```
import pandas as pd

df = pd.read_csv('melb_data.csv')
```

```
C:\Users\woosh\AppData\Local\Temp\ipykernel_10640\3565767594.py:1:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major
release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type,
and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at
https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

```
df
```

```
               Suburb           Address  Rooms Type      Price
Method  \
0          Abbotsford      85 Turner St      2    h  1480000.0
S

1          Abbotsford   25 Bloomburg St      2    h  1035000.0
S

2          Abbotsford      5 Charles St      3    h  1465000.0
SP

3          Abbotsford  40 Federation La      3    h   850000.0
PI

4          Abbotsford       55a Park St      4    h  1600000.0
VB

...               ...               ...    ...  ...        ...
...

13575  Wheelers Hill      12 Strada Cr      4    h  1245000.0
S

13576   Williamstown     77 Merrett Dr      3    h  1031000.0
SP

13577   Williamstown       83 Power St      3    h  1170000.0
S

13578   Williamstown      96 Verdon St      4    h  2500000.0
PI

13579     Yarraville        6 Agnes St      4    h  1285000.0
SP


        SellerG        Date  Distance  Postcode  ...  Bathroom  Car
Landsize  \
0        Biggin   3/12/2016       2.5    3067.0  ...       1.0  1.0
202.0
1        Biggin   4/02/2016       2.5    3067.0  ...       1.0  0.0
156.0
2        Biggin   4/03/2017       2.5    3067.0  ...       2.0  0.0
```

```
134.0
3        Biggin   4/03/2017       2.5    3067.0  ...       2.0  1.0
94.0
4        Nelson   4/06/2016       2.5    3067.0  ...       1.0  2.0
120.0
...         ...         ...       ...       ...  ...       ...  ...
...
13575     Barry  26/08/2017      16.7    3150.0  ...       2.0  2.0
652.0
13576  Williams  26/08/2017       6.8    3016.0  ...       2.0  2.0
333.0
13577     Raine  26/08/2017       6.8    3016.0  ...       2.0  4.0
436.0
13578   Sweeney  26/08/2017       6.8    3016.0  ...       1.0  5.0
866.0
13579   Village  26/08/2017       6.3    3013.0  ...       1.0  1.0
362.0

       BuildingArea  YearBuilt  CouncilArea Lattitude  Longtitude  \
0               NaN        NaN        Yarra -37.79960   144.99840
1              79.0     1900.0        Yarra -37.80790   144.99340
2             150.0     1900.0        Yarra -37.80930   144.99440
3               NaN        NaN        Yarra -37.79690   144.99690
4             142.0     2014.0        Yarra -37.80720   144.99410
...             ...        ...          ...       ...         ...
13575           NaN     1981.0          NaN -37.90562   145.16761
13576         133.0     1995.0          NaN -37.85927   144.87904
13577           NaN     1997.0          NaN -37.85274   144.88738
13578         157.0     1920.0          NaN -37.85908   144.89299
13579         112.0     1920.0          NaN -37.81188   144.88449

                     Regionname Propertycount
0           Northern Metropolitan        4019.0
1           Northern Metropolitan        4019.0
2           Northern Metropolitan        4019.0
3           Northern Metropolitan        4019.0
4           Northern Metropolitan        4019.0
...                           ...           ...
13575  South-Eastern Metropolitan        7392.0
13576        Western Metropolitan        6380.0
13577        Western Metropolitan        6380.0
13578        Western Metropolitan        6380.0
13579        Western Metropolitan        6543.0

[13580 rows x 21 columns]
```

# Display info (size,shape,number of dimensions and overview information)

```
print("size: ",df.size)
print("shape: ",df.shape)
print("number of dimensions: ",df.ndim)
print(df.info())
```

```
size:  285180
shape:  (13580, 21)
number of dimensions:  2
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13580 entries, 0 to 13579
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Suburb         13580 non-null  object 
 1   Address        13580 non-null  object 
 2   Rooms          13580 non-null  int64  
 3   Type           13580 non-null  object 
 4   Price          13580 non-null  float64
 5   Method         13580 non-null  object 
 6   SellerG        13580 non-null  object 
 7   Date           13580 non-null  object 
 8   Distance       13580 non-null  float64
 9   Postcode       13580 non-null  float64
 10  Bedroom2       13580 non-null  float64
 11  Bathroom       13580 non-null  float64
 12  Car            13518 non-null  float64
 13  Landsize       13580 non-null  float64
 14  BuildingArea   7130 non-null   float64
 15  YearBuilt      8205 non-null   float64
 16  CouncilArea    12211 non-null  object 
 17  Lattitude      13580 non-null  float64
 18  Longtitude     13580 non-null  float64
 19  Regionname     13580 non-null  object 
 20  Propertycount  13580 non-null  float64
dtypes: float64(12), int64(1), object(8)
memory usage: 2.2+ MB
None
```

# Display Statistics (min, max, average, S.D.)

## all attributes

I don't sure if this is what you mean to calculate the min, max, average, and SD for the df. But, I choose to do only numbers because it sounds sensible to me.

```
print(df.select_dtypes(include=['number']).min(),"\n")
print(df.select_dtypes(include=['number']).max(),"\n")
print(df.select_dtypes(include=['number']).mean(),"\n")
print(df.select_dtypes(include=['number']).std())
```

```
Rooms                1.00000
Price            85000.00000
Distance             0.00000
Postcode          3000.00000
Bedroom2             0.00000
Bathroom             0.00000
Car                  0.00000
Landsize             0.00000
BuildingArea         0.00000
YearBuilt         1196.00000
Lattitude          -38.18255
Longtitude         144.43181
Propertycount      249.00000
dtype: float64

Rooms             1.000000e+01
Price             9.000000e+06
Distance          4.810000e+01
Postcode          3.977000e+03
Bedroom2          2.000000e+01
Bathroom          8.000000e+00
Car               1.000000e+01
Landsize          4.330140e+05
BuildingArea      4.451500e+04
YearBuilt         2.018000e+03
Lattitude        -3.740853e+01
Longtitude        1.455264e+02
Propertycount     2.165000e+04
dtype: float64

Rooms             2.937997e+00
Price             1.075684e+06
Distance          1.013778e+01
Postcode          3.105302e+03
Bedroom2          2.914728e+00
Bathroom          1.534242e+00
Car               1.610075e+00
Landsize          5.584161e+02
BuildingArea      1.519676e+02
YearBuilt         1.964684e+03
Lattitude        -3.780920e+01
Longtitude        1.449952e+02
Propertycount     7.454417e+03
dtype: float64
```

```
Rooms                  0.955748
Price             639310.724296
Distance               5.868725
Postcode              90.676964
Bedroom2               0.965921
Bathroom               0.691712
Car                    0.962634
Landsize            3990.669241
BuildingArea         541.014538
YearBuilt             37.273762
Lattitude              0.079260
Longtitude             0.103916
Propertycount       4378.581772
dtype: float64
```

## Selected Attributes: Price, Landsize, Propertycount

```
df[['Price','Landsize','Propertycount']].min()
```

```
Price             85000.0
Landsize              0.0
Propertycount       249.0
dtype: float64
```

## Selected attributes with a specific condition:

```
Landsize < 500 and Bedroom = 2 and Bathroom =1 and Car = 1
```

```
df[(df['Landsize']<500)]
```

```
             Suburb           Address  Rooms Type      Price Method
SellerG  \
0         Abbotsford      85 Turner St      2    h  1480000.0      S
Biggin
1         Abbotsford   25 Bloomburg St      2    h  1035000.0      S
Biggin
2         Abbotsford      5 Charles St      3    h  1465000.0     SP
Biggin
3         Abbotsford  40 Federation La      3    h   850000.0     PI
Biggin
4         Abbotsford       55a Park St      4    h  1600000.0     VB
Nelson
...              ...               ...    ...  ...        ...    ...
...
13572       Watsonia     76 Kenmare St      2    h   650000.0     PI
Morrison
13574   Westmeadows        9 Black St      3    h   582000.0      S
Red
```

```
13576  Williamstown     77 Merrett Dr      3    h  1031000.0     SP
Williams
13577  Williamstown       83 Power St      3    h  1170000.0      S
Raine
13579    Yarraville        6 Agnes St      4    h  1285000.0     SP
Village

             Date  Distance  Postcode  ...  Bathroom  Car  Landsize  \
0       3/12/2016       2.5    3067.0  ...       1.0  1.0     202.0
1       4/02/2016       2.5    3067.0  ...       1.0  0.0     156.0
2       4/03/2017       2.5    3067.0  ...       2.0  0.0     134.0
3       4/03/2017       2.5    3067.0  ...       2.0  1.0      94.0
4       4/06/2016       2.5    3067.0  ...       1.0  2.0     120.0
...           ...       ...       ...  ...       ...  ...       ...
13572  26/08/2017      14.5    3087.0  ...       1.0  1.0     210.0
13574  26/08/2017      16.5    3049.0  ...       2.0  2.0     256.0
13576  26/08/2017       6.8    3016.0  ...       2.0  2.0     333.0
13577  26/08/2017       6.8    3016.0  ...       2.0  4.0     436.0
13579  26/08/2017       6.3    3013.0  ...       1.0  1.0     362.0

       BuildingArea  YearBuilt  CouncilArea Lattitude  Longtitude  \
0               NaN        NaN        Yarra -37.79960   144.99840
1              79.0     1900.0        Yarra -37.80790   144.99340
2             150.0     1900.0        Yarra -37.80930   144.99440
3               NaN        NaN        Yarra -37.79690   144.99690
4             142.0     2014.0        Yarra -37.80720   144.99410
...             ...        ...          ...       ...         ...
13572          79.0     2006.0          NaN -37.70657   145.07878
13574           NaN        NaN          NaN -37.67917   144.89390
13576         133.0     1995.0          NaN -37.85927   144.87904
13577           NaN     1997.0          NaN -37.85274   144.88738
13579         112.0     1920.0          NaN -37.81188   144.88449

                  Regionname Propertycount
0      Northern Metropolitan        4019.0
1      Northern Metropolitan        4019.0
2      Northern Metropolitan        4019.0
3      Northern Metropolitan        4019.0
4      Northern Metropolitan        4019.0
...                      ...           ...
13572  Northern Metropolitan        2329.0
13574  Northern Metropolitan        2474.0
13576   Western Metropolitan        6380.0
13577   Western Metropolitan        6380.0
13579   Western Metropolitan        6543.0

[7392 rows x 21 columns]

df[(df['Bedroom2']==2) & (df['Bathroom']==1) & (df['Car']==1)]
```

```
             Suburb              Address  Rooms Type      Price Method
\
0         Abbotsford        85 Turner St      2    h  1480000.0      S
13        Abbotsford       45 William St      2    h  1172500.0      S
17        Abbotsford         78 Yarra St      3    h  1176500.0      S
19        Abbotsford       42 Valiant St      2    h   890000.0      S
23        Abbotsford  6/219 Nicholson St      2    u   500000.0      S
...              ...                 ...    ...  ...        ...    ...
13482  Malvern East     2002 Malvern Rd      2    u   651000.0     SP
13495  Moonee Ponds     1/53 Buckley St      2    u   435000.0      S
13510    Nunawading       3/39 Lemon Gr      2    u   710000.0      S
13511      Oak Park        18 Jessie St      2    h  1006000.0      S
13572      Watsonia       76 Kenmare St      2    h   650000.0     PI

          SellerG        Date  Distance  Postcode  ...  Bathroom
Car  \
0          Biggin   3/12/2016       2.5    3067.0  ...       1.0  1.0
13         Biggin  13/08/2016       2.5    3067.0  ...       1.0  1.0
17         LITTLE  16/07/2016       2.5    3067.0  ...       1.0  1.0
19         Biggin  17/09/2016       2.5    3067.0  ...       1.0  1.0
23        Collins  18/06/2016       2.5    3067.0  ...       1.0  1.0
...           ...         ...       ...       ...  ...       ...  ...
13482      Jellis  26/08/2017       8.4    3145.0  ...       1.0  1.0
13495      Nelson  26/08/2017       6.2    3039.0  ...       1.0  1.0
13510      Jellis  26/08/2017      15.4    3131.0  ...       1.0  1.0
13511  Stockdale  26/08/2017      11.2    3046.0  ...       1.0  1.0
13572   Morrison  26/08/2017      14.5    3087.0  ...       1.0  1.0

       Landsize  BuildingArea  YearBuilt  CouncilArea Lattitude
Longtitude  \
```

```
0         202.0           NaN        NaN        Yarra -37.79960
144.99840
13        195.0           NaN        NaN        Yarra -37.80840
144.99730
17        138.0         105.0     1890.0        Yarra -37.80210
144.99650
19        150.0          73.0     1985.0        Yarra -37.80110
145.00040
23          0.0          60.0     1970.0        Yarra -37.80150
144.99720
...         ...           ...        ...          ...       ...
...
13482     129.0          97.0     1940.0          NaN -37.87798
145.06731
13495    1475.0          66.0     1970.0          NaN -37.75799
144.92354
13510     903.0           NaN     1985.0          NaN -37.80640
145.18452
13511     716.0           NaN        NaN          NaN -37.71589
144.92176
13572     210.0          79.0     2006.0          NaN -37.70657
145.07878

                   Regionname Propertycount
0      Northern Metropolitan        4019.0
13     Northern Metropolitan        4019.0
17     Northern Metropolitan        4019.0
19     Northern Metropolitan        4019.0
23     Northern Metropolitan        4019.0
...                      ...           ...
13482  Southern Metropolitan        8801.0
13495   Western Metropolitan        6232.0
13510   Eastern Metropolitan        4973.0
13511  Northern Metropolitan        2651.0
13572  Northern Metropolitan        2329.0

[2137 rows x 21 columns]
```

# Inspecting if there are any missing values: and if there are, performing the following:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13580 entries, 0 to 13579
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
```

```
---  ------          --------------  -----
 0   Suburb          13580 non-null  object
 1   Address         13580 non-null  object
 2   Rooms           13580 non-null  int64
 3   Type            13580 non-null  object
 4   Price           13580 non-null  float64
 5   Method          13580 non-null  object
 6   SellerG         13580 non-null  object
 7   Date            13580 non-null  object
 8   Distance        13580 non-null  float64
 9   Postcode        13580 non-null  float64
 10  Bedroom2        13580 non-null  float64
 11  Bathroom        13580 non-null  float64
 12  Car             13518 non-null  float64
 13  Landsize        13580 non-null  float64
 14  BuildingArea    7130 non-null   float64
 15  YearBuilt       8205 non-null   float64
 16  CouncilArea     12211 non-null  object
 17  Lattitude       13580 non-null  float64
 18  Longtitude      13580 non-null  float64
 19  Regionname      13580 non-null  object
 20  Propertycount   13580 non-null  float64
dtypes: float64(12), int64(1), object(8)
memory usage: 2.2+ MB
```

Remove row

```
remove_row = df.dropna()
print(remove_row.shape)
print(remove_row.info())
```

```
(6196, 21)
<class 'pandas.core.frame.DataFrame'>
Index: 6196 entries, 1 to 12212
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Suburb          6196 non-null   object
 1   Address         6196 non-null   object
 2   Rooms           6196 non-null   int64
 3   Type            6196 non-null   object
 4   Price           6196 non-null   float64
 5   Method          6196 non-null   object
 6   SellerG         6196 non-null   object
 7   Date            6196 non-null   object
 8   Distance        6196 non-null   float64
 9   Postcode        6196 non-null   float64
 10  Bedroom2        6196 non-null   float64
 11  Bathroom        6196 non-null   float64
```

```
 12  Car             6196 non-null   float64
 13  Landsize        6196 non-null   float64
 14  BuildingArea    6196 non-null   float64
 15  YearBuilt       6196 non-null   float64
 16  CouncilArea     6196 non-null   object
 17  Lattitude       6196 non-null   float64
 18  Longtitude      6196 non-null   float64
 19  Regionname      6196 non-null   object
 20  Propertycount   6196 non-null   float64
dtypes: float64(12), int64(1), object(8)
memory usage: 1.0+ MB
None
```

Replace missing values with zeros and compare the average value of the before versus after

```
print(df.isnull().sum(),"\n")
replace_zero = df.fillna(value=0)
# print(replace_zero)
print(df["Car"].mean())
print(df["BuildingArea"].mean())
print(df["YearBuilt"].mean())
print(replace_zero["Car"].mean())
print(replace_zero["BuildingArea"].mean())
print(replace_zero["YearBuilt"].mean())
```

```
Suburb              0
Address             0
Rooms               0
Type                0
Price               0
Method              0
SellerG             0
Date                0
Distance            0
Postcode            0
Bedroom2            0
Bathroom            0
Car                62
Landsize            0
BuildingArea     6450
YearBuilt        5375
CouncilArea      1369
Lattitude           0
Longtitude          0
Regionname          0
Propertycount       0
dtype: int64

1.6100754549489569
```

```
151.96764988779805
1964.6842169408897
1.6027245949926363
79.78861146539029
1187.0569955817377
```

Data imputation for numeric value colums (mean) remove row that contain missing value (non numeric)

```
df.isnull().sum()
df['Car'] = df['Car'].fillna(df['Car'].median())
df = df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6196 entries, 1 to 12212
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Suburb         6196 non-null   object 
 1   Address        6196 non-null   object 
 2   Rooms          6196 non-null   int64  
 3   Type           6196 non-null   object 
 4   Price          6196 non-null   float64
 5   Method         6196 non-null   object 
 6   SellerG        6196 non-null   object 
 7   Date           6196 non-null   object 
 8   Distance       6196 non-null   float64
 9   Postcode       6196 non-null   float64
 10  Bedroom2       6196 non-null   float64
 11  Bathroom       6196 non-null   float64
 12  Car            6196 non-null   float64
 13  Landsize       6196 non-null   float64
 14  BuildingArea   6196 non-null   float64
 15  YearBuilt      6196 non-null   float64
 16  CouncilArea    6196 non-null   object 
 17  Lattitude      6196 non-null   float64
 18  Longtitude     6196 non-null   float64
 19  Regionname     6196 non-null   object 
 20  Propertycount  6196 non-null   float64
dtypes: float64(12), int64(1), object(8)
memory usage: 1.0+ MB
```

I think it is ok to do this because for the non-numeric columns because you don't know what to replace it with.

```
df["Date"] = pd.to_datetime(df["Date"], format="mixed") #doesn't
actually work along the format.
df['Date'] = df['Date'].dt.strftime('%Y/%m/%d')
```

```
C:\Users\woosh\AppData\Local\Temp\ipykernel_10640\1004160185.py:1:
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/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df["Date"] = pd.to_datetime(df["Date"], format="mixed") #doesn't
actually work along the format.
C:\Users\woosh\AppData\Local\Temp\ipykernel_10640\1004160185.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/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df['Date'] = df['Date'].dt.strftime('%Y/%m/%d')
```