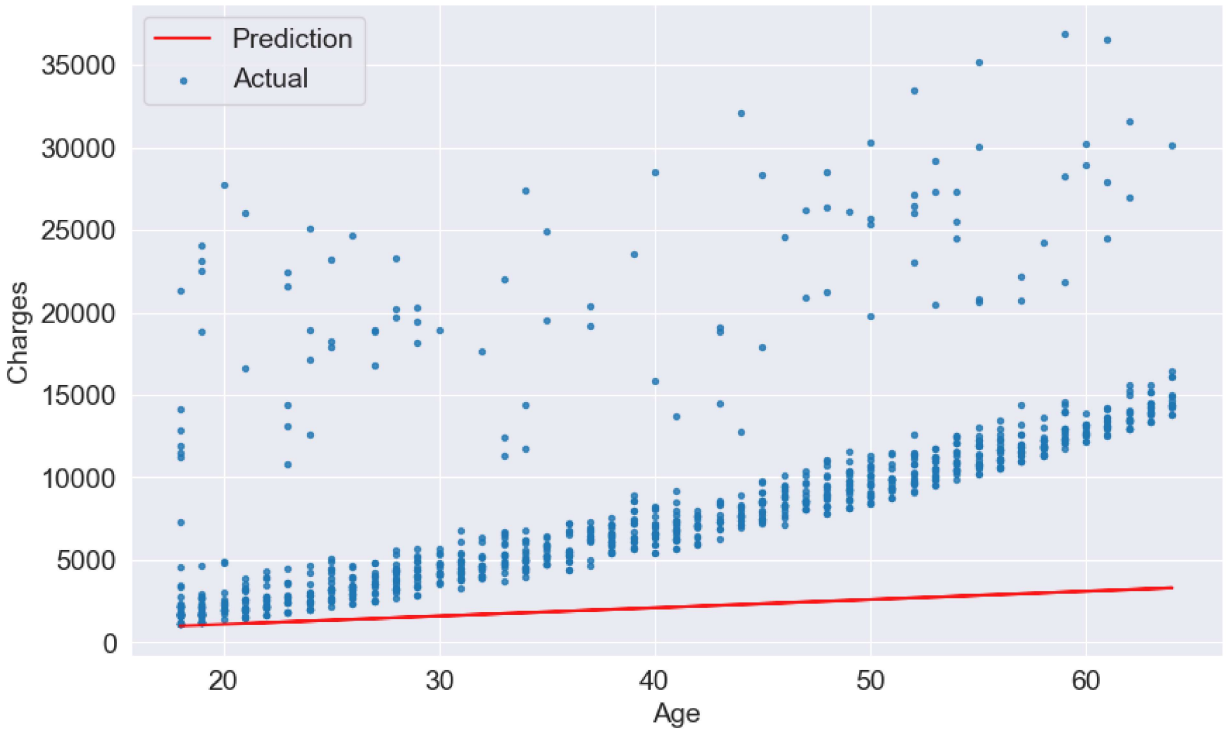
In [58]:
```
def try_parameters(w, b):
    ages = non_smoker_df.age
    target = non_smoker_df.charges
    predictions = estimate_charges(ages, w, b)

    plt.plot(ages, predictions, 'r', alpha=0.9);
    plt.scatter(ages, target, s=8,alpha=0.8);
    plt.xlabel('Age');
    plt.ylabel('Charges')
    plt.legend(['Prediction', 'Actual']);

    loss = rmse(target, predictions)
    print("RMSE Loss: ", loss)
```

In [59]:
```
try_parameters(50, 100)
```

```
RMSE Loss:  8461.949562575493
```


Prediction
Actual
35000
30000
25000
20000
15000
10000
5000
0
Charges
20
30
40
50
60
Age