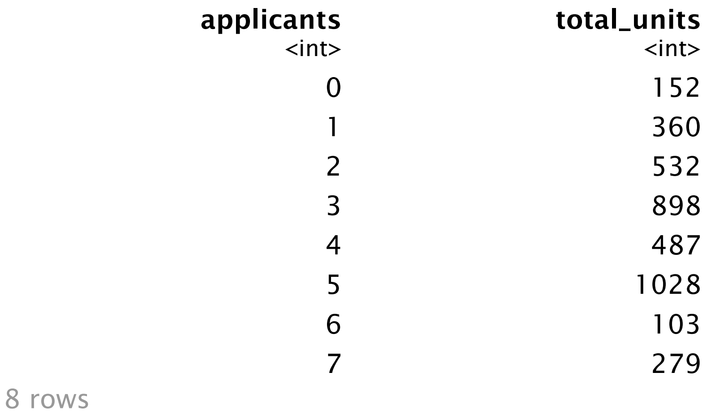
Apartment applicants – Solution

Task 1 – Provide a summary of the target variable (5 points)

The data consists of 1,430 observations of apartment buildings which have sold. This includes 3,839 individual apartment units. We are interested in the number of applicants per unit, and so we use a weighted average of applicants by num_units and see that this is 3.6.

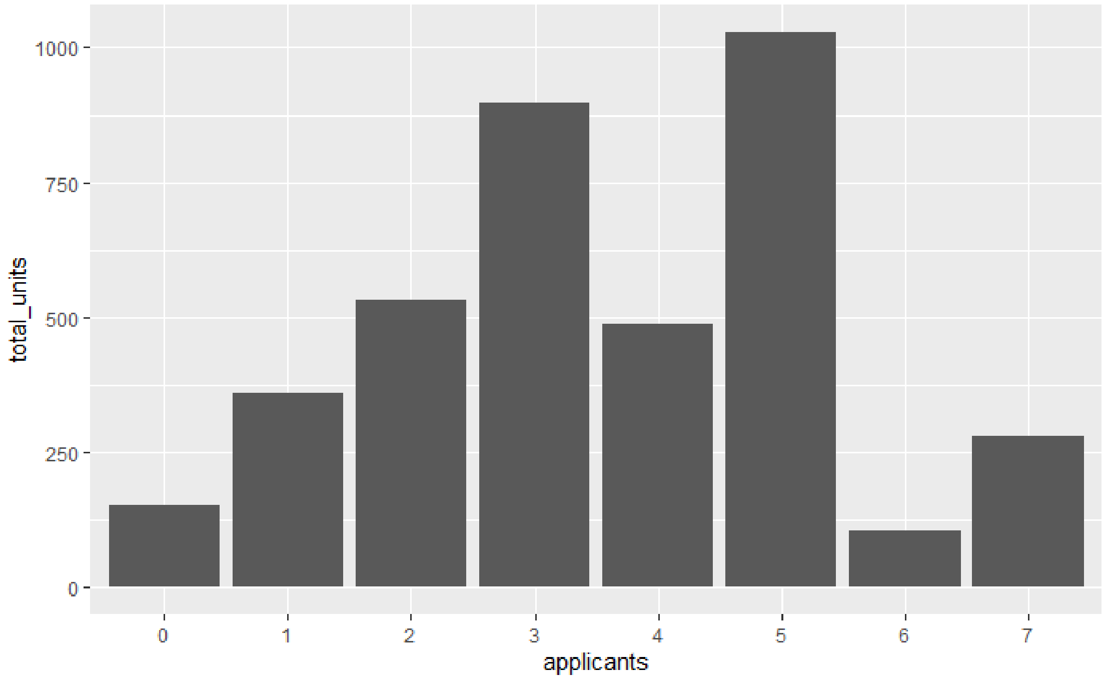
I am interested in the distribution of the number of applicants. I want to know how many units had 0 applicants, 1 applicant, 2 applicants, and so fourth. I see the code provided gives me this info.



applicants <int>	total_units <int>
0	152
1	360
2	532
3	898
4	487
5	1028
6	103
7	279

8 rows

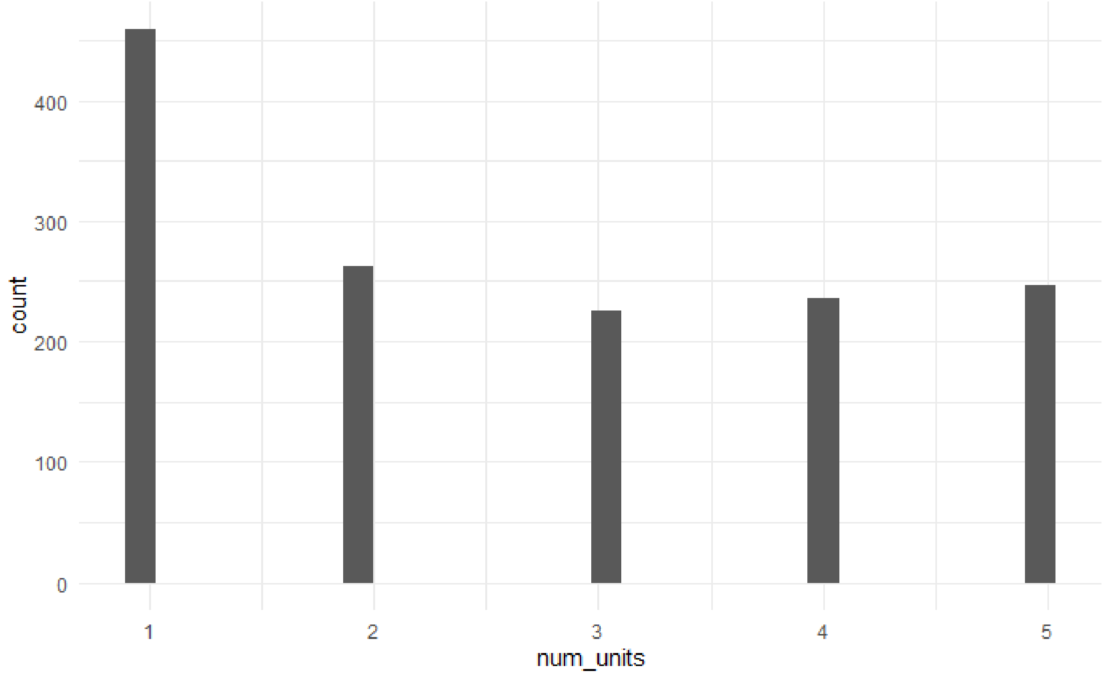
The histogram is incorrect because it shows the count of the number of records instead of the number of units on the y-axis. The bar plot below shows the total units in each apartment building, which is what the client asked to see. The applicants range from 0 to 7 on the x-axis.



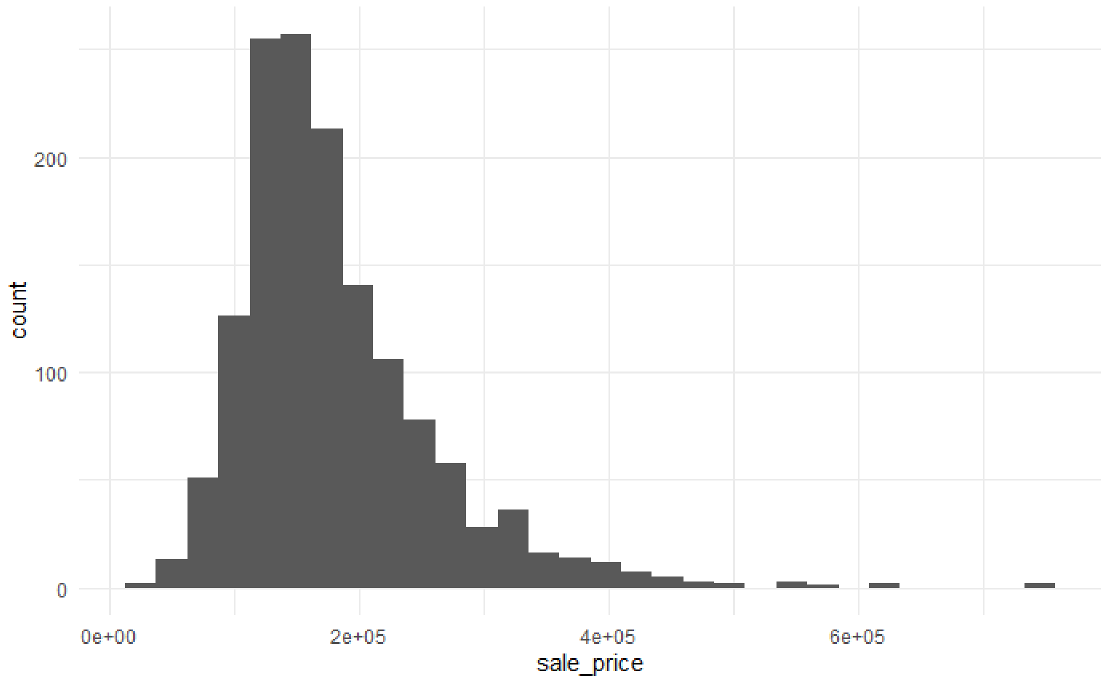
Task 2 – Explore the predictor variables (10 points)

Because the client asked us to use the number of units as the weights, I'm using the weighted averages as opposed to the straight averages. The mean sale price is $181,146 and the mean overall quality is 6.06.

The number of units ranges from 1 to 5, with most buildings having 1 unit.

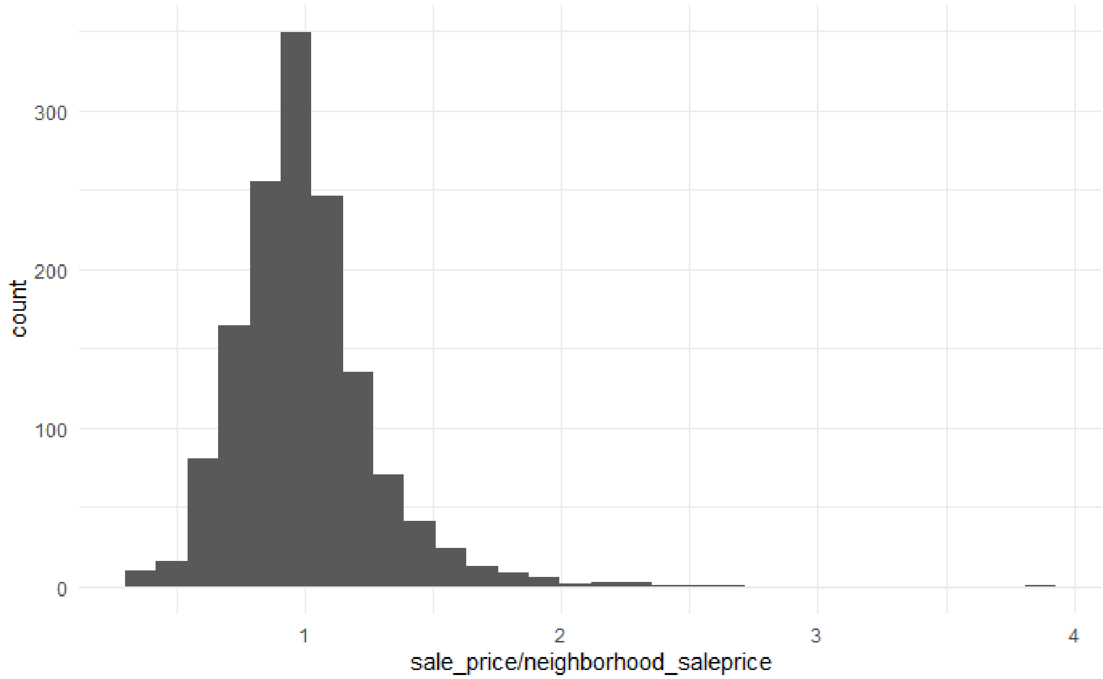


The sale price is positive and right skewed. The below histogram shows the unweighted distribution. To correct this skewness, I apply a log transform.

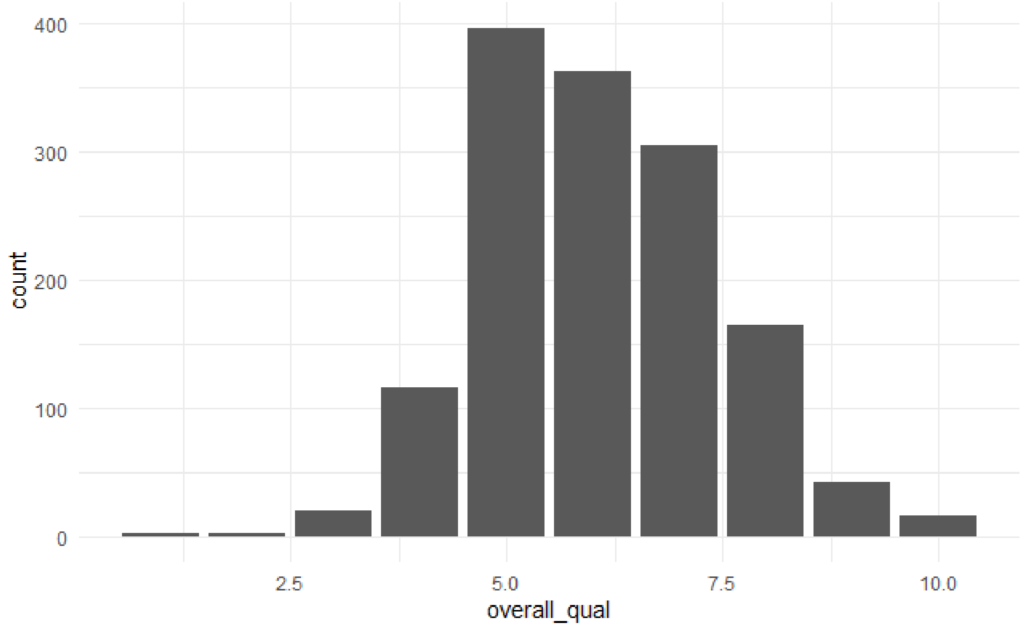


I adjusted the sale price so that it shows the relative price to other apartments in the same neighborhood. I did this by dividing by the mean sale price of the neighboring apartments. Then I applied the log transform. Then I removed the original sale_price and neighborhood_sale_price variables.

I expect that this relative sale price will be predictive of demand and that apartments which are relatively less expensive than their neighbors will have many people apply for them. The number of units is not a predictor variable but a weighting variable. When an apartment has more units there will automatically be more people who apply.



The overall quality measures the state of repair on the apartment. This ranges from 0 to 10, and is approximately symmetric, so no transformation is applied. It has a mean os 6.082 (unweighted) and a weighted mean of 6.06.



Task 3 – Use insights from the Marketing manager (5 points)

I added the insights from the marketing manager into the data in three new ways. This expands on the information which was already available by taking into account the facts about apartments.

First, I calculated the sale price per square foot for each apartment. Because larger apartments which have a low selling price will be in high demand, I expect that more people will apply for these. Conversely, smaller apartments which have a high selling price will be in lower demand.

Secondly, I flagged apartments which are likely to have been rented by students based on when the lease was signed in July or August.

Finally, I calculated the number of bathrooms per square foot. People like to have their own private bathroom and so larger apartments with greater square footage will also take into account the number of half bathrooms and full-sized bathrooms.

This info will improve our predictive models because these calculations will take into account multiple factors at once.

Task 4 – Inspect the garage_type variables (5 points)

I inspected the data found 15 records which had all garage variables listed as zero. These must be in error and so I removed them. The data dictionary says that every apartments needs to have either an attached garage, a basement garage, a detached garage, built-in garage, or no garage at all. However, these 15 records had zeros for all of these values.

Task 5 – Select GLM parameters (10 points)

The target variable is the number of people who are applying for the apartment, which is discrete, and so before fitting the model, I can rule out the continuous distributions of Gaussian, inverse gaussian, and Gamma. There are 8 unique values from 0 to 7, and so I also rule out the binomial, which only takes on 2 values. This only leaves the Poisson, which is appropriate because it models a number of events in a given interval of time, such as the number of customers at a checkout or the number of claims.

There are two ways of modeling the number of applicants. The first approach is to use a *counting model*, where the target is the total number of applicants and the log of the number of units is an offset. An offset is a fixed term that is added to the linear predictor. This can also be thought of as the variable log(num_units) being added with a coefficient of 1.0. The second approach is to use a *frequency* model where the target is the average number of applicants per unit, and the weights are the number of units. Both of these models would yield the same predictions. I chose to use the first type. This means that I do not need to use a weighting term in the model.

The only link function which results in multiplicative model is the log link. The inverse of the log is the exponential, and so the resulting GLM can be expressed as a product of factors, called relativities. The log link is also the canonical link for the Poisson, which makes the model more likely to converge.

Task 6 - Fit a GLM (10 points)

I split the data into training and test sets and fit a GLM on the training data.

I trained the glm on the training data. Following my selections from task 5, I used the log of the num_units as the offset term and removed this variable from being one of the predictors. I noticed that there was a linear dependency between the neighborhood columns because they formed an over-determined system of equations (i.e., they were not linearly independent), and so I removed the north ames neighbhorhood. This had the most observations, and so the intercept term which has the base level is the most stable. I also removed the mean_sale_price variable because this was linearly dependent on the neighborhood variables. Finally, the garage type variable was also set to so that the reference level is garage_type_attachd, which has the most observations.

The AIC is 5,202. This is a measure of how well the model fits the data when taking into consideration the number of parameters in the model as well as the log likelihood. A lower value is better. I also note that my three engineered features have p-values less than 0.001, which indicates that they are predictive.

```
Call:
glm(formula = applicants ~ . + offset(log(num_units)) - neighborhood_n_ames - 
    garage_type_attchd - num_units - log_sale_price, family = poisson(link = "log"), 
    data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5500  -0.8463   0.1564   1.1433   2.8501  

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                 1.598e+01  2.188e+01   0.730 0.465279    
year_sold                  -8.963e-03  1.089e-02  -0.823 0.410353    
month_sold                 -1.302e-02  6.444e-03  -2.021 0.043241 *  
overall_qual               -3.946e-02  2.455e-02  -1.607 0.107973    
total_sq_feet               7.009e-04  1.222e-04   5.734 9.83e-09 ***
gr_liv_area                -8.316e-05  7.742e-05  -1.074 0.282761    
tot_bathrooms              -3.907e-01  1.071e-01  -3.650 0.000263 ***
lot_area                    3.867e-05  2.318e-05   1.668 0.095231 .  
exter_qual                  2.414e-01  4.818e-02   5.011 5.42e-07 ***
full_bath                   1.638e-02  2.481e-02   0.660 0.509254    
central_airno              -1.780e+00  1.616e-01 -11.012  < 2e-16 ***
garage_type_basment        -3.022e-01  1.712e-01  -1.765 0.077508 .  
garage_type_builtIn        -3.153e-01  7.828e-02  -4.027 5.64e-05 ***
garage_type_detchd         -3.110e-01  5.093e-02  -6.106 1.02e-09 ***
garage_type_no_garage      -3.166e-01  9.522e-02  -3.325 0.000885 ***
NeighborhoodBrDale         -4.813e-01  2.528e-01  -1.904 0.056940 .  
neighborhood_brk_side      -1.827e-01  1.121e-01  -1.630 0.103207    
neighborhood_clear_cr      -1.527e-01  1.144e-01  -1.334 0.182168    
neighborhood_collg_cr      -9.105e-02  7.137e-02  -1.276 0.202018    
neighborhood_crawfor       -9.837e-02  1.025e-01  -0.960 0.336973    
neighborhood_edwards        5.976e-02  8.650e-02   0.691 0.489658    
neighborhood_gilbert       -4.415e-02  9.124e-02  -0.484 0.628504    
neighborhood_idottrr       -1.096e+00  2.376e-01  -4.614 3.95e-06 ***
neighborhood_meadowv       -2.908e-01  1.892e-01  -1.537 0.124232    
neighborhood_mitchel       -2.643e-01  9.916e-02  -2.666 0.007683 ** 
neighborhood_n_ridge       -3.659e-01  1.230e-01  -2.974 0.002937 ** 
neighborhood_n_ridge_hghts -3.663e-01  1.115e-01  -3.285 0.001020 ** 
neighborhood_n_w_ames      -4.150e-02  8.234e-02  -0.504 0.614255    
neighborhood_old_town      -1.727e-02  8.711e-02  -0.198 0.842812    
neighborhood_sawyer        -4.492e-01  9.964e-02  -4.508 6.55e-06 ***
neighborhood_sawyer_w      -7.559e-02  8.902e-02  -0.849 0.395821    
neighborhood_somerst       -3.964e-02  9.281e-02  -0.427 0.669305    
neighborhood_stone_br      -6.524e-02  1.421e-01  -0.459 0.646253    
neighborhood_swisu          2.335e-01  1.551e-01   1.505 0.132227    
neighborhood_timber        -2.670e-01  1.144e-01  -2.334 0.019603 *  
neighborhood_veenker       -2.984e-01  1.520e-01  -1.963 0.049675 *  
log_rel_price              -6.434e-01  3.318e-01  -1.939 0.052469 .  
sale_price_per_sqft        -2.118e+03  1.129e+03  -1.875 0.060755 .  
bath_pr_sqft                1.468e+03  3.645e+02   4.026 5.67e-05 ***
student_apt                 5.287e-01  3.440e-02  15.367  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2857.7  on 1144  degrees of freedom
Residual deviance: 1709.1  on 1105  degrees of freedom
AIC: 5202.9

Number of Fisher Scoring iterations: 5
```

Task 7 – Use AIC to select features (5 points)

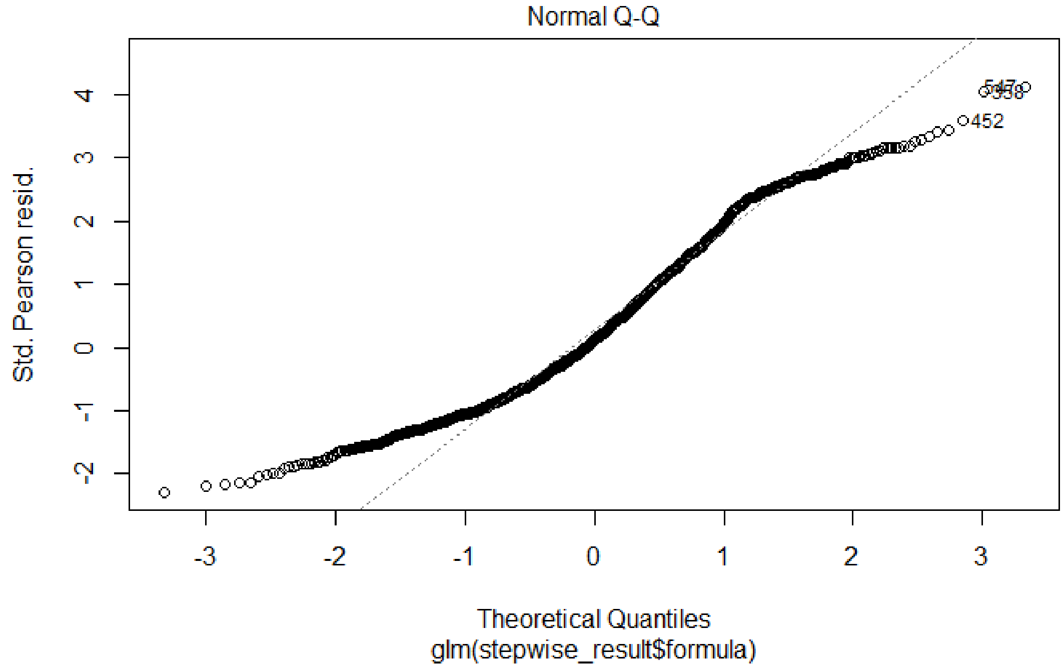
The step AIC procedure begins with all of the variables that I used in task 6 and removes those which are not improving the AIC. This makes the result better because the AIC increases to 5225.

```
Call:
glm(formula = stepwise_result$formula, family = glm$family, data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.4354  -0.8492   0.1187   1.1505   3.0177  

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                 3.857e+00  2.187e+01   0.176 0.860029    
year_sold                  -2.805e-03  1.088e-02  -0.258 0.796525    
month_sold                 -7.662e-03  6.155e-03  -1.245 0.213229    
overall_qual               -4.240e-02  2.477e-02  -1.712 0.086970 .  
total_sq_feet               6.121e-04  1.233e-04   4.964 6.90e-07 ***
gr_liv_area                -2.766e-05  7.907e-05  -0.350 0.726441    
tot_bathrooms              -2.951e-01  1.062e-01  -2.778 0.005472 ** 
lot_area                    8.309e-06  2.345e-05   0.354 0.723119    
exter_qual                  1.622e-01  4.571e-02   3.549 0.000386 ***
full_bath                   9.631e-03  2.479e-02   0.388 0.697693    
central_airno              -1.776e+00  1.607e-01 -11.057  < 2e-16 ***
garage_type_basment        -6.449e-01  1.789e-01  -3.605 0.000313 ***
garage_type_builtIn        -2.743e-01  7.485e-02  -3.665 0.000248 ***
garage_type_detchd         -3.403e-01  4.937e-02  -6.893 5.46e-12 ***
garage_type_no_garage      -3.882e-01  9.569e-02  -4.057 4.97e-05 ***
NeighborhoodBrDale         -4.743e-01  2.507e-01  -1.892 0.058506 .  
neighborhood_brk_side      -1.906e-01  1.091e-01  -1.747 0.080628 .  
neighborhood_clear_cr      -4.249e-03  1.166e-01  -0.036 0.970937    
neighborhood_collg_cr      -4.005e-02  7.142e-02  -0.561 0.574956    
neighborhood_crawfor       -6.505e-02  1.031e-01  -0.631 0.528105    
neighborhood_edwards        7.106e-02  8.121e-02   0.875 0.381560    
neighborhood_gilbert        5.613e-02  8.456e-02   0.664 0.506819    
neighborhood_idottrr       -8.628e-01  2.035e-01  -4.240 2.23e-05 ***
neighborhood_meadowv       -3.219e-01  1.975e-01  -1.630 0.103083    
neighborhood_mitchel       -1.107e-01  9.416e-02  -1.176 0.239554    
neighborhood_n_ridge       -2.067e-01  1.236e-01  -1.672 0.094617 .  
neighborhood_n_ridge_hghts -2.668e-01  1.113e-01  -2.397 0.016523 *  
neighborhood_n_w_ames      -5.038e-02  8.046e-02  -0.626 0.531196    
neighborhood_sawyer        -4.065e-01  9.743e-02  -4.172 3.02e-05 ***
neighborhood_sawyer_w      -1.528e-02  8.376e-02  -0.182 0.855224    
neighborhood_somerst        3.898e-02  9.300e-02   0.419 0.675099    
neighborhood_stone_br      -6.644e-02  1.437e-01  -0.462 0.643799    
neighborhood_swisu          2.700e-01  1.470e-01   1.836 0.066339 .  
neighborhood_timber        -1.111e-01  1.175e-01  -0.946 0.344125    
neighborhood_veenker       -2.594e-01  1.923e-01  -1.349 0.177348    
log_rel_price              -7.163e-01  3.295e-01  -2.174 0.029697 *  
sale_price_per_sqft        -1.245e+03  1.114e+03  -1.117 0.263939    
bath_pr_sqft                1.161e+03  3.631e+02   3.198 0.001382 ** 
student_apt                 5.010e-01  3.450e-02  14.523  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2784.1  on 1144  degrees of freedom
Residual deviance: 1723.6  on 1106  degrees of freedom
AIC: 5225.8

Number of Fisher Scoring iterations: 5
```

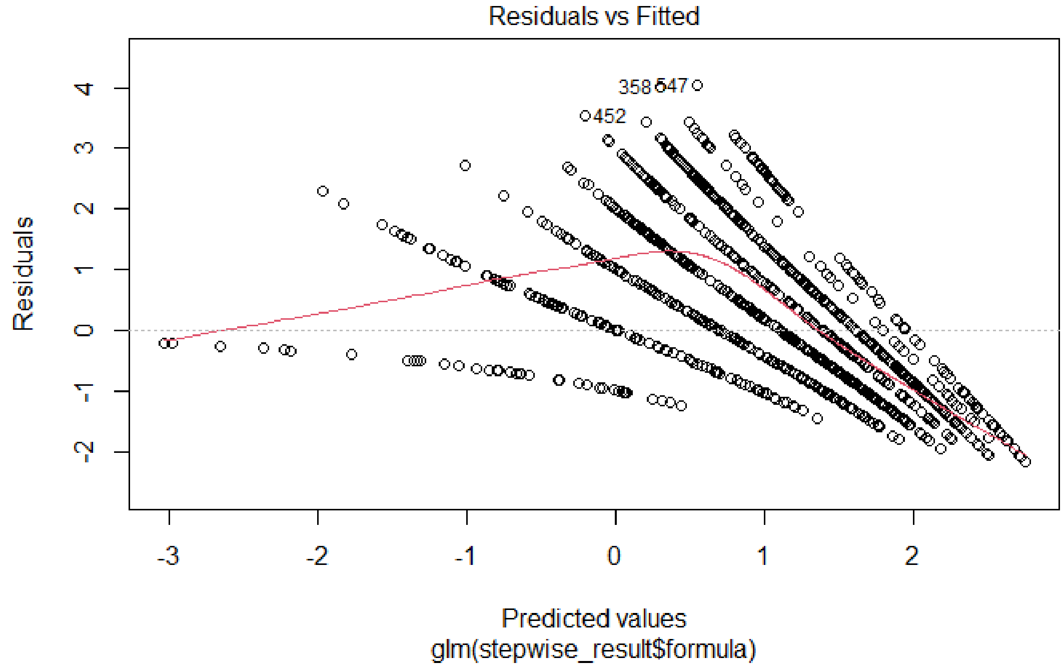
I interpret the coefficients has having a percentage change on the target variable. Because the log link function is used, I need to first take the exponent of the coefficients and then subtract 1. This that as the over_qual increases by one unit, the number of applicants decreases by 4.85%. As the tot_bathrooms increases by 1 unit, the number of applicants decreases by 20.58%. The other variables can be interpreted in the same way. For the neighborhood or garage type variables, the coefficients represent the percentage change over the reference (base) level, which is the North Ames Neighborhood and apartments which have an attached garage. For instance, apartments in Brook Side have 19.71% lower predicted number of applicants than North Ames, and apartments with a built in garage have 25.41% lower predicted number of applicants. The interpretation for bathrooms per square foot was more ambiguous because of the large size of the coefficient. We see that the sign is positive, indicating that having more bathrooms per square foot increases the number of applicants.

Variable	Coefficient	Impact on Applicants
(Intercept)	-2.20E+00	-88.96%
overall_qual	-4.97E-02	-4.85%
total_sq_feet	6.17E-04	0.06%
tot_bathrooms	-2.30E-01	-20.58%
exter_qual	1.69E-01	18.45%
central_airno	-1.80E+00	-83.39%
garage_type_basment	-6.48E-01	-47.71%
garage_type_builtIn	-2.93E-01	-25.41%
garage_type_detchd	-3.29E-01	-28.05%
garage_type_no_garage	-3.73E-01	-31.16%
NeighborhoodBrDale	-5.13E-01	-40.12%
neighborhood_brk_side	-2.20E-01	-19.71%
neighborhood_gilbert	9.72E-02	10.20%
neighborhood_idottrr	-8.89E-01	-58.89%
neighborhood_meadowv	-3.49E-01	-29.43%
neighborhood_n_ridge	-1.76E-01	-16.15%
neighborhood_n_ridge_hghts	-2.25E-01	-20.17%
neighborhood_sawyer	-3.95E-01	-32.65%
neighborhood_swisu	2.80E-01	32.27%
log_rel_price	-1.00E+00	-63.25%
bath_pr_sqft	9.27E+02	
student_apt	4.89E-01	63.13%

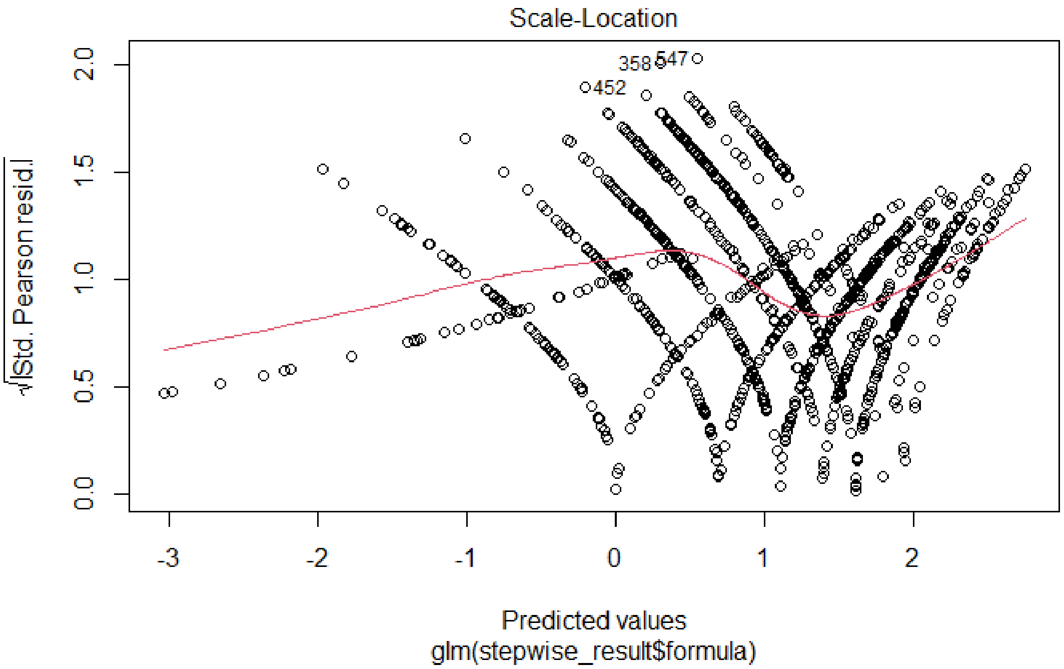
The QQ-plot validates that my choice of the Poisson distribution and link function is not a great fit. There is significant deviance along the upper and lower tails. This shows the theoretical quantiles against the actual quantiles, which are all approximately along a straight line.



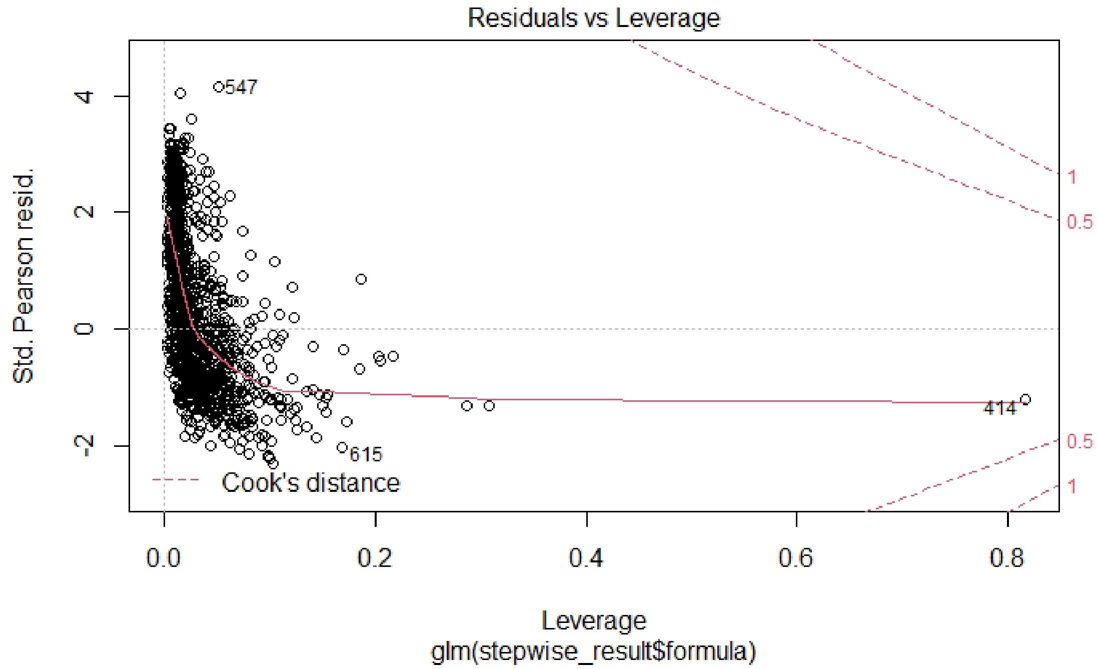
The residuals vs. fitted shows the raw residuals against the predicted values. This is different than the studentized residuals above, which are adjusted by the shape of the Poisson distribution. They are approximately centered at zero, although there is some decreasing trend (the residuals get smaller as the predicted values increase).



The scale-location plot below shows that there is no clear pattern when I adjust for the Poisson distribution with the studentized residuals.



The scale-location plot shows that no points are having too much of an impact on the model's results. There are no points in the upper right quadrant of the graph.



Task 8 – Fit a LASSO (5 points)

The lasso is a type of penalized regression which makes the model simpler by imposing a penalty term on the number and size of the parameters. This is a special case of the elastic net, which takes two parameters, called lambda and alpha. When Alpha = 0, we have Ridge Regression, which penalizes the sum of the square of the coefficients, and when alpha = 1, we have the Lasso, which penalizes the sum of the absolute value of the coefficients. This has the resulting effect of setting some variables' coefficients exactly equal to zero, which removes them from the model.

When I ran the LASSO using the same formula from the GLM in task 6, I found that these variables were removed:

```
year_sold
tot_bathrooms
neighborhood_gilbert
```

Task 9 – Create a bagged tree model (10 points)

The code was set up to create 8 bootstrapped samples of the training data. Eight samples of the training data were taken with replacement and then bagging was performed.

Bagging is where multiple samples of the same data are taken, different models are fit over each sample, and then the average of the model's predictions based on the test set is used as the final prediction. I updated the formula of the predictions to use the average of the eight trees. This improves performance because of the bias variance tradeoff – taking the average of several trees reduces the variance, because variance decreases as the sample size increased, and the bias decreases as each tree uses different variables and so when a single tree overfits, this has less of an impact on the model overall because this tree is only given partial credibility.

I fit three different models. I varied the min bucket, which controls the minimum number of observations in a split, cp, which is the complexity parameter, and max.depth, which is the max height of the tree. Then I compared the log likelihood of the Poisson model based on the test set.

minbucket	cp	maxdepth	log likelihood
5	0.01	5	295
10	**0.001**	**5**	**356**
10	0.01	7	343

The log likelihood is the probability of generating the response variable given the data. A higher value implies that the model is more accurate. This first model with min bucket = 5, cp = 0.01, and max depth = 5 is the best and so this is what I chose.

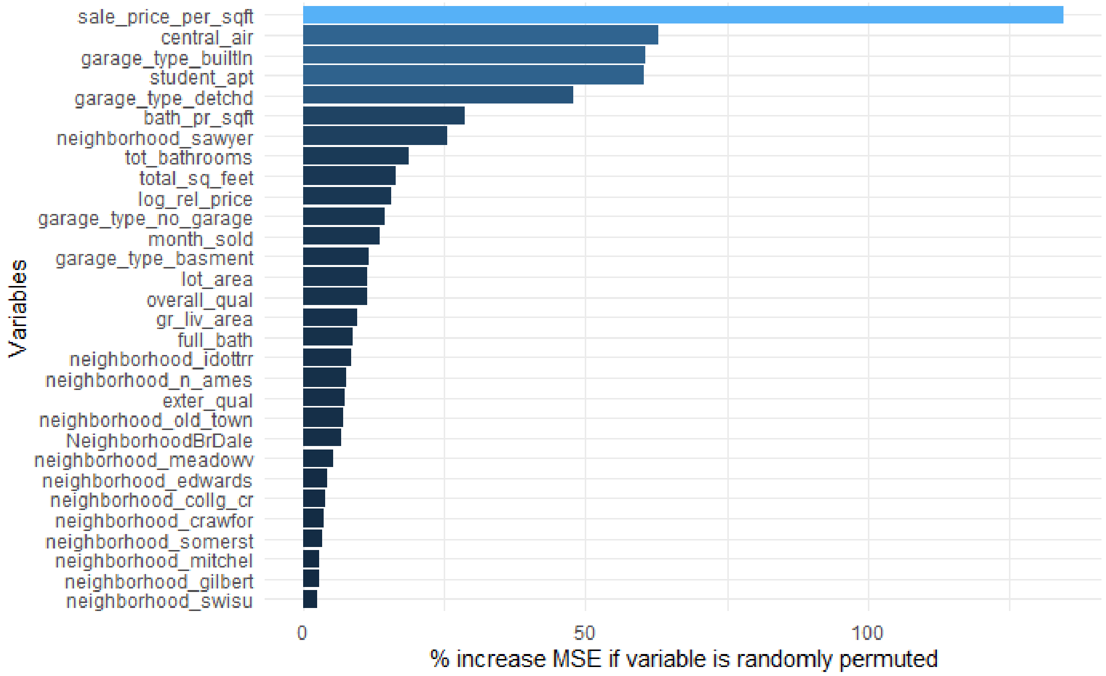
A decision tree works by recursively partitioning the training data based on variables which separate out apartments which have a lot of applicants from those which have relatively fewer applicants. The variables that are higher up in the tree as well as those which appear more often have greater predictive power, also known as variable importance. In my case, the variable that I engineered, sale_price_per_sq appeared as the top split in 7 out of 8 trees, and of the one tree where it was not the top split it was still the second-level split. This is the most important variable in the bagged tree model.

Not surprisingly, the log likelihood of a single tree was worse (lower) than the average of the eight trees. This was 128 for tree1 under the third model as compared to 356 for the bagged trees.

Task 10 – Fit a random random forest (5 points)

I fit a random forest on the training data using the num_units as the weight. The most important variable was the sale_price_per_sq, which is the same as the what we found in the bagged decision trees. This is not surprising given that both models use bagging, and both are based on decision trees. Some possible reasons why these importance rankings could be different are 1) The random forest below uses the % decrease in MSE to rank importance whereas my bagged trees were just looking at the frequency and height in each tree, which is a crude measure, 2) The trees in the random forest are using different hyperparameters, 3) My bagged tree model used only 8 trees whereas this random forest uses 400. This allows for deeper interactions and could be why we see the difference.

As compared to the LASSO, the variables are similar for year_sold and neighborhood_gilbert, which are both unimportance. Year_sold doesn't even make the top 30 in the random forest. Total bathrooms was higher in the random forest, though, as it is ranked 8 out of 39, which is fairly high. One possible reason is that the random forest is able to account for interactions whereas the LASSO isn't.



Task 11 – Compare model performance (10 points)

I computed the log likelihood based on the test set for each of the models. The random forest is the best, followed by the LASSO, bagged trees, and then the GLM. Some ways that I could improve these models if I were to do this analysis again are:

GLM:

- I could try out interaction terms, because my model used only the primary features. An interaction is when the impact that a variable has on the target changes depending on multiple variables at once.
- I could try out alternative link functions instead of just the log, as well as the quasi-poisson distribution. This is used when overdispersion is present, which I could check by comparing the mean and the variance of the number of applicants. If the mean is equal to the variance, then the Poisson is appropriate.

LASSO:

- I could test out more values for the parameters. I could try a ridge regression model be setting alpha = 0, or just tune alpha using cross validation.
- I could try out more values of lambda using cross-validation
- I could add interaction terms, because the lasso doesn't take these into account automatically
- I could also try adding non-linear transformations or smoothing splines for the continuous variables sale price and quality

Bagged Trees:

- I try out more parameters combinations, because I only tried three.
- I could use pruning based on the complexity parameter and choose the best CP based on cross-validation.

Random forest:

- Use more trees, as I only used 400
- I could try alternative values for mtry as well

For all models, I could try engineering additional features, because I only tried out three, which were all predictive.

Model	LogLikelihood
<chr>	<dbl>
GLM	238.5346
LASSO	429.3319
Bagged Trees	356.8725
Random Forest	447.5109

Task 12 – Executive summary (20 points)

Our client is a property management firm in NYC and enlisted our help to increase the number of people who apply for their apartments. This will increase their sales revenue by letting them sign more leases. This analysis used real data from 1,430 properties between the years of 2007 – 2011. For each of these buildings, which include 3,839 units, we had information on the number of people who applied for a lease, the size, number of bathrooms, size of the lot, overall quality both inside and outside, the neighborhood that it was located, type of garage, sale price, number of bathrooms, number of half bathrooms, and so forth. We used this information to construct a model which can predict how people will apply for any given unit to help our client understand what factors contribute to the demand for an apartment.

Here are the factors which impact how many people apply to sign a lease. This can be useful for your sales team to help them understand why some apartments may have many people apply and others have only few people apply.

Variable	% Impact on Number of Applicants	Business Interpretation
(Intercept)	-89%	Apartments which are in the North Ames Neighborhood with no garage are the most common apartment, and these have 89% fewer applicants than the average
overall_qual	-5%	Decreases as overall quality increases, perhaps because customers are shy from purchasing luxury apartments that are too expensive for their budget
total_sq_feet	0%	Size by itself does not make a difference
tot_bathrooms	-21%	Apartments which have more bathrooms are less desirable. This can be misleading because it doesn't take the size into
exter_qual	18%	Demand increases with higher quality on the outside of the apartment. This makes because people may look at photos of apartment building before visiting and submitting an application. Sometimes photos in the apartment itself are not available.
central_airno	-83%	Apartments without air conditioning are in lower demand. Customers like having comforts such as AC.
garage_type_basment	-48%	Apartments which don't have an attached garage are less demanded than apartments which have one that is attached to the building
garage_type_builtIn	-25%	Apartments which don't have an attached garage are less demanded than apartments which have one that is attached to the building
garage_type_detchd	-28%	Apartments which don't have an attached garage are less demanded than apartments which have one that is attached to the building
garage_type_no_garage	-31%	Customers would prefer to have an attached garage than to having none
NeighborhoodBrDale	-40%	Apartments in the Dale neighborhood have 40% fewer applicants than those same apartments if they had been in the N Ames Neighborhood
neighborhood_brk_side	-20%	Apartments in Brookside have 20% fewer applicants that those in N Ames
neighborhood_gilbert	10%	Apartments in Gilbert have 20% fewer applicants that those in N Ames
neighborhood_idottrr	-59%	Apartments in IDDOTTRR have 20% fewer applicants that those in N Ames
neighborhood_meadowv	-29%	Apartments in Meadow V have 20% fewer applicants that those in N Ames

neighborhood_n_ridge	-16%	Apartments in N Ridge have 20% fewer applicants that those in N Ames
neighborhood_n_ridge_hghts	-20%	Apartments in N Ridge Heights have 20% fewer applicants that those in N Ames
neighborhood_sawyer	-33%	Apartments in Sawyer have 20% fewer applicants that those in N Ames
neighborhood_swisu	32%	Apartments in SWISU have 20% fewer applicants that those in N Ames
log_rel_price	-63%	As the sale price increase relative to neighboring apartments, the demand decreases. This makes sense because customers will choose the apartment which has a lower price if the other characteristics, such as the size and number of bathrooms, are the same
bath_pr_sqft		Customers greatly prefer having a private bathroom. When adjusting the number of bathrooms by the size of the unit, more people applied to apartments which had a greater number of bathrooms relative to the size of the unit
student_apt	63%	Units which students apply for have 63% more applicants than those without students. We didn't have precise data as to which people where students but relied on the fact that most student apartments change hands during July and August.

We measured how important each variable is in predicting the number of applicants and ranked them from highest to lowest. This can help your property management team to prioritize the most relevant info when going after sales leads. The top 10 variables were

1. Sale price per square foot
2. Whether or not the apartment has central AC
3. If the apartment has a built-in garage
4. If the apartment is rented by a student
5. If there is a detached garage
6. The number of bathrooms per square foot
7. If the apartment is in the Sawyer neighborhood
8. The total number of bathrooms
9. The total square feet
10. The sale price relative to the sale price of other apartments in the same neighborhood

If the sale price per square is too high, we recommend using a lower price so that more applicants will apply for the unit. If the apartment incudes AC, add on an additional dollar amount to the price. We recommend purchasing new apartment buildings which have attached garages, and avoiding apartments which have building or detached garages, because these were less desireable. We recommend adding more bathrooms to apartments which have shared bathrooms. Even adding a half-bathroom to a large bedroom would help. Finally, we recommend considering the geography closely and pricing so that the price is competitive with other apartments in the area.

We began with the data and performed cleaning to make sure that it was appropriate for modeling. We used the info on the number of units in each apartment so that these results can be interpreted for each unit individually. Each unit had between 0 and 7 people apply for it.

We checked that each of the values made sense given your descriptions of what the fields represented in the Data Dictionary. We wanted to consider the sale price relative to the neighborhood, because there is usually significant variation in price depending on the geography. To do this, we divided the sale price by the average sale price in that neighborhood.

We gathered insights from the Marketing manager to expand on our analysis in three ways.

1) The size of the unit determines the price. We calculated the sale price per square foot for each apartment unit so that larger apartments with a lower price will stand out, as well as smaller apartments which have high price.
2) Many apartments have their leases end in July and August, as students are moving in at the beginning of the academic school year, and so we flagged these apartments as being student-rented.
3) People like having a private, rather than shared, bathroom, and so we looked the number of bathrooms relative to the size of the unit. Units which have a greater number of bathrooms per square foot turned out to have more people apply for them.

We did discover several errors with the data related to the type of garage. There were 15 records which we needed to remove because of a recording error. We can follow up with your data team to see if these records can be corrected.

We tried out several different types of model. The first, was a generalized linear model. This can help you to understand the relationships between each of the variables and the number of applicants. GLMs are a commonly used type of model that can be explained using simple arithmetic while also being powerful in certain use cases. We chose the type of GLM based on the description of the data, for counting the number of applicants, as well as making the result easy to interpret. We adjusted each record for the number of units that are in the apartment building using a statistical concept known as an offset. We then validated that the model was working by splitting the data into two groups and testing out the model on a “blind” group of apartments. This showed that our statistical assumptions will also apply in the real world, when you use the results on new apartments which are not represented in this data. We then used an automatic procedure to remove variables which were irrelevant to the problem known as stepwise selection. This resulted in a simpler model which was also powerful.

We wanted to find variables which were not needed, and so we used a model called a LASSO, which automatically removes variables which are not predictive. This showed us that the year sold, total number of bathrooms, and whether the apartment was in the Gilbert neighborhood are not important variables. Our client can use this info to reduce the costs of gathering data in the future.

The next type of model was based on decision trees. These have several advantages over the GLM, and in some cases can be more powerful, and so we wanted to give our client another option. Trees are able to detect more nuanced relationships within the data, such as interaction effects, which is when multiple variables work together to change the number applicants, as well as non-linearities, which are when the variables do not effect the number of applicants in a consistent way. To improve the performance, we used a technique known as bagging, which combined the results of eight separate models into a single model.

Lastly, we used a common machine learning algorithm known as a random forest. This is a tree-based model which requires extraordinarily little customization while still delivering high performance. This also let us determine the relative predictive power of each of the variables.

Depending on the goals of our client, we have a model which can help. If the goal is to be as accurate as possible, then we recommend the random forest. If the goal is to easily explain the results using a spreadsheet, we recommend the LASSO.

To expand upon this analysis, we have several recommended ways of further improving the model results. This analysis clearly shows that the demand for an apartment can be predicted based on the characteristics of the apartment. Each of our models were statistically validated using a blind hold-out set, and so we are confident that the resulting models can be deployed in real-time to help your management team to better serve their customers.

There are several limitations to this analysis. When we were identifying if an apartment was student-rented, we only considered whether it was rented in July or August. If we had data on the demographics of the customer, we could use this info precisely. These models are dependent on the geography and can only apply for apartments around the N Ames area. For these results to be used elsewhere, new data for that region would need to be collected. Similarly, this data is several years old, as the latest records only go through 2011. The market for apartments has undoubtedly changed since then and so we recommend getting new data. We only tested a limited number of parameters when fitting these models. We can improve the performance by testing out a broader range of parameters if this is necessary.

When fitting the GLM, we noticed that the diagnostics of the model were not ideal. This indicates that the statistical assumptions do not fully match the data. We recommend trying out variations of the GLM to see if better results can be achieved.