Regression Week 1: Simple Linear Regression Assignment

Predicting House Prices (One feature)

In this notebook we will use data on house sales in King County, where Seattle is located, to predict house prices using simple (one feature) linear regression. You will:

- Use SArray and SFrame functions to compute important summary statistics

- Write a function to compute the Simple Linear Regression weights using the closed form solution

- Write a function to make predictions of the output given the input feature

- Turn the regression around to predict the input/feature given the output

- Compare two different models for predicting house prices

If you are doing the assignment with IPython Notebook

An IPython Notebook has been provided below to you for this assignment. This notebook contains the instructions, quiz questions and partially-completed code for you to use as well as some cells to test your code.

What you need to download

If you are using GraphLab Create:

- Download the King County House Sales data In SFrame format: kc_house_data.gl.zip (https://eventing.coursera.org/api/redirectStrict/bM1fwxhfTy7GFp7rTHwWV1R1DjP7bkIa t-dLI1d8bSAesj61Uf6GBeHFLurYeUMXIMGcFR-hqjQoPWcr-K4w-w.EqymiFp2NcrjarD7jVWoGA.Z0R3FVUEiIU6CG2JoBYunykeAL2J34LHjr6TLCsUaK2df-Plm0PSPbFfE0KJi3zV506a-pDnLybtngXz9KNBm8M7AEVhEVDkZERC1ADWO9nWNaEuaFsAobs41zmWr09zTaYgTVyX udYe_Fj1glKU2aUX7zTP6tlY81ZjDZ55Bi6A5PqNNzl1HoO7PAGMguHhIE7PhjP0F66LfTzYn

MSYxtcszAHkyDyR_xO4yFm03soaMnzSOxa7Q5TxnQG141Dwv2lZbpFf8rGARSRdFIGtAtc4

MSYxtcsZAHKyDyP_xQ4yFm03seaMp23Qxa7Q51xnQG141Dwy2JZbnEI8iGAP3PuFJGtAtc4
hxlXuiVApFRcoE8MwiTpS1eM1WtG5Wb4hhjkbeccm_cc0MZvw6ZAGQLu1A3YocQGpKEiB
8uOwrtYfRhSrhoH89LLX-
HL1zys8qhwPtQNTnBBFFS6elr2sGEbHdlkIHQeG1hml0rhvahx9xdnZll6WW5oLZPze0y7iL2
7P6ah)

- Download the companion IPython Notebook: week-1-simple-regression-quiz-
  blank.ipynb
  (https://eventing.coursera.org/api/redirectStrict/4yOcocpC3AmB2AuOWJf4zziiOgGfptGIL
  X0Jn1jaqhB6CPqOr__4QDMOBi0B7qDc1tIZo614v3dqXPlQ033Ijg.jlnNq99ZI1sT9E8MHkyB
  rQ.h2LnRfv_u8lV_pIDvhaq5OBN3AaTEwrBvZygFw_GScTD1kCcuc9cyMY_z16r9R1b5Tez49
  RdmyzNJ6z8Lzv5m98NLfGvE_qykghyIU0CNKuYaZ79fPK0WCj3qMru4J2gB3srPK1qlvZgz76
  xiSWOYxsflKyTS6v11nouMXIamAb7NL78yhBPUpQJeawuCtkFBxVg8j7A-
  3J5cJPSbPEu7yGszXpKsAFSxth7zXXquMujIMY7Yim6y7fNS-CSx8CmFFXa3DLW-
  1VSGHasePENzyMstHQMptaSEBB-
  HqlqiwmaXeMYyJtaJVL_CBr0d7CG8TkBTrvUE0g0LmJmdzNfQv-VpvVin-p1eOk4eJjKxUaTO-
  za9tzWJYxzRHSATN65rZ7zK8hL79nZ5YUHJJo2NIYOunPcb56qykb64W41_FO_WcBRPKb3T
  8-o4Li57mGKgY_sBcA0kQUoZ1m_Bxn3VmgEUc1dLwNntBjRrJQF2KFs9OUrsc_LWK1D-
  RwQTiDO)

- Save both of these files in the same directory (where you are calling IPython notebook
  from) and unzip the data file.

If you are not using GraphLab Create:

- Download the King County House Sales data csv file: kc_house_data.csv
  (https://eventing.coursera.org/api/redirectStrict/_PJzEhreTpSyhYj8gkdQa41ZuQqLhPzNd
  WtQJI8aZe0C4XXJiPOERgSDbn4SXfKpr0brkrVcPEfm95bKjMZliA.BTAfKWa957-
  f6sTOgrhgxQ.Sj04iL5dE0EHFl5yO232b_DMyw587JadWRFYeVpx1DSRANKt6u1enQIcZq9o
  CMYGpJYJK4e3qQ5TGkS_ssFQ8_6l4PKwTJXxF76etnzDQ_XOKy7wx_zU1DGvcVbj6MAZuq3
  3RI5RNj6LrQar4StnyIkLkSEvmH1lTs-
  cOYWgilwS9y5BLijLdkjmDAT1fDVaTqVKHnWkJCDqOBWcYdFmfvPvY97rRHK34aagzJW74
  N01m01-
  KCPKNE2S_T4ENr1y7HR_ZZrMBijkvgzdXyAMdt5wYwZvMandS8lK6Z5YMW20uGTDGRDFv
  7Au6JPFD41WUcNm_LNXmVV1xzZh3FuMd1Tnlp8W9-
  z3pprY0Dps6lgeynPYUczRl5UH6h1ktFeLhYMwg38BhBMAlZFsHEa9hAJ-
  eyZJCMASQOgXkQ3zOQANWVj0PnkCG5YkAw-_qmtt)

- Download the King County House Sales training data csv file: kc_house_train_data.csv
  (https://eventing.coursera.org/api/redirectStrict/P46GnjbWlfRyJKkd9kOXpL5pVaTCpPJOi
  OY-
  U5zMYVM_KZ1w3DxxkydE6Em2LWUFRY2eO5TB9mJavGehjTUpnA.lraBFnjgRX8uAiuN0f4
  Uew.8ysx0hLkt2wbyeW4Aebtin7JZayd4UQy4r3_wWyBxxZQGRNfRy7F3z2xSdekEh8IkfkD6
  sMxan11LqzvHyiXj9H-
  Uw7kw2OcmDLQyvEn66P3IXR5RI845nTpCRm5IVPIhpeT65er3griXdM6TFoz9otUIEdvV_tU
  cZaXfwlG9cT8TS3BD1ksNSfghI2Yrmeggh7v9OzKvOLd074WzVYPuMvfnBJvycXpGJx2gvskx
  hJku_bcttyYt4GGjamtnkG3cQyUW45pM7ulsiVjddZKeoqWcyUIGzFN7rAMcjL13Sp8Dg0uW
  DHIhNbNrzign8_VfPo_6mPcG

DHInNhNIZIgh8_vIRe_omPcG-
vNCdvFCmPgJiBriTEjiakm3b1fJowTaCZE4RCwXSDXPAgRmmzE-QMKivsHDCzNOctWJF05-
9_Fn8BkWPh9ODWUv2s_oil_5Gw4volcmy_7I86OmxzyMp4e)

- Download the King County House Sales testing data csv file: kc_house_test_data.csv
  (https://eventing.coursera.org/api/redirectStrict/hbPznZdwwW2F2_4oyzJxCvmq4DglAWn
  dNWqoFJS9cLcFA6Cxvp4jeODqyNaPxwrFH2WWQhGmVmez6-
  FLm3XrmA.x3GGw3sK43o0qqFDAwz3Kw.nG-G6M-bOblmIQWYoxLMmjLz-
  OHDLJuHvCTflXiWNDFoL2a5B0BTcrFt7Egwk4-
  cDdQUS4fIOEkiX4IpefMF1lRXv7eV5Gv74IddNWlxyiA-3_kwA_4HIj6oScKlHXmX-
  tRQCvo1Z5hHb0pQH_kYHOJur4jaJCmEjJHbUSQqNC0V6mTcfWPeJGxYBL99FvEtP881xmyT
  3In-lOKEADleL-
  jBkhI4oPCNkbvwxQh_isO8sB6Mbe2P5_64chd_HWAgWIbt92b1S86_jZMetJ93r2ZCuaZp_G
  HbxDqNA712GDYAdj44AMZbVbpRLMRG6t3B3TgdPoAqVWuWcrkc_39O7CAckRdILICsXPJ
  hLThSRc8iOjXZpDnRVAhWh8vcv_2I5rSFfRqKpEv5F2ODvi2kRw-
  ocjfvTKxhnSLbOaa7FLWM34mZvEYu02qlD3ju-p7H)

Useful resources

You may need to install the software tools or use the free Amazon EC2 machine.
Instructions for both options are provided in the reading for Module 1.

If instead you are using other tools to do your homework

You are welcome, however, to write your own code and use any other libraries, like Pandas
or R, to help you in the process. If you would like to take this path, follow the instructions
below.

1. If you are using SFrame, import graphlab and load in the house data, otherwise you can
also download the csv. (Note that we will be using the training and testing csv files
provided). e.g in python with SFrames:

```
sales = graphlab.SFrame('kc_house_data.gl/')
```

2. Split data into 80% training and 20% test data. Using SFrame, use this command to set
the same seed for everyone. e.g. in python with SFrames:

```
train_data,test_data = sales.random_split(.8,seed=0)
```

For those students not using graphlab please download the training and testing data csv
files.

From now on we will train the models using train_data. It will be important that we use the
same split here to ensure the results are the same.

3. Write a generic function that accepts a column of data (e.g. an SArray) 'input_feature' and

3. Write a generic function that accepts a column of data (e.g. an SArray) 'input_feature' and another column 'output' and returns the Simple Linear Regression parameters 'intercept' and 'slope'. Use the closed form solution from lecture to calculate the slope and intercept. e.g. in python:

```
def simple_linear_regression(input_feature, output):
    [your code here]
return(intercept, slope)
```

4. Use your function to calculate the estimated slope and intercept on the training data to predict 'price' given 'sqft_living'. e.g. in python with SFrames using:

```
input_feature = train_data['sqft_living']
output = train_data['price']
```

save the value of the slope and intercept for later (you might want to call them e.g. squarefeet_slope, and squarefeet_intercept)

5. Write a function that accepts a column of data 'input_feature', the 'slope', and the 'intercept' you learned, and returns an a column of predictions 'predicted_output' for each entry in the input column. e.g. in python:

```
def get_regression_predictions(input_feature, intercept, slope)
    [your code here]
return(predicted_output)
```

6. Quiz Question: Using your Slope and Intercept from (4), What is the predicted price for a house with 2650 sqft?

7. Write a function that accepts column of data: 'input_feature', and 'output' and the regression parameters 'slope' and 'intercept' and outputs the Residual Sum of Squares (RSS). e.g. in python:

```
def get_residual_sum_of_squares(input_feature, output, intercept,slope):
    [your code here]
return(RSS)
```

Recall that the RSS is the sum of the squares of the prediction errors (difference between output and prediction).

8. Quiz Question: According to this function and the slope and intercept from (4) What is the RSS for the simple linear regression using squarefeet to predict prices on TRAINING data?

9. Note that although we estimated the regression slope and intercept in order to predict the output from the input, since this is a simple linear relationship with only two variables we can invert the linear function to estimate the input given the output!

Write a function that accept a column of data:'output' and the regression parameters

Write a function that accept a column of data: output and the regression parameters 'slope' and 'intercept' and outputs the column of data: 'estimated_input'. Do this by solving the linear function output = intercept + slope*input for the 'input' variable (i.e. 'input' should be on one side of the equals sign by itself). e.g. in python:

```
def inverse_regression_predictions(output, intercept, slope):
    [your code here]
return(estimated_input)
```

10. Quiz Question: According to this function and the regression slope and intercept from (3) what is the estimated square-feet for a house costing $800,000?

11. Instead of using 'sqft_living' to estimate prices we could use 'bedrooms' (a count of the number of bedrooms in the house) to estimate prices. Using your function from (3) calculate the Simple Linear Regression slope and intercept for estimating price based on bedrooms. Save this slope and intercept for later (you might want to call them e.g. bedroom_slope, bedroom_intercept).

12. Now that we have 2 different models compute the RSS from BOTH models on TEST data.

13. Quiz Question: Which model (square feet or bedrooms) has lowest RSS on TEST data? Think about why this might be the case.