```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score


data = pd.read_csv('luna_pets.csv')


data.head()
```

	X	Y	Z	R	G	B	Intensity
0	731000.31	9246012.06	3317.59	126	119	163	5911.0
1	731002.53	9246010.16	3316.46	118	110	160	6939.0
2	731000.50	9246012.30	3316.79	127	121	159	3855.0
3	731000.25	9246012.73	3317.08	126	120	157	5654.0
4	731001.47	9246010.60	3317.28	121	114	159	4369.0

Next steps: Generate code with `data` View recommended plots

```
X = data.drop(columns=['X','Y'])
y = data['X']


X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.3, random_state=42)


model = LinearRegression()


model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

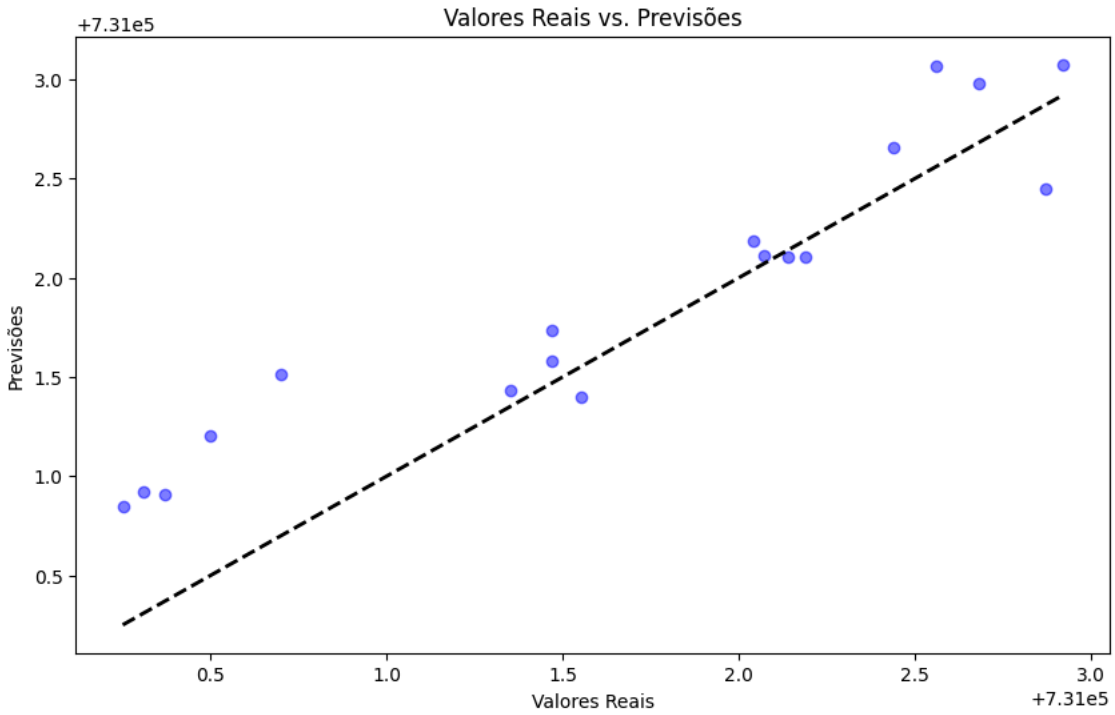
```
predictions = model.predict(X_val)


mae = mean_absolute_error(y_val, predictions)
mse = mean_squared_error(y_val, predictions)
rmse = mean_squared_error(y_val, predictions, squared=False)
r2 = r2_score(y_val, predictions)


print("Erro Médio Absoluto (MAE):", mae)
print("Erro Quadrático Médio (MSE):", mse)
print("Raiz do Erro Quadrático Médio (RMSE):", rmse)
print("R-quadrado (R²):", r2)
```

```
Erro Médio Absoluto (MAE): 0.32036522548231816
Erro Quadrático Médio (MSE): 0.1615418833312438
Raiz do Erro Quadrático Médio (RMSE): 0.4019227330361444
R-quadrado (R²): 0.7960415597031454
```

```
plt.figure(figsize=(10, 6))
plt.scatter(y_val, predictions, color='blue', alpha=0.5)
plt.plot([y_val.min(), y_val.max()], [y_val.min(), y_val.max()], 'k--', lw=2)
plt.xlabel('Valores Reais')
plt.ylabel('Previsões')
plt.title('Valores Reais vs. Previsões')
plt.show()
```


Valores Reais vs. Previsões
+7.31e5
3.0
2.5
2.0
1.5
1.0
0.5
Previsões
0.5
1.0
1.5
2.0
2.5
3.0
+7.31e5
Valores Reais