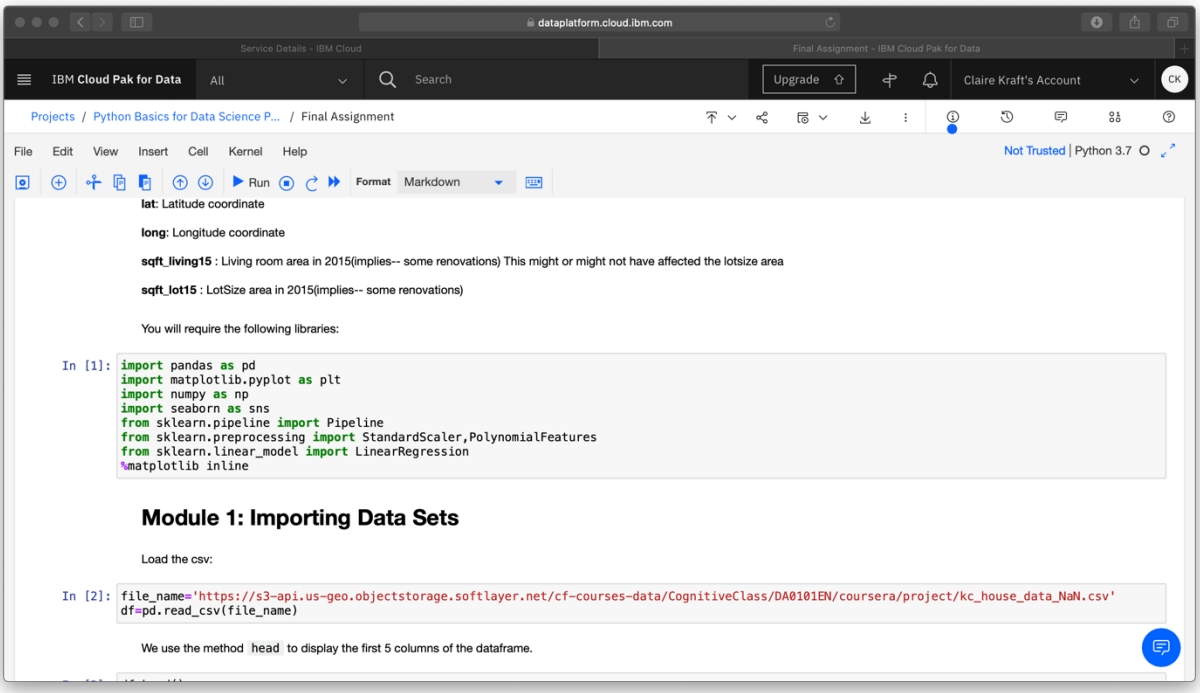
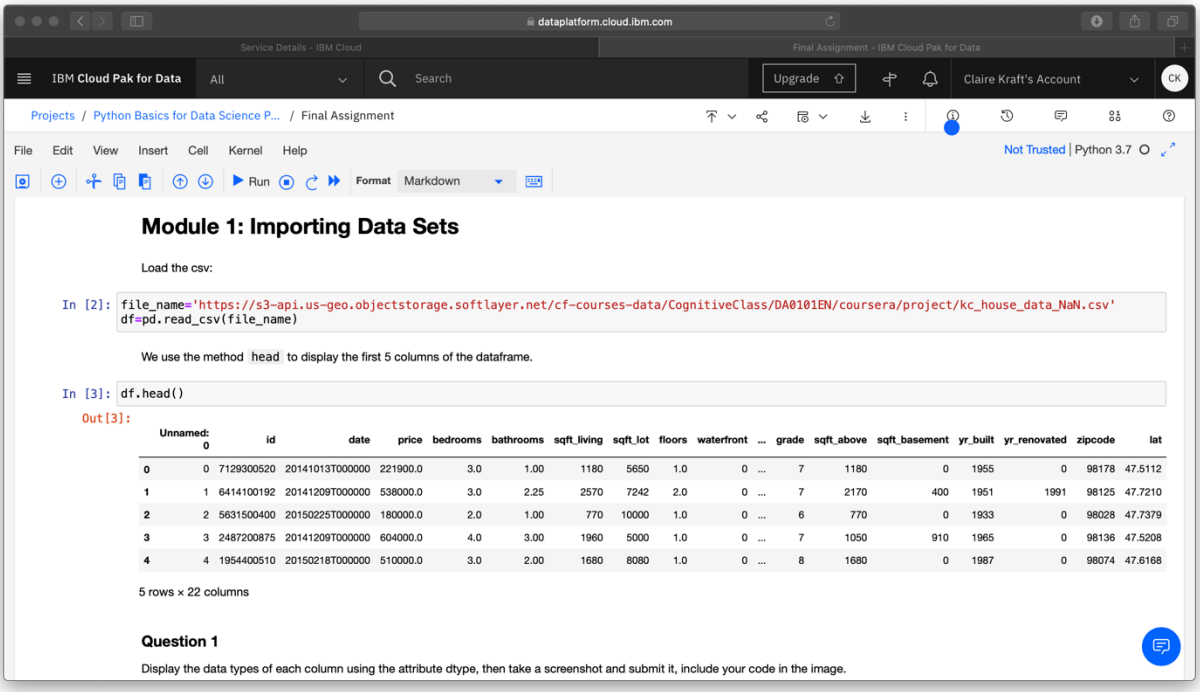
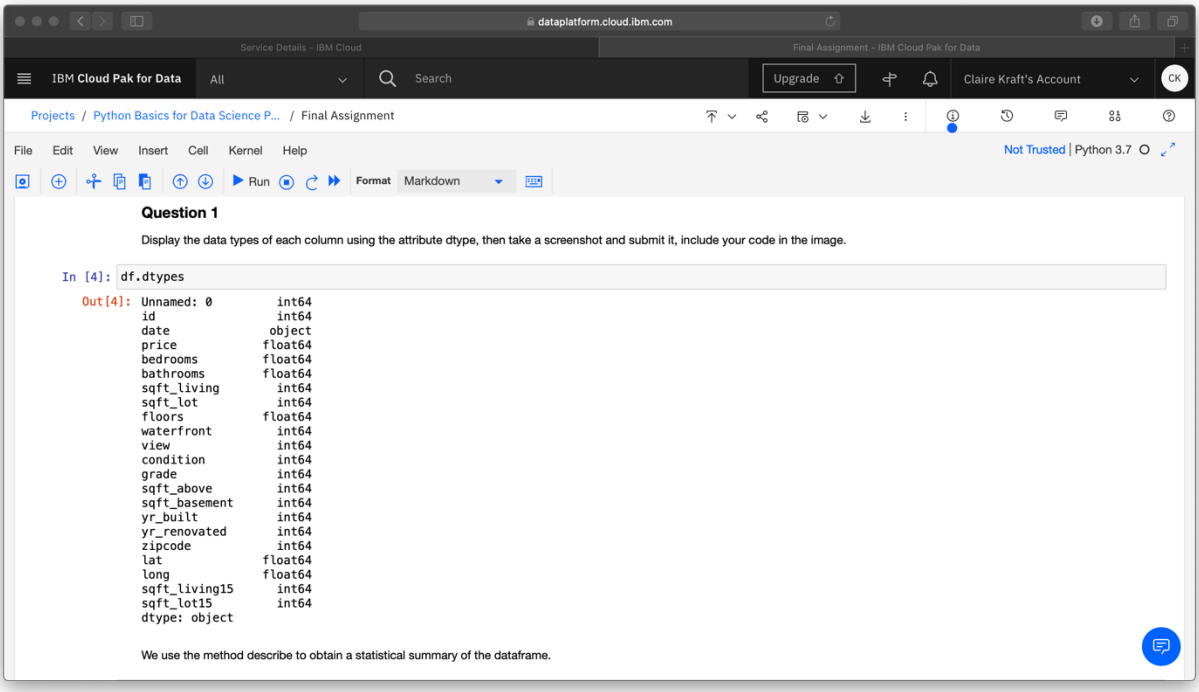
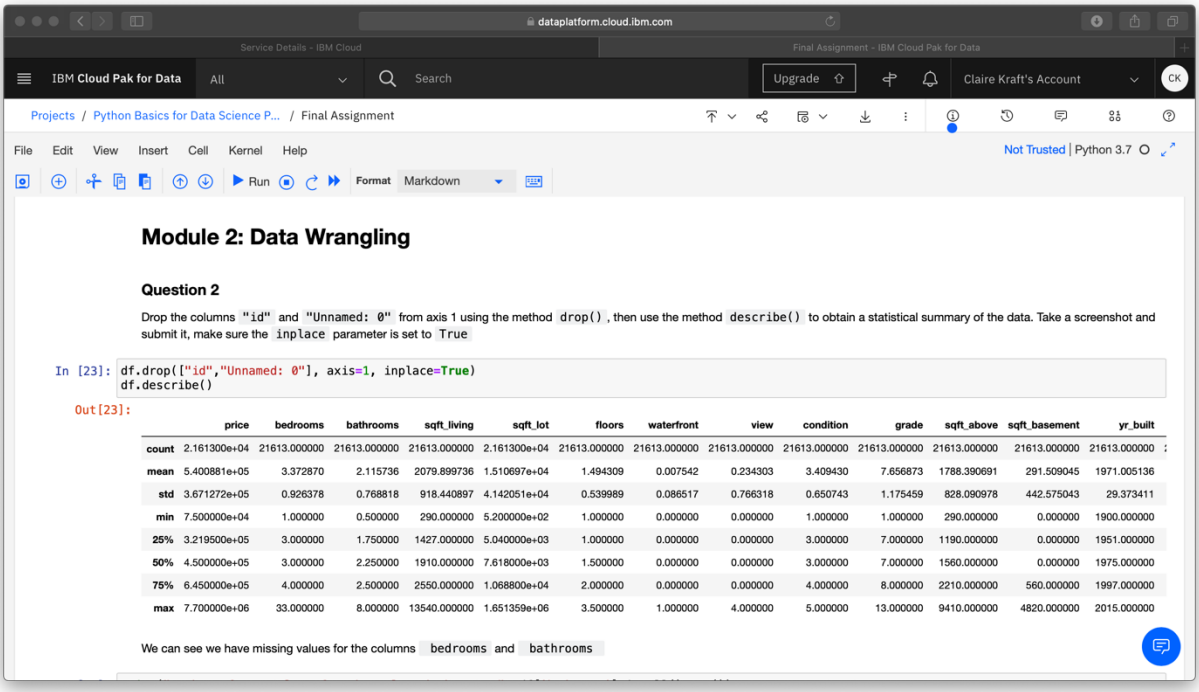
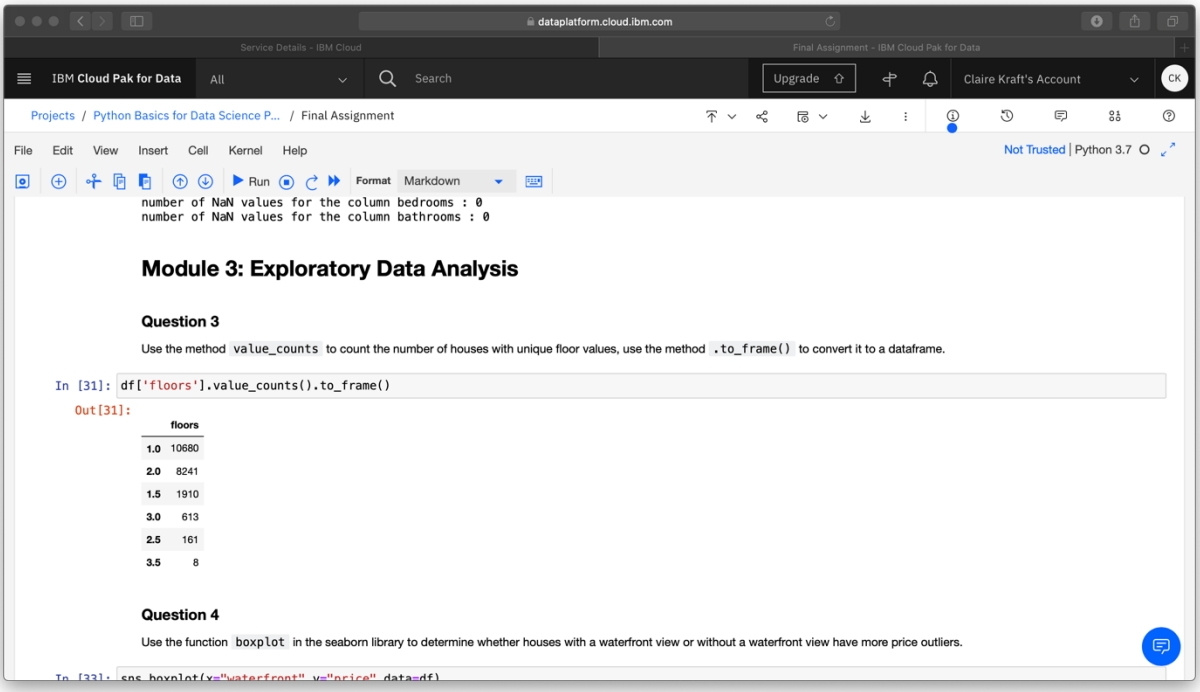
House Sales in King County, USA

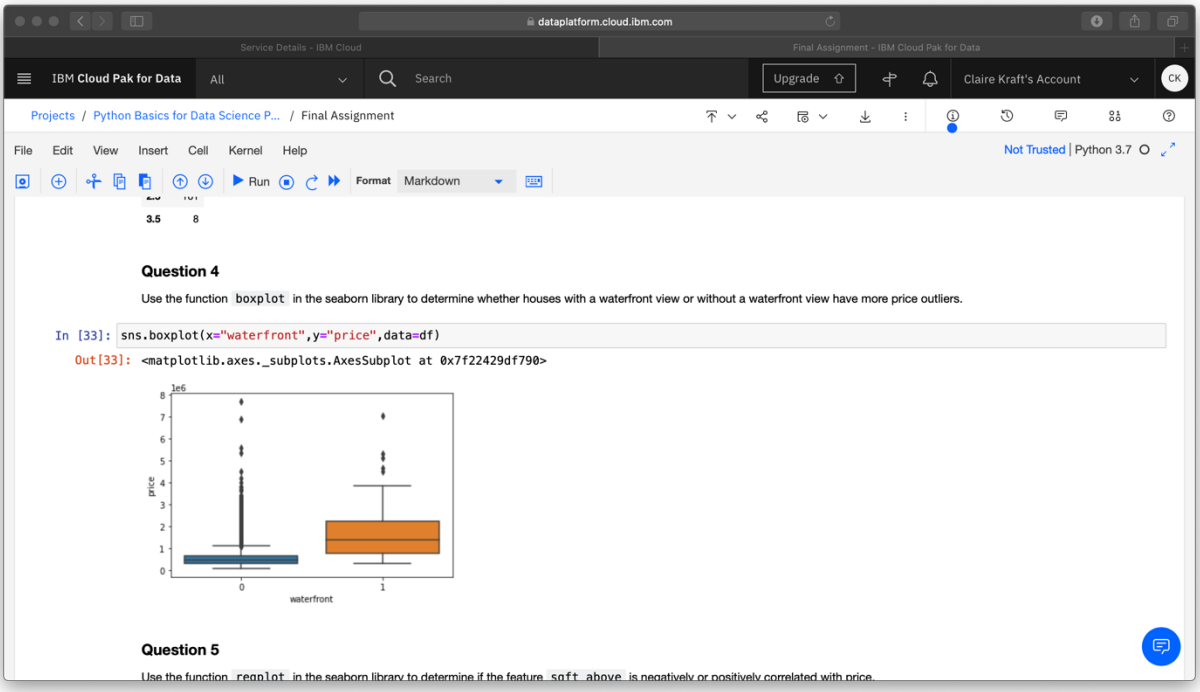





dataplatform.cloud.ibm.com
IBM Cloud Pak for Data
Search
Upgrade
Claire Kraft's Account
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
Question 1
Display the data types of each column using the attribute dtype, then take a screenshot and submit it, include your code in the image.
In [4]: df.dtypes
Out[4]:
Unnamed: 0 int64
id int64
date object
price float64
bedrooms float64
bathrooms float64
sqft_living int64
sqft_lot int64
floors float64
waterfront int64
view int64
condition int64
grade int64
sqft_above int64
sqft_basement int64
yr_built int64
yr_renovated int64
zipcode int64
lat float64
long float64
sqft_living15 int64
sqft_lot15 int64
dtype: object
We use the method describe to obtain a statistical summary of the dataframe.


dataplatform.cloud.ibm.com
IBM Cloud Pak for Data
Search
Upgrade
Claire Kraft's Account
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
Module 2: Data Wrangling
Question 2
Drop the columns "id" and "Unnamed: 0" from axis 1 using the method drop() , then use the method describe() to obtain a statistical summary of the data. Take a screenshot and submit it, make sure the inplace parameter is set to True
In [23]: df.drop(["id","Unnamed: 0"], axis=1, inplace=True)
df.describe()
Out[23]:
price bedrooms bathrooms sqft_living sqft_lot floors waterfront view condition grade sqft_above sqft_basement yr_built
count 2.161300e+04 21613.000000 21613.000000 21613.000000 2.161300e+04 21613.000000 21613.000000 21613.000000 21613.000000 21613.000000 21613.000000 21613.000000 21613.000000
mean 5.400881e+05 3.372870 2.115736 2079.899736 1.510697e+04 1.494309 0.007542 0.234303 3.409430 7.656873 1788.390691 291.509045 1971.005136
std 3.671272e+05 0.926378 0.768818 918.440897 4.142051e+04 0.539989 0.086517 0.766318 0.650743 1.175459 828.090978 442.575043 29.373411
min 7.500000e+04 1.000000 0.500000 290.000000 5.200000e+02 1.000000 0.000000 0.000000 1.000000 1.000000 290.000000 0.000000 1900.000000
25% 3.219500e+05 3.000000 1.750000 1427.000000 5.040000e+03 1.000000 0.000000 0.000000 3.000000 7.000000 1190.000000 0.000000 1951.000000
50% 4.500000e+05 3.000000 2.250000 1910.000000 7.618000e+03 1.500000 0.000000 0.000000 3.000000 7.000000 1560.000000 0.000000 1975.000000
75% 6.450000e+05 4.000000 2.500000 2550.000000 1.068800e+04 2.000000 0.000000 0.000000 4.000000 8.000000 2210.000000 560.000000 1997.000000
max 7.700000e+06 33.000000 8.000000 13540.000000 1.651359e+06 3.500000 1.000000 4.000000 5.000000 13.000000 9410.000000 4820.000000 2015.000000
We can see we have missing values for the columns bedrooms and bathrooms


dataplatform.cloud.ibm.com
IBM Cloud Pak for Data
Search
Upgrade
Claire Kraft's Account
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
number of NaN values for the column bedrooms : 0
number of NaN values for the column bathrooms : 0
Module 3: Exploratory Data Analysis
Question 3
Use the method value_counts to count the number of houses with unique floor values, use the method .to_frame() to convert it to a dataframe.
In [31]: df['floors'].value_counts().to_frame()
Out[31]:
floors
1.0 10680
2.0 8241
1.5 1910
3.0 613
2.5 161
3.5 8
Question 4
Use the function boxplot in the seaborn library to determine whether houses with a waterfront view or without a waterfront view have more price outliers.


dataplatform.cloud.ibm.com
IBM Cloud Pak for Data
Search
Upgrade
Claire Kraft's Account
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
3.5 8
Question 4
Use the function boxplot in the seaborn library to determine whether houses with a waterfront view or without a waterfront view have more price outliers.
In [33]: sns.boxplot(x="waterfront",y="price",data=df)
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7f22429df790>
1e6
price
waterfront
Question 5
Use the function regplot in the seaborn library to determine if the feature sqft_above is negatively or positively correlated with price.

dataplatform.cloud.ibm.com

Service Details - IBM Cloud | Final Assignment - IBM Cloud Pak for Data

IBM Cloud Pak for Data | All | Search | Upgrade | Claire Kraft's Account | CK

Projects / Python Basics for Data Science P... / Final Assignment

File Edit View Insert Cell Kernel Help | Not Trusted | Python 3.7

Run | Format Markdown

waterfront

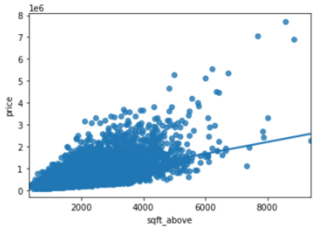
0 1

Question 5

Use the function `regplot` in the seaborn library to determine if the feature `sqft_above` is negatively or positively correlated with price.

```
In [34]: sns.regplot(x="sqft_above",y="price",data=df,ci=None)
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2242163c90>
```



We can use the Pandas method `corr()` to find the feature other than price that is most correlated with price.

```
In [35]: df.corr()['price'].sort_values()
```

dataplatform.cloud.ibm.com

Service Details - IBM Cloud | Final Assignment - IBM Cloud Pak for Data

IBM Cloud Pak for Data | All | Search | Upgrade | Claire Kraft's Account | CK

Projects / Python Basics for Data Science P... / Final Assignment

File Edit View Insert Cell Kernel Help | Not Trusted | Python 3.7

Run | Format Markdown

Module 4: Model Development

We can Fit a linear regression model using the longitude feature `'long'` and caculate the R^2.

```
In [36]: X = df[['long']]
         Y = df['price']
         lm = LinearRegression()
         lm.fit(X,Y)
         lm.score(X, Y)
```

```
Out[36]: 0.00046769430149007363
```

Question 6

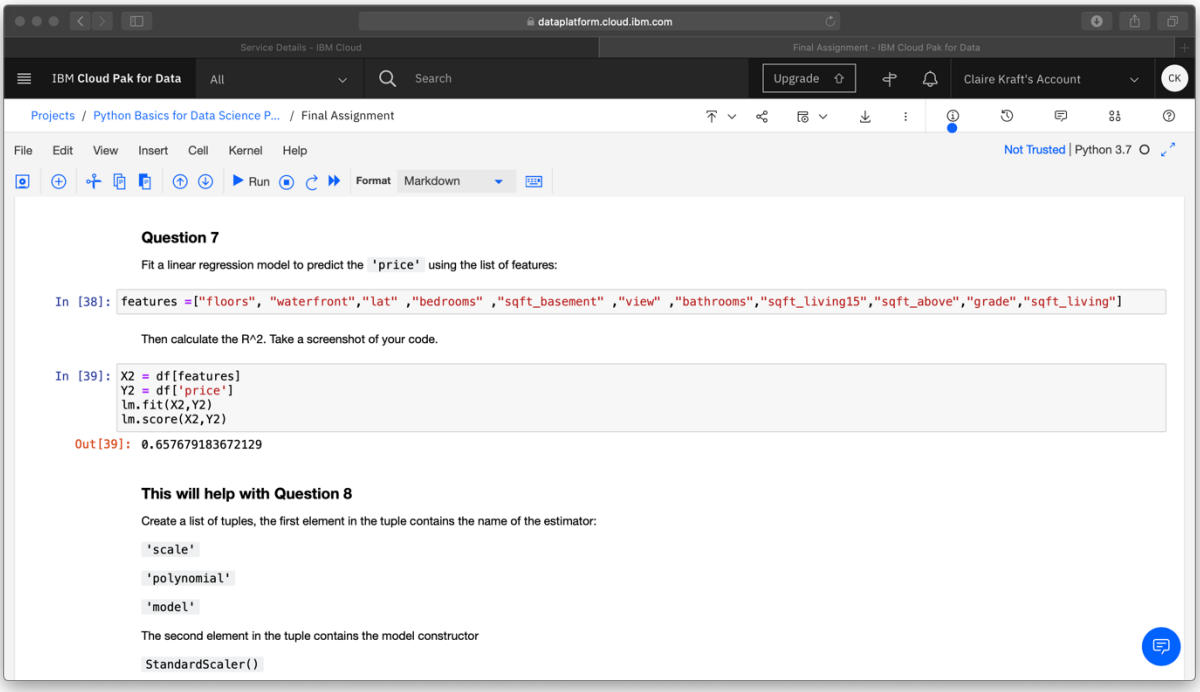
Fit a linear regression model to predict the `'price'` using the feature `'sqft_living'` then calculate the R^2. Take a screenshot of your code and the value of the R^2.

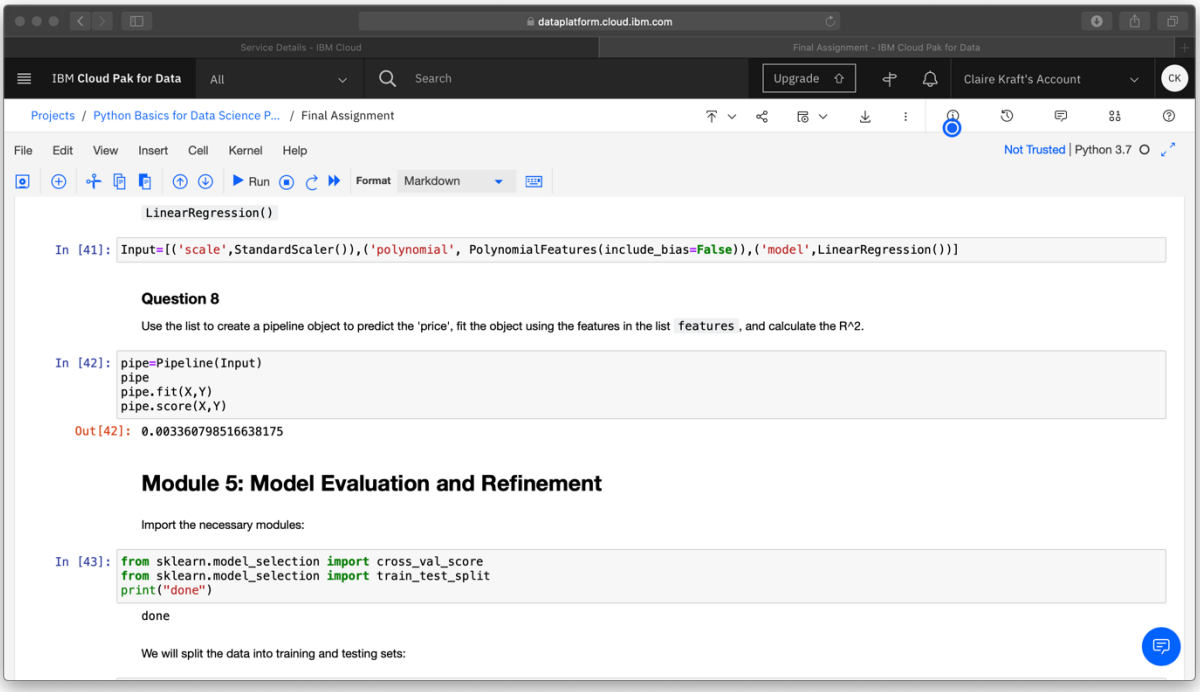
```
In [37]: X1 = df[['sqft_living']]
         Y1 = df['price']
         lm = LinearRegression()
         lm.fit(X1,Y1)
         lm.score(X1, Y1)
```

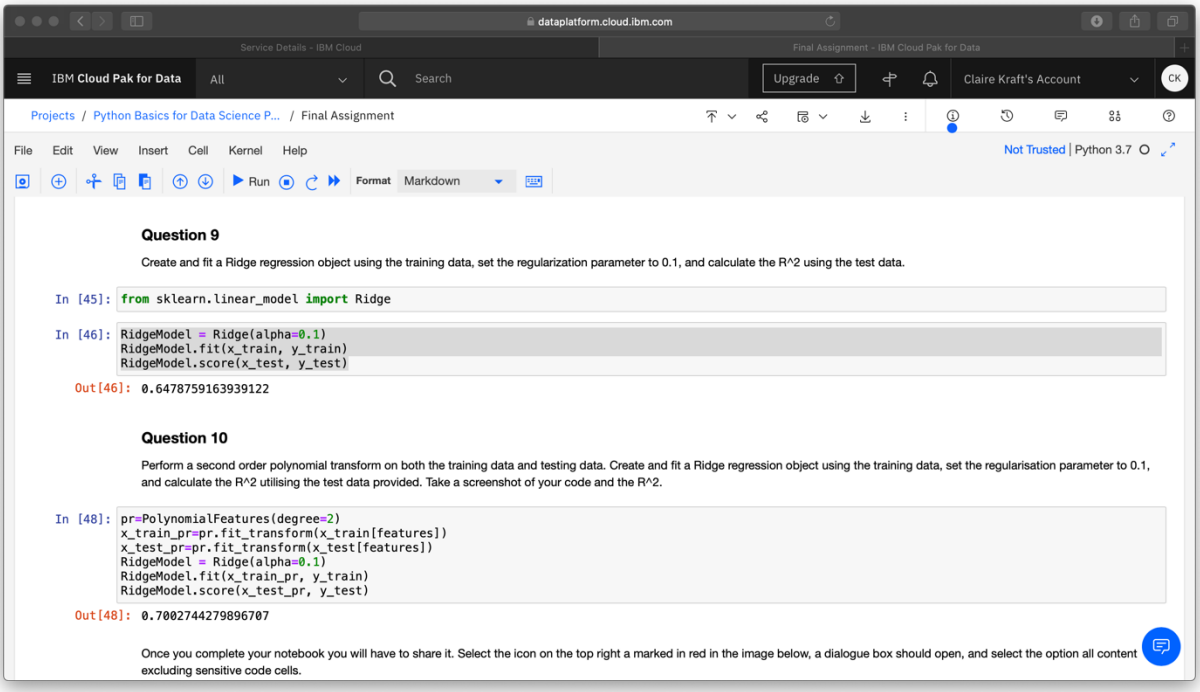
```
Out[37]: 0.4928532179037931
```

Question 7

Fit a linear regression model to predict the `'price'` using the list of features:


dataplatform.cloud.ibm.com
IBM Cloud Pak for Data
Search
Upgrade
Claire Kraft's Account
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
Question 7
Fit a linear regression model to predict the 'price' using the list of features:
In [38]: features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms","sqft_living15","sqft_above","grade","sqft_living"]
Then calculate the R^2. Take a screenshot of your code.
In [39]: X2 = df[features]
Y2 = df['price']
lm.fit(X2,Y2)
lm.score(X2,Y2)
Out[39]: 0.657679183672129
This will help with Question 8
Create a list of tuples, the first element in the tuple contains the name of the estimator:
'scale'
'polynomial'
'model'
The second element in the tuple contains the model constructor
StandardScaler()


dataplatform.cloud.ibm.com
IBM Cloud Pak for Data
Search
Upgrade
Claire Kraft's Account
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
LinearRegression()
In [41]: Input=[('scale',StandardScaler()),('polynomial', PolynomialFeatures(include_bias=False)),('model',LinearRegression())]
Question 8
Use the list to create a pipeline object to predict the 'price', fit the object using the features in the list features , and calculate the R^2.
In [42]: pipe=Pipeline(Input)
pipe
pipe.fit(X,Y)
pipe.score(X,Y)
Out[42]: 0.0033607985166381 75
Module 5: Model Evaluation and Refinement
Import the necessary modules:
In [43]: from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
print("done")
done
We will split the data into training and testing sets:


dataplatform.cloud.ibm.com
Service Details - IBM Cloud
Final Assignment - IBM Cloud Pak for Data
IBM Cloud Pak for Data
All
Search
Upgrade
Claire Kraft's Account
CK
Projects / Python Basics for Data Science P... / Final Assignment
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7
Run
Format Markdown
Question 9
Create and fit a Ridge regression object using the training data, set the regularization parameter to 0.1, and calculate the R^2 using the test data.
In [45]: from sklearn.linear_model import Ridge
In [46]: RidgeModel = Ridge(alpha=0.1)
RidgeModel.fit(x_train, y_train)
RidgeModel.score(x_test, y_test)
Out[46]: 0.6478759163939122
Question 10
Perform a second order polynomial transform on both the training data and testing data. Create and fit a Ridge regression object using the training data, set the regularisation parameter to 0.1, and calculate the R^2 utilising the test data provided. Take a screenshot of your code and the R^2.
In [48]: pr=PolynomialFeatures(degree=2)
x_train_pr=pr.fit_transform(x_train[features])
x_test_pr=pr.fit_transform(x_test[features])
RidgeModel = Ridge(alpha=0.1)
RidgeModel.fit(x_train_pr, y_train)
RidgeModel.score(x_test_pr, y_test)
Out[48]: 0.7002744279896707
Once you complete your notebook you will have to share it. Select the icon on the top right a marked in red in the image below, a dialogue box should open, and select the option all content excluding sensitive code cells.