In [1]:
```python
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np

data = pd.read_csv("water_potability.csv")
data.head()
```

Out[1]:

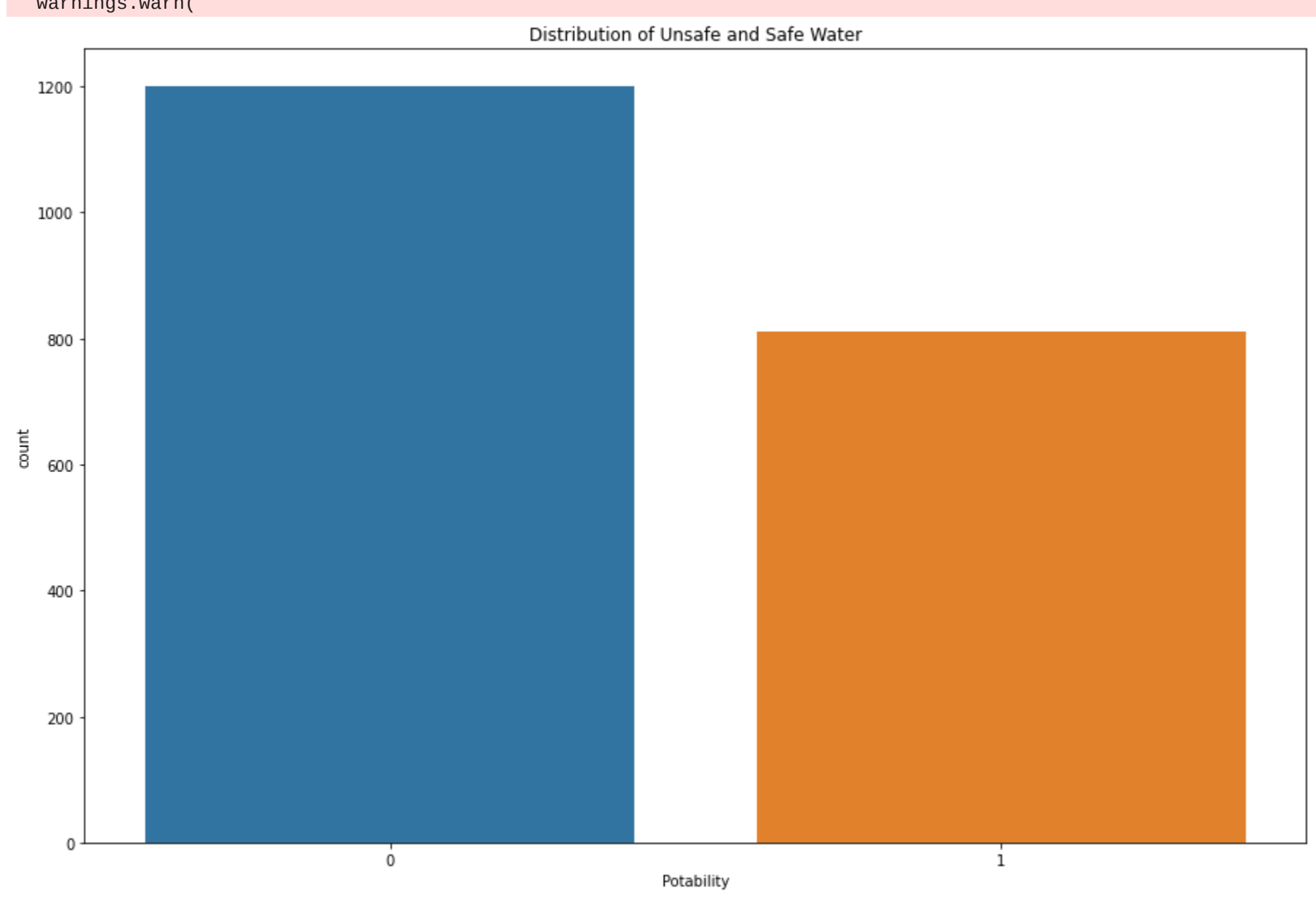
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

In [2]:
```python
data = data.dropna()
data.isnull().sum()
```

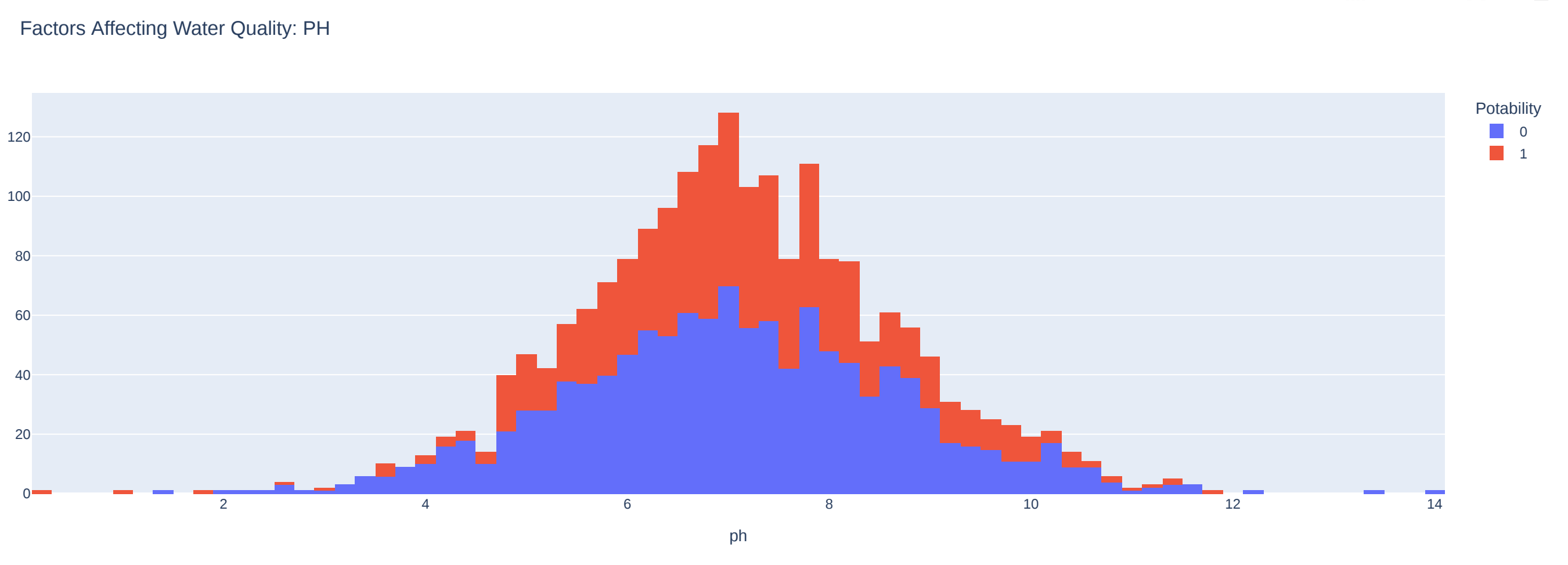
Out[2]:
```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

In [3]:
```python
plt.figure(figsize=(15, 10))
sns.countplot(data.Potability)
plt.title("Distribution of Unsafe and Safe Water")
plt.show()
```

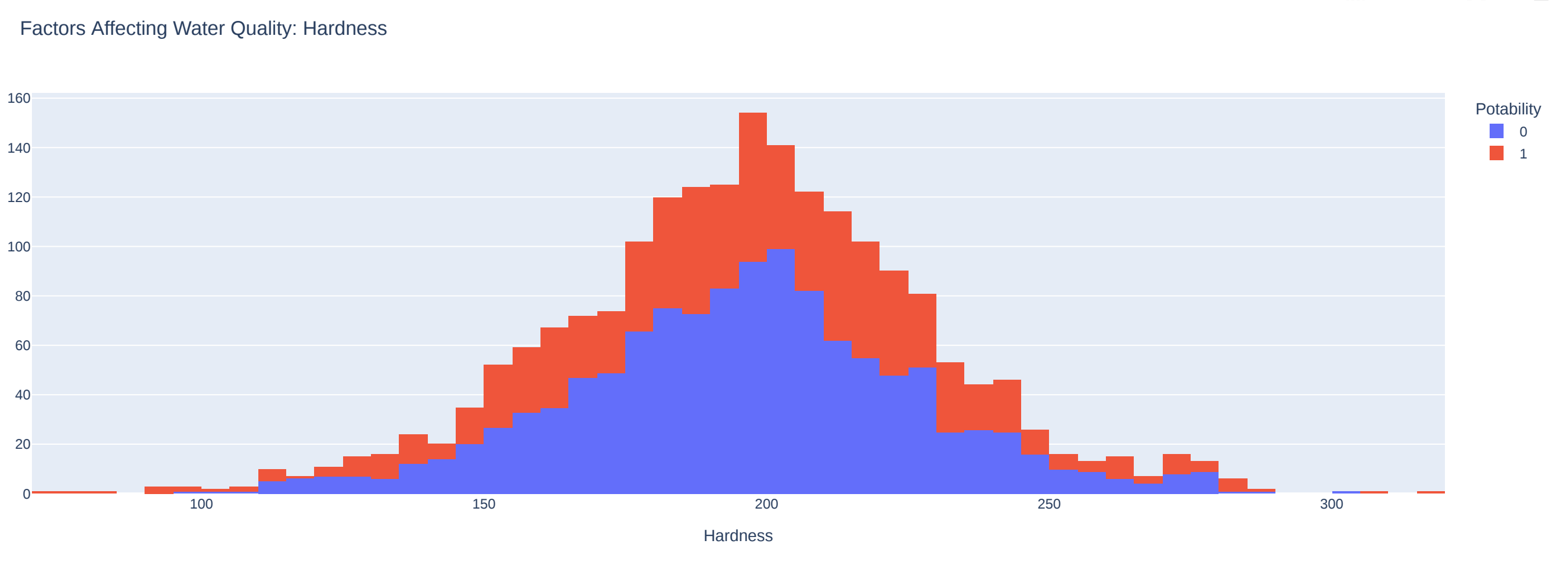
```
c:\users\vijay raj\appdata\local\programs\python\python38\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



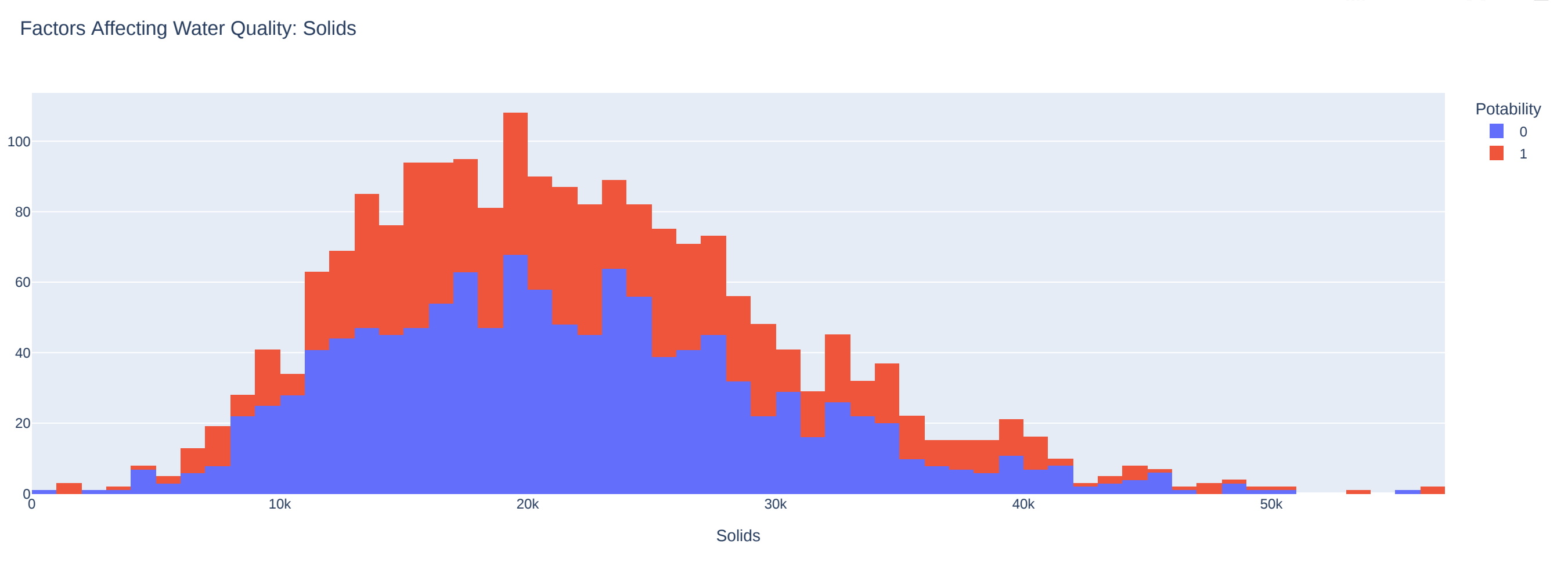
In [4]:
```python
import plotly.express as px
data = data
figure = px.histogram(data, x = "ph",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: PH")
figure.show()
```



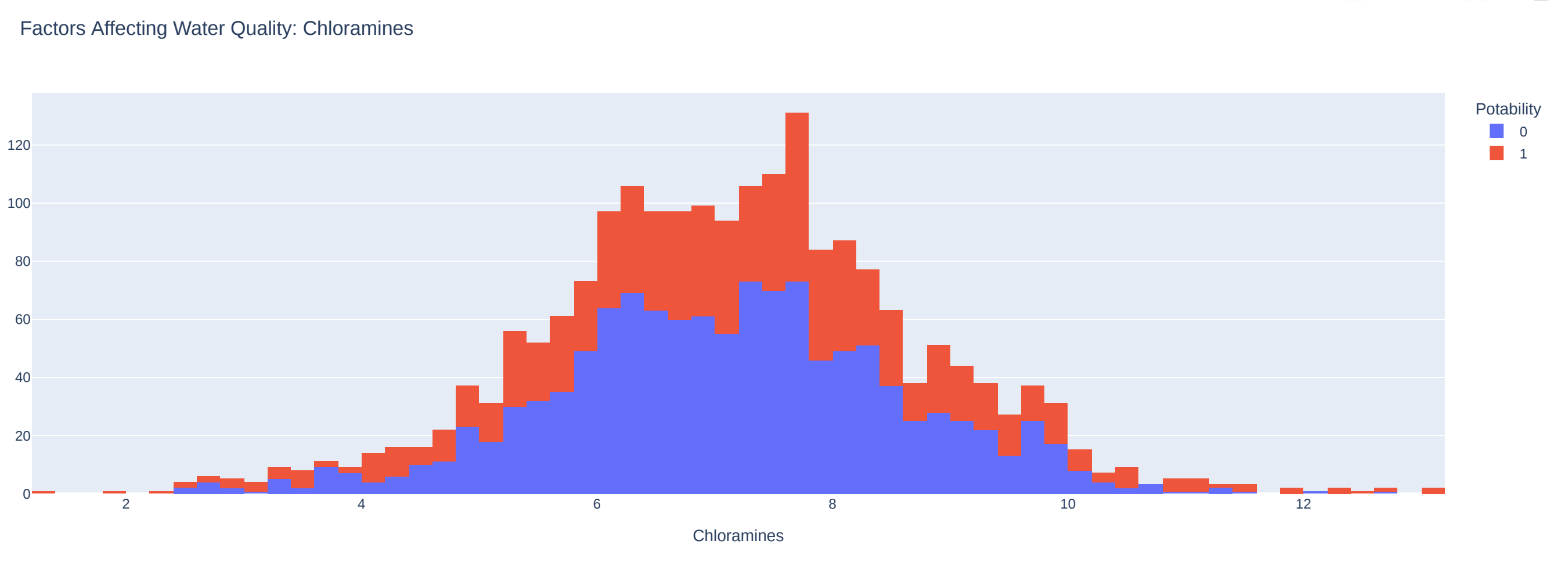
In [5]:
```python
figure = px.histogram(data, x = "Hardness",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Hardness")
figure.show()
```



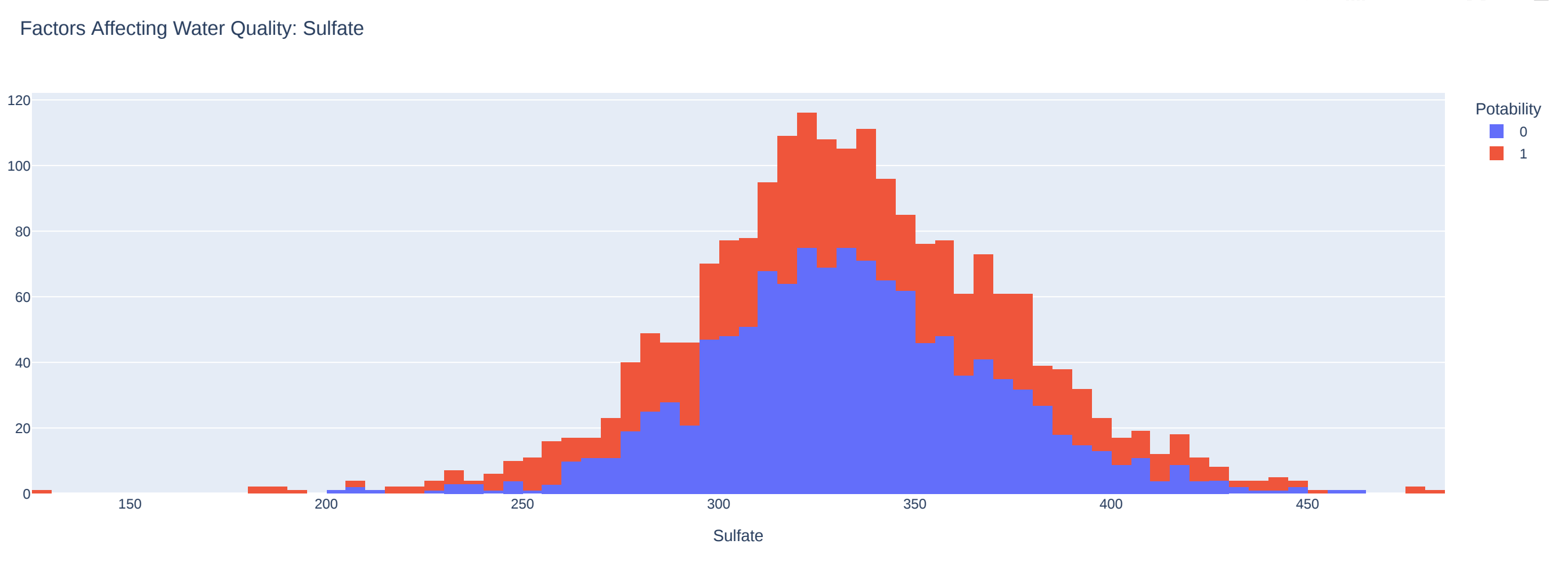
In [6]:
```python
figure = px.histogram(data, x = "Solids",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Solids")
figure.show()
```



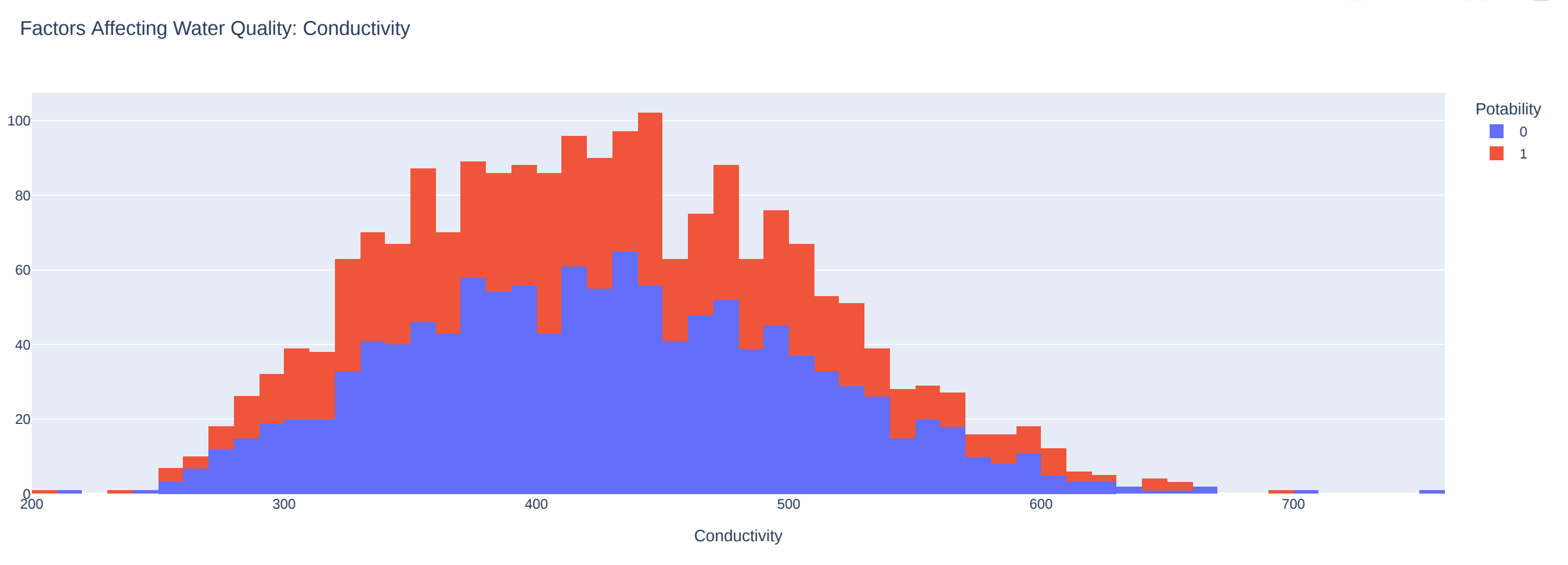
In [7]:
```python
figure = px.histogram(data, x = "Chloramines",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Chloramines")
figure.show()
```



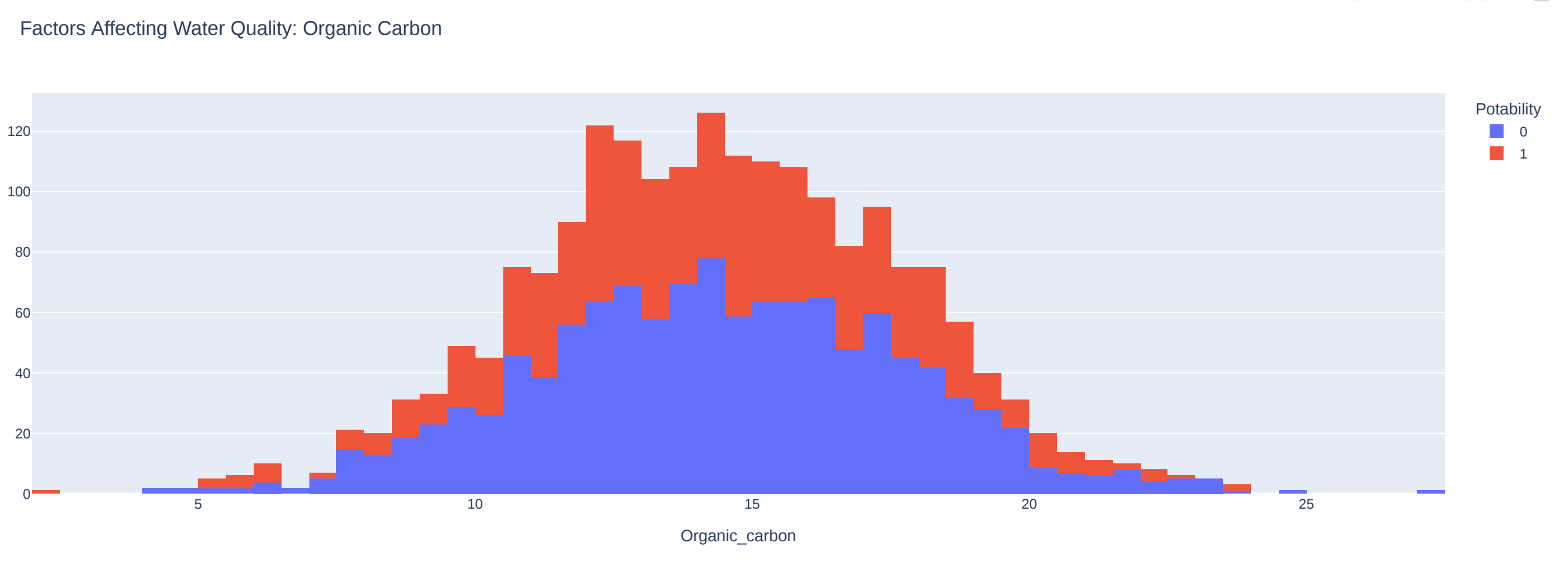
In [8]:
```python
figure = px.histogram(data, x = "Sulfate",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Sulfate")
figure.show()
```



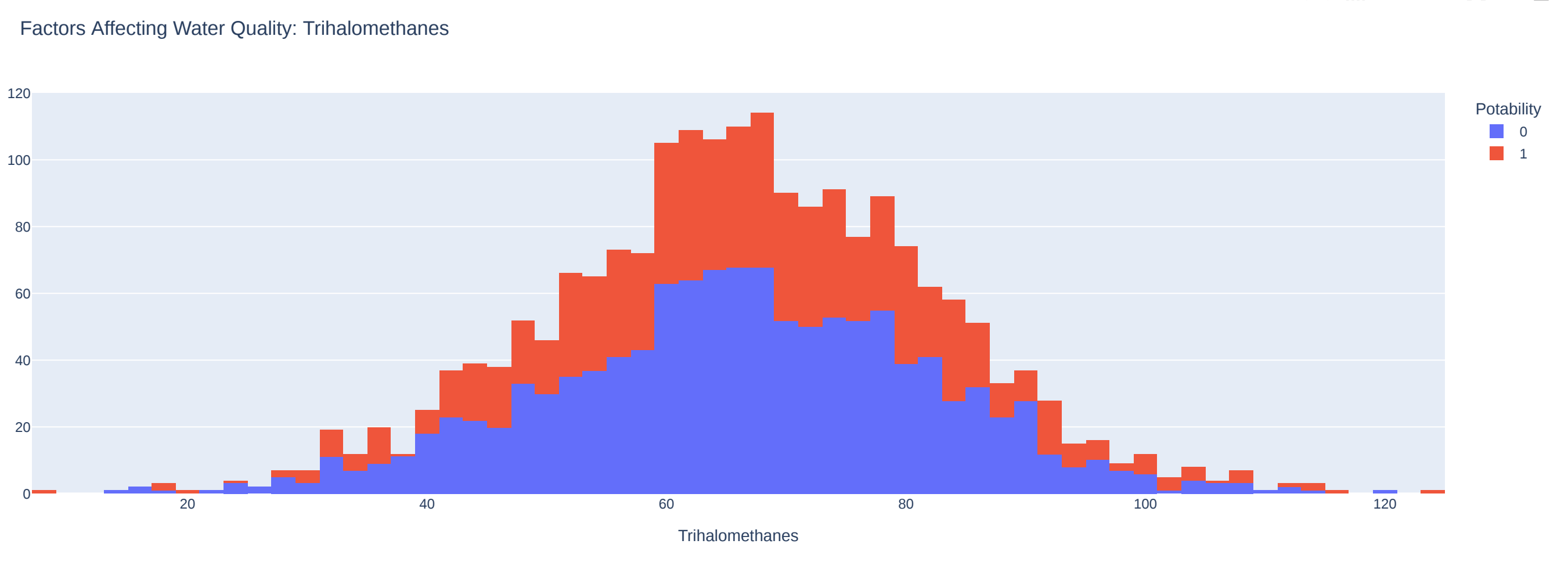
In [9]:
```python
figure = px.histogram(data, x = "Conductivity",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Conductivity")
figure.show()
```



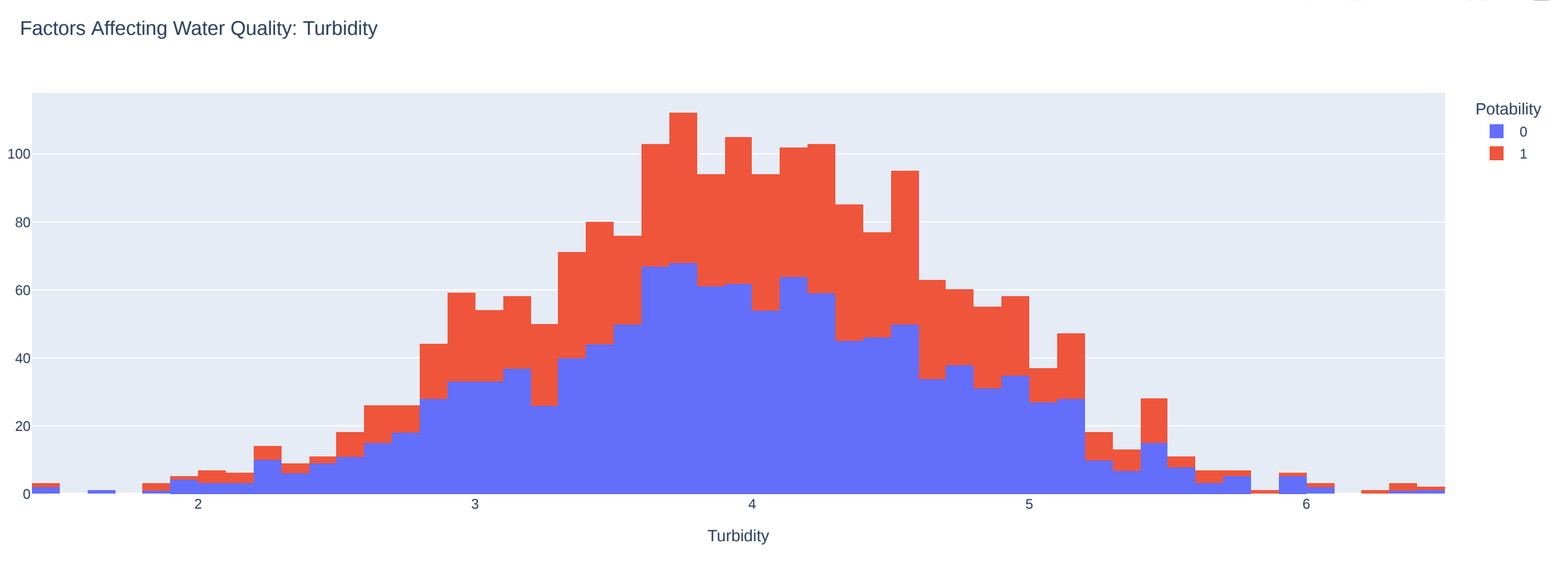
In [10]:
```python
figure = px.histogram(data, x = "Organic_carbon",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Organic Carbon")
figure.show()
```



In [11]:
```python
figure = px.histogram(data, x = "Trihalomethanes",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Trihalomethanes")
figure.show()
```



In [12]:
```python
figure = px.histogram(data, x = "Turbidity",
                      color = "Potability",
                      title= "Factors Affecting Water Quality: Turbidity")
figure.show()
```



In [13]:
```python
correlation = data.corr()
correlation["ph"].sort_values(ascending=False)
```

Out[13]:
```
ph                 1.000000
Hardness           0.108948
Organic_carbon     0.028375
Trihalomethanes    0.018278
Potability         0.014530
Conductivity       0.014128
Sulfate            0.010524
Chloramines       -0.024768
Turbidity         -0.035849
Solids            -0.087615
Name: ph, dtype: float64
```

In [14]:
```python
from pycaret.classification import *
clf = setup(data, target = "Potability", silent = True, session_id = 786)
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.6830	0.7005	0.4197	0.6744	0.5133	0.2976	0.3182	0.1330
qda	Quadratic Discriminant Analysis	0.6823	0.7192	0.3985	0.6883	0.5013	0.2917	0.3174	0.0050
et	Extra Trees Classifier	0.6816	0.6941	0.3861	0.6858	0.4916	0.2863	0.3123	0.0990
lightgbm	Light Gradient Boosting Machine	0.6652	0.6916	0.4762	0.6078	0.5324	0.2781	0.2840	0.0390
gbc	Gradient Boosting Classifier	0.6610	0.6738	0.3718	0.6323	0.4672	0.2432	0.2619	0.0980
nb	Naive Bayes	0.6184	0.6078	0.2478	0.5545	0.3412	0.1261	0.1462	0.0050
dt	Decision Tree Classifier	0.6034	0.5895	0.5186	0.5049	0.5097	0.1775	0.1784	0.0090
lr	Logistic Regression	0.5984	0.5198	0.0071	0.1900	0.0134	0.0028	0.0127	0.7530
ridge	Ridge Classifier	0.5984	0.0000	0.0089	0.1583	0.0168	0.0035	0.0056	0.0060
dummy	Dummy Classifier	0.5984	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0080
lda	Linear Discriminant Analysis	0.5977	0.4903	0.0089	0.1500	0.0167	0.0021	0.0024	0.0110
ada	Ada Boost Classifier	0.5956	0.5671	0.2919	0.4896	0.3644	0.0972	0.1034	0.0440
knn	K Neighbors Classifier	0.5743	0.5423	0.3644	0.4642	0.4070	0.0826	0.0846	0.0150
svm	SVM - Linear Kernel	0.5194	0.0000	0.3982	0.1604	0.2287	-0.0014	-0.0104	0.0070

Out[14]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=-1, oob_score=False, random_state=786, verbose=0,
                       warm_start=False)
```

In [16]:
```python
model = create_model("rf")
predict = predict_model(model, data=data)
predict.head()
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7376	0.7545	0.4737	0.7941	0.5934	0.4174	0.4478
1	0.6525	0.6687	0.3860	0.6111	0.4731	0.2331	0.2468
2	0.6879	0.7056	0.4386	0.6757	0.5319	0.3134	0.3299
3	0.6738	0.7172	0.3684	0.6774	0.4773	0.2691	0.2955
4	0.6667	0.6885	0.3158	0.6923	0.4337	0.2417	0.2791
5	0.6312	0.6404	0.3929	0.5500	0.4583	0.1904	0.1966
6	0.7092	0.7192	0.5357	0.6667	0.5941	0.3717	0.3771
7	0.6786	0.6988	0.5000	0.6222	0.5545	0.3077	0.3122
8	0.7071	0.7090	0.3750	0.7778	0.5060	0.3322	0.3769
9	0.6857	0.7033	0.4107	0.6765	0.5111	0.2994	0.3196
Mean	0.6830	0.7005	0.4197	0.6744	0.5133	0.2976	0.3182
Std	0.0288	0.0290	0.0637	0.0690	0.0523	0.0640	0.0677

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Random Forest Classifier	0.9040	0.9691	0.8237	0.9304	0.8738	0.7968	0.8007

Out[16]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability	Label	Score
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0	0	0.87
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0	0	0.91
5	5.584087	188.313324	28748.687739	7.544869	326.678363	280.467916	8.399735	54.917862	2.559708	0	0	0.83
6	10.223862	248.071735	28749.716544	7.513408	393.663396	283.651634	13.789695	84.603556	2.672989	0	0	0.89
7	8.635849	203.361523	13672.091764	4.563009	303.309771	474.607645	12.363817	62.798309	4.401425	0	0	0.94

In [ ]:

In [ ]: