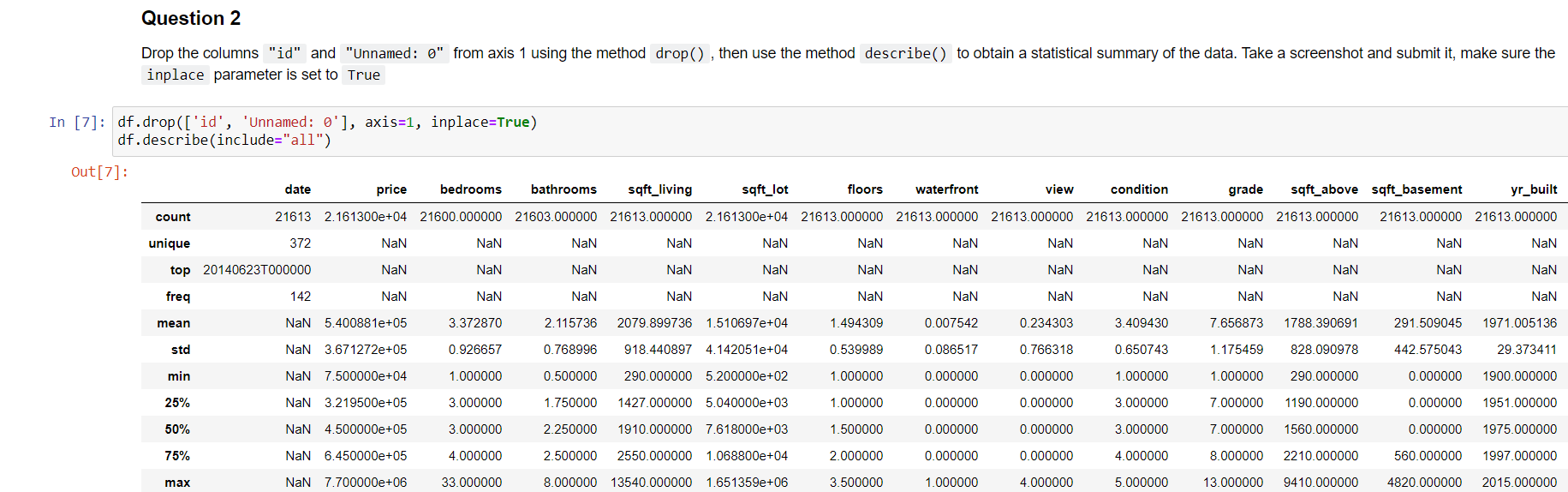
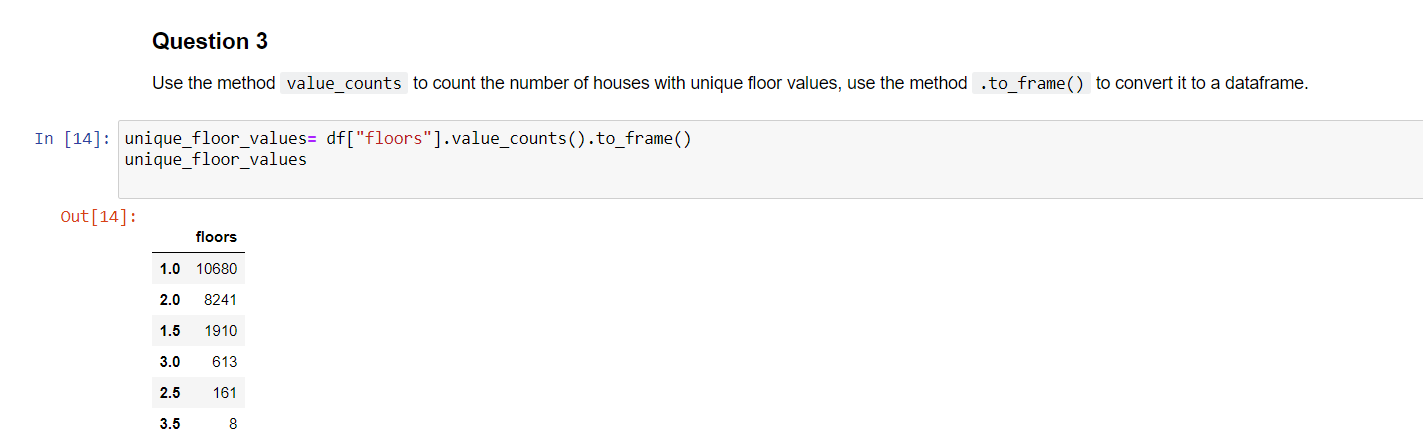


df.dtypes



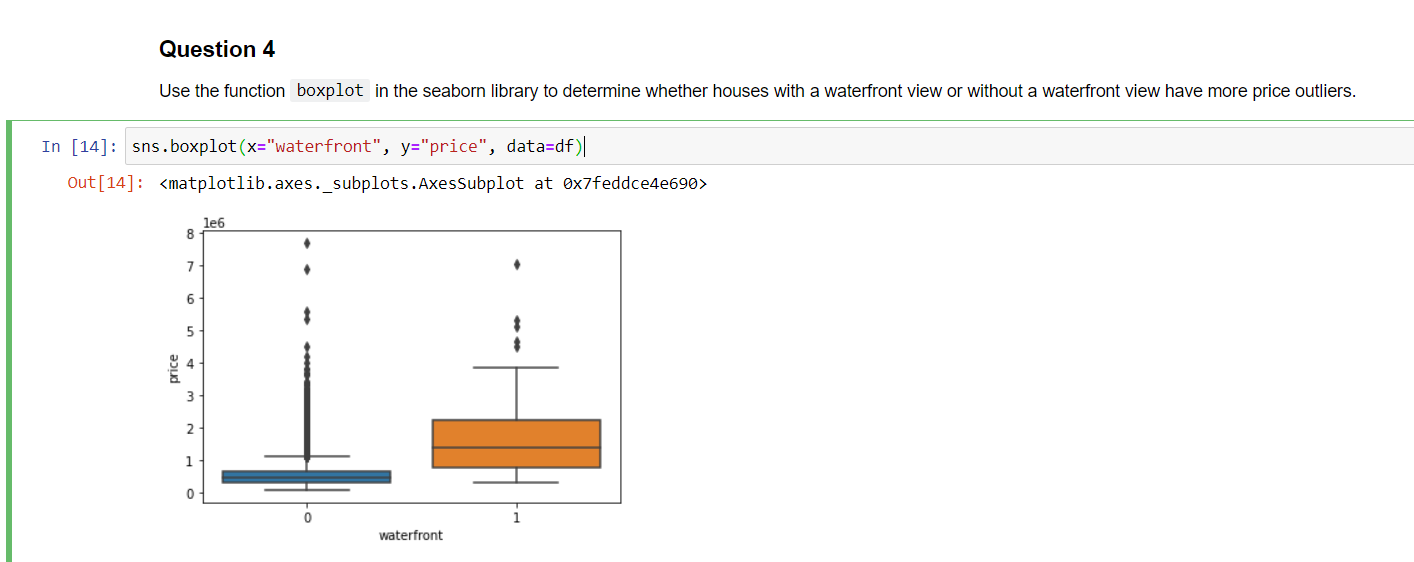
df.drop(['id', 'Unnamed: 0'], axis=1, inplace=True)

df.describe(include="all")

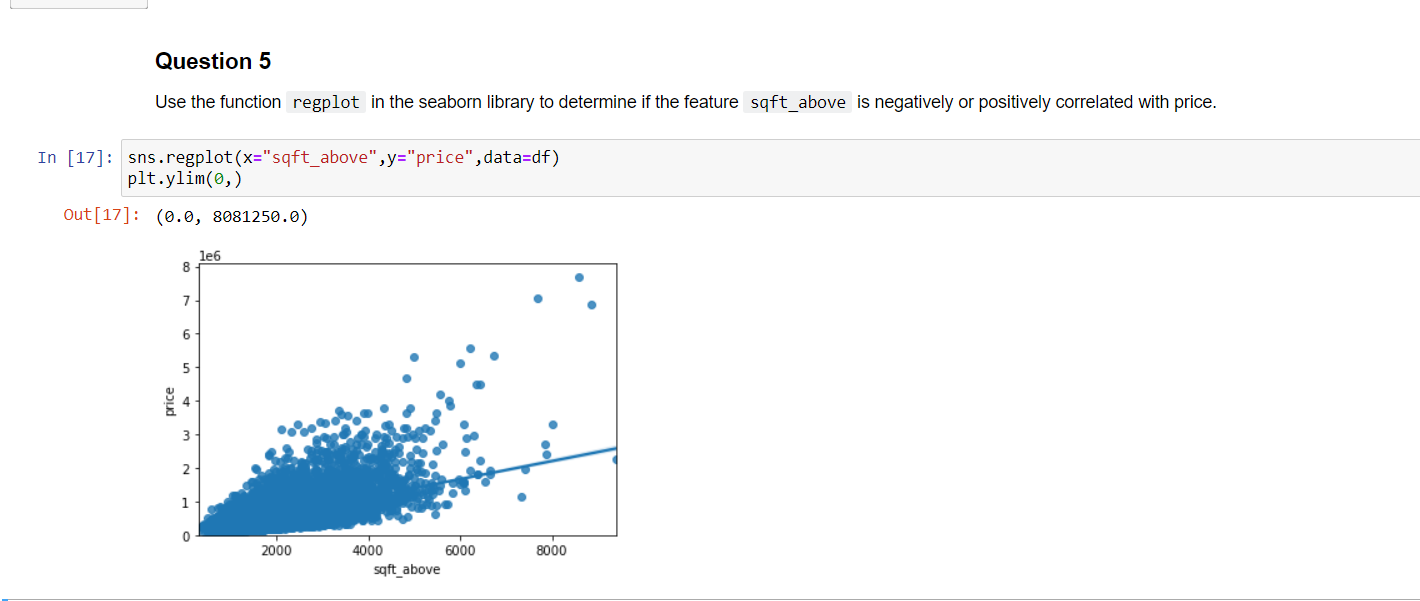


unique\_floor\_values= df["floors"].value\_counts().to\_frame()

unique\_floor\_values

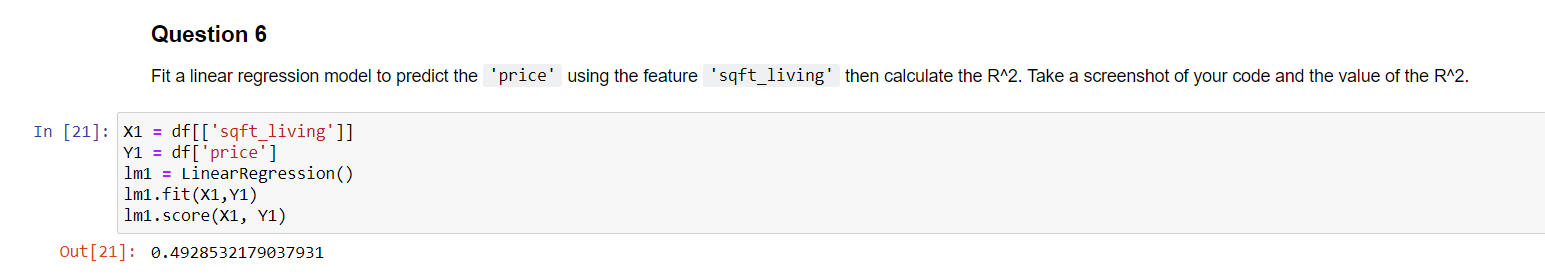


sns.boxplot(x="waterfront", y="price", data=df)



sns.regplot(x="sqft\_above",y="price",data=df)

plt.ylim(0,)



X1 = df[['sqft\_living']]

Y1 = df['price']

lm1 = LinearRegression()

lm1.fit(X1,Y1)

lm1.score(X1, Y1)



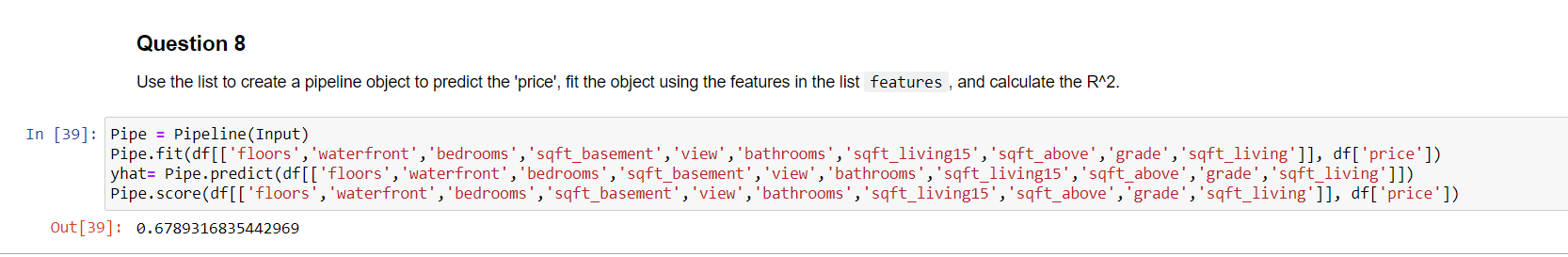
lm2 = LinearRegression()

X3= df[['floors','waterfront','bedrooms','sqft\_basement','view','bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']]

Y3=df['price']

lm2.fit(X3,Y3)

lm2.score(X3,Y3)

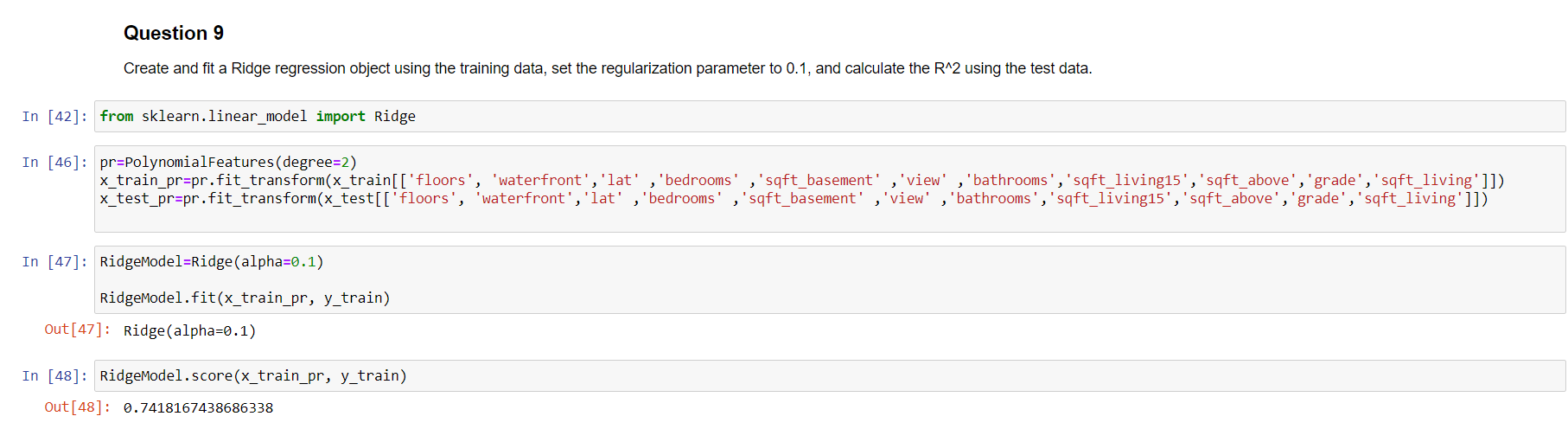


Pipe = Pipeline(Input)

Pipe.fit(df[['floors','waterfront','bedrooms','sqft\_basement','view','bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']], df['price'])

yhat= Pipe.predict(df[['floors','waterfront','bedrooms','sqft\_basement','view','bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

Pipe.score(df[['floors','waterfront','bedrooms','sqft\_basement','view','bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']], df['price'])



pr=PolynomialFeatures(degree=2)

x\_train\_pr=pr.fit\_transform(x\_train[['floors', 'waterfront','lat' ,'bedrooms' ,'sqft\_basement' ,'view' ,'bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

x\_test\_pr=pr.fit\_transform(x\_test[['floors', 'waterfront','lat' ,'bedrooms' ,'sqft\_basement' ,'view' ,'bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

RidgeModel=Ridge(alpha=0.1)

RidgeModel.fit(x\_train\_pr, y\_train)

RidgeModel.score(x\_train\_pr, y\_train)



from sklearn.preprocessing import PolynomialFeatures

pr=PolynomialFeatures(degree=2)

pr

x\_train\_pr=pr.fit\_transform(x\_train[['floors', 'waterfront','lat' ,'bedrooms' ,'sqft\_basement' ,'view' ,'bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

x\_polly=pr.fit\_transform(x\_train[['floors', 'waterfront','lat' ,'bedrooms' ,'sqft\_basement' ,'view' ,'bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

RidgeModel=Ridge(alpha=0.1)

RidgeModel.fit(x\_train\_pr, y\_train)

RidgeModel.score(x\_train\_pr, y\_train)

x\_test\_pr=pr.fit\_transform(x\_test[['floors', 'waterfront','lat' ,'bedrooms' ,'sqft\_basement' ,'view' ,'bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

x\_polly=pr.fit\_transform(x\_test[['floors', 'waterfront','lat' ,'bedrooms' ,'sqft\_basement' ,'view' ,'bathrooms','sqft\_living15','sqft\_above','grade','sqft\_living']])

RidgeModel=Ridge(alpha=0.1)

RidgeModel.fit(x\_test\_pr, y\_test)

RidgeModel.score(x\_test\_pr, y\_test)

sns.boxplot(x="waterfront", y="price", data=df)

https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/7a3998cc-cd1f-4906-b24a-327ed953a9c0/view?access\_token=b319e528bf074238cab7a9b0ea5f5daf2edce0bbad1332a04169f39d2fd20790