Prediction of House Price

Yuxiang (Kevin) Hu, July 2020

Executive Summary

The analysis is conducted on a given dataset of New Zealand houses' capital value. The dataset contains details of house composition and its environment including deprive index and population and location of suburb and statistical area 1

The analysis is based on 1043 distinct observations for each of the 15 numerical variables out of 17 variables collected. The fifth variable is capital value, abbr. CV which is the response variables with integer values.

The rest of the variables are explanatory variables and each describes a measurement or feature of the house. The eighth variable is ID for SA1 (statistical area 1). Column 9 to 14 represents numbers of occupents for each year group in its SA1. The sixteenth (second last) variable is deprive index for its SA1. All population and deprive index are based on 2018 census. The names of the rest variables are self-explanatory.

After exploration and brief calculation on statistics of data, and by creating visualization of each and the correlation between each numerical variables, three moderate positively correlated variables are found. Two linear regression models are tested for this dataset the the best model has been chosen on coefficient of determination

Initial Data Exploration

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
dataset = pd.read_csv('prepared data.csv')
```

In [3]:

```
# drop index row which was created by to_csv in preparation of data
dataset = dataset.drop('Unnamed: 0', axis=1)
```

In [4]:

```
dataset.loc[dataset.isnull().any(axis=1)]
```

Out[4]:

	Bedrooms	Bathrooms	Address	Land area	CV	Latitude	Longitude	SA1	0-19 years	20-29 years	30-39 years

In [5]:

```
dataset.dtypes
```

Out[5]:

```
Bedrooms          int64
Bathrooms         int64
Address          object
Land area         int64
CV                int64
Latitude        float64
Longitude       float64
SA1               int64
0-19 years        int64
20-29 years       int64
30-39 years       int64
40-49 years       int64
50-59 years       int64
60+ years         int64
Suburbs          object
NZDep2018       float64
Population18      int64
dtype: object
```

In [6]:

```
dataset.head()
```

Out[6]:

	Bedrooms	Bathrooms	Address	Land area	CV	Latitude	Longitude	SA1	0-1 yea
0	5	3	106 Lawrence Crescent Hill Park, Auckland	714	960000	-37.012920	174.904069	7009770	4
1	5	3	8 Corsica Way Karaka, Auckland	564	1250000	-37.063672	174.922912	7009991	4
2	6	4	243 Harbourside Drive Karaka, Auckland	626	1250000	-37.063580	174.924044	7009991	4
3	2	1	2/30 Hardington Street Onehunga, Auckland	65	740000	-36.912996	174.787425	7007871	4
4	3	1	59 Israel Avenue Clover Park, Auckland	601	630000	-36.979037	174.892612	7008902	9

In [7]:

```
dataset.shape
```

Out[7]:

(1043, 17)

All data are present.

There is no null or duplicate values after preparation of data in DataCollection.ipynb.

Data has correct data types.

The initial exploration of the data began with summary and descriptive statistics. Individual Feature Statistics Summary statistics are recorded in the following table for 1043 observations in 2018.

In [8]:

```
dataset.describe()
```

Out[8]:

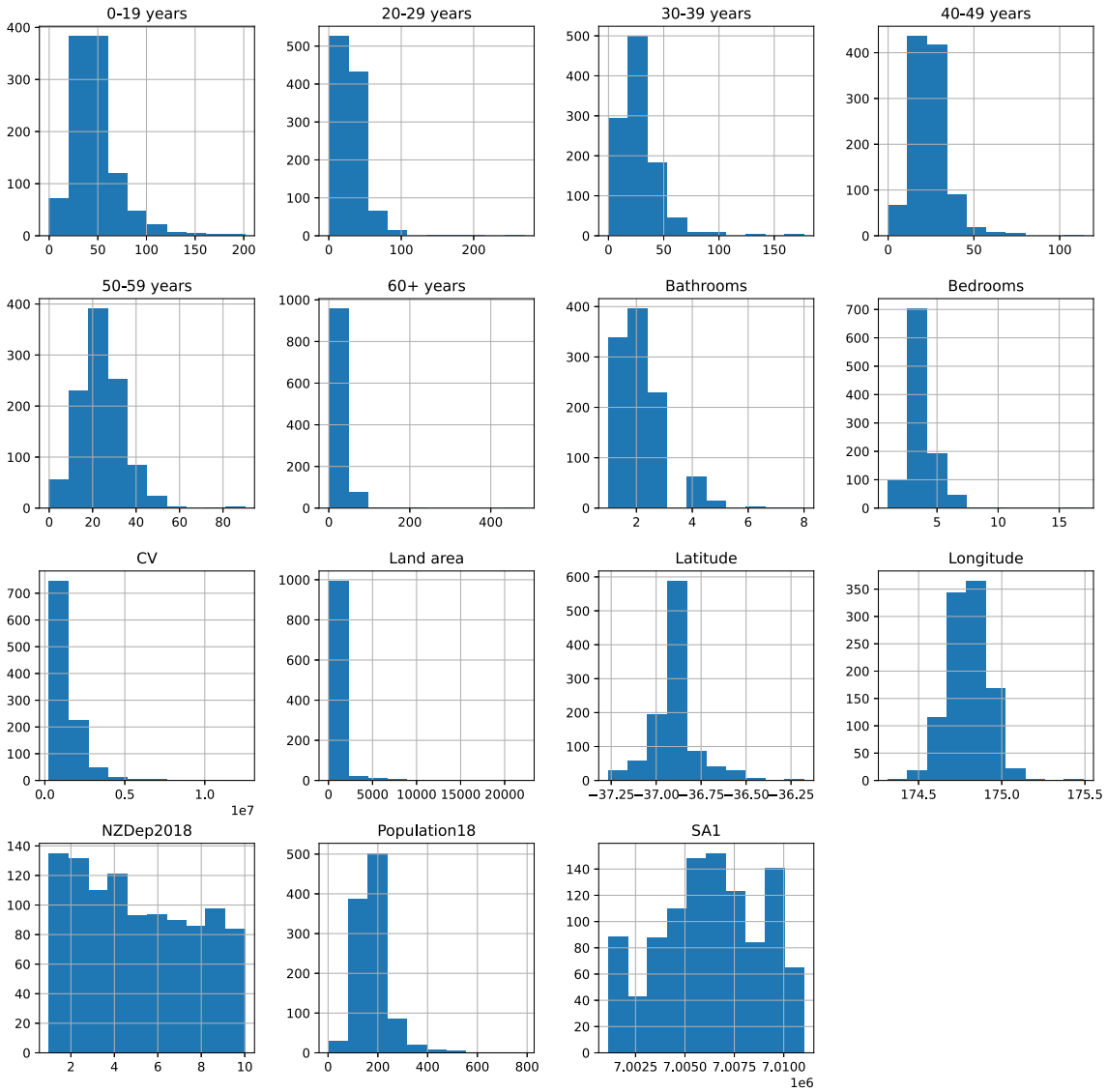
	Bedrooms	Bathrooms	Land area	CV	Latitude	Longitude	
count	1043.000000	1043.000000	1043.000000	1.043000e+03	1043.000000	1043.000000	1.0
mean	3.780441	2.072867	850.817833	1.365014e+06	-36.893368	174.799023	7.0
std	1.172592	0.993483	1579.533876	1.042246e+06	0.130153	0.119779	2.5
min	1.000000	1.000000	40.000000	2.700000e+05	-37.265021	174.317078	7.0
25%	3.000000	1.000000	321.000000	7.800000e+05	-36.950183	174.719666	7.0
50%	4.000000	2.000000	570.000000	1.080000e+06	-36.893368	174.797892	7.0
75%	4.000000	3.000000	825.000000	1.600000e+06	-36.855643	174.880944	7.0
max	17.000000	8.000000	22240.000000	1.250000e+07	-36.177655	175.492424	7.0

In [10]:

```
dataset.hist(figsize=(15,15))
```

Out[10]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x23CD64A8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23D05EB0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23D288C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23D492C8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x23D15B38>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23D76610>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23D76DD8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23D95808>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x23DC9B98>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23DEB598>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23E00F70>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23E1F970>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x23E3E370>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23E51D48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23E73748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x23E950E8>]],
      dtype=object)
```

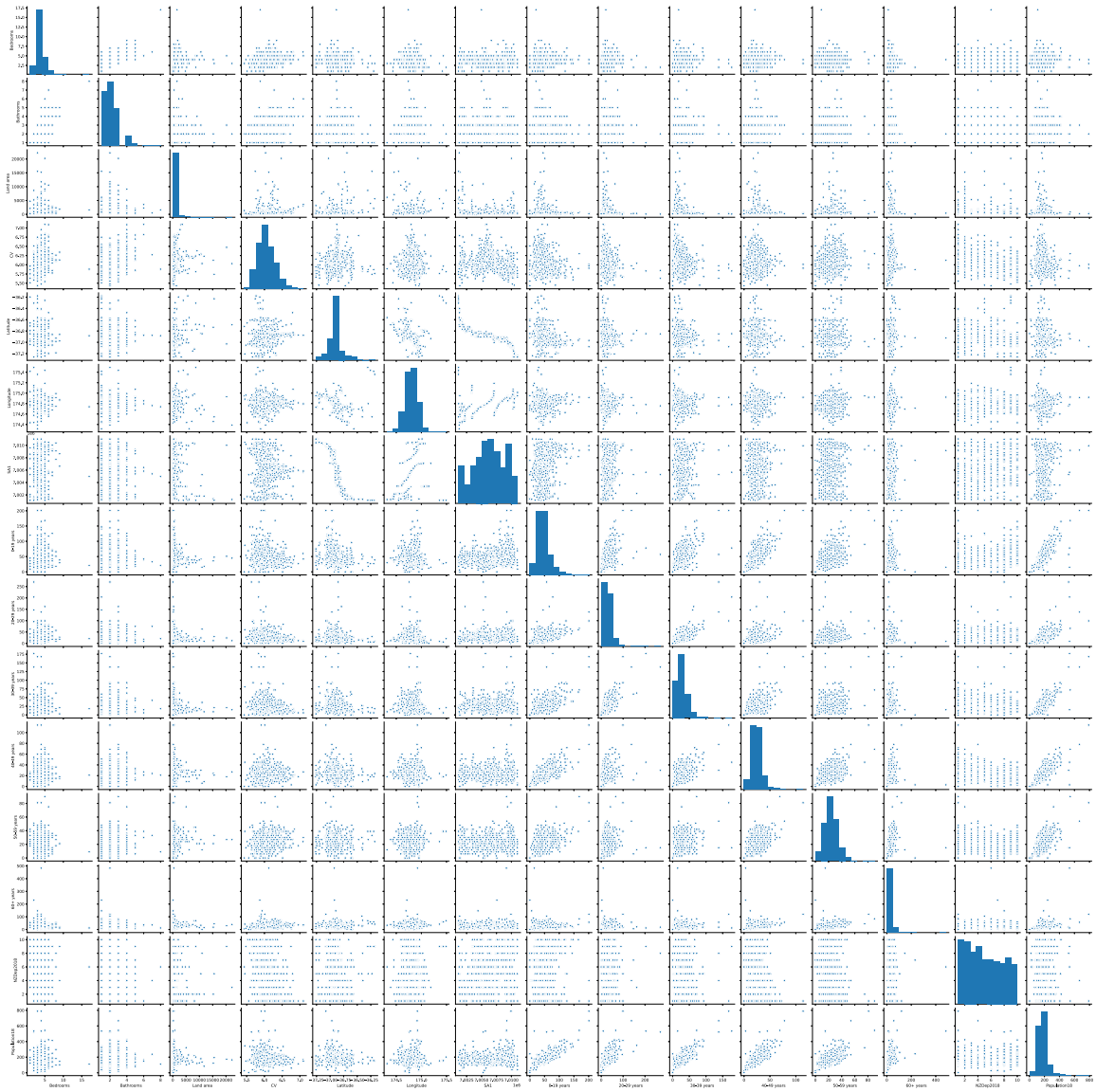


In [37]:

```
sns.pairplot(dataset)
```

Out[37]:

<seaborn.axisgrid.PairGrid at 0x27820130>



Correlation and Relationships

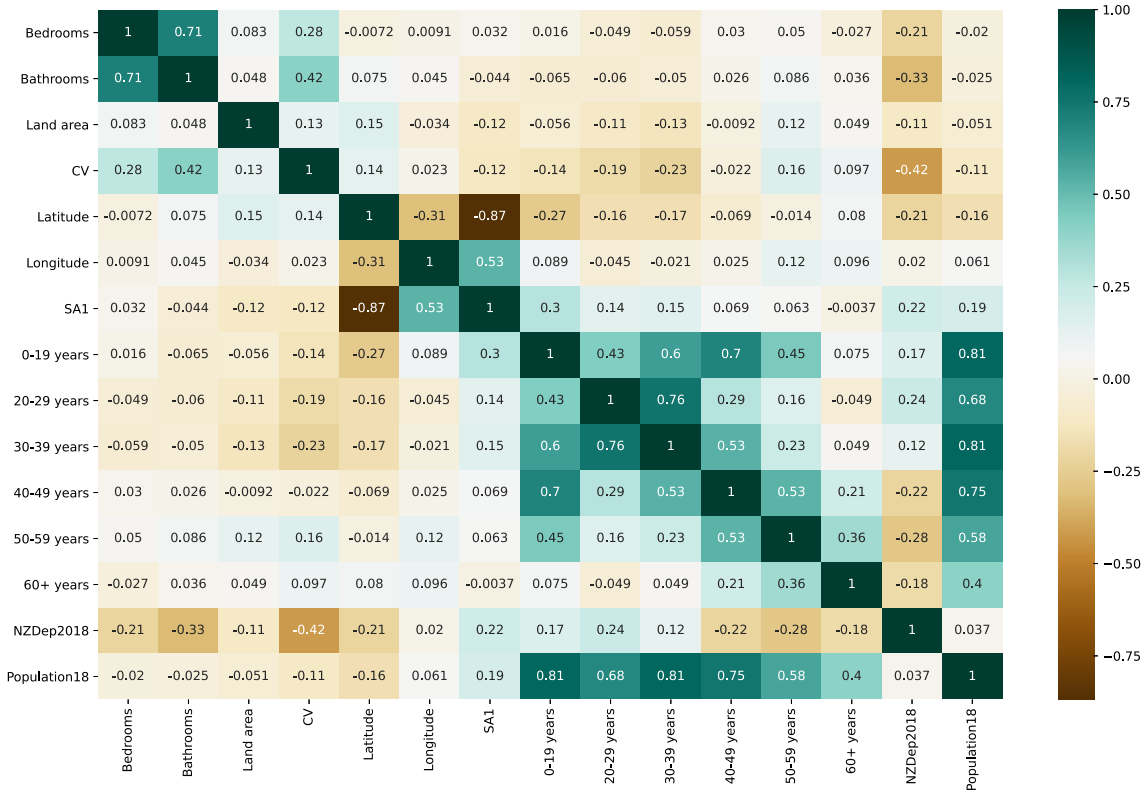
A table of coefficient of determination is generated as below where dark green means correlation value of 1 and brown means correlation value of -1.

In [12]:

```
fig, ax = plt.subplots(figsize=(16,10))
correlation_matrix = dataset.corr()
sns.heatmap(correlation_matrix, annot=True,cmap="BrBG")
```

Out[12]:

<matplotlib.axes._subplots.AxesSubplot at 0x251310b8>

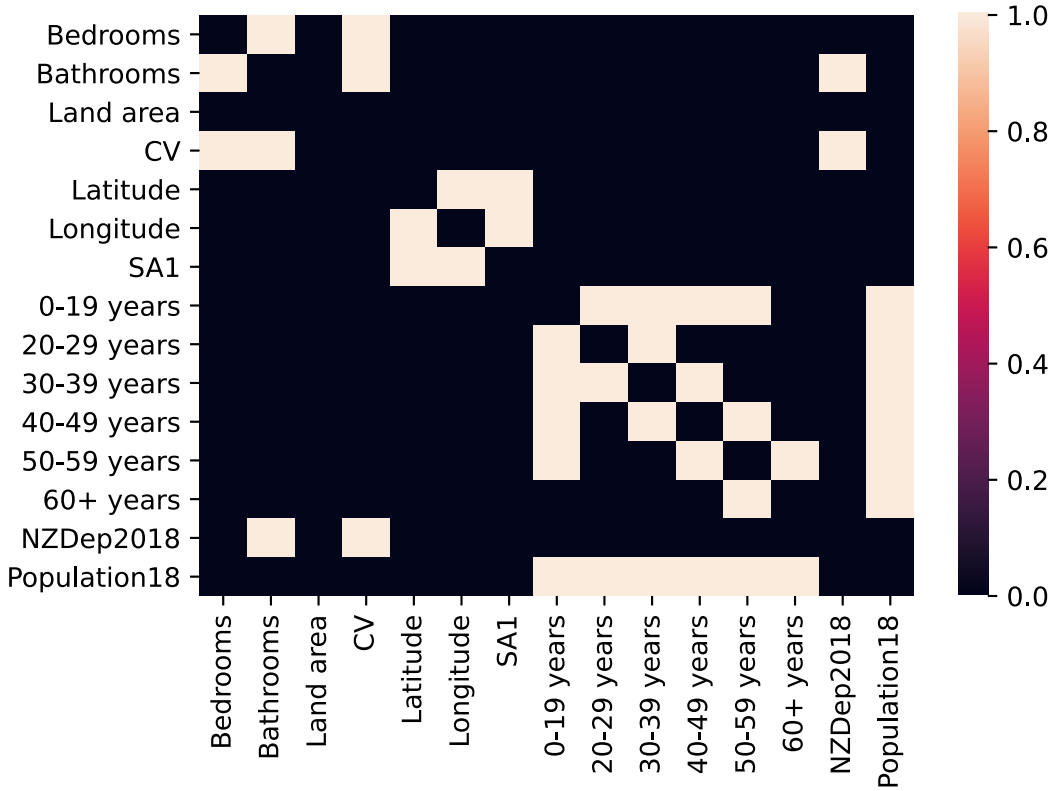


In [38]:

```
# display all block with moderate correlation that isn't with itself
sns.heatmap(correlation_matrix.apply(lambda x: (abs(x) > 0.3) & (x != 1) ))
```

Out[38]:

<matplotlib.axes._subplots.AxesSubplot at 0x2da6cd78>



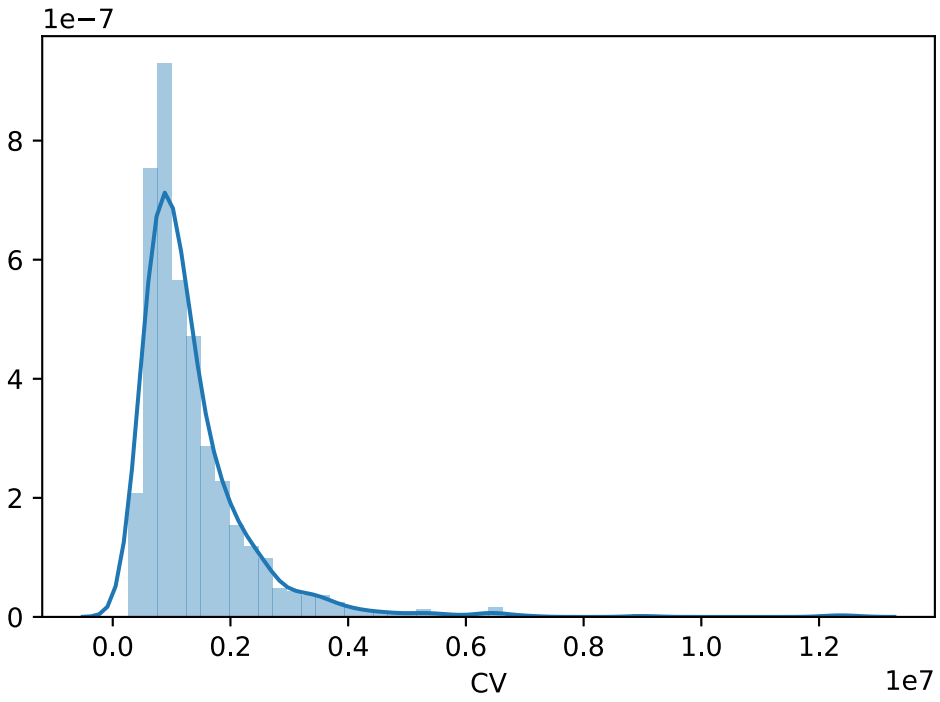
It can be seen from either of the graphs above, that there is strong correlation between total population in each SA1 and individual age group and in between different age groups. There is also strong correlation between SA1 ID and lattitude. The capital value shows moderate positive correlation with number of bedrooms and bathrooms and deprive index.

In [14]:

```
sns.distplot(dataset['CV'])
# CV values looks to be right skewed
```

Out[14]:

<matplotlib.axes._subplots.AxesSubplot at 0x25d3d2f8>



The variable CV looks right skewed to a small extent.

Model fitted

Direct linear regression model

A direct linear regression model will be fitted first. String variables including address and suburbs cannot be handled by a linear regression model and will be removed.

In [15]:

```
from sklearn.model_selection import train_test_split
x = dataset.drop(['CV', 'Address', 'Suburbs',], axis=1)
x.head()
```

Out[15]:

	Bedrooms	Bathrooms	Land area	Latitude	Longitude	SA1	0-19 years	20-29 years	30-39 years	40-49 years
0	5	3	714	-37.012920	174.904069	7009770	48	27	24	21
1	5	3	564	-37.063672	174.922912	7009991	42	18	12	21
2	6	4	626	-37.063580	174.924044	7009991	42	18	12	21
3	2	1	65	-36.912996	174.787425	7007871	42	6	21	21
4	3	1	601	-36.979037	174.892612	7008902	93	27	33	30

In [16]:

```
y = dataset['CV']
y.head()
```

Out[16]:

```
0     960000
1    1250000
2    1250000
3     740000
4     630000
Name: CV, dtype: int64
```

In [17]:

```
train_x, test_x, train_y, test_y = train_test_split(x,y, test_size = 0.3, random_state = 42)
```

In [18]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

In [19]:

```
model.fit(train_x, train_y)
```

Out[19]:

```
LinearRegression()
```

In [20]:

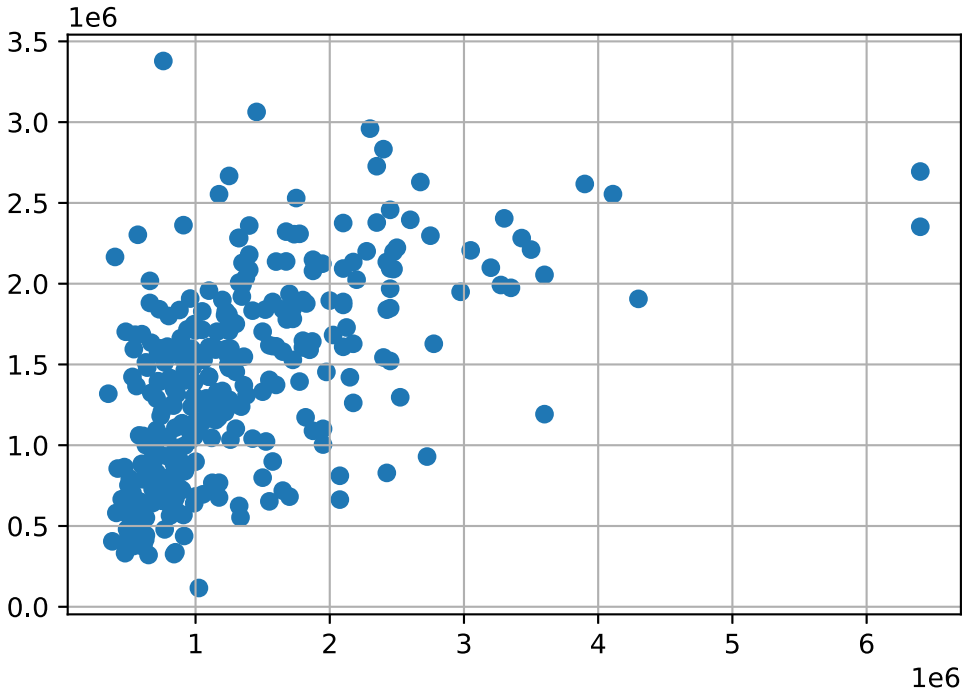
```
list(zip(x.columns,model.coef_))
```

Out[20]:

```
[('Bedrooms', 6632.003592908645),
 ('Bathrooms', 353632.8984720462),
 ('Land area', 28.01586721505737),
 ('Latitude', -18026.0525885713),
 ('Longitude', 96696.83874232024),
 ('SA1', -13.098936293099541),
 ('0-19 years', 7881.026225626225),
 ('20-29 years', 9225.602139826147),
 ('30-39 years', -11650.60595966825),
 ('40-49 years', -3741.676429879403),
 ('50-59 years', 13121.67351534961),
 ('60+ years', 4515.703404461472),
 ('NZDep2018', -111985.73896780939),
 ('Population18', -4448.72178165781)]
```

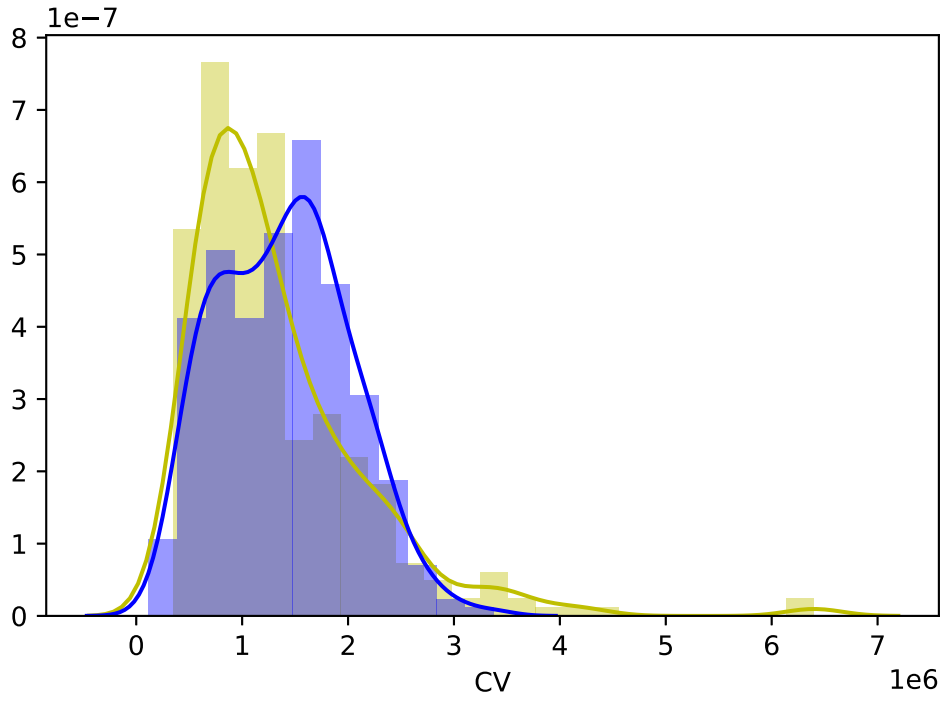
In [21]:

```
predicted = model.predict(test_x)
plt.grid(True)
plt.scatter(test_y,predicted,)
plt.show()
```



In [22]:

```
# yellow is tested, blue is predidcted
fig,ax = plt.subplots()
sns.distplot(test_y,ax=ax,color="y")
sns.distplot(predicted,ax=ax, color='b')
plt.show()
```



In [23]:

```
model.score(test_x,test_y)
```

Out[23]:

0.2781075021344699

Direct linear regression model explains 28% of variation in the data.

The model tends to predict the majority of houses to be more expensive and might not be able to predict expensive houses as there is no prediction of capital value above $400000 in the test data set (30% of full data set). The model may predict capital value close to or below zero when a large number of predictions is required.

Linear regression model with transformed response variable

The value of response variable Capital Value will be transformed by log base 10 for model fitting purposes and would be transformed back for interpretation.

In [24]:

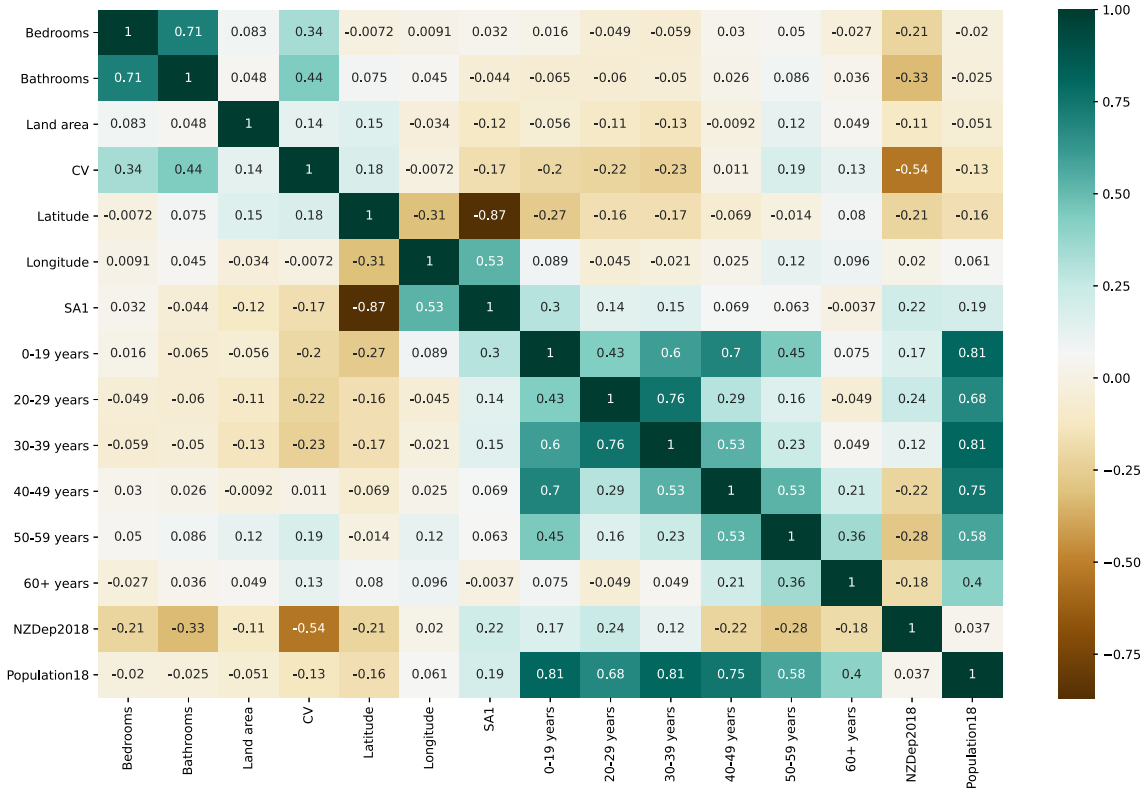
```
import math
dataset['CV'] = dataset['CV'].apply(math.log10)
```

In [25]:

```
fig, ax = plt.subplots(figsize=(16,10))
correlation_matrix = dataset.corr()
sns.heatmap(correlation_matrix, annot=True,cmap="BrBG")
```

Out[25]:

<matplotlib.axes._subplots.AxesSubplot at 0x2779ace8>

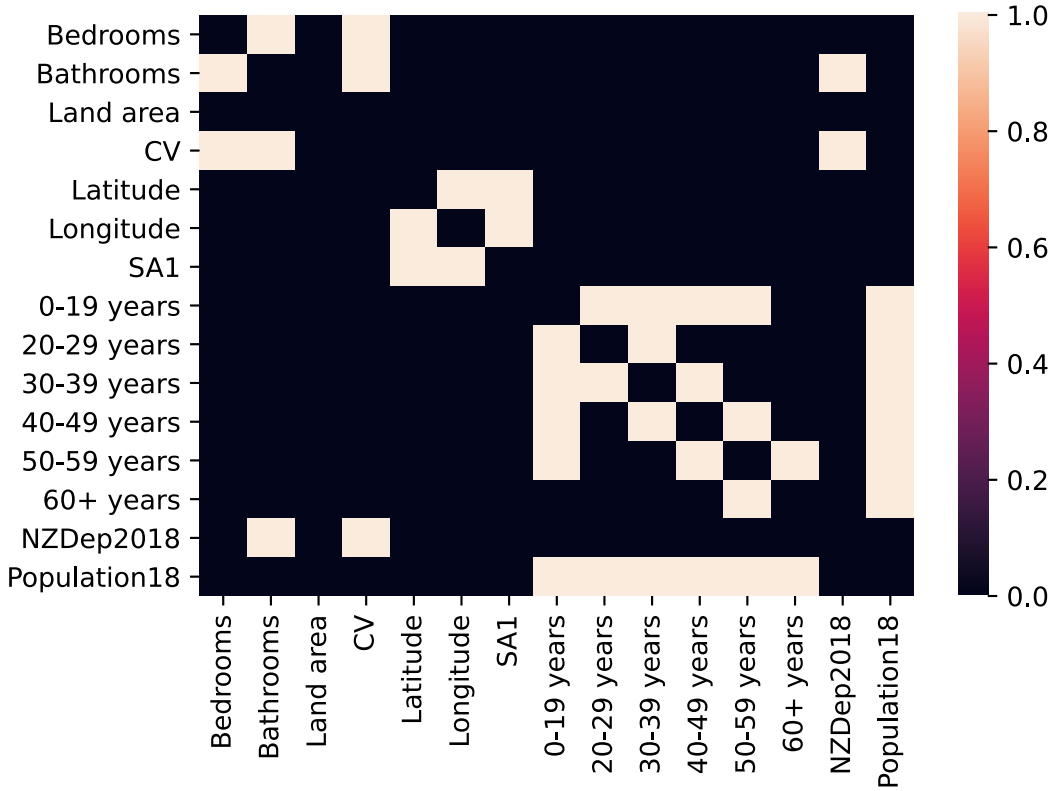


In [39]:

```
# display all block with strong correlation that isn't with itself
sns.heatmap(correlation_matrix.apply(lambda x: (abs(x) > 0.3) & (x != 1) ))
```

Out[39]:

<matplotlib.axes._subplots.AxesSubplot at 0x2dcbf2e0>

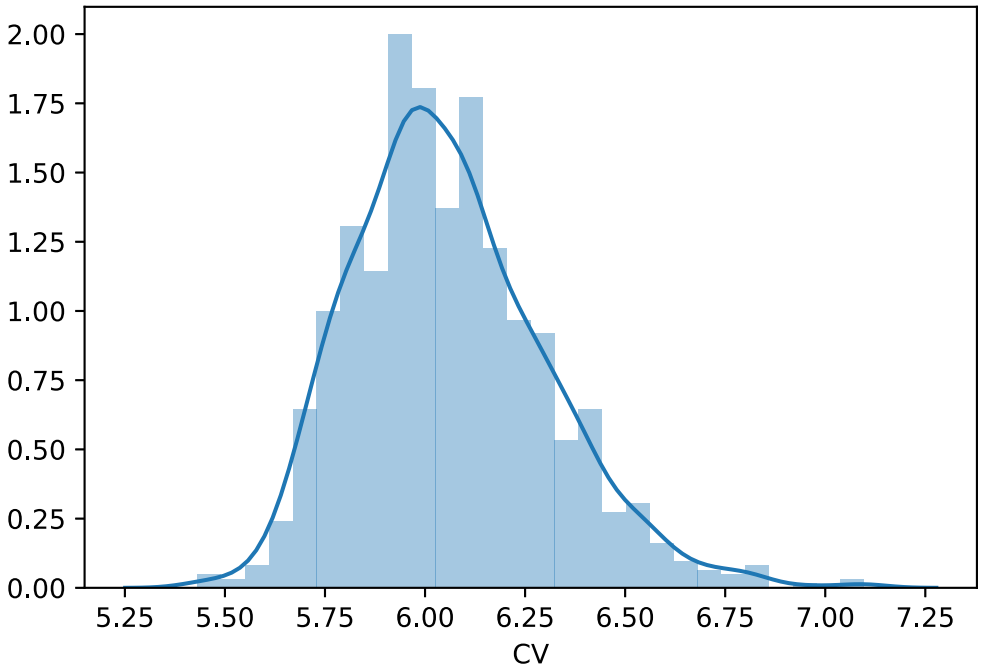


In [27]:

```
sns.distplot(dataset['CV'])
```

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2775da48>
```



Only the same three variables including number of bedrooms and bathrooms and land area show moderate positive correlation with capital value although all three coefficients of determination increases.

The capital value after transformation is approximately normal.

In [28]:

```
x = dataset.drop(['CV', 'Address', 'Suburbs'], axis=1)
x.head()
```

Out[28]:

	Bedrooms	Bathrooms	Land area	Latitude	Longitude	SA1	0-19 years	20-29 years	30-39 years	40-4 years
0	5	3	714	-37.012920	174.904069	7009770	48	27	24	2
1	5	3	564	-37.063672	174.922912	7009991	42	18	12	2
2	6	4	626	-37.063580	174.924044	7009991	42	18	12	2
3	2	1	65	-36.912996	174.787425	7007871	42	6	21	2
4	3	1	601	-36.979037	174.892612	7008902	93	27	33	3

In [29]:

```
y = dataset['CV']
y.head()
```

Out[29]:

```
0    5.982271
1    6.096910
2    6.096910
3    5.869232
4    5.799341
Name: CV, dtype: float64
```

In [30]:

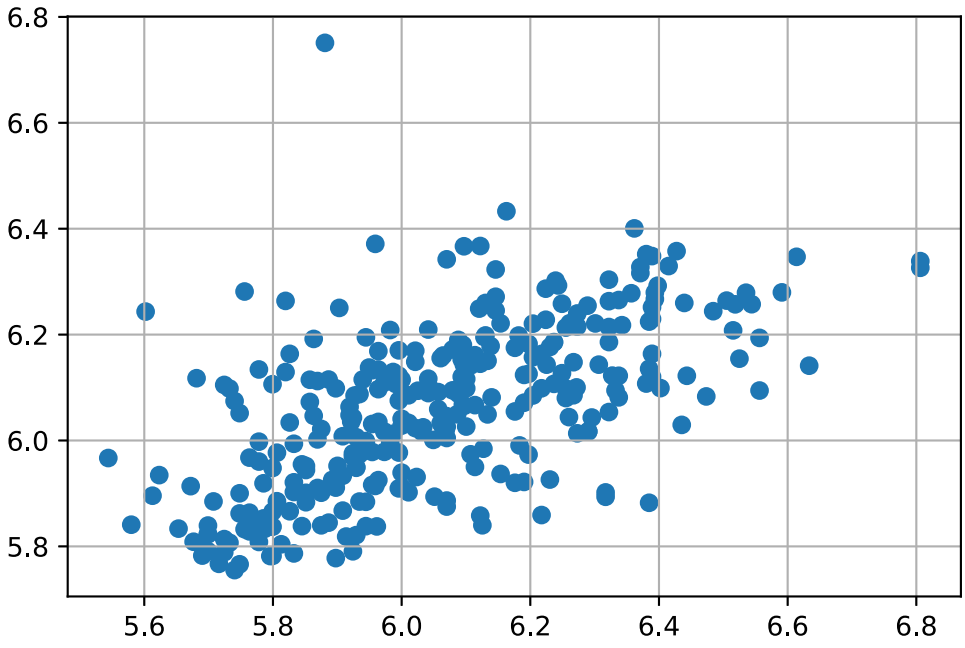
```
train_x, test_x, train_y, test_y = train_test_split(x,y, test_size = 0.3, random_state = 42)
model = LinearRegression()
model.fit(train_x,train_y)
list(zip(x.columns,model.coef_))
```

Out[30]:

```
[('Bedrooms', 0.03304194437260333),
 ('Bathrooms', 0.04842683485821104),
 ('Land area', 2.7145117694853116e-06),
 ('Latitude', -0.039339169655350274),
 ('Longitude', 0.0177742776548647),
 ('SA1', -5.475941329179113e-06),
 ('0-19 years', 0.00026645505141770594),
 ('20-29 years', 0.0017996220488152351),
 ('30-39 years', -0.0018749226921119713),
 ('40-49 years', 0.0009283196814485348),
 ('50-59 years', 0.003748703070089749),
 ('60+ years', 0.0011164577471639574),
 ('NZDep2018', -0.03117360192180079),
 ('Population18', -0.000957784571491653)]
```

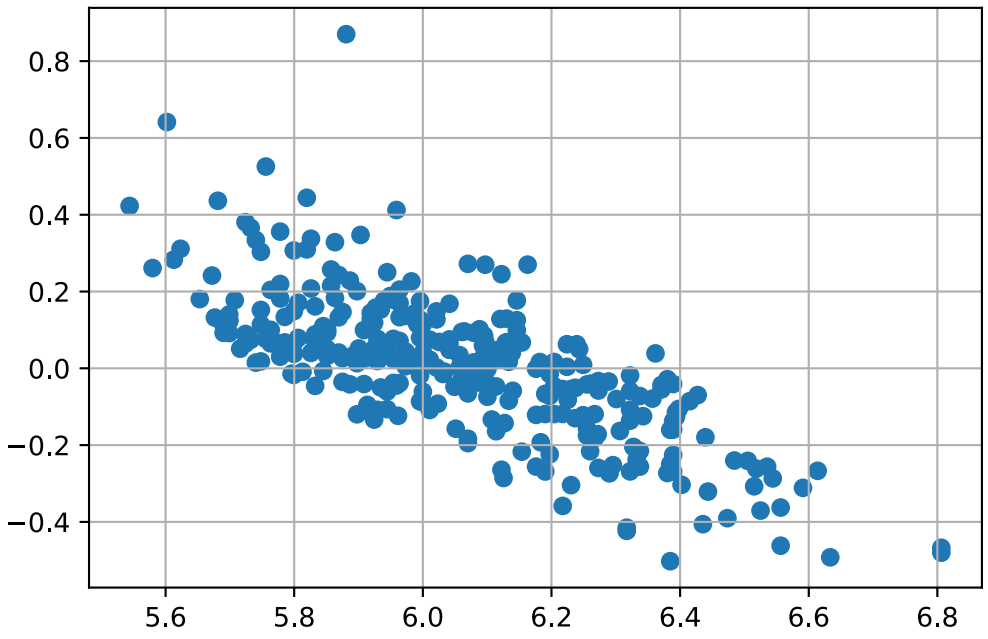
In [31]:

```
predicted = model.predict(test_x)
plt.grid(True)
plt.scatter(test_y,predicted)
plt.show()
```



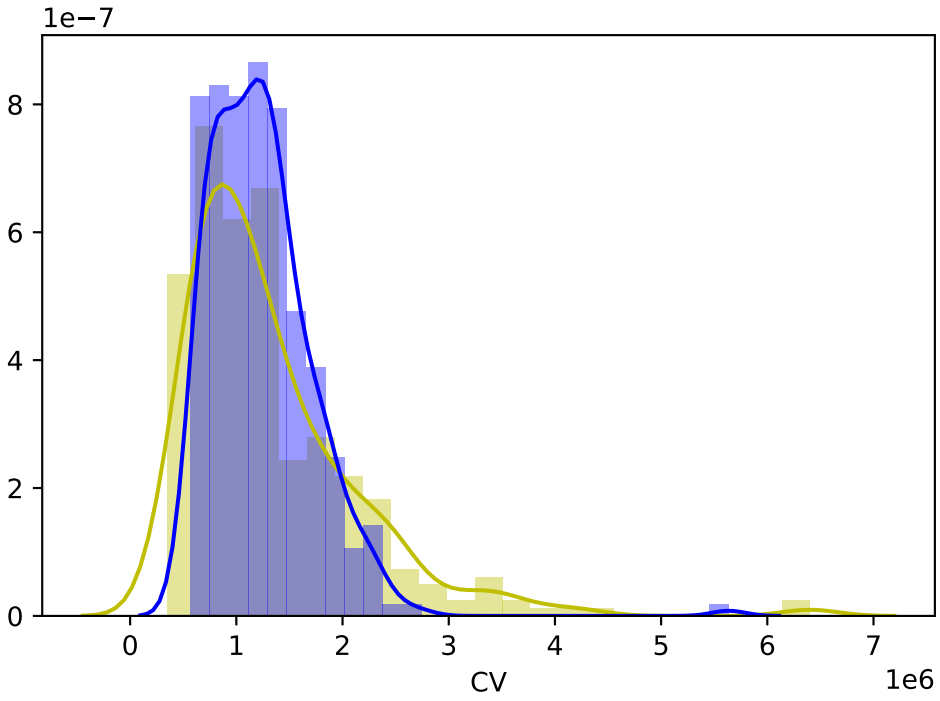
In [36]:

```
# show difference against true values across the range of CV
# the x-axis is in million of dollars worth of capital value of true record
# the y-axis is the difference between true and predicted capital value in log base 10 scale.
predicted = model.predict(test_x)
plt.grid(True)
plt.scatter(test_y,predicted-test_y)
plt.show()
```



In [33]:

```
# yellow is tested, blue is predidcted
fig,ax = plt.subplots()
sns.distplot(10**test_y,ax=ax,color="y")
sns.distplot(10**predicted,ax=ax, color='b')
plt.show()
```



In [34]:

```
model.score(test_x,test_y)
```

Out[34]:

0.3626769859406884

The linear regression model with transformed response variable explains 36% of variation in the data.

The best fit curve demonstrate a reasonable range and a right-skewed shape. A value very close to or below zero is unlikely to occur in future predictions and the model is able to predict expensive models to a certain extent but is more likely to predict a value much closer to median.

The prediction has less spread than expected.

Conclusion

The linear regression model with transformed response variable is the best model of the two already analysised. It explains 36% of variation in the test data.