Project_Ames_Part1_zm57_xz573

February 20, 2018

In [1]:
```
import pandas as pd
import numpy as np
from __future__ import division
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from pylab import pcolor

%config InlineBackend.figure_format = 'png' #set 'png' here when working on notebook
%matplotlib inline
```

In [3]:
```
#Q1_A
data = pd.read_csv("ames_data.txt", sep = '\t')
for i in data:
    print i
'''
We chose the Ames Real Estate data set.
This dataset provides 1460 house sale prices with
79 attributes of the sold houses in the city of Ames, Iowa.
The meaning of each individual columns
is provided in the data set (3 examples, others can be found in dictionary):
MS SubClass: The building class
MS Zoning: The general zoning classification
Lot Frontage: Linear feet of street connected to property
We are concerned about how some of the ordinal data is recorded.
For example, Lot Shape has three categories: Regular,
Slightly irregular, Moderately Irregular, Irregular.
We are not sure if this data is recorded in a
very disciplinary and quantitative fashion.
If not, then we will have to interpret accordingly.

'''
```

```
Order
PID
MS SubClass
MS Zoning
Lot Frontage
```

Lot Area
Street
Alley
Lot Shape
Land Contour
Utilities
Lot Config
Land Slope
Neighborhood
Condition 1
Condition 2
Bldg Type
House Style
Overall Qual
Overall Cond
Year Built
Year Remod/Add
Roof Style
Roof Matl
Exterior 1st
Exterior 2nd
Mas Vnr Type
Mas Vnr Area
Exter Qual
Exter Cond
Foundation
Bsmt Qual
Bsmt Cond
Bsmt Exposure
BsmtFin Type 1
BsmtFin SF 1
BsmtFin Type 2
BsmtFin SF 2
Bsmt Unf SF
Total Bsmt SF
Heating
Heating QC
Central Air
Electrical
1st Flr SF
2nd Flr SF
Low Qual Fin SF
Gr Liv Area
Bsmt Full Bath
Bsmt Half Bath
Full Bath
Half Bath
Bedroom AbvGr

```
Kitchen AbvGr
Kitchen Qual
TotRms AbvGrd
Functional
Fireplaces
Fireplace Qu
Garage Type
Garage Yr Blt
Garage Finish
Garage Cars
Garage Area
Garage Qual
Garage Cond
Paved Drive
Wood Deck SF
Open Porch SF
Enclosed Porch
3Ssn Porch
Screen Porch
Pool Area
Pool QC
Fence
Misc Feature
Misc Val
Mo Sold
Yr Sold
Sale Type
Sale Condition
SalePrice
```

```
In [14]: #Q2_B
         data.isnull().any().any()
```

```
Out[14]: True
```

```
In [ ]: #Are any values in your dataset NULL or NA? There is according to result
        #Think of what you will do with rows with such
        #entries: do you plan to delete them, or still work with
        #the remaining columns for such rows?
        '''
        There are many NA values in this dataset.
        However, these NA values have actual meanings
        rather than being missing values.
        Therefore, we plan to replace the NA that has actual
        meanings to names that correspond to its actual meaning.
        Overall, the dataset has very few missing values.
        There are about 350 missing values for Lot Frontage,
```

```
        Mas Vnr Type, and Garage Yr Blt.
        We plan to either delete the missing entries or
        replace with the average value.

        '''
```

In [19]:
```python
#Q2_C
train, test = train_test_split(data, test_size=0.2)#it is not time-series data
train.to_csv('train.csv')
test.to_csv('test.csv')
```

In [23]:
```python
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

In [18]:
```python
#Q2_D
a = np.mean(data)
b = np.var(data)
c = pd.DataFrame(dict(mean = a, variance = b))
c.assign(variance_mean_ratio = lambda x: (x['variance'] /x['mean']))
```

Out[18]:

	mean	variance	variance_mean_ratio
Order	1.465500e+03	7.154082e+05	4.881667e+02
PID	7.144645e+08	3.560717e+16	4.983757e+07
MS SubClass	5.738737e+01	1.817381e+03	3.166865e+01
Lot Frontage	6.922459e+01	5.457151e+02	7.883256e+00
Lot Area	1.014792e+04	6.207349e+07	6.116867e+03
Overall Qual	6.094881e+00	1.990315e+00	3.265552e-01
Overall Cond	5.563140e+00	1.235092e+00	2.220134e-01
Year Built	1.971356e+03	9.144696e+02	4.638784e-01
Year Remod/Add	1.984267e+03	4.350030e+02	2.192261e-01
Mas Vnr Area	1.018968e+02	3.207029e+04	3.147331e+02
BsmtFin SF 1	4.426296e+02	2.074921e+05	4.687715e+02
BsmtFin SF 2	4.972243e+01	2.860820e+04	5.753581e+02
Bsmt Unf SF	5.592625e+02	1.930892e+05	3.452567e+02
Total Bsmt SF	1.051615e+03	1.940754e+05	1.845499e+02
1st Flr SF	1.159558e+03	1.535261e+05	1.324005e+02
2nd Flr SF	3.354560e+02	1.834603e+05	5.468982e+02
Low Qual Fin SF	4.676792e+00	2.143931e+03	4.584192e+02
Gr Liv Area	1.499690e+03	2.554520e+05	1.703365e+02
Bsmt Full Bath	4.313525e-01	2.753422e-01	6.383229e-01
Bsmt Half Bath	6.113388e-02	6.012877e-02	9.835589e-01
Full Bath	1.566553e+00	3.056390e-01	1.951029e-01
Half Bath	3.795222e-01	2.525499e-01	6.654418e-01
Bedroom AbvGr	2.854266e+00	6.849050e-01	2.399583e-01
Kitchen AbvGr	1.044369e+00	4.581300e-02	4.386669e-02
TotRms AbvGrd	6.443003e+00	2.473373e+00	3.838850e-01
Fireplaces	5.993174e-01	4.196582e-01	7.002270e-01
Garage Yr Blt	1.978132e+03	6.514646e+02	3.293332e-01
Garage Cars	1.766815e+00	5.782637e-01	3.272917e-01

```
Garage Area      4.728197e+02  4.622923e+04         9.777348e+01
Wood Deck SF     9.375188e+01  1.596179e+04         1.702557e+02
Open Porch SF    4.753345e+01  4.552455e+03         9.577372e+01
Enclosed Porch   2.301160e+01  4.112415e+03         1.787105e+02
3Ssn Porch       2.592491e+00  6.318708e+02         2.437311e+02
Screen Porch     1.600205e+01  3.144719e+03         1.965198e+02
Pool Area        2.243345e+00  1.266727e+03         5.646599e+02
Misc Val         5.063515e+01  3.206364e+05         6.332288e+03
Mo Sold          6.216041e+00  7.365954e+00         1.184991e+00
Yr Sold          2.007790e+03  1.732878e+00         8.630771e-04
SalePrice        1.807961e+05  6.379705e+09         3.528675e+04
```

In [30]:
```
c.assign(variance_mean_ratio = lambda x: (x['variance'] /x['mean'])).sort_values(['va
```

Out[30]:

	mean	variance	variance_mean_ratio
BsmtFin SF 2	4.972243e+01	2.860820e+04	5.753581e+02
Lot Area	1.014792e+04	6.207349e+07	6.116867e+03
Misc Val	5.063515e+01	3.206364e+05	6.332288e+03
SalePrice	1.807961e+05	6.379705e+09	3.528675e+04
PID	7.144645e+08	3.560717e+16	4.983757e+07

In [ ]:
```
#Are there any columns that appear to be random noise?
#(what is the definiton of random noise?)
#SalePrice is really outstanding
#in term of being noisy if considering the variance_mean_ratio
```

In [36]:
```
#Q2_E
'''
We choose SalePrice as our continuous response variable.
We think it is very intuitive to use all the features
of the house to predict its sale price,
and it will be quite interesting to find a good model
to conduct such prediction.

'''
```

In [54]:
```
#Q2_F
'''
There is no intuitive binary response variables in our dataset.
Therefore, we create a new variable:
those houses have sale prices over $200,000
are expensive houses, and otherwise cheap houses.
This intuitively makes sense.
The reason we choose $ 200,000 is because
the mean for house sale price is $180,000,
which is close to $200,000.
'''
#mean_binary
len(data[data['SalePrice']>200000])/len(data['SalePrice'])
```

Out[54]: 0.2924914675767918

In [93]:
```
#Q2_G
data.corr()['SalePrice'].sort_values()
```

Out[93]:
```
PID               -0.246521
Enclosed Porch    -0.128787
Kitchen AbvGr     -0.119814
Overall Cond      -0.101697
MS SubClass       -0.085092
Low Qual Fin SF   -0.037660
Bsmt Half Bath    -0.035835
Order             -0.031408
Yr Sold           -0.030569
Misc Val          -0.015691
BsmtFin SF 2       0.005891
3Ssn Porch         0.032225
Mo Sold            0.035259
Pool Area          0.068403
Screen Porch       0.112151
Bedroom AbvGr      0.143913
Bsmt Unf SF        0.182855
Lot Area           0.266549
2nd Flr SF         0.269373
Bsmt Full Bath     0.276050
Half Bath          0.285056
Open Porch SF      0.312951
Wood Deck SF       0.327143
Lot Frontage       0.357318
BsmtFin SF 1       0.432914
Fireplaces         0.474558
TotRms AbvGrd      0.495474
Mas Vnr Area       0.508285
Garage Yr Blt      0.526965
Year Remod/Add     0.532974
Full Bath          0.545604
Year Built         0.558426
1st Flr SF         0.621676
Total Bsmt SF      0.632280
Garage Area        0.640401
Garage Cars        0.647877
Gr Liv Area        0.706780
Overall Qual       0.799262
SalePrice          1.000000
Name: SalePrice, dtype: float64
```

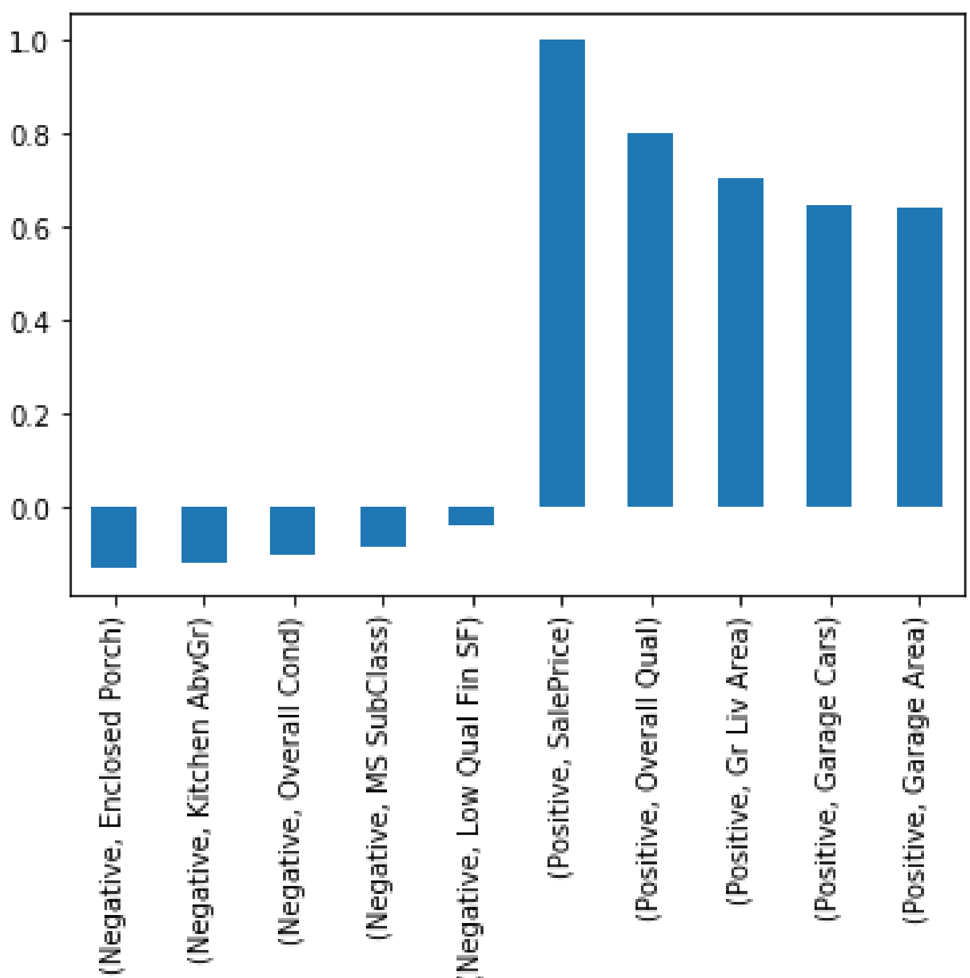
In [ ]:
```
'''
The most strongly positively correlated: Overall Qual, Gr Liv Area,
```

```
Garage Cars, Garage Area, Total Bsmt SF
The most strongly negatively correlated: "Enclosed Porch", "Kitchen AbvGr",
"Overall Cond", "MS SubClass", "Low Qual Fin SF"
Obviously, Garage Cars, Garage Area, Total Bsmt SF are somewhat
all correlated to Gr Liv Area.
And gross living area is definitely a very intuitive measure on
how much the cost cost.
On the other hand, one of the most negatively correlated is
Overall Condition of the house,
which doesnt make sense to us.
We would imagine a positive correlation between this variable and the sale price.

'''
```

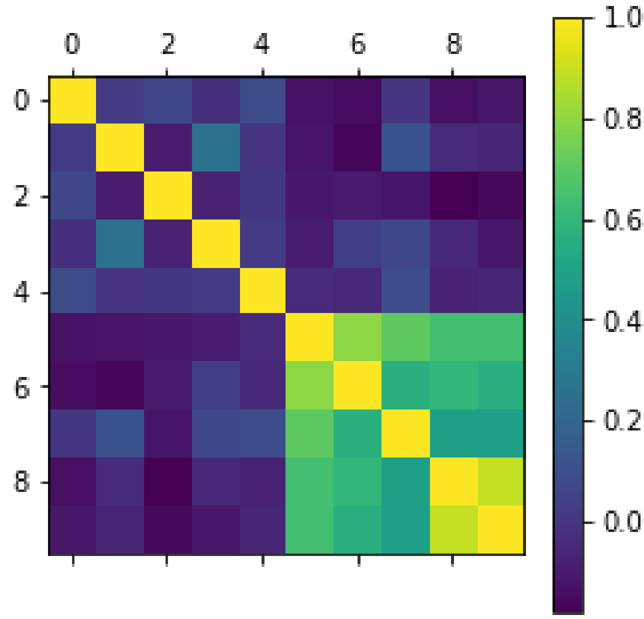
In [10]:
```
a = data.corr()['SalePrice'].sort_values()[1:6]
b = data.corr()['SalePrice'].sort_values(ascending = False)[0:5]
top10 = pd.concat([a,b], keys=['Negative', 'Positive'])
top10.plot(kind = 'bar')
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x117b75e10>



In [13]:
```python
#Q2_H
a = data.corr()['SalePrice'].sort_values()[1:6]
b = data.corr()['SalePrice'].sort_values(ascending = False)[0:5]
plt.matshow(data[a.index].join(data[b.index]).corr())
colorbar()
```

Out[13]: <matplotlib.colorbar.Colorbar at 0x118ae5610>



In [14]:
```python
data[a.index].join(data[b.index]).corr()
```

Out[14]:

	Enclosed Porch	Kitchen AbvGr	Overall Cond	MS SubClass \
Enclosed Porch	1.000000	0.027911	0.071459	-0.022866
Kitchen AbvGr	0.027911	1.000000	-0.086386	0.257698
Overall Cond	0.071459	-0.086386	1.000000	-0.067349
MS SubClass	-0.022866	0.257698	-0.067349	1.000000
Low Qual Fin SF	0.087326	0.000517	0.009175	0.025765
SalePrice	-0.128787	-0.119814	-0.101697	-0.085092
Overall Qual	-0.140332	-0.159744	-0.094812	0.039419
Gr Liv Area	0.004030	0.117836	-0.115643	0.068061
Garage Cars	-0.132840	-0.037092	-0.181557	-0.045883
Garage Area	-0.106272	-0.057779	-0.153754	-0.103239

	Low Qual Fin SF	SalePrice	Overall Qual	Gr Liv Area \
Enclosed Porch	0.087326	-0.128787	-0.140332	0.004030

```
Kitchen AbvGr           0.000517  -0.119814     -0.159744     0.117836
Overall Cond            0.009175  -0.101697     -0.094812    -0.115643
MS SubClass             0.025765  -0.085092      0.039419     0.068061
Low Qual Fin SF         1.000000  -0.037660     -0.048680     0.097050
SalePrice              -0.037660   1.000000      0.799262     0.706780
Overall Qual           -0.048680   0.799262      1.000000     0.570556
Gr Liv Area             0.097050   0.706780      0.570556     1.000000
Garage Cars            -0.067327   0.647877      0.599545     0.488829
Garage Area            -0.053510   0.640401      0.563503     0.484892

                 Garage Cars  Garage Area
Enclosed Porch     -0.132840    -0.106272
Kitchen AbvGr      -0.037092    -0.057779
Overall Cond       -0.181557    -0.153754
MS SubClass        -0.045883    -0.103239
Low Qual Fin SF    -0.067327    -0.053510
SalePrice           0.647877     0.640401
Overall Qual        0.599545     0.563503
Gr Liv Area         0.488829     0.484892
Garage Cars         1.000000     0.889676
Garage Area         0.889676     1.000000
```

In [ ]:
```
#Are correlations associative in your data?
#Yes, by looking at the sheet, Kitchen AbvGr is postively correlated with MS SubClass,
#and MS SubClass is correlated with Gr Liv Area, therefore Kitchen AbvGr is positively
#correlated with Gr Liv Area
```

In [ ]:
```
#Q2_I
'''
There arent any other variables we want to add to the dataset,
but we will surely do some substantial change to the current variables.
For example, we might change some ordinal values to numerical values or vice versa.
Also, for variables like Garage Finish, which has values: Finished, Rough Finished,
Unfinished, and No Garage, we might simply change the values to be
Have Garage and No Garage.

'''
```