

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

```
df = pd.read_csv('MELBOURNE_HOUSE_PRICES_LESS.csv')
df.head()
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-2-b03f3ba969b6> in <module>()
----> 1 df = pd.read_csv('MELBOURNE_HOUSE_PRICES_LESS.csv')
      2 df.head()

----- 4 frames -----
/usr/local/lib/python3.6/dist-packages/pandas/io/parsers.py in __init__(self, src,
**kwargs)
    2008         kws["usecols"] = self.usecols
    2009
-> 2010         self._reader = parsers.TextReader(src, **kws)
    2011         self.unnamed_cols = self._reader.unnamed_cols
    2012

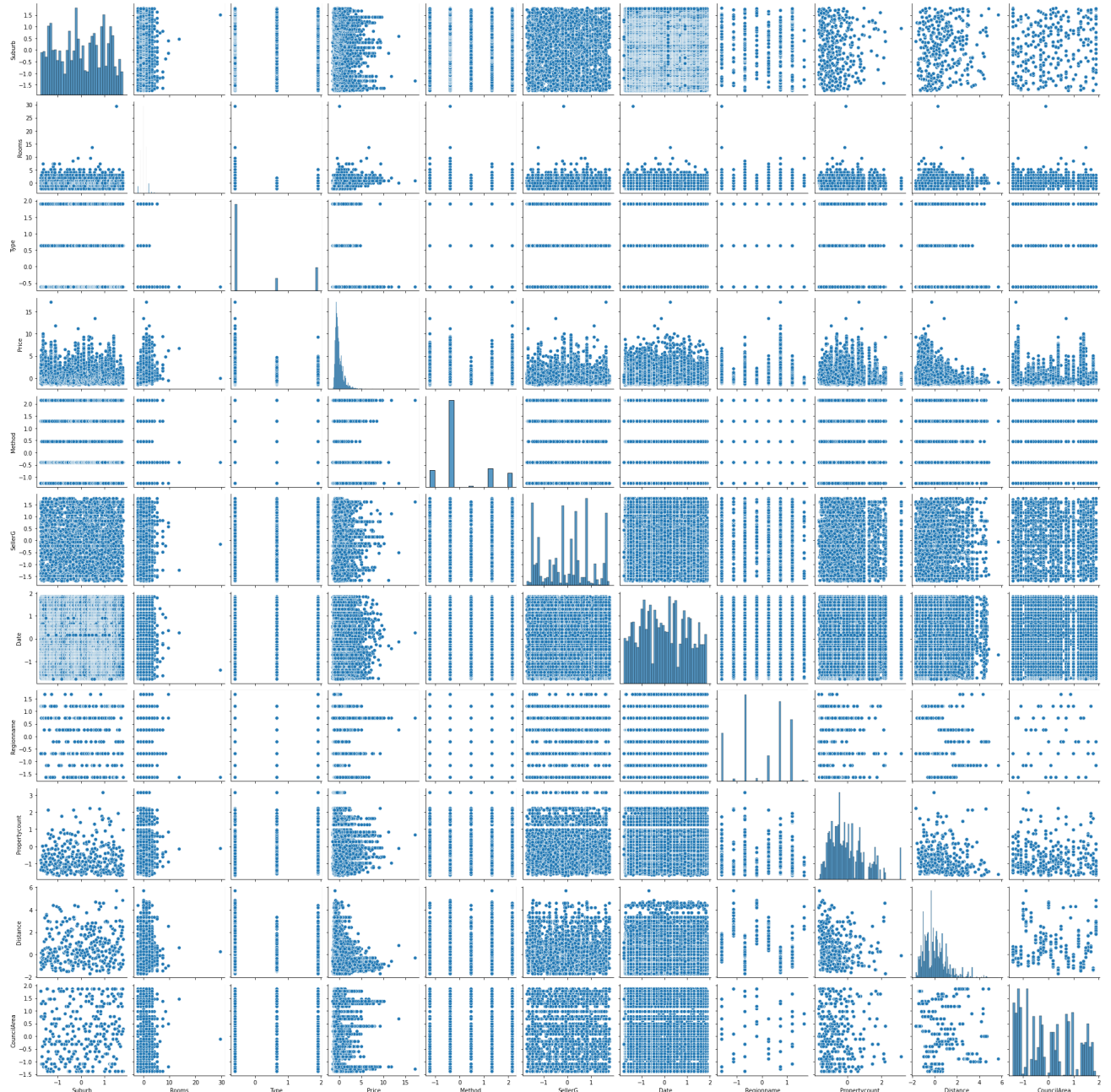
pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()

FileNotFoundError: [Errno 2] No such file or directory:
'MELBOURNE_HOUSE_PRICES_LESS.csv'
```

```
sns.pairplot(df)
```

<seaborn.axisgrid.PairGrid at 0x7fc53839e2b0>



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63023 entries, 0 to 63022
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Suburb          63023 non-null  object
1   Address         63023 non-null  object
2   Rooms           63023 non-null  int64
3   Type            63023 non-null  object
4   Price           48433 non-null  float64
5   Method          63023 non-null  object
6   SellerG         63023 non-null  object
```

```

7   Date          63023 non-null object
8   Postcode      63023 non-null int64
9   Regionname    63023 non-null object
10  Propertycount 63023 non-null int64
11  Distance      63023 non-null float64
12  CouncilArea   63023 non-null object
dtypes: float64(2), int64(3), object(8)
memory usage: 6.3+ MB

```

```
df['Date'] = pd.to_datetime(df['Date'])
```

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63023 entries, 0 to 63022
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Suburb          63023 non-null object
1   Address         63023 non-null object
2   Rooms           63023 non-null int64
3   Type            63023 non-null object
4   Price           48433 non-null float64
5   Method          63023 non-null object
6   SellerG         63023 non-null object
7   Date            63023 non-null datetime64[ns]
8   Postcode        63023 non-null int64
9   Regionname      63023 non-null object
10  Propertycount   63023 non-null int64
11  Distance         63023 non-null float64
12  CouncilArea     63023 non-null object
dtypes: datetime64[ns](1), float64(2), int64(3), object(7)
memory usage: 6.3+ MB

```

```
df = df.dropna()
```

```

print('Number of uniques in the columns:')
for i in df.columns:
    print(i, len(df[i].unique()))

```

```

Number of uniques in the columns:
Suburb 370
Address 44739
Rooms 14
Type 3
Price 3417
Method 5
SellerG 422
Date 112
Postcode 221
Regionname 8
Propertycount 359
Distance 176
CouncilArea 34

```

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```

label = LabelEncoder()
df['Suburb']=label.fit_transform(df['Suburb'])
df['Type']=label.fit_transform(df['Type'])
df['Method']=label.fit_transform(df['Method'])
df['SellerG']=label.fit_transform(df['SellerG'])
df['Date']=label.fit_transform(df['Date'])
df['Regionname']=label.fit_transform(df['Regionname'])
df['CouncilArea']=label.fit_transform(df['CouncilArea'])

df

```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Postcode	Re
<b>0</b>	0	49 Lithgow St	3	0	1490000.0	1	180	0	3067	
<b>1</b>	0	59A Turner St	3	0	1220000.0	1	225	0	3067	
<b>2</b>	0	119B Yarra St	3	0	1420000.0	1	253	0	3067	
<b>3</b>	1	68 Vida St	3	0	1515000.0	1	28	0	3040	
<b>4</b>	2	92 Clydesdale Rd	2	0	670000.0	1	253	0	3042	
...	...	...	...	...	...	...	...	...	...	
<b>63016</b>	147	4/34 Petrie St	2	2	347700.0	3	14	88	3199	
<b>63017</b>	275	229 Murray Rd	3	0	808000.0	1	305	88	3072	

```
df = df[['Suburb', 'Rooms', 'Type', 'Price', 'Method', 'SellerG', 'Date', 'Regionname', 'Propertycount', 'Distance', 'CouncilArea']]
```

```

scaler = StandardScaler()
df[['Suburb']] = scaler.fit_transform(df[['Suburb']])
df[['Rooms']] = scaler.fit_transform(df[['Rooms']])
df[['Type']] = scaler.fit_transform(df[['Type']])
df[['Price']] = scaler.fit_transform(df[['Price']])
df[['Method']] = scaler.fit_transform(df[['Method']])
df[['SellerG']] = scaler.fit_transform(df[['SellerG']])
df[['Date']] = scaler.fit_transform(df[['Date']])
df[['Regionname']] = scaler.fit_transform(df[['Regionname']])
df[['Propertycount']] = scaler.fit_transform(df[['Propertycount']])
df[['Distance']] = scaler.fit_transform(df[['Distance']])
df[['CouncilArea']] = scaler.fit_transform(df[['CouncilArea']])

df

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
isetter(loc, value[:, i].tolist())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

This is separate from the ipykernel package so we can avoid doing imports until

```
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
isetter(loc, value[:, i].tolist())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

after removing the cwd from sys.path.

```
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
isetter(loc, value[:, i].tolist())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
"""
```

```
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
isetter(loc, value[:, i].tolist())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/using\\_indexers.html](https://pandas.pydata.org/pandas-docs/stable/using_indexers.html)

```
isetter(loc, value[:, i].tolist())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
import sys  
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
isetter(loc, value[:, i].tolist())  
/usr/local/lib/python3.6/dist-packages/ipykernel\_launcher.py:8: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
isetter(loc, value[:, i].tolist())  
/usr/local/lib/python3.6/dist-packages/ipykernel\_launcher.py:9: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
if \_\_name\_\_ == '\_\_main\_\_':  
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
isetter(loc, value[:, i].tolist())  
/usr/local/lib/python3.6/dist-packages/ipykernel\_launcher.py:10: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
# Remove the CWD from sys.path while we load stuff.  
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
isetter(loc, value[:, i].tolist())  
/usr/local/lib/python3.6/dist-packages/ipykernel\_launcher.py:11: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
# This is added back by InteractiveShellApp.init\_path()  
/usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>  
isetter(loc, value[:, i].tolist())  
/usr/local/lib/python3.6/dist-packages/ipykernel\_launcher.py:12: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

```
try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html#loc-i-loc-r>  
 if sys.path[0] == '':  
 /usr/local/lib/python3.6/dist-packages/pandas/core/indexing.py:1734: SettingWithCopyWarning:  
 A value is trying to be set on a copy of a slice from a DataFrame.  
 Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html#loc-i-loc-r>  
 isetter(loc, value[:, i].tolist())

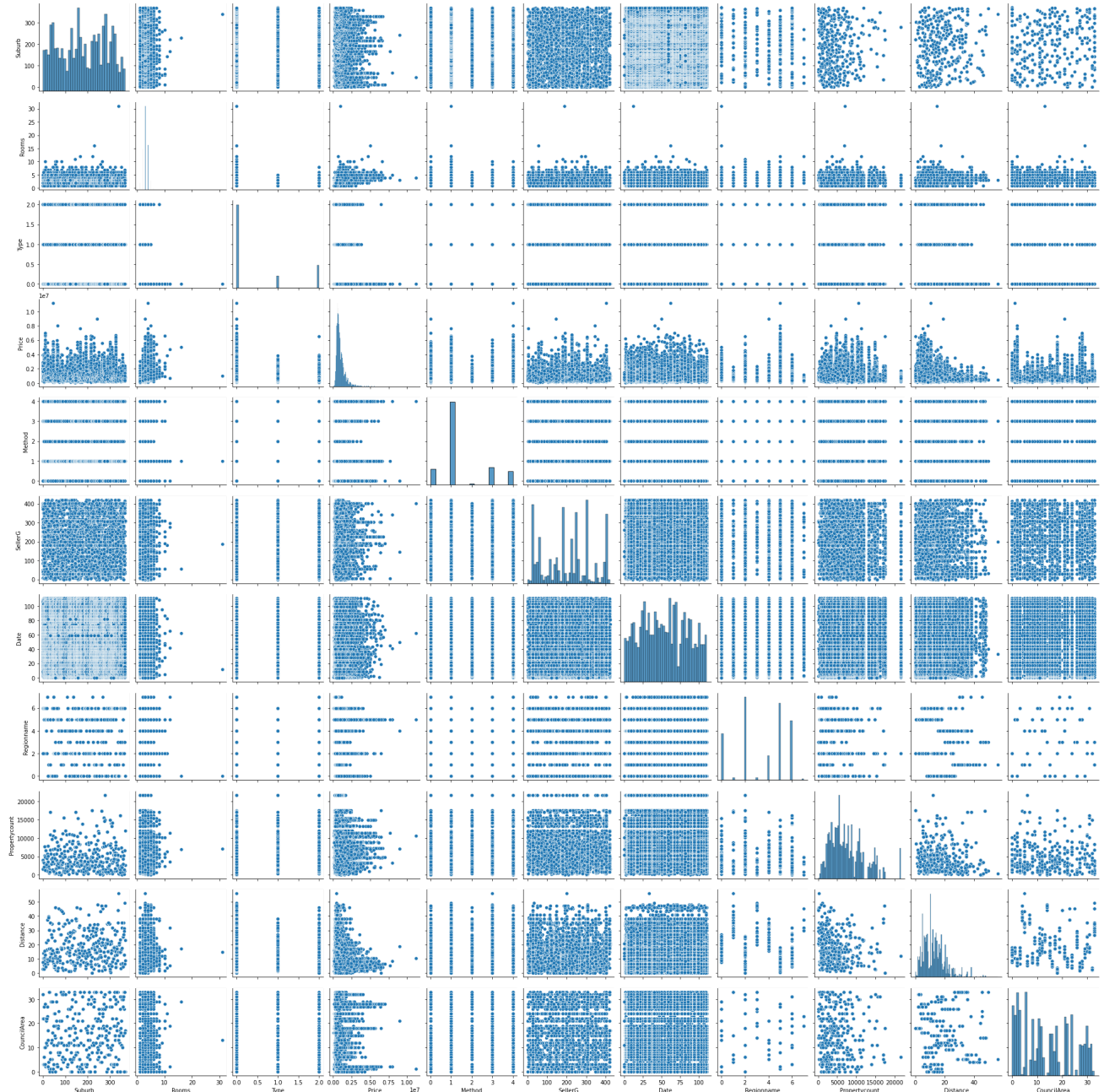
	Suburb	Rooms	Type	Price	Method	SellerG	Date	Region
<b>0</b>	-1.743259	-0.075861	-0.611260	0.829162	-0.394141	-0.218651	-1.777147	-0.68
<b>1</b>	-1.743259	-0.075861	-0.611260	0.374228	-0.394141	0.152343	-1.777147	-0.68
<b>2</b>	-1.743259	-0.075861	-0.611260	0.711216	-0.394141	0.383185	-1.777147	-0.68
<b>3</b>	-1.733626	-0.075861	-0.611260	0.871286	-0.394141	-1.471789	-1.777147	1.20
<b>4</b>	-1.723993	-1.134401	-0.611260	-0.552489	-0.394141	0.383185	-1.777147	1.20
...	...	...	...	...	...	...	...	...
<b>63016</b>	-0.327250	-1.134401	1.901485	-1.095545	1.305947	-1.587210	1.028760	0.25
<b>63017</b>	0.905737	-0.075861	-0.611260	-0.319967	-0.394141	0.811890	1.028760	-0.68
<b>63018</b>	1.030962	-0.075861	-0.611260	-0.727723	-0.394141	0.820134	1.028760	-0.68
<b>63019</b>	1.030962	-0.075861	-0.611260	-0.838929	-0.394141	0.820134	1.028760	-0.68
<b>63020</b>	1.030962	-0.075861	-0.611260	-0.763107	-0.394141	0.820134	1.028760	-0.68

48433 rows × 11 columns

```
sns.pairplot(df)
```



<seaborn.axisgrid.PairGrid at 0x7ff487c512b0>



```
import statsmodels.api as sm
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning
import pandas.util.testing as tm
```

```
df.columns
```

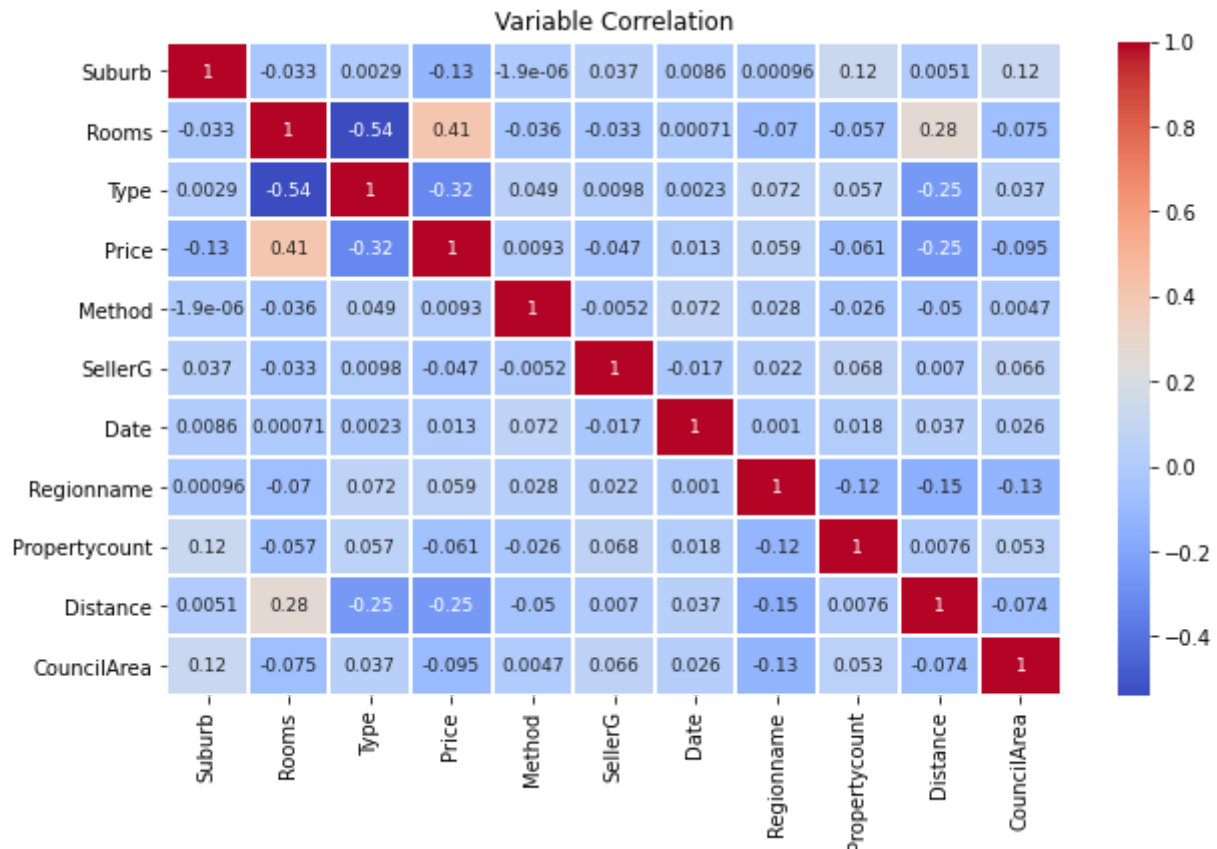
```
Index(['Suburb', 'Rooms', 'Type', 'Price', 'Method', 'SellerG', 'Date',
       'Regionname', 'Propertycount', 'Distance', 'CouncilArea'],
      dtype='object')
```

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(),cmap = 'coolwarm',linewidth = 1,annot= True, annot_kws={"size": 9})
```



```
plt.title('Variable Correlation')
```

```
Text(0.5, 1.0, 'Variable Correlation')
```



```
x = df[['Suburb', 'Rooms', 'Type', 'Method', 'SellerG', 'Date',
        'Regionname', 'Propertycount', 'Distance', 'CouncilArea']]
y = df[['Price']]
```

```
x = sm.add_constant(x)
model = sm.OLS(y,x).fit()
predictions = model.predict(x)
print_model = model.summary()
print(print_model)
```

#### OLS Regression Results

```
=====
Dep. Variable:          Price      R-squared:                0.364
Model:                  OLS        Adj. R-squared:           0.364
Method:                 Least Squares    F-statistic:           2771.
Date:                   Thu, 17 Dec 2020    Prob (F-statistic):    0.00
Time:                   16:45:33    Log-Likelihood:        -7.0162e+05
No. Observations:       48433    AIC:                   1.403e+06
Df Residuals:           48422    BIC:                   1.403e+06
Df Model:               10
Covariance Type:        nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	8.07e+05	1.37e+04	58.794	0.000	7.8e+05	8.34e+05
Suburb	-581.6586	21.038	-27.648	0.000	-622.894	-540.423
Rooms	2.611e+05	2752.588	94.845	0.000	2.56e+05	2.66e+05
Type	-1.48e+05	3236.006	-45.749	0.000	-1.54e+05	-1.42e+05

Method	5322.7650	1838.107	2.896	0.004	1720.052	8925.478
SellerG	-96.0071	17.842	-5.381	0.000	-130.977	-61.037
Date	578.0809	70.835	8.161	0.000	439.243	716.918
Regionname	7842.1044	1046.373	7.495	0.000	5791.199	9893.010
Propertycount	-0.2012	0.493	-0.408	0.683	-1.167	0.764
Distance	-3.302e+04	303.300	-108.863	0.000	-3.36e+04	-3.24e+04
CouncilArea	-4217.7244	217.541	-19.388	0.000	-4644.108	-3791.341
=====						
Omnibus:	30916.324	Durbin-Watson:	1.438			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	892274.564			
Skew:	2.618	Prob(JB):	0.00			
Kurtosis:	23.365	Cond. No.	5.69e+04			
=====						

#### Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
 [2] The condition number is large, 5.69e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import ensemble
from sklearn import metrics
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
from xgboost import XGBRegressor
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
```

```
df1 = pd.read_csv('housing-1.csv')
df1
```

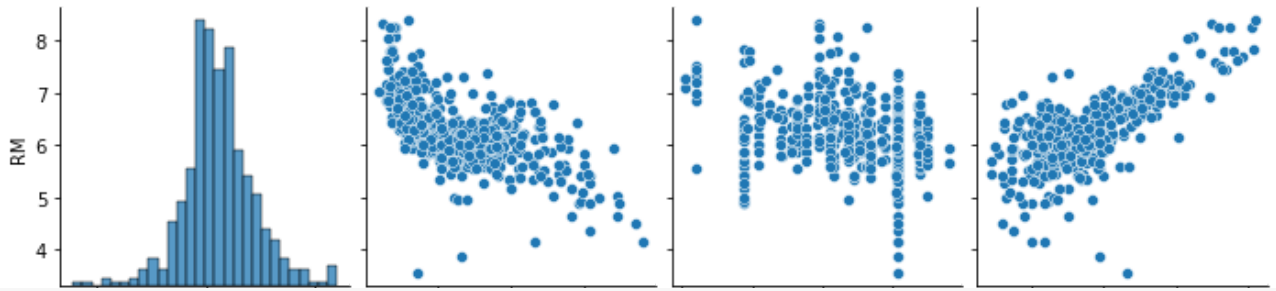
	RM	LSTAT	PTRATIO	MEDV
0	6.575	4.98	15.3	504000.0
1	6.421	9.14	17.8	453600.0
2	7.185	4.03	17.8	728700.0
3	6.998	2.94	18.7	701400.0

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 489 entries, 0 to 488
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype  
---  -
0    RM          489 non-null   float64
1    LSTAT       489 non-null   float64
2    PTRATIO     489 non-null   float64
3    MEDV       489 non-null   float64
dtypes: float64(4)
memory usage: 15.4 KB
```

```
sns.pairplot(df1)
```

```
<seaborn.axisgrid.PairGrid at 0x7f6bc5e0bda0>
```



```
df1.columns
```

```
Index(['RM', 'LSTAT', 'PTRATIO', 'MEDV'], dtype='object')
```



```
x = df1[['RM', 'LSTAT', 'PTRATIO']]
```

```
y = df1[['MEDV']]
```



```
x = sm.add_constant(x)
model = sm.OLS(y,x).fit()
predictions = model.predict(x)
print_model = model.summary()
print(print_model)
```

#### OLS Regression Results

```
=====
Dep. Variable:          MEDV    R-squared:                0.718
Model:                  OLS      Adj. R-squared:           0.716
Method:                 Least Squares    F-statistic:         410.9
Date:                  Sun, 20 Dec 2020    Prob (F-statistic):    9.96e-133
Time:                  05:13:32    Log-Likelihood:       -6259.9
No. Observations:      489    AIC:                  1.253e+04
Df Residuals:          485    BIC:                  1.254e+04
Df Model:              3
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	4.155e+05	6.88e+04	6.035	0.000	2.8e+05	5.51e+05
RM	8.657e+04	7888.895	10.973	0.000	7.11e+04	1.02e+05
LSTAT	-1.085e+04	732.138	-14.819	0.000	-1.23e+04	-9410.786
PTRATIO	-1.949e+04	2039.047	-9.559	0.000	-2.35e+04	-1.55e+04

```
=====
Omnibus:                57.976    Durbin-Watson:           1.049
Prob(Omnibus):           0.000    Jarque-Bera (JB):        91.413
Skew:                    0.772    Prob(JB):                1.41e-20
Kurtosis:                4.451    Cond. No.                416.
=====
```

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly spec

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = .20, random_state= 0)
```

```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

```
lr = LinearRegression()
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
lr.score(x_test, y_test)
```

```
0.6574622113312862
```

```
gbr = ensemble.GradientBoostingRegressor(n_estimators = 400, max_depth = 2, min_samples_sp
    learning_rate = 0.1, loss = 'ls')
gbr.fit(x_train, y_train)
gbr.score(x_test, y_test)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/ensemble/_gb.py:1454: DataConversionWarning:
  y = column_or_1d(y, warn=True)
0.8111816188588544
```

```
rf = RandomForestRegressor(n_estimators=1000, max_depth=5)
rf.fit(x_train, y_train)
rf.score(x_test, y_test)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: DataConversionWarning:
0.818449928806004
```

```
from sklearn.neural_network import MLPRegressor
```

```
mlpr = MLPRegressor(hidden_layer_sizes=(4,32,64,128,64,32), max_iter=4000)
mlpr.fit(x_train,y_train)
mlpr.score(x_test,y_test)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron
  y = column_or_1d(y, warn=True)
0.7852554244718521
```

```
xg = XGBRegressor(n_estimators=300, max_depth='2')
xg.fit(x_train, y_train)
print(xg.score(x_test, y_test))
yhat = xg.predict(x_test)
```

```
[05:14:55] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now
0.8118291295071038
```

```
print('MAE:',metrics.mean_absolute_error(y_test,yhat))
print('MSE:',metrics.mean_squared_error(y_test,yhat))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,yhat)))
print('R Square:',metrics.r2_score(y_test,yhat))
```

```
metrics.explained_variance_score(y_test, yhat)
```

```
MAE: 54435.35283801021
MSE: 5343738476.644528
RMSE: 73100.87876793636
R Square: 0.8118291295071038
0.8122205065195367
```

```
x_test.shape
```

```
(98, 4)
```

```
model = Sequential([
    Dense(32, activation='relu', input_shape = (4,)),
    Dense(32, activation='relu'),
    Dense(1, activation='relu'),
])
```

```
model.compile(optimizer='adam',
              loss='mean_squared_error')
```

```
hist = model.fit(x_train, y_train,
                 batch_size=32, epochs=100,
                 validation_split = 0.2)
```

```
yhat = model.predict(x_test)
```

```
Epoch 1/100
10/10 [=====] - 0s 17ms/step - loss: 230163415040.0000 - 
Epoch 2/100
10/10 [=====] - 0s 6ms/step - loss: 223731341125.8182 - 
Epoch 3/100
10/10 [=====] - 0s 5ms/step - loss: 241088126231.2727 - 
Epoch 4/100
10/10 [=====] - 0s 5ms/step - loss: 221636322024.7273 - 
Epoch 5/100
10/10 [=====] - 0s 5ms/step - loss: 239976592477.0909 - 
Epoch 6/100
10/10 [=====] - 0s 5ms/step - loss: 223983724730.1818 - 
Epoch 7/100
10/10 [=====] - 0s 5ms/step - loss: 231876347345.4546 - 
Epoch 8/100
10/10 [=====] - 0s 5ms/step - loss: 228932559592.7273 - 
Epoch 9/100
10/10 [=====] - 0s 6ms/step - loss: 222435454603.6364 - 
Epoch 10/100
10/10 [=====] - 0s 6ms/step - loss: 234665453754.1818 - 
Epoch 11/100
10/10 [=====] - 0s 6ms/step - loss: 219560274106.1818 - 
Epoch 12/100
10/10 [=====] - 0s 6ms/step - loss: 243256718429.0909 - 
Epoch 13/100
10/10 [=====] - 0s 5ms/step - loss: 224747785122.9091 - 
Epoch 14/100
10/10 [=====] - 0s 5ms/step - loss: 233335455744.0000 - 
Epoch 15/100
10/10 [=====] - 0s 5ms/step - loss: 230568959255.2727 - 
Epoch 16/100
```