```
import pandas as pd
```

```
medical_df = pd.read_csv('medical.csv')
```

```
medical_df
```

| | 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

```
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
```

```
medical_df.describe()
```

| | 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 |

```
!pip install plotly matplotlib seaborn --quiet
```

```
import plotly.express as px
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

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

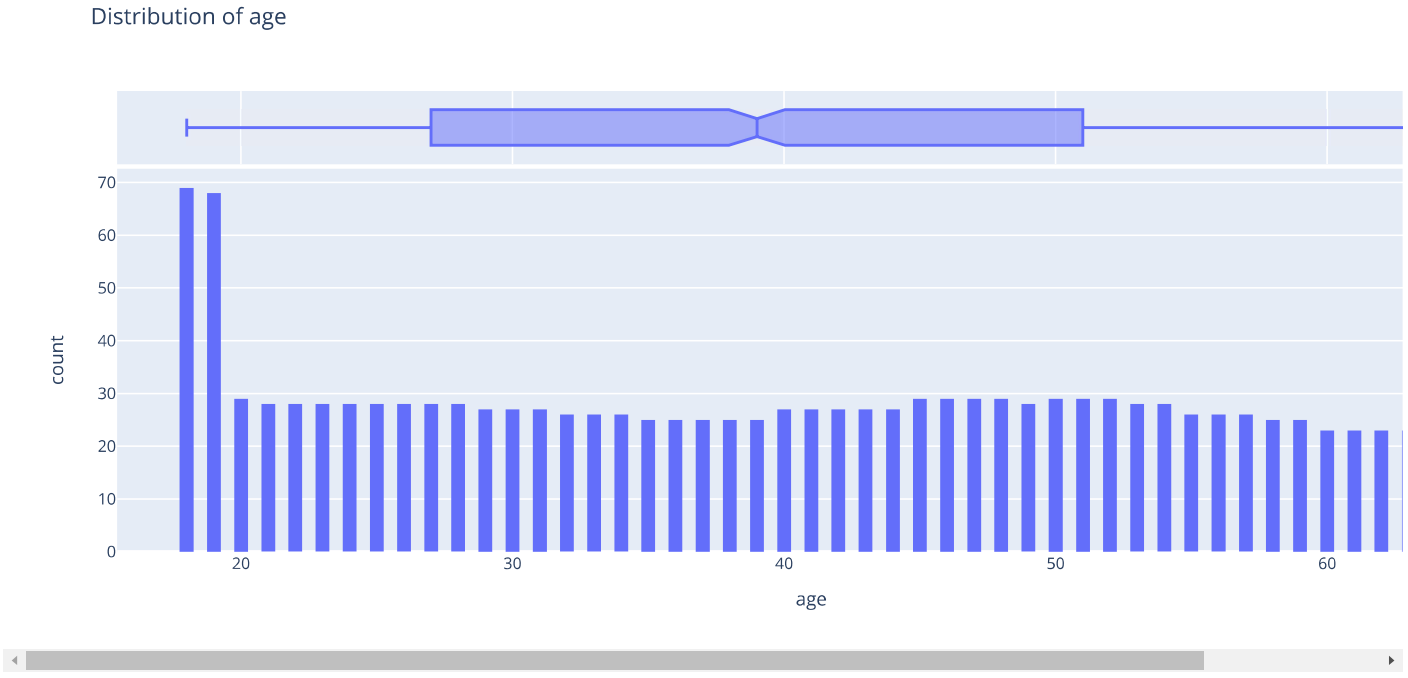
```
medical_df.age.describe()
```

```
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
```

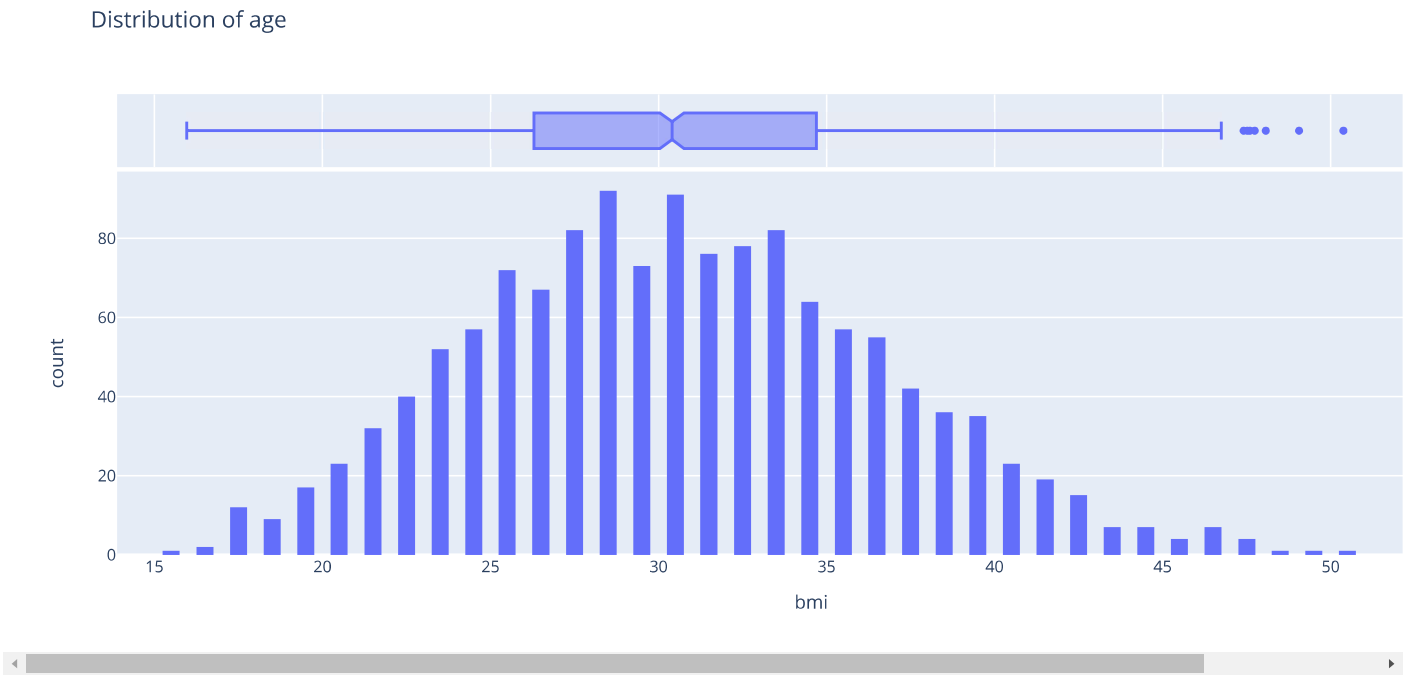
```
medical_df.charges.describe()
```

```
count     1338.000000
mean     13270.422265
std      12110.011237
min       1121.873900
25%       4740.287150
50%       9382.033000
75%      16639.912515
max      63770.428010
Name: charges, dtype: float64
```

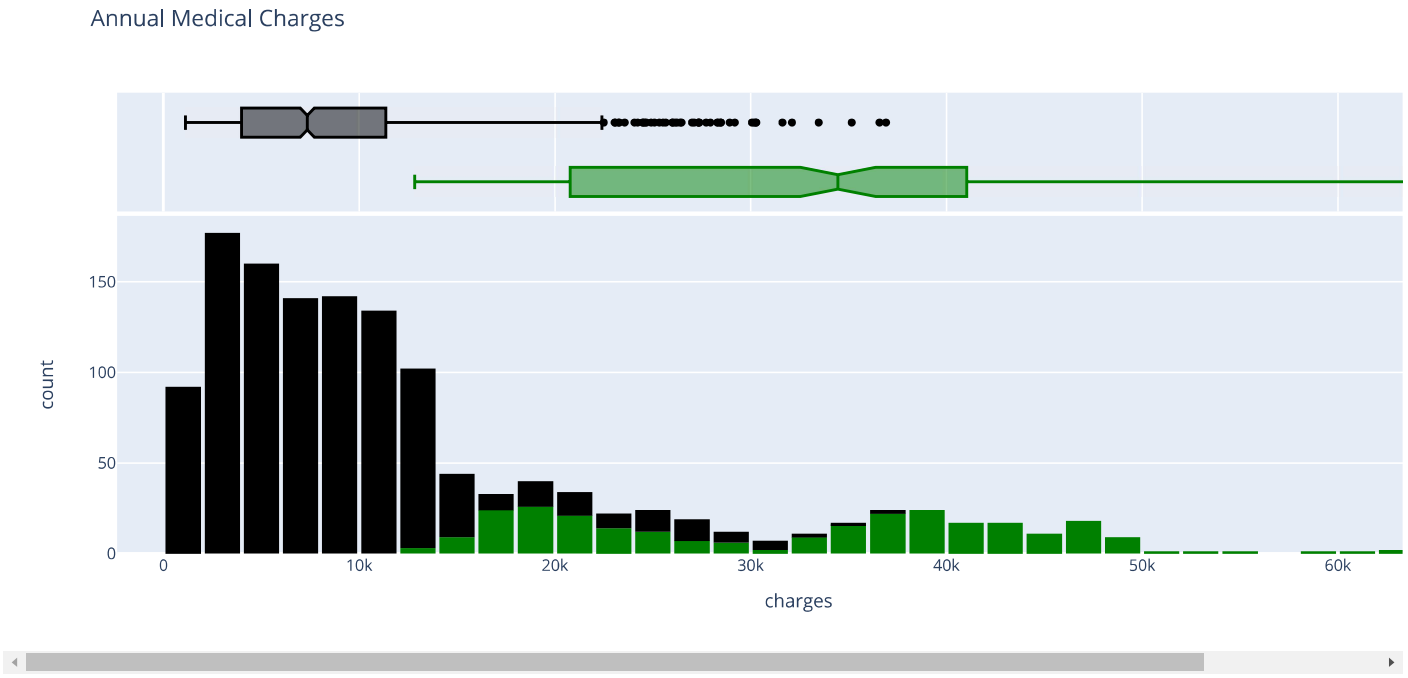
```
fig=px.histogram(medical_df,x="age",marginal="box",nbins=47,title="Distribution of age")
fig.update_layout(bargap=0.5)
fig.show()
```



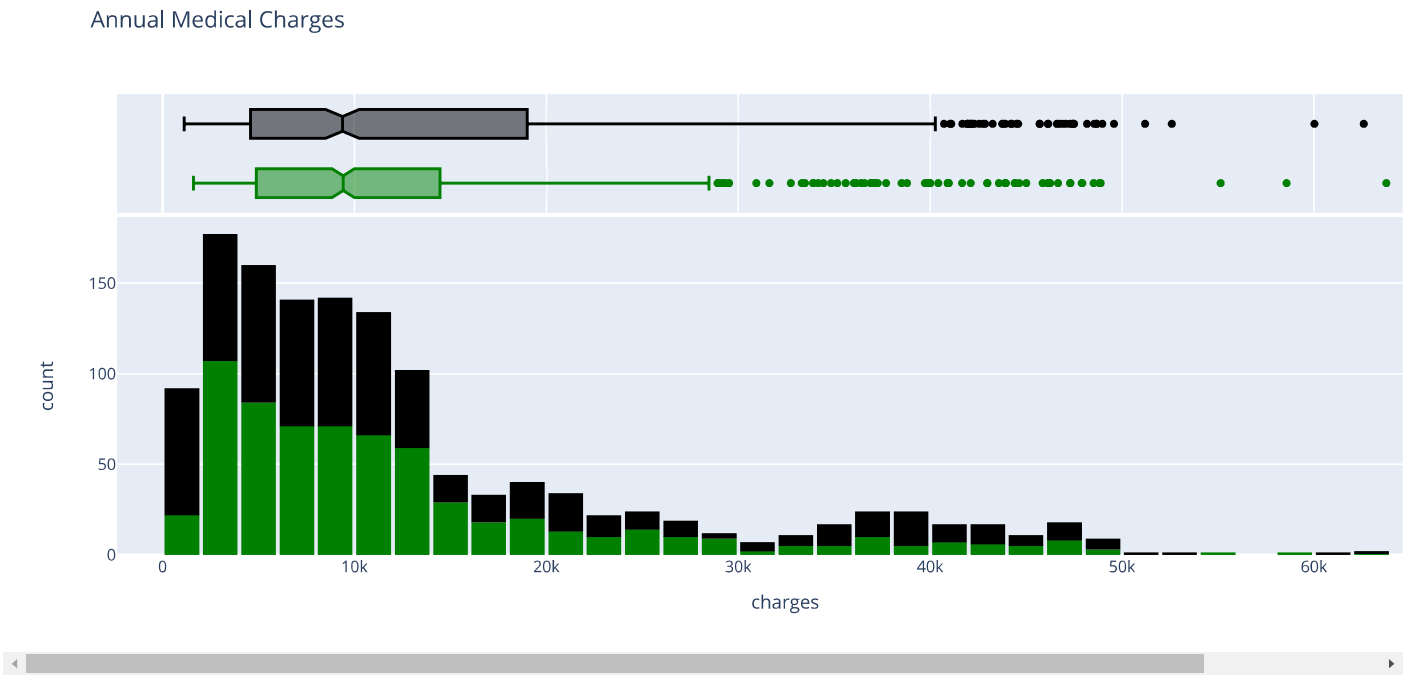

```
fig=px.histogram(medical_df,x="bmi",marginal="box",nbins=47,title="Distribution of age")
fig.update_layout(bargap=0.5)
fig.show()
```



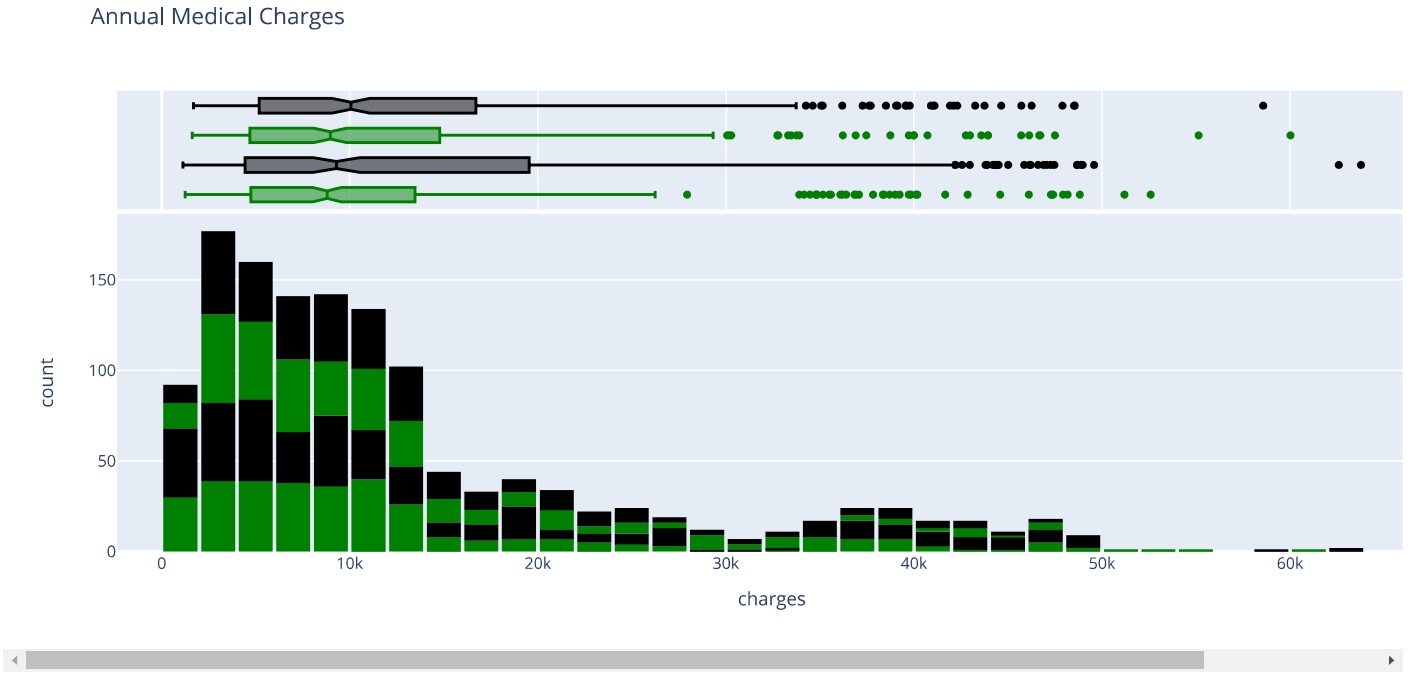

```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal='box',
                   color='smoker',
                   color_discrete_sequence=['green', 'black'],
                   title='Annual Medical Charges')
fig.update_layout(bargap=0.1)
fig.show()
```




```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal='box',
                   color='sex',
                   color_discrete_sequence=['green', 'black'],
                   title='Annual Medical Charges')
fig.update_layout(bargap=0.1)
fig.show()
```



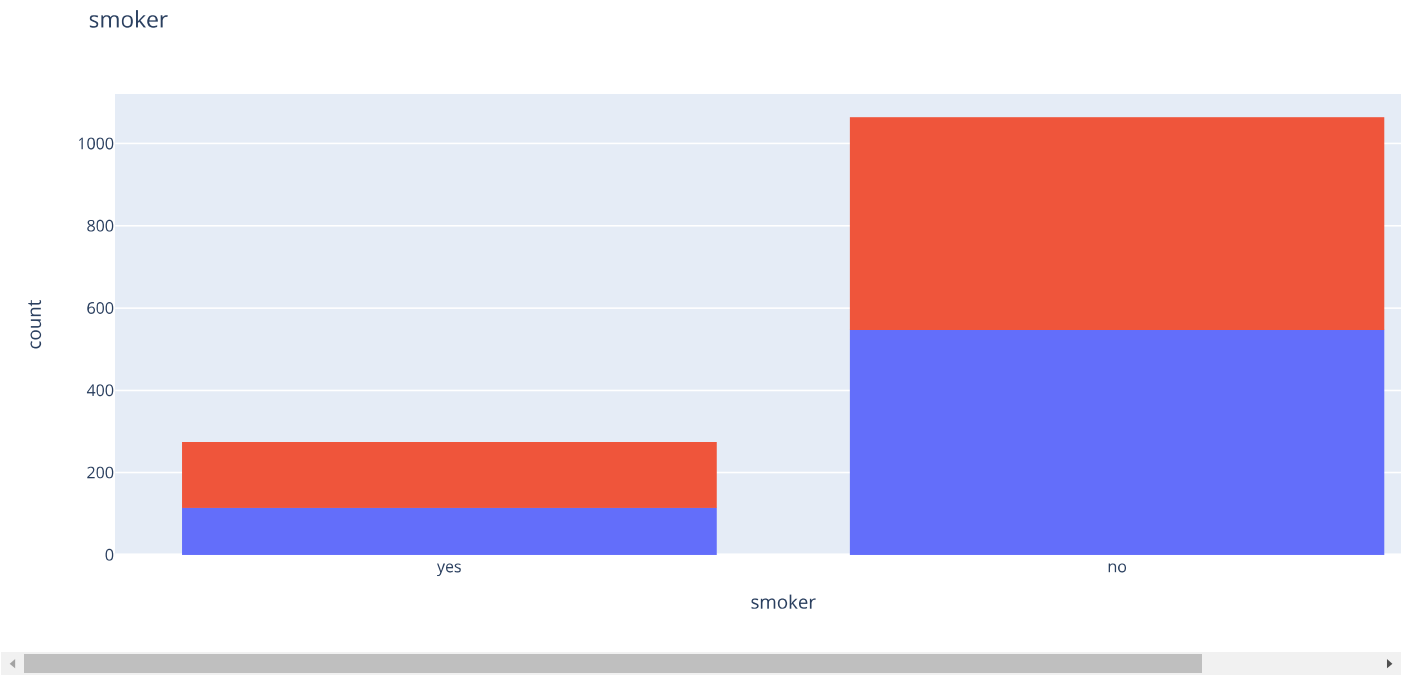

```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal='box',
                   color='region',
                   color_discrete_sequence=['green', 'black'],
                   title='Annual Medical Charges')
fig.update_layout(bargap=0.1)
fig.show()
```



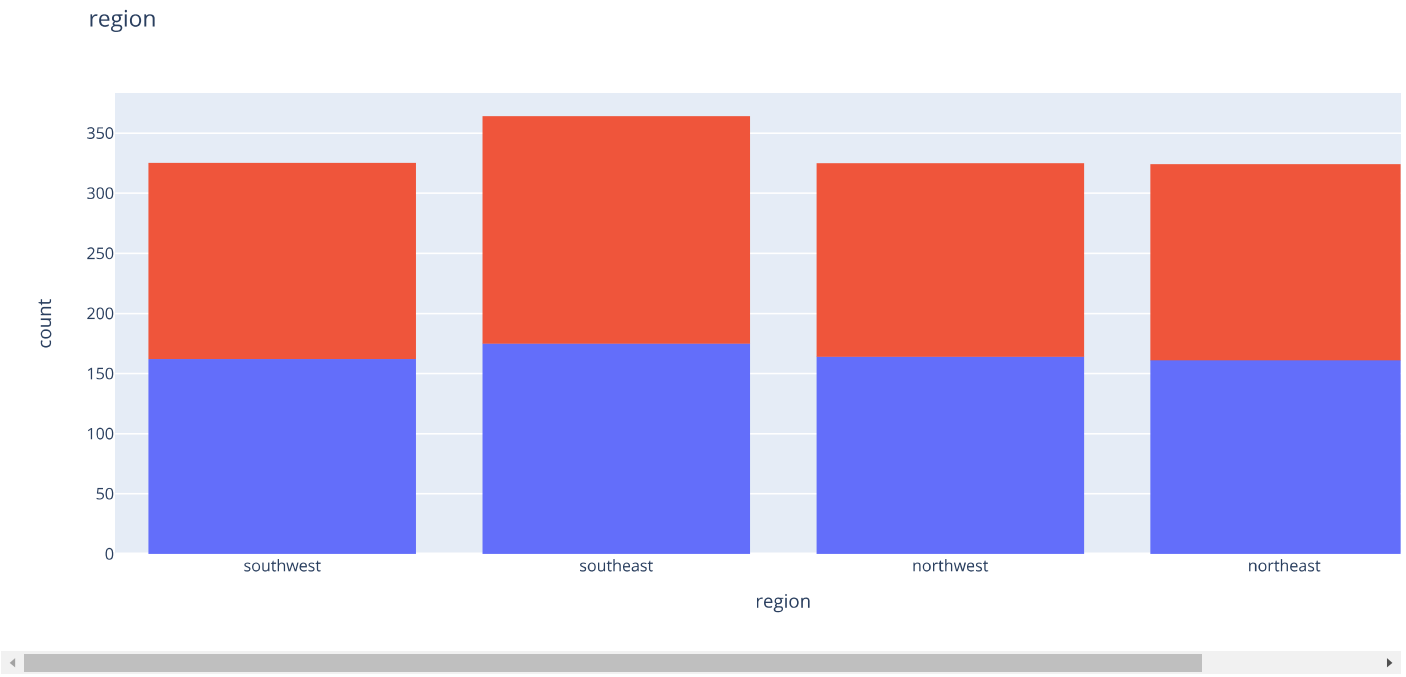

```
medical_df.smoker.value_counts()
```

```
no     1064
yes     274
Name: smoker, dtype: int64
```

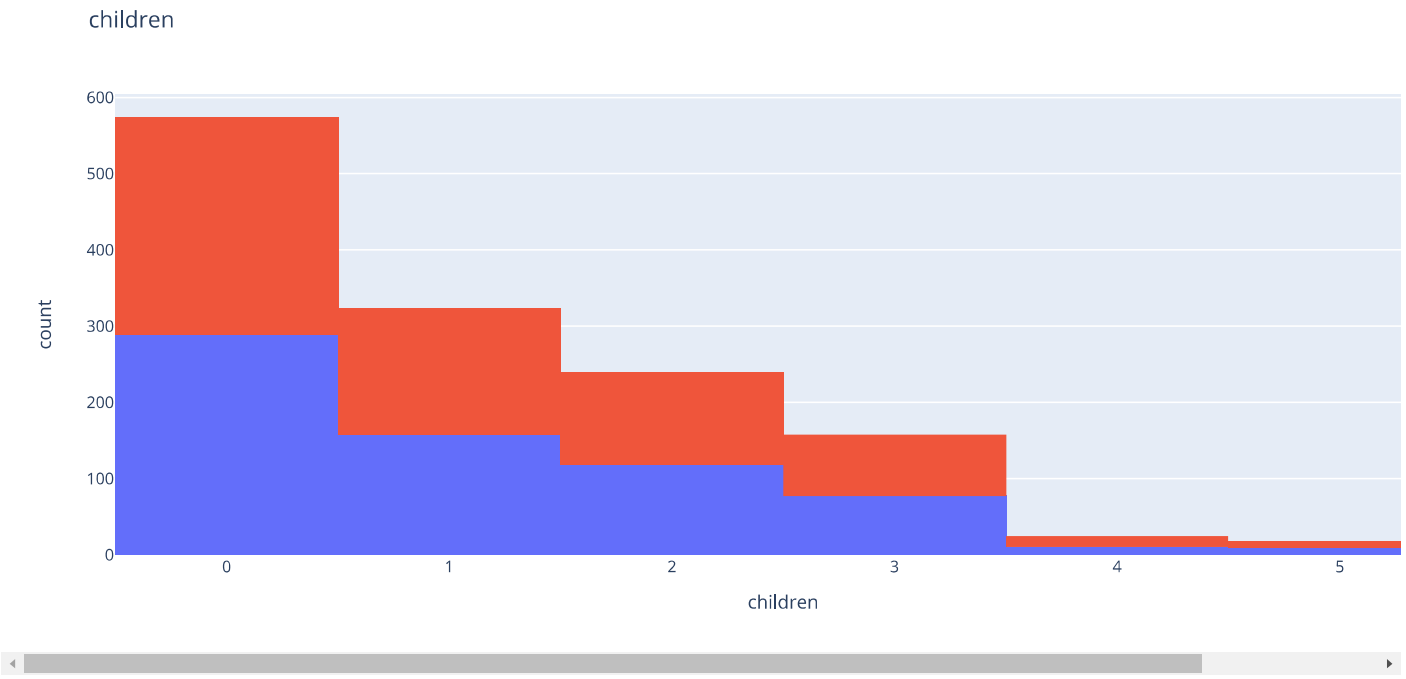
```
px.histogram(medical_df,x="smoker",color="sex",title="smoker")
```



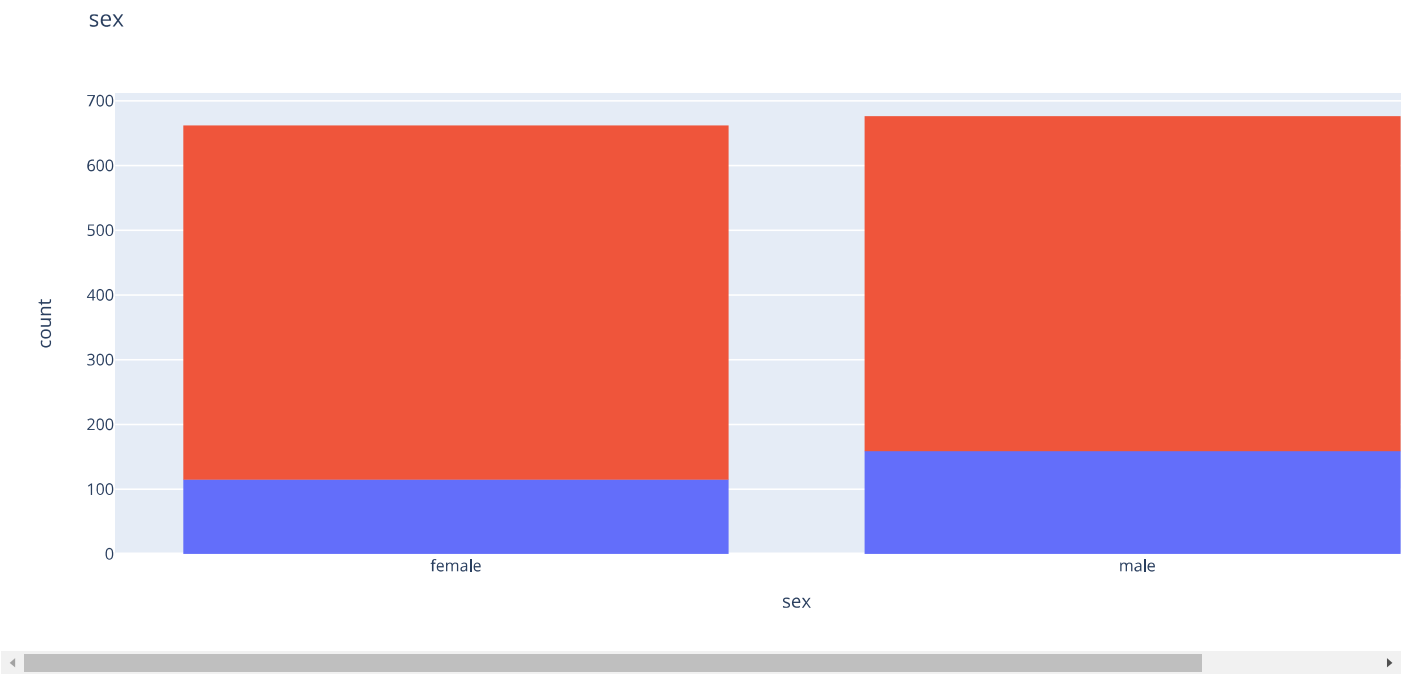

```
px.histogram(medical_df,x="region",color="sex",title="region")
```




```
px.histogram(medical_df,x="children",color="sex",title="children")
```

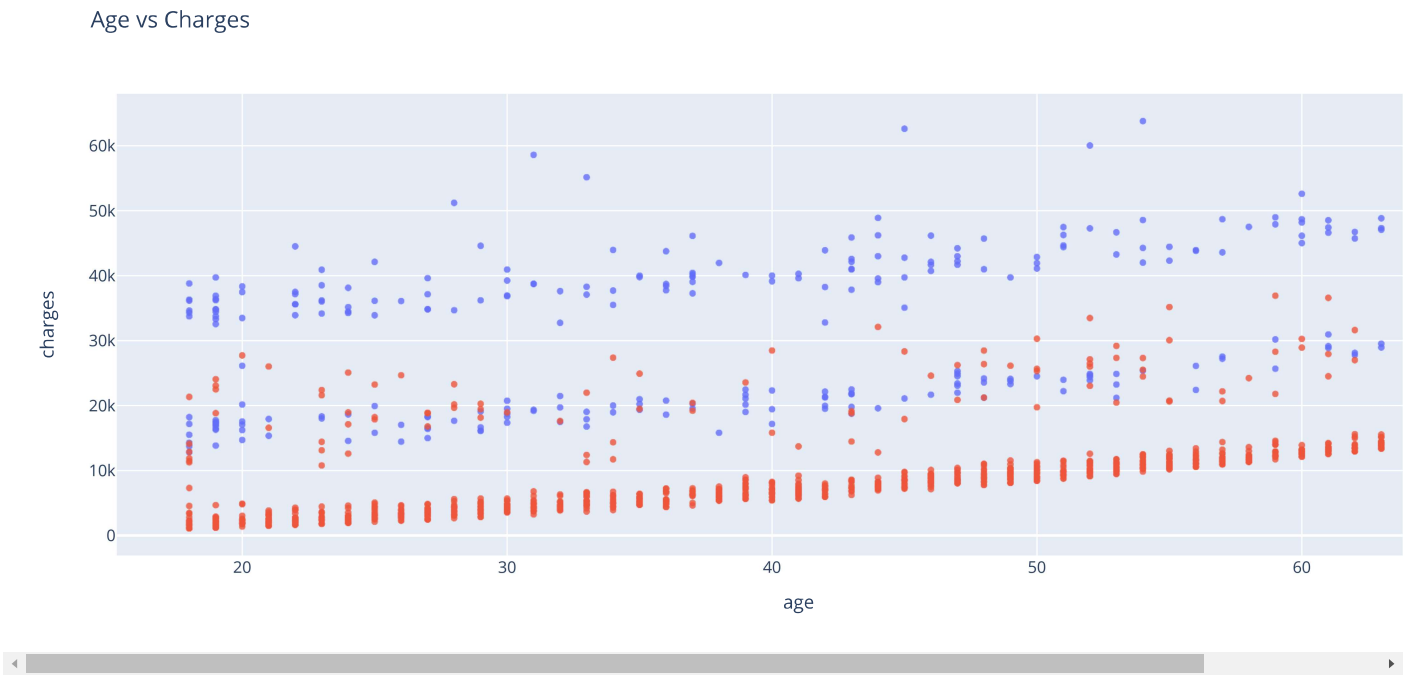



```
px.histogram(medical_df,x="sex",color="smoker",title="sex")
```

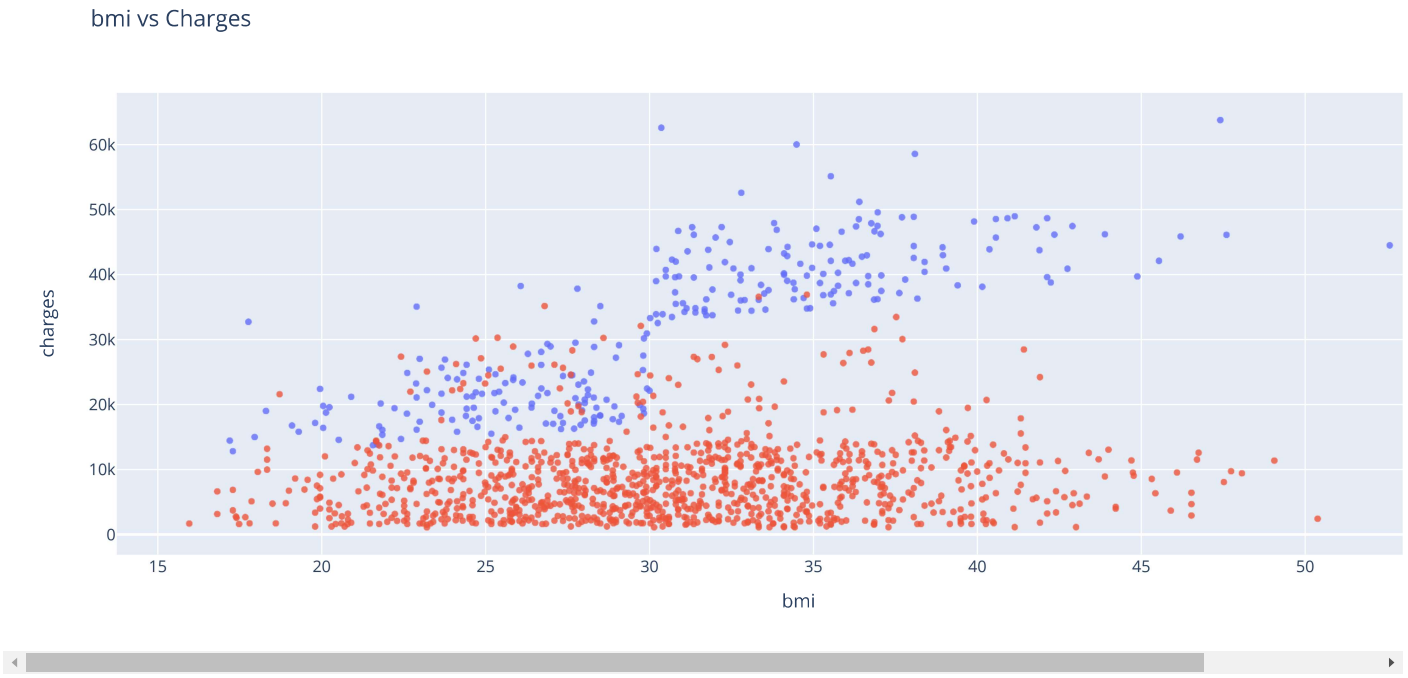



## Age and charges

```
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()
```

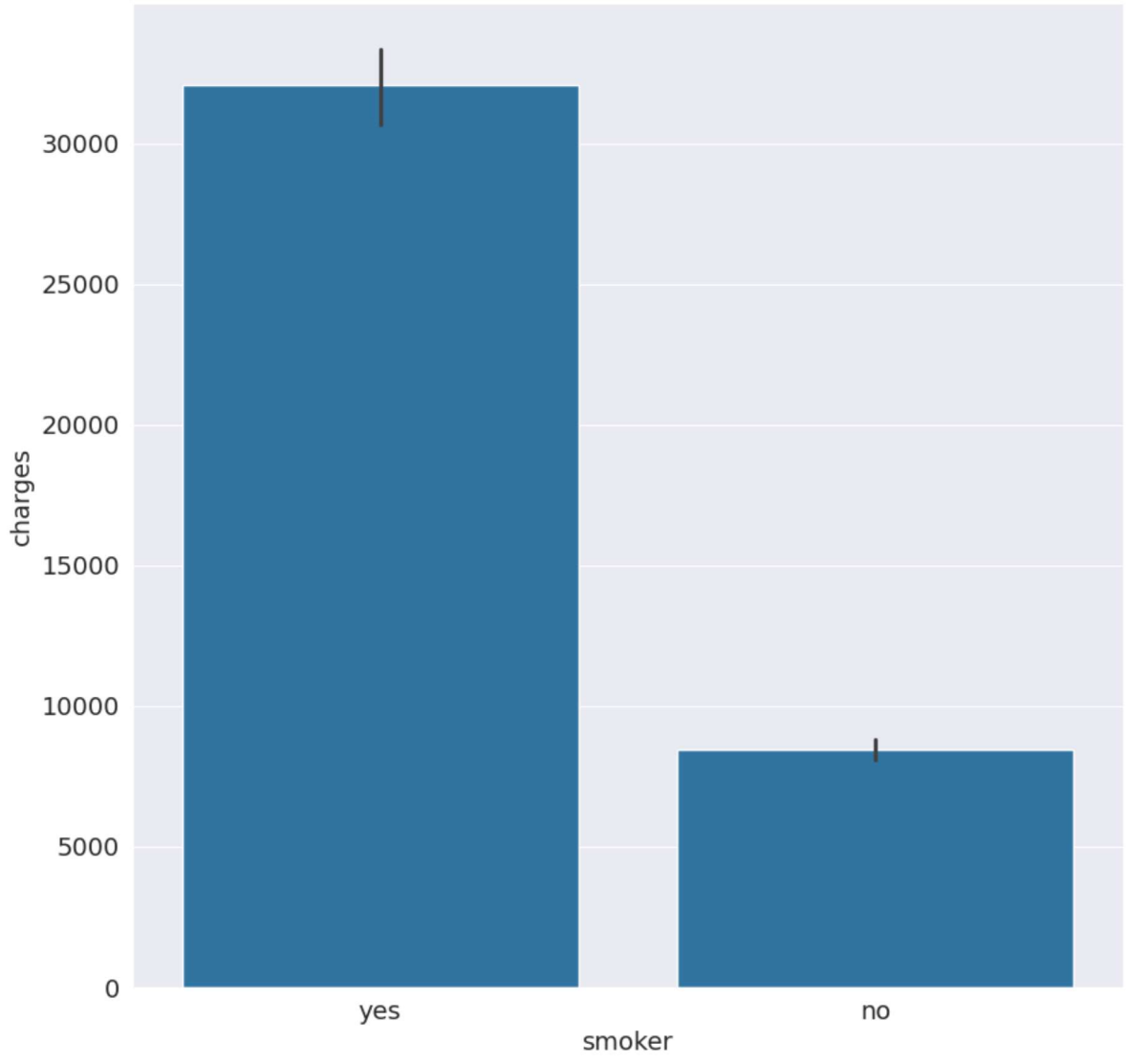



```
fig=px.scatter(medical_df,x="bmi",y="charges",color="smoker",opacity=0.8,hover_data=["sex"],title="bmi vs Charges")
fig.update_traces(marker_size=5)
fig.show()
```

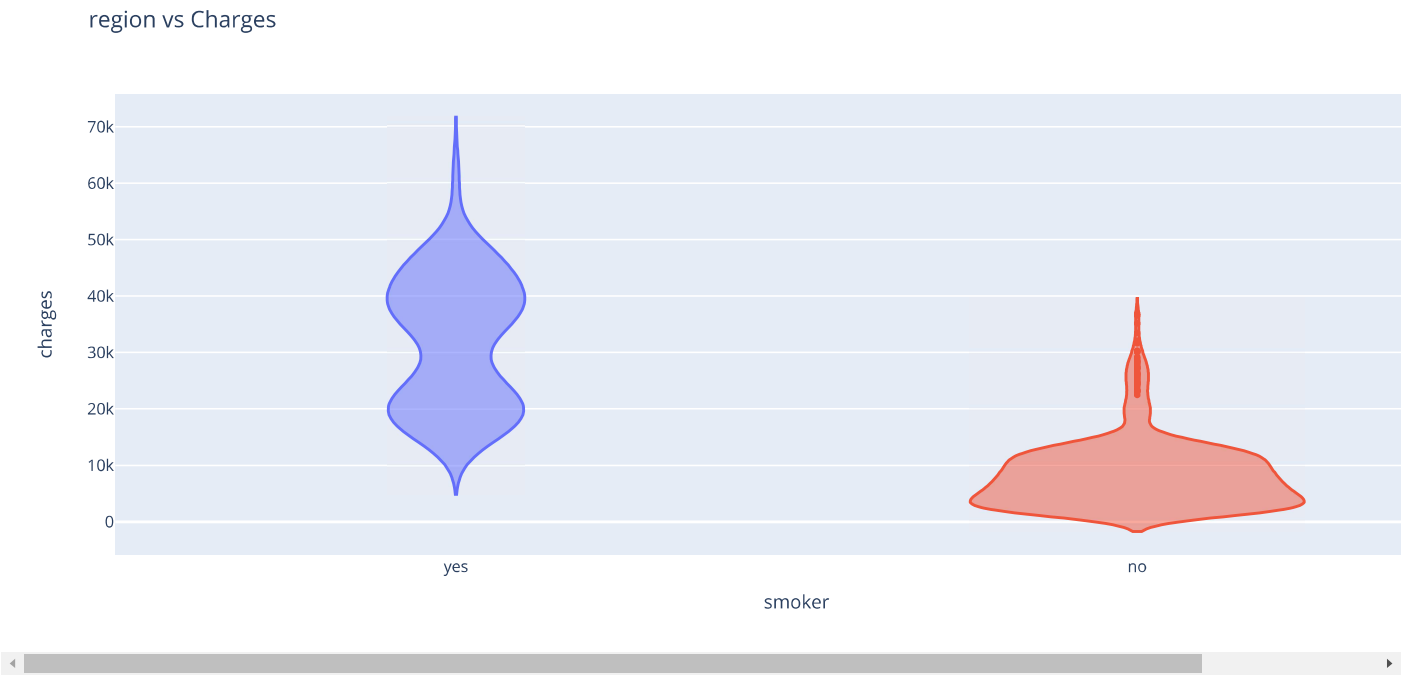



```
fig=px.scatter(medical_df,x="smoker",y="charges",color="smoker",opacity=0.8,hover_data=["sex"],title="region vs Charges")
fig.update_traces(marker_size=5)
fig.show()
```

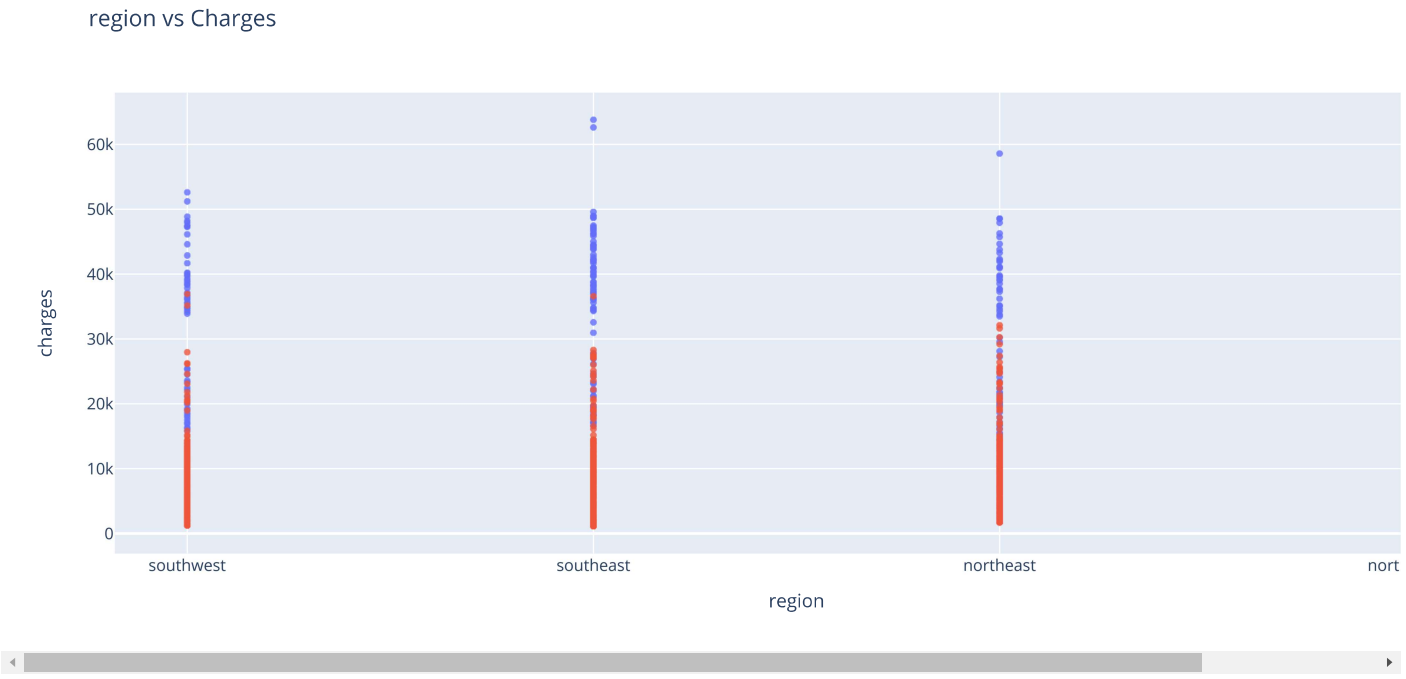

```
fig=sns.barplot(data=medical_df,x="smoker",y="charges")
```



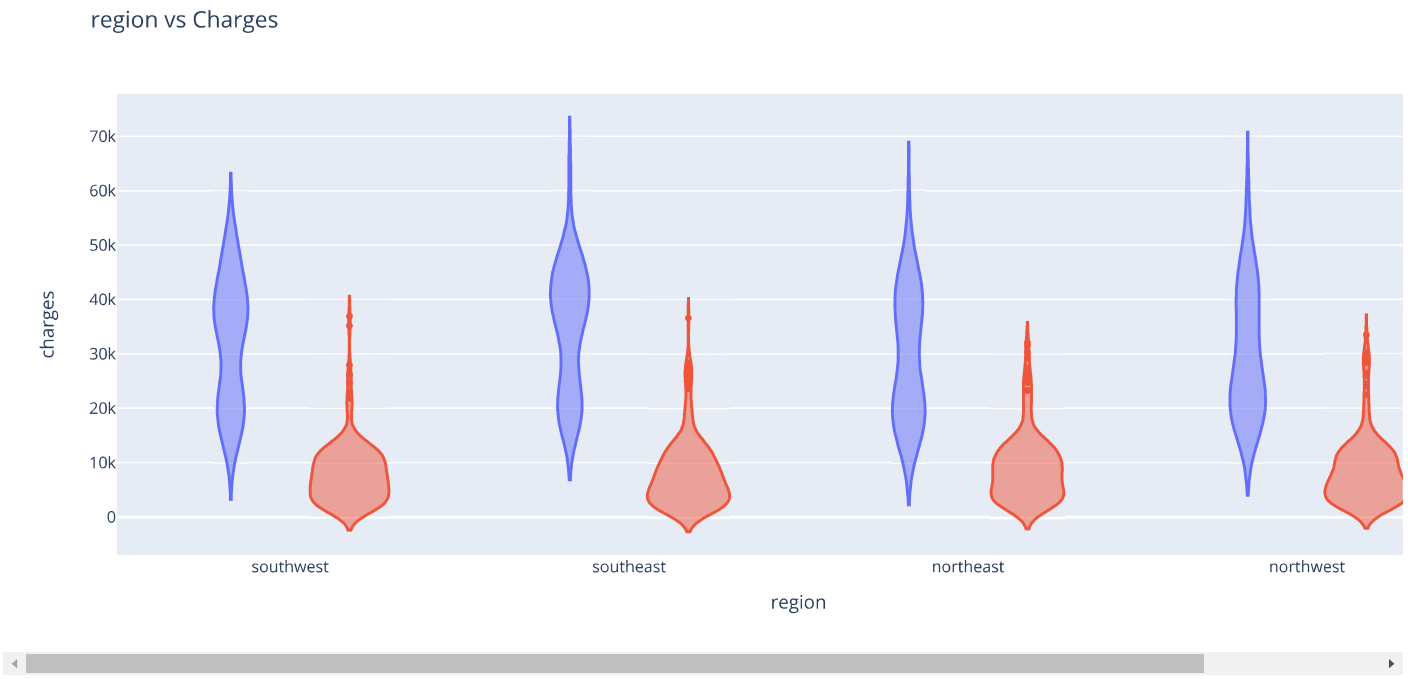
```
fig=px.violin(medical_df,x="smoker",y="charges",color="smoker",hover_data=["sex"],title="region vs Charges")
fig.update_traces(marker_size=5)
fig.show()
```



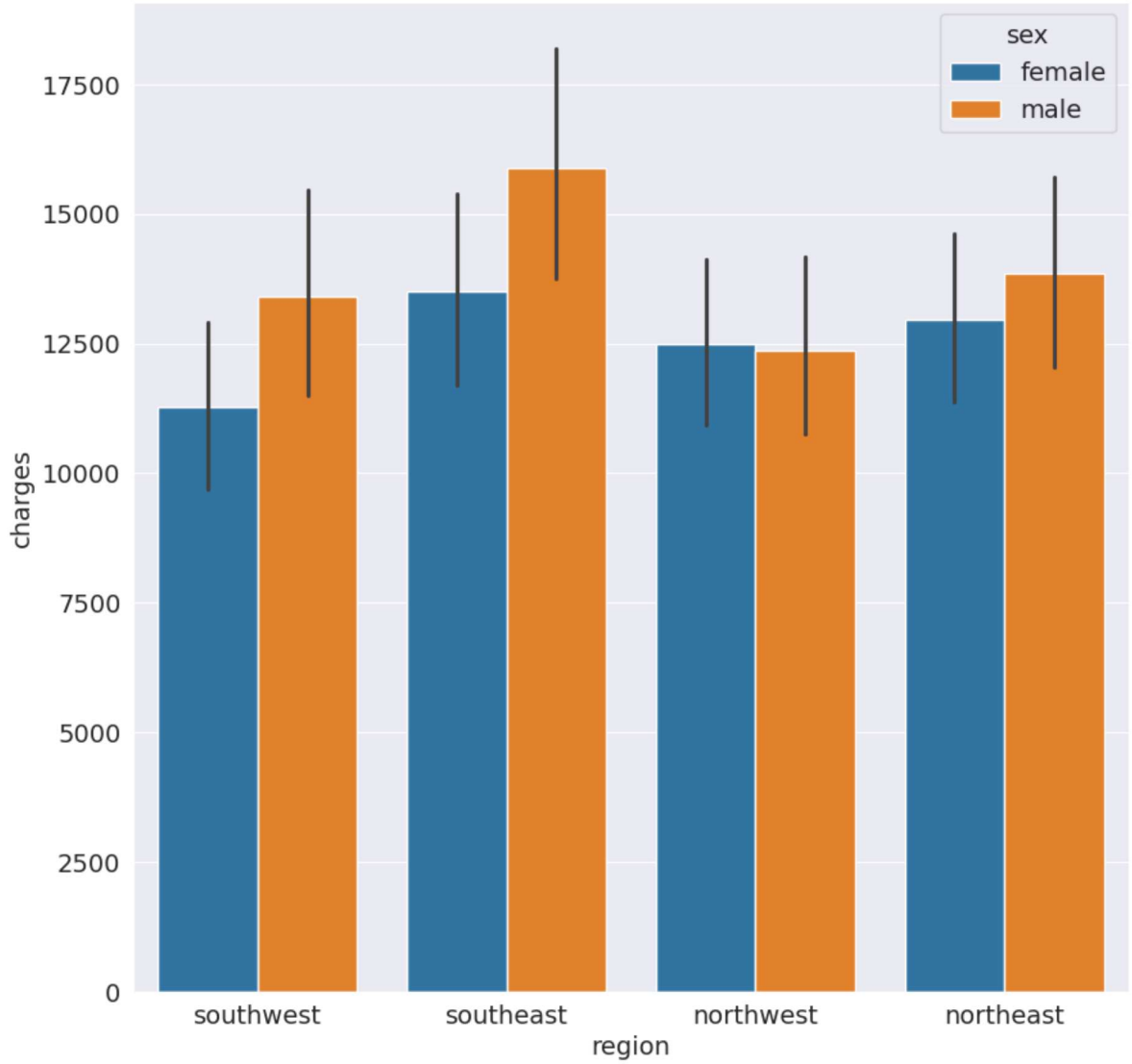

```
fig=px.scatter(medical_df,x="region",y="charges",color="smoker",opacity=0.8,hover_data=["sex"],title="region vs Charges")
fig.update_traces(marker_size=5)
fig.show()
```



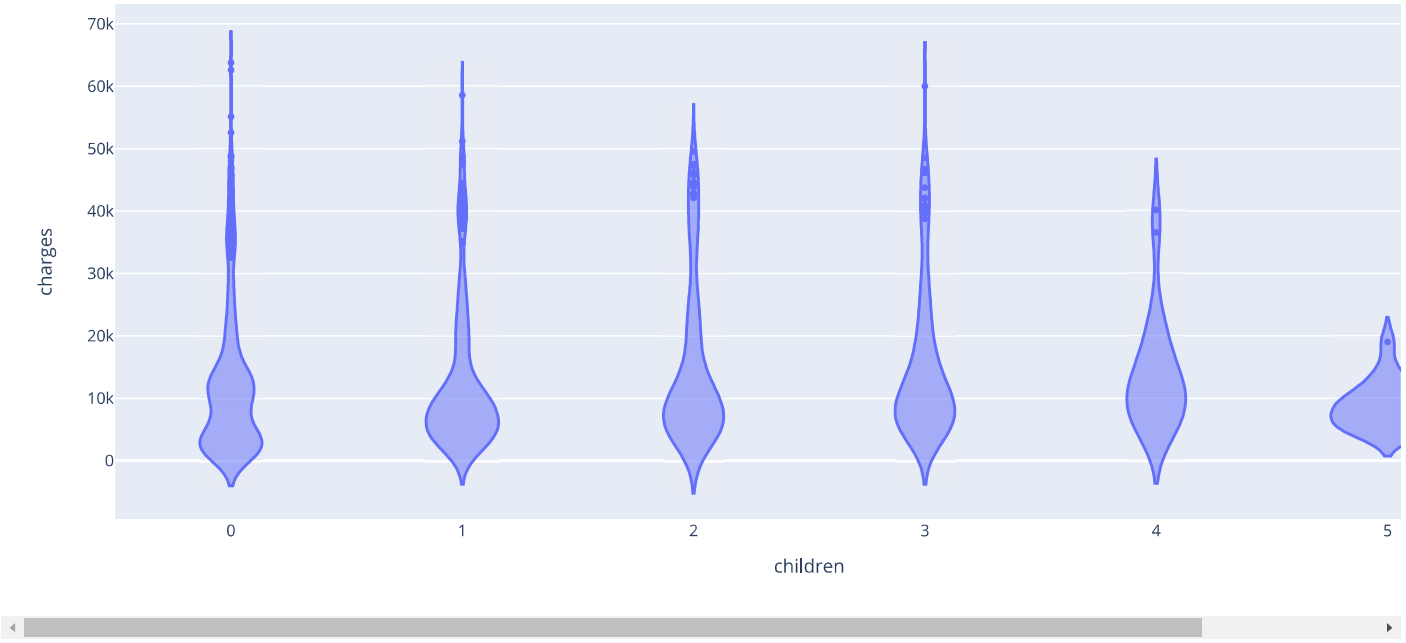

```
fig=px.violin(medical_df,x="region",y="charges",color="smoker",hover_data=["sex"],title="region vs Charges")
fig.update_traces(marker_size=5)
fig.show()
```



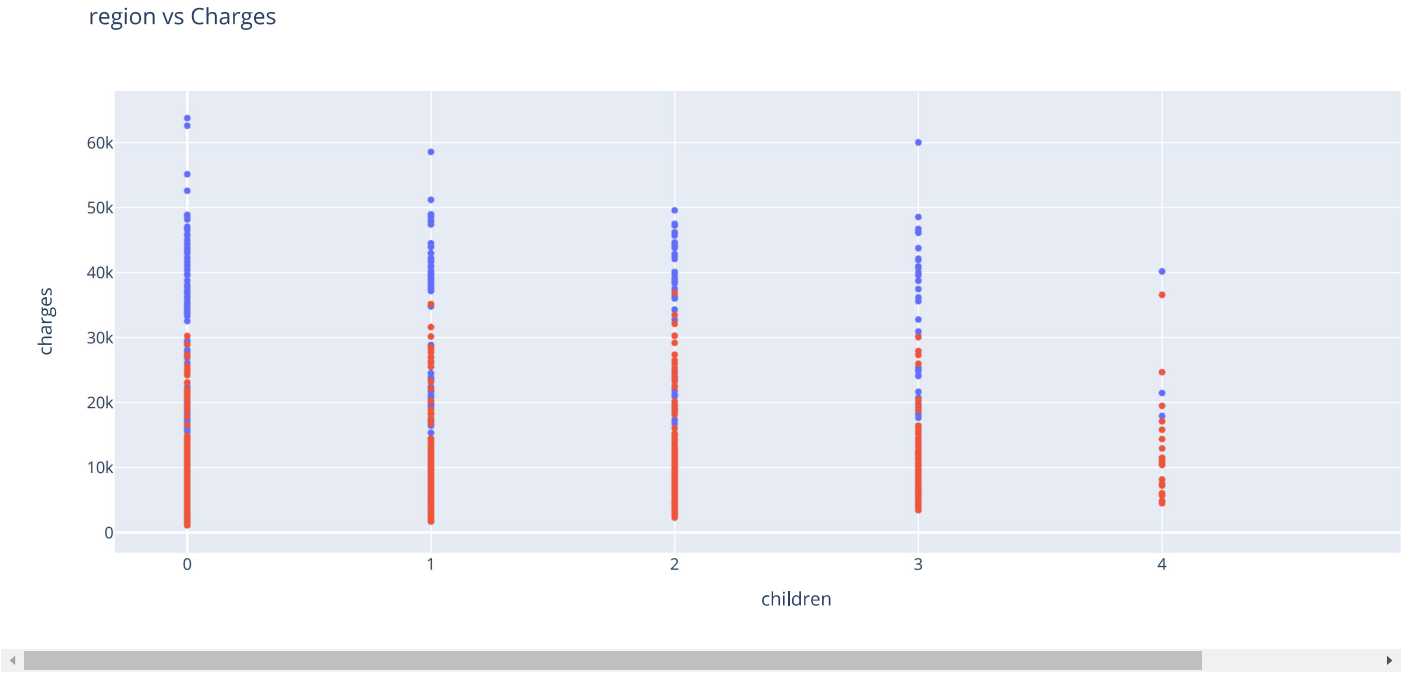

```
fig=sns.barplot(data=medical_df,x="region",y="charges",hue="sex")
#fig.update_traces(marker_size=5)
#fig.show()
```



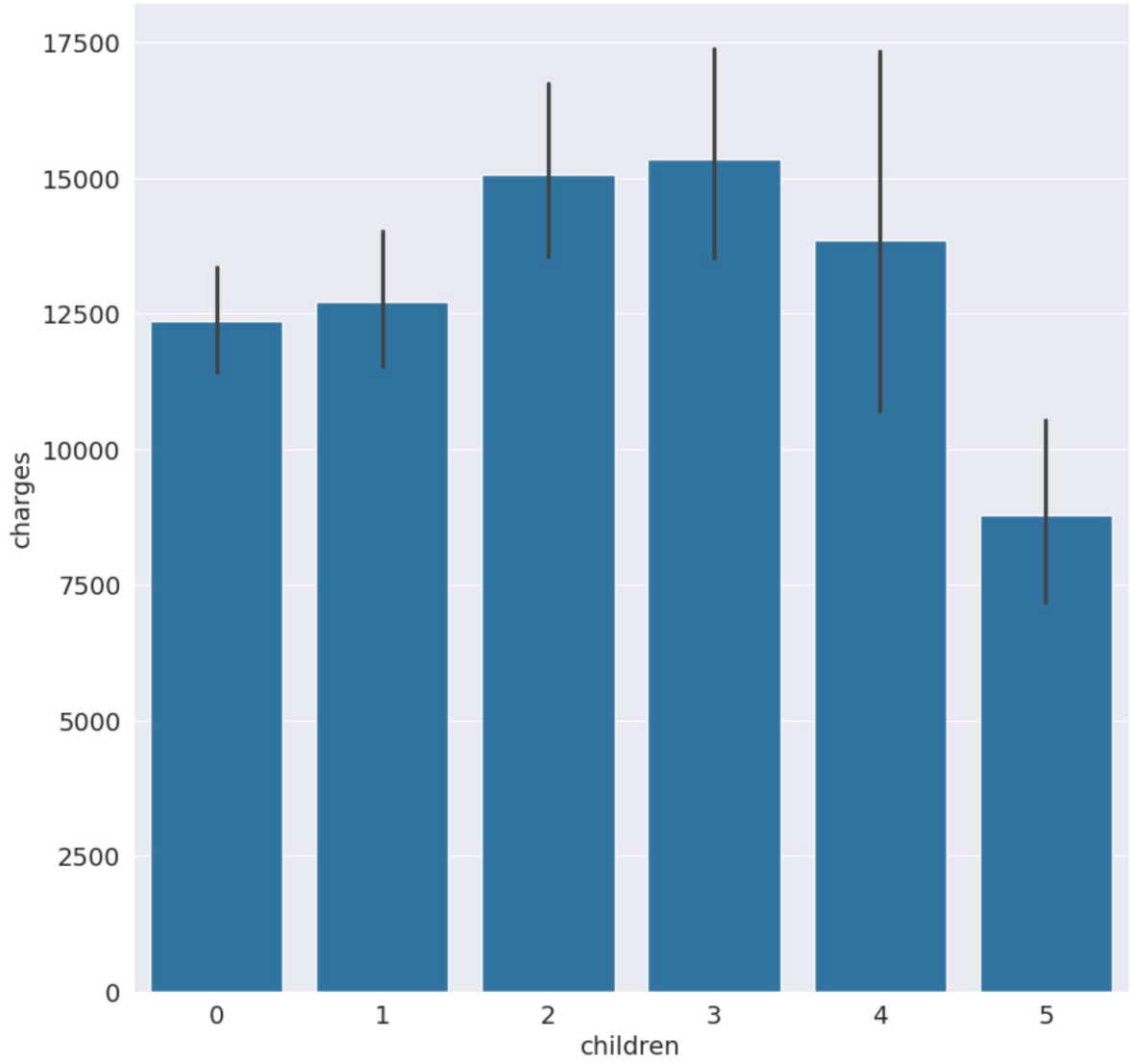

```
fig=px.violin(medical_df,x="children",y="charges")
fig.update_traces(marker_size=5)
fig.show()
```




```
fig=px.scatter(medical_df,x="children",y="charges",color="smoker",hover_data=["sex"],title="region vs Charges")
fig.update_traces(marker_size=5)
fig.show()
```




```
fig=sns.barplot(data=medical_df,x="children",y="charges")
```




## correlation

```
medical_df.charges.corr(medical_df.age)
```

```
0.2990081933306476
```

```
medical_df.charges.corr(medical_df.bmi)
```

```
0.19834096883362895
```

```
medical_df.charges.corr(medical_df.children)
```

```
0.06799822684790478
```

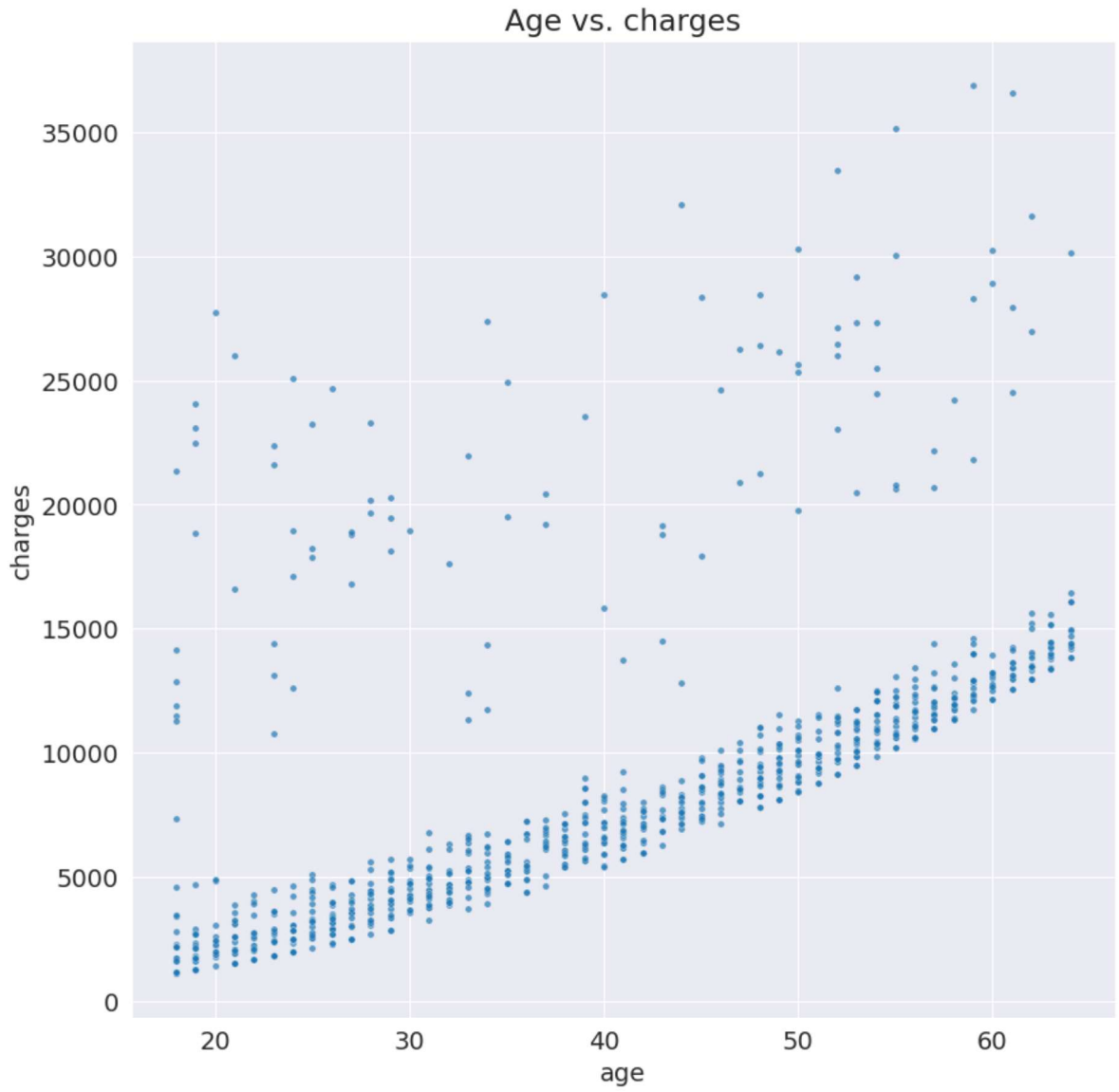
```
smoker_values={"no":0,"yes":1}
smoker_numeric=medical_df.smoker.map(smoker_values)
medical_df.charges.corr(smoker_numeric)
```

```
0.787251430498478
```

## linear regression using single feature

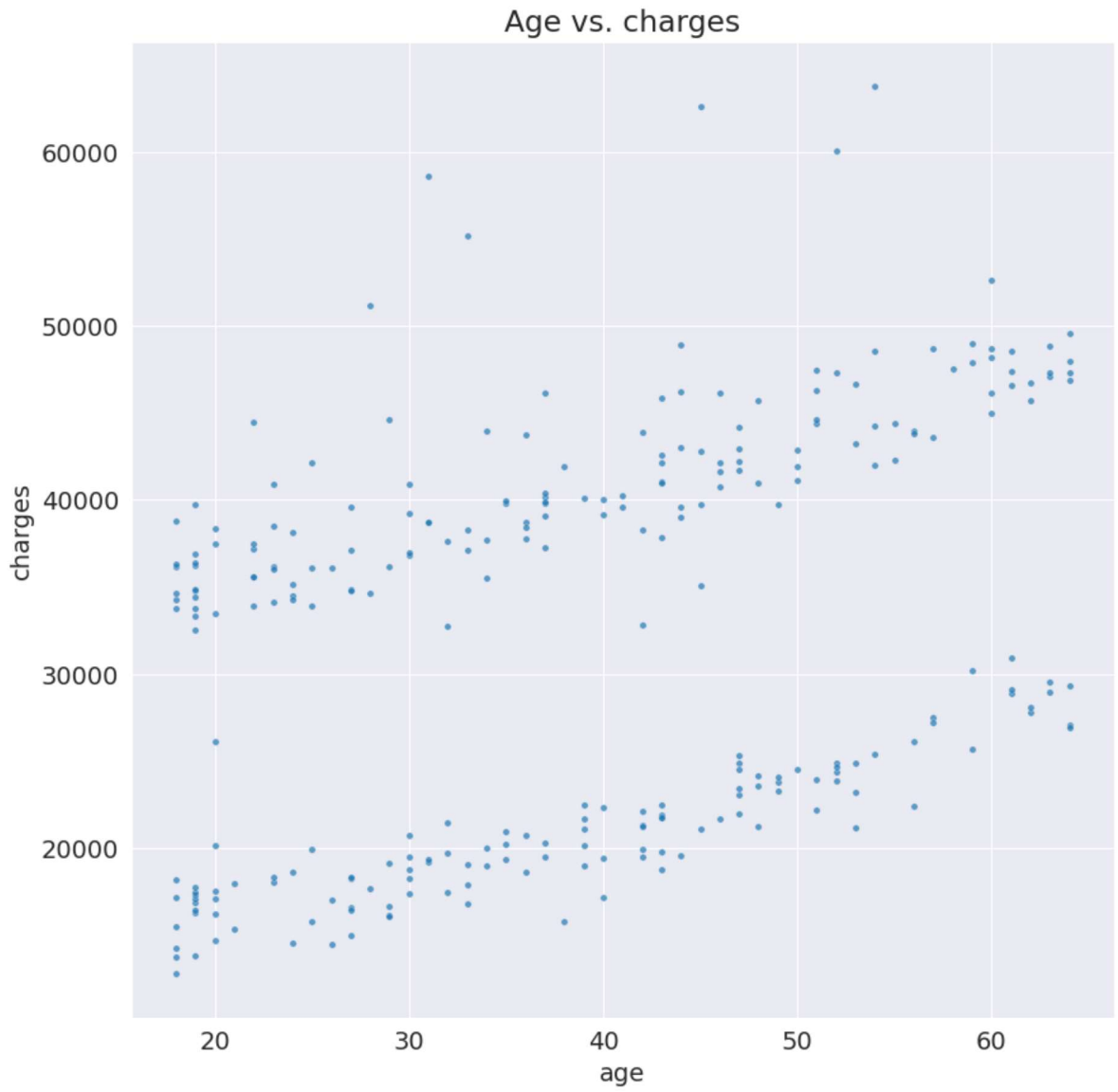
```
non_smoker_df=medical_df[medical_df.smoker=="no"]
```

```
plt.title("Age vs. charges")
sns.scatterplot(data=non_smoker_df, x='age', y='charges', alpha=0.7, s=15);
```




```
smoker_df=medical_df[medical_df.smoker=="yes"]
```

```
plt.title("Age vs. charges")
sns.scatterplot(data=smoker_df, x='age', y='charges', alpha=0.7, s=15);
```




```
def estimate_charges(age,w,b):
    return w*age+b


w=50
b=100


estimate_charges(30,w,b)
```

```
1600
```

```
ages=non_smoker_df.age
ages
```

```
1       18
2       28
3       33
4       32
5       31
        ..
1332    52
1333    50
1334    18
1335    18
1336    21
Name: age, Length: 1064, dtype: int64
```

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

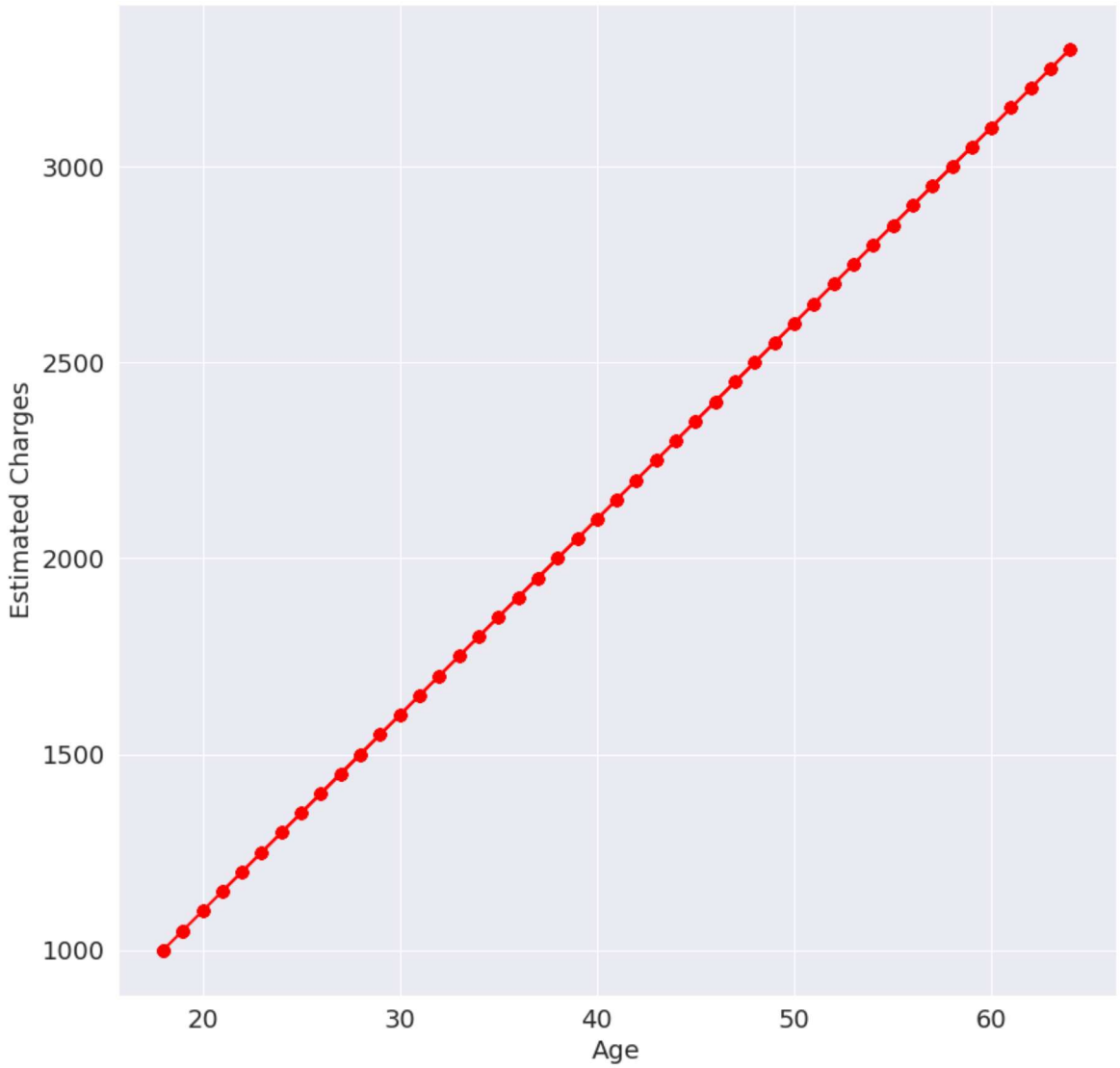
```
1       1000
2       1500
3       1750
4       1700
5       1650
        ...
1332    2700
1333    2600
1334    1000
1335    1000
1336    1150
Name: age, Length: 1064, dtype: int64
```

```
non_smoker_df.charges
```

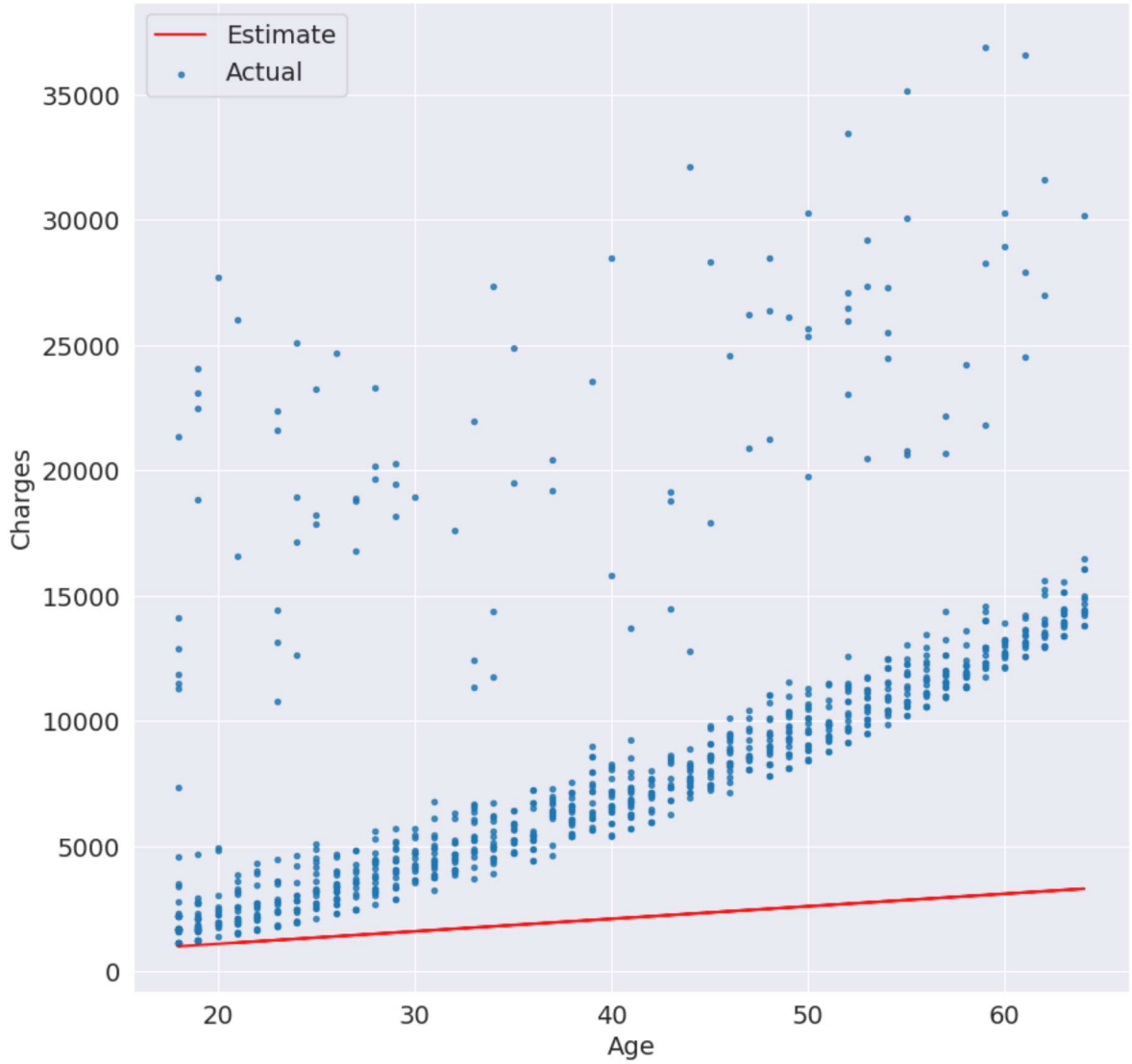
```
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
5        3756.62160
           ...
1332    11411.68500
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
Name: charges, Length: 1064, dtype: float64
```

```
ages = non_smoker_df.age
estimated_charges = estimate_charges(ages, w, b)
```

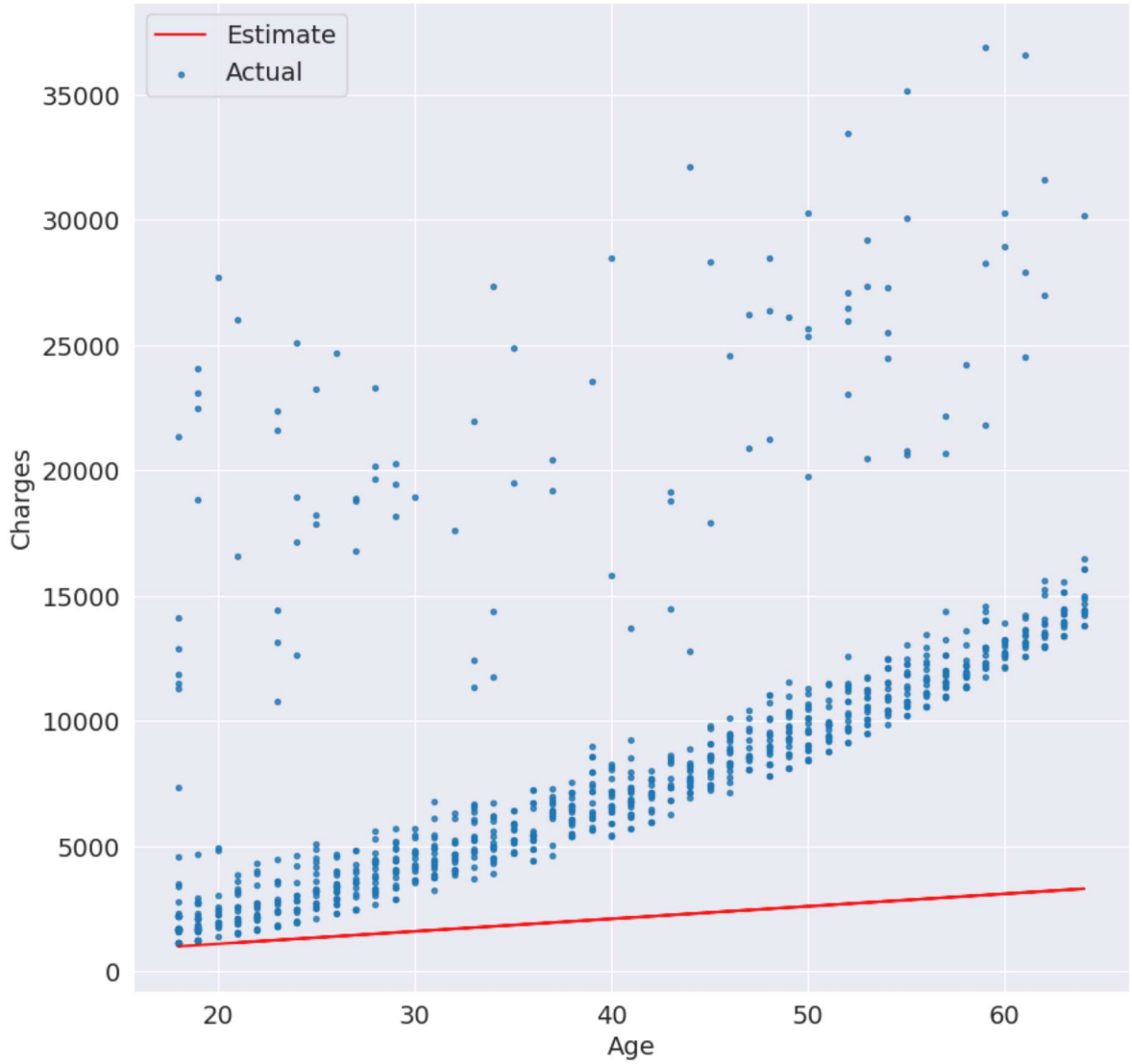
```
plt.plot(ages, estimated_charges, 'r-o');
plt.xlabel('Age');
plt.ylabel('Estimated Charges');
```




```
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']);
```




```
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']);
#w=int(input("enter w: "))
#b=int(input("enter b: "))
try_parameters(w, b)
```




```
targets=non_smoker_df.charges
targets
```

```
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
5        3756.62160
           ...
1332    11411.68500
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
Name: charges, Length: 1064, dtype: float64
```

```
predictions=estimated_charges
predictions
```

```
1       1000
2       1500
3       1750
4       1700
5       1650
        ...
1332    2700
1333    2600
1334    1000
1335    1000
1336    1150
Name: age, Length: 1064, dtype: int64
```

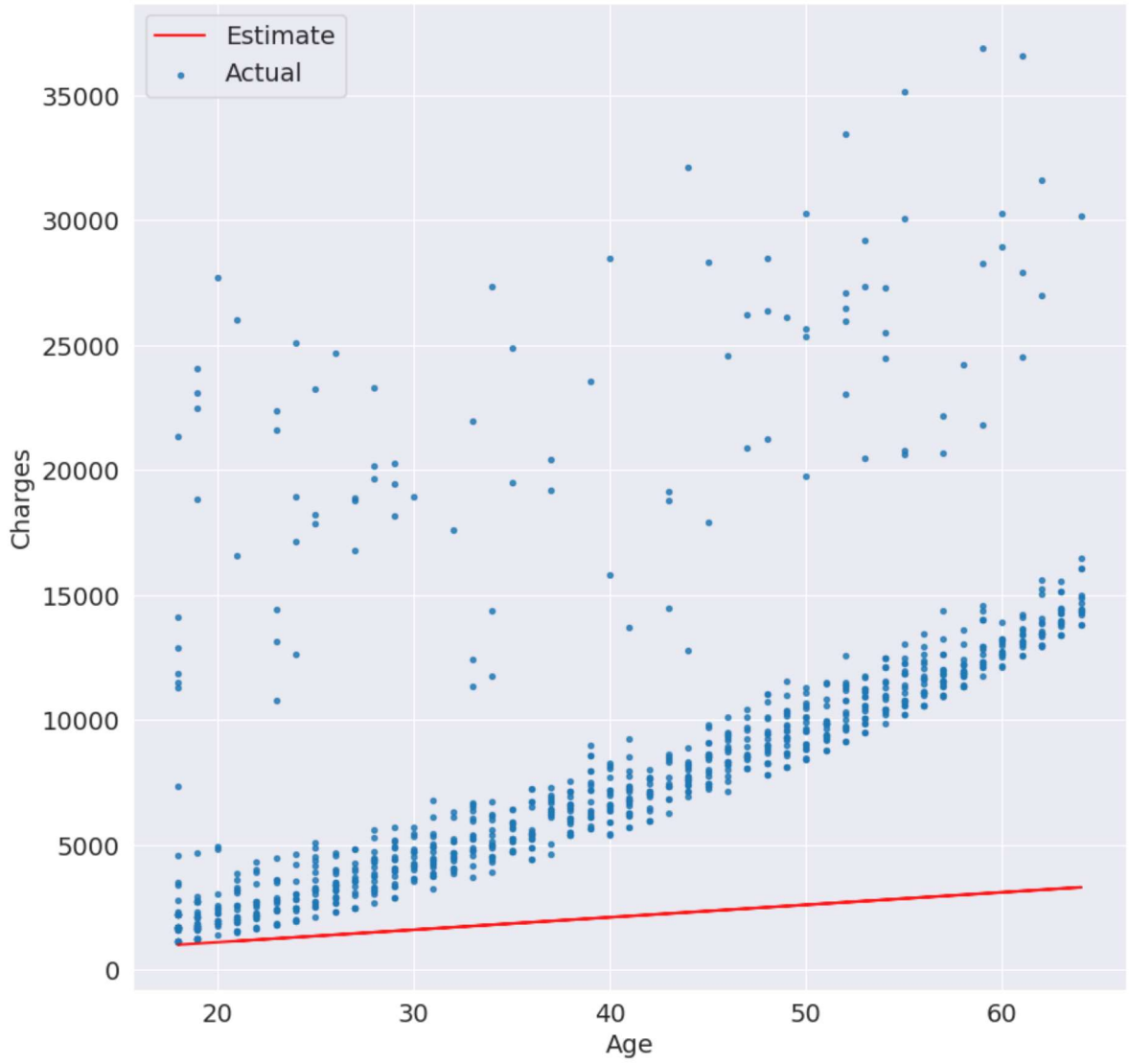
```
!pip install numpy --quiet
```

```
import numpy as np
```

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

```
w = 50
b = 100
```

```
try_parameters(w,b)
```




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

```
rmse(targets, predicted)
```

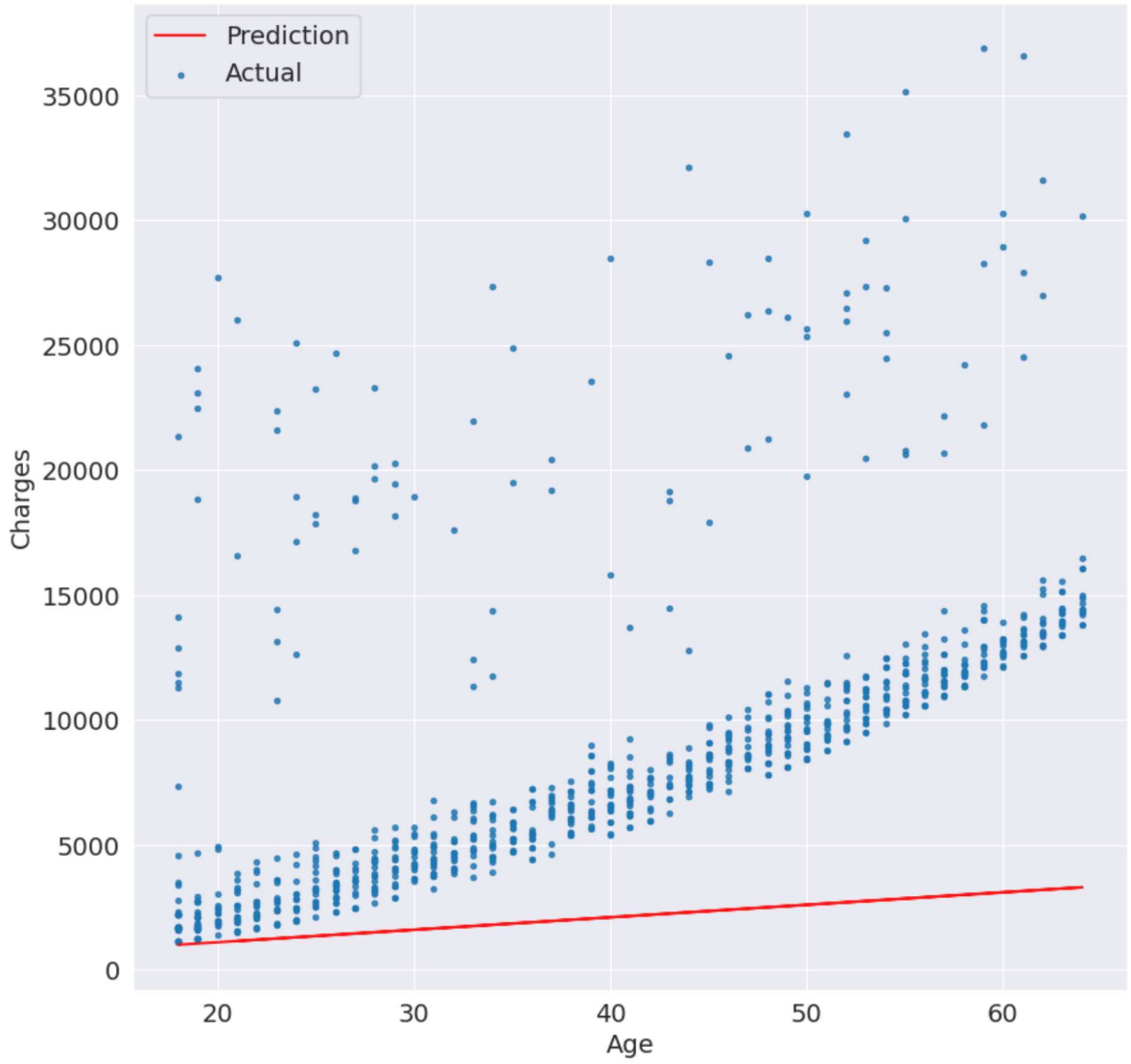
```
8461.949562575493
```

```
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)
```

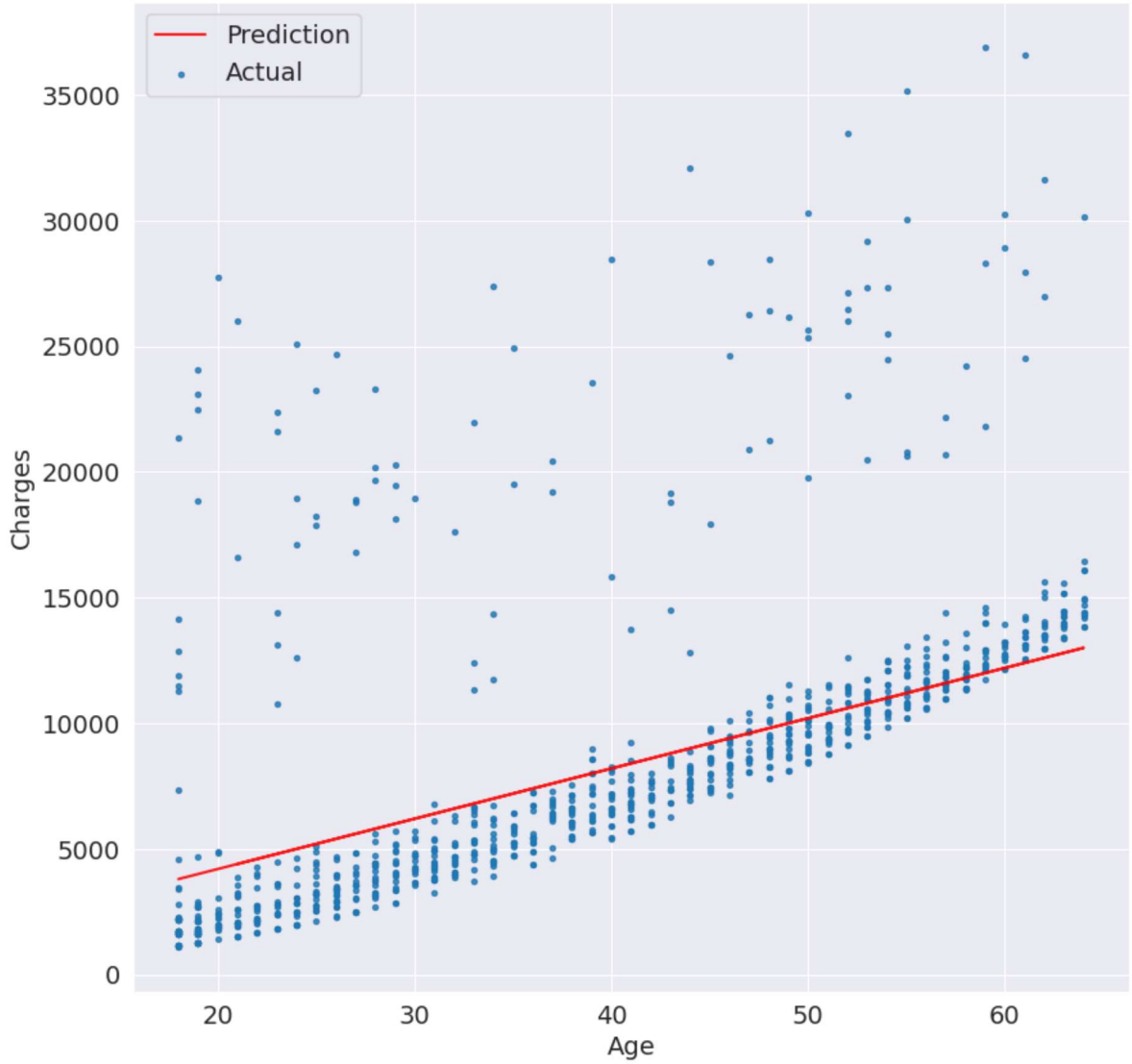
```
try_parameters(50, 100)
```

```
RMSE Loss:  8461.949562575493
```




```
try_parameters(200, 200)
```

```
RMSE Loss:  4771.026760433572
```




## linear regression with scikit-learn

```
!pip install scikit-learn --quiet
```

```
from sklearn.linear_model import LinearRegression
```

```
model = LinearRegression()
```

```
help(model.fit)
```

```
Help on method fit in module sklearn.linear_model._base:

fit(X, y, sample_weight=None) method of sklearn.linear_model._base.LinearRegression instance
    Fit linear model.

    Parameters
    ----------
    X : {array-like, sparse matrix} of shape (n_samples, n_features)
        Training data.

    y : array-like of shape (n_samples,) or (n_samples, n_targets)
        Target values. Will be cast to X's dtype if necessary.

    sample_weight : array-like of shape (n_samples,), default=None
        Individual weights for each sample.

        .. versionadded:: 0.17
```

```
            parameter *sample_weight* support to LinearRegression.

        Returns
        -------
        self : object
            Fitted Estimator.
```

```
inputs = non_smoker_df[['age']]
targets = non_smoker_df.charges
print('inputs.shape :', inputs.shape)
print('targes.shape :', targets.shape)
```

```
inputs.shape : (1064, 1)
targes.shape : (1064,)
```

```
model.fit(inputs, targets)
```

```
LinearRegression()
```

```
model.predict(np.array([[23],
                        [37],
                        [61]]))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:

X does not have valid feature names, but LinearRegression was fitted with feature names

array([ 4055.30443855,  7796.78921819, 14210.76312614])
```

```
predictions = model.predict(inputs)
```

```
predictions
```

```
array([2719.0598744 , 5391.54900271, 6727.79356686, ..., 2719.0598744 ,
       2719.0598744 , 3520.80661289])
```

```
inputs
```

| | age |
|---|---|
| **1** | 18 |
| **2** | 28 |
| **3** | 33 |
| **4** | 32 |
| **5** | 31 |
| **...** | ... |
| **1332** | 52 |
| **1333** | 50 |
| **1334** | 18 |
| **1335** | 18 |
| **1336** | 21 |

1064 rows × 1 columns

```
targets
```

```
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
5        3756.62160
           ...
1332    11411.68500
1333    10600.54830
```

```
1334     2205.98080
1335     1629.83350
1336     2007.94500
Name: charges, Length: 1064, dtype: float64
```

```
rmse(targets, predictions)
```

```
4662.505766636395
```

```
# w
model.coef_
```

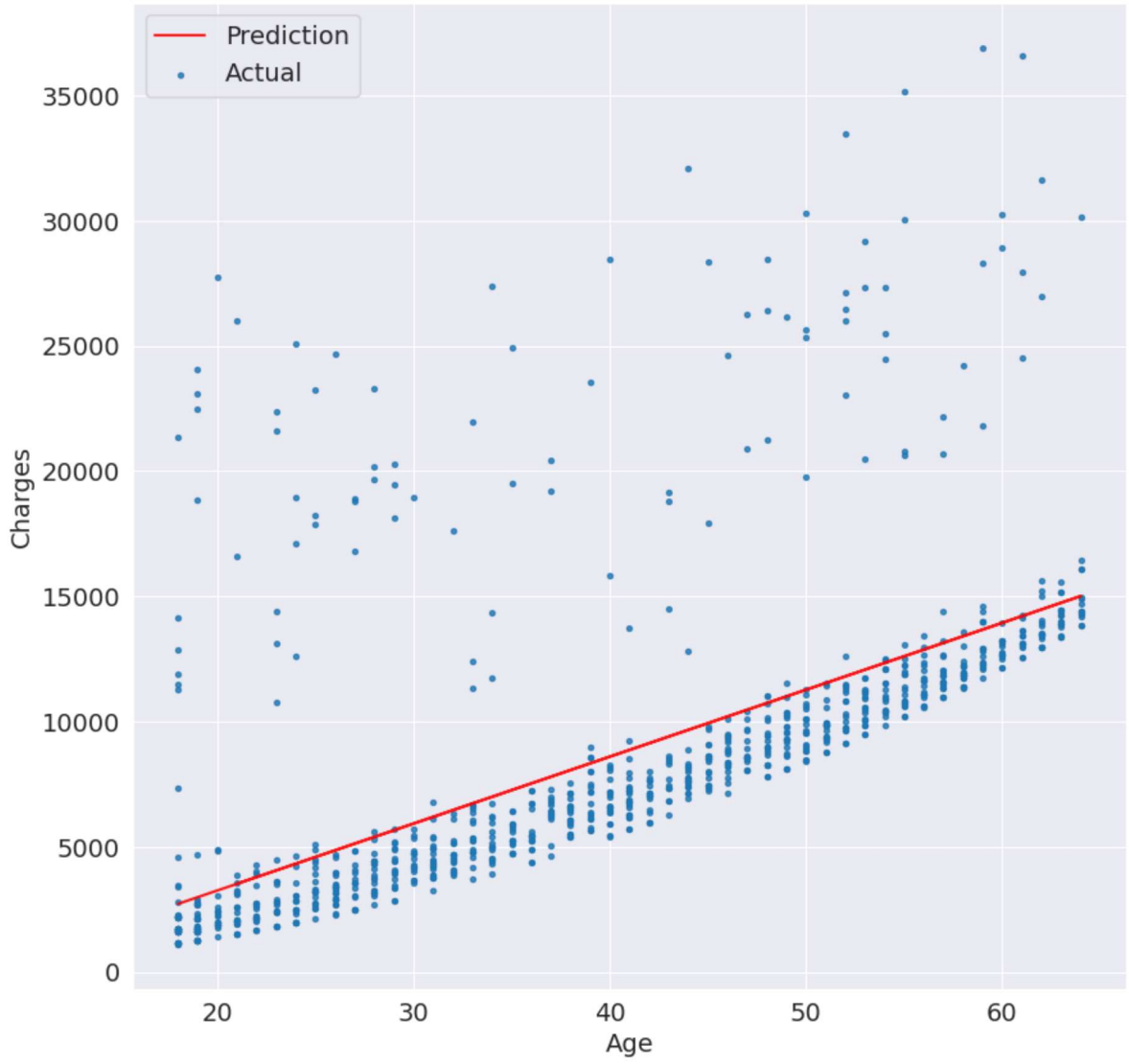
```
array([267.24891283])
```

```
# b
model.intercept_
```

```
-2091.4205565650864
```

```
try_parameters(model.coef_, model.intercept_)
```

```
RMSE Loss:  4662.505766636395
```




## linear regression with multiple features

```
inputs,targets=non_smoker_df[["age","bmi"]],non_smoker_df["charges"]
model=LinearRegression().fit(inputs,targets)
predictions=model.predict(inputs)
```

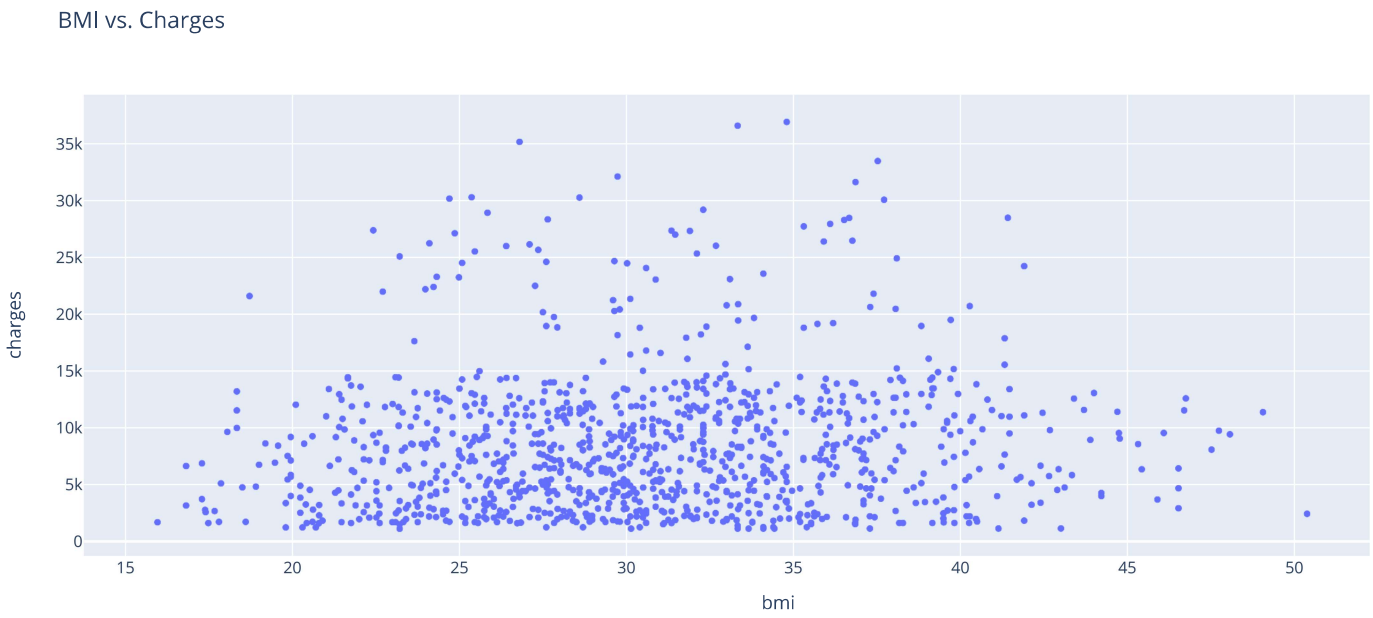
```
loss=rmse(targets,predictions)
print("Loss:",loss)
```

```
Loss: 4662.3128354612945
```

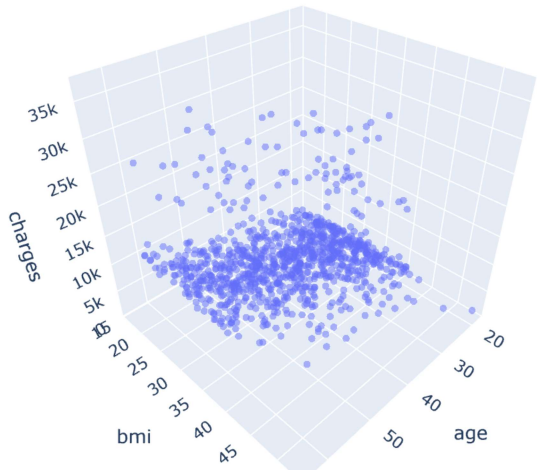
```
non_smoker_df.charges.corr(non_smoker_df.bmi)
```

```
0.0840365431283327
```

```
fig = px.scatter(non_smoker_df, x='bmi', y='charges', title='BMI vs. Charges')
fig.update_traces(marker_size=5)
fig.show()
```




```
fig = px.scatter_3d(non_smoker_df, x='age', y='bmi', z='charges')
fig.update_traces(marker_size=3, marker_opacity=0.5)
fig.show()
```



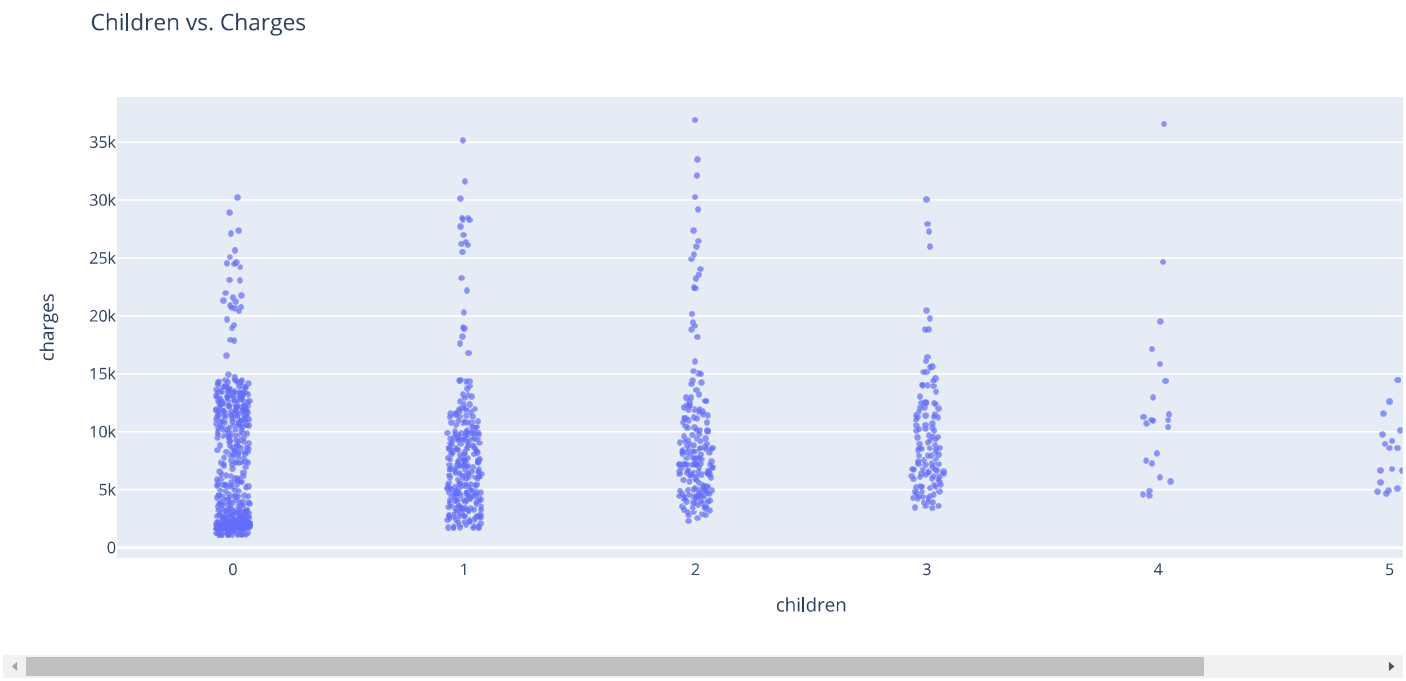
```
model.coef_, model.intercept_
```

```
(array([266.87657817,   7.07547666]), -2293.6320906488727)
```

```
non_smoker_df.charges.corr(non_smoker_df.children)
```

```
0.13892870453542192
```

```
fig = px.strip(non_smoker_df, x='children', y='charges', title= "Children vs. Charges")
fig.update_traces(marker_size=4, marker_opacity=0.7)
fig.show()
```



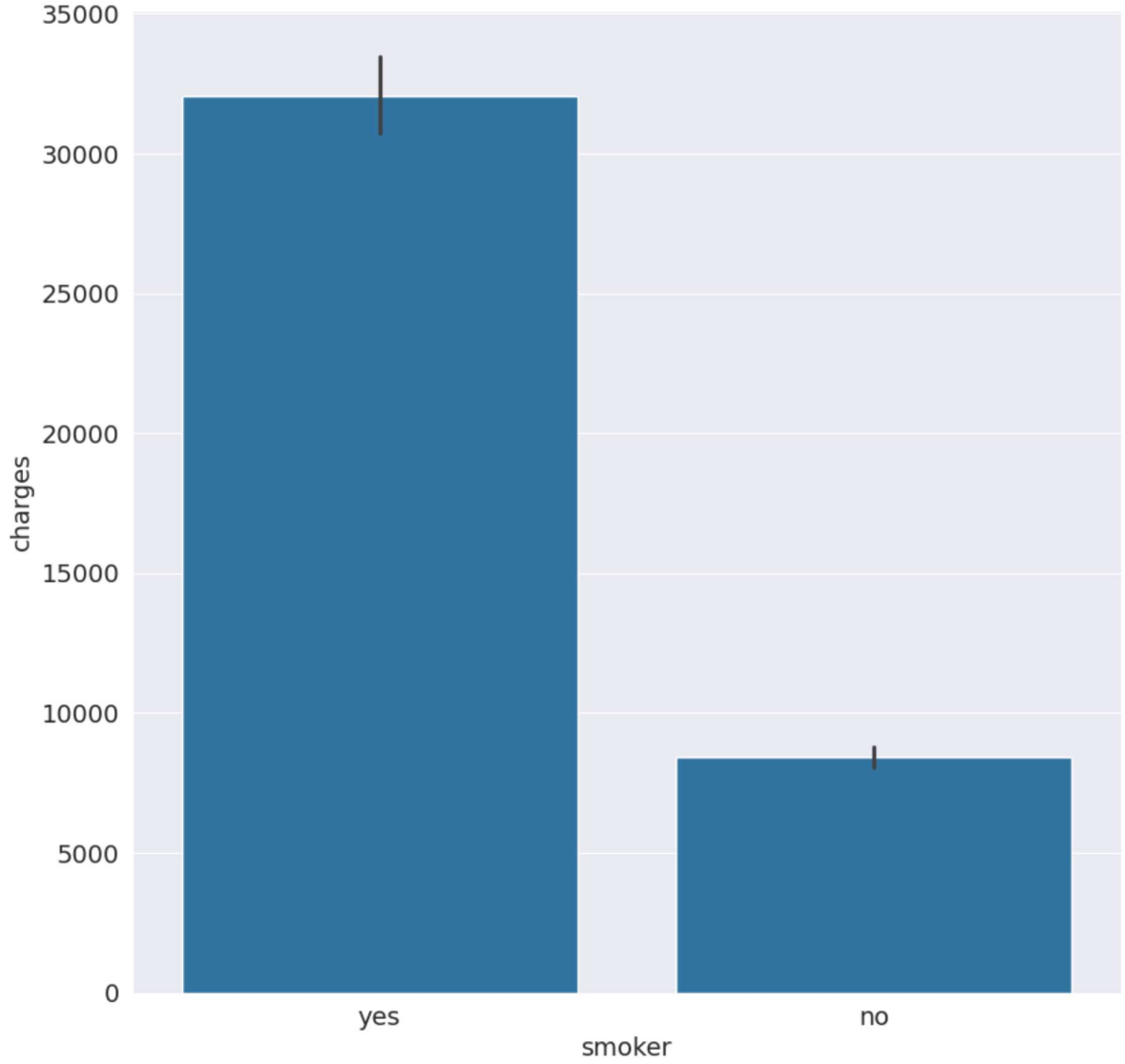

```
inputs,targets=non_smoker_df[["age","bmi","children"]],non_smoker_df["charges"]
model=LinearRegression().fit(inputs,targets)
predictions=model.predict(inputs)
loss=rmse(targets,predictions)
print("Loss:",loss)
```

```
Loss: 4608.470405038247
```

```
inputs,targets=medical_df[["age","bmi","children",]],medical_df["charges"]
model=LinearRegression().fit(inputs,targets)
predictions=model.predict(inputs)
loss=rmse(targets,predictions)
print("Loss:",loss)
```

```
Loss: 11355.317901125973
```

```
sns.barplot(data=medical_df, x='smoker', y='charges');
```




```
smoker_codes = {'no': 0, 'yes': 1}
medical_df['smoker_code'] = medical_df.smoker.map(smoker_codes)
```

```
medical_df.charges.corr(medical_df.smoker_code)
```

```
0.787251430498478
```

```
medical_df
```

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

1338 rows × 8 columns

```
inputs,targets=medical_df[["age","bmi","children","smoker_code"]],medical_df["charges"]
model=LinearRegression().fit(inputs,targets)
predictions=model.predict(inputs)
loss=rmse(targets,predictions)
print("Loss:",loss)
```

```
Loss: 6056.439217188081
```

```
sns.barplot(data=medical_df, x='sex', y='charges')
```

```
<Axes: xlabel='sex', ylabel='charges'>
```

```
sex_codes = {'female': 0, 'male': 1}
```

```
medical_df['sex_code'] = medical_df.sex.map(sex_codes)
```

```
medical_df.charges.corr(medical_df.sex_code)
```

```
0.057292062202025484
```

```
inputs,targets=medical_df[["age","bmi","children","smoker_code",]],medical_df["charges"]
model=LinearRegression().fit(inputs,targets)
predictions=model.predict(inputs)
loss=rmse(targets,predictions)
print("Loss:",loss)
```

```
Loss: 6056.439217188081
```

```
sns.barplot(data=medical_df, x='region', y='charges');
```