Task 1

Code 1:
```
head -n 1 property_transaction_victoria.csv
```

Output:
```
ID      ,postcode ,suburb           ,sold_time  ,sold_type      ,sold_price     ,address
,beds ,baths ,parking ,area       ,property_type
,features
,description
```

Purpose:
To see the first header row in the property file and find the location of the sold_time column.

Code 2:
```
LC_ALL=C cut -d, -f 4 property_transaction_victoria.csv | grep -v sold_time | grep -v NA \
| grep -v  '^$'  | awk -F'[/: ]'  '{ printf "%04d-%02d-%02d %02d:%02d\n", $3,$2,$1,$4, $5}' | sort >
sorted_sold_times
```

Output:
```
2021-01-01 00:19
2021-01-01 00:20
2021-01-01 00:50
2021-01-01 01:14
2021-01-01 01:37
2021-01-01 01:52
2021-01-01 02:05
2021-01-01 02:31
2021-01-01 02:36
2021-01-01 02:37
2021-01-01 02:59
2021-01-01 03:06
2021-01-01 03:33
2021-01-01 03:53
2021-01-01 05:28
2021-01-01 05:28
2021-01-01 05:34
2021-01-01 05:46
2021-01-01 06:09
```
...click on the icon below to see the full output!

Purpose:
I selected the sold_time column for further processing by first using -d to specify the comma as the delimiter and -f to specify which column I wanted from the file.
Then I added a pipeline to remove the 'sold_time' header from the output. Next, I sorted the data from the earliest to the latest time and removed NA values as well as missing values. I used awk to rearrange the date and time values because otherwise, it will not order them correctly as strings. I saved the output to a file 'sorted_sold_times'
and added the .doc extension so you can view the full output by clicking on the icon link above. Note LC_ALL=C was added to all lines of code containing cut and paste so that the command worked properly for mac users.

Code 3:

```
echo "Earliest sold date record:"
LC_ALL=C cut -d, -f 4 property_transaction_victoria.csv | grep -v sold_time | grep -v NA \
| grep -v  '^$'  | awk -F'[/: ]'  ' { printf "%04d-%02d-%02d %02d:%02d\n", $3,$2,$1,$4, $5}' | sort | head -n 1
```

Output:

```
Earliest sold date record:
2021-01-01 00:19
```

Purpose:

I added to code 1 by using a pipeline. I used head -n 1 to extract the top 1 row from the sold_time column. This gave me an output of the first sold date and time record. Echo is a command I used to display the text: ‘Earliest sold date record’.

Code 4:

```
echo " Last sold date record:"
LC_ALL=C cut -d, -f 4 property_transaction_victoria.csv | grep -v sold_time | grep -v NA \
| grep -v  '^$'  | awk -F'[/: ]'  ' { printf "%04d-%02d-%02d %02d:%02d\n", $3,$2,$1,$4, $5}' | sort | tail -n 1
```

Output:

```
Last sold date record:
2021-12-31 23:58
```

Purpose:

Using the same method as code 3, I found the last sold time record by using tail -n 1 to output the last row. Now we can answer the question, what is the sold_time range of the records.
The range: 2021-01-01 00:19 to 2021-12-31 23:58

Task 2

Code 1:

```
LC_ALL=C cut -d, -f1 property_transaction_victoria.csv | grep -v ID | grep -v '^[0-9]\{6\}$' | wc -l
```

Output:

```
10
```

Purpose:

- I selected the ID column for further processing by first using -d to specify the comma as the delimiter and -f to specify the first 'ID' column I wanted from the file. Then I used grep -v to remove the lines that do not match the regex. The regex expression ^[0-9]\{6\}$ means any 6 digit number. I used wc -1 to count the number of these rows. This answers the question: count lines with an id that is not a number of 6 digits long, i.e., id values that contain anything other than numbers OR are of a length more/less than 6. The code output has counted 10 invalid ID values.

Code 2:

```
head -n 1 property_transaction_victoria.csv > header.csv

grep  '^[0-9]\{6\},' property_transaction_victoria.csv  > only_valid_id.csv

LC_ALL=C cut -d, -f1-3 only_valid_id.csv > before_sold_time.csv

LC_ALL=C cut -d, -f4 only_valid_id.csv  | grep -v NA | grep -v  '^$' | cut -d' ' -f1 > sold_time.csv

LC_ALL=C cut -d, -f5-14 only_valid_id.csv > after_sold_time.csv

LC_ALL=C paste -d, before_sold_time.csv sold_time.csv after_sold_time.csv > combined_noheader.csv

cat header.csv combined_noheader.csv > filtered_property.csv

head -n 5 filtered_property.csv
```

Output:

```
ID     ,postcode ,suburb          ,sold_time  ,sold_type      ,sold_price    ,address
,beds ,baths ,parking ,area      ,property_type
,features
,description
294290,3040,Essendon         ,27/03/2021,auction       ,1655000,1/53 Nimmo Street Essendon VIC 3040
,5,3,2,      ,Townhouse
,
,Property Description Family Flexibility With A Luxury Edge An immaculate home of distinction and quality with bright open spaces at every turn this extensive entertainers residence is a showpiece of contemporary elegance and premium family living. Designed with flexibility and generosity in mind it features open-plan living and dining an elaborate kitchen and butlers pantry each with Smeg appliances up to five bedrooms or four plus home office three sleek fully-tiled bathrooms a first floor retreat and the luxury of two undercover alfresco options with heating.    Read less
169586,3981,Koo Wee Rup       ,18/02/2021,private treaty   ,554000,8 William Street Koo Wee Rup VIC 3981                                  ,3,2,4,1231,House
,
,Property Description Brick Veneer Home - HUGE Development Block!!! This house has 3 bedrooms the spacious master bedroom has a large walk-in-robe plus a full ensuite. The other 2 large bedrooms have BIR's . There is a wide entrance that gives you the option of either turning left into a magic lounge that has access to the meals/kitchen or continuing on into a short passage to the other end of the kitchen or the massive laundry. The passage turns towards the bedrooms and the big bathroom. There is a large alcove
```

under roof line which adjoins the kitchen this can been enclosed to make an office or you could extend and update the kitcken/meals area at some time in the future. The large double garage under roof line has large windows on one side and a doorway that allows access under a recess to the front door of the house. Read less
237723,3006,Southbank ,29/04/2021,private treaty ,540000,2205/180 City Road Southbank VIC 3006 ,2,1,1, ,Apartment / Unit / Flat ,Property Features* Unverified featureInternal Laundry*Intercom*Heating*Dishwasher*Secure ParkingSwimming PoolView less
,Property Description Central Southbank Sanctuary with Breathtaking Panorama from a Corner Position A captivating combination of sunlit space and designer quality from a commanding corner position this impeccable 2 bedroom retreat showcases striking views stretching across the horizon. Set 22 floors high in the award-winning SouthbankONE complex venture downstairs and walk to Crown entertainment riverfront restaurants supermarket choice Queensbridge Street trams and Flinders Street trains. This is the life! Discover wide-reaching open-plan living and dining complemented by a stone-finished kitchen with stainless-steel appliances including a dishwasher and a waterfall-edged breakfast bar for relaxed meal times. Framed by floor-to-ceiling glass step outside to an undercover balcony boasting a spectacular panorama sweeping across the neighbourhood skyline and the blue waters of Port Phillip Bay. The sun-drenched pair of mirror-robed bedrooms are generous in size serviced by a luxe bathroom with slick floor-to-ceiling tiles and a stone-topped vanity. Read less
116018,3121,Richmond ,8/11/2021,auction ,1180000,210/84 Cutter Street Richmond VIC 3121 ,3,2,2, ,Apartment / Unit / Flat
,
,Property Description Every imaginable convenience This property is open for inspection. In accordance with Victorian Government requirements only fully vaccinated people will be able to attend the open for inspection and auction for this property. Enjoying a privileged north facing position within a landmark address distinguished finishes a serene material palette and outstanding proportions define this exquisite apartment residence. Designed by award winning MAA architects and occupying approximately 110sqm of living space two alfresco spaces two basement car parks and three storage cages deliver contemporary practicality. The impressive open plan living room showcases warm timber floors that contrast beautifully with cool luxurious marble elements within the stylish kitchen complete with a suite of Miele appliances. Floor to ceiling glass connects to an inviting terrace perfect for relaxing recharging and entertaining. The main bedroom features a deluxe ensuite and two additionally sensationally sized bedrooms come complete with built in robes complemented by a central bathroom. Includes Euro laundry heating and cooling in a compelling location for convenience with Swan Street Burnley train station Burnley Park freeway access and the Yarra River all close by. Read less

Purpose:

In this task I successfully removed the invalid ID rows, removed time values, stored the data into a new file named filtered_property.csv and displayed the first 5 rows.

To do this I first, I used head -n 1 to save the header row to a new file called header.csv. Then I filtered all rows that started with the correct 6 digit ID numbers using the same regex that was used in Task 1. I then saved these to another file named only_valid_id.csv. I then split the only_valid_id.csv file into parts. A before_sold_time.csv file which contained the columns 1 to 3. Then I spliced sold_time wherever there was a space, then I selected the first slice which had the date values. I saved this sold_time column with just date values to a separate file. Next I saved the remaining rows into another file: after_sold_time.csv. I then pasted and re-joined all the data to file named combined_noheader.csv. Then I used cat to add the header.csv file back to this one to create the final file named filtered_property.csv. I then used head -n 5 to show the first 5 rows of the file, including the header row.

Task 3

Code 1:

```
LC_ALL=C cut -d, -f14 filtered_property.csv | grep -i "Alfresco" | grep -i "Renovation" | grep -v description| wc -l
```

Output:

548

Purpose:

To answer the question: how many transaction records contain both “Alfresco” and “Renovation” in their description value in the dataset? (Note: Please ignore cases). I first selected the description column for further processing by first using -d to specify the comma as the delimiter and -f to specify which column I wanted from the file. Then I added a pipeline to filter the words Alfresco and Renovation using grep. ‘-i’ was used to ignore cases. I then used wc -l to count the number of lines that contained these words.

Code 2:

```
LC_ALL=C cut -d, -f14 filtered_property.csv | sed -E 's/m2|sqms|sq[[:space:]]+metre|sq[[:space:]]+metres|sq\.metres|sq\.metre|sq\.meters|sq\.[[:space:]]*m|sq\.meter|sq[[:space:]]+meter|sq[[:space:]]+meters|sq[[:space:]]+m|m\^2/sqm/Ig'| grep -i sqm|wc -l
```

Output:

37564

Purpose:

The task asked how many transaction records contain the property size information (e.g.: 1249m2, 758 sq metres) in their description value in the dataset? (Note: Please ignore cases). For this, I decided to do some small data engineering to replace all the variations of m^2 into ‘sqm’ in the description column. For this step, I used the sed command. To see the different variations, I had to go to the excel file and try Command-Find each variation to see if it existed. Once I had confirmation, I added each condition to the sed command so it would replace each variation with sqm. After this, I ensured the matching would ignore cases. The grep command was used to find all lines with sqm in them, and then I counted the lines using wc-l.

Task 4

Code 1:

```
LC_ALL=C cut -d',' -f1,4,5,6,7,8,11,12,14 filtered_property.csv \
| awk -F',' 'NR>1{split($2,d,"/"); m=d[2]+0;area=$7; gsub(/[[:space:]]/,"",area); \
if (area~/ha$/) { sub(/ha$/,"",area); area=area*10000 } else { area+=0 }; \
p=$8; sub(/^[[:space:]]+/,"",p); sub(/[[:space:]]+$/,"",p); \
if (m%2==1 && p=="Townhouse" && area>300) printf "%04d-%02d-%02d\n", d[3], d[2], d[1]}' | sort \
| awk 'NR==1{first=$0}{last=$0} END{print "first_sold_time,last_sold_time"; if(NR>0) print first "," last}'
```

Output:

```
first_sold_time,last_sold_time
2021-01-03,2021-11-30
```

Purpose:

For this task, I first filtered the dataset to only include the columns: ID, sold_time, sold_type, sold_price, address, beds, area, property_type, and description. Then I used an awk pipeline to split the data by commas and use NR>1 to skip the header. I then split the sold_time wherever there is a slash '/' so that I have day, month and year split up. I then assigned the m variable to the second slice, which was month. I made sure to remove leading and trailing whitespaces for area and property_type to ensure that all relevant rows will be matched. Then, for area, I used an if statement to convert ha values to m^2 values by multiplying by 10,000. A final if statement to pass all the conditions from the question, m%2==1 to ensure only odd months were included. Then if the rows matched, I printed these in the correct date format so they could be sorted. I did not worry about the case because all Townhouse values had this exact case when I used Command-Find. I then ordered the dates using sort and then printed the first and last dates which were 2021-01-03 and 2021-11-30; respectively.