```
!ls
```

```
insurance.csv   insurance.zip   sample_data
```

```
import pandas as pd

df = pd.read_csv("insurance.csv")
df.head()
```

	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

Next steps: (Generate code with df) (New interactive sheet)

```
# Encode binary categorical columns
df['sex'] = df['sex'].map({'male': 1, 'female': 0})
df['smoker'] = df['smoker'].map({'yes': 1, 'no': 0})

# One-hot encode region
df = pd.get_dummies(df, columns=['region'], drop_first=True)

df.head()
```

	age	sex	bmi	children	smoker	charges	region_northwest	region_s
0	19	0	27.900	0	1	16884.92400	False	
1	18	1	33.770	1	0	1725.55230	False	
2	28	1	33.000	3	0	4449.46200	False	
3	33	1	22.705	0	0	21984.47061	True	
4	32	1	28.880	0	0	3866.85520	True	

Next steps: (Generate code with df) (New interactive sheet)

```
X = df[['age', 'bmi', 'children', 'smoker']]
y = df['charges']
```

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```python
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

lr = LinearRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)

print("Linear Regression Results")
print("MSE:", mean_squared_error(y_test, y_pred_lr))
print("R2 Score:", r2_score(y_test, y_pred_lr))
```

```
Linear Regression Results
MSE: 33981653.95019776
R2 Score: 0.7811147722517886
```

```python
from sklearn.linear_model import Ridge

alphas = [0.1, 1, 10, 100]
results = []

for alpha in alphas:
    ridge = Ridge(alpha=alpha)
    ridge.fit(X_train, y_train)

    y_pred = ridge.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    results.append([alpha, mse, r2])
```

```python
results_df = pd.DataFrame(
    results, columns=['Alpha', 'MSE', 'R2 Score']
)

results_df
```

	Alpha	MSE	R2 Score
0	0.1	3.398517e+07	0.781092
1	1.0	3.401934e+07	0.780872
2	10.0	3.458037e+07	0.777258
3	100.0	4.831900e+07	0.688764

Next steps: (Generate code with `results_df`) (New interactive sheet)

```
best_alpha = results_df.loc[results_df['MSE'].idxmin()]
best_alpha
```

	0
Alpha	1.000000e-01
MSE	3.398517e+07
R2 Score	7.810921e-01

dtype: float64