q1a

November 7, 2023

1 Project 3 - Chris O'Brien 4638722

https://github.com/chrisobi02/Chris-OBrien-2504-2023-PROJECT3

I did no data recovery strategy (although I'm sure the EM algorithm or nearest neighbour would have been sufficient). I dropped any msising prices from task 1.2 onwards when concerned with price, but for single variables included all datapoints

[ ]:
```
using Pkg; Pkg.activate(".")
```

```
  Activating project at `~/Chris-OBrien-2504-2023-PROJECT3`
```

[ ]:
```
using DataFrames, CSV, DataFrameMacros, StatsBase, StatsPlots, Dates
```

[ ]:
```
df = DataFrame(CSV.File("data/Melbourne_housing_FULL.csv"))
# @transform Distance = :value == "n/a" ? nothing : parse(Float64, :value)
transform!(df, :Distance => ByRow(x -> x == "#N/A" ? missing : parse(Float64,␣
 ↪x)) => :Distance)
#Convert strings to proper Date objects
transform!(df, :Date => ByRow((x) -> Date(x, dateformat"d/m/y")) => :Date)
# Quantise the dates to months. Note this will show as 1/month/year but␣
 ↪represents sales for the month.
transform!(df, :Date => ByRow((x) -> Date(year(x), month(x))) => :Month)
# Drop all entries without price
price_df = dropmissing(df, :Price)
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	
	String31	String31	Int64	String1	Int64	String3	String31	
1	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	...
2	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	...
3	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	...
4	Abbotsford	40 Federation La	3	h	850000	PI	Biggin	...
5	Abbotsford	55a Park St	4	h	1600000	VB	Nelson	...
6	Abbotsford	129 Charles St	2	h	941000	S	Jellis	...
7	Abbotsford	124 Yarra St	3	h	1876000	S	Nelson	...
8	Abbotsford	98 Charles St	2	h	1636000	S	Nelson	...
9	Abbotsford	217 Langridge St	3	h	1000000	S	Jellis	...
10	Abbotsford	18a Mollison St	2	t	745000	S	Jellis	...
11	Abbotsford	6/241 Nicholson St	1	u	300000	S	Biggin	...
12	Abbotsford	10 Valiant St	2	h	1097000	S	Biggin	...
13	Abbotsford	403/609 Victoria St	2	u	542000	S	Dingle	...
14	Abbotsford	25/84 Trenerry Cr	2	u	760000	SP	Biggin	...
15	Abbotsford	106/119 Turner St	1	u	481000	SP	Purplebricks	...
16	Abbotsford	411/8 Grosvenor St	2	u	700000	VB	Jellis	...
17	Abbotsford	40 Nicholson St	3	h	1350000	VB	Nelson	...
18	Abbotsford	123/56 Nicholson St	2	u	750000	S	Biggin	...
19	Abbotsford	22 Park St	4	h	1985000	S	Biggin	...
20	Abbotsford	13/84 Trenerry Cr	1	u	500000	S	Biggin	...
21	Abbotsford	45 William St	2	h	1172500	S	Biggin	...
22	Abbotsford	7/20 Abbotsford St	1	u	441000	SP	Greg	...
23	Abbotsford	16 William St	2	h	1310000	S	Jellis	...
24	Abbotsford	42 Henry St	3	h	1200000	S	Jellis	...
...	...	...	...	...	...	...	...	

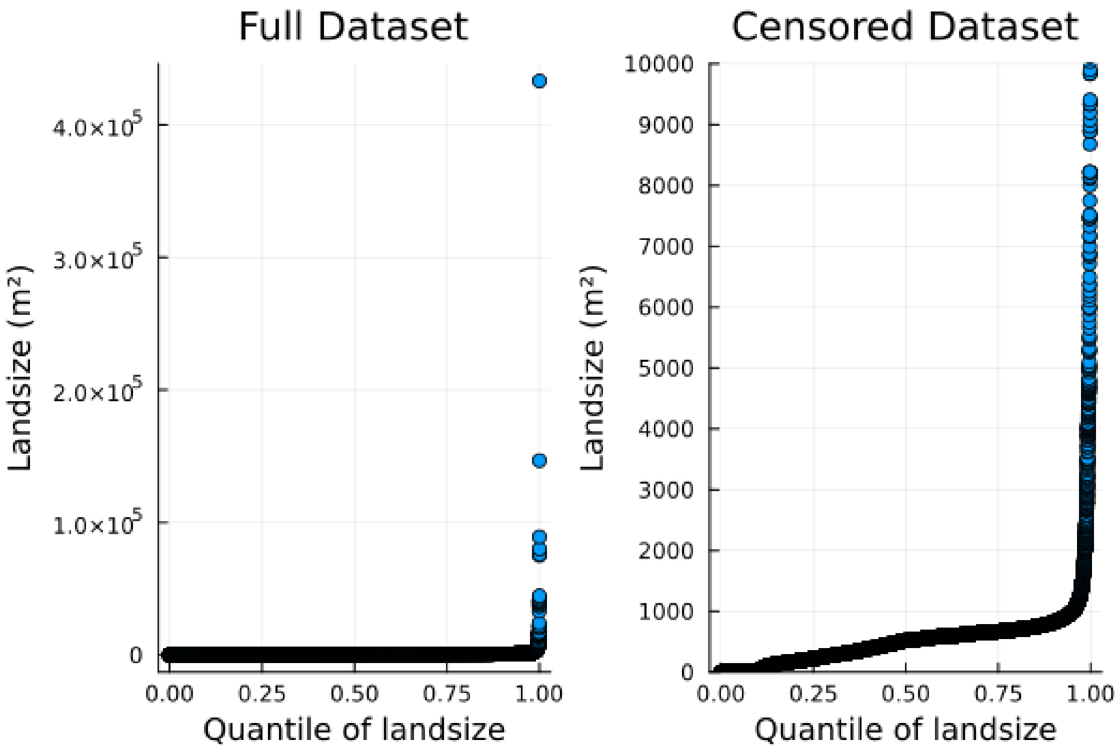
2 1.1a

2.0.1 Landsize

In examining this data, there were some significant outliers on the largest landplot size. As such the full distribution is shown on the left, and the right censors sizes above 10000. The dataset has been plotted as a quantile-quantile plot

[ ]:
```
distance_df = dropmissing(df, :Landsize)
distance_full = scatter([i/length(distance_df.Landsize) for i in 1:
 ↪length(distance_df.Landsize)],
                        sort(distance_df.Landsize), xlabel="Quantile of␣
 ↪landsize", ylabel="Landsize (m²)", legend=false,
                        title="Full Dataset")

distance_truncated = scatter([i/length(distance_df.Landsize) for i in 1:
 ↪length(distance_df.Landsize)],
                             sort(distance_df.Landsize), xlabel="Quantile of␣
 ↪landsize", ylabel="Landsize (m²)",
                             legend=false,
```

```
                                                title="Censored Dataset", ylims=(0,10000), yticks=0:
↪1000:maximum(distance_df.Landsize)+1000)
plot(distance_full, distance_truncated, layout=(1,2))
```

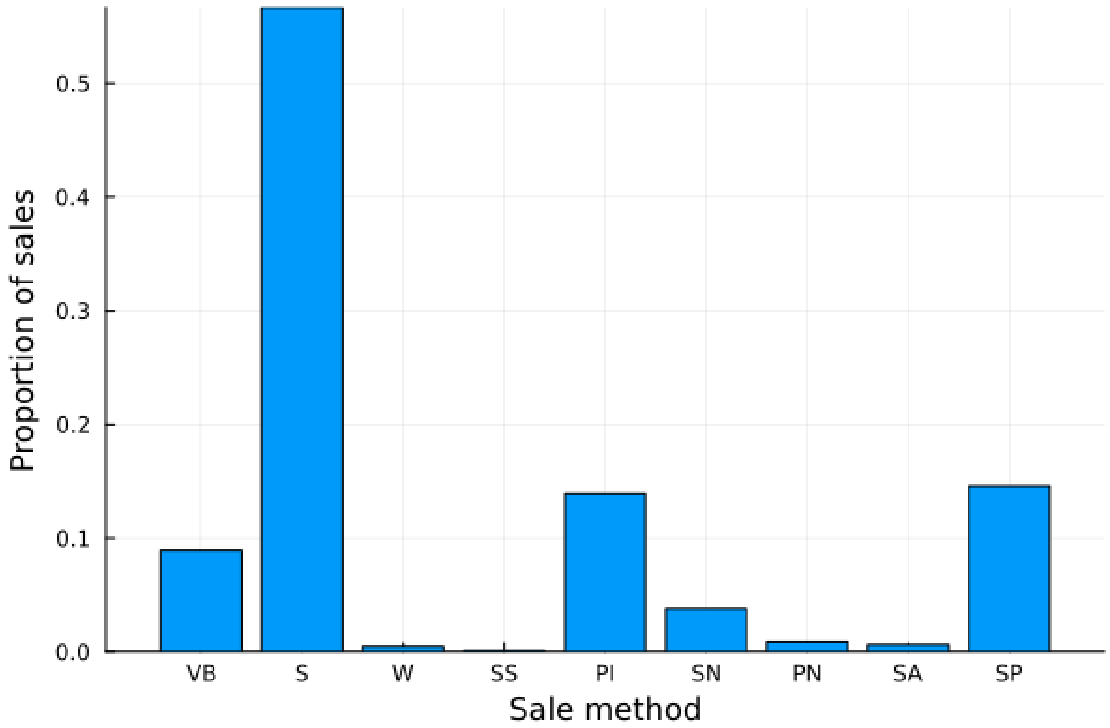


As can be seen, almost all houses (~90%) have less than a plot size of less than 1000m(?)

3 Method

The method has been visualised as a bar chart counting the occurences of each type.

[ ]:
```
method_prop = Dict()
method_count = countmap(df.Method)
total= sum(values(method_count))
for k in keys(method_count)
    method_prop[k] = method_count[k]/total
end
method_prop
bar(method_prop, xlabel="Sale method", ylabel="Proportion of sales",␣
↪legend=false)
```

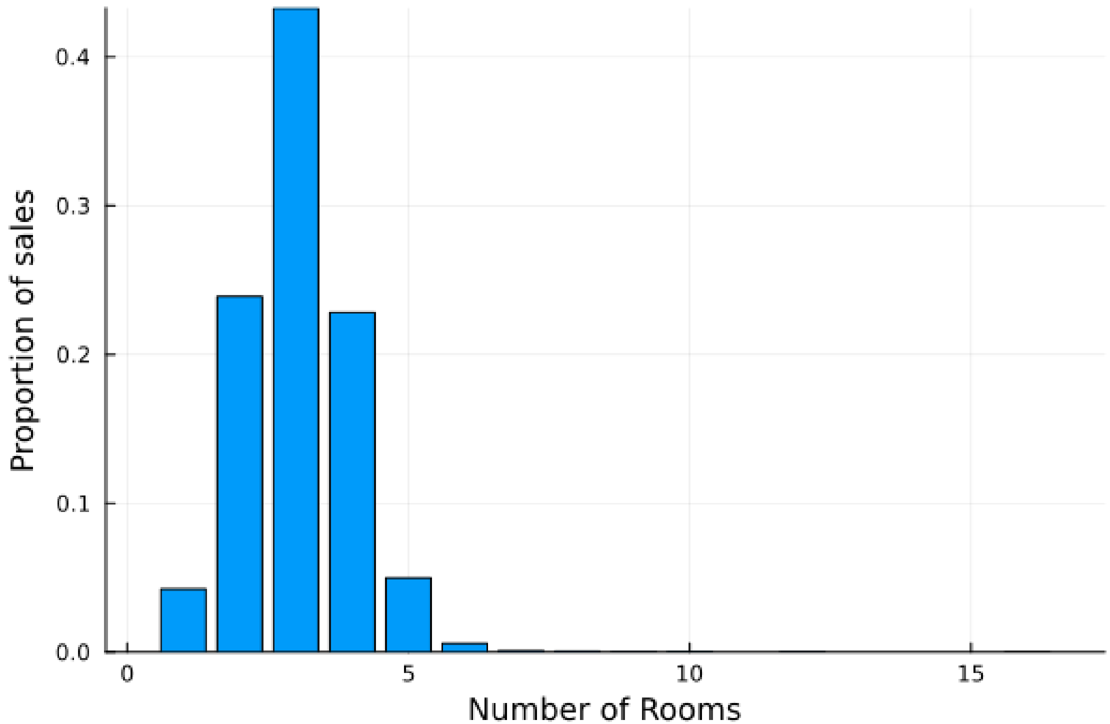


The vast majority of property sales (more than 50%) occur through direct sale

4 Rooms

The proportion of sales for each number of rooms is shown below

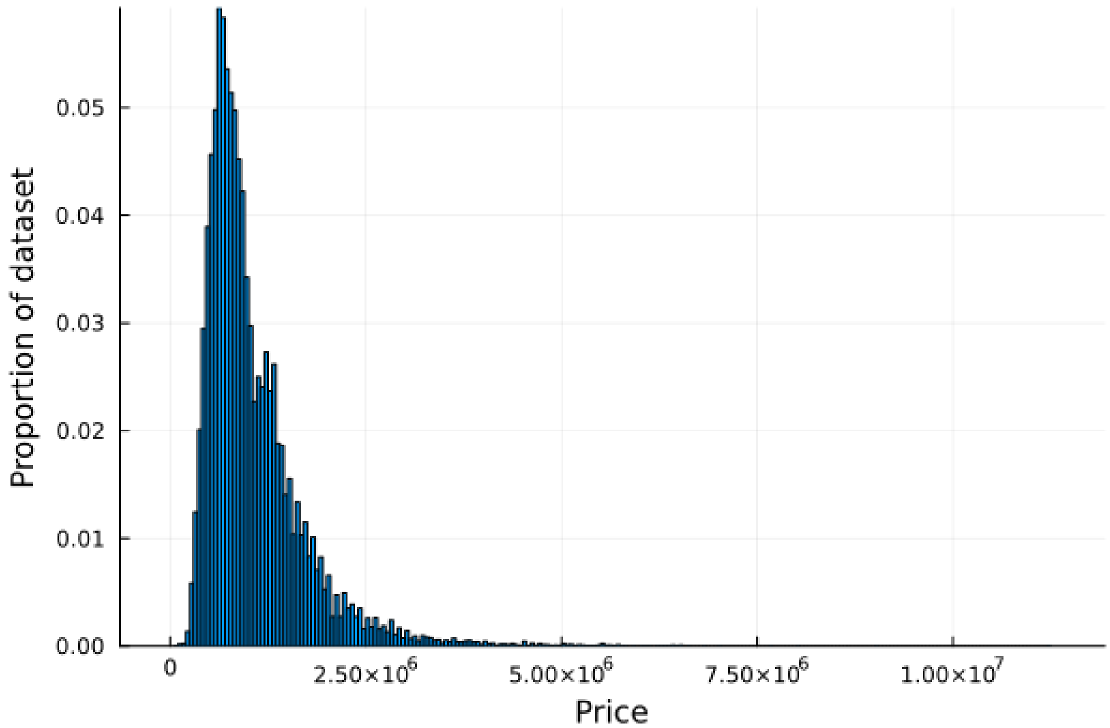
[ ]:
```
room_prop = Dict()
room_count = countmap(df.Rooms)
total= sum(values(room_count))
for k in keys(room_count)
    room_prop[k] = room_count[k]/total
end
room_prop
bar(room_prop, xlabel="Number of Rooms", ylabel="Proportion of sales",␣
  ↪legend=false)
# bar(countmap(df.Rooms), bins=1:20, normalize=:probability)
```



5 Price

A histogram of the price distribution can be seen below. It has been normalised to represent the proportion of houses sold that sold for that given price bucket.

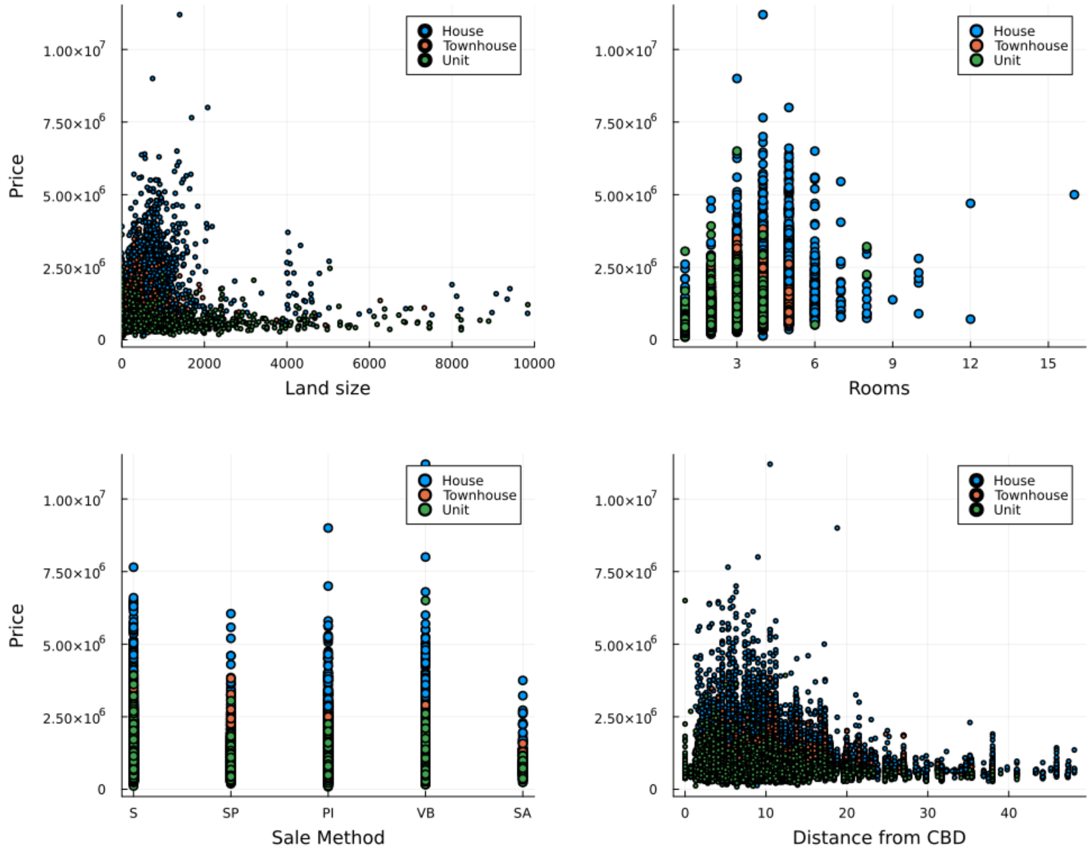
[ ]:
```
histogram(skipmissing(df.Price), normalize=:probability, legend=false,␣
↪xlabel="Price", ylabel="Proportion of dataset")
```



6 1.2a

These plots are presented with grouping by type, since we would expect significant difference between the nature of each property

[ ]:
```
p_land = scatter(price_df.Landsize, price_df.Price, group=price_df.Type, ms=2,␣
↪xlabel="Land size", ylabel="Price", label=["House" "Townhouse" "Unit"],␣
↪xlim=(0,10000))
p_room = scatter(price_df.Rooms, price_df.Price, group=price_df.Type,␣
↪xlabel="Rooms", label=["House" "Townhouse" "Unit"])#, ylabel="Price"
p_method = scatter(price_df.Method, price_df.Price, group=price_df.Type,␣
↪xlabel="Sale Method", ylabel="Price", label=["House" "Townhouse" "Unit"])
p_dist= @df price_df scatter(
    :Distance, :Price, group=:Type, ms=2, label=["House" "Townhouse" "Unit"],␣
↪xlabel="Distance from CBD"
)

plot(p_land, p_room, p_method, p_dist, layout=(2,2), size=(1000,800),␣
↪margin=5Plots.mm)
```

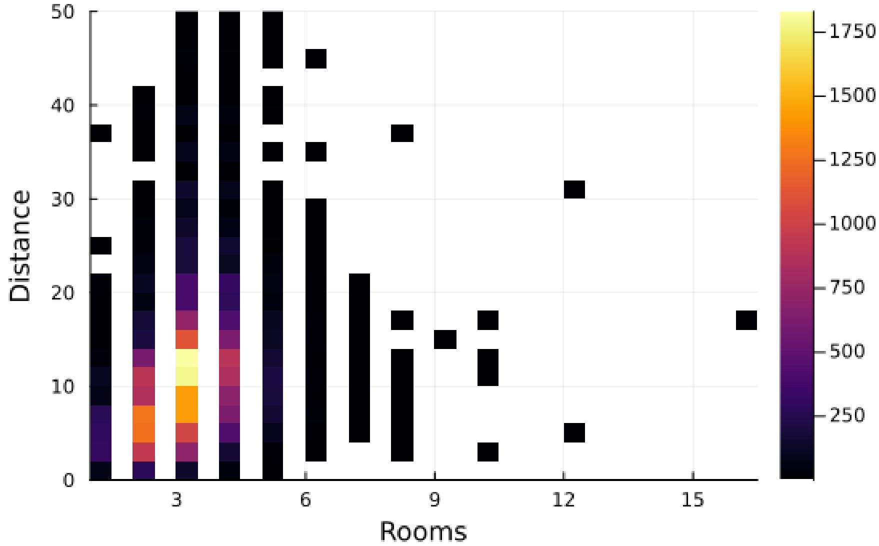


It can be noted across all these graphs, the housing series tended to attain the upper echelon of values, while the units were the cheapest. Trends are not as clear when it comes to other metrics. Room prices are roughly symetric about 4, hwoever the larger room counts do not follow this trend Price does seem to increase as distances increase, most specifically notable for houses. The other two are less clear, but seems though the variance in Unit price increases. Nothing clear about sale method, except that houses were consistently most valuable.

7 Other fun variable relations examined

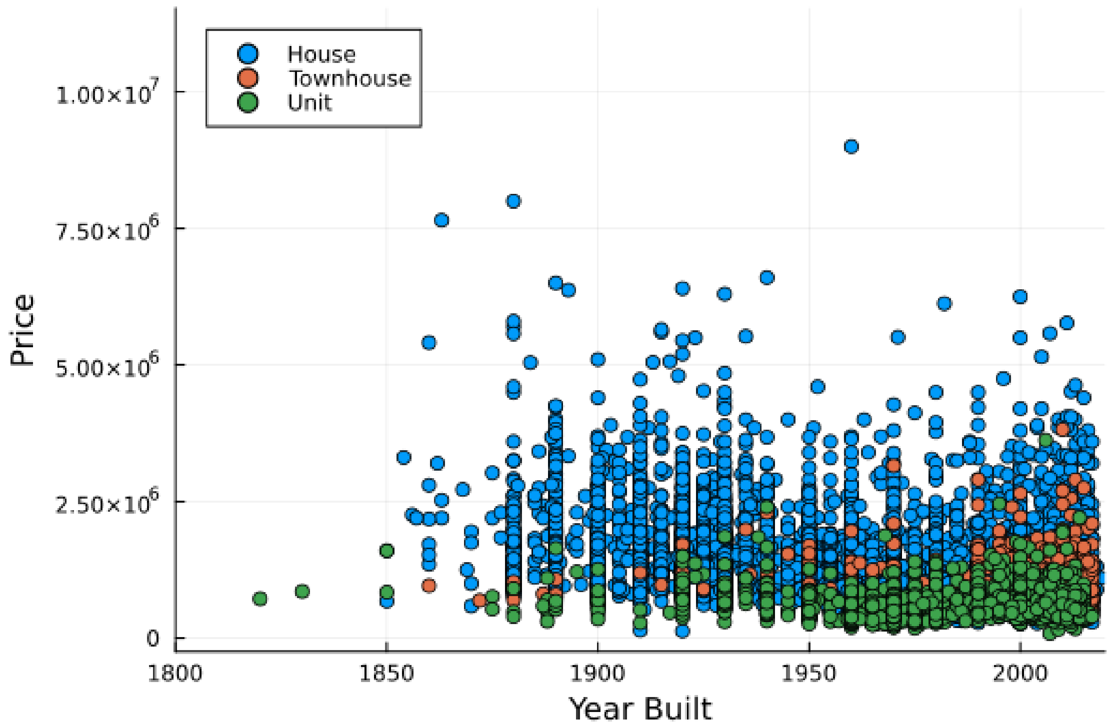
First we have the distribution of Rooms and Distance from cbd and their impact on price.

[ ]:
```
room_distance = dropmissing(df, [:Price, :Rooms, :Distance])
plot(histogram2d(room_distance.Rooms, room_distance.Distance, xlabel="Rooms",␣
  ↪ylabel="Distance"), size=(700,500), margin=20Plots.mm)
```



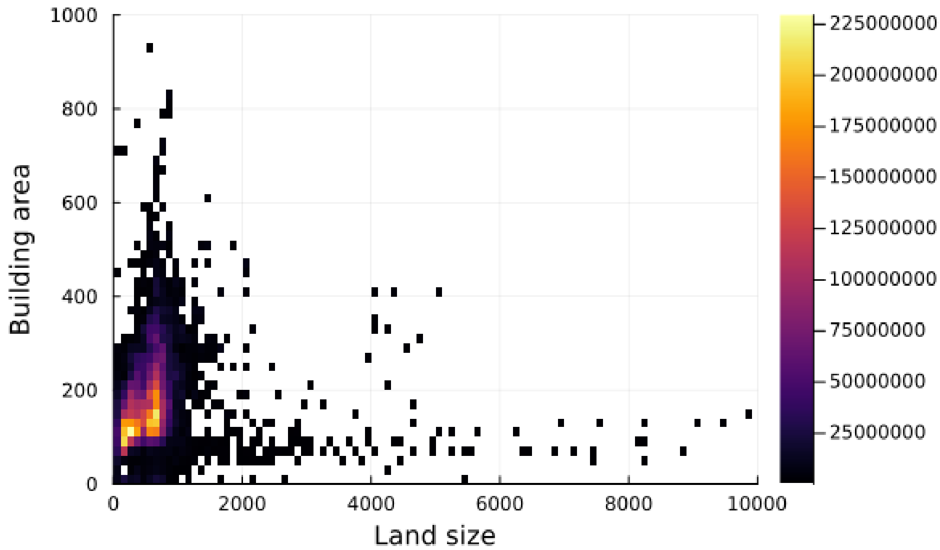
It appears most properties are at distances of up to 10km from the city, with 3 bedrooms. The counts of properties are shown on the colour axis

```
[ ]: @df df scatter(
         :YearBuilt, :Price, group=:Type, xlim=(1800,2020), xlabel="Year Built",␣
      ↪ylabel="Price", label=["House" "Townhouse" "Unit"]
     )
```



Note the bounds have been set to 1800, one house was allegedly built in 1200, however this was omitted. No clear trend can be observed here.

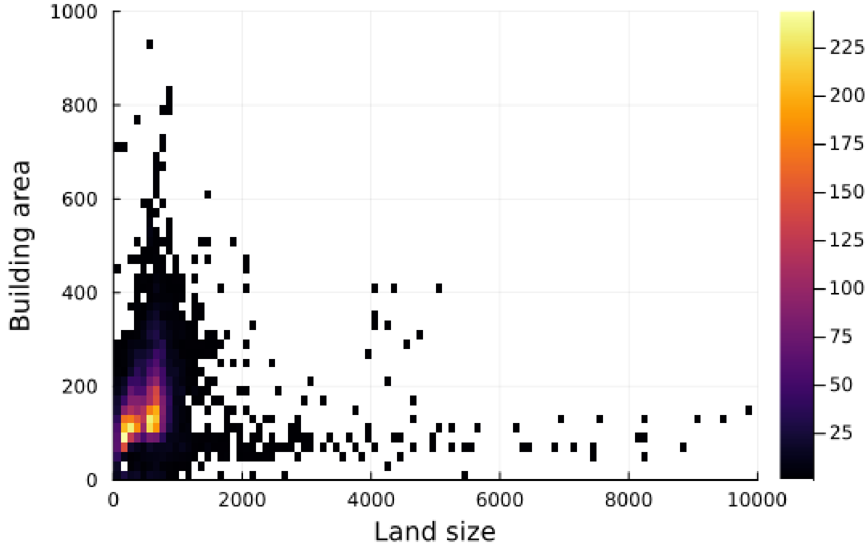
```
[ ]: @df df scatter(
         :BuildingArea, :Price, xlim=(0,1500)
     )
     building_area = dropmissing(df, [:Price, :Landsize, :BuildingArea])
     building_area = combine(groupby(building_area, [:Landsize, :BuildingArea]), :
      ↪Price => mean)
     plot(histogram2d(building_area.Landsize, building_area.BuildingArea,␣
      ↪weights=building_area.Price_mean, xlabel="Land size", ylabel="Building␣
      ↪area", ylim=(0,1000), xlim=(0,10000)), size=(700,500), margin=20Plots.mm)
```



I believe this plot is broken, as its colour axis is orders of magnitude too large. I believe it is blending the prices into bucekts. Irrespective of this, it indicates most sales are occurring with small building area and land sizes

7.1 As counts

[ ]:
```
@df df scatter(
    :BuildingArea, :Price, xlim=(0,1500)
)
building_area = dropmissing(df, [:Price, :Landsize, :BuildingArea])
building_area = combine(groupby(building_area, [:Landsize, :BuildingArea]), :
 ↪Price => mean)
plot(histogram2d(building_area.Landsize, building_area.BuildingArea,␣
 ↪xlabel="Land size", ylabel="Building area", ylim=(0,1000), xlim=(0,10000)),␣
 ↪size=(700,500), margin=20Plots.mm)
```



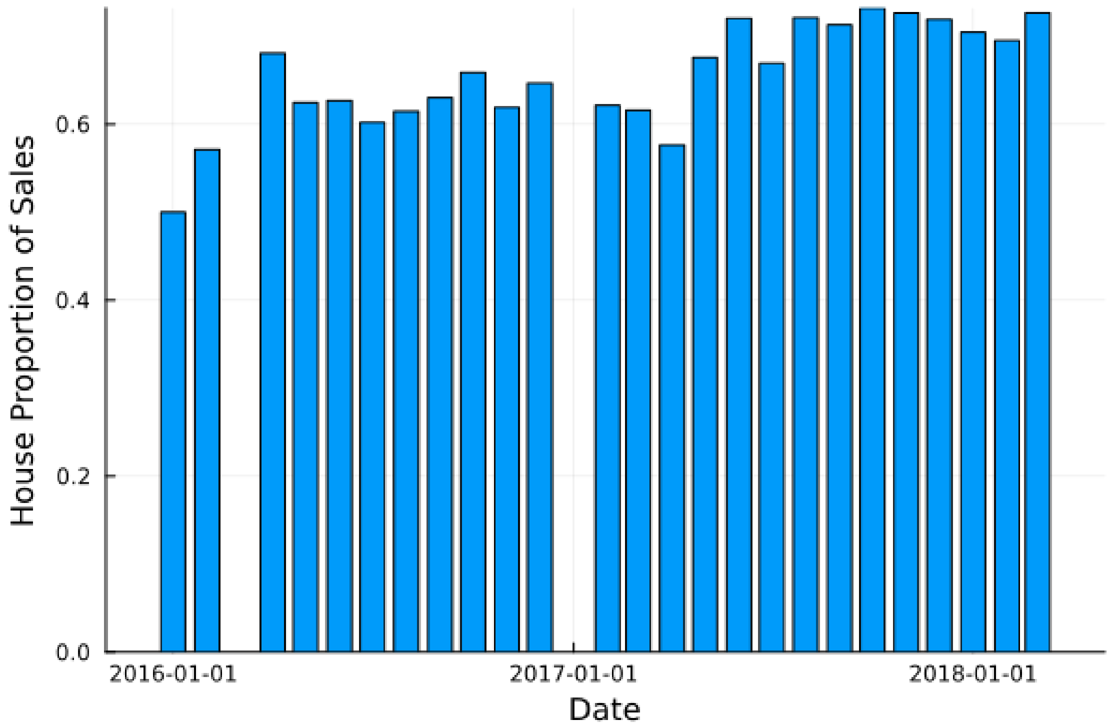
Here it shows the distribution of property sales - most are for smaller properties.

8 1.3a

9 Proportion of House sales

Below is the chart for the proportions of sales houses occupy. It seems as though it has been slowly increasing as the years progress.

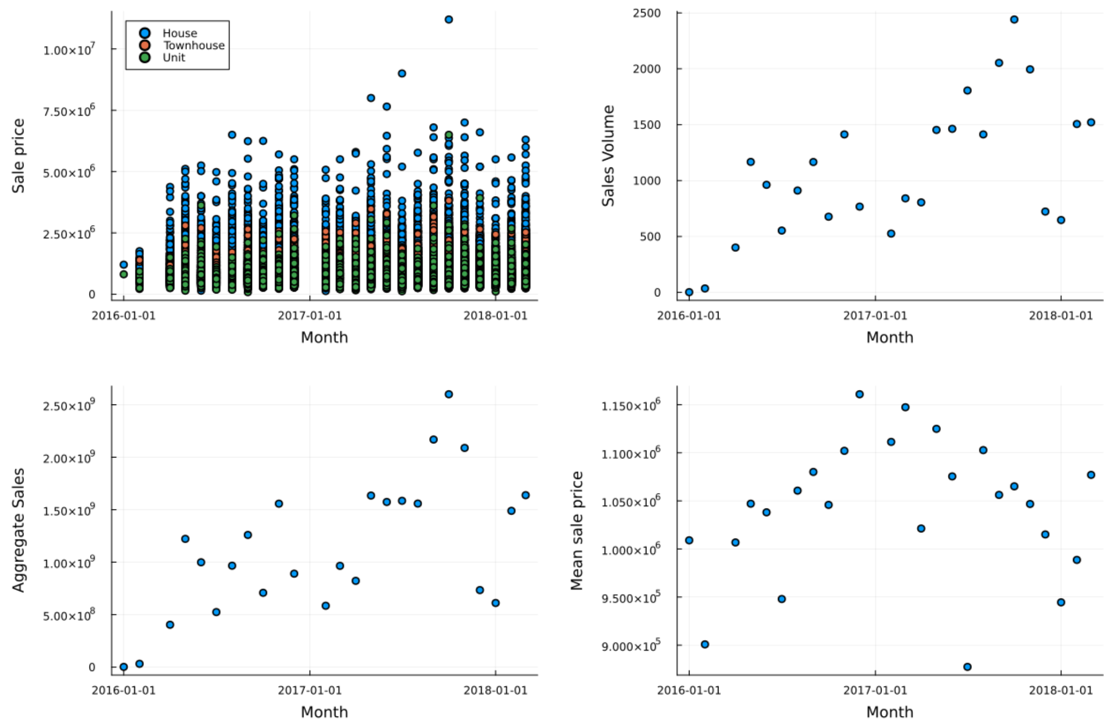
[ ]:
```
gf = groupby(price_df, :Month)
price_sum = combine(gf, :Price => sum)
count_f = combine(groupby(price_df, [:Month, :Type]), :Type => length => :Num)
total_f = combine(groupby(price_df, [:Month]), nrow => :Mnum)
final = innerjoin(count_f, total_f, on= :Month)
final[!, :Prop] = final.Num./final.Mnum

@df filter(:Type => x->x=="h", final) bar(
    :Month,
    :Prop,
    legend=false, xlabel="Date", ylabel="House Proportion of Sales"
)
```



10 Sales volume and revenue

Below is a side-by-side comparison of the total number of sales and the net revenue from those sales. It should be noted these follow the same general trend almost one to one. This makes sense given one must sell a house to make money. This was further examined by considering the mean house price, which does not significantly vary between months implying sales volume is a good indicator of revenue

[ ]:
```
using StatsPlots
c = combine(gf, nrow => :Count)
mean_price = combine(gf, :Price => mean)
raw_sales = @df price_df scatter(:Month, :Price, group=:Type, xlabel="Month",␣
 ↪ylabel="Sale price", label=["House" "Townhouse" "Unit"])
volume_time = scatter(c.Month, c.Count, legend=false, xlabel="Month",␣
 ↪ylabel="Sales Volume")
agg_price_time = scatter(price_sum.Month, price_sum.Price_sum, legend=false,␣
 ↪xlabel="Month", ylabel="Aggregate Sales")
mean_price_time = scatter(mean_price.Month, mean_price.Price_mean,␣
 ↪legend=false, xlabel="Month", ylabel="Mean sale price")
plot(raw_sales, volume_time, agg_price_time, mean_price_time, size=(1200,800),␣
 ↪layout=(2,2), margin=5Plots.mm)
```



11 1.4

Note when I say significant, I refer to the statistical threshold of 0.05

[ ]:
```julia
using GLM
```

[ ]:
```julia
everything_model = lm(@formula(Price ~ Distance + Landsize + Bedroom2 + Car + Rooms + Type + Method + BuildingArea + YearBuilt), price_df)
```

```
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredChol{Float64, LinearAlgebra.CholeskyPivoted{Float64, Matrix{Float64}, Vector{Int64}}}}, Matrix{Float64}}

Price ~ 1 + Distance + Landsize + Bedroom2 + Car + Rooms + Type + Method + BuildingArea + YearBuilt

Coefficients:

                          Coef.     Std. Error       t  Pr(>|t|)        Lower 95%     Upper 95%

(Intercept)          8.57629e6      3.17639e5   27.00    <1e-99        7.95364e6     9.19893e6
```

```
Distance       -32632.9          875.461      -37.28    <1e-99    -34349.0
↪-30916.8
Landsize           27.8362         4.87927      5.70    <1e-07        18.2717
↪    37.4007
Bedroom2       46082.7         20036.6          2.30    0.0215     6806.38
↪85359.1
Car            53652.6          5876.45         9.13    <1e-19    42133.4
↪65171.8
Rooms                1.53102e5  20344.5         7.53    <1e-13         1.13222e5
↪      1.92982e5
Type: t        -14167.1         20741.3        -0.68    0.4946    -54824.8
↪26490.7
Type: u             -1.56044e5  17772.0        -8.78    <1e-17        -1.90881e5
↪    -1.21207e5
Method: S        1905.22        15969.0         0.12    0.9050    -29397.8
↪33208.2
Method: SA      15939.8         62265.4         0.26    0.7980        -1.06115e5
↪      1.37994e5
Method: SP     -72559.5         19845.9        -3.66    0.0003   -111462.0
↪-33657.0
Method: VB      60487.2         22005.8         2.75    0.0060     17350.8
↪      1.03624e5
BuildingArea     2560.54           73.9468     34.63    <1e-99      2415.59
↪ 2705.49
YearBuilt       -4166.54          165.227     -25.22    <1e-99     -4490.42
↪-3842.65
```

From this we refine the model to remove Method and type as it lacks significance. The townhouse type does not have significance, but being a unit is significant. This makes sense due to the significant differences in accomodation style

[ ]:
```
reduced = lm(@formula(Price ~ Distance + Landsize + Rooms +Car + Bedroom2+
↪BuildingArea + YearBuilt), price_df)
```

```
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.
↪DensePredChol{Float64, LinearAlgebra.CholeskyPivoted{Float64, Matrix{Float64},
↪Vector{Int64}}}}, Matrix{Float64}}

Price ~ 1 + Distance + Landsize + Rooms + Car + Bedroom2 + BuildingArea +
↪YearBuilt

Coefficients:

                       Coef.     Std. Error       t  Pr(>|t|)       Lower 95%
↪    Upper 95%
```

```
(Intercept)        9.2926e6       2.88094e5   32.26    <1e-99       8.72787e6  ␣
 ↪    9.85733e6
Distance       -31474.3          840.463      -37.45    <1e-99  -33121.8      ␣
 ↪-29826.8
Landsize           25.7742         4.89073      5.27    <1e-06      16.1872    ␣
 ↪  35.3611
Rooms               1.74489e5  20366.8          8.57    <1e-16       1.34565e5  ␣
 ↪    2.14413e5
Car            57999.2          5891.05         9.85    <1e-22   46451.4       ␣
 ↪69547.0
Bedroom2       52932.5         20140.8          2.63    0.0086   13451.9       ␣
 ↪92413.2
BuildingArea    2629.49           74.0112      35.53    <1e-99    2484.41      ␣
 ↪2774.57
YearBuilt      -4606.51          147.072      -31.32    <1e-99   -4894.81      ␣
 ↪-4318.21
```

All of these variables are significant in terms of P values. Landsize has a small coefficient. Recall most sales occur over a small window of land sizes, while few have significantly larger sizes.

[ ]:
```julia
reduced = lm(@formula(Price ~ Distance + Landsize + Rooms + Bedroom2 +␣
 ↪BuildingArea + Car+YearBuilt), price_df)
```

```
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.
 ↪DensePredChol{Float64, LinearAlgebra.CholeskyPivoted{Float64, Matrix{Float64},␣
 ↪Vector{Int64}}}}, Matrix{Float64}}

Price ~ 1 + Distance + Landsize + Rooms + Bedroom2 + BuildingArea + Car +␣
 ↪YearBuilt

Coefficients:

                       Coef.      Std. Error        t   Pr(>|t|)       Lower 95%  ␣
 ↪    Upper 95%

(Intercept)        9.2926e6       2.88094e5   32.26    <1e-99       8.72787e6  ␣
 ↪    9.85733e6
Distance       -31474.3          840.463      -37.45    <1e-99  -33121.8      ␣
 ↪-29826.8
Landsize           25.7742         4.89073      5.27    <1e-06      16.1872    ␣
 ↪  35.3611
Rooms               1.74489e5  20366.8          8.57    <1e-16       1.34565e5  ␣
 ↪    2.14413e5
Bedroom2       52932.5         20140.8          2.63    0.0086   13451.9       ␣
 ↪92413.2
```

```
BuildingArea    2629.49          74.0112      35.53    <1e-99    2484.41
  2774.57
Car            57999.2         5891.05        9.85    <1e-22   46451.4
  69547.0
YearBuilt      -4606.51         147.072      -31.32    <1e-99   -4894.81
  -4318.21
```

Consider a model based purely on attributes which impact the size of the house:

[ ]:
```julia
house_size = lm(@formula(Price ~ Landsize + Rooms + Bedroom2 + Car +
    BuildingArea), price_df)
```

```
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.
  DensePredChol{Float64, LinearAlgebra.CholeskyPivoted{Float64, Matrix{Float64},
  Vector{Int64}}}}, Matrix{Float64}}

Price ~ 1 + Landsize + Rooms + Bedroom2 + Car + BuildingArea

Coefficients:

                     Coef.  Std. Error      t  Pr(>|t|)    Lower 95%      Upper
  95%

(Intercept)   55772.4       21270.9      2.62    0.0088   14076.8      97468.1
Landsize         -4.92995       5.5713  -0.88    0.3762     -15.8509       5.
  99103
Rooms             3.04549e5  24470.0     12.45    <1e-34  256582.0          3.
  52515e5
Bedroom2      20747.8       24452.1      0.85    0.3962  -27183.7      68679.4
Car           13900.3        7026.58     1.98    0.0479     126.667   27674.0
BuildingArea     52.3667       14.0532   3.73    0.0002      24.8193     79.914
```

In terms of considering house size based attributes only, landsize and the existence of second bedrooms lose their statistical significance. The bedroom loses significance since it is linked naturally to the Rooms variable. As such we will exclude them in future

[ ]:
```julia
final_model = lm(@formula(Price ~ Distance + Car + Rooms + YearBuilt +
    BuildingArea), price_df)
```

```
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.
  DensePredChol{Float64, LinearAlgebra.CholeskyPivoted{Float64, Matrix{Float64},
  Vector{Int64}}}}, Matrix{Float64}}

Price ~ 1 + Distance + Car + Rooms + YearBuilt + BuildingArea

Coefficients:
```

```
                       Coef.     Std. Error       t  Pr(>|t|)       Lower 95%     ␣
 ↪  Upper 95%

(Intercept)        9.56844e6      2.72368e5   35.13    <1e-99       9.03455e6     ␣
 ↪  1.01023e7
Distance       -30434.9          755.199      -40.30    <1e-99  -31915.3         ␣
 ↪-28954.6
Car             58365.3         5540.83        10.53    <1e-25   47504.2          ␣
 ↪69226.4
Rooms              2.19954e5   6774.17         32.47    <1e-99       2.06676e5 ␣
 ↪233233.0
YearBuilt       -4739.18         139.001      -34.09    <1e-99   -5011.65         ␣
 ↪-4466.71
BuildingArea    2701.11           70.0674      38.55    <1e-99    2563.76          ␣
 ↪2838.45
```

This final model has all significant values.