txuinohpj

March 31, 2025

1 Experiment Notebook

1.1 0. Setup Environment

1.1.1 0.a Install Mandatory Packages

Do not modify this code before running it

[1]:
```
# Do not modify this code

import os
import sys
from pathlib import Path

COURSE = "36106"
ASSIGNMENT = "AT1"
DATA = "data"

asgmt_path = f"{COURSE}/assignment/{ASSIGNMENT}"
root_path = "./"

print("###### Install required Python packages ######")
! pip install -r https://raw.githubusercontent.com/aso-uts/labs_datasets/main/
  ↪36106-mlaa/requirements.txt

if os.getenv("COLAB_RELEASE_TAG"):

    from google.colab import drive
    from pathlib import Path

    print("\n###### Connect to personal Google Drive ######")
    gdrive_path = "/content/gdrive"
    drive.mount(gdrive_path)
    root_path = f"{gdrive_path}/MyDrive/"

print("\n###### Setting up folders ######")
folder_path = Path(f"{root_path}/{asgmt_path}/") / DATA
```

```
folder_path.mkdir(parents=True, exist_ok=True)
print(f"\nYou can now save your data files in: {folder_path}")

if os.getenv("COLAB_RELEASE_TAG"):
    %cd {folder_path}
```

```
###### Install required Python packages ######
Requirement already satisfied: pandas==2.2.2 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from -r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2.2.2)
Requirement already satisfied: scikit-learn==1.6.1 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from -r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (1.6.1)
Requirement already satisfied: altair==5.5.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from -r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (5.5.0)
Requirement already satisfied: numpy>=1.23.2 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from pandas==2.2.2->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (1.24.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from pandas==2.2.2->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from pandas==2.2.2->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.7 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from pandas==2.2.2->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2023.3)
Requirement already satisfied: scipy>=1.6.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.6.1->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (1.11.1)
Requirement already satisfied: joblib>=1.2.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.6.1->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (1.2.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from scikit-
learn==1.6.1->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (3.5.0)
```

```
Requirement already satisfied: jinja2 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (3.1.2)
Requirement already satisfied: jsonschema>=3.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (4.17.3)
Requirement already satisfied: narwhals>=1.14.2 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (1.31.0)
Requirement already satisfied: packaging in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (23.1)
Requirement already satisfied: typing-extensions>=4.10.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (4.12.2)
Requirement already satisfied: attrs>=17.4.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from
jsonschema>=3.0->altair==5.5.0->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (22.1.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from
jsonschema>=3.0->altair==5.5.0->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (0.18.0)
Requirement already satisfied: six>=1.5 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from python-
dateutil>=2.8.2->pandas==2.2.2->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/Users/ratikpant/anaconda3/lib/python3.11/site-packages (from
jinja2->altair==5.5.0->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (2.1.1)

###### Setting up folders ######

You can now save your data files in: 36106/assignment/AT1/data
```

1.1.2 0.b Disable Warnings Messages

Do not modify this code before running it

[2]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

1.1.3 0.c Install Additional Packages

If you are using additional packages, you need to install them here using the command:
`! pip install <package_name>`

[3]:
```
# <Student to fill this section>
```

1.1.4 0.d Import Packages

[3]:
```
import ipywidgets as widgets
import numpy as np
import pandas as pd
import altair as alt
import matplotlib.pyplot as plt
import seaborn as sns
import re
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

1.2 A. Project Description

[5]:
```
# @title Student Information
wgt_student_name = widgets.Text(
    value=None,
    placeholder='<student to fill this section>',
    description='Student Name:',
    style={'description_width': 'initial'},
    disabled=False
)

wgt_student_id = widgets.Text(
    value=None,
    placeholder='<student to fill this section>',
    description='Student Id:',
    style={'description_width': 'initial'},
    disabled=False
)

widgets.HBox([wgt_student_name, wgt_student_id])
```

[5]:
```
HBox(children=(Text(value='', description='Student Name:', placeholder='<student
to fill this section>', style…
```

[6]:
```
# @title Experiment ID

wgt_experiment_id = widgets.BoundedIntText(
    value=0,
    min=0,
    max=3,
    step=1,
    description='Experiment ID:',
    style={'description_width': 'initial'},
    disabled=False
)
wgt_experiment_id
```

[6]:
```
BoundedIntText(value=0, description='Experiment ID:', max=3,
style=DescriptionStyle(description_width='initial…
```

[7]:
```
# @title Business Objective

wgt_business_objective = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Business Objective:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_business_objective
```

[7]:
```
Textarea(value='', description='Business Objective:',
layout=Layout(height='100%', width='auto'), placeholder=…
```

1.3 B. Experiment Description

[8]:
```
# @title Experiment Hypothesis

wgt_experiment_hypothesis = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Experiment Hypothesis:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_experiment_hypothesis
```

[8]:
```
Textarea(value='', description='Experiment Hypothesis:',
layout=Layout(height='100%', width='auto'), placehold…
```

[9]:
```
# @title Experiment Expectations

wgt_experiment_expectations = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Experiment Expectations:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_experiment_expectations
```

[9]:
```
Textarea(value='', description='Experiment Expectations:',
layout=Layout(height='100%', width='auto'), placeho…
```

1.4 C. Data Understanding

1.4.1 C.1 Load Datasets

Do not change this code

[10]:
```
pwd
```

[10]:
```
'/Users/ratikpant/Desktop'
```

[11]:
```
# Load training data
training_df = pd.read_csv('/Users/ratikpant/Desktop/machine learning/
 ↪rental_training.csv')
```

[12]:
```
# Load validation data
validation_df = pd.read_csv( "/Users/ratikpant/Desktop/machine learning/
 ↪rental_validation.csv")
```

[13]:
```
# Load testing data
testing_df = pd.read_csv( "/Users/ratikpant/Desktop/machine learning/
 ↪rental_testing.csv")
```

1.4.2 C.2 Explore Training Set

You can add more cells in this section

[14]:
```
#checking dimensionality.
```

[15]:
```
training_df.shape
```

[15]: (3434, 20)

[16]:
```
validation_df.shape
```

[16]: (1320, 20)

[17]:
```
testing_df.shape
```

[17]: (1364, 20)

[18]:
```
#eda on training set
```

[19]:
```
training_df
```

[19]:

	advertised_date	number_of_bedrooms	rent	floor_area	level	\
0	2022-05-18	2	568.0	1100	Ground out of 2	
1	2022-05-13	2	581.0	800	1 out of 3	
2	2022-05-16	2	577.0	1000	1 out of 3	
3	2022-05-09	2	565.0	850	1 out of 2	
4	2022-04-29	2	564.0	600	Ground out of 1	
…	…	…	…	…	…	
3429	2022-06-08	3	600.0	1250	4 out of 5	
3430	2022-06-02	2	571.0	1350	2 out of 2	
3431	2022-05-18	2	574.0	1000	3 out of 5	
3432	2022-05-15	3	592.0	2000	1 out of 4	
3433	2022-05-04	2	574.0	1000	4 out of 5	

	suburb	furnished	tenancy_preference	number_of_bathrooms	\
0	Canberra	Unfurnished	Bachelors/Family	2	
1	Canberra	Semi-Furnished	Bachelors/Family	1	
2	Canberra	Semi-Furnished	Bachelors/Family	1	
3	Canberra	Unfurnished	Bachelors	1	
4	Canberra	Unfurnished	Bachelors/Family	2	
…	…	…	…	…	
3429	Perth	Furnished	Bachelors	2	
3430	Perth	Unfurnished	Bachelors/Family	2	
3431	Perth	Semi-Furnished	Bachelors/Family	2	
3432	Perth	Semi-Furnished	Bachelors/Family	3	
3433	Perth	Unfurnished	Bachelors	2	

	point_of_contact	secondary_address	building_number	street_name	\
0	Contact Owner	02/	1	Mcdowell Edge	
1	Contact Owner	667/	6	Lewis Parkway	
2	Contact Owner	859/	459	Daniel Copse	
3	Contact Owner	Flat 54	482	Young Walkway	

```
4       Contact Owner          Unit 75               838       Michael Port
…               …                …                  …              …
3429    Contact Owner              14/                8  Elizabeth Laneway
3430    Contact Owner           Flat 86              65    Michael Landing
3431    Contact Owner           Level 7             314      Flores Siding
3432    Contact Owner          Apt. 131             211      Jason Viaduct
3433    Contact Owner              72/               23       Taylor Corso

     street_suffix prefix first_name   last_name gender     phone_number  \
0         Driveway    Mr.     Robert       Jones      m   (08) 8174 5701
1          Viaduct   Mrs.       Lisa    Mcknight      f   (08).5553.7944
2          Meander    NaN    Annette      Lester      u   (03).6394.3934
3        Firetrail   Mrs.       Emma        Hill      f     +61836311377
4        Esplanade   Miss     Ariana  Richardson      f  +61 409 341 340
…              …    …        …          …    …               …
3429        Subway   Miss        Dr.     Thabani      f   (02).6367.5421
3430        Access    Dr.       Dawn     Spencer      f        3690 6564
3431          Park    Dr.      Craig      Garner      m     0469-517-332
3432     Deviation   Mrs.   Samantha       Silva      f     0485-687-657
3433     Anchorage    NaN      David       Dixon      u  +61 487 589 767

                          email
0       georgelopez@example.org
1      robertdorsey@example.net
2     rodriguezkaren@example.net
3      johnsonjeremy@example.com
4             sbrown@example.net
…                             …
3429   shannonharvey@example.net
3430  vincentheather@example.net
3431   jessicahowell@example.net
3432         gmiller@example.net
3433        ehiggins@example.com

[3434 rows x 20 columns]
```

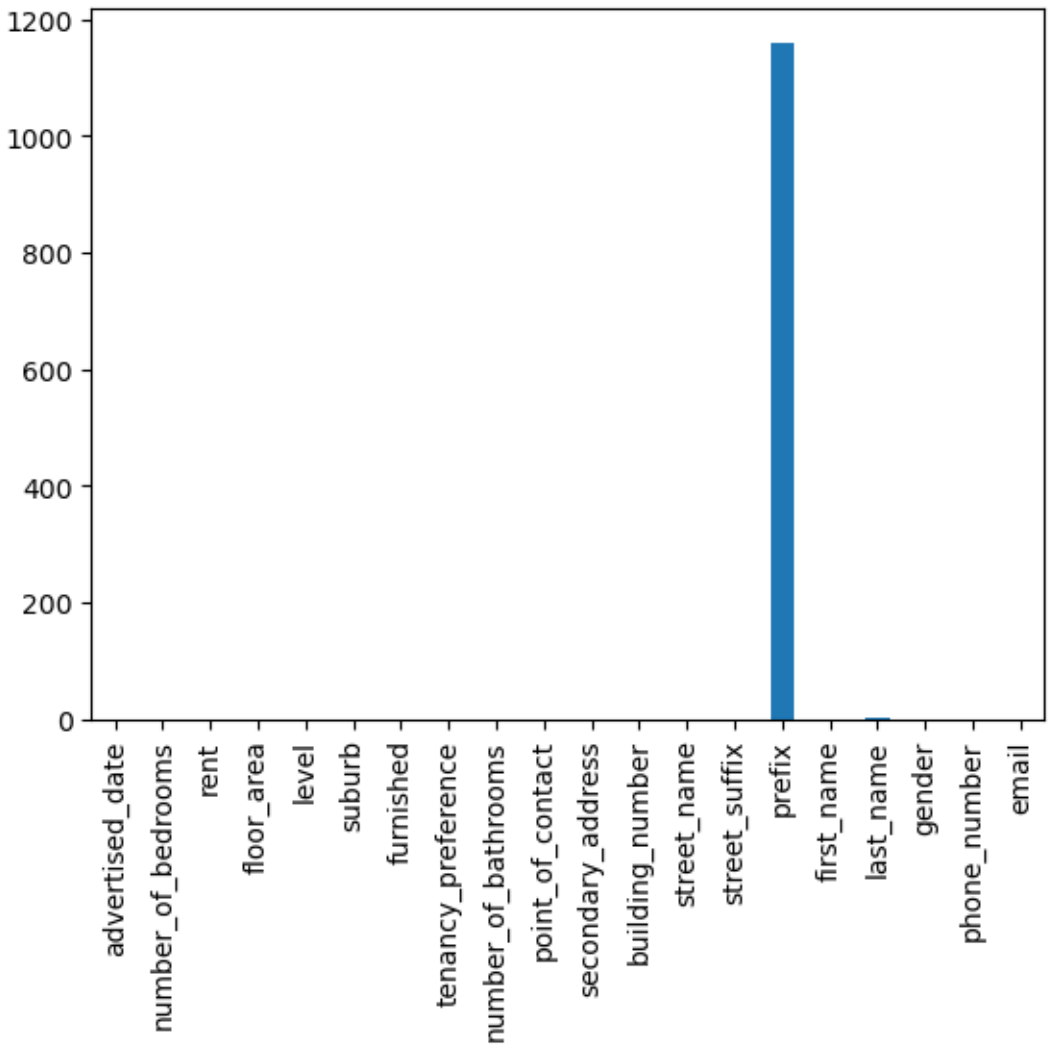
[20]:
```
#duplicates
```

[21]:
```
training_df.duplicated().sum()
```

[21]:
```
0
```

[22]:
```
#missing/null values
```

[23]:
```
training_df.isnull().sum().plot(kind = 'bar')
plt.show()
```



[24]:
```
#prefix has some missing values
```

[25]:
```
training_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3434 entries, 0 to 3433
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   advertised_date      3434 non-null   object
 1   number_of_bedrooms   3434 non-null   int64
 2   rent                 3434 non-null   float64
 3   floor_area           3434 non-null   int64
 4   level                3434 non-null   object
 5   suburb               3434 non-null   object
```

```
 6   furnished             3434 non-null   object
 7   tenancy_preference    3434 non-null   object
 8   number_of_bathrooms   3434 non-null   int64
 9   point_of_contact      3434 non-null   object
 10  secondary_address     3434 non-null   object
 11  building_number       3434 non-null   int64
 12  street_name           3434 non-null   object
 13  street_suffix         3434 non-null   object
 14  prefix                2274 non-null   object
 15  first_name            3434 non-null   object
 16  last_name             3433 non-null   object
 17  gender                3434 non-null   object
 18  phone_number          3434 non-null   object
 19  email                 3434 non-null   object
dtypes: float64(1), int64(4), object(15)
memory usage: 536.7+ KB
```

[26]:
```
#one mssing value in column-> "last_name". Which, as per the business , the␣
 ↪column last_name is not imprtant.
#we will later drop the entire column along with not so required columns.
```

[27]:
```
#lets explore column advertised date, and also change the dtype to date-time.
```

[28]:
```
training_df['advertised_date']
```

[28]:
```
0       2022-05-18
1       2022-05-13
2       2022-05-16
3       2022-05-09
4       2022-04-29
           ...
3429    2022-06-08
3430    2022-06-02
3431    2022-05-18
3432    2022-05-15
3433    2022-05-04
Name: advertised_date, Length: 3434, dtype: object
```

[29]:
```
training_df['advertised_date'] = pd.to_datetime(training_df['advertised_date'])
```

[30]:
```
#extracting year, month , day into separate columns for better data␣
 ↪visualisations,
#improving model features and analyse trends over time.
```

[31]:
```
training_df['yearmonth'] = training_df['advertised_date'].dt.to_period('M')
```

[32]:
```
training_df['advertised_year'] = training_df['advertised_date'].dt.year
training_df['advertised_month'] = training_df['advertised_date'].dt.month
training_df['advertised_day'] = training_df['advertised_date'].dt.day
```

[33]:
```
#using regular expression to clean up phone number column
# cleaning special characters eg: '+*' '\ /'
```

[34]:
```
training_df['phone_number'] = training_df['phone_number'].apply(lambda x: re.
 ↪sub('\D', ' ', x))
```

[35]:
```
#adjusting abnormal gaps between phone numbers.
#also replacing starting 61 with 0
```

[36]:
```
training_df['phone_number'] = training_df['phone_number'].replace(" ", "",␣
 ↪regex=True)
training_df['phone_number'] = training_df['phone_number'].replace("61", "0",␣
 ↪regex=True)
```

[37]:
```
training_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3434 entries, 0 to 3433
Data columns (total 24 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   advertised_date      3434 non-null   datetime64[ns]
 1   number_of_bedrooms   3434 non-null   int64
 2   rent                 3434 non-null   float64
 3   floor_area           3434 non-null   int64
 4   level                3434 non-null   object
 5   suburb               3434 non-null   object
 6   furnished            3434 non-null   object
 7   tenancy_preference   3434 non-null   object
 8   number_of_bathrooms  3434 non-null   int64
 9   point_of_contact     3434 non-null   object
 10  secondary_address    3434 non-null   object
 11  building_number      3434 non-null   int64
 12  street_name          3434 non-null   object
 13  street_suffix        3434 non-null   object
 14  prefix               2274 non-null   object
 15  first_name           3434 non-null   object
 16  last_name            3433 non-null   object
 17  gender               3434 non-null   object
 18  phone_number         3434 non-null   object
 19  email                3434 non-null   object
 20  yearmonth            3434 non-null   period[M]
 21  advertised_year      3434 non-null   int32
```

```
 22  advertised_month     3434 non-null   int32
 23  advertised_day       3434 non-null   int32
dtypes: datetime64[ns](1), float64(1), int32(3), int64(4), object(14),
period[M](1)
memory usage: 603.8+ KB
```

[38]:
```
# lETS look at column "level"
```

[39]:
```
training_df['level']
```

[39]:
```
0       Ground out of 2
1            1 out of 3
2            1 out of 3
3            1 out of 2
4       Ground out of 1
            …
3429         4 out of 5
3430         2 out of 2
3431         3 out of 5
3432         1 out of 4
3433         4 out of 5
Name: level, Length: 3434, dtype: object
```

[40]:
```
# we have to transform this column into two separate columns
#1st --> current level
#2nd --> total_level
```

[41]:
```
training_df['level'].unique()
```

[41]:
```
array(['Ground out of 2', '1 out of 3', '1 out of 2', 'Ground out of 1',
       'Ground out of 4', '1 out of 4', '1 out of 1', 'Ground out of 3',
       '2 out of 3', '4 out of 5', '2 out of 2', '2 out of 5',
       '4 out of 14', '3 out of 3', '5 out of 5', '7 out of 8',
       '2 out of 4', '4 out of 4', '3 out of 4', '1 out of 5',
       '8 out of 5', 'Ground out of 6', 'Ground out of 5', '3 out of 5',
       '11 out of 19', '5 out of 10', '11 out of 14',
       'Lower Basement out of 2', '2 out of 7', '4 out of 10',
       '7 out of 10', '2 out of 13', '6 out of 7', '4 out of 7',
       '14 out of 14', '2 out of 8', '5 out of 12', '3 out of 7',
       '7 out of 19', '14 out of 23', 'Upper Basement out of 9',
       '3 out of 21', '1 out of 22', '8 out of 8', '6 out of 12',
       'Upper Basement out of 16', '60 out of 66', '5 out of 8',
       '5 out of 7', '12 out of 18', '26 out of 44', '1 out of 8',
       '53 out of 78', 'Ground out of 7', '13 out of 20', '10 out of 18',
       '39 out of 60', '16 out of 21', '12 out of 24', '4 out of 8',
       '11 out of 21', '28 out of 30', '6 out of 21', '8 out of 16',
       '8 out of 28', '9 out of 15', '14 out of 22', '12 out of 45',
```

```
'25 out of 35', '2 out of 6', '7 out of 15',
'Upper Basement out of 20', '5 out of 20',
'Upper Basement out of 40', '5 out of 18', '4 out of 6',
'15 out of 18', '65 out of 78', '17 out of 22', '40 out of 75',
'11 out of 28', '10 out of 22', '17 out of 24', '15 out of 19',
'9 out of 10', '11 out of 13', '9 out of 19', '6 out of 11',
'11 out of 20', '10 out of 23', '14 out of 18', '6 out of 10',
'7 out of 7', '14 out of 58', '18 out of 23', '19 out of 19',
'9 out of 20', '13 out of 14', '7 out of 11', '11 out of 22',
'Upper Basement out of 30', '12 out of 14', '12 out of 13',
'2 out of 12', '9 out of 22', '7 out of 14', '10 out of 12',
'9 out of 14', '8 out of 20', '8 out of 15', '3 out of 6',
'17 out of 20', '9 out of 30', '3 out of 8', '11 out of 26',
'10 out of 32', '12 out of 16', '65 out of 76', '1 out of 7',
'5 out of 14', '17 out of 60', '10 out of 16', '20 out of 22',
'18 out of 25', '15 out of 17', '15 out of 23', '5 out of 17',
'3 out of 28', '5 out of 24', '16 out of 32', '21 out of 22',
'9 out of 12', '15 out of 32', '16 out of 23', '7 out of 12',
'14 out of 20', '18 out of 45', '15 out of 15', '1 out of 20',
'2 out of 9', '12 out of 22', '4 out of 12', '4 out of 9',
'2 out of 22', '6 out of 18', '35 out of 55', '16 out of 29',
'30 out of 45', '12 out of 19', '13 out of 23', '9 out of 38',
'6 out of 8', '8 out of 13', '19 out of 30', '7 out of 21',
'4 out of 15', '3 out of 9', '8 out of 12', '1 out of 9',
'5 out of 22', '9 out of 40', 'Ground out of 8', '18 out of 24',
'8 out of 17', '4 out of 11', '10 out of 11', '10 out of 28',
'14 out of 17', '5 out of 13', '18 out of 32', '10 out of 25',
'13 out of 16', '8 out of 10', '18 out of 21', '27 out of 58',
'19 out of 25', '10 out of 14', '8 out of 14', '12 out of 20',
'10 out of 13', '45 out of 77', '18 out of 19', '10 out of 20',
'15 out of 24', '15 out of 20', '16 out of 22', '18 out of 30',
'24 out of 55', 'Upper Basement out of 7', '11 out of 27',
'11 out of 23', '6 out of 15', '3 out of 12', '15 out of 36',
'15 out of 25', '10 out of 24', '15 out of 28', '6 out of 20',
'23 out of 23', '5 out of 15', '16 out of 18',
'Upper Basement out of 22', '9 out of 31', '6 out of 14',
'5 out of 21', '32 out of 59', '20 out of 32', '25 out of 43',
'9 out of 18', '10 out of 37', '16 out of 36', '4 out of 22',
'Upper Basement out of 10', '8 out of 18', '11 out of 11',
'5 out of 23', '60 out of 77', '4 out of 20', '6 out of 16',
'5 out of 16', '15 out of 22', '3 out of 13', '30 out of 58',
'7 out of 16', '5 out of 6', '5 out of 9', '18 out of 28',
'14 out of 27', '9 out of 16', '25 out of 50', '6 out of 30',
'8 out of 58', '20 out of 41', '12 out of 21', '28 out of 39',
'15 out of 58', '6 out of 23', '21 out of 58', '7 out of 28',
'7 out of 23', '2 out of 17', '6 out of 24', '76 out of 78',
'3 out of 10', '20 out of 27', '8 out of 36', '9 out of 21',
```

```
'12 out of 25', '7 out of 20', '9 out of 35', '11 out of 15',
'15 out of 60', '18 out of 20', '14 out of 21', '8 out of 22',
'20 out of 31', '27 out of 45', '19 out of 20', '19 out of 85',
'3 out of 23', '4 out of 27', '35 out of 60', '21 out of 33',
'25 out of 52', '2 out of 24', '1 out of 6', '18 out of 33',
'1 out of 10', '45 out of 60', '36 out of 81', '24 out of 60',
'16 out of 38', '8 out of 45', '8 out of 32', '10 out of 10',
'7 out of 18', '8 out of 19', '6 out of 17', '18 out of 22',
'16 out of 34', 'Ground out of 12', '2 out of 10', '6 out of 9',
'Ground out of 18', '20 out of 25', '11 out of 18', '11 out of 25',
'24 out of 25', '17 out of 19', 'Upper Basement out of 4',
'8 out of 9', 'Lower Basement out of 3', '12 out of 23',
'9 out of 11', 'Ground out of 9', '1 out of 24', '1 out of 12',
'3', 'Ground', '17 out of 31', '15 out of 29', '3 out of 17',
'Lower Basement out of 1', '1 out of 14',
'Upper Basement out of 2', '2 out of 14', '10 out of 19',
'10 out of 15', '24 out of 31', '2 out of 32', '2 out of 16',
'9 out of 13', '6 out of 29', '28 out of 31', '3 out of 11',
'7 out of 9', '2 out of 11', '11 out of 12', '3 out of 14',
'1 out of 16', '25 out of 32', '11 out of 16', '10 out of 31',
'7 out of 17', 'Upper Basement out of 3', 'Ground out of 13',
'13 out of 25', '23 out of 35', '5 out of 34', '1', '4 out of 31',
'4 out of 26', '1 out of 35', '12 out of 17'], dtype=object)
```

[42]:
```
#treating all the lower upper and ground levels into separate column 'current level'.
```

[43]:
```
training_df['current_level'] = training_df['level'].apply(lambda x : -1 if 'Lower Basement' in x else
                                                           -0.5 if  'Upper Basement' in x else
                                                           0 if 'Ground' in x else
                                                           int(x.split(' ')[0]))
```

[44]:
```
#now since there is a pattern "out of" in describing the total number of leveles.
#we extract  it using regex (regular expression)
```

[45]:
```
training_df['total_level'] = training_df['level'].str.extract(r'out of (\d+)')
```

[46]:
```
#converting total_level into float
training_df['total_level'] = training_df['total_level'].astype(float)
```

[47]:
```
training_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 3434 entries, 0 to 3433
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   advertised_date       3434 non-null   datetime64[ns]
 1   number_of_bedrooms    3434 non-null   int64
 2   rent                  3434 non-null   float64
 3   floor_area            3434 non-null   int64
 4   level                 3434 non-null   object
 5   suburb                3434 non-null   object
 6   furnished             3434 non-null   object
 7   tenancy_preference    3434 non-null   object
 8   number_of_bathrooms   3434 non-null   int64
 9   point_of_contact      3434 non-null   object
 10  secondary_address     3434 non-null   object
 11  building_number       3434 non-null   int64
 12  street_name           3434 non-null   object
 13  street_suffix         3434 non-null   object
 14  prefix                2274 non-null   object
 15  first_name            3434 non-null   object
 16  last_name             3433 non-null   object
 17  gender                3434 non-null   object
 18  phone_number          3434 non-null   object
 19  email                 3434 non-null   object
 20  yearmonth             3434 non-null   period[M]
 21  advertised_year       3434 non-null   int32
 22  advertised_month      3434 non-null   int32
 23  advertised_day        3434 non-null   int32
 24  current_level         3434 non-null   float64
 25  total_level           3430 non-null   float64
dtypes: datetime64[ns](1), float64(3), int32(3), int64(4), object(14),
period[M](1)
memory usage: 657.4+ KB
```

[48]:
```
# after separting level columns we discover few missing values in total_level.
#LETS LOOK ITNO THAT
```

[49]:
```
training_df[training_df['total_level'].isnull()]
```

[49]:

	advertised_date	number_of_bedrooms	rent	floor_area	level	suburb	furnished	tenancy_preference	number_of_bathrooms	point_of_contact
1868	2022-06-18	2	581.0	400	3	Adelaide	Unfurnished	Bachelors/Family	1	Contact Owner
2127	2022-05-23	1	578.0	450	Ground	Adelaide				
3265	2022-06-12	3	574.0	900	1	Perth				
3320	2022-05-31	3	574.0	1270	1	Perth				

```
2127       Furnished   Bachelors/Family                    1    Contact Owner
3265  Semi-Furnished   Bachelors/Family                    3    Contact Owner
3320       Furnished            Family                    2    Contact Owner

      … last_name  gender phone_number                         email  \
1868  …  Phillips       m     69131429  cristianbrowning@example.com
2127  …     Huang       f    079010546          xbarrett@example.net
3265  …      Diaz       u   0349246026            juan33@example.com
3320  …     Moore       m   0316258209           smiller@example.org

     yearmont advertised_year advertised_month advertised_day current_level  \
1868   2022-06            2022                6             18           3.0
2127   2022-05            2022                5             23           0.0
3265   2022-06            2022                6             12           1.0
3320   2022-05            2022                5             31           1.0

     total_level
1868         NaN
2127         NaN
3265         NaN
3320         NaN

[4 rows x 26 columns]
```

[50]:
```
#It is because not all of the level were in "out of " format. we will replace it with values of level.
```

[51]:
```
training_df['total_level'].fillna(training_df['level'],inplace = True)
```

[52]:
```
training_df['total_level'].replace({'Ground' : '0'},inplace = True)
```

[53]:
```
training_df
```

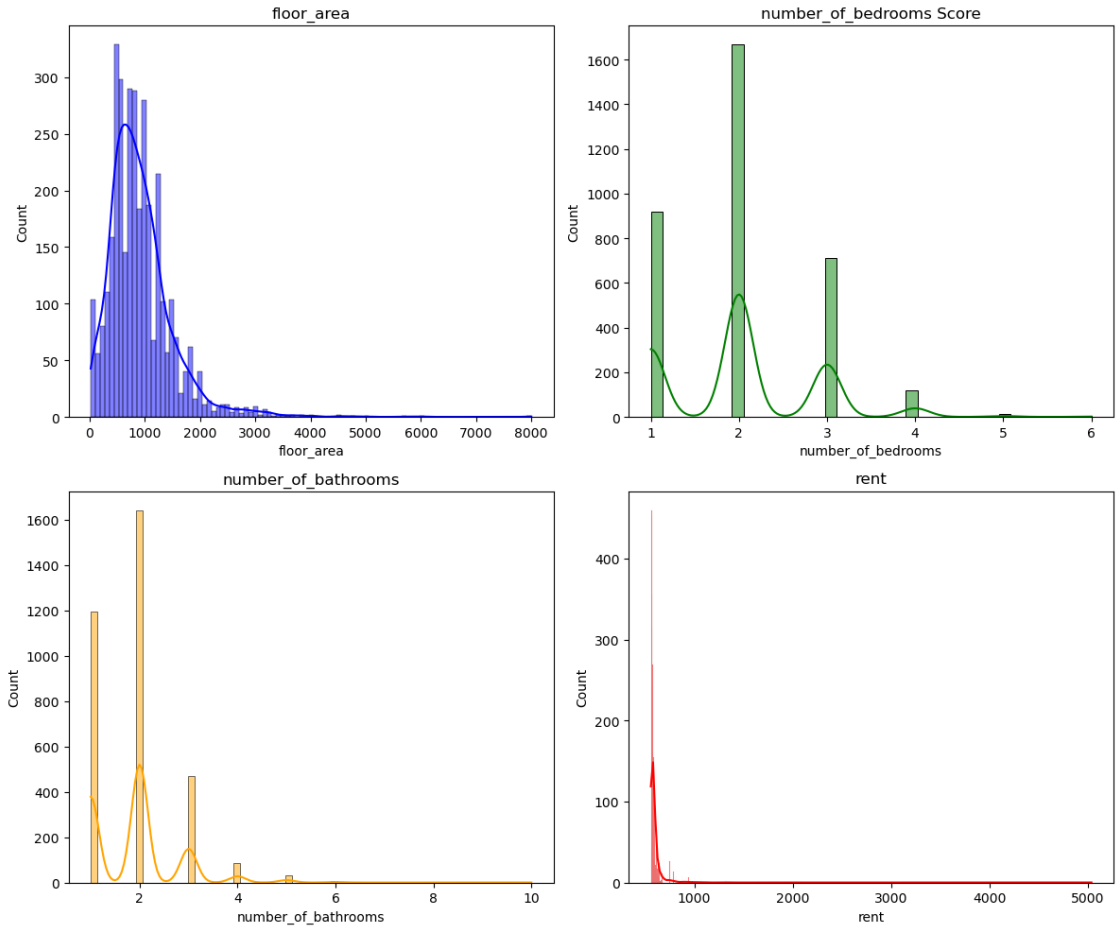
[53]:
```
     advertised_date  number_of_bedrooms   rent  floor_area            level  \
0         2022-05-18                   2  568.0        1100  Ground out of 2
1         2022-05-13                   2  581.0         800        1 out of 3
2         2022-05-16                   2  577.0        1000        1 out of 3
3         2022-05-09                   2  565.0         850        1 out of 2
4         2022-04-29                   2  564.0         600  Ground out of 1
…                …                   …    …           …                …
3429      2022-06-08                   3  600.0        1250        4 out of 5
3430      2022-06-02                   2  571.0        1350        2 out of 2
3431      2022-05-18                   2  574.0        1000        3 out of 5
3432      2022-05-15                   3  592.0        2000        1 out of 4
3433      2022-05-04                   2  574.0        1000        4 out of 5

        suburb       furnished tenancy_preference  number_of_bathrooms  \
```

```
0     Canberra     Unfurnished   Bachelors/Family                    2
1     Canberra  Semi-Furnished   Bachelors/Family                    1
2     Canberra  Semi-Furnished   Bachelors/Family                    1
3     Canberra     Unfurnished          Bachelors                    1
4     Canberra     Unfurnished   Bachelors/Family                    2
…        …             …                …                    …
3429     Perth       Furnished          Bachelors                    2
3430     Perth     Unfurnished   Bachelors/Family                    2
3431     Perth  Semi-Furnished   Bachelors/Family                    2
3432     Perth  Semi-Furnished   Bachelors/Family                    3
3433     Perth     Unfurnished          Bachelors                    2

     point_of_contact  …   last_name  gender phone_number  \
0       Contact Owner  …       Jones       m   0881745701
1       Contact Owner  …    Mcknight       f   0855537944
2       Contact Owner  …      Lester       u   0363943934
3       Contact Owner  …        Hill       f   0836311377
4       Contact Owner  …  Richardson       f   0409341340
…                 …  …         …     …          …
3429    Contact Owner  …     Thabani       f   0263675421
3430    Contact Owner  …     Spencer       f     36906564
3431    Contact Owner  …      Garner       m   0469517332
3432    Contact Owner  …       Silva       f   0485687657
3433    Contact Owner  …       Dixon       u   0487589767

                          email yearmonth advertised_year advertised_month  \
0        georgelopez@example.org   2022-05            2022                5
1       robertdorsey@example.net   2022-05            2022                5
2     rodriguezkaren@example.net   2022-05            2022                5
3      johnsonjeremy@example.com   2022-05            2022                5
4             sbrown@example.net   2022-04            2022                4
…                            …       …              …              …
3429   shannonharvey@example.net   2022-06            2022                6
3430  vincentheather@example.net   2022-06            2022                6
3431   jessicahowell@example.net   2022-05            2022                5
3432         gmiller@example.net   2022-05            2022                5
3433        ehiggins@example.com   2022-05            2022                5

     advertised_day current_level total_level
0                18           0.0         2.0
1                13           1.0         3.0
2                16           1.0         3.0
3                 9           1.0         2.0
4                29           0.0         1.0
…              …            …          …
3429              8           4.0         5.0
3430              2           2.0         2.0
```

```
3431              18           3.0         5.0
3432              15           1.0         4.0
3433               4           4.0         5.0

[3434 rows x 26 columns]
```

[54]:
```
#checking the data distibtion
```

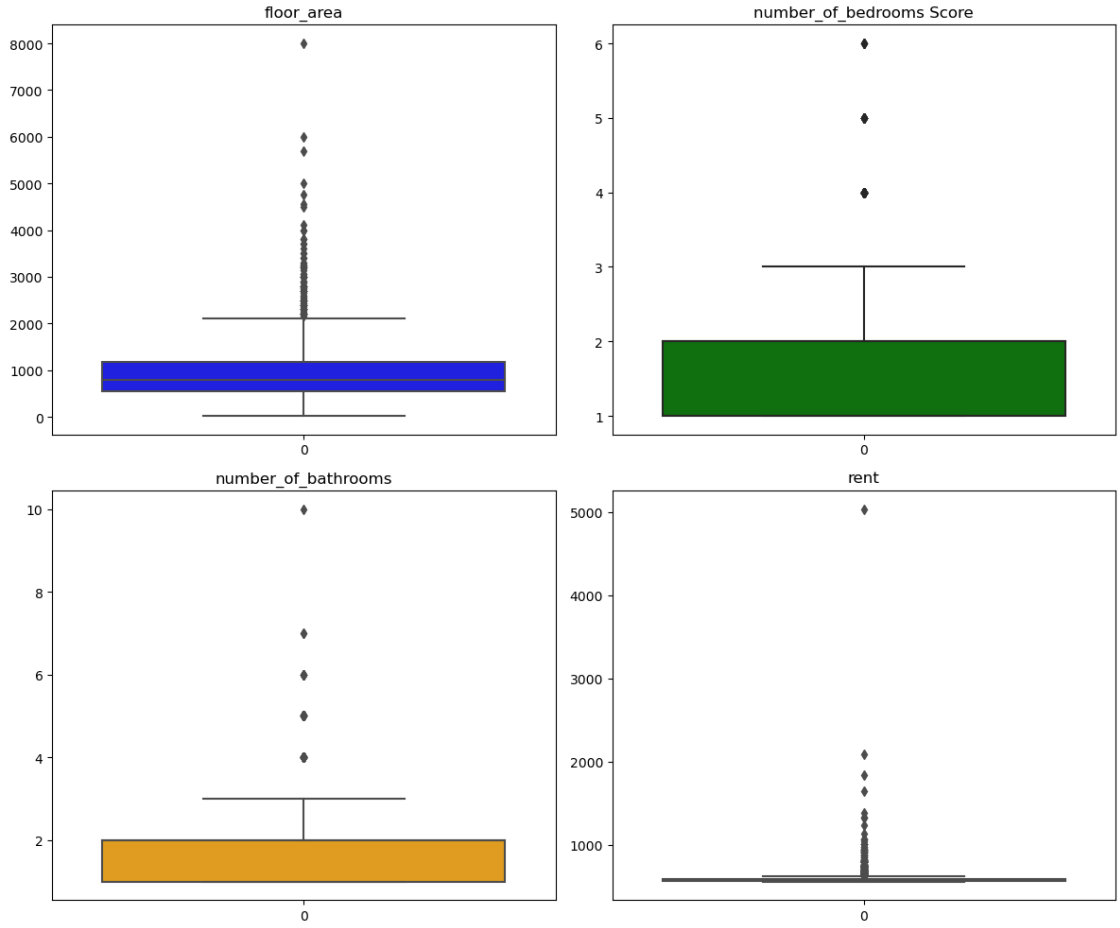
[55]:
```
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# Plot the histograms for each of the specified columns
sns.histplot(training_df['floor_area'], ax=axes[0, 0], kde=True, color='blue')
axes[0, 0].set_title('floor_area')

sns.histplot(training_df['number_of_bedrooms'], ax=axes[0, 1], kde=True,␣
 ↪color='green')
axes[0, 1].set_title('number_of_bedrooms Score')

sns.histplot(training_df['number_of_bathrooms'], ax=axes[1, 0], kde=True,␣
 ↪color='orange')
axes[1, 0].set_title('number_of_bathrooms')

sns.histplot(training_df['rent'], ax=axes[1, 1], kde=True, color='red')
axes[1, 1].set_title('rent')

# Adjust layout for better spacing
plt.tight_layout()

# Show the plot
plt.show()
```



[56]:
```
#1)"floor area" seems to be skewed to its right, however some of the houses␣
 ↪having close to 0 floor area doesn't seem real
#2)number of bedrooms, majority of them have standard number of bedrooms,
#3)number of bathroom again appears to have some large numbers which seem to be␣
 ↪off
#4)most of the weekly rent seem to have mid 500 -700 dollarsweely , however␣
 ↪some seem relatively high.
```

[57]:
```
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# Plot the histograms for each of the specified columns
sns.boxplot(training_df['floor_area'], ax=axes[0, 0], color='blue')
axes[0, 0].set_title('floor_area')

sns.boxplot(training_df['number_of_bedrooms'], ax=axes[0, 1],  color='green')
axes[0, 1].set_title('number_of_bedrooms Score')

sns.boxplot(training_df['number_of_bathrooms'], ax=axes[1, 0],  color='orange')
```

```
axes[1, 0].set_title('number_of_bathrooms')

sns.boxplot(training_df['rent'], ax=axes[1, 1], color='red')
axes[1, 1].set_title('rent')

# Adjust layout for better spacing
plt.tight_layout()

# Show the plot
plt.show()
```



[58]:
```
#as we acknlowdged above there are some anomalies in the dataset, and therefore
↪we will fix them.
```

[59]:
```
#checking on the outliers of 'floor area' using IQR
```

[60]:
```
Q1 = training_df['floor_area'].quantile(0.25)
Q3 = training_df['floor_area'].quantile(0.75)
```

```
IQR = Q3-Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

[61]:
```
#SINCE WE HAVE 118 OUTLIERS AND REMOVING THEM SHOULD NOT CAUSE ANY PROBLEMS. WE␣
↪DROP THEM
```

[62]:
```
outliers = training_df[(training_df['floor_area'] < lower_bound) |␣
↪(training_df['floor_area'] > upper_bound)]
outliers
```

[62]:

	advertised_date	number_of_bedrooms	rent	floor_area	level	\
151	2022-05-13	1	619.0	2200	2 out of 5	
225	2022-06-24	1	563.0	2160	1 out of 2	
319	2022-06-08	3	606.0	3500	1 out of 2	
431	2022-06-05	3	638.0	2210	Ground out of 2	
480	2022-05-14	4	1003.0	3000	39 out of 60	
…	…	…	…	…	…	
3315	2022-06-17	3	615.0	2400	1 out of 3	
3368	2022-06-20	4	613.0	2300	4 out of 5	
3389	2022-06-18	3	632.0	2170	4 out of 5	
3412	2022-06-17	3	600.0	2500	2 out of 2	
3415	2022-04-29	4	696.0	3250	12 out of 17	

	suburb	furnished	tenancy_preference	number_of_bathrooms	\
151	Canberra	Furnished	Bachelors/Family	3	
225	Canberra	Semi-Furnished	Bachelors	1	
319	Canberra	Semi-Furnished	Bachelors/Family	2	
431	Canberra	Unfurnished	Bachelors/Family	3	
480	Sydney	Semi-Furnished	Bachelors/Family	5	
…	…	…	…	…	
3315	Perth	Furnished	Bachelors/Family	3	
3368	Perth	Semi-Furnished	Bachelors/Family	4	
3389	Perth	Semi-Furnished	Bachelors/Family	3	
3412	Perth	Unfurnished	Bachelors/Family	2	
3415	Perth	Semi-Furnished	Bachelors/Family	5	

	point_of_contact	…	last_name	gender	phone_number	\
151	Contact Owner	…	Watkins	u	025058096	
225	Contact Owner	…	Sutton	m	0484251748	
319	Contact Owner	…	Garde	u	0419505956	
431	Contact Owner	…	Wilson	u	0341251215	
480	Contact Agent	…	Zavala	f	0280477824	
…	…	…	…	…	…	
3315	Contact Agent	…	Schmidt	f	0292513578	
3368	Contact Owner	…	Willis	m	0746953294	
3389	Contact Owner	…	Simpson	u	040677258	

```
3412    Contact Owner  …       May       m     42958405
3415    Contact Owner  …    Garcia       u     22914006

                          email yearmonth advertised_year advertised_month  \
151    burgesscolin@example.net   2022-05            2022                5
225   smithcourtney@example.com   2022-06            2022                6
319   christopher30@example.org   2022-06            2022                6
431   lewispatricia@example.org   2022-06            2022                6
480      tinamoreno@example.com   2022-05            2022                5
…                           …       …               …               …
3315         iklein@example.org   2022-06            2022                6
3368          hcain@example.com   2022-06            2022                6
3389     leonjohnny@example.org   2022-06            2022                6
3412  charlesvargas@example.com   2022-06            2022                6
3415    owenrebecca@example.net   2022-04            2022                4

     advertised_day current_level total_level
151              13           2.0         5.0
225              24           1.0         2.0
319               8           1.0         2.0
431               5           0.0         2.0
480              14          39.0        60.0
…               …            …           …
3315             17           1.0         3.0
3368             20           4.0         5.0
3389             18           4.0         5.0
3412             17           2.0         2.0
3415             29          12.0        17.0

[118 rows x 26 columns]
```

[63]:
```
#storing evertything but outliers in new df
training_cleaned = training_df[(training_df['floor_area'] > lower_bound) &␣
 ↪(training_df['floor_area'] < upper_bound)]
```

[ ]:
```

```

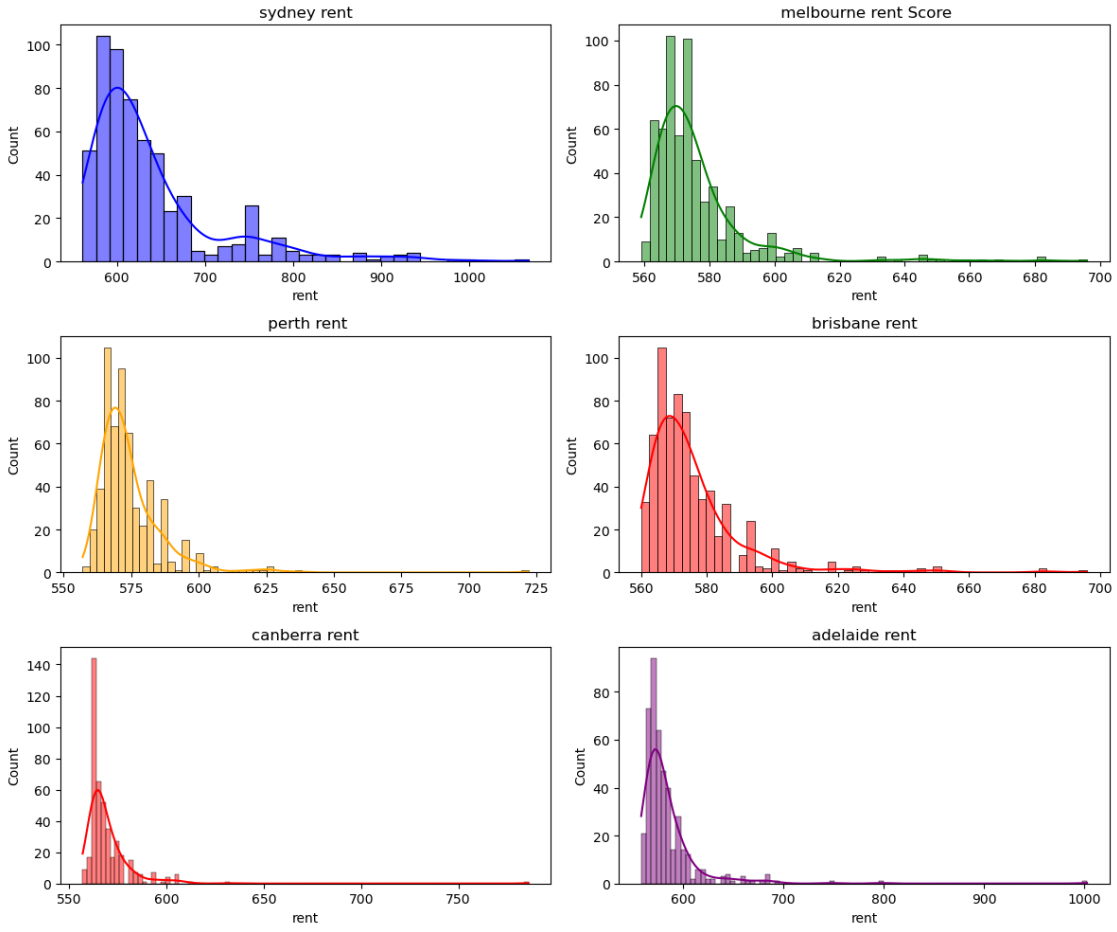
2 RENTAL ANALYSIS BY SUBURBS

[64]:
```
#Lets store each suburbs and perform exploration on if still some extreme␣
 ↪rental values can be detected.
```

[65]:
```
adelaide  = training_cleaned[training_cleaned['suburb'] == 'Adelaide']
sydney = training_cleaned[training_cleaned['suburb'] == 'Sydney']
melbn = training_cleaned[training_cleaned['suburb'] == 'Melbourne']
perth = training_cleaned[training_cleaned['suburb'] == 'Perth']
```

```
brisbn = training_cleaned[training_cleaned['suburb'] == 'Brisbane']
canb = training_cleaned[training_cleaned['suburb'] == 'Canberra']
```

[66]:
```
#distribution of rental prices by suburbs
fig, axes = plt.subplots(3, 2, figsize=(12, 10))

# Plot the histograms for each of the specified columns
sns.histplot(sydney['rent'], ax=axes[0, 0], kde=True, color='blue')
axes[0, 0].set_title('sydney rent')

sns.histplot(melbn['rent'], ax=axes[0, 1], kde=True, color='green')
axes[0, 1].set_title(' melbourne rent Score')

sns.histplot(perth['rent'], ax=axes[1, 0], kde=True, color='orange')
axes[1, 0].set_title(' perth rent')

sns.histplot(brisbn['rent'], ax=axes[1, 1], kde=True, color='red')
axes[1, 1].set_title('brisbane rent')

sns.histplot(canb['rent'], ax=axes[2, 0], kde=True, color='red')
axes[2, 0].set_title('canberra rent')

sns.histplot(adelaide['rent'], ax = axes[2,1], kde = True, color = 'purple')
axes[2,1].set_title('adelaide rent')

# Adjust layout for better spacing
plt.tight_layout()

# Show the plot
plt.show()
```



[67]:
```
# we can observe, all of them are heavily right skewed, some are more than␣
 ↪others:
#brisbane has extremities where rent is close to 5000 dollars a week.
#sydney has more than 2000 dollars a week
#adelaide has 1200 dollar a week
```

3 BRISBANE

[68]:
```
brisbn[brisbn['rent'] > 650]
```

[68]:

	advertised_date	number_of_bedrooms	rent	floor_area	level	\
1370	2022-05-12	3	651.0	1850	6 out of 11	
1527	2022-05-27	3	683.0	1800	2 out of 4	
1560	2022-05-23	4	651.0	2000	Ground out of 2	
1598	2022-06-10	3	651.0	2000	1 out of 5	
1616	2022-06-12	3	696.0	1500	6 out of 16	
1646	2022-06-02	3	683.0	1950	3 out of 3	

```
        suburb       furnished tenancy_preference  number_of_bathrooms  \
1370  Brisbane  Semi-Furnished          Bachelors                    3
1527  Brisbane  Semi-Furnished   Bachelors/Family                    3
1560  Brisbane       Furnished   Bachelors/Family                    3
1598  Brisbane       Furnished          Bachelors                    2
1616  Brisbane       Furnished   Bachelors/Family                    3
1646  Brisbane  Semi-Furnished   Bachelors/Family                    3

     point_of_contact  … last_name  gender phone_number  \
1370    Contact Agent  …   Kennedy       m    040477360
1527    Contact Agent  …     Moore       f   0328283216
1560    Contact Owner  …    Taylor       m   0365562111
1598    Contact Agent  …    Keller       f   0427801944
1616    Contact Agent  …     Lewis       u   0267385764
1646    Contact Agent  …     Brown       f   0353491570

                         email yearmonth advertised_year advertised_month  \
1370  tiffanythomas@example.com   2022-05            2022                5
1527   petersonjill@example.com   2022-05            2022                5
1560  christopher77@example.com   2022-05            2022                5
1598        awilson@example.net   2022-06            2022                6
1616    dalekennedy@example.net   2022-06            2022                6
1646      michael64@example.org   2022-06            2022                6

     advertised_day current_level total_level
1370             12           6.0        11.0
1527             27           2.0         4.0
1560             23           0.0         2.0
1598             10           1.0         5.0
1616             12           6.0        16.0
1646              2           3.0         3.0

[6 rows x 26 columns]
```

[69]:
```
#WE FIND EVERYTHING TO BE NORMAL IN THE SPREAD SO WE LEAVE IT AS IS
```

4 CANBERRA

[70]:
```
#understanding variation in canberra rent is logical or just something odd
canb_outlier = canb[canb['rent'] > 650].index
canb_outlier
```

[70]:
```
Index([90], dtype='int64')
```

[71]:
```
#since this is an extreme value we change it with the mean of canberra in the␣
↪"training_cleaned"
```

[72]:
```
training_cleaned.loc[canb_outlier, 'rent'] = canb['rent'].mean()
```

[73]:
```
#changing the training_cleaned values too.
```

5 Melbourne

[74]:
```
melbn[melbn['rent'] > 650]
```

[74]:

	advertised_date	number_of_bedrooms	rent	floor_area	level	\
2362	2022-05-27	3	658.0	2036	6 out of 11	
2388	2022-06-22	3	664.0	1725	10 out of 14	
2520	2022-06-17	3	670.0	2100	2 out of 4	
2564	2022-06-08	3	683.0	1975	12 out of 18	
2573	2022-05-20	3	696.0	1435	5 out of 15	
2795	2022-05-27	3	683.0	2000	3 out of 14	
2833	2022-06-14	3	651.0	2000	3 out of 4	

	suburb	furnished	tenancy_preference	number_of_bathrooms	\
2362	Melbourne	Semi-Furnished	Bachelors	3	
2388	Melbourne	Furnished	Bachelors/Family	2	
2520	Melbourne	Unfurnished	Bachelors	3	
2564	Melbourne	Semi-Furnished	Family	3	
2573	Melbourne	Furnished	Bachelors/Family	3	
2795	Melbourne	Semi-Furnished	Bachelors	3	
2833	Melbourne	Semi-Furnished	Bachelors/Family	3	

	point_of_contact	…	last_name	gender	phone_number	\
2362	Contact Agent	…	Turner	m	0282006211	
2388	Contact Agent	…	Terry	f	0329231977	
2520	Contact Agent	…	Ellis	u	0466301168	
2564	Contact Agent	…	Baker	m	32245886	
2573	Contact Owner	…	Harris	u	0738053811	
2795	Contact Agent	…	Moreno	m	7935017	
2833	Contact Owner	…	Ryan	u	0855717350	

	email	yearmonth	advertised_year	advertised_month	\
2362	leahcharles@example.org	2022-05	2022	5	
2388	payneanthony@example.org	2022-06	2022	6	
2520	williamssherri@example.com	2022-06	2022	6	
2564	tcastro@example.org	2022-06	2022	6	
2573	amyrice@example.net	2022-05	2022	5	
2795	ischneider@example.org	2022-05	2022	5	
2833	gerald20@example.net	2022-06	2022	6	

```
     advertised_day current_level total_level
2362             27           6.0        11.0
2388             22          10.0        14.0
2520             17           2.0         4.0
2564              8          12.0        18.0
2573             20           5.0        15.0
2795             27           3.0        14.0
2833             14           3.0         4.0

[7 rows x 26 columns]
```

[75]:
```
##WE FIND EVERYTHING TO BE NORMAL IN THE SPREAD SO WE LEAVE IT AS IS
```

6 ADELAIDE

[76]:
```
adelaide[adelaide['rent'] > 750]
```

[76]:
```
     advertised_date  number_of_bedrooms    rent  floor_area       level  \
1755      2022-06-20                   4  1003.0         800  1 out of 4
1954      2022-05-22                   5   798.0         200  2 out of 2

        suburb    furnished tenancy_preference  number_of_bathrooms  \
1755  Adelaide  Unfurnished   Bachelors/Family                    4
1954  Adelaide  Unfurnished          Bachelors                    5

     point_of_contact  … last_name  gender phone_number  \
1755    Contact Agent  …    Cortez       u   0803577431
1954    Contact Agent  …   Jimenez       u   0283813014

                     email yearmonth advertised_year advertised_month  \
1755    james79@example.com   2022-06            2022                6
1954  ythompson@example.net   2022-05            2022                5

     advertised_day current_level total_level
1755             20           1.0         4.0
1954             22           2.0         2.0

[2 rows x 26 columns]
```

[77]:
```
ade_outier =  adelaide[adelaide['rent'] > 750].index
training_cleaned.loc[ade_outier, 'rent'] = adelaide['rent'].mean()
```

[78]:
```
#we change these two extreme values to mean
```

7 PERTH

[79]:
```
perth[perth['rent']> 650]
```

[79]:
```
     advertised_date  number_of_bedrooms   rent  floor_area       level  \
2930      2022-05-06                   2  722.0         130  1 out of 1

     suburb    furnished tenancy_preference  number_of_bathrooms  \
2930  Perth  Unfurnished   Bachelors/Family                    2

     point_of_contact  … last_name  gender phone_number               email  \
2930    Contact Owner  …     Drake       f   0245334978  eric90@example.org

     yearmonth advertised_year advertised_month advertised_day current_level  \
2930   2022-05            2022                5              6           1.0

     total_level
2930         1.0

[1 rows x 26 columns]
```

[80]:
```
#PERTH IS FINE
```

8 SYDNEY

[81]:
```
sydney[sydney['rent'] > 950]
```

[81]:
```
    advertised_date  number_of_bedrooms    rent  floor_area         level  \
757      2022-05-19                   3  1067.0        1800   7 out of 14
908      2022-06-15                   3   978.0        1500  20 out of 41
995      2022-06-08                   3  1003.0        1663  19 out of 85

     suburb       furnished tenancy_preference  number_of_bathrooms  \
757  Sydney  Semi-Furnished          Bachelors                    4
908  Sydney  Semi-Furnished   Bachelors/Family                    2
995  Sydney  Semi-Furnished   Bachelors/Family                    2

    point_of_contact  … last_name  gender phone_number  \
757    Contact Agent  …  Thompson       m   0787124204
908    Contact Agent  …     Doyle       f   0456368495
995    Contact Agent  …     Baker       f   0824473521

                         email yearmonth advertised_year advertised_month  \
757       zerickson@example.com   2022-05            2022                5
908    robertssusan@example.net   2022-06            2022                6
995  douglasmarquez@example.org   2022-06            2022                6
```

```
     advertised_day current_level total_level
757              19           7.0        14.0
908              15          20.0        41.0
995               8          19.0        85.0

[3 rows x 26 columns]
```

[82]:
```
#considering sydney hosuing market is swelled up we let these values live in␣
↪the dataframe.
```

[83]:
```
#So far everything related to "floor area" and "rent" is reconciled and stored␣
↪in "training_cleaned"
```

9 GENDER affect on rental prices

[84]:
```
#lets check the gender column
```

[85]:
```
training_cleaned['gender'].value_counts()
```

[85]:
```
gender
u    1119
f    1106
m    1091
Name: count, dtype: int64
```

[86]:
```
#we have too many unknown columns.
# Question to ask:
#1) Does the gender affect the rental prices.
# Hypothesis: Gender does not have an affect on rental prices.
```

[87]:
```
#running a statistical tests
#Checking if DATA of rental_price in each gender group is normally distributed.
```

[88]:
```
from scipy.stats import shapiro
for gender in training_cleaned['gender'].unique():
    rents = training_cleaned[training_cleaned['gender'] == gender]['rent']
shapiro_stat , shapiro_p_value =  shapiro(rents)
if shapiro_p_value > 0.05:
    print("ANOVA")
else:
    print("Kruskal-Wallis")
```

```
Kruskal-Wallis
```

[89]:
```
#creating boolean series to make each groups "male", "female", "unknown"
```

[90]:
```
male_rent =training_cleaned[training_cleaned['gender'] == 'm']['rent']
female_rent = training_cleaned[training_cleaned['gender'] == 'f']['rent']
unknown_rent = training_cleaned[training_cleaned['gender'] == 'u']['rent']
```

[91]:
```
from scipy.stats import kruskal
```

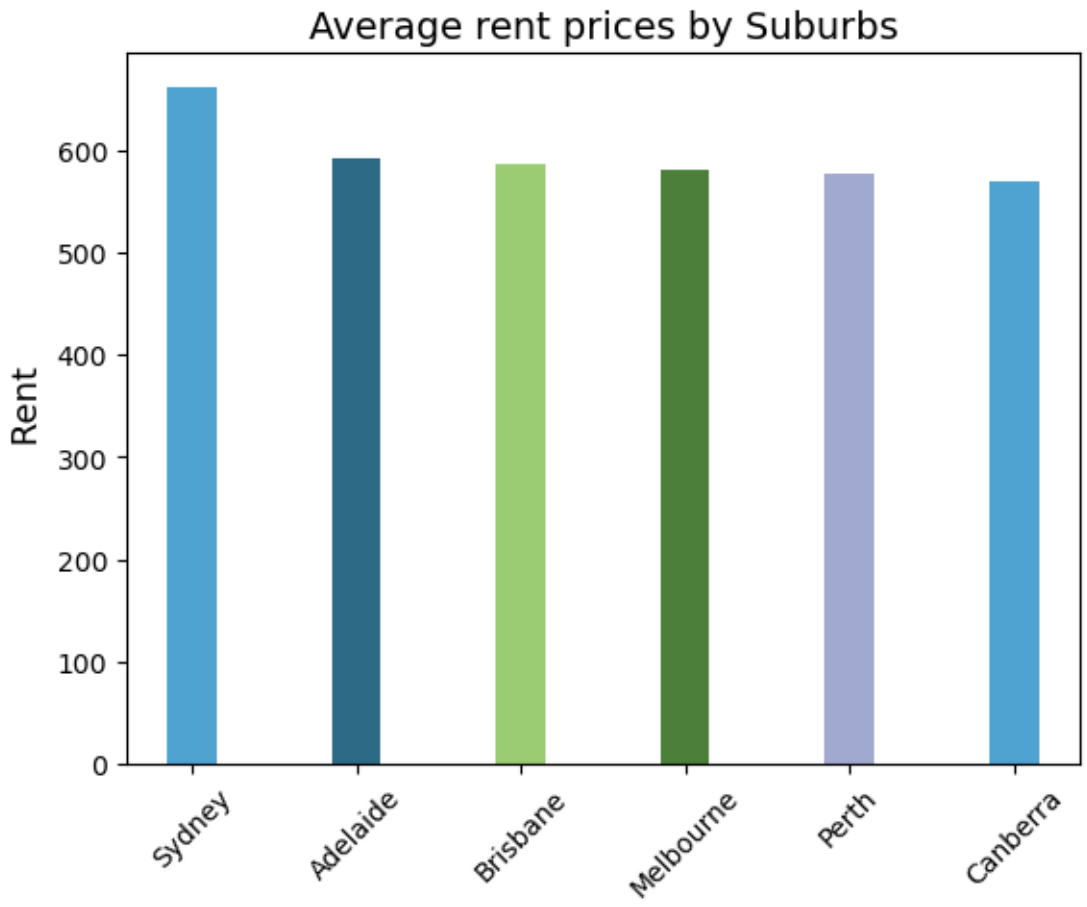
[92]:
```
kruskal_stat , kruskal_p_value = kruskal(male_rent, female_rent, unknown_rent)
#interpretation
if kruskal_p_value <= 0.05:
    print("There is a significant difference , reject hypothesis.")
else:
    print("There is no significant difference, failed to reject hypothesis.")
```

```
There is no significant difference, failed to reject hypothesis.
```

10 CONCLUSION: GENDER DO NOT AFFECT THE RENTAL PRICES

[93]:
```
#visualising rental prices by suburbs
```

[94]:
```
training_df.groupby('suburb')['rent'].mean().sort_values(ascending = False).
 ↪plot(kind = 'bar', width = 0.3,
                                        color =['#4FA3D1', '#2D6A85',␣
 ↪'#9BCB72', '#4B7F3A', '#A2A9D0'])
plt.title('Average rent prices by Suburbs', fontsize =14)
plt.xlabel(" ")
plt.ylabel(" Rent", fontsize = 13)
plt.xticks(rotation =45)
plt.show()
```

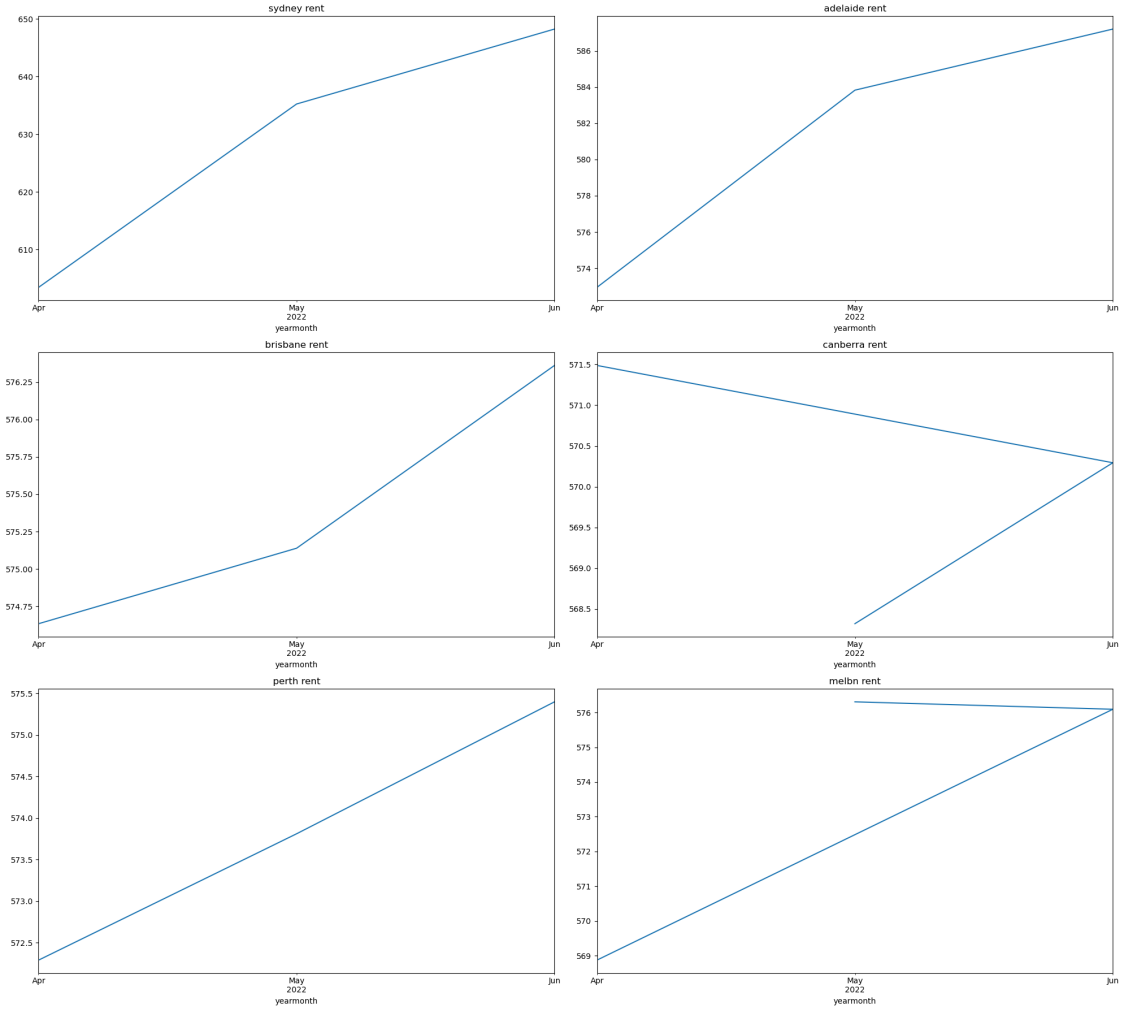


11 rental by suburb illustration

[95]:
```
#sydney has the highest average rental price.
```

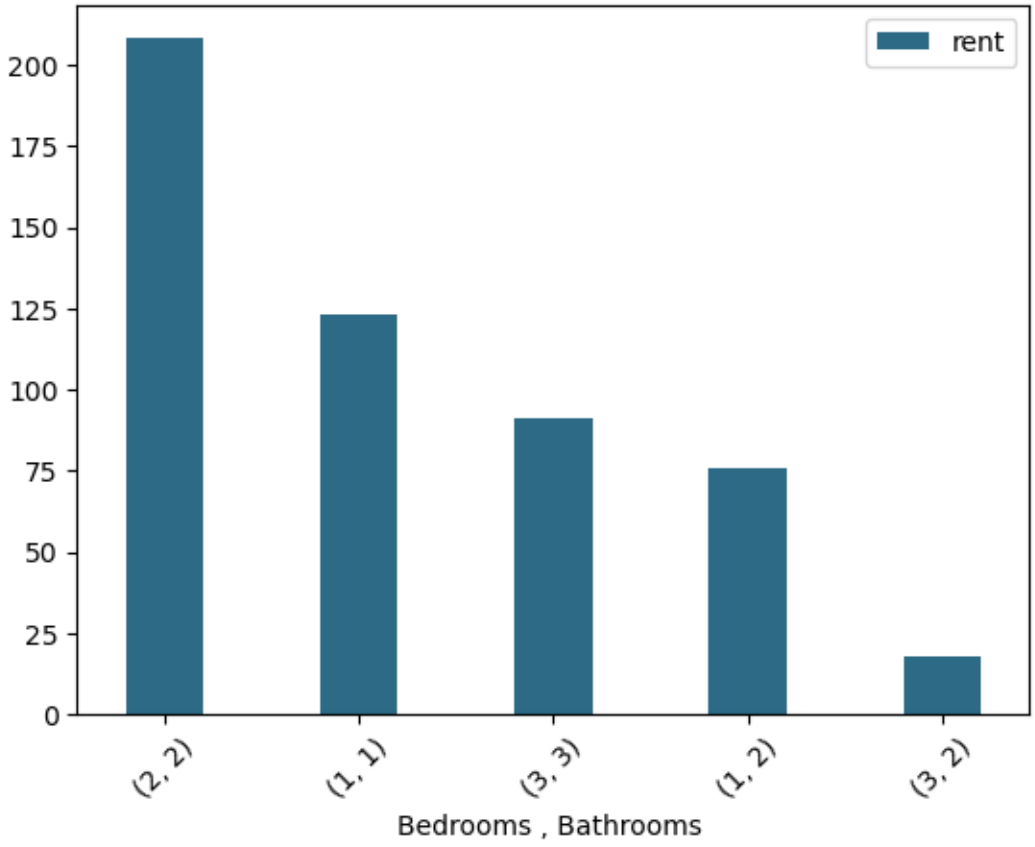
[96]:
```
fig ,axes = plt.subplots(3,2, figsize = (20 , 18))
sydney.groupby('yearmonth')['rent'].mean().sort_values(ascending =True).
 ↪plot(kind = 'line',ax =axes[0,0])
axes[0,0].set_title('sydney rent')

adelaide.groupby('yearmonth')['rent'].mean().sort_values(ascending =True).
 ↪plot(kind = 'line',ax =axes[0,1])
axes[0,1].set_title('adelaide rent')

brisbn.groupby('yearmonth')['rent'].mean().sort_values(ascending =True).
 ↪plot(kind = 'line',ax =axes[1,0])
axes[1,0].set_title('brisbane rent')
```

```
canb.groupby('yearmonth')['rent'].mean().sort_values(ascending =True).plot(kind␣
 ↪= 'line',ax =axes[1,1])
axes[1,1].set_title('canberra rent')

perth.groupby('yearmonth')['rent'].mean().sort_values(ascending =True).
 ↪plot(kind = 'line',ax =axes[2,0])
axes[2,0].set_title('perth rent')

melbn.groupby('yearmonth')['rent'].mean().sort_values(ascending =True).
 ↪plot(kind = 'line',ax =axes[2,1])
axes[2,1].set_title('melbn rent')

plt.tight_layout()

plt.show()
```



[97]:
```
#top 5 house bed bath in sydney in dataset
```

[98]:
```
bed_bath =sydney.groupby(['number_of_bedrooms','number_of_bathrooms' ])['rent'].
 ↪size().sort_values(
    ascending = False).head(5).reset_index()
bed_bath.set_index(['number_of_bedrooms', 'number_of_bathrooms']).plot(kind =␣
 ↪'bar', width = 0.4,
                                                color = '#2D6A85')
plt.xticks(rotation = 45)
plt.xlabel("Bedrooms , Bathrooms ")
plt.show()
```



12 average house prices by bedroom and bathroom

13 SYDNEY

[99]:
```
plt.figure(figsize = (12,8))
sydney.groupby(['number_of_bedrooms', 'number_of_bathrooms'])['rent'].mean().
 ↪sort_values(ascending =False).plot(
    kind = 'bar', color =['#4FA3D1', '#2D6A85', '#9BCB72', '#4B7F3A',␣
 ↪'#A2A9D0'] )
plt.xticks(rotation = 45)
plt.title(" AVG. RENT PRICES OF SYDNEY BY BEDROOM & BATHROOM ", fontsize =14)
plt.xlabel(" BEDROOMS , BATHROOMS", fontsize =18)
plt.ylabel(' WEEKLY RENT', fontsize =18)
plt.show()
```

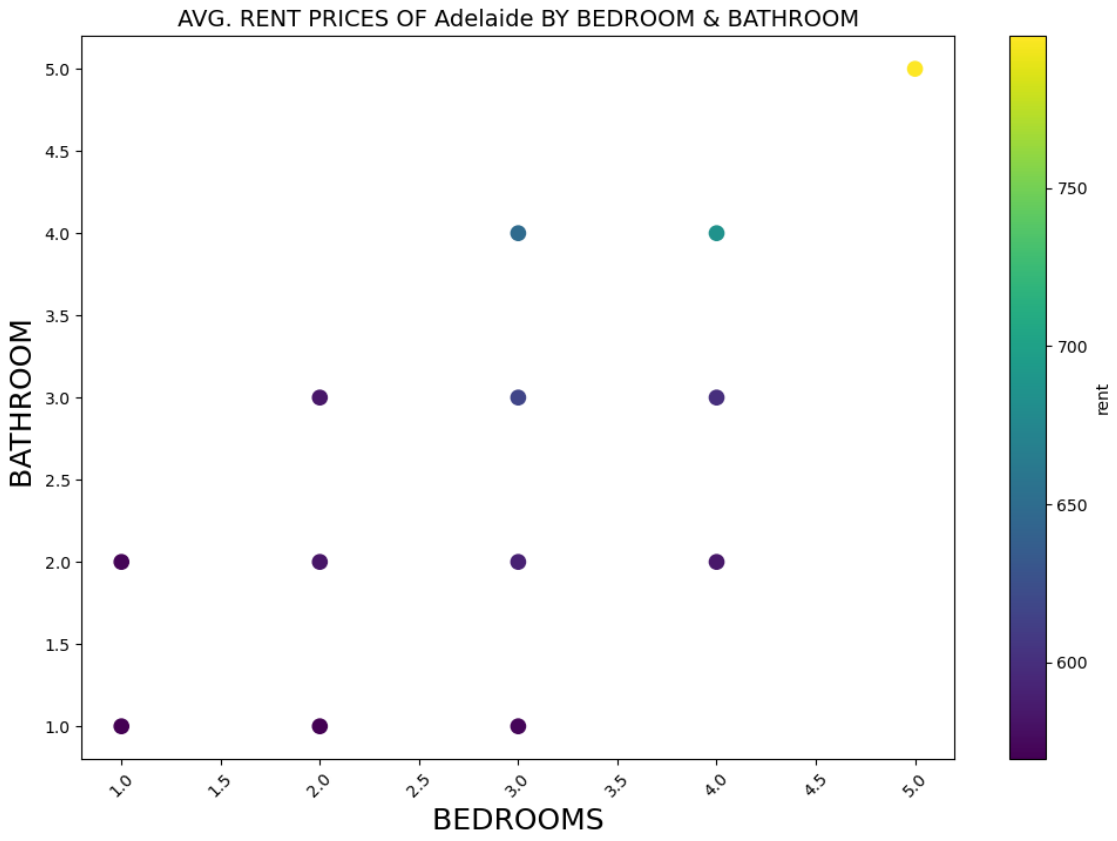
14 MELBOURNE

[100]:
```
plt.figure(figsize = (12,8))
melbn.groupby(['number_of_bedrooms', 'number_of_bathrooms'])['rent'].mean().
 ↪sort_values(ascending =False).plot(
```

```
    kind = 'bar', color =['#4FA3D1', '#2D6A85', '#9BCB72', '#4B7F3A',␣
 ↪'#A2A9D0'] )
plt.xticks(rotation = 45)
plt.title(" AVG. RENT PRICES OF Melbn BY BEDROOM & BATHROOM ", fontsize =14)
plt.xlabel(" BEDROOMS , BATHROOMS", fontsize =18)
plt.ylabel(' WEEKLY RENT', fontsize =18)
plt.show()
```

15 Adelaide

[101]:
```
adelaide_bed_bath = adelaide.groupby(['number_of_bedrooms',␣
 ↪'number_of_bathrooms'])['rent'].mean().sort_values(ascending =False).
 ↪reset_index()
adelaide_bed_bath.plot.scatter( x = 'number_of_bedrooms', y =␣
 ↪'number_of_bathrooms', c = 'rent',s = 78, figsize = (12,8) )
plt.xticks(rotation = 45)
plt.title(" AVG. RENT PRICES OF Adelaide BY BEDROOM & BATHROOM ", fontsize =14)
plt.xlabel(" BEDROOMS ", fontsize =18)
plt.ylabel('BATHROOM ',fontsize = 18)
plt.show()
```



[102]:
```
#checking whether Rents by suburbs are statistically significant
# Null Hypothesis - Suburbs have no significant impact on rent prices.
#Alternative hypothesis - Suburbs have significant impact on rent prices.
```

[103]:
```
from scipy.stats import kruskal
```

[104]:
```
syndey_rent = training_cleaned[training_cleaned['suburb'] == 'Sydney']['rent']
perth_rent = training_cleaned[training_cleaned['suburb'] == 'Perth']['rent']
canberra_rent = training_cleaned[training_cleaned['suburb'] ==␣
 ↪'Canberra']['rent']
melbourne_rent = training_cleaned[training_cleaned['suburb'] ==␣
 ↪'Melbourne']['rent']
brisbane_rent = training_cleaned[training_cleaned['suburb'] ==␣
 ↪'Brisbane']['rent']
adelaide_rent = training_cleaned[training_cleaned['suburb'] ==␣
 ↪'Adelaide']['rent']
```

[105]:
```
kk_stat , kk_p_value = kruskal (syndey_rent, perth_rent, canberra_rent,␣
 ↪melbourne_rent, brisbane_rent, adelaide_rent )
```

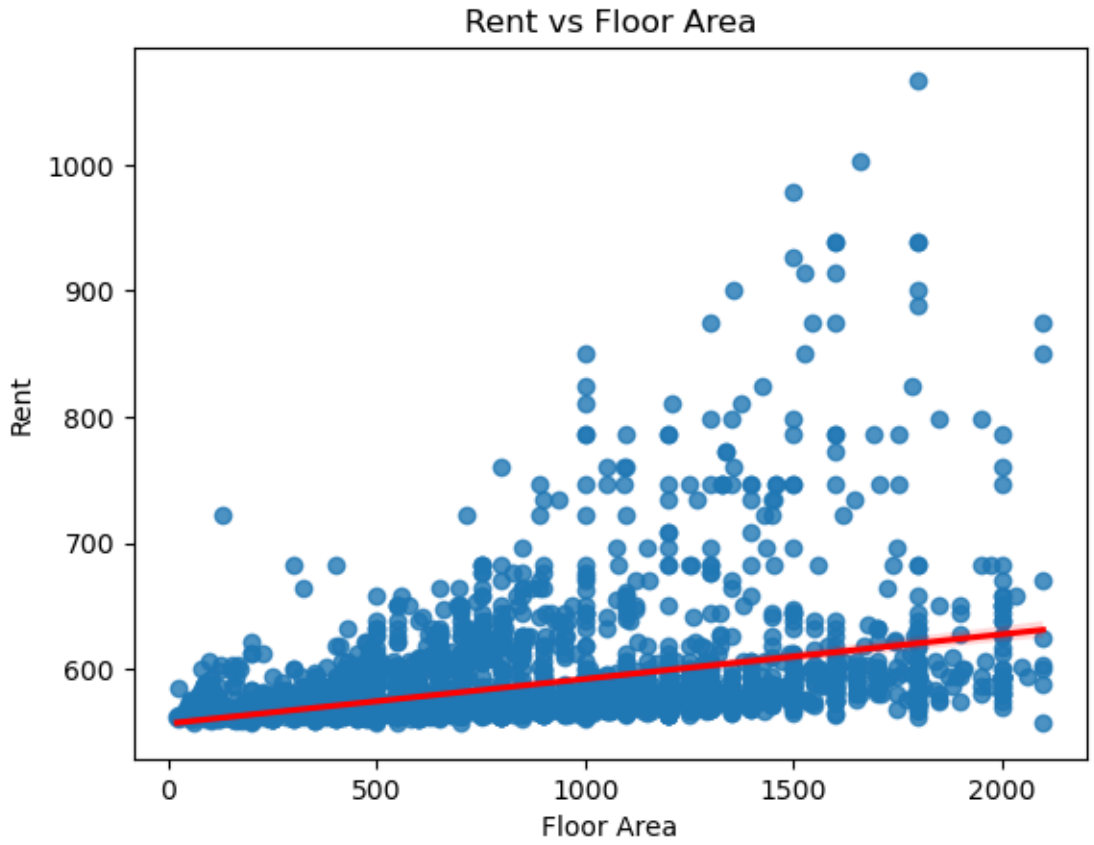
[106]:
```
if kk_p_value < 0.05:
    print('reject the null hypothesis')
else:
    print('Cannot Reject the null hopthesis')
```

```
reject the null hypothesis
```

16 the above test conducted implies that suburbs have a significant impact on the rent prices.

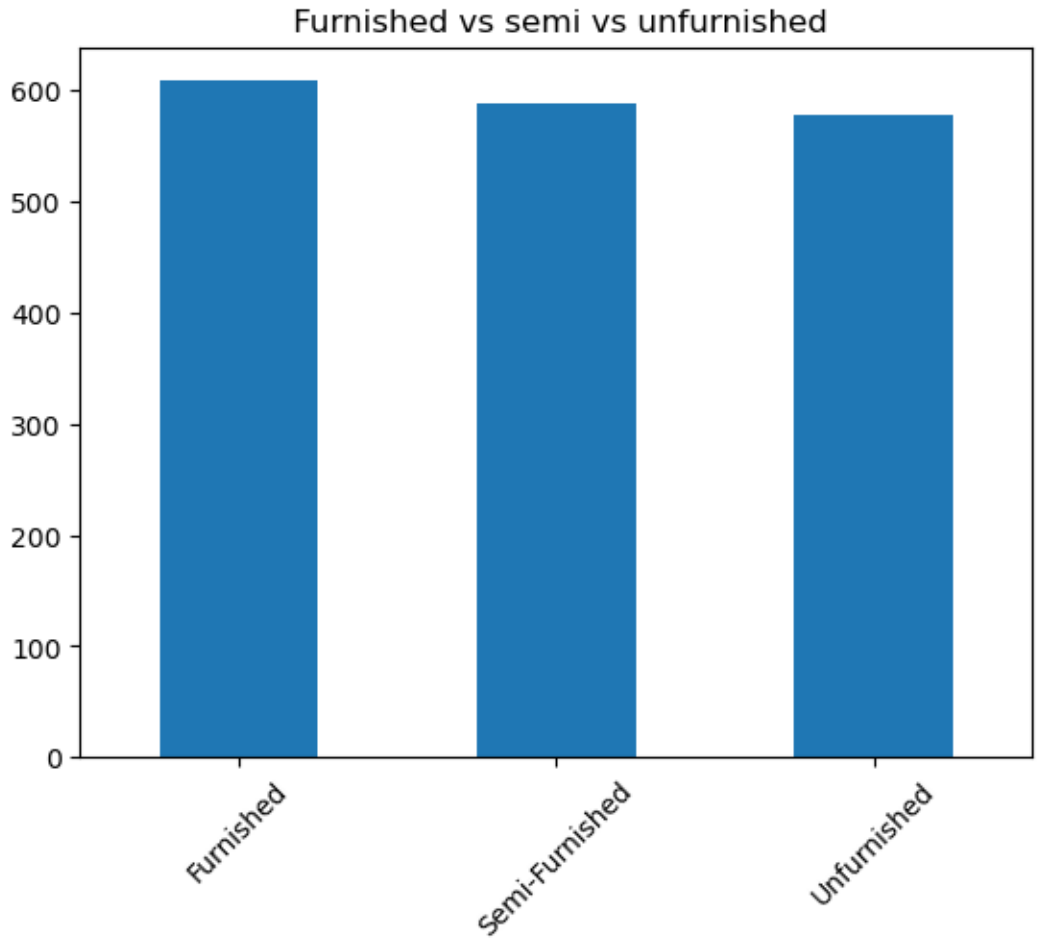
[107]:
```
#understanding the relationship of floor area vs rent
```

[108]:
```
sns.regplot(data=training_cleaned, x='floor_area', y='rent' ,line_kws =␣
↪{'color' :  'red'}  )
plt.title('Rent vs Floor Area')
plt.xlabel('Floor Area')
plt.ylabel('Rent')
plt.show()
```



[109]:
```
#average rent by furnish ,semi-furnish, unfurnished
```

[110]:
```
training_cleaned.groupby(['furnished'])['rent'].mean().plot(kind = 'bar')
plt.title(" Furnished vs semi vs unfurnished")
plt.xticks(rotation =45)
plt.xlabel(" ")
plt.show()
```



[111]:
```
training_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 3433
Data columns (total 26 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   advertised_date      3316 non-null   datetime64[ns]
```

```
 1   number_of_bedrooms   3316 non-null   int64
 2   rent                 3316 non-null   float64
 3   floor_area           3316 non-null   int64
 4   level                3316 non-null   object
 5   suburb               3316 non-null   object
 6   furnished            3316 non-null   object
 7   tenancy_preference   3316 non-null   object
 8   number_of_bathrooms  3316 non-null   int64
 9   point_of_contact     3316 non-null   object
 10  secondary_address    3316 non-null   object
 11  building_number      3316 non-null   int64
 12  street_name          3316 non-null   object
 13  street_suffix        3316 non-null   object
 14  prefix               2197 non-null   object
 15  first_name           3316 non-null   object
 16  last_name            3315 non-null   object
 17  gender               3316 non-null   object
 18  phone_number         3316 non-null   object
 19  email                3316 non-null   object
 20  yearmonth            3316 non-null   period[M]
 21  advertised_year      3316 non-null   int32
 22  advertised_month     3316 non-null   int32
 23  advertised_day       3316 non-null   int32
 24  current_level        3316 non-null   float64
 25  total_level          3316 non-null   object
dtypes: datetime64[ns](1), float64(2), int32(3), int64(4), object(15),
period[M](1)
memory usage: 789.6+ KB
```

[112]:
```
# <Student to fill this section>
```

[113]:
```
# @title Training Set Insights

wgt_eda_training_set_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Training Set Insights:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_eda_training_set_insights
```

[113]:
```
Textarea(value='', description='Training Set Insights:',
layout=Layout(height='100%', width='auto'), placehold…
```

16.0.1 C.3 Explore Validation Set

You can add more cells in this section

[114]:
```
# <Student to fill this section>
```

[115]:
```
validation_df
```

[115]:
```
     advertised_date  number_of_bedrooms   rent  floor_area  \
0         2022-06-13                   2  571.0         560
1         2022-06-04                   2  683.0         750
2         2022-04-29                   3  574.0         950
3         2022-05-18                   1  565.0         500
4         2022-04-28                   2  565.0         600
…                ...                 ...    ...         ...
1315      2022-06-29                   3  581.0        1100
1316      2022-07-02                   3  623.0        2300
1317      2022-06-28                   3  594.0         214
1318      2022-06-28                   1  562.0         500
1319      2022-06-28                   3  574.0        1500

                         level     suburb       furnished tenancy_preference  \
0              Ground out of 1  Melbourne  Semi-Furnished             Family
1     Upper Basement out of 30     Sydney     Unfurnished   Bachelors/Family
2              Ground out of 3   Adelaide     Unfurnished   Bachelors/Family
3                   2 out of 2     Sydney  Semi-Furnished          Bachelors
4                   2 out of 3   Brisbane  Semi-Furnished   Bachelors/Family
…                          ...        ...             ...                ...
1315                2 out of 5      Perth  Semi-Furnished   Bachelors/Family
1316                1 out of 5      Perth       Furnished          Bachelors
1317                2 out of 2      Perth       Furnished          Bachelors
1318           Ground out of 1      Perth       Furnished   Bachelors/Family
1319    Lower Basement out of 2      Perth  Semi-Furnished             Family

      number_of_bathrooms point_of_contact secondary_address  building_number  \
0                       2    Contact Owner          Level 1                1
1                       2    Contact Agent                1/               31
2                       2    Contact Owner          Unit 37               89
3                       1    Contact Owner               16/              82
4                       2    Contact Owner          Flat 64                9
…                     ...              ...                ...              ...
1315                    3    Contact Owner         Apt. 393                2
1316                    3    Contact Agent          Level 6              262
1317                    4    Contact Owner          Level 6              301
1318                    1    Contact Owner        Suite 718                3
1319                    3    Contact Owner        Suite 748               28

            street_name street_suffix prefix first_name last_name gender  \
```

```
0        Baldwin Towers       Footway    NaN        Jay    Glover      u
1        Cox Fire Track       Lookout    Dr.   Danielle      Tran      f
2       Davidson Ground          Part    NaN     Ashley   Pacheco      u
3       Fitzpatrick Key       Heights    NaN   Victoire     Weber      u
4          Heidi Access          Mews   Mrs.      Kerry      Koch      f
…                   …           …     …        …       …     …
1315       Wilson Elbow         Round    NaN      Scott    Warren      u
1316  Roberson Roadside         Brace   Mrs.  Christina   Roberts      f
1317    Rebecca Parkway         Plaza   Mrs.     Kimaya     Bobal      f
1318     Gregory Subway          Mall   Mrs.     Andrea      Wood      f
1319      Adam Crossing         Close   Mrs.     Nicole       May      f

         phone_number                        email
0        (03)08687820   brettkennedy@example.net
1      (03)-0313-6072         dana35@example.net
2        08-9358-6662       justin89@example.org
3      (02).9817.8199  pruittmichael@example.net
4           4124.0210    hansendiana@example.com
…                 …                          …
1315     0414.594.227          nayala@example.net
1316  +61-495-764-167        zjacobs@example.com
1317  +61.434.281.837        rharper@example.org
1318  +61-475-031-953        orivera@example.net
1319        8233 8936        kelli49@example.com

[1320 rows x 20 columns]
```

[116]:
```
validation_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1320 entries, 0 to 1319
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   advertised_date      1320 non-null   object
 1   number_of_bedrooms   1320 non-null   int64
 2   rent                 1320 non-null   float64
 3   floor_area           1320 non-null   int64
 4   level                1320 non-null   object
 5   suburb               1320 non-null   object
 6   furnished            1320 non-null   object
 7   tenancy_preference   1320 non-null   object
 8   number_of_bathrooms  1320 non-null   int64
 9   point_of_contact     1320 non-null   object
 10  secondary_address    1320 non-null   object
 11  building_number      1320 non-null   int64
 12  street_name          1320 non-null   object
```

```
 13  street_suffix        1320 non-null   object
 14  prefix               855 non-null    object
 15  first_name           1320 non-null   object
 16  last_name            1319 non-null   object
 17  gender               1320 non-null   object
 18  phone_number         1320 non-null   object
 19  email                1320 non-null   object
dtypes: float64(1), int64(4), object(15)
memory usage: 206.4+ KB
```

[117]:
```
validation_df.duplicated().sum()
```

[117]:
```
0
```

[118]:
```
#change to datetime
```

[119]:
```
validation_df['advertised_date'] = pd.
 ↪to_datetime(validation_df['advertised_date'])
```

[120]:
```
#extracting same dd,yy,mm like done in training
```

[121]:
```
validation_df['advertised_year'] = validation_df['advertised_date'].dt.year
validation_df['advertised_month'] = validation_df['advertised_date'].dt.month
validation_df['advertised_day'] = validation_df['advertised_date'].dt.day
```

[122]:
```
# separting level into current_level and total_level
```

[123]:
```
validation_df['level'].unique()
```

[123]:
```
array(['Ground out of 1', 'Upper Basement out of 30', 'Ground out of 3',
       '2 out of 2', '2 out of 3', '1 out of 2', '14 out of 23',
       '3 out of 4', '2 out of 5', '2 out of 4', 'Ground out of 7',
       '1 out of 3', '3 out of 5', 'Ground out of 2', '6 out of 7',
       '1 out of 1', 'Ground out of 4', '1 out of 10', '3 out of 10',
       '4 out of 20', '5 out of 21', '2 out of 22', '11 out of 25',
       'Upper Basement out of 3', '3 out of 9', '5 out of 7',
       '8 out of 5', '18 out of 19', '2 out of 6', '1 out of 5',
       '7 out of 8', '1 out of 6', '1 out of 4', '19 out of 85',
       '2 out of 7', '28 out of 31', '2 out of 13', 'Ground out of 5',
       '4 out of 26', '4 out of 10', '5 out of 8', '7 out of 9',
       '15 out of 23', '4 out of 7', '1 out of 7', '4 out of 4',
       '7 out of 20', '10 out of 18', '3 out of 7', '6 out of 12',
       '7 out of 19', '15 out of 28', '8 out of 17', '5 out of 14',
       '3 out of 3', '1 out of 8', '4 out of 9', '5 out of 5',
       '30 out of 58', '6 out of 11', '18 out of 33', '15 out of 15',
       '12 out of 19', '11 out of 13', '3 out of 6', '8 out of 9',
       '5 out of 18', '9 out of 15', '4 out of 8', '12 out of 18',
```

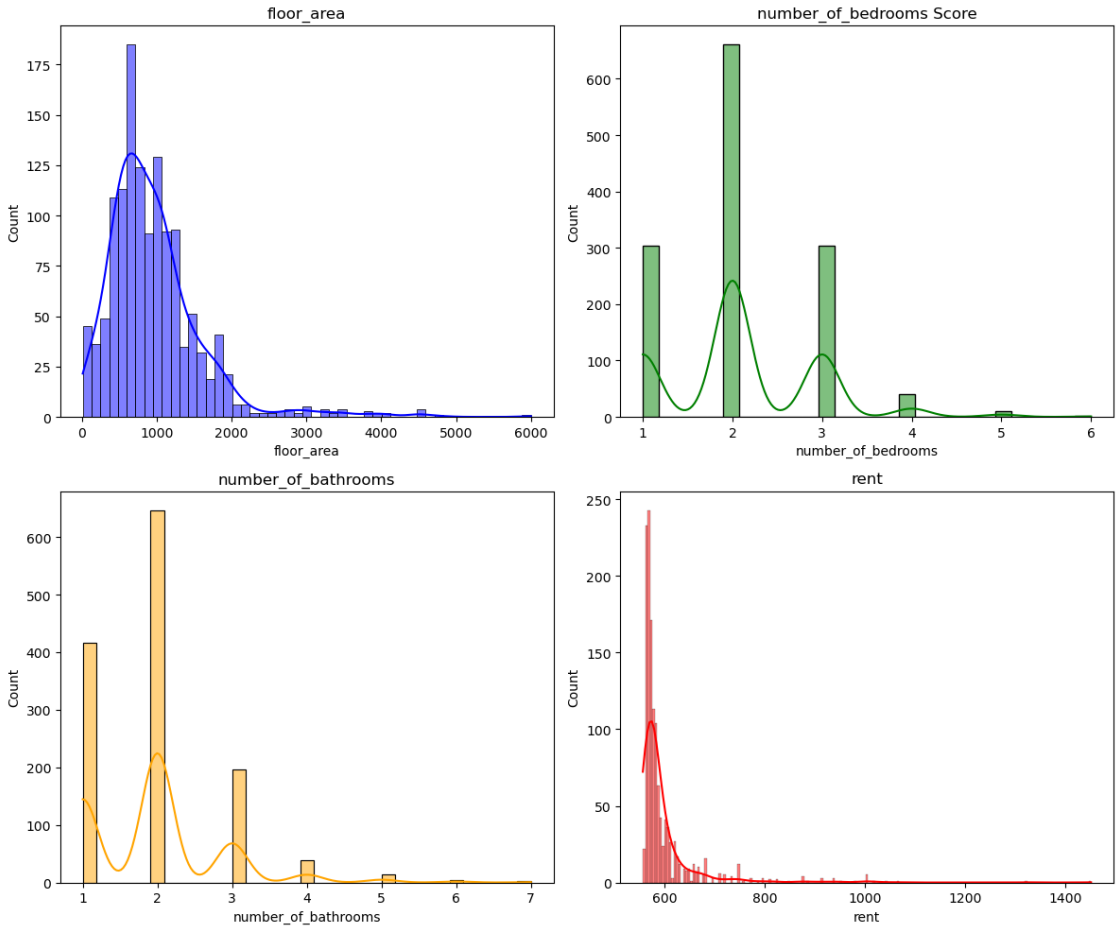
```
'4 out of 5', '9 out of 12', '9 out of 13', 'Ground out of 8',
'10 out of 14', '4 out of 12', '9 out of 38', 'Ground out of 12',
'15 out of 60', '8 out of 16', '4 out of 6', '5 out of 12',
'5 out of 6', '65 out of 76', '10 out of 13', '45 out of 77',
'10 out of 10', '18 out of 32', '8 out of 45', '4 out of 14',
'5 out of 15', '15 out of 17', '10 out of 23', '14 out of 18',
'20 out of 32', '14 out of 14', '6 out of 8', '12 out of 13',
'11 out of 21', '6 out of 23', '1 out of 16', '8 out of 10',
'3 out of 8', '25 out of 52', '35 out of 55', '10 out of 19',
'9 out of 35', '20 out of 22', '60 out of 77', '23 out of 35',
'7 out of 7', '15 out of 18', '5 out of 20', '16 out of 36',
'4 out of 31', '2 out of 12', '1 out of 12', '6 out of 20',
'24 out of 60', '2 out of 14', '5 out of 24', '18 out of 24',
'11 out of 16', '8 out of 22', 'Upper Basement out of 16',
'2 out of 1', 'Upper Basement out of 4', '17 out of 31',
'19 out of 24', '25 out of 41', '16 out of 23', '13 out of 15',
'5 out of 19', '10 out of 12', '15 out of 31', '7 out of 12',
'8 out of 20', '7 out of 23', '4 out of 11', '16 out of 31',
'10 out of 22', '7 out of 21', '17 out of 27', '10 out of 24',
'7 out of 10', '6 out of 10', '18 out of 22', '15 out of 16',
'6 out of 14', '5 out of 9', '33 out of 42', '11 out of 24',
'26 out of 42', '14 out of 22', '17 out of 24', '15 out of 20',
'12 out of 20', '17 out of 29', '10 out of 31', '11 out of 12',
'13 out of 14', '7 out of 11', '12 out of 27', '6 out of 15',
'14 out of 15', '8 out of 11', '25 out of 28', '12 out of 17',
'4 out of 15', '15 out of 43', '13 out of 21', '9 out of 55',
'49 out of 55', '21 out of 23', '23 out of 23', '11 out of 27',
'5 out of 17', '11 out of 15', 'Upper Basement out of 7',
'19 out of 33', '2 out of 8', '6 out of 13', '18 out of 23',
'4 out of 13', '3 out of 12', '24 out of 24', 'Ground out of 16',
'11 out of 19', '8 out of 13', 'Lower Basement out of 2',
'14 out of 30', '20 out of 20', '9 out of 9', '7 out of 18',
'1 out of 9', '10 out of 20', '15 out of 30', '12 out of 30',
'8 out of 14'], dtype=object)
```

[124]:
```
validation_df['current_level'] = validation_df['level'].apply(lambda x: -1 if␣
↪'Lower Basement' in x else
                                                               - 0.5 if 'Upper␣
↪Basement' in x else
                                                                0 if 'Ground' in␣
↪x else
                                                               int(x.split('␣
↪')[0]))
```

[125]:
```
validation_df['total_level'] = validation_df['level'].str.extract(r'out of␣
↪(\d+)')
```

[126]:
```
#converting total_level to float
validation_df['total_level'] = validation_df['total_level'].astype(float)
```

[127]:
```
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# Plot the histograms for each of the specified columns
sns.histplot(validation_df['floor_area'], ax=axes[0, 0], kde=True, color='blue')
axes[0, 0].set_title('floor_area')

sns.histplot(validation_df['number_of_bedrooms'], ax=axes[0, 1], kde=True,␣
 ↪color='green')
axes[0, 1].set_title('number_of_bedrooms Score')

sns.histplot(validation_df['number_of_bathrooms'], ax=axes[1, 0], kde=True,␣
 ↪color='orange')
axes[1, 0].set_title('number_of_bathrooms')

sns.histplot(validation_df['rent'], ax=axes[1, 1], kde=True, color='red')
axes[1, 1].set_title('rent')

# Adjust layout for better spacing
plt.tight_layout()

# Show the plot
plt.show()
```



[128]:
```
validation_df[validation_df['floor_area'] > 4000]
```

[128]:

	advertised_date	number_of_bedrooms	rent	floor_area	level	\
39	2022-06-08	4	709.0	4105	11 out of 25	
139	2022-05-27	5	1323.0	4500	7 out of 20	
303	2022-05-24	5	683.0	6000	9 out of 12	
930	2022-07-02	4	722.0	4800	11 out of 19	
947	2022-06-30	4	760.0	4500	4 out of 4	
1157	2022-06-28	6	914.0	4500	Ground out of 1	
1213	2022-06-30	4	875.0	4500	Ground out of 2	
1223	2022-07-02	5	939.0	4050	Ground out of 5	

	suburb	furnished	tenancy_preference	number_of_bathrooms	\
39	Brisbane	Semi-Furnished	Bachelors/Family	4	
139	Sydney	Furnished	Bachelors	5	
303	Melbourne	Semi-Furnished	Bachelors/Family	4	
930	Brisbane	Semi-Furnished	Bachelors/Family	4	
947	Brisbane	Unfurnished	Bachelors	4	

```
1157  Melbourne  Semi-Furnished          Bachelors                   5
1213      Perth  Semi-Furnished             Family                   4
1223      Perth     Unfurnished   Bachelors/Family                   4

     point_of_contact  … first_name  last_name gender    phone_number  \
39      Contact Agent  …     Alicia      Wolfe      f    03-4185-2520
139     Contact Agent  …      Gokul     Khatri      m    02.7642.8725
303     Contact Agent  …       Cory     Murphy      u    03-9280-5043
930     Contact Agent  …     Thomas      Ortiz      m    0440.741.808
947     Contact Agent  …      Jenna     Walker      f    08-0358-2545
1157    Contact Agent  …     Daniel     Newman      u  (02)-9983-7439
1213    Contact Agent  …     Meghan   Thompson      f    03.1857.7968
1223    Contact Owner  …     Amanda        Ray      f       6256.9986

                            email advertised_year advertised_month  \
39         thomasmeyer@example.org            2022                6
139           zroberts@example.com            2022                5
303           ohoffman@example.net            2022                5
930           asanders@example.org            2022                7
947      kathrynwilson@example.com            2022                6
1157        greenjames@example.org            2022                6
1213          qesparza@example.net            2022                6
1223  christopherknapp@example.net            2022                7

     advertised_day current_level total_level
39                8          11.0        25.0
139              27           7.0        20.0
303              24           9.0        12.0
930               2          11.0        19.0
947              30           4.0         4.0
1157             28           0.0         1.0
1213             30           0.0         2.0
1223              2           0.0         5.0

[8 rows x 25 columns]
```

[129]:
```
#handling outliers through IQR METHOD.
```

[130]:
```
q1 = validation_df['floor_area'].quantile(0.25)
q3 = validation_df['floor_area'].quantile(0.75)
iqr = q3-q1
```

[131]:
```
lower_bound = q1 - 1.5 *iqr
upper_bound = q3 +1.5 *iqr
```

[132]:
```
outliers = validation_df[(validation_df['floor_area'] < lower_bound) |␣
 ↪(validation_df['floor_area'] > upper_bound)]
```

```
outliers
```

[132]:

	advertised_date	number_of_bedrooms	rent	floor_area	\
22	2022-04-29	3	1003.0	3200	
37	2022-06-12	5	939.0	3000	
39	2022-06-08	4	709.0	4105	
64	2022-05-17	4	939.0	2200	
96	2022-06-21	3	645.0	2780	
132	2022-05-12	4	638.0	3200	
139	2022-05-27	5	1323.0	4500	
264	2022-05-31	5	1003.0	3250	
282	2022-06-10	3	613.0	2200	
300	2022-05-23	4	619.0	2900	
303	2022-05-24	5	683.0	6000	
312	2022-06-14	4	914.0	3800	
340	2022-06-22	3	574.0	2200	
423	2022-06-02	3	626.0	2430	
439	2022-06-16	4	734.0	2800	
485	2022-06-05	3	638.0	2210	
525	2022-05-21	4	875.0	2800	
557	2022-06-18	3	582.0	2200	
620	2022-06-18	4	888.0	3000	
633	2022-05-25	4	914.0	2500	
669	2022-05-12	6	600.0	3400	
726	2022-07-04	4	594.0	4000	
767	2022-07-04	5	1067.0	2308	
883	2022-06-29	5	952.0	2800	
896	2022-06-27	4	1451.0	3200	
897	2022-06-30	3	645.0	3354	
919	2022-06-29	4	1042.0	3500	
930	2022-07-02	4	722.0	4800	
947	2022-06-30	4	760.0	4500	
962	2022-07-04	3	626.0	2430	
1047	2022-07-04	4	609.0	2700	
1048	2022-06-30	3	722.0	3000	
1052	2022-07-04	4	601.0	2700	
1058	2022-06-30	3	683.0	2925	
1069	2022-06-28	2	597.0	2200	
1079	2022-06-28	4	722.0	3500	
1080	2022-06-28	4	747.0	4000	
1081	2022-06-28	3	651.0	2500	
1082	2022-06-28	4	747.0	3500	
1157	2022-06-28	6	914.0	4500	
1204	2022-07-05	4	658.0	3000	
1212	2022-06-28	4	651.0	3800	
1213	2022-06-30	4	875.0	4500	
1223	2022-07-02	5	939.0	4050	

1234	2022-07-05		3	583.0	3000
1242	2022-06-27		4	709.0	3800
1272	2022-07-02		5	619.0	3500
1284	2022-07-05		3	609.0	2671
1316	2022-07-02		3	623.0	2300

	level	suburb	furnished	tenancy_preference	\
22	1 out of 2	Adelaide	Semi-Furnished	Bachelors/Family	
37	2 out of 22	Sydney	Furnished	Bachelors/Family	
39	11 out of 25	Brisbane	Semi-Furnished	Bachelors/Family	
64	7 out of 8	Sydney	Furnished	Family	
96	4 out of 26	Perth	Semi-Furnished	Bachelors/Family	
132	18 out of 19	Brisbane	Semi-Furnished	Bachelors/Family	
139	7 out of 20	Sydney	Furnished	Bachelors	
264	12 out of 18	Sydney	Unfurnished	Family	
282	4 out of 4	Melbourne	Unfurnished	Bachelors/Family	
300	5 out of 7	Brisbane	Furnished	Bachelors/Family	
303	9 out of 12	Melbourne	Semi-Furnished	Bachelors/Family	
312	1 out of 4	Adelaide	Semi-Furnished	Bachelors/Family	
340	1 out of 2	Melbourne	Unfurnished	Bachelors	
423	7 out of 9	Perth	Semi-Furnished	Bachelors	
439	Ground out of 4	Adelaide	Semi-Furnished	Bachelors/Family	
485	Ground out of 2	Canberra	Unfurnished	Bachelors/Family	
525	2 out of 3	Adelaide	Unfurnished	Bachelors/Family	
557	Ground out of 2	Adelaide	Semi-Furnished	Bachelors/Family	
620	4 out of 4	Adelaide	Semi-Furnished	Bachelors/Family	
633	24 out of 60	Sydney	Semi-Furnished	Bachelors/Family	
669	Ground out of 2	Perth	Unfurnished	Bachelors/Family	
726	Ground out of 3	Canberra	Furnished	Bachelors/Family	
767	17 out of 31	Sydney	Furnished	Family	
883	19 out of 33	Sydney	Semi-Furnished	Bachelors/Family	
896	24 out of 24	Sydney	Furnished	Bachelors/Family	
897	Ground out of 16	Brisbane	Furnished	Bachelors/Family	
919	2 out of 4	Brisbane	Semi-Furnished	Bachelors	
930	11 out of 19	Brisbane	Semi-Furnished	Bachelors/Family	
947	4 out of 4	Brisbane	Unfurnished	Bachelors	
962	4 out of 5	Brisbane	Semi-Furnished	Bachelors/Family	
1047	2 out of 4	Adelaide	Semi-Furnished	Bachelors	
1048	1 out of 5	Adelaide	Semi-Furnished	Bachelors/Family	
1052	2 out of 3	Adelaide	Semi-Furnished	Bachelors/Family	
1058	1 out of 4	Adelaide	Semi-Furnished	Bachelors/Family	
1069	1 out of 1	Adelaide	Semi-Furnished	Family	
1079	1 out of 3	Adelaide	Semi-Furnished	Bachelors	
1080	3 out of 3	Adelaide	Semi-Furnished	Bachelors	
1081	1 out of 3	Adelaide	Semi-Furnished	Bachelors	
1082	3 out of 3	Adelaide	Semi-Furnished	Bachelors	
1157	Ground out of 1	Melbourne	Semi-Furnished	Bachelors	

```
1204        1 out of 2  Melbourne     Unfurnished          Bachelors
1212       3 out of 10      Perth  Semi-Furnished   Bachelors/Family
1213   Ground out of 2      Perth  Semi-Furnished             Family
1223   Ground out of 5      Perth     Unfurnished   Bachelors/Family
1234        3 out of 5      Perth       Furnished   Bachelors/Family
1242        1 out of 1      Perth  Semi-Furnished   Bachelors/Family
1272        2 out of 3      Perth     Unfurnished   Bachelors/Family
1284       2 out of 14      Perth  Semi-Furnished             Family
1316        1 out of 5      Perth       Furnished          Bachelors

      number_of_bathrooms point_of_contact  … first_name    last_name  \
22                      4    Contact Owner  …    Rebecca      Jimenez
37                      5    Contact Agent  …     Travis      Hampton
39                      4    Contact Agent  …     Alicia        Wolfe
64                      4    Contact Agent  …     Manuel       Cooper
96                      3    Contact Agent  …       Jack        Mccoy
132                     3    Contact Agent  …      Jamie      Schultz
139                     5    Contact Agent  …      Gokul       Khatri
264                     5    Contact Agent  …     Samuel        Hurst
282                     3    Contact Agent  …       Lisa       Fields
300                     4    Contact Owner  …     Amanda    Patterson
303                     4    Contact Agent  …       Cory       Murphy
312                     5    Contact Agent  …    Theresa       Carter
340                     3    Contact Owner  …      Allen     Gonzalez
423                     3    Contact Agent  …      Holly        Cline
439                     4    Contact Agent  …    Kathryn         Buck
485                     3    Contact Owner  …      Tyler       Wilson
525                     4    Contact Owner  …    Matthew       Nelson
557                     2    Contact Agent  …  Stephanie         Gill
620                     5    Contact Agent  …   Jonathan        Wolfe
633                     6    Contact Agent  …  Elizabeth        Smith
669                     7    Contact Owner  …     Lauren    Donaldson
726                     3    Contact Agent  …    Dhanush      Kothari
767                     5    Contact Agent  …      Peter       Jordan
883                     5    Contact Agent  …      Purab     Choudhry
896                     4    Contact Agent  …     Andrea         Gill
897                     3    Contact Agent  …    Dishani     Krishnan
919                     5    Contact Agent  …      Jenny       Martin
930                     4    Contact Agent  …     Thomas        Ortiz
947                     4    Contact Agent  …      Jenna       Walker
962                     3    Contact Owner  …  Christine      Ramirez
1047                    4    Contact Agent  …      Logan         Love
1048                    4    Contact Agent  …       John       Cooper
1052                    3    Contact Agent  …      James       Brooks
1058                    3    Contact Agent  …      Sarah        Quinn
1069                    2    Contact Owner  …      Tyler       Garcia
1079                    6    Contact Agent  …   Nicholas         Mayo
```

1080	7	Contact Agent	…	Steven	Woods	
1081	4	Contact Agent	…	Nomvula	Duze	
1082	6	Contact Agent	…	Dr.	Paul	
1157	5	Contact Agent	…	Daniel	Newman	
1204	4	Contact Agent	…	David	Fitzpatrick	
1212	4	Contact Agent	…	Lacey	Durham	
1213	4	Contact Agent	…	Meghan	Thompson	
1223	4	Contact Owner	…	Amanda	Ray	
1234	2	Contact Owner	…	Catherine	Yates	
1242	4	Contact Agent	…	Marie	Cohen	
1272	6	Contact Owner	…	Lisa	Brown	
1284	3	Contact Owner	…	Dhanush	Bakshi	
1316	3	Contact Agent	…	Christina	Roberts	

	gender	phone_number	email	advertised_year	\
22	f	7493.2263	andreaellis@example.com	2022	
37	m	0407-124-172	ginaparsons@example.org	2022	
39	f	03-4185-2520	thomasmeyer@example.org	2022	
64	m	02 7375 6683	westpaula@example.com	2022	
96	u	0481-593-709	barry42@example.org	2022	
132	f	+61.431.662.086	floydjessica@example.com	2022	
139	m	02.7642.8725	zroberts@example.com	2022	
264	m	+61.2.2186.0016	carlpatel@example.net	2022	
282	f	(02)97411827	lgordon@example.org	2022	
300	u	0241961692	jameshicks@example.org	2022	
303	u	03-9280-5043	ohoffman@example.net	2022	
312	u	03.3116.0664	nicholasbell@example.net	2022	
340	m	0435-327-862	lisa26@example.net	2022	
423	u	+61.482.589.638	kristy24@example.net	2022	
439	u	+61.2.2363.7428	brenda67@example.org	2022	
485	u	(03)41251215	lewispatricia@example.org	2022	
525	m	7795 2427	danielclark@example.com	2022	
557	m	(02).5571.4409	jeremyparker@example.com	2022	
620	m	0488 333 533	jonathan79@example.com	2022	
633	u	03 9916 7924	zschneider@example.net	2022	
669	f	(03)-4501-7490	gabrielle09@example.net	2022	
726	m	0264018245	thomaskaiser@example.com	2022	
767	m	08 2770 7697	richard89@example.net	2022	
883	u	9623 5954	sarahrodriguez@example.com	2022	
896	f	+61.445.335.832	jameswillis@example.org	2022	
897	u	+61.8.3821.3463	estone@example.net	2022	
919	f	0499-294-432	carpenterdavid@example.org	2022	
930	m	0440.741.808	asanders@example.org	2022	
947	f	08-0358-2545	kathrynwilson@example.com	2022	
962	u	(02)-4937-8304	qjohnson@example.org	2022	
1047	m	+61884193791	brownshelly@example.net	2022	
1048	m	(03).8178.2273	kingeric@example.com	2022	

1052	m	+61 425 101 092	cathyking@example.org	2022
1058	f	+61-435-654-008	christine36@example.net	2022
1069	m	6527-8703	rachelguzman@example.org	2022
1079	u	1234 9666	robertsonelizabeth@example.org	2022
1080	u	9284.7479	brendan25@example.com	2022
1081	u	03 2806 5721	sarah28@example.net	2022
1082	u	(08)-3760-1593	dmejia@example.com	2022
1157	u	(02)-9983-7439	greenjames@example.org	2022
1204	m	(07)-3485-9205	dean01@example.com	2022
1212	u	(07)01507784	grhodes@example.com	2022
1213	f	03.1857.7968	qesparza@example.net	2022
1223	f	6256.9986	christopherknapp@example.net	2022
1234	f	0476 156 723	lisarichardson@example.org	2022
1242	f	+61-478-780-949	linda11@example.org	2022
1272	u	5253 9187	atkinsandrea@example.net	2022
1284	f	6337-6844	lperez@example.net	2022
1316	f	+61-495-764-167	zjacobs@example.com	2022

	advertised_month	advertised_day	current_level	total_level
22	4	29	1.0	2.0
37	6	12	2.0	22.0
39	6	8	11.0	25.0
64	5	17	7.0	8.0
96	6	21	4.0	26.0
132	5	12	18.0	19.0
139	5	27	7.0	20.0
264	5	31	12.0	18.0
282	6	10	4.0	4.0
300	5	23	5.0	7.0
303	5	24	9.0	12.0
312	6	14	1.0	4.0
340	6	22	1.0	2.0
423	6	2	7.0	9.0
439	6	16	0.0	4.0
485	6	5	0.0	2.0
525	5	21	2.0	3.0
557	6	18	0.0	2.0
620	6	18	4.0	4.0
633	5	25	24.0	60.0
669	5	12	0.0	2.0
726	7	4	0.0	3.0
767	7	4	17.0	31.0
883	6	29	19.0	33.0
896	6	27	24.0	24.0
897	6	30	0.0	16.0
919	6	29	2.0	4.0
930	7	2	11.0	19.0

```
947                6             30           4.0          4.0
962                7              4           4.0          5.0
1047               7              4           2.0          4.0
1048               6             30           1.0          5.0
1052               7              4           2.0          3.0
1058               6             30           1.0          4.0
1069               6             28           1.0          1.0
1079               6             28           1.0          3.0
1080               6             28           3.0          3.0
1081               6             28           1.0          3.0
1082               6             28           3.0          3.0
1157               6             28           0.0          1.0
1204               7              5           1.0          2.0
1212               6             28           3.0         10.0
1213               6             30           0.0          2.0
1223               7              2           0.0          5.0
1234               7              5           3.0          5.0
1242               6             27           1.0          1.0
1272               7              2           2.0          3.0
1284               7              5           2.0         14.0
1316               7              2           1.0          5.0

[49 rows x 25 columns]
```

[133]:
```
#saving the validation_df without the outliers
```

[134]:
```
validation_df =validation_df[(validation_df['floor_area'] > lower_bound) &␣
 ↪(validation_df['floor_area'] < upper_bound)]
validation_df
```

[134]:
```
     advertised_date  number_of_bedrooms   rent  floor_area  \
0         2022-06-13                   2  571.0         560
1         2022-06-04                   2  683.0         750
2         2022-04-29                   3  574.0         950
3         2022-05-18                   1  565.0         500
4         2022-04-28                   2  565.0         600
…                …                   …     …           …
1314      2022-07-02                   2  581.0        1350
1315      2022-06-29                   3  581.0        1100
1317      2022-06-28                   3  594.0         214
1318      2022-06-28                   1  562.0         500
1319      2022-06-28                   3  574.0        1500

                         level     suburb       furnished tenancy_preference  \
0              Ground out of 1  Melbourne  Semi-Furnished             Family
1     Upper Basement out of 30     Sydney     Unfurnished   Bachelors/Family
2              Ground out of 3   Adelaide     Unfurnished   Bachelors/Family
```

```
3                   2 out of 2     Sydney  Semi-Furnished          Bachelors
4                   2 out of 3   Brisbane  Semi-Furnished   Bachelors/Family
…                          …          …             …                  …
1314               8 out of 14      Perth  Semi-Furnished          Bachelors
1315                2 out of 5      Perth  Semi-Furnished   Bachelors/Family
1317                2 out of 2      Perth       Furnished          Bachelors
1318           Ground out of 1      Perth       Furnished   Bachelors/Family
1319   Lower Basement out of 2      Perth  Semi-Furnished             Family

      number_of_bathrooms point_of_contact  … first_name  last_name gender  \
0                       2    Contact Owner  …        Jay     Glover      u
1                       2    Contact Agent  …   Danielle       Tran      f
2                       2    Contact Owner  …     Ashley    Pacheco      u
3                       1    Contact Owner  …   Victoire      Weber      u
4                       2    Contact Owner  …      Kerry       Koch      f
…                     …                 …  …         …          …    …
1314                    2    Contact Owner  …    Brandon   Robinson      m
1315                    3    Contact Owner  …      Scott     Warren      u
1317                    4    Contact Owner  …     Kimaya      Bobal      f
1318                    1    Contact Owner  …     Andrea       Wood      f
1319                    3    Contact Owner  …     Nicole        May      f

         phone_number                      email advertised_year  \
0        (03)08687820   brettkennedy@example.net            2022
1      (03)-0313-6072         dana35@example.net            2022
2        08-9358-6662       justin89@example.org            2022
3      (02).9817.8199  pruittmichael@example.net            2022
4           4124.0210    hansendiana@example.com            2022
…                 …                          …             …
1314     +61800919982     bobbywhite@example.net            2022
1315     0414.594.227          nayala@example.net            2022
1317  +61.434.281.837        rharper@example.org            2022
1318  +61-475-031-953        orivera@example.net            2022
1319        8233 8936        kelli49@example.com            2022

     advertised_month advertised_day current_level total_level
0                   6             13           0.0         1.0
1                   6              4          -0.5        30.0
2                   4             29           0.0         3.0
3                   5             18           2.0         2.0
4                   4             28           2.0         3.0
…                 …             …            …          …
1314                7              2           8.0        14.0
1315                6             29           2.0         5.0
1317                6             28           2.0         2.0
1318                6             28           0.0         1.0
1319                6             28          -1.0         2.0
```

```
[1271 rows x 25 columns]
```

[135]:
```
validation_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1271 entries, 0 to 1319
Data columns (total 25 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   advertised_date       1271 non-null   datetime64[ns]
 1   number_of_bedrooms    1271 non-null   int64
 2   rent                  1271 non-null   float64
 3   floor_area            1271 non-null   int64
 4   level                 1271 non-null   object
 5   suburb                1271 non-null   object
 6   furnished             1271 non-null   object
 7   tenancy_preference    1271 non-null   object
 8   number_of_bathrooms   1271 non-null   int64
 9   point_of_contact      1271 non-null   object
 10  secondary_address     1271 non-null   object
 11  building_number       1271 non-null   int64
 12  street_name           1271 non-null   object
 13  street_suffix         1271 non-null   object
 14  prefix                824 non-null    object
 15  first_name            1271 non-null   object
 16  last_name             1270 non-null   object
 17  gender                1271 non-null   object
 18  phone_number          1271 non-null   object
 19  email                 1271 non-null   object
 20  advertised_year       1271 non-null   int32
 21  advertised_month      1271 non-null   int32
 22  advertised_day        1271 non-null   int32
 23  current_level         1271 non-null   float64
 24  total_level           1271 non-null   float64
dtypes: datetime64[ns](1), float64(3), int32(3), int64(4), object(14)
memory usage: 243.3+ KB
```

[136]:
```
# @title Validation Set Insights

wgt_eda_validation_set_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Validation Set Insights:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
```

```
)
wgt_eda_validation_set_insights
```

[136]:
```
Textarea(value='', description='Validation Set Insights:',
layout=Layout(height='100%', width='auto'), placeho…
```

16.0.2 C.4 Explore Testing Set

You can add more cells in this section

[137]:
```
# <Student to fill this section>
```

17 TESTING SET

[138]:
```
testing_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1364 entries, 0 to 1363
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   advertised_date       1364 non-null   object
 1   number_of_bedrooms    1364 non-null   int64
 2   rent                  1364 non-null   float64
 3   floor_area            1364 non-null   int64
 4   level                 1364 non-null   object
 5   suburb                1364 non-null   object
 6   furnished             1364 non-null   object
 7   tenancy_preference    1364 non-null   object
 8   number_of_bathrooms   1364 non-null   int64
 9   point_of_contact      1364 non-null   object
 10  secondary_address     1364 non-null   object
 11  building_number       1364 non-null   int64
 12  street_name           1364 non-null   object
 13  street_suffix         1364 non-null   object
 14  prefix                877 non-null    object
 15  first_name            1364 non-null   object
 16  last_name             1364 non-null   object
 17  gender                1364 non-null   object
 18  phone_number          1364 non-null   object
 19  email                 1364 non-null   object
dtypes: float64(1), int64(4), object(15)
memory usage: 213.3+ KB
```

[139]:
```
testing_df.duplicated().sum()
```

[139]:
```
0
```

[140]:
```
#converting 'advertised_date' into datetime
```

[141]:
```
testing_df['advertised_date'] = pd.to_datetime(testing_df['advertised_date'] )
```

[142]:
```
#separating current level and total level column from column "level"
```

[143]:
```
testing_df['current_level'] = testing_df['level'].apply(lambda x: -1 if 'Lower␣
↪Basement' in x else
                                                          -0.5 if 'Upper␣
↪Basement' in x else
                                                           0 if 'Ground' in␣
↪x else
                                                          int(x.split('␣
↪')[0]))
```

[144]:
```
testing_df['total_level'] = testing_df['level'].str.extract(r' out of (\d+)')
```

[145]:
```
testing_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1364 entries, 0 to 1363
Data columns (total 22 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   advertised_date      1364 non-null   datetime64[ns]
 1   number_of_bedrooms   1364 non-null   int64
 2   rent                 1364 non-null   float64
 3   floor_area           1364 non-null   int64
 4   level                1364 non-null   object
 5   suburb               1364 non-null   object
 6   furnished            1364 non-null   object
 7   tenancy_preference   1364 non-null   object
 8   number_of_bathrooms  1364 non-null   int64
 9   point_of_contact     1364 non-null   object
 10  secondary_address    1364 non-null   object
 11  building_number      1364 non-null   int64
 12  street_name          1364 non-null   object
 13  street_suffix        1364 non-null   object
 14  prefix               877 non-null    object
 15  first_name           1364 non-null   object
 16  last_name            1364 non-null   object
 17  gender               1364 non-null   object
 18  phone_number         1364 non-null   object
 19  email                1364 non-null   object
 20  current_level        1364 non-null   float64
 21  total_level          1364 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(4), object(15)
```

```
memory usage: 234.6+ KB
```

[146]:
```
#notice how current level is float and total level as object, this will cause␣
 ↪inconsitency while building model.
#We change it to float
```

[147]:
```
testing_df['total_level'] = testing_df['total_level'].astype(float)
```

[148]:
```
testing_df.dtypes
```

[148]:
```
advertised_date        datetime64[ns]
number_of_bedrooms              int64
rent                          float64
floor_area                      int64
level                          object
suburb                         object
furnished                      object
tenancy_preference             object
number_of_bathrooms             int64
point_of_contact               object
secondary_address              object
building_number                 int64
street_name                    object
street_suffix                  object
prefix                         object
first_name                     object
last_name                      object
gender                         object
phone_number                   object
email                          object
current_level                 float64
total_level                   float64
dtype: object
```

[149]:
```
#extracting year month and day column "advertised time"
```

[150]:
```
testing_df['advertised_year'] = testing_df['advertised_date'].dt.year
testing_df['advertised_month'] = testing_df['advertised_date'].dt.month
testing_df['advertised_day'] = testing_df['advertised_date'].dt.day
```

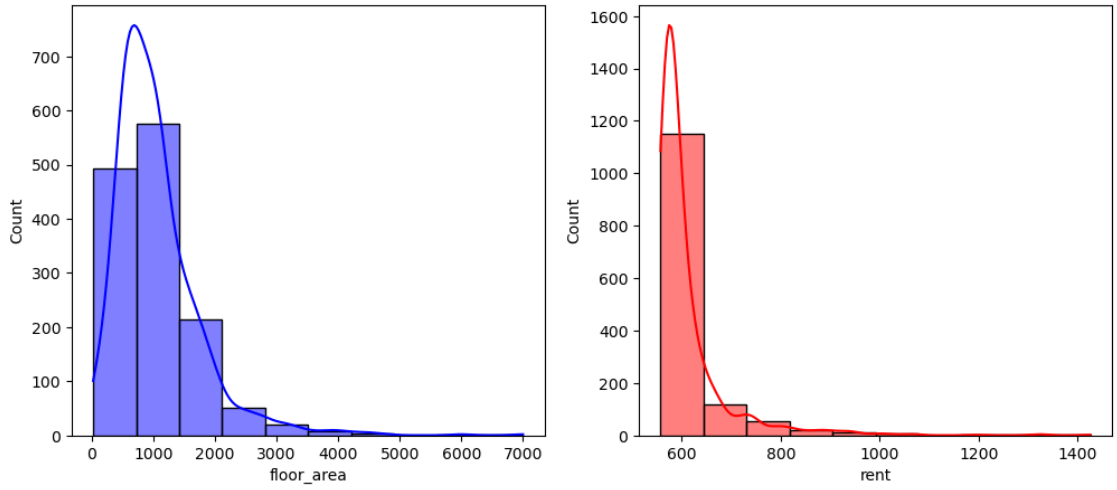
[151]:
```
testing_df['total_level'].astype(float)
```

[151]:
```
0        1.0
1       30.0
2        3.0
3        2.0
4        3.0
```

```
        …
1359     2.0
1360     5.0
1361     4.0
1362     5.0
1363    34.0
Name: total_level, Length: 1364, dtype: float64
```

[152]:
```
fig, axes = plt.subplots(1,2, figsize = (12,5))
sns.histplot(testing_df['floor_area'],ax = axes[0], bins = 10, kde = True,␣
 ↪color = 'blue')
sns.histplot(testing_df['rent'], bins = 10, ax = axes[1], kde = True, color =␣
 ↪'red')
```

[152]:
```
<Axes: xlabel='rent', ylabel='Count'>
```



[153]:
```
testing_df[testing_df['rent'] > 1000]
```

[153]:

	advertised_date	number_of_bedrooms	rent	floor_area	\
22	2022-04-29	3	1003.0	3200	
78	2022-06-08	3	1003.0	1663	
139	2022-05-27	5	1323.0	4500	
264	2022-05-31	5	1003.0	3250	
811	2022-07-09	5	1195.0	3900	
828	2022-07-09	4	1042.0	2800	
911	2022-07-09	4	1067.0	2080	
921	2022-07-07	4	1426.0	1962	
1160	2022-07-06	2	1323.0	950	
1308	2022-07-06	4	1067.0	7000	

```
                        level     suburb       furnished tenancy_preference  \
22                  1 out of 2   Adelaide  Semi-Furnished   Bachelors/Family
78                19 out of 85     Sydney  Semi-Furnished   Bachelors/Family
139                7 out of 20     Sydney       Furnished          Bachelors
264               12 out of 18     Sydney     Unfurnished             Family
811                 4 out of 6     Sydney       Furnished   Bachelors/Family
828               50 out of 75     Sydney  Semi-Furnished   Bachelors/Family
911               34 out of 46     Sydney  Semi-Furnished   Bachelors/Family
921               18 out of 20     Sydney  Semi-Furnished   Bachelors/Family
1160                1 out of 1  Melbourne     Unfurnished          Bachelors
1308  Lower Basement out of 2      Perth  Semi-Furnished   Bachelors/Family

      number_of_bathrooms point_of_contact  …   first_name  last_name  \
22                      4    Contact Owner  …      Rebecca    Jimenez
78                      2    Contact Agent  …    Christine      Baker
139                     5    Contact Agent  …        Gokul     Khatri
264                     5    Contact Agent  …       Samuel      Hurst
811                     5    Contact Agent  …        Kathy     Mendez
828                     4    Contact Agent  …       Joshua   Fletcher
911                     5    Contact Agent  …        Andre     Daniel
921                     5    Contact Agent  …  Christopher   Thompson
1160                    2    Contact Owner  …       Teresa     Taylor
1308                    6    Contact Agent  …       Eugene       Cook

     gender     phone_number                       email current_level  \
22        f        7493.2263     andreaellis@example.com           1.0
78        f     (08)24473521  douglasmarquez@example.org          19.0
139       m     02.7642.8725        zroberts@example.com           7.0
264       m  +61.2.2186.0016       carlpatel@example.net          12.0
811       f        9377 5298         qgeorge@example.net           4.0
828       m         68397365          adamle@example.org          50.0
911       u  +61.495.517.273      pricejames@example.net          34.0
921       u  +61.419.781.592       webbbrian@example.net          18.0
1160      f       0876215458         wayne30@example.org           1.0
1308      m   (03).6616.8618        ejohnson@example.com          -1.0

     total_level advertised_year advertised_month advertised_day
22           2.0            2022                4            29
78          85.0            2022                6             8
139         20.0            2022                5            27
264         18.0            2022                5            31
811          6.0            2022                7             9
828         75.0            2022                7             9
911         46.0            2022                7             9
921         20.0            2022                7             7
1160         1.0            2022                7             6
1308         2.0            2022                7             6
```

```
[10 rows x 25 columns]
```

[ ]:

[154]:
```
# @title Testing Set Insights

wgt_eda_testing_set_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Testing Set Insights:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_eda_testing_set_insights
```

[154]:
```
Textarea(value='', description='Testing Set Insights:',
layout=Layout(height='100%', width='auto'), placeholde…
```

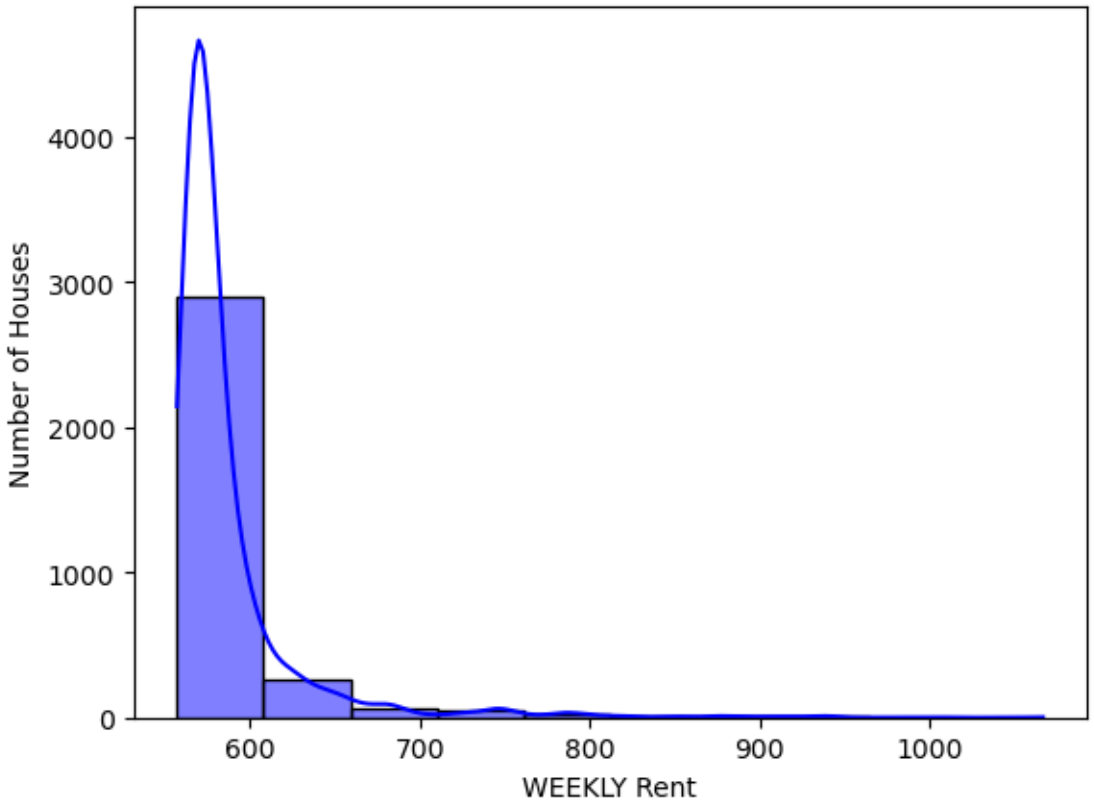
17.0.1 C.5 Explore Target Variable

Save the name of column used as the target variable and call it `target_name`

You can add more cells in this section

[155]:
```
# <Student to fill this section>

target_name = 'rent'
```

[156]:
```
sns.histplot(training_cleaned['rent'], bins =10,kde = True, color = 'blue')
plt.xlabel('WEEKLY Rent')
plt.ylabel('Number of Houses')
plt.show()
```



[157]:
```
# @title Target Variable Insights

wgt_eda_target_variable_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Target Variable Insights:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_eda_target_variable_insights
```

[157]:
```
Textarea(value='', description='Target Variable Insights:',
layout=Layout(height='100%', width='auto'), placeh…
```

17.0.2 C.6 Explore Feature of Interest

You can add more cells in this section

[158]:
```
# <Student to fill this section>
```

[159]:
```
training_cleaned.dtypes
```

[159]:
```
advertised_date        datetime64[ns]
number_of_bedrooms              int64
rent                          float64
floor_area                      int64
level                          object
suburb                         object
furnished                      object
tenancy_preference             object
number_of_bathrooms             int64
point_of_contact               object
secondary_address              object
building_number                 int64
street_name                    object
street_suffix                  object
prefix                         object
first_name                     object
last_name                      object
gender                         object
phone_number                   object
email                          object
yearmonth                    period[M]
advertised_year                 int32
advertised_month                int32
advertised_day                  int32
current_level                 float64
total_level                    object
dtype: object
```

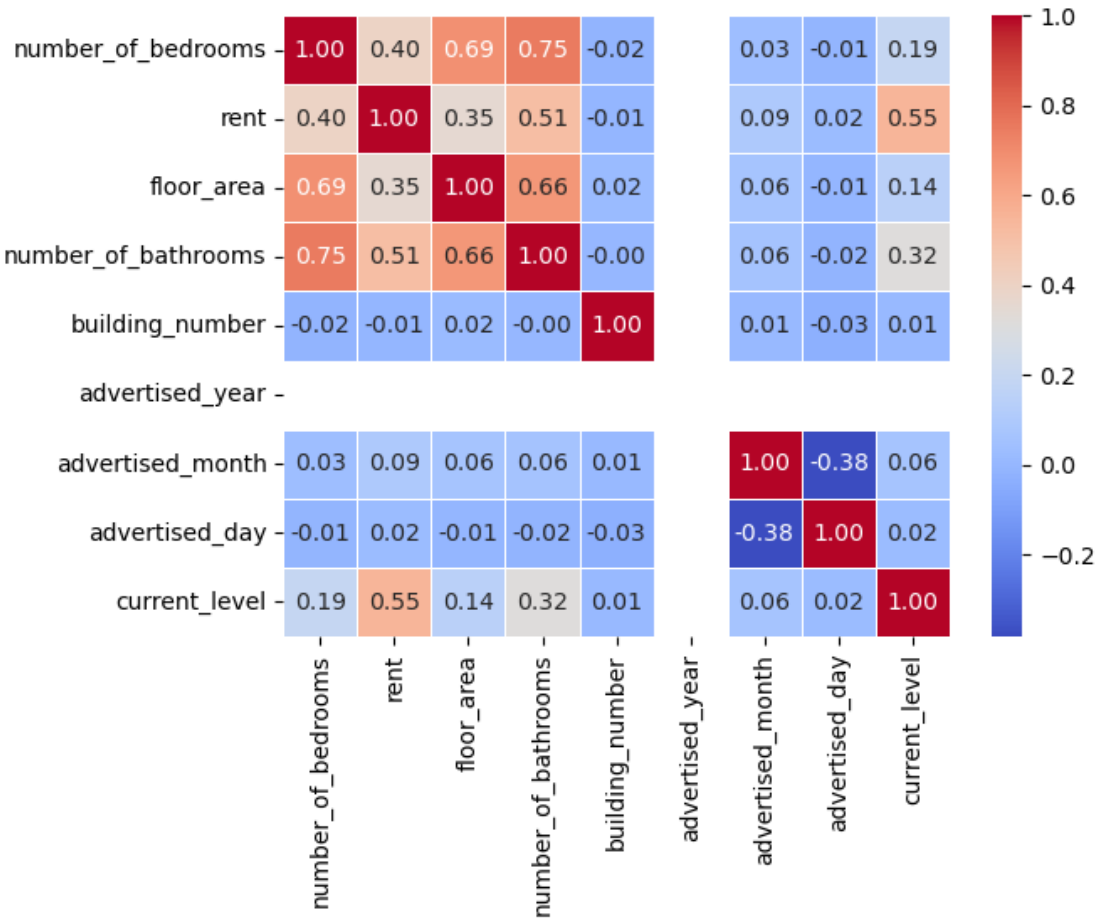
[160]:
```
training_cleaned.describe().T
```

[160]:
```
                       count                           mean  \
advertised_date         3316  2022-05-28 11:26:33.727382528
number_of_bedrooms   3316.0                      1.965018
rent                 3316.0                    586.813314
floor_area           3316.0                    848.947226
number_of_bathrooms  3316.0                      1.809107
building_number      3316.0                    189.539204
advertised_year      3316.0                        2022.0
advertised_month     3316.0                      5.372437
advertised_day       3316.0                     16.863088
current_level        3316.0                      2.844542

                                     min                  25%  \
advertised_date      2022-04-13 00:00:00  2022-05-14 00:00:00
number_of_bedrooms                   1.0                  1.0
```

```
rent                                557.0                567.0
floor_area                           20.0                510.0
number_of_bathrooms                   1.0                  1.0
building_number                       0.0                  7.0
advertised_year                    2022.0               2022.0
advertised_month                      4.0                  5.0
advertised_day                        1.0                 10.0
current_level                        -1.0                  1.0

                                      50%                  75%  \
advertised_date      2022-05-27 00:00:00  2022-06-13 06:00:00
number_of_bedrooms                    2.0                  2.0
rent                                574.0                587.0
floor_area                          800.0                1100.0
number_of_bathrooms                   2.0                  2.0
building_number                      46.0                269.0
advertised_year                    2022.0               2022.0
advertised_month                      5.0                  6.0
advertised_day                       18.0                 23.0
current_level                         2.0                  3.0

                                      max         std
advertised_date      2022-06-26 00:00:00         NaN
number_of_bedrooms                    6.0    0.751458
rent                                1067.0   43.304978
floor_area                          2100.0  433.457728
number_of_bathrooms                    6.0    0.739362
building_number                      998.0  284.592786
advertised_year                    2022.0         0.0
advertised_month                      6.0    0.608398
advertised_day                       31.0    8.364109
current_level                        76.0    4.909747
```

[161]:
```
#exploring numerical features of interest
```

[162]:
```
corr_matrix = training_cleaned.select_dtypes(include = ['number'])
matrix = corr_matrix.corr()
sns.heatmap(matrix, annot=True, cmap='coolwarm',fmt=".2f", linewidths=0.5 )
plt.show()
```



18 insights from the heat map

[163]:
```
#Number of Bedrooms is highly correlated with Floor Area (0.69) and Number of␣
 ↪Bathrooms (0.75). This makes sense since larger homes tend to have more␣
 ↪rooms.
#Rent shows a moderate correlation with Number of Bedrooms (0.40) and a␣
 ↪stronger one with Current Level (0.55), suggesting higher floors might␣
 ↪command higher rents.
#Floor Area and Number of Bathrooms also have a strong correlation (0.66),␣
 ↪indicating that larger homes often have more bathrooms.
```

[164]:
```
# @title Feature Insights

wgt_eda_feature_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Insights:',
```

```
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_eda_feature_insights
```

[164]:
```
Textarea(value='', description='Feature Insights:', layout=Layout(height='100%',
width='auto'), placeholder='<…
```

18.1 D. Feature Selection

18.1.1 D.1 Approach 1

[165]:
```
# <Student to fill this section>
```

[166]:
```
# @title Feature Selection 1 Insights

wgt_feat_selection_1_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Selection 1:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feat_selection_1_insights
```

[166]:
```
Textarea(value='', description='Feature Selection 1:',
layout=Layout(height='100%', width='auto'), placeholder…
```

18.1.2 D.2 Approach 2

[167]:
```
# <Student to fill this section>
```

[168]:
```
# @title Feature Selection 2 Insights

wgt_feat_selection_2_insights = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Selection 2:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feat_selection_2_insights
```

[168]:
```
Textarea(value='', description='Feature Selection 2:',
layout=Layout(height='100%', width='auto'), placeholder…
```

18.2 D.3 Final Selection of Features

Save the names of selected features into a list called `features_list`

[169]:
```
training_cleaned.columns
```

[169]:
```
Index(['advertised_date', 'number_of_bedrooms', 'rent', 'floor_area', 'level',
       'suburb', 'furnished', 'tenancy_preference', 'number_of_bathrooms',
       'point_of_contact', 'secondary_address', 'building_number',
       'street_name', 'street_suffix', 'prefix', 'first_name', 'last_name',
       'gender', 'phone_number', 'email', 'yearmonth', 'advertised_year',
       'advertised_month', 'advertised_day', 'current_level', 'total_level'],
      dtype='object')
```

[170]:
```
# <Student to fill this section>

features_list = ['number_of_bedrooms', 'rent', 'floor_area', 'current_level',␣
 ↪'total_level','suburb', 'furnished'
                ,'tenancy_preference','number_of_bathrooms','advertised_month']
```

[171]:
```
# @title Feature Selection Explanation

wgt_feat_selection_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Selection Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feat_selection_explanation
```

[171]:
```
Textarea(value='', description='Feature Selection Explanation:',
layout=Layout(height='100%', width='auto'), p…
```

18.3 E. Data Cleaning

18.3.1 E.1 Copy Datasets

Create copies of the datasets and called them `training_df_clean`, `validation_df_clean` and `testing_df_clean`

Do not change this code

19 I have changed training_df to training_cleaned because I changed it above during analysis, nothing changes.

[172]:
```
# Create copy of datasets

training_df_clean = training_cleaned[features_list].copy()
validation_df_clean = validation_df[features_list].copy()
testing_df_clean = testing_df[features_list].copy()
```

19.0.1 E.2 Fixing "<CHANGING DTYPES.>"

Provide some explanations on why you believe it is important to fix this issue and its impacts

1) converted total_levels to float. 2)REASON: CURRENT_LEVEL HAS UPPER AND LOWER BASEMENT WHICH WERE CONVERTED TO -0.5 AND -1 BASED ON THE BUSINESS LOGIC BECAUSE GROUND FLOOR WAS CONVERTED TO 0.

You can add more cells in this section

[173]:
```
# <Student to fill this section>
```

[174]:
```
training_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 3433
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   3316 non-null   int64
 1   rent                 3316 non-null   float64
 2   floor_area           3316 non-null   int64
 3   current_level        3316 non-null   float64
 4   total_level          3316 non-null   object
 5   suburb               3316 non-null   object
 6   furnished            3316 non-null   object
 7   tenancy_preference   3316 non-null   object
 8   number_of_bathrooms  3316 non-null   int64
 9   advertised_month     3316 non-null   int32
dtypes: float64(2), int32(1), int64(3), object(4)
memory usage: 401.1+ KB
```

[175]:
```
#converting total_levels to float because current level is in float.
```

[176]:
```
training_df_clean['total_level'] = training_df_clean['total_level'].
↪astype(float)
```

[177]:
```
#VALIDATION
validation_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1271 entries, 0 to 1319
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   1271 non-null   int64
 1   rent                 1271 non-null   float64
 2   floor_area           1271 non-null   int64
 3   current_level        1271 non-null   float64
 4   total_level          1271 non-null   float64
 5   suburb               1271 non-null   object
 6   furnished            1271 non-null   object
 7   tenancy_preference   1271 non-null   object
 8   number_of_bathrooms  1271 non-null   int64
 9   advertised_month     1271 non-null   int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 104.3+ KB
```

[178]:
```
#TESTING
testing_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1364 entries, 0 to 1363
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   1364 non-null   int64
 1   rent                 1364 non-null   float64
 2   floor_area           1364 non-null   int64
 3   current_level        1364 non-null   float64
 4   total_level          1364 non-null   float64
 5   suburb               1364 non-null   object
 6   furnished            1364 non-null   object
 7   tenancy_preference   1364 non-null   object
 8   number_of_bathrooms  1364 non-null   int64
 9   advertised_month     1364 non-null   int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 101.4+ KB
```

[179]:
```
# @title Data Cleaning 1 Explanation

wgt_data_cleaning_1_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
```

```
    description='Data Cleaning 1 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_data_cleaning_1_explanation
```

[179]:
```
Textarea(value='', description='Data Cleaning 1 Explanation:',
layout=Layout(height='100%', width='auto'), pla…
```

19.0.2 E.3 Fixing "<describe_issue_here>"

Provide some explanations on why you believe it is important to fix this issue and its impacts

You can add more cells in this section

[180]:
```
# <Student to fill this section>
```

[181]:
```
# @title Data Cleaning 2 Explanation

wgt_data_cleaning_2_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Data Cleaning 1 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_data_cleaning_2_explanation
```

[181]:
```
Textarea(value='', description='Data Cleaning 1 Explanation:',
layout=Layout(height='100%', width='auto'), pla…
```

19.0.3 E.4 Fixing "<describe_issue_here>"

Provide some explanations on why you believe it is important to fix this issue and its impacts

You can add more cells in this section

[182]:
```
# <Student to fill this section>
```

[183]:
```
# @title Data Cleaning 3 Explanation

wgt_data_cleaning_3_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Data Cleaning 3 Explanation:',
```

```
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_data_cleaning_3_explanation
```

[183]: Textarea(value='', description='Data Cleaning 3 Explanation:', layout=Layout(height='100%', width='auto'), pla…

19.1 F. Feature Engineering

19.1.1 F.1 Copy Datasets

Create copies of the datasets and called them `training_df_eng`, `validation_df_eng` and `testing_df_eng`

Do not change this code

[184]:
```
# Create copy of datasets

training_df_eng = training_df_clean.copy()
validation_df_eng = validation_df_clean.copy()
testing_df_eng = testing_df_clean.copy()
```

[185]:
```
training_df_eng.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 3433
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   3316 non-null   int64
 1   rent                 3316 non-null   float64
 2   floor_area           3316 non-null   int64
 3   current_level        3316 non-null   float64
 4   total_level          3316 non-null   float64
 5   suburb               3316 non-null   object
 6   furnished            3316 non-null   object
 7   tenancy_preference   3316 non-null   object
 8   number_of_bathrooms  3316 non-null   int64
 9   advertised_month     3316 non-null   int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 401.1+ KB
```

[186]:
```
validation_df_eng.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 1271 entries, 0 to 1319
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   1271 non-null   int64
 1   rent                 1271 non-null   float64
 2   floor_area           1271 non-null   int64
 3   current_level        1271 non-null   float64
 4   total_level          1271 non-null   float64
 5   suburb               1271 non-null   object
 6   furnished            1271 non-null   object
 7   tenancy_preference   1271 non-null   object
 8   number_of_bathrooms  1271 non-null   int64
 9   advertised_month     1271 non-null   int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 104.3+ KB
```

[187]:
```
testing_df_eng.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1364 entries, 0 to 1363
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   1364 non-null   int64
 1   rent                 1364 non-null   float64
 2   floor_area           1364 non-null   int64
 3   current_level        1364 non-null   float64
 4   total_level          1364 non-null   float64
 5   suburb               1364 non-null   object
 6   furnished            1364 non-null   object
 7   tenancy_preference   1364 non-null   object
 8   number_of_bathrooms  1364 non-null   int64
 9   advertised_month     1364 non-null   int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 101.4+ KB
```

[188]:
```
training_df_eng['advertised_month'].value_counts()
```

[188]:
```
advertised_month
5    1629
6    1461
4     226
Name: count, dtype: int64
```

[189]:
```
validation_df_eng['advertised_month'].value_counts()
```

[189]:
```
advertised_month
6    601
5    343
7    288
4     39
Name: count, dtype: int64
```

[190]:
```
testing_df_eng['advertised_month'].value_counts()
```

[190]:
```
advertised_month
7    678
5    352
6    294
4     40
Name: count, dtype: int64
```

[191]:
```
month_07_val = validation_df_eng[validation_df_eng['advertised_month'] == 7]
```

[192]:
```
month_07_test = testing_df_eng[testing_df_eng['advertised_month'] == 7]
```

20 Note: The training set does not contain data for the month of July (07).

21 We will analyze the validation and testing sets to determine how many values correspond to the month of July (07).

22 Due to data limitation in the training set we will have to drop the the months 07 from validation and testing

[193]:
```
testing_df_eng = testing_df_eng[testing_df_eng['advertised_month'] != 7]
validation_df_eng = validation_df_eng[validation_df_eng['advertised_month'] !=␣
↪7]
```

[194]:
```
testing_df_eng.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 686 entries, 0 to 685
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   686 non-null    int64
 1   rent                 686 non-null    float64
 2   floor_area           686 non-null    int64
 3   current_level        686 non-null    float64
 4   total_level          686 non-null    float64
```

```
 5   suburb               686 non-null    object
 6   furnished            686 non-null    object
 7   tenancy_preference   686 non-null    object
 8   number_of_bathrooms  686 non-null    int64
 9   advertised_month     686 non-null    int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 56.3+ KB
```

[195]:
```
validation_df_eng.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 983 entries, 0 to 1319
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   983 non-null    int64
 1   rent                 983 non-null    float64
 2   floor_area           983 non-null    int64
 3   current_level        983 non-null    float64
 4   total_level          983 non-null    float64
 5   suburb               983 non-null    object
 6   furnished            983 non-null    object
 7   tenancy_preference   983 non-null    object
 8   number_of_bathrooms  983 non-null    int64
 9   advertised_month     983 non-null    int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 80.6+ KB
```

22.0.1 F.2 New Feature "<CURRENT_LEVEL>"

Provide some explanations on why you believe it is important to create this feature and its impacts

1) Separated the column 'level' which was in a format eg: x out of y format. where x was converted into current_level

[196]:
```
# <Student to fill this section>
```

[197]:
```
training_df_eng['current_level']
```

[197]:
```
0       0.0
1       1.0
2       1.0
3       1.0
4       0.0
       ...
3429    4.0
3430    2.0
3431    3.0
```

```
3432    1.0
3433    4.0
Name: current_level, Length: 3316, dtype: float64
```

[198]:
```
# @title Feature Engineering 1 Explanation

wgt_feature_engineering_1_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Engineering 1 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feature_engineering_1_explanation
```

[198]:
```
Textarea(value='', description='Feature Engineering 1 Explanation:',
layout=Layout(height='100%', width='auto'…
```

22.0.2 F.3 New Feature "<TOTAL_LEVEL , ADVERTISED_MONTH>"

Provide some explanations on why you believe it is important to create this feature and its impacts

1) these were the total levels of the house. For eg: x out of y, here y- was converted into total_level
2) advertised month was extracted from the original feature advertised time. Since the data holds information on 2022 year and only accounts for 3 months. Hence, became important for removing redundant features like year .

[199]:
```
# <Student to fill this section>
```

[200]:
```
# @title Feature Engineering 2 Explanation

wgt_feature_engineering_2_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Engineering 2 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feature_engineering_2_explanation
```

[200]:
```
Textarea(value='', description='Feature Engineering 2 Explanation:',
layout=Layout(height='100%', width='auto'…
```

22.0.3 F.4 New Feature "<average_rent_bath&bed>"

Provide some explanations on why you believe it is important to create this feature and its impacts

[201]:
```
#since bed bath have a affect on rental prices, my thinking behind adding this␣
 ↪feature was to give model a chance to
#understand the pattern better.
```

[202]:
```
# <Student to fill this section>
```

[203]:
```
training_df_eng.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 3433
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   number_of_bedrooms   3316 non-null   int64
 1   rent                 3316 non-null   float64
 2   floor_area           3316 non-null   int64
 3   current_level        3316 non-null   float64
 4   total_level          3316 non-null   float64
 5   suburb               3316 non-null   object
 6   furnished            3316 non-null   object
 7   tenancy_preference   3316 non-null   object
 8   number_of_bathrooms  3316 non-null   int64
 9   advertised_month     3316 non-null   int32
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 401.1+ KB
```

23 creating new variables for experiment 2

24 average rent per bedroom &bathroom.

[204]:
```
training_df_eng['average_rent_bath&bed'] = training_df_eng.
 ↪groupby(['number_of_bedrooms' , 'number_of_bathrooms'])['rent'].
 ↪transform('mean').round(2)
```

[205]:
```
training_df_eng
```

[205]:

	number_of_bedrooms	rent	floor_area	current_level	total_level	\
0	2	568.0	1100	0.0	2.0	
1	2	581.0	800	1.0	3.0	
2	2	577.0	1000	1.0	3.0	
3	2	565.0	850	1.0	2.0	
4	2	564.0	600	0.0	1.0	

```
…                  …    …          …              …           …
3429                   3  600.0        1250            4.0           5.0
3430                   2  571.0        1350            2.0           2.0
3431                   2  574.0        1000            3.0           5.0
3432                   3  592.0        2000            1.0           4.0
3433                   2  574.0        1000            4.0           5.0

        suburb       furnished tenancy_preference  number_of_bathrooms  \
0     Canberra     Unfurnished   Bachelors/Family                    2
1     Canberra  Semi-Furnished   Bachelors/Family                    1
2     Canberra  Semi-Furnished   Bachelors/Family                    1
3     Canberra     Unfurnished          Bachelors                    1
4     Canberra     Unfurnished   Bachelors/Family                    2
…        …              …                …                     …
3429     Perth       Furnished          Bachelors                    2
3430     Perth     Unfurnished   Bachelors/Family                    2
3431     Perth  Semi-Furnished   Bachelors/Family                    2
3432     Perth  Semi-Furnished   Bachelors/Family                    3
3433     Perth     Unfurnished          Bachelors                    2

      advertised_month  average_rent_bath&bed
0                    5                 583.42
1                    5                 569.09
2                    5                 569.09
3                    5                 569.09
4                    4                 583.42
…                  …                   …
3429                 6                 592.44
3430                 6                 583.42
3431                 5                 583.42
3432                 5                 621.25
3433                 5                 583.42

[3316 rows x 11 columns]
```

[206]:
```
#validation df
```

[207]:
```
validation_df_eng['average_rent_bath&bed'] = validation_df_eng.
↪groupby(['number_of_bedrooms' , 'number_of_bathrooms'])['rent'].
↪transform('mean').round(2)
```

[208]:
```
#testing df
```

[209]:
```
testing_df_eng['average_rent_bath&bed'] = testing_df_eng.
↪groupby(['number_of_bedrooms' , 'number_of_bathrooms'])['rent'].
↪transform('mean').round(2)
```

[210]:
```
# @title Feature Engineering 3 Explanation

wgt_feature_engineering_3_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Feature Engineering 3 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feature_engineering_3_explanation
```

[210]:
```
Textarea(value='', description='Feature Engineering 3 Explanation:',
layout=Layout(height='100%', width='auto'…
```

24.1 G. Data Preparation for Modeling

24.1.1 G.1 Copy Datasets

Create copies of the datasets and split them into X and y

Do not change this code

[211]:
```
# Create copy of datasets

X_train = training_df_eng.copy()
X_val = validation_df_eng.copy()
X_test = testing_df_eng.copy()

y_train = X_train.pop(target_name)
y_val = X_val.pop(target_name)
y_test = X_test.pop(target_name)
```

24.1.2 G.2 Data Transformation

Provide some explanations on why you believe it is important to perform this data transformation and its impacts

[212]:
```
# <Student to fill this section>
```

[213]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 3433
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
```

```
 0   number_of_bedrooms     3316 non-null   int64
 1   floor_area             3316 non-null   int64
 2   current_level          3316 non-null   float64
 3   total_level            3316 non-null   float64
 4   suburb                 3316 non-null   object
 5   furnished              3316 non-null   object
 6   tenancy_preference     3316 non-null   object
 7   number_of_bathrooms    3316 non-null   int64
 8   advertised_month       3316 non-null   int32
 9   average_rent_bath&bed  3316 non-null   float64
dtypes: float64(3), int32(1), int64(3), object(3)
memory usage: 401.1+ KB
```

[214]:
```
# using one-hot-encoding approach we convert furnished & tenancy_preference &␣
↪suburbs
```

[215]:
```
#train
X_train = pd.get_dummies(X_train, columns = ['suburb', 'furnished',␣
↪'tenancy_preference'], dtype = int)
```

[216]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 3433
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   number_of_bedrooms                    3316 non-null   int64
 1   floor_area                            3316 non-null   int64
 2   current_level                         3316 non-null   float64
 3   total_level                           3316 non-null   float64
 4   number_of_bathrooms                   3316 non-null   int64
 5   advertised_month                      3316 non-null   int32
 6   average_rent_bath&bed                 3316 non-null   float64
 7   suburb_Adelaide                       3316 non-null   int64
 8   suburb_Brisbane                       3316 non-null   int64
 9   suburb_Canberra                       3316 non-null   int64
 10  suburb_Melbourne                      3316 non-null   int64
 11  suburb_Perth                          3316 non-null   int64
 12  suburb_Sydney                         3316 non-null   int64
 13  furnished_Furnished                   3316 non-null   int64
 14  furnished_Semi-Furnished              3316 non-null   int64
 15  furnished_Unfurnished                 3316 non-null   int64
 16  tenancy_preference_Bachelors          3316 non-null   int64
 17  tenancy_preference_Bachelors/Family   3316 non-null   int64
 18  tenancy_preference_Family             3316 non-null   int64
dtypes: float64(3), int32(1), int64(15)
memory usage: 634.2 KB
```

[217]:
```
#val
X_val = pd.get_dummies(X_val, columns = ['suburb', 'furnished',␣
 ↪'tenancy_preference'],  dtype = int)
```

[218]:
```
X_val.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 983 entries, 0 to 1319
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   number_of_bedrooms                    983 non-null    int64
 1   floor_area                            983 non-null    int64
 2   current_level                         983 non-null    float64
 3   total_level                           983 non-null    float64
 4   number_of_bathrooms                   983 non-null    int64
 5   advertised_month                      983 non-null    int32
 6   average_rent_bath&bed                 983 non-null    float64
 7   suburb_Adelaide                       983 non-null    int64
 8   suburb_Brisbane                       983 non-null    int64
 9   suburb_Canberra                       983 non-null    int64
 10  suburb_Melbourne                      983 non-null    int64
 11  suburb_Perth                          983 non-null    int64
 12  suburb_Sydney                         983 non-null    int64
 13  furnished_Furnished                   983 non-null    int64
 14  furnished_Semi-Furnished              983 non-null    int64
 15  furnished_Unfurnished                 983 non-null    int64
 16  tenancy_preference_Bachelors          983 non-null    int64
 17  tenancy_preference_Bachelors/Family   983 non-null    int64
 18  tenancy_preference_Family             983 non-null    int64
dtypes: float64(3), int32(1), int64(15)
memory usage: 149.8 KB
```

[219]:
```
#test
X_test =pd.get_dummies(X_test, columns = ['suburb', 'furnished',␣
 ↪'tenancy_preference'],  dtype = int)
```

[220]:
```
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 686 entries, 0 to 685
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   number_of_bedrooms                    686 non-null    int64
 1   floor_area                            686 non-null    int64
 2   current_level                         686 non-null    float64
 3   total_level                           686 non-null    float64
```

```
 4   number_of_bathrooms                  686 non-null    int64
 5   advertised_month                     686 non-null    int32
 6   average_rent_bath&bed                686 non-null    float64
 7   suburb_Adelaide                      686 non-null    int64
 8   suburb_Brisbane                      686 non-null    int64
 9   suburb_Canberra                      686 non-null    int64
 10  suburb_Melbourne                     686 non-null    int64
 11  suburb_Perth                         686 non-null    int64
 12  suburb_Sydney                        686 non-null    int64
 13  furnished_Furnished                  686 non-null    int64
 14  furnished_Semi-Furnished             686 non-null    int64
 15  furnished_Unfurnished                686 non-null    int64
 16  tenancy_preference_Bachelors         686 non-null    int64
 17  tenancy_preference_Bachelors/Family  686 non-null    int64
 18  tenancy_preference_Family            686 non-null    int64
dtypes: float64(3), int32(1), int64(15)
memory usage: 104.5 KB
```

[221]:
```
# @title Data Preparation 1 Explanation

wgt_data_preparation_1_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Data Preparation 1 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_data_preparation_1_explanation
```

[221]:
```
Textarea(value='', description='Data Preparation 1 Explanation:',
layout=Layout(height='100%', width='auto'), …
```

24.1.3 G.3 Data Transformation

Provide some explanations on why you believe it is important to perform this data transformation and its impacts

[222]:
```
# @title Data Preparation 2 Explanation

wgt_data_preparation_2_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Data Preparation 2 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
```

```
wgt_data_preparation_2_explanation
```

[222]:
```
Textarea(value='', description='Data Preparation 2 Explanation:',
layout=Layout(height='100%', width='auto'), …
```

24.1.4 G.4 Data Transformation

Provide some explanations on why you believe it is important to perform this data transformation and its impacts

[223]:
```
# @title Data Preparation 3 Explanation

wgt_data_preparation_3_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Data Preparation 3 Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_data_preparation_3_explanation
```

[223]:
```
Textarea(value='', description='Data Preparation 3 Explanation:',
layout=Layout(height='100%', width='auto'), …
```

24.2 H. Save Datasets

Do not change this code

[224]:
```
pwd
```

[224]:
```
'/Users/ratikpant/Desktop'
```

[225]:
```
# Save training set

X_train.to_csv('/Users/ratikpant/Desktop/machine learning/ X_train.csv',␣
 ↪index=False)
y_train.to_csv('/Users/ratikpant/Desktop/machine learning/ y_train.csv',␣
 ↪index=False)
```

[226]:
```
# Save validation set

X_val.to_csv('/Users/ratikpant/Desktop/machine learning/ X_val.csv',␣
 ↪index=False)
y_val.to_csv('/Users/ratikpant/Desktop/machine learning/ y_val.csv',␣
 ↪index=False)
```

[227]:
```
# Save testing set

X_test.to_csv('/Users/ratikpant/Desktop/machine learning/X_test.csv',␣
 ↪index=False)
y_test.to_csv('/Users/ratikpant/Desktop/machine learning/y_test.csv',␣
 ↪index=False)
```

25 saving future month 07 validation

[242]:
```
month_07_val
```

[242]:

	number_of_bedrooms	rent	floor_area	current_level	total_level	\
686	2	568.0	800	1.0	2.0	
687	2	565.0	650	1.0	2.0	
690	2	571.0	650	0.0	1.0	
691	2	562.0	800	0.0	1.0	
692	2	564.0	650	0.0	3.0	
…	…	…	…	…	…	
1308	2	565.0	900	0.0	2.0	
1311	2	565.0	800	1.0	6.0	
1312	1	564.0	650	3.0	3.0	
1313	2	568.0	1125	2.0	3.0	
1314	2	581.0	1350	8.0	14.0	

	suburb	furnished	tenancy_preference	number_of_bathrooms	\
686	Canberra	Unfurnished	Bachelors/Family	1	
687	Canberra	Unfurnished	Family	1	
690	Canberra	Unfurnished	Family	2	
691	Canberra	Unfurnished	Bachelors/Family	1	
692	Canberra	Semi-Furnished	Bachelors/Family	2	
…	…	…	…	…	
1308	Perth	Semi-Furnished	Bachelors	2	
1311	Perth	Furnished	Bachelors/Family	2	
1312	Perth	Semi-Furnished	Bachelors/Family	1	
1313	Perth	Unfurnished	Bachelors	2	
1314	Perth	Semi-Furnished	Bachelors	2	

	advertised_month
686	7
687	7
690	7
691	7
692	7
…	…
1308	7
1311	7

```
1312                 7
1313                 7
1314                 7

[288 rows x 10 columns]
```

[243]:
```
month_07_val.to_csv('/Users/ratikpant/Desktop/machine learning/ month_07_val',␣
↪index=False)
```

26 saving future month 07 test

[245]:
```
month_07_test
```

[245]:
```
      number_of_bedrooms   rent  floor_area  current_level  total_level  \
686                    2  566.0         720            4.0          4.0
687                    2  587.0        1100            2.0          2.0
688                    3  571.0         800            0.0          1.0
689                    2  564.0         600            0.0          2.0
690                    3  583.0        1150            1.0          2.0
…                    …    …          …              …            …
1359                   3  574.0        1500           -1.0          2.0
1360                   2  577.0         855            4.0          5.0
1361                   2  587.0        1040            2.0          4.0
1362                   3  600.0        1750            3.0          5.0
1363                   3  613.0        1500           23.0         34.0

        suburb       furnished tenancy_preference  number_of_bathrooms  \
686   Canberra  Semi-Furnished   Bachelors/Family                    2
687   Canberra       Furnished          Bachelors                    2
688   Canberra     Unfurnished   Bachelors/Family                    2
689   Canberra     Unfurnished          Bachelors                    1
690   Canberra     Unfurnished   Bachelors/Family                    2
…        …              …                 …                    …
1359     Perth  Semi-Furnished   Bachelors/Family                    3
1360     Perth     Unfurnished          Bachelors                    2
1361     Perth     Unfurnished          Bachelors                    2
1362     Perth  Semi-Furnished   Bachelors/Family                    3
1363     Perth  Semi-Furnished            Family                    2

      advertised_month
686                  7
687                  7
688                  7
689                  7
690                  7
…                  …
```

```
1359                7
1360                7
1361                7
1362                7
1363                7

[678 rows x 10 columns]
```

[246]:
```
month_07_test.to_csv('/Users/ratikpant/Desktop/machine learning/␣
↪month_07_test', index=False)
```

26.1 I. Assess Baseline Model

26.1.1 I.1 Generate Predictions with Baseline Model

[228]:
```
# <Student to fill this section>
```

26.1.2 I.2 Selection of Performance Metrics

Provide some explanations on why you believe the performance metrics you chose is appropriate

[229]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error as mse
```

[230]:
```
base = LinearRegression()
```

[231]:
```
base = base.fit(X_train,y_train)
```

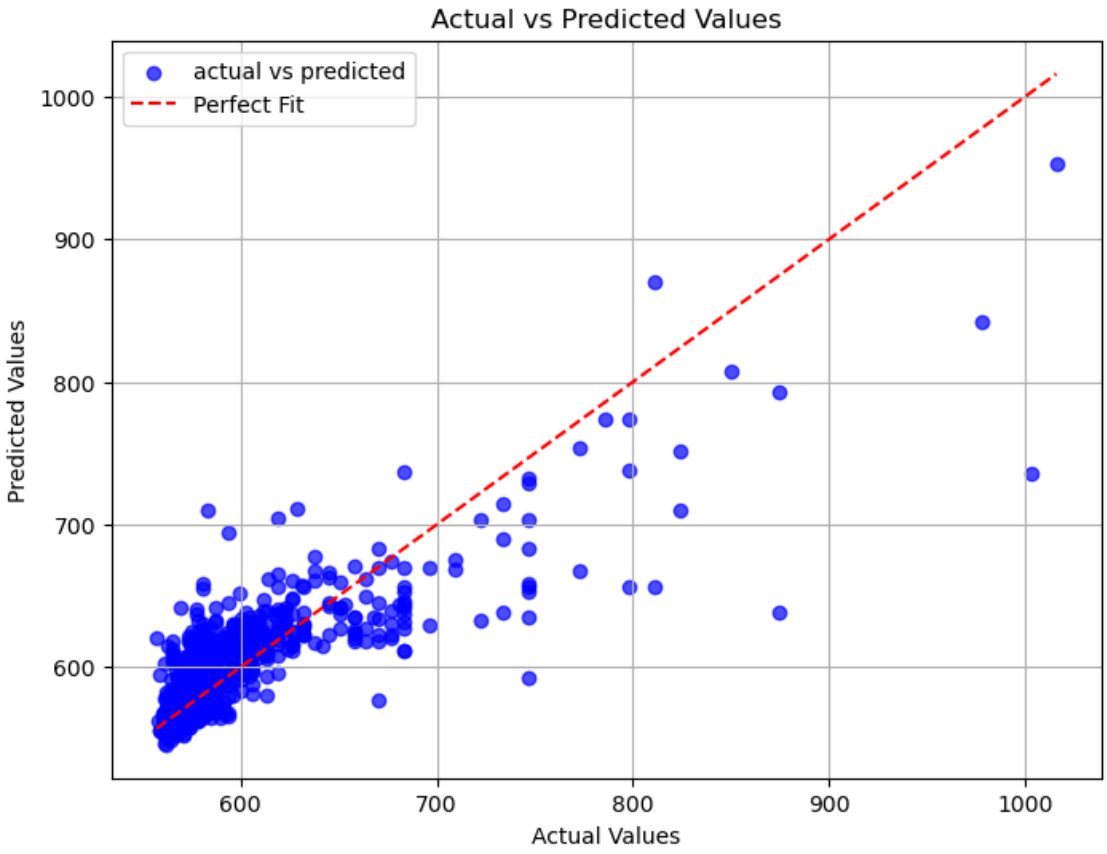
[232]:
```
y_val_pred = base.predict(X_val)
```

[233]:
```
mse_val = mse(y_val_pred, y_val)
rmse = np.sqrt(mse_val)
print("rmse score is:", rmse)
```

```
rmse score is: 26.140506655579017
```

[234]:
```
plt.figure(figsize=(8, 6))
plt.scatter(y_val, y_val_pred, alpha=0.7, color='blue', label='actual vs␣
↪predicted')
plt.plot([min(y_val), max(y_val)], [min(y_val), max(y_val)], color='red',␣
↪linestyle='--', label='Perfect Fit')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values')
plt.legend()
```

```
plt.grid(True)
plt.show()
```



[ ]:

[ ]:

[ ]:

[235]:
```
y_test_pred = base.predict(X_test)
```

[236]:
```
mse_test = mse(y_test_pred, y_test)
rmsee = np.sqrt(mse_test)
print("rmse score is:", rmsee)
```

```
rmse score is: 30.693638176631456
```

[237]:
```
# <Student to fill this section>
```

[238]:
```
# @title Performance Metrics Explanation

wgt_perf_metrics_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Performance Metrics Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_perf_metrics_explanation
```

[238]:
```
Textarea(value='', description='Performance Metrics Explanation:',
layout=Layout(height='100%', width='auto'),…
```

26.1.3 I.3 Baseline Model Performance

Provide some explanations on model performance

[239]:
```
# <Student to fill this section>
```

[240]:
```
# @title Performance Metrics Explanation

wgt_model_performance_explanation = widgets.Textarea(
    value=None,
    placeholder='<student to fill this section>',
    description='Model Performance Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_model_performance_explanation
```

[240]:
```
Textarea(value='', description='Model Performance Explanation:',
layout=Layout(height='100%', width='auto'), p…
```