Hospital Length of Stay (LOS) Case Stud

Andrew Goh (3457439)

QUESTION ONE (30 MARKS)

Based on Table 2 from the medical review paper, select a paper (article) which models Length of Stay (LOS) and use it to answer the following questions:
(remember to include a reference to the paper and other sources as appropriate):
Link to article:
https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-10-9#Sec14

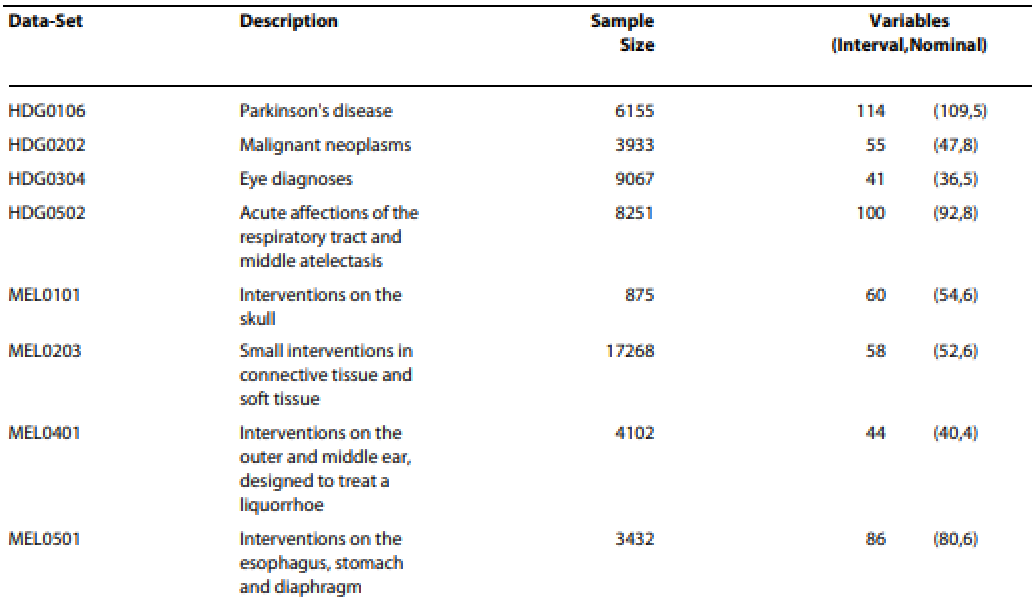
(a) What is the problem being modelled? What question(s) are they asking?

The problem that the article is trying to model would be an efficient and fair way of allocating resources to hospitals based on their performance. The article states that the aim of diagnosis-related group (DRG) systems is to classify hospital patients into groups that consume such resources which is measured by their length of stay (LOS). LOS also has a good relationship with the total cost and availability. The groups are described by variables such as the patient's information and diagnoses. The resource allocation uses conjunctive rules where all rules must be satisfied for a resource to be allocated. The conjunctive rules can be interpreted as a tree structure where the rules can be created by regression tree methods.

A bootstrap-based method called bumping is used to build diverse and more accurate trees. However, the most statistically accurate models often have to be readjusted according to medical knowledge which decreases the predictive accuracy. Additionally, those models can be too complex for the DRG systems to apply. The question that the article addresses is if bumping allows more diverse and accurate trees to be built compared to the Classification and Regression Trees (CART) algorithm without increasing the complexity. The article also asks if this model selection approach is also applicable to other medical decision and prognosis tasks.

(b) How do they initially describe/present the data being used for the model?

The data described by the article comes from the Austrian DRG system 2006. It consists of 8 datasets from 4 *Medizinische Einzelleistung* (MEL) and 4 *Hautdiagnosegruppe* (HDG) groups. The MEL and HDG groups are one of the 979 patient groups resulting from a three-step classification procedure which will be explained in part c. The datasets consist of patient information such as their diagnosis, number of procedures, sex, age and LOS. The details of the 4 MEL and HDG groups can be seen on the next page which is taken from table 1 of the article.

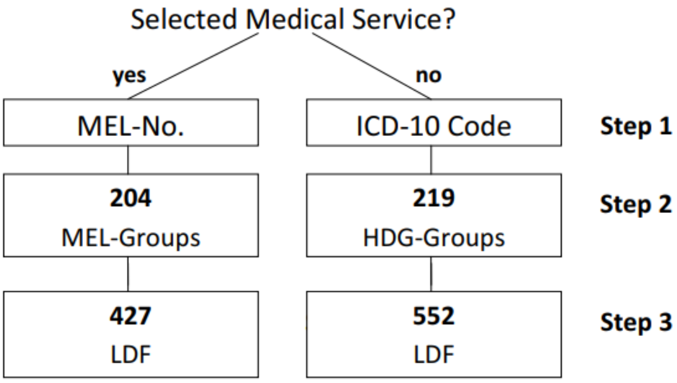


Data-Set	Description	Sample Size	Variables (Interval,Nominal)	
HDG0106	Parkinson's disease	6155	114	(109,5)
HDG0202	Malignant neoplasms	3933	55	(47,8)
HDG0304	Eye diagnoses	9067	41	(36,5)
HDG0502	Acute affections of the respiratory tract and middle atelectasis	8251	100	(92,8)
MEL0101	Interventions on the skull	875	60	(54,6)
MEL0203	Small interventions in connective tissue and soft tissue	17268	58	(52,6)
MEL0401	Interventions on the outer and middle ear, designed to treat a liquorrhoe	4102	44	(40,4)
MEL0501	Interventions on the esophagus, stomach and diaphragm	3432	86	(80,6)

The article only indicates that the 8 datasets are chosen by evaluating their possibility of producing diverse and accurate models for predefined tree complexities.

(c) How do they analyse the data prior to modelling? For example, they may consider how to select a subset of the explanatory variables as input to the model using an information gain measure, or correlation, clustering, PCA, etc.

The article explains how the Austrian DRG system classifies patients into patient groups called *Leistungsorientierte Diagnosefallgruppen* (LDF). It uses a three-step classification model below.



Step 1 checks if a patient consumes a predefined individual medical service, it is placed in the MEL-No where a procedure-oriented LDF. Otherwise, the patient is placed in an ICD-10 Code which leads to an HDG group in Step 2 where it places the patient in an LDF group related to the patient's main diagnosis. In Step 3, the patients corresponding to either a MEL or HDG group are divided into specific LDF groups to find groups with more homogeneous LOS. This not only subsets the patient data into homogeneous groups with similar LOS and diagnosis but also, it makes the variables to be used in rules which can be simple enough to be used in decision tree models without requiring knowledge about the algorithm itself. Additionally, by having simple rules, would allow further analysis by medical experts where they can verify if the model is realistic for hospital management and budgeting.

(d) What data transformations (e.g., scaling, normalisation) are applied (if any) to the dataset prior to modelling? This might also include constructing new (transformed) variables using PCA, or new calculated values (e.g., taking log).

The article did not mention any data transformations to the datasets. It was mentioned that the 8 datasets used for model building produce more diverse and accurate tree models and they are from MEL and HDG groups. The article focuses on the tree models and compares CART tree models to model search by bootstrapping and alternative tree model methods.

(e) What is the architecture of the model (if appropriate)?

The article uses different tree-building models.

Firstly, the CART algorithm is used to build a tree model. It splits the data into two groups based on a splitting rule and stops when the splitting rule cannot improve the homogeneity of the nodes significantly. The number of terminal nodes represents the number of patient groups in the model and is interpreted as the required number of rules for classifying patients. The article references the three steps in building a CART model from Chapter 2 in "Classification and Regression Trees" by Breiman L, and Friedman J and comments that the trees constructed in that fashion tend to be too big and have only a few observations in the terminal nodes. Therefore, the model's internal node is pruned back iteratively which leads to the smallest loss in accuracy and only one internal node remains.

The CART model is compared to the best Boostrapped Tree model which uses model search by Bootstrap where it combines and averages multiple models to reduce prediction error. Bumping is used in this case where it uses only single trees instead of an ensemble of trees and the single trees are all built on different bootstrap samples. The bumping procedure will be discussed in part g, but the resulting model sizes are limited to a set size and only models with the same number of nodes are compared in the article's analysis.

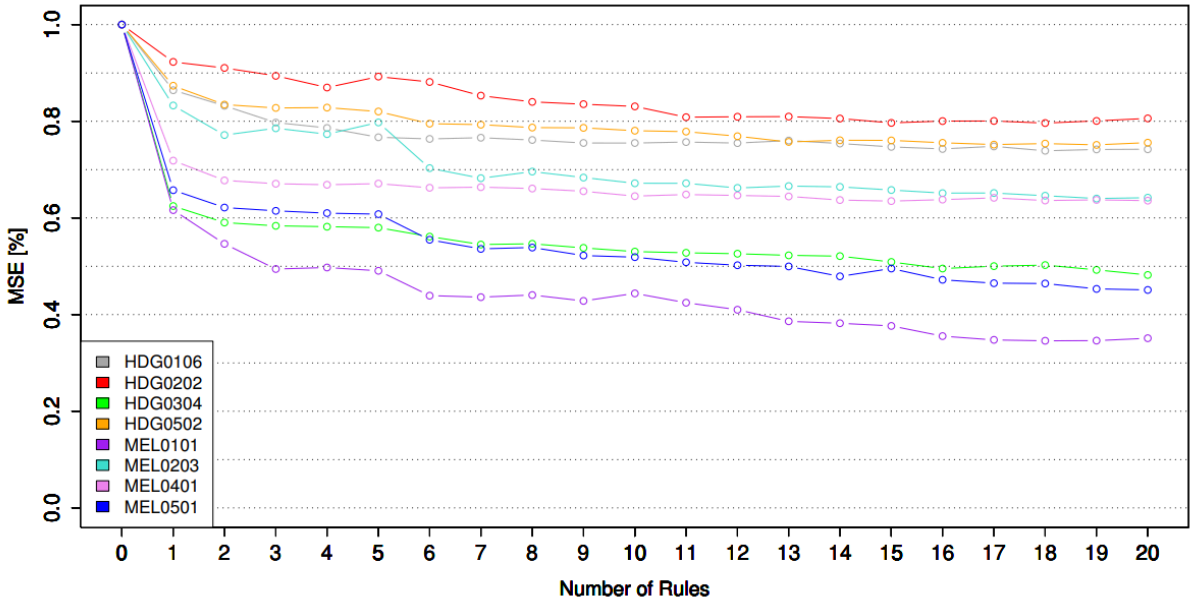
(f) How do they determine the model parameters (if appropriate)? This might be that they just set them arbitrarily, or that they did some trial runs to find “appropriate” settings, or something else?

The article considered the requirements of the CART algorithm to determine the model parameters but mainly focused on the model structure. It states that the CART algorithm is greedy where it builds trees in a forward stepwise search, making its results only locally optimal due to splits at each node being chosen to maximise homogeneity at the next step only. This is addressed with the bumping method where despite it identifying different trees in a greedy manner, some of the models may be closer to a global or local maximum.

The article discussed improvements on the CART model to make it more globally optimal by calculating the effects of the choice of the attribute in deeper layers of the tree. However, this is computationally intractable for larger datasets such as the datasets used in the model where the search is an exhaustive search. The article discusses controversies of using look-ahead search on tree algorithms where oversearching the hypothesis space can result in an overfit of the training data, giving less accurate and bigger tree models. It came to a conclusion that it was better to have an approach of constructing a set of models first and leaving the selection of the tree model to the users of the software.

Other optimisation methods were performed to improve the CART model tree structure after it is built in a greedy manner. Evolutionary algorithms were used by having a fitness function that takes the misclassification rate and tree size into account. This would prune the tree to be more computationally efficient. Bayesian CART algorithms were used to stochastically optimise the CART tree by using Monte Carlo methods to give an approximation of the probability distribution of the space of possible trees. Each tree is modified one at a time by having "grow", "prune" and "change" steps which change the split of an internal node. Simulated annealing and tabu search were also used to optimise the model. The article briefly explains how they work such as how simulated annealing prevents the model from being "stuck" in a local minimum too early and finds the local optimal solution. Tabu search allows the model to search for solutions beyond the local optimum while still making the best possible move at each direction.

For the bootstrap models, a constraint is added to the models in order for them to have similar complexity which is given by the number of internal nodes. The article only compares models with the same number of nodes and limits the maximum number of internal nodes to 16, resulting in 17 patient groups and a tree depth of 5 which corresponds to 5 rules to classify patients. They performed a test to see the reduction of Mean Squared Error by the best-boostrapped tree for different tree sizes and from the results of Figure 3 in the article (pasted below), more complex models only had small improvements to the predictive error and was too complex to be used in the DRG system.



(g) How is the problem dataset used for training/testing? Here we are assuming that they are doing a supervised learning problem. For example, do they do repeated training/test randomised splits, cross-validation, bootstrapping or some other method?
For the CART tree, the article did not mention any specific training and test data used for it. It can be assumed that the original 8 datasets were split into training and test sets for the model.

For the bootstrap model search, the bootstrap samples are formed by random sampling with replacement from the original training data and each bootstrap sample has the same size as the original training data set. This is used in the bumping procedure where a set of bootstrap samples are drawn from the training set and models are fit to each bootstrap sample. For each tree complexity, the best tree models are selected based on their average prediction error on the original training set.

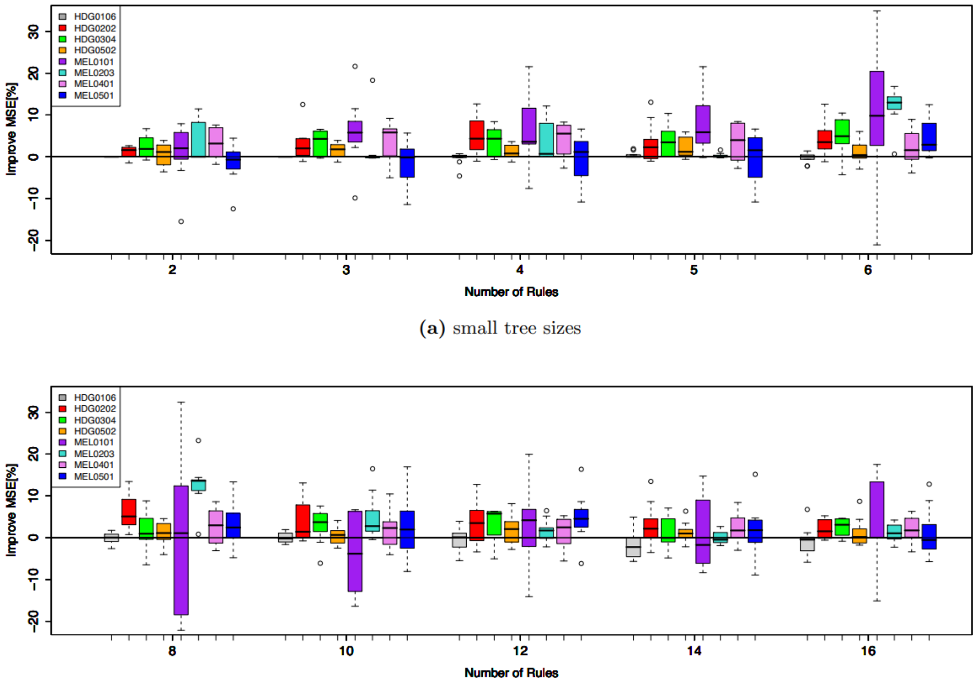
A 10-fold cross-validation was also used to assess the accuracy of the best-bootstrapped tree. The data is partitioned into complementary subsets called folds. The model is built on 9 folds and the last fold is used as a test set. The analysis is repeated 10 times where each fold is used as a test set once. This allows the predictive accuracy to be calculated from the average performance of the 10 models on the test sets. The article mentioned that there was no restrictions on the minimum number of observations for each terminal node. The argument would be that by comparing to a CART tree that has a minimum number of observations in a node where it would stop splitting to avoid overfitting. The bootstrap tree splits at the same node where it can lead to significantly different predictive accuracy which would lead to more diverse trees to provide more possible ideal models.

The article did not mention any scaling of the data but since the 8 datasets were picked specifically because on their possibility to produce diverse and accurate trees, scaling the data might not be necessary.

(h) How do they measure the performance of the model on training/test data? i.e. what error measurements did they use? For example, they might calculate the root mean squared error (RMSE), or R2 , or use a ROC curve, etc.
For the CART tree, the cost complexity criterion was used to assess the accuracy-complexity tradeoff to determine the right-sized tree. The formula is below.

$$R_{\alpha}(T) = R(T) + \alpha \mid \tilde{T} \mid$$

Where R(T) is the Mean Squared Error (MSE) and T is the number of terminal nodes. alpha would be a non-negative constant which regulates the additional cost for complex tree models.
To address the main goal of the article, the CART algorithm tree model is compared to the best-bootstrapped model by bumping by evaluating their ability in predicting the LOS of hospital patients. The relative predictive accuracy of the models is compared using the improvement of their MSE. This is used for the comparison in Figure 4 and Table 2 displayed below, which were taken from the article where it shows the performance results of the CART tree and the best bootstrap tree.



(a) small tree sizes

(b) large tree sizes

Figure 4 Comparison of the best bootstrap based tree with the standard CART tree.

Table 2: Relative average improvement.

Tree Size	HDG0106	HDG0202	HDG0304	HDG0502	MEL0101	MEL0203	MEL0401	MEL0501	Average
2	0.00	1.12	2.55	0.71	1.20	3.74	3.34	-1.52	**1.39**
3	0.00	2.78	3.33	1.65	5.96	1.88	3.92	-1.97	**2.19**
4	-0.36	5.57	3.52	1.23	5.77	3.30	4.28	-1.05	**2.78**
5	0.42	3.18	3.85	2.30	7.43	0.26	3.81	-0.84	**2.55**
6	-0.24	4.38	5.47	1.13	9.65	12.03	2.33	4.41	**4.90**
8	-0.11	6.05	1.75	1.15	1.06	12.91	2.67	3.63	**3.64**
10	-0.06	3.99	3.16	0.69	-2.93	5.09	1.94	2.83	**1.84**
12	-0.42	4.14	3.24	1.75	2.89	1.61	1.24	4.95	**2.43**
14	-1.87	3.35	1.82	1.20	-0.36	0.00	2.15	2.17	**1.06**
16	-0.76	2.11	2.52	1.27	1.38	1.18	1.89	0.65	**1.28**

Relative average improvement of the best bootstrapped tree compared to the standard CART tree using 10-fold cross validation.

(i) How successful was the model? In other words, did the authors compare the performance with other modelling approaches, and did the model tell them something interesting? i.e. Did the model help answer the questions that were being addressed?

The article found that the use of bumping can construct more diverse and accurate models for DRG systems that predict patients' LOS compared to the standard CART algorithm. Based on the results of comparing the MSE improvement of the best-bootstrapped tree compared to the CART tree in Figure 4 and Table 2 displayed above, the CART tree performs better than the bootstrap tree for the HDG0106 dataset and on models with smaller tree sizes specifically 2 - 4 internal nodes on the MEL0501 dataset and 10 - 14 internal nodes for the MEL0101 dataset. The article could not address why the bootstrap model performed worse on those datasets and tree sizes. However, the bootstrap model performs better on the other tree sizes and datasets where it averaged an improvement of 1.06 - 4.9% for the different tree sizes.

The article also found that alternative tree models can be found despite having a very low tree complexity whereas simple models can have different tree models with a + or - 1% difference in accuracy. For more complex models, accuracy can be improved by over 1% with having a larger number of different trees. However, for the models to be used by the DRG system, the article is only interested in the choices of split variables in the model where it can relate to their medical meaning in order to be used practically.

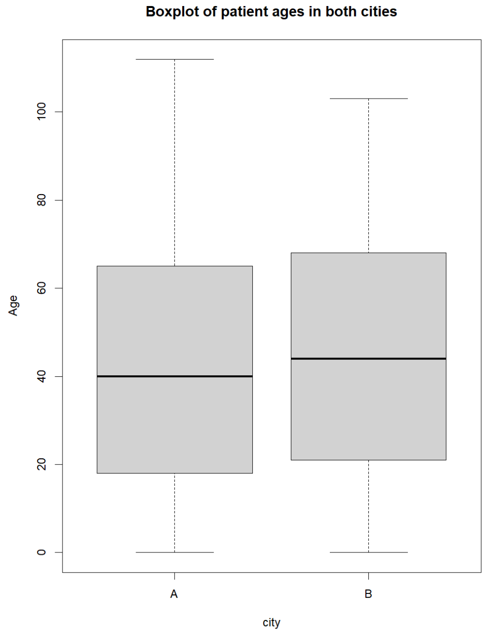
Despite using bumping to find the best bootstrap tree model, the statistically optimal tree model was not selected because of medical expert opinion. The evaluation of resulting tree models is evaluated by medical experts in order for the model to meet economic and medical considerations. The article does address this by constructing diverse models to allow a wide range of candidate models to be considered. The article does answer the question they address by showing that bumping does produce more diverse and accurate tree models and it shows that the model selection approach of bumping can be useful in any classification or regression problems in medical decision and prognosis tasks but it must require domain specific knowledge to guide the selection of a model that is both medically meaningful and statistically accurate.

The method of modelling in the article was considered successful due to three factors. Firstly, the data used for the models contains a lot of important information about each patient such as their diagnoses, secondary diagnoses, specific procedures, number of diagnoses and procedures and their basic details such as sex and age which would make the model more robust due to it being better in generalising LOS based on the given information.
Secondly, the models were trained and tested on 8 different datasets containing the same explanatories. Each dataset belongs to a specific diagnosis group and by testing the same model across all datasets, it can accurately reflect how the model can generalise LOS.
Lastly, the article had help from medical experts who could verify if the chosen model would be realistic. As mentioned, often the statistical optimal model was not selected but by creating multiple tree models, this allowed a wider range of models to be considered and be possibly used.

QUESTION TWO (70 MARKS) HOSPITAL ADMISSION – LENGTH OF STAY (LOS) MODELLING

1. Describe and visualise the datasets for both cities, focussing on the length of stay (LOS) and the relationship to the explanatories that are known at time of admission. Include a discussion that compares the cities in terms of health delivery, the population cohort, bed count over time, etc.

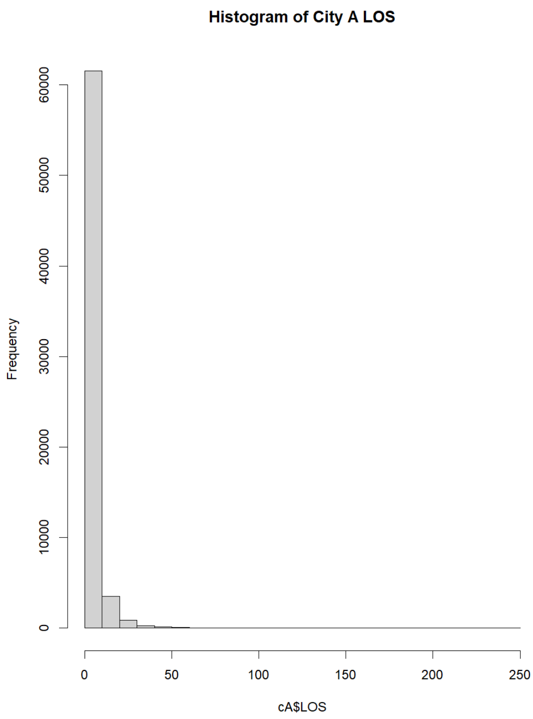
At first glance, the data from city A would be taken from a larger city compared to city B with city A having 66533 observations compared to city B with 33512 observations. The boxplot below shows the distribution of the patient ages in each city where the range of patient ages are relatively similar despite city A having a higher maximum value for a patient age.

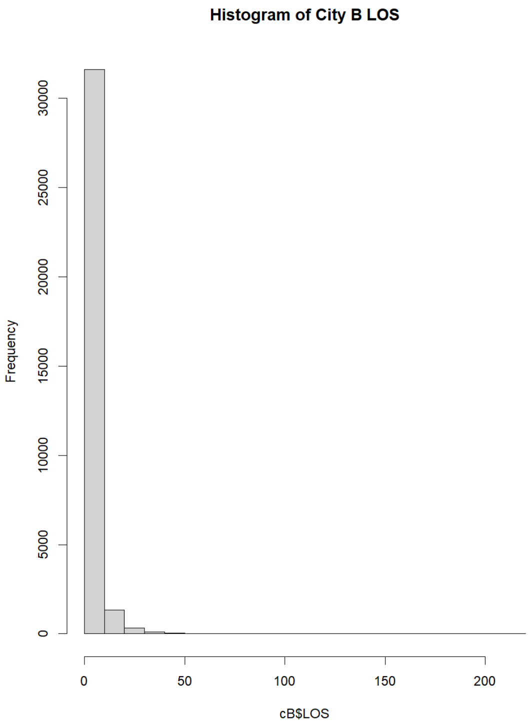


We are more interested in the LOS variable. Below is the summary of the LOS variable of both cities.

```
summary(cA$LOS)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 1.000   1.000   2.000   4.089   5.000 249.000 
summary(cB$LOS)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 1.000   1.000   2.000   3.699   4.000 216.000 
```

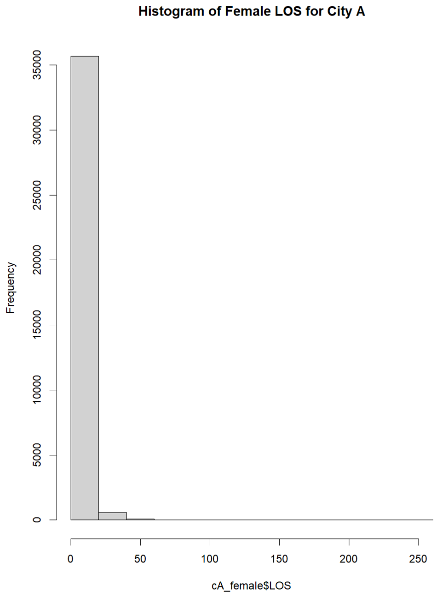
Both cities have a min LOS of 1 where a patient stays for 1 day which is very possible. Both cities have a median of 2 but have different mean values which can be explained by city A having almost double the observations compared to city B. City A has a higher maximum value than city B with a patient staying for 249 days compared to 216. It is important to consider the distribution of the data where the following histograms would show.

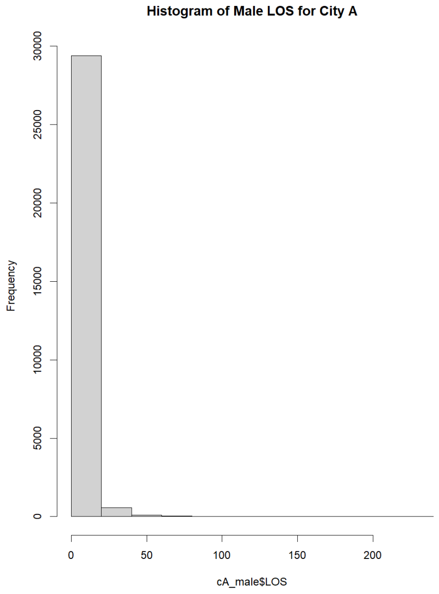




We can see that the LOS variable for both cities are very skewed due to most patients staying for at most 1 day at the hospital. Having very skewed data would make it hard to model as it would make it biased towards the class that has the majority of the data, in this case having 1 day for LOS. A possibility would be to normalise the LOS variable and exclude some observations that has an LOS of 1.

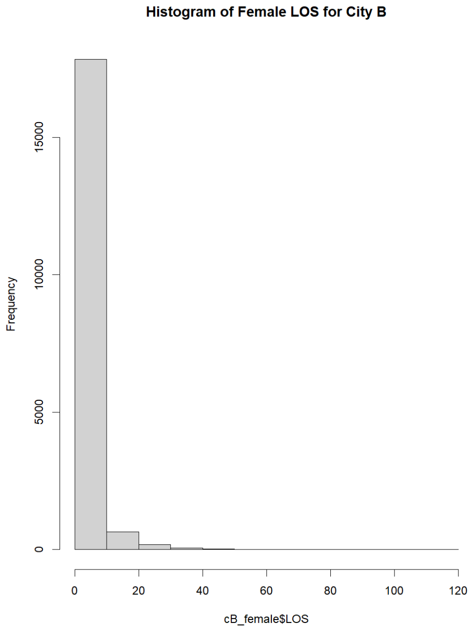
It would be important to visualise the relationship between the explanatories and LOS.
Firstly, the relationship between LOS and GENDER is explored.
The histograms below display the Female and Male LOS of city A.

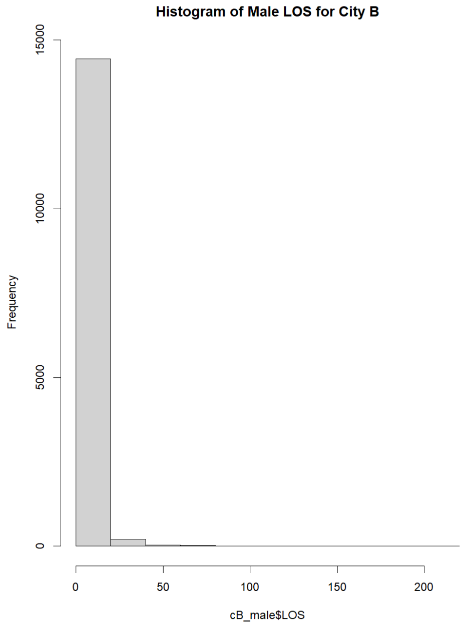




We can see that the LOS is equally skewed for both genders, despite females having a higher frequency than males.

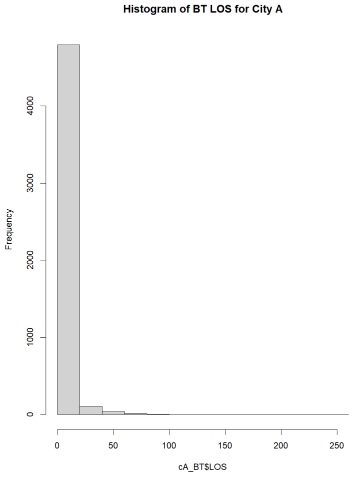
The histograms below display the Female and Male LOS of city B.

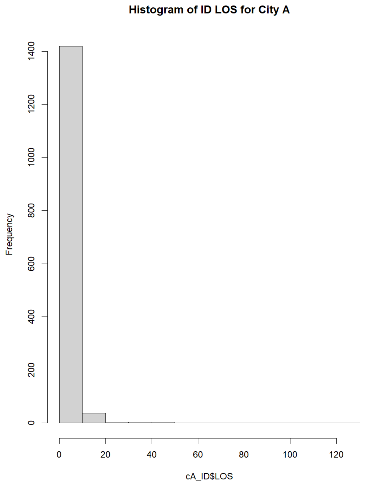

Histogram of Female LOS for City B

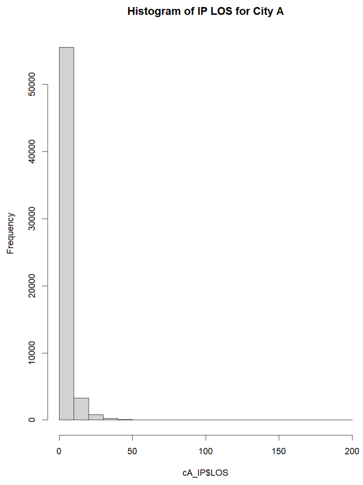

Histogram of Male LOS for City B

The LOS variable is skewed for both genders. The frequency is lesser than city A due to city B having less observations. There is a difference in frequency between females and males like in city A. For both cities A and B, the number of females and males is relatively balanced with city A having 36392 female and 30141 male observations and city B having 18789 female and 14723 male observations.

For EVENT_TYPE, there are 3 values IP, BT and ID. It would be worth exploring if the LOS would be equally skewed for the 3 event types.
The histograms below are the 3 event types for city A.

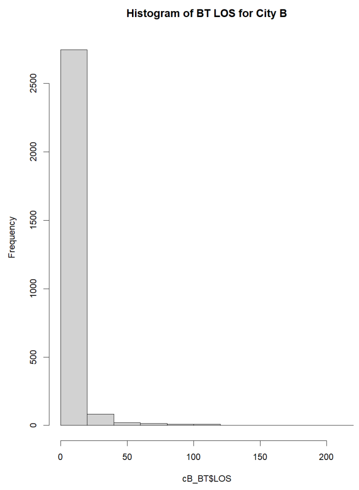

Histogram of BT LOS for City A

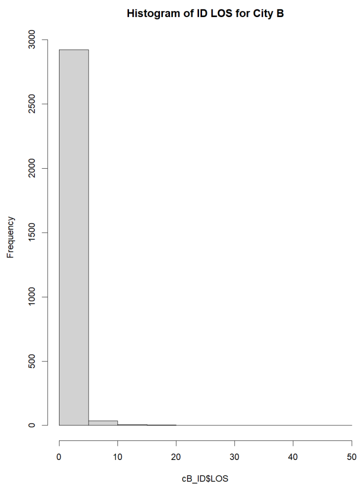

Histogram of ID LOS for City A

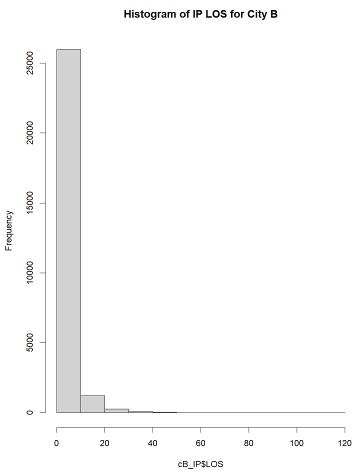

Histogram of IP LOS for City A

Similar to the relationship between GENDER and LOS, each of the event types are skewed due to the distribution of LOS. However, the IP event appears to be slightly less skewed than the other 2 events.

For city B, the histograms look similar to city A with highly skewed distributions.



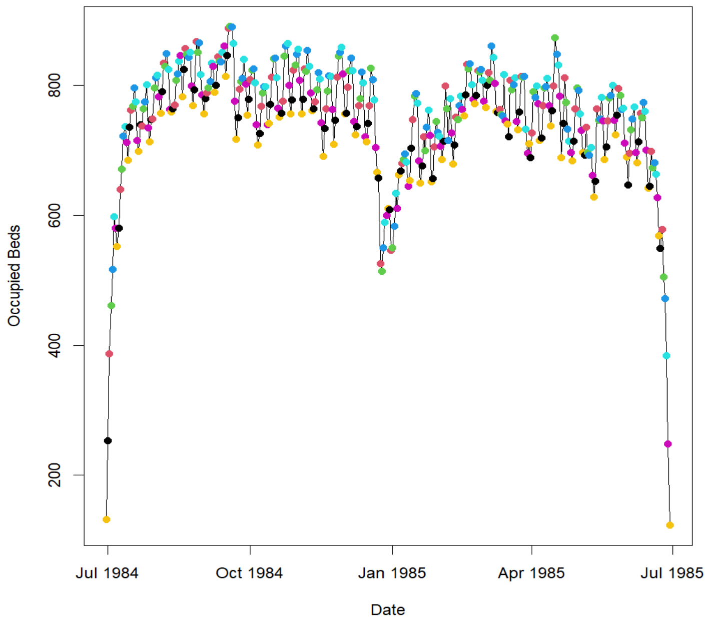




However, the number of observations per event type is imbalanced as city A has 60088 observations of event IP compared to event BT with 4973. City B has 27654 IP event observations compared to 2885 BT event observations. This would add more difficulty to modelling LOS by making the model more biased towards IP event observations. Therefore the event type should not be a major explanatory for modelling LOS.

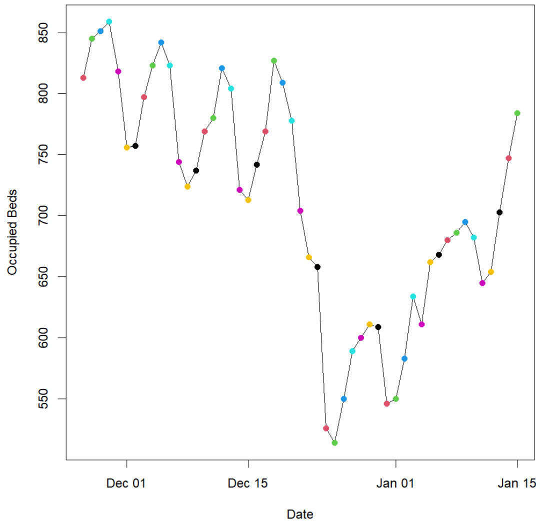
Based on the histograms of the categories of GENDER and EVENT_TYPE, the other explanatories such as Dep06 where it has 10 categories can be assumed to be skewed as well. The HLTHSPEC and diag01 has too many unique values to compare the relationship of each category's relationship to LOS. The EVSTDATE and EVENDATE were included in the datasets to find any potential patterns and confirmation of LOS values. They can be used to display the bed count over time.

The plot below displays the bed count over time for city A over the EVSTDATE and EVENDATE variables which dates span over 1 year.

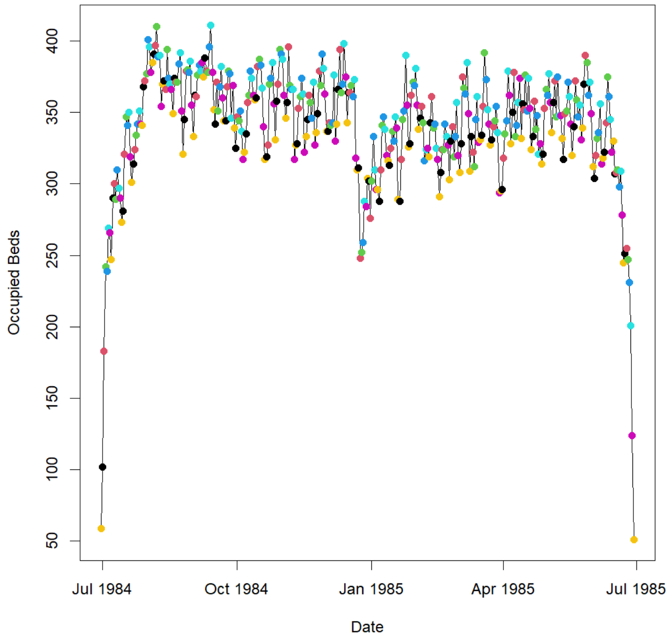


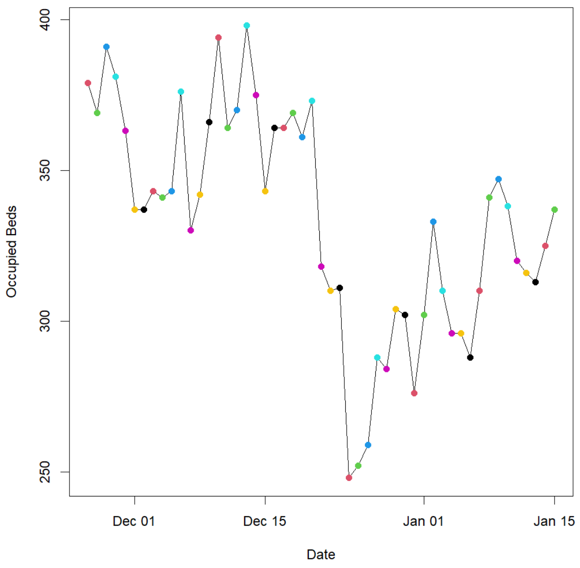
There is an increase in occupied beds after July 1984 and a decrease towards the following July. There is also a decrease in bed counts during the New Year period. It seems that the bed counts are decreasing within each week which a closer look is needed.

The plot below displays the bed count over the new year period.



There is a clear decrease in beds occupied every weekend, especially during the Christmas and New Year period. The same plots can be reproduced for city B which are displayed below.

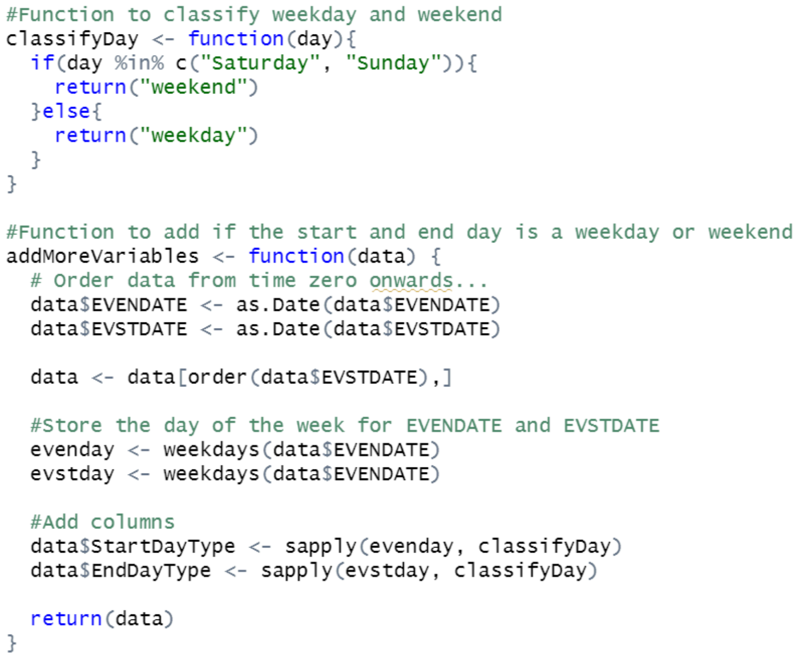




Although the trend of bed count for the Christmas and New Year period of city B is slightly different to city A, there is a clear trend of the bedcount dropping during the weekends for both cities.

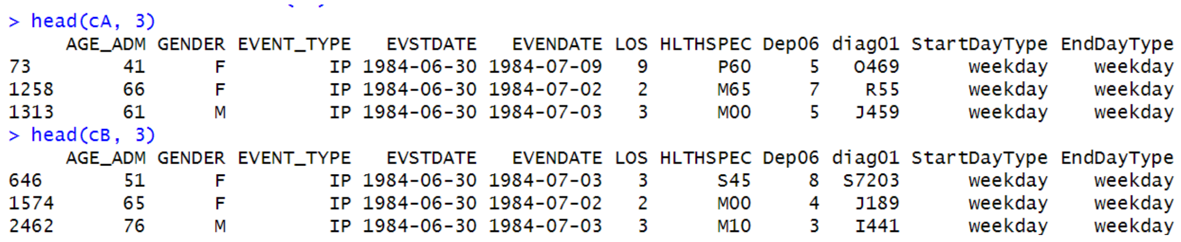
Based on the behaviour of bed count (or other observations) what other explanatory variable(s) could be created? Create any additional explanatory variables and save to cityA and cityB data prior to step 2 in the assignment. What (if any) transformations are suggested by the data? Why?

From the previous section, it would be useful to include the day of the week due to the trend of bedcounts dropping over the weekend. However, due to the EVSTDATE and EVENDATE only indicating the day that the patient started and ended their stay at the hospital, two new explanations will be added to give indication of whether the patient enters and discharges on a weekend. This is done with the code below.



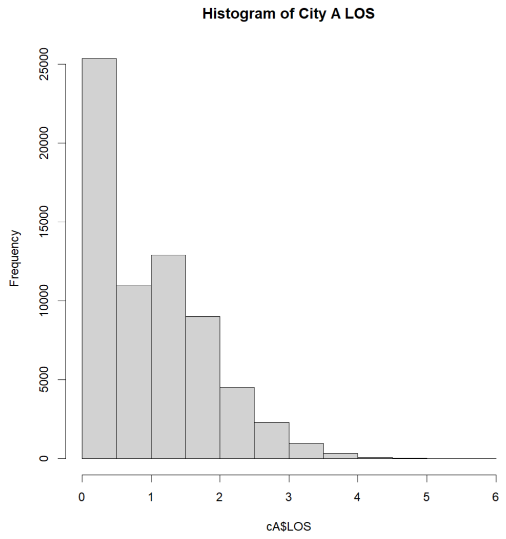
```
#Function to classify weekday and weekend
classifyDay <- function(day){
  if(day %in% c("Saturday", "Sunday")){
    return("weekend")
  }else{
    return("weekday")
  }
}

#Function to add if the start and end day is a weekday or weekend
addMoreVariables <- function(data) {
  # Order data from time zero onwards...
  data$EVENDATE <- as.Date(data$EVENDATE)
  data$EVSTDATE <- as.Date(data$EVSTDATE)

  data <- data[order(data$EVSTDATE),]

  #Store the day of the week for EVENDATE and EVSTDATE
  evenday <- weekdays(data$EVENDATE)
  evstday <- weekdays(data$EVENDATE)

  #Add columns
  data$StartDayType <- sapply(evenday, classifyDay)
  data$EndDayType <- sapply(evstday, classifyDay)

  return(data)
}
```

The resulting datasets now have 2 additional explanatories that indicate what type of day the patient began and ended their stay.

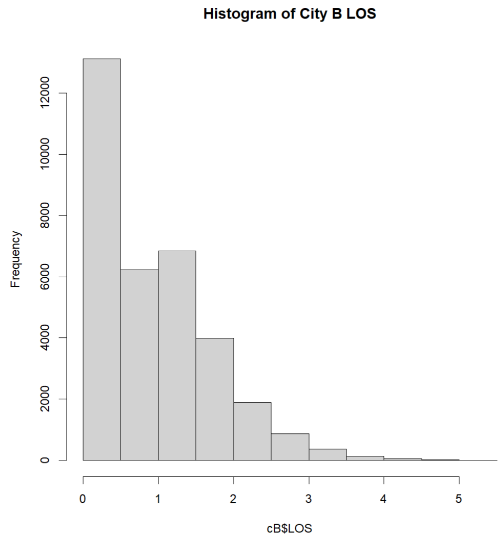


```
> head(cA, 3)
     AGE_ADM GENDER EVENT_TYPE   EVSTDATE   EVENDATE LOS HLTHSPEC Dep06 diag01 StartDayType EndDayType
73        41      F         IP 1984-06-30 1984-07-09   9      P60     5   O469      weekday    weekday
1258      66      F         IP 1984-06-30 1984-07-02   2      M65     7    R55      weekday    weekday
1313      61      M         IP 1984-06-30 1984-07-03   3      M00     5   J459      weekday    weekday
> head(cB, 3)
     AGE_ADM GENDER EVENT_TYPE   EVSTDATE   EVENDATE LOS HLTHSPEC Dep06 diag01 StartDayType EndDayType
646       51      F         IP 1984-06-30 1984-07-03   3      S45     8  S7203      weekday    weekday
1574      65      F         IP 1984-06-30 1984-07-02   2      M00     4   J189      weekday    weekday
2462      76      M         IP 1984-06-30 1984-07-03   3      M10     3   I441      weekday    weekday
```

To address how skewed the LOS variable is for both city datasets, a log transform is needed in order to reduce how bias the model would be towards the most common value which would be a LOS of 1 day. The following histograms reflect the new LOS distribution for both city A and B.



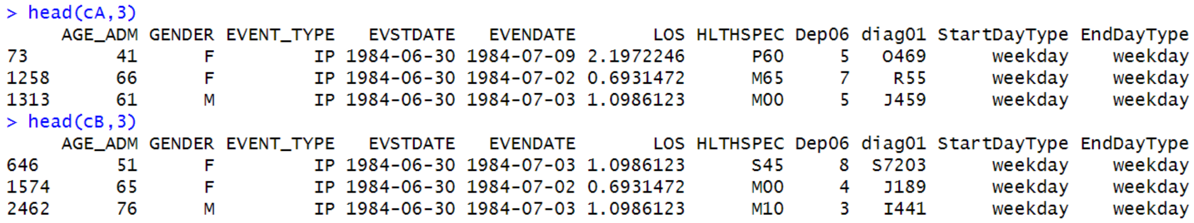
Histogram of City A LOS



Histogram of City B LOS

The LOS distribution is still skewed but much less skewed compared to the original values which would give a less bias model. Log transformation cannot be used for the EVENT_TYPE explanatory due to it being categorical in nature. Unlike LOS which has many observations within many different values of LOS, there are only 3 event types where it can be left unchanged.

After the adjustments, both datasets now have an additional 2 explanatories and a log-transformed LOS variable.



```
> head(cA,3)
     AGE_ADM GENDER EVENT_TYPE   EVSTDATE   EVENDATE       LOS HLTHSPEC Dep06 diag01 StartDayType EndDayType
73        41      F         IP 1984-06-30 1984-07-09 2.1972246      P60     5   O469      weekday    weekday
1258      66      F         IP 1984-06-30 1984-07-02 0.6931472      M65     7    R55      weekday    weekday
1313      61      M         IP 1984-06-30 1984-07-03 1.0986123      M00     5   J459      weekday    weekday
> head(cB,3)
     AGE_ADM GENDER EVENT_TYPE   EVSTDATE   EVENDATE       LOS HLTHSPEC Dep06 diag01 StartDayType EndDayType
646       51      F         IP 1984-06-30 1984-07-03 1.0986123      S45     8  S7203      weekday    weekday
1574      65      F         IP 1984-06-30 1984-07-02 0.6931472      M00     4   J189      weekday    weekday
2462      76      M         IP 1984-06-30 1984-07-03 1.0986123      M10     3   I441      weekday    weekday
```

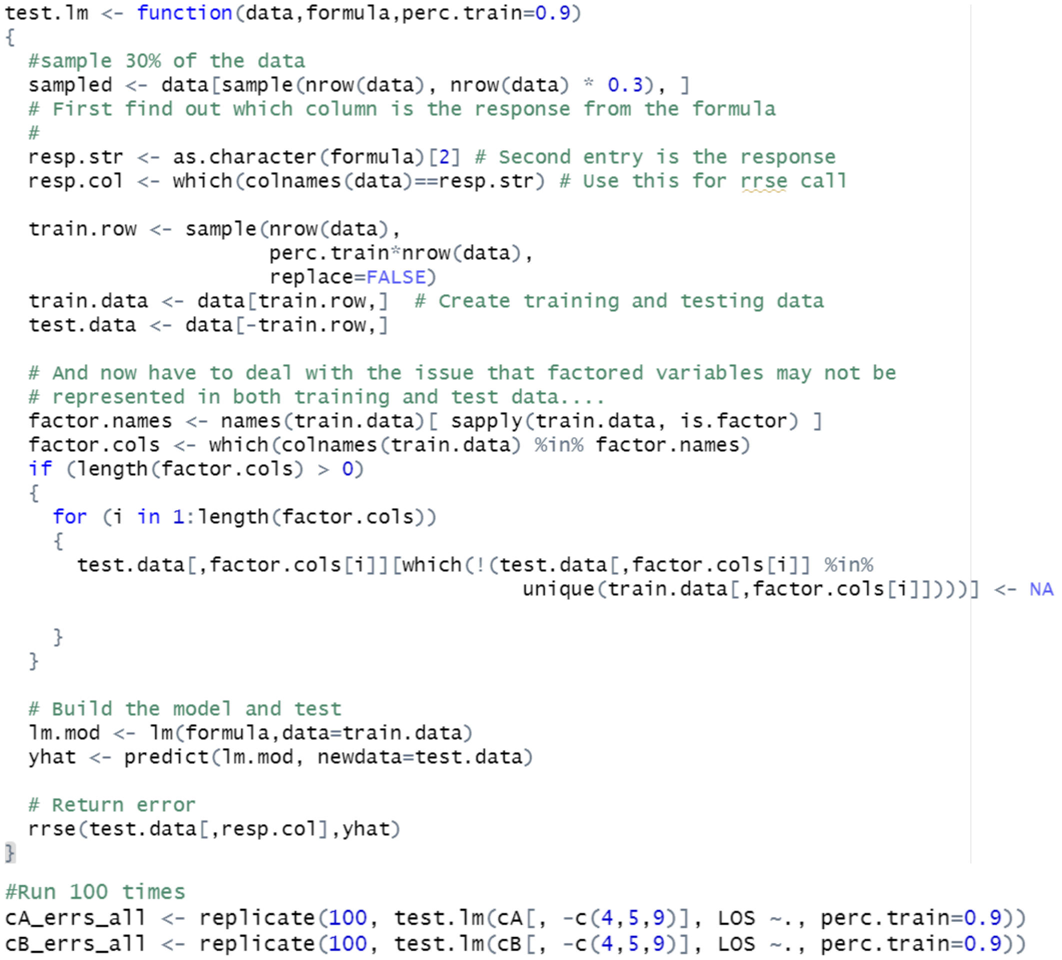
An additional explanatory could be added to reflect the month as there was a large decrease in bedcount for the month of July. However, the large decrease only happened in 1 out of 12 total months compared to the trend of bedcount decreasing every weekend and is not added.

2. Build separate linear models (with additional explanatories and transformations) to predict LOS for cityA and cityB using all of the data – DO NOT USE the initial diagnosis class (diag01).

Discuss why LOS is difficult to model. How might you go about making the problem easier? Explore and discuss several options for improving LOS modelling and present some initial approaches. This can all be done just using linear regression, logistic regression, random forests, etc., depending on how you modify the data. To assess model accuracy, use a 90% training-10% test split over 100 replicates using RRSE as the error measurement (if appropriate).

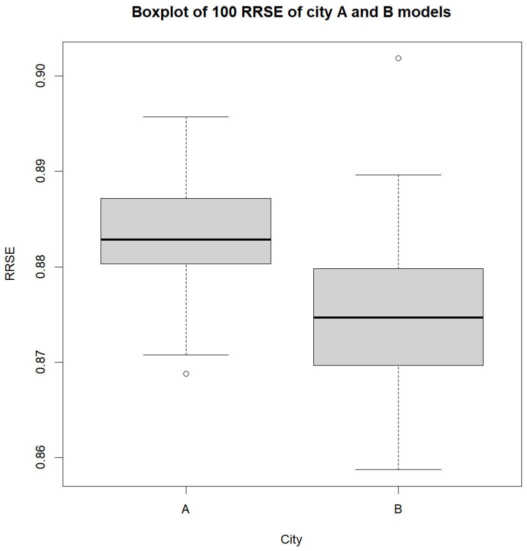
Firstly, separate linear models with the adjusted datasets were built.

In addition to the diag01 variable, the EVSTDATE and EVENDATE variables are removed as they are date-type values. The code below builds and tests the model with 100 replicates and a 90% training, 10% test split for both cities. The datasets are relatively large and running 100 replicates would take a long time. I modified the given test.lm() function to sample 30% of the data each time it is run.

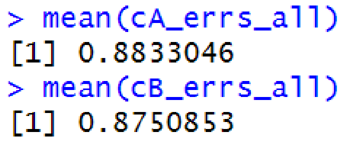


```
test.lm <- function(data,formula,perc.train=0.9)
{
  #sample 30% of the data
  sampled <- data[sample(nrow(data), nrow(data) * 0.3), ]
  # First find out which column is the response from the formula
  #
  resp.str <- as.character(formula)[2] # Second entry is the response
  resp.col <- which(colnames(data)==resp.str) # Use this for rrse call

  train.row <- sample(nrow(data),
                      perc.train*nrow(data),
                      replace=FALSE)
  train.data <- data[train.row,]  # Create training and testing data
  test.data <- data[-train.row,]

  # And now have to deal with the issue that factored variables may not be
  # represented in both training and test data....
  factor.names <- names(train.data)[ sapply(train.data, is.factor) ]
  factor.cols <- which(colnames(train.data) %in% factor.names)
  if (length(factor.cols) > 0)
  {
    for (i in 1:length(factor.cols))
    {
      test.data[,factor.cols[i]][which(!(test.data[,factor.cols[i]] %in%
                                     unique(train.data[,factor.cols[i]])))] <- NA

    }
  }

  # Build the model and test
  lm.mod <- lm(formula,data=train.data)
  yhat <- predict(lm.mod, newdata=test.data)

  # Return error
  rrse(test.data[,resp.col],yhat)
}

#Run 100 times
cA_errs_all <- replicate(100, test.lm(cA[, -c(4,5,9)], LOS ~., perc.train=0.9))
cB_errs_all <- replicate(100, test.lm(cB[, -c(4,5,9)], LOS ~., perc.train=0.9))
```

The boxplot below displays the distribution of RRSE values for the models of city A and B over 100 replications.



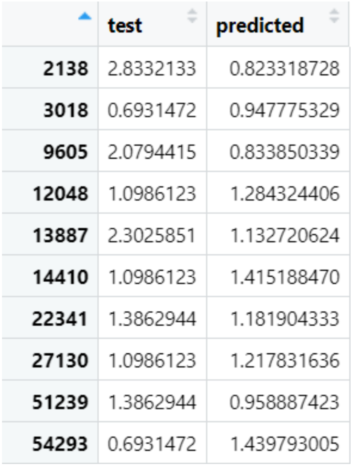
The interquartile range and median for city B is lower than A which could be caused by city B having fewer observations than city A.



```
> mean(cA_errs_all)
[1] 0.8833046
> mean(cB_errs_all)
[1] 0.8750853
```

The mean RRSE error of the 100 replications for city A and city B is displayed above. Having a RRSE value of 0.8 shows that the model has a poor performance as the closer the RRSE value is to zero, the better the model performance. Additionally, values around 1 suggest that the model is not doing any better than predicting the mean of the response.

By taking a modified version of test.lm() and making it return a data frame of the test LOS values and the predicted LOS values. I ran the function once using city A and displayed the first few rows.



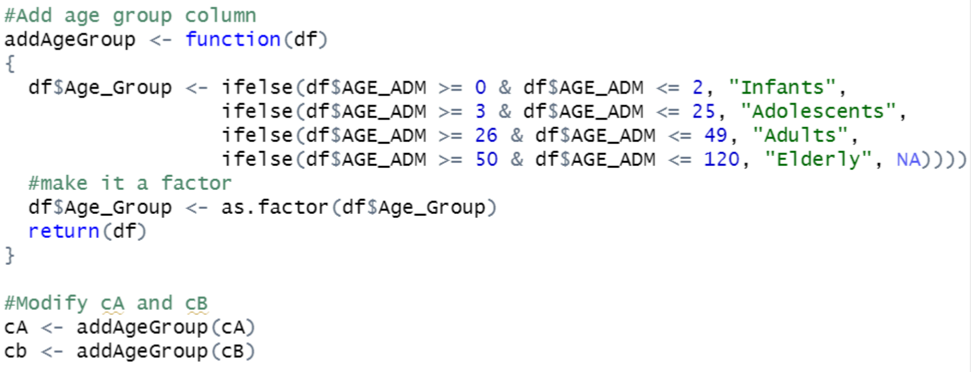
	test	predicted
2138	2.8332133	0.823318728
3018	0.6931472	0.947775329
9605	2.0794415	0.833850339
12048	1.0986123	1.284324406
13887	2.3025851	1.132720624
14410	1.0986123	1.415188470
22341	1.3862944	1.181904333
27130	1.0986123	1.217831636
51239	1.3862944	0.958887423
54293	0.6931472	1.439793005

We can see that the model does not have good predicted values which the RRSE values reflect. From this, LOS is difficult to model mainly due to the different types of patient variables being the age range or HLTHSPEC and the imbalance in the data with the LOS and EVENT_TYPE variables. Having many variables for the dataset would result in complex interactions between the variables. The linear model fails in this case possibly due to to LOS having a non-linear relationship with the rest of the explanatories.

The problem of modelling LOS could be made easier if we focus on a specific aspect of patients which reduces the amount of factors for the model to account for. One way would be to split the patients into age groups in this case 4 categories:

- Infants: 0-2
- Adolescents: 3-25
- Adults: 26 - 49
- Elderly: 50 - 120

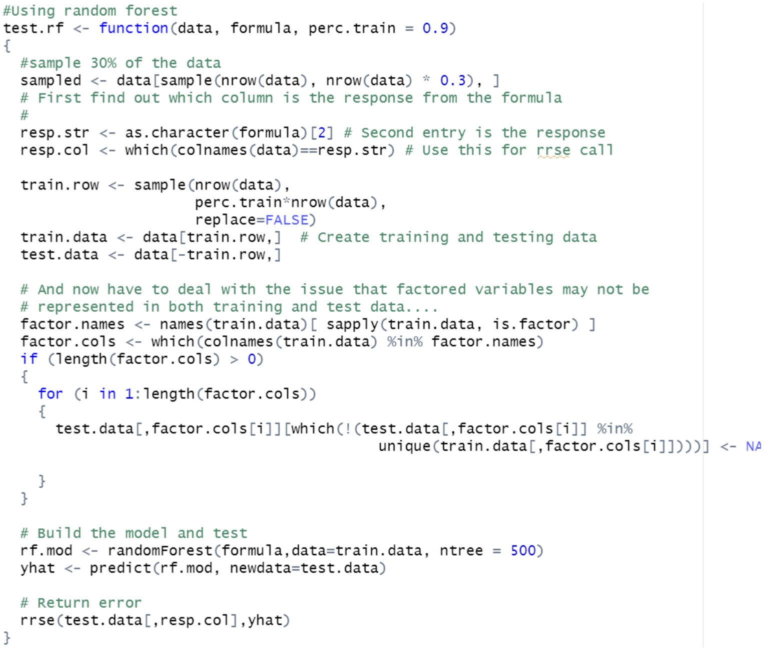
The imbalance in the categories is necessary to reflect realistic age groups. The code below adds the age groups to city A and city B.



```
#Add age group column
addAgeGroup <- function(df)
{
  df$Age_Group <- ifelse(df$AGE_ADM >= 0 & df$AGE_ADM <= 2, "Infants",
                  ifelse(df$AGE_ADM >= 3 & df$AGE_ADM <= 25, "Adolescents",
                  ifelse(df$AGE_ADM >= 26 & df$AGE_ADM <= 49, "Adults",
                  ifelse(df$AGE_ADM >= 50 & df$AGE_ADM <= 120, "Elderly", NA))))
  #make it a factor
  df$Age_Group <- as.factor(df$Age_Group)
  return(df)
}

#Modify cA and cB
cA <- addAgeGroup(cA)
cb <- addAgeGroup(cB)
```

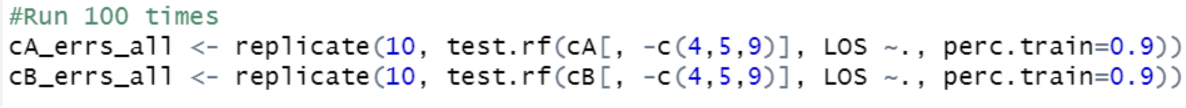
Since the linear model failed, it is worth using other models such as logistic regression and random forests.
A logistic regression has only two possible outputs where it is not suitable for predicting continuous variables such as LOS. Therefore a random forest is worth exploring.

An initial approach to improve LOS modelling would be to modify the test.lm() function to make it use a random forest instead of a linear regression. The function uses the R library randomForest and has a forest size of 500.



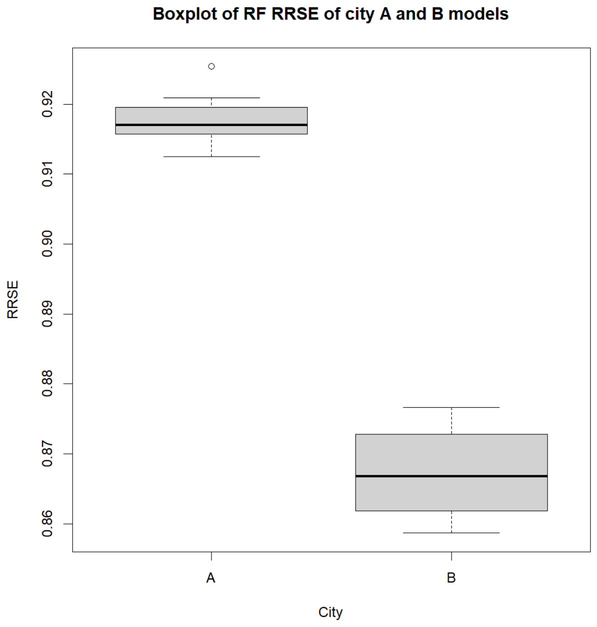
```
#Using random forest
test.rf <- function(data, formula, perc.train = 0.9)
{
  #sample 30% of the data
  sampled <- data[sample(nrow(data), nrow(data) * 0.3), ]
  # First find out which column is the response from the formula
  #
  resp.str <- as.character(formula)[2] # Second entry is the response
  resp.col <- which(colnames(data)==resp.str) # Use this for rrse call

  train.row <- sample(nrow(data),
                      perc.train*nrow(data),
                      replace=FALSE)
  train.data <- data[train.row,]  # Create training and testing data
  test.data <- data[-train.row,]

  # And now have to deal with the issue that factored variables may not be
  # represented in both training and test data....
  factor.names <- names(train.data)[ sapply(train.data, is.factor) ]
  factor.cols <- which(colnames(train.data) %in% factor.names)
  if (length(factor.cols) > 0)
  {
    for (i in 1:length(factor.cols))
    {
      test.data[,factor.cols[i]][which(!(test.data[,factor.cols[i]] %in%
                                         unique(train.data[,factor.cols[i]])))] <- NA

    }
  }

  # Build the model and test
  rf.mod <- randomForest(formula,data=train.data, ntree = 500)
  yhat <- predict(rf.mod, newdata=test.data)

  # Return error
  rrse(test.data[,resp.col],yhat)
}
```

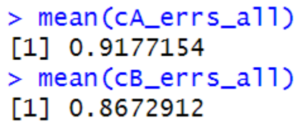
Due to the computation limits of my laptop, I replicated the function 10 times instead of 100 times in order for R to keep running.



```
#Run 100 times
cA_errs_all <- replicate(10, test.rf(cA[, -c(4,5,9)], LOS ~., perc.train=0.9))
cB_errs_all <- replicate(10, test.rf(cB[, -c(4,5,9)], LOS ~., perc.train=0.9))
```

The following boxplot displays the distribution of the RRSE of the city A model and city B model.



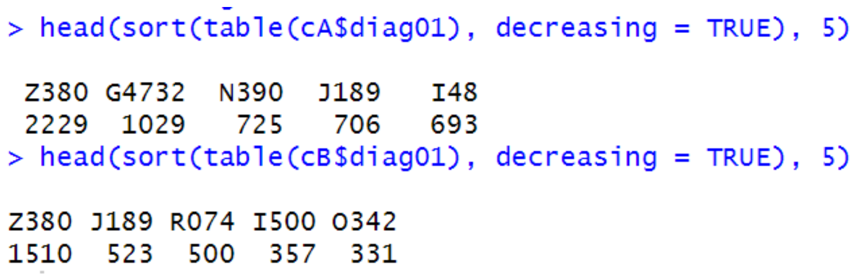


```
> mean(cA_errs_all)
[1] 0.9177154
> mean(cB_errs_all)
[1] 0.8672912
```

The random forest produces a better RRSE value for city B but has a higher value for city A. This could be caused by city A having more observations than city B which may contain more noisy relationships and have bigger class imbalances in the dataset. Another possibility would be that the model for city A was overfitted due to not controlling the depth of the tree. Additionally, the sample size for city B may serve as a more representative sample of the patients which would result in a better model performance. Moreover, reducing the number of replications would have a major effect on the RRSE values of the Random Forest. Unfortunately, by replicating the random forest 100 times, my Rstudio crashed and I had to resort to 10 runs. This would mean that the comparison between the models would not be reliable. However, if the random Forest model for city B could have a lower RRSE than the linear model, it is likely that doing 100 replications would result in a lower RRSE.

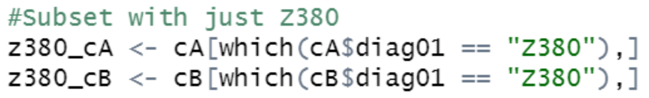
3. The diag01 variable defines the initial diagnosis for each patient on admission. Select a diagnosis that has a reasonable number of occurrences for both cities (i.e., > 100), and build a model to predict LOS just for patients with this diagnosis. To assess the quality of the model use a 90% training-10% test split for 100 replicates and measure error using RRSE (so the result gives an measure effectively against the mean – see lab notes). Discuss the results in relation to the type of diagnosis you have chosen, it's distribution of LOS, patient characteristics, etc. Can you model any signal for this diagnosis and LOS?

The code below produces the most common diag01 diagnoses for both cities.



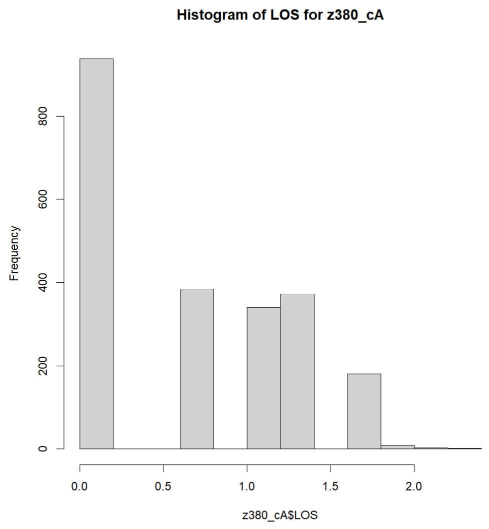
```
> head(sort(table(cA$diag01), decreasing = TRUE), 5)

 Z380 G4732  N390  J189   I48 
 2229  1029   725   706   693 
> head(sort(table(cB$diag01), decreasing = TRUE), 5)

Z380 J189 R074 I500 O342 
1510  523  500  357  331 
```

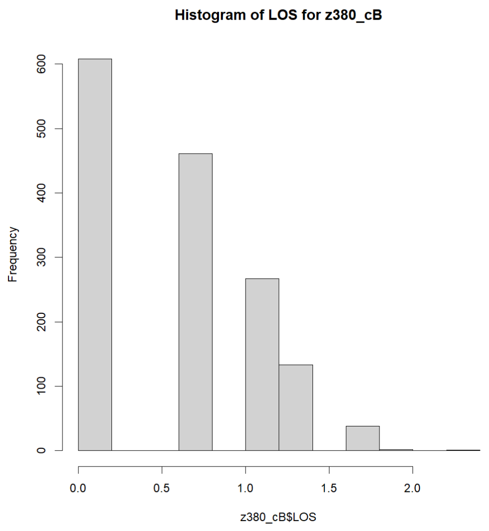
The diagnosis Z380 is the most common diagnosis for both cities. Therefore it would be useful to build a model to predict LOS for this diagnosis. I would repeat the same model-building steps as the previous question with the subset data for city A and city B containing only the Z380 diagnosis.



```
#Subset with just Z380
z380_cA <- cA[which(cA$diag01 == "Z380"),]
z380_cB <- cB[which(cB$diag01 == "Z380"),]
```

The following histograms show the LOS distribution within the z380 diagnosis for both cities.

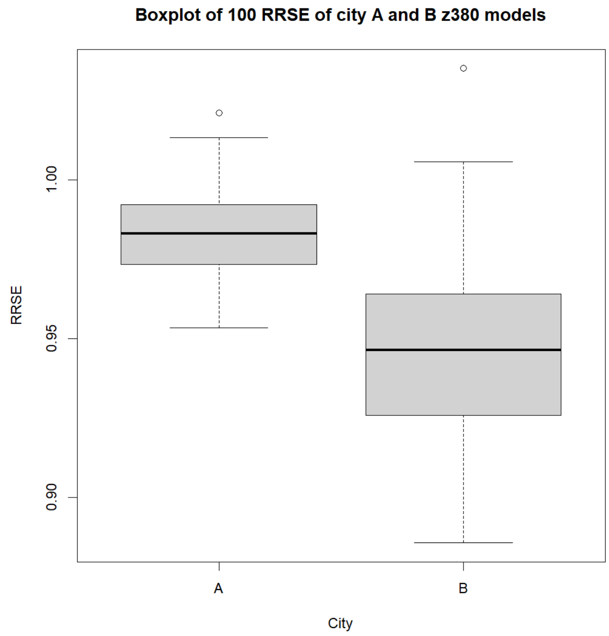


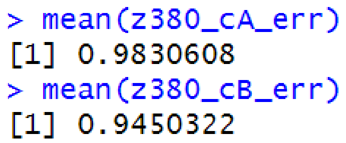


LOS distribution is less skewed and we can see that there are groupings of LOS values. However, there could be many categories and LOS is a continuous variable which it can be changed therefore it should not be made into a factor.

With the LOS variable less skewed, the linear model from test.lm() is run again with the same code previous. The Age_Group and AGE_ADM variables are excluded due to the Z380 diagnosis only involved the infant age group.

The boxplots display the RRSE ranges of the 100 runs per model

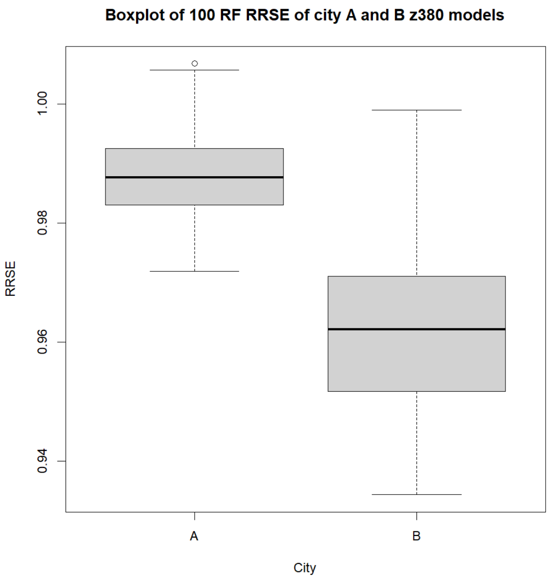




```
> mean(z380_cA_err)
[1] 0.9830608
> mean(z380_cB_err)
[1] 0.9450322
```

Despite narrowing down the datasets by focusing on a specific diagnosis, the models perform worse than the linear model using all the data with a higher RRSE value. Perhaps, important relationships between other age groups and variables were excluded in this model which affected its performance. Additionally, the LOS distribution is skewed and seems to be catagorised which may have affected the model's accuracy.

I repeated the model using random forests where the boxplots display the RRSE values over 100 runs. The dataset is small enough that my laptop does not crash while building the model.



The random forest in this case performed worse than the linear models which is very unusual. This means that either overfitting of the data occurred or that the LOS in the subsetted dataset has a linear relationship where a linear model is more appropriate.

Relating back to what the Z380 diagnosis type where it is given to newly born infants, the distribution of LOS would be understandable as there may be set times when an infant is able to be discharged from the hospital, resulting in the categorised effect of the LOS distribution. The HLTHSPEC values would be limited to fewer possibilities in this case 2 which may have reduced the relationships between LOS and may have impacted the model.

From the model results, no signal can be observed and cannot be modelled.

4. Discuss why predicting LOS is a difficult task and reflect on the way bed counts (i.e., admissions) are currently handled and therefore why this pattern might be observed. Discuss some approaches to break the problem down into more tractable aspects for managing admissions and predicting LOS. What does this suggest about building a single model for predicting LOS over the entire country?

From the attempts to model LOS, it is a difficult variable to predict due to it having many factors that can affect a patient's LOS. There are many different diagnosis categories for the dataset for both cities and a wide range of patient types with different ages and different diagnoses where a one-size-fits-all model for LOS is impossible due to the need to account for all the different factors.

Regarding the observed trend of bedcount decreasing during the weekends and notable holidays, this could be explained by the hospital having either reduced services on weekends and holidays or staffing constraints where bed counts decrease to compensate. Additionally, the trend can be a result of the hospital's elective procedures where the elective medical procedures could be scheduled in the weekdays which leads to a higher occupancy rate in weekdays and decreases in the weekends. It could also be a result of patient preferences where patients might delay treatments to the next week. All these possibilities add extra factors for the model to consider, thus adding to the difficulty of predicting LOS.

As discussed beforehand, breaking down the LOS predicting problem into more tractable aspects would be more realistic such as modelling LOS for a specific diagnosis or age group. This would decrease the amount of possibilities of patient types which also makes the problem easier as the amount of relationships would be significantly reduced. From the article discussed in question one, making multiple LOS models and testing them over multiple evaluated datasets, each specific to one diagnosis group while having medical expert opinion results in a better performing model that can meet realistic expectations. However, as seen from the previous models built in question 3, narrowing down the data could remove important relationships that result in a more accurate model for LOS which complicates how the LOS predicting problem should be broken down.

From this, it is possible to build a single LOS model for a local perspective where each LOS model is specific to either a diagnosis group within a local area. This is due to breaking the problem down into tractable aspects where the model is able to account for. Therefore, building a single model for predicting LOS over the entire country is impossible as not only would the model need to consider the individual factors respective to each local area but also, the model would need to process a large amount of data where it would be computationally expensive and inefficient compared to having LOS models for respective local areas.