MA679 Final Project - PT Data Analysis

Team 1

5/5/2020

Abstract

Physical therapy has always been an intriguing topic on the data end, because a lot of features in this field would be hard to quantify. As statisticians, we always wonder the relationship between different variables. With the data from our client, we can now better understand what it would take to make a successful therapy and thus carry out further analysis.

Introduction

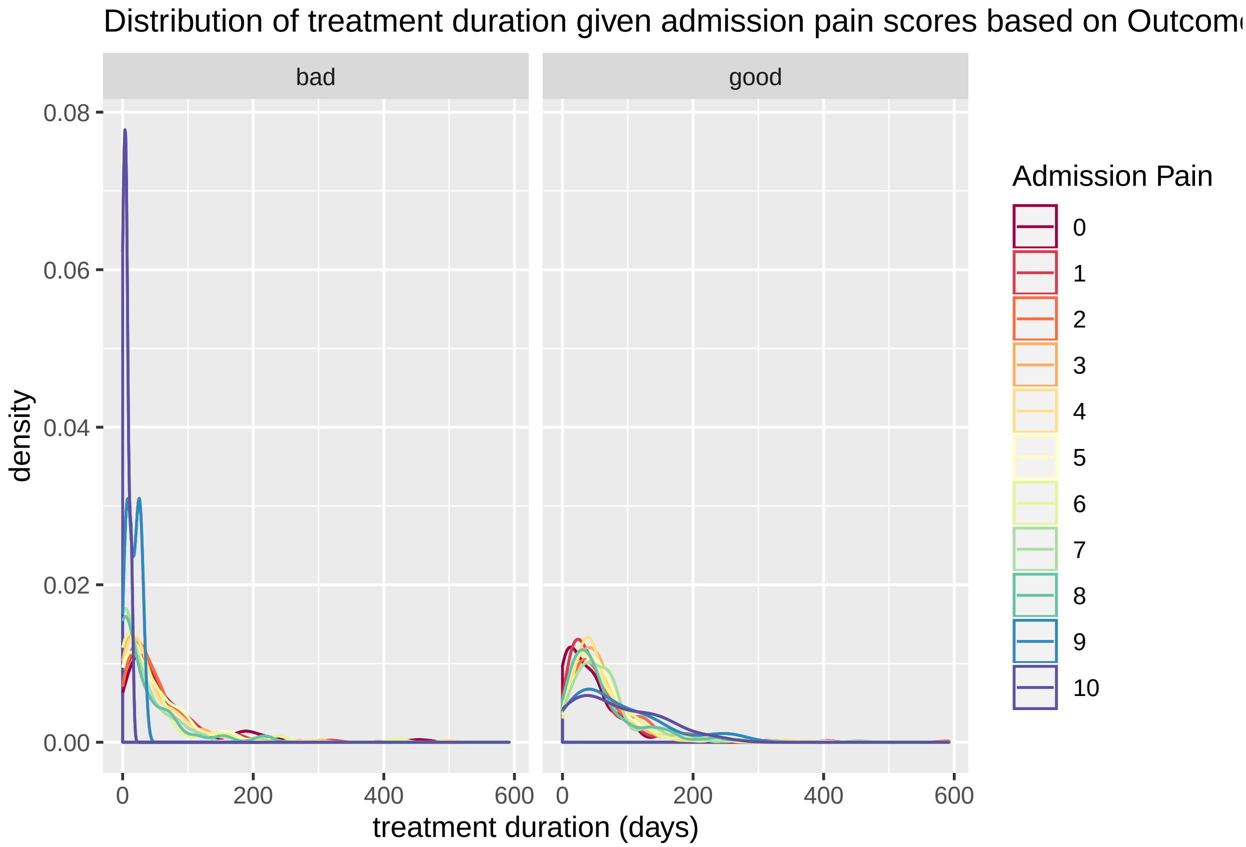
For this project, our team decided on taking on the task of trying to determine what attributes may best differentiate multiple clusters of patients and how they vary within a provided PT patient data set. In theory, this would help our client better understand their overall patient audience, and potentially find niches they may have not be aware of before.

To do this, we made a plan to perform unsupervised learning methods that included Autoencoders and K-Means Clustering to our provided data. Autoencoders are a common type of neural network used to encode high dimensional data into lower dimensions in an unsupervised fashion. Once our data is brought to a lower dimensions, we then planned to apply a K-means clustering layer.

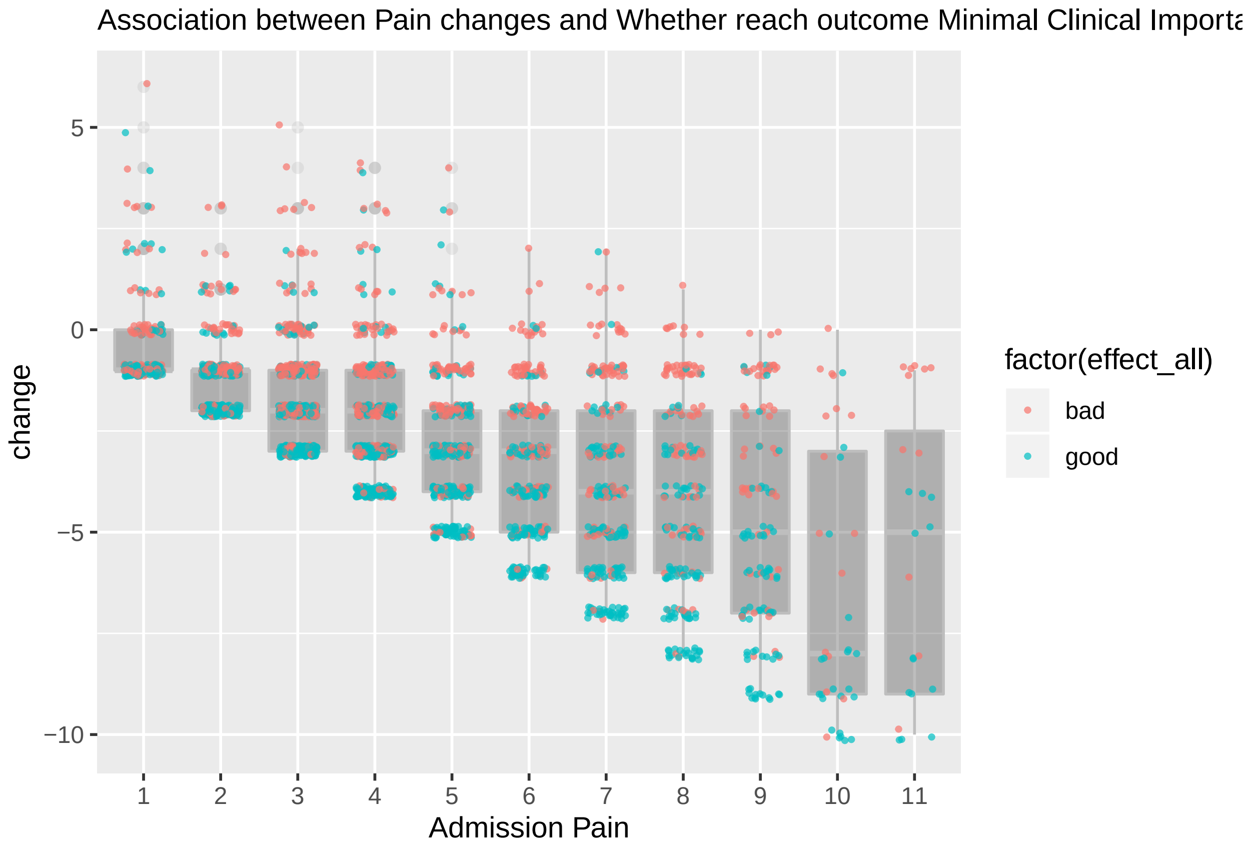
Once the data was clustered, we then could look to make interpretations and conclusions based on our original research question. Prior to EDA, we cleaned the provided data set and the information regarding our data cleaning process can be found in the Appendix.

EDA

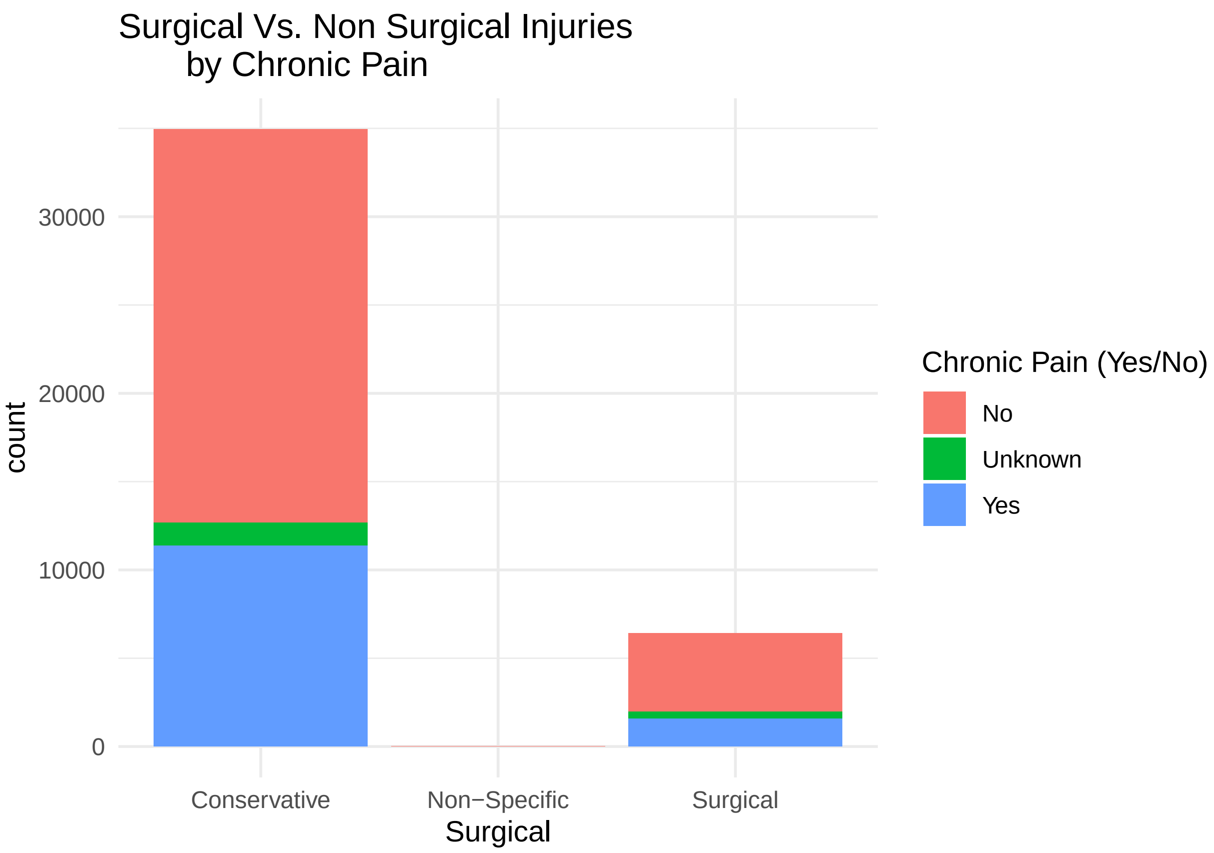
Once we had the data cleaned, we began EDA to gain a better understanding of the data that we were working with. We particularly wanted to look at the variance of variables within the PT patient data, as overall higher variance can compliment more decisive clustering.



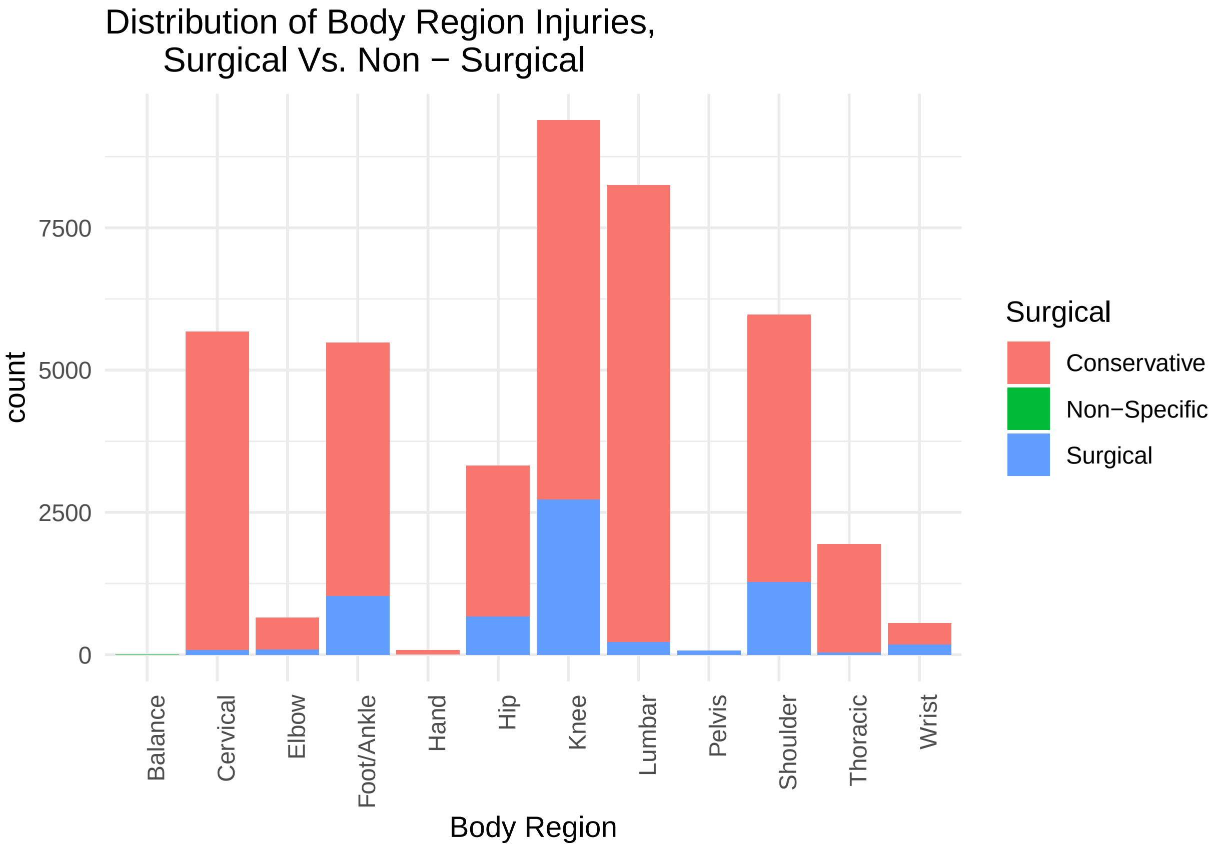
From the plot on the left side, we can see that patients with higher admission pain score tend to have shorter durations which might be due to their change to surgery because of the bad outcome. On the other hand, for the good outcome score, patients with lower pain score would tend to stay for a certain amount of time because of the length of treatment. Patients with higher admission pain score would have longer treatment durations than others.



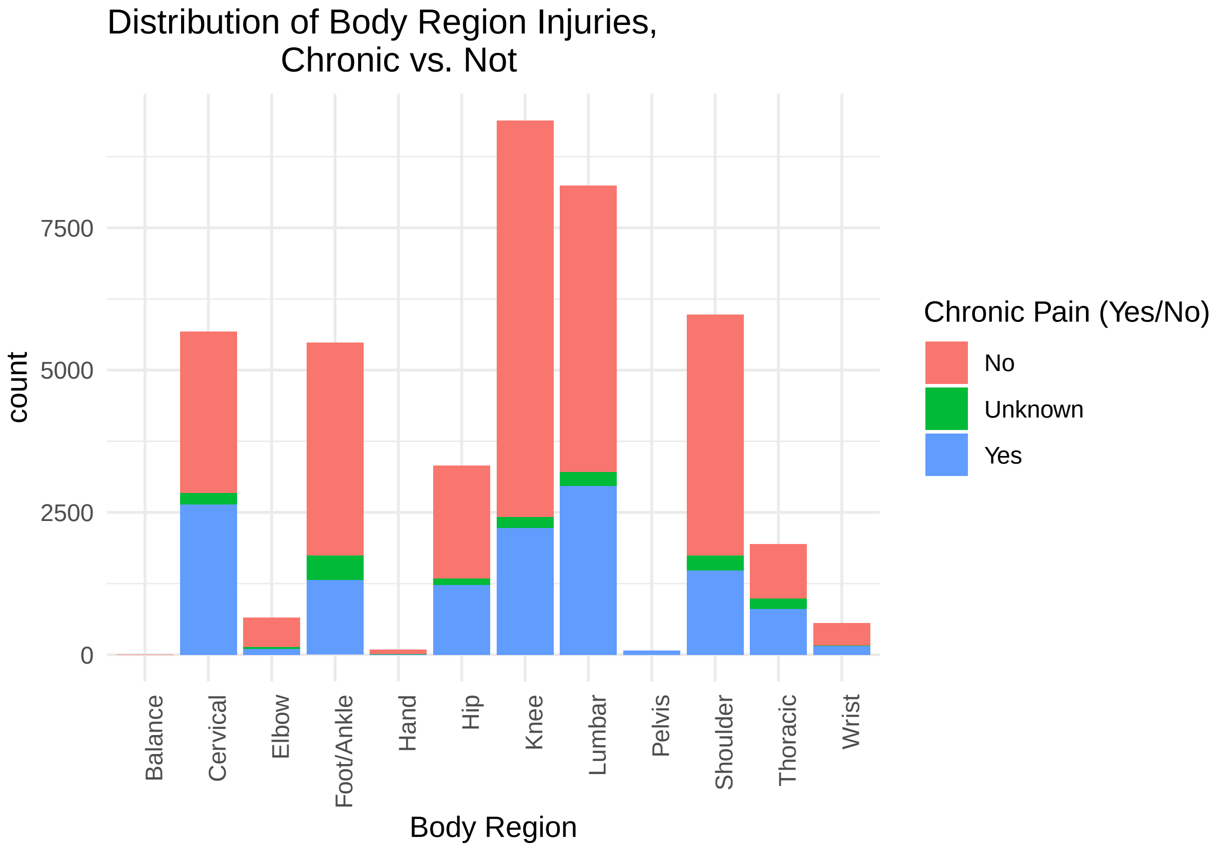
From this plot, we can see the change in pain score between the admission and discharge of patients. The closer to the right of the graph, the more pain the patient had when he/she first came to the treatment. And the lower half of the graph represents a declining of pain score implay an improvement in the condition from the therapy.



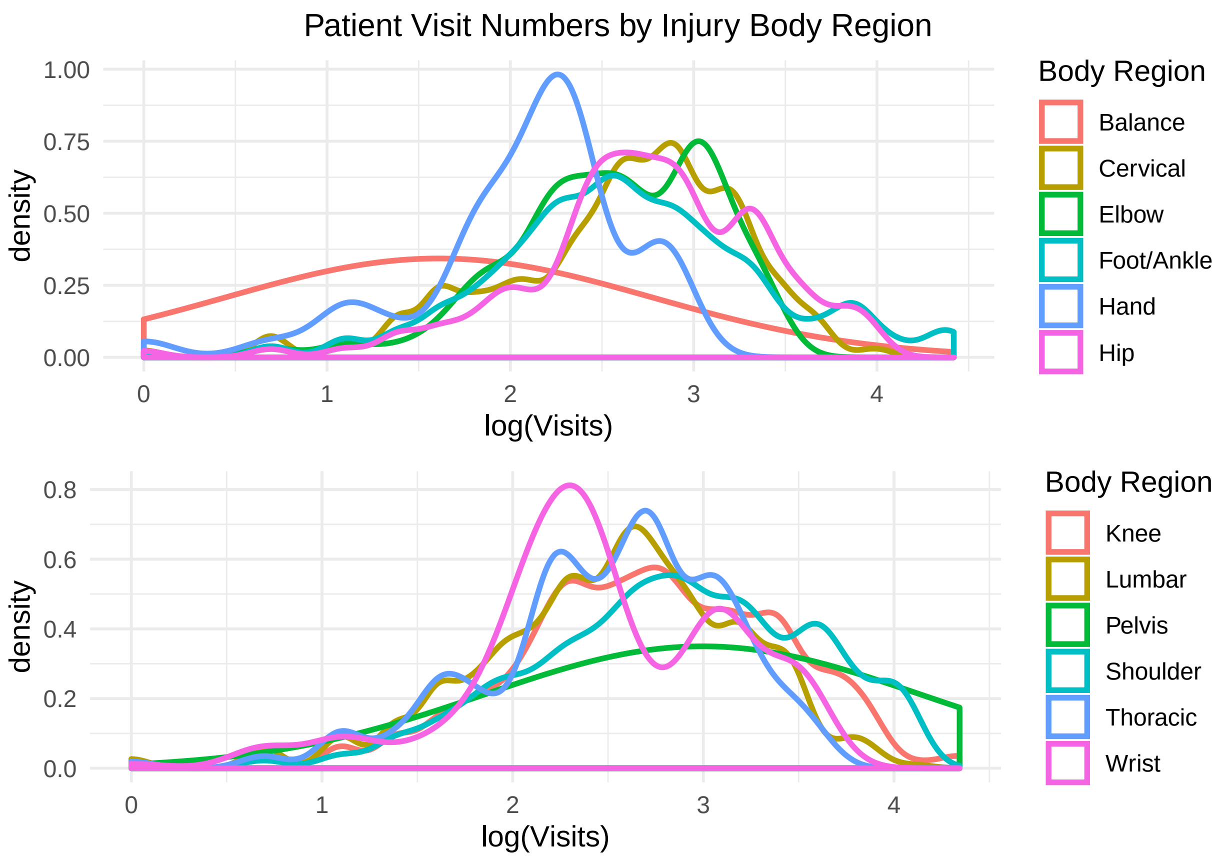
From this plot, surgical injuries did not necessarily have more to do with the chronic pain. However, this plot could not show enough details that we need so we plotted the chronic pain and surgical condition versus body regions in the next part.



Overall, we can see that not many of the patients would take the surgery since most of them are seeking physical therapies. However, we observed that the highest proportion for body injuries for which the patient would take a surgery is a knee pain. Besides knee, we also noticed that foot/ankle and shoulder injuries would be more likely to result in a surgery than injuries in other body regions.



Injuries over most of body regions would result in a chronic pain more likely than taking a surgery. comparing to the former plot, the rise on the rate is obvious excpet for the body regions that were more likely to result in surgeries.



We used logrithm on the number of visits to arrange them on the same scale and we can see that most of visit by injury body region are around 2 to 3. The exponential of 2.5 is around 12 so we suppose that was a treatment phase. Noticeably, for injury at pelvis, the log of visit is more smoothly distributed throughout the region rather than had a densed range like other body regions. This might be helpful for our later analysis.

Autoencoder & K-means Clustering

Once EDA was complete, we had a clean 20-dimension data set that needed to be condensed before we then applied clustering. Therefore, we chose to use a variational autoencoder as a means of encoding the data to 2-dimensions in an unsupervised fashion. Once we get the data in 2-dimensions, we can then apply the k-means algorithm to the data and then work towards interpretation. (ie. Determining what variables from the original PT data set may be responsible to the seperation/break-outs of clusters in 2-dimensions.)

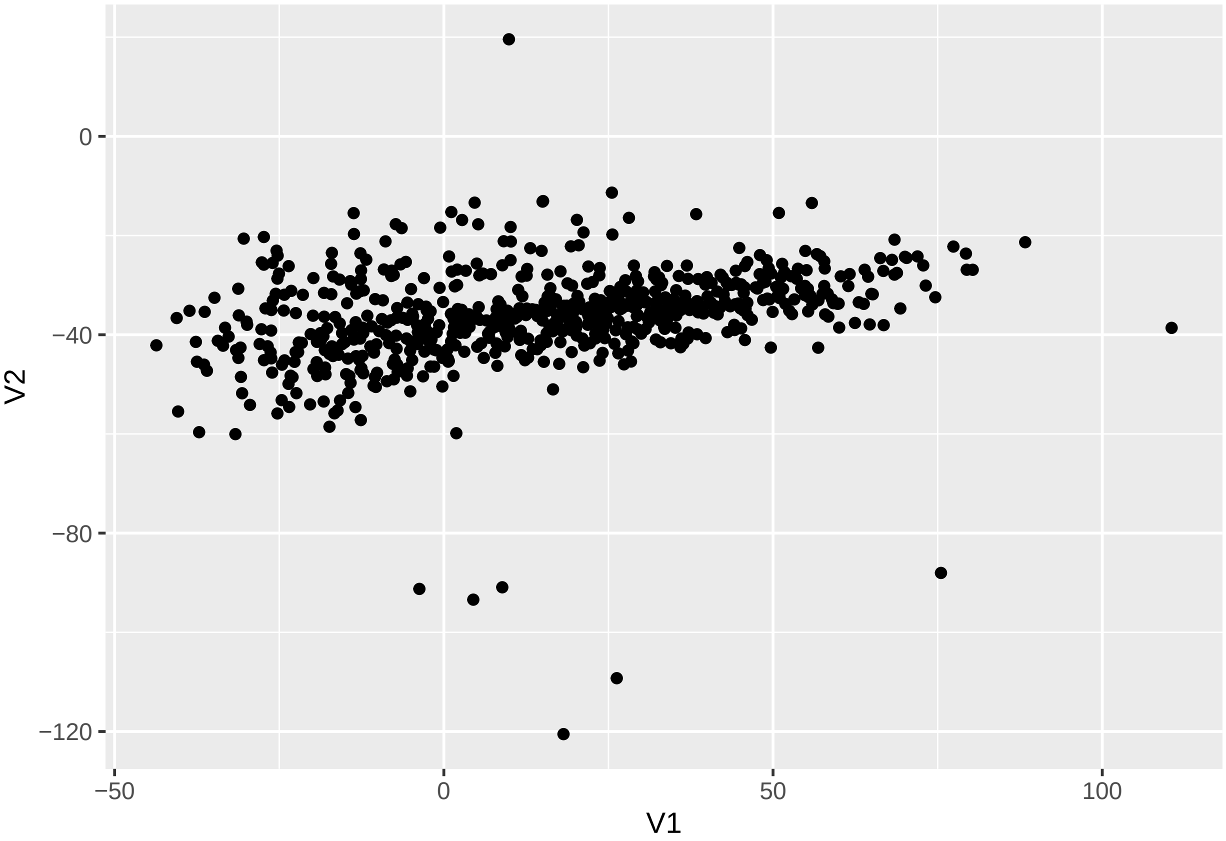
It is important to note, that during the autoencoding process, we had troubles with maintaining the consistency of the autoencoder's neural network 2-dimensional output. Therefore, after significant tweaking and testing, we decided that the best thing to do to ensure reproducibility of the clustering was to save our optimal output from the autoencoder.

The specific of the code run for the autoencoder can be found in the corresponding .Rmd file to this .pdf report. We had our autoencoder hyperparameters set to:

- batch size = 50
- epochs = 20
- original dimension = 20
- intermediate dimension = 12
- latent dimension = 2
- epsilon = 1

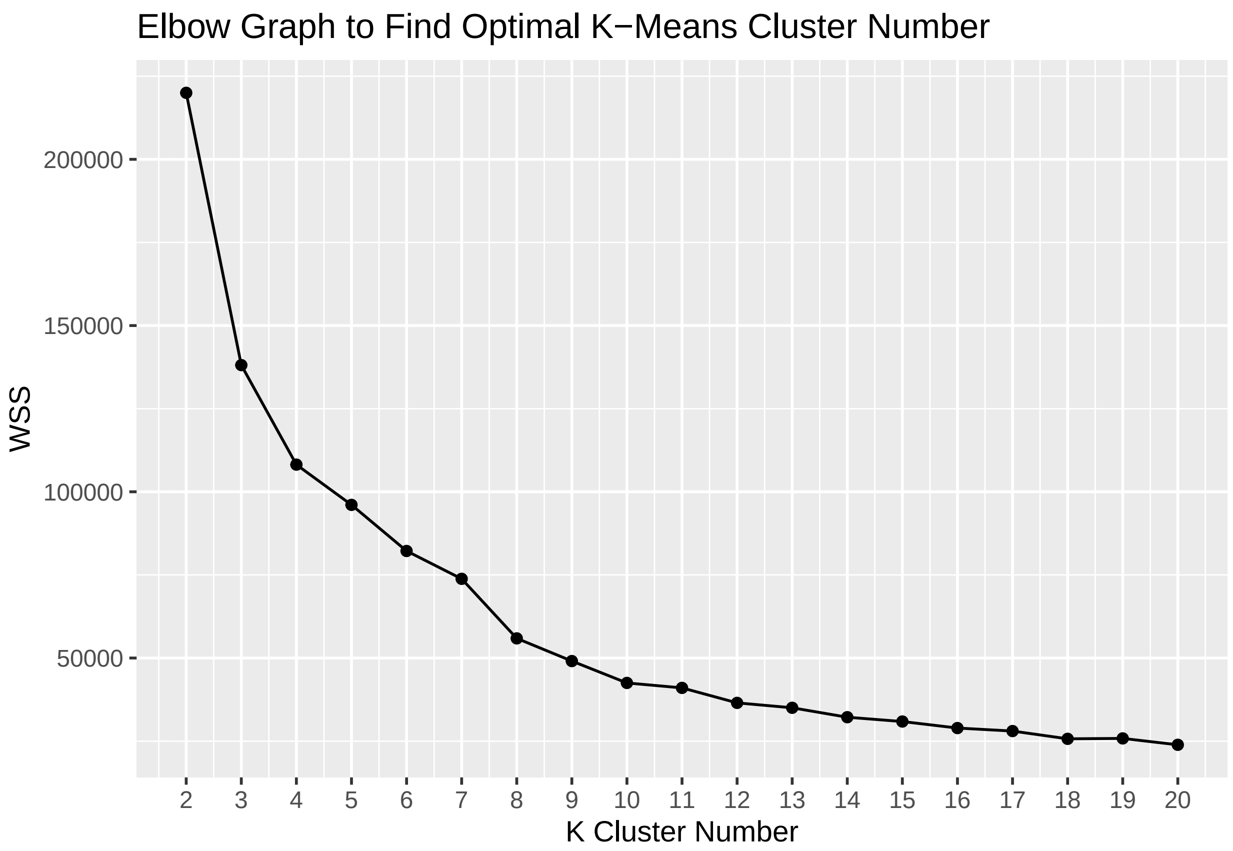
The activation function used when encoded into lower dimensions was “relu”.

Visualizing Encoded Data



K-Means Clustering

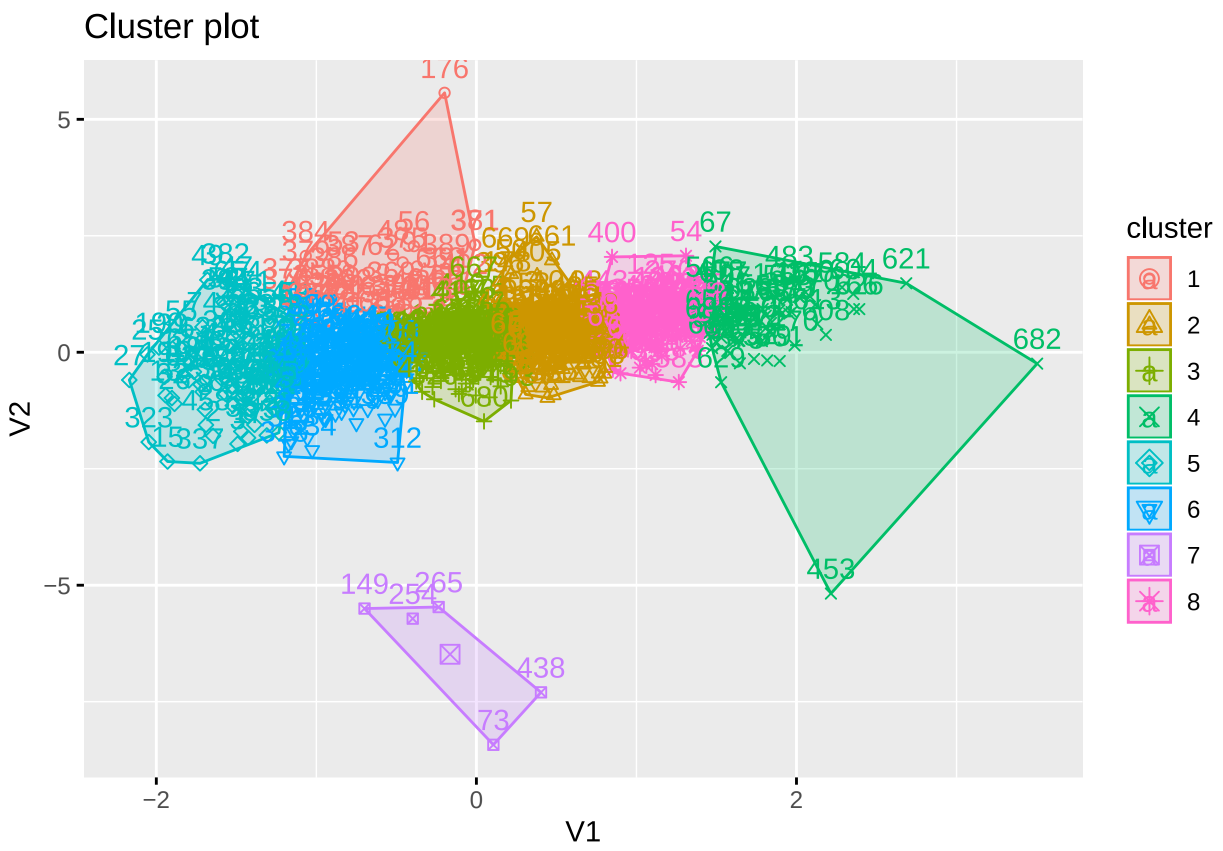
Once our data is encoded in two dimension as seen above, we then want to apply k-means clustering and visualize our results. The first thing we need to do is determine the optimal cluster count.



From this graph we can see that the total within sum of squares for clusters drops significantly until about 8, and from 8 the drop of WSS begins to plateau. Therefore 8 clusters is an efficient cluster number, and we'll set that number for our clustering going forward.

```
## List of 9
##  $ cluster     : int [1:682] 6 5 5 5 5 5 5 5 5 6 ...
##  $ centers     : num [1:8, 1:2] -1.14 28.24 12.59 64.6 -25.88 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:8] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:2] "V1" "V2"
##  $ totss       : num 571421
##  $ withinss    : num [1:8] 6772 7721 5629 10678 10961 ...
##  $ tot.withinss: num 55655
##  $ betweenss   : num 515767
##  $ size        : int [1:8] 46 144 123 58 91 119 5 96
##  $ iter        : int 5
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

From this output we can see that the clustering is showing a significantly lower value of total within sum of squares, compared to it's between sum of squares value. This is telling us that when running the k-means algorithm on our encoded data with 8 clusters, we are getting clusters that encapsulate points that are overall much closer to eachother, compared to their distance to points in *other* clusters, and that is a good thing to see when running k-means.

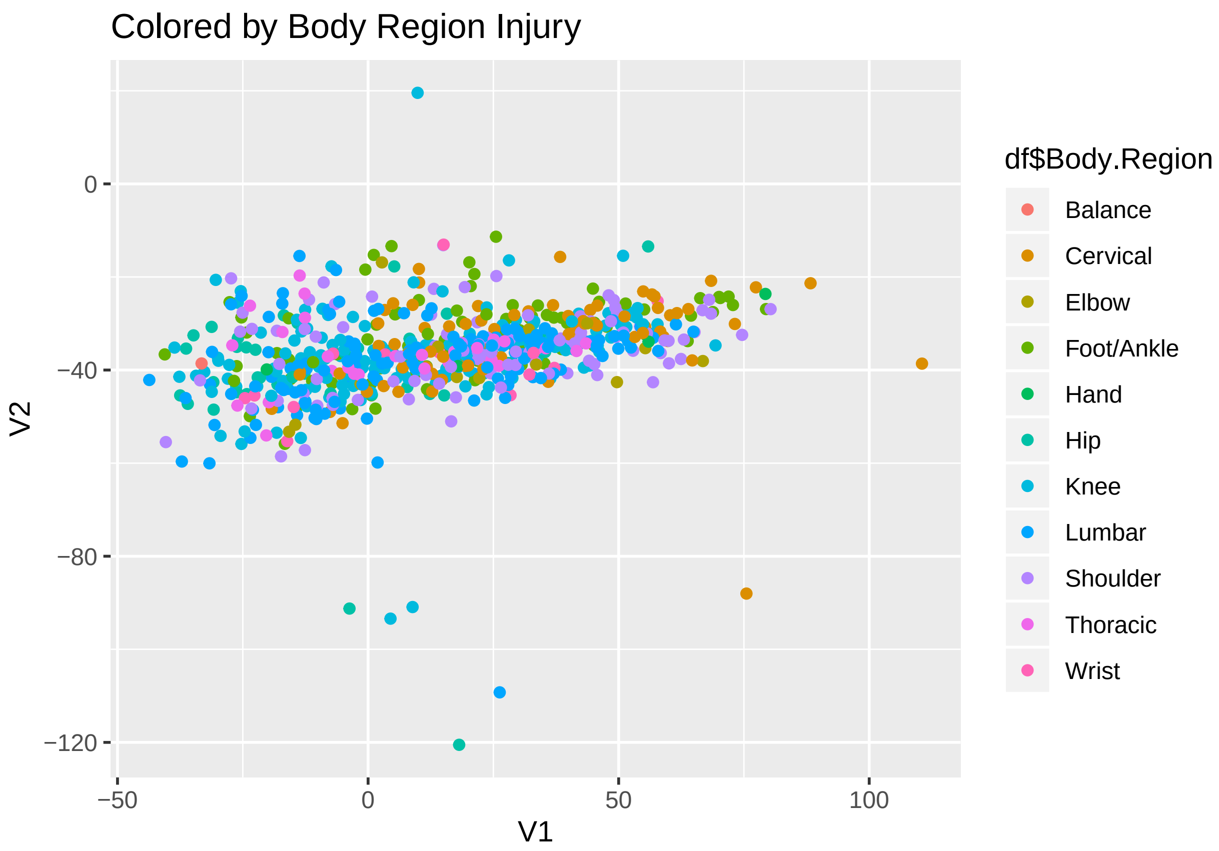


Now, here we can see that the clusters broke out fairly nicely. Clearly the largest divide of clusters is along the horizontal 'V1' axis.

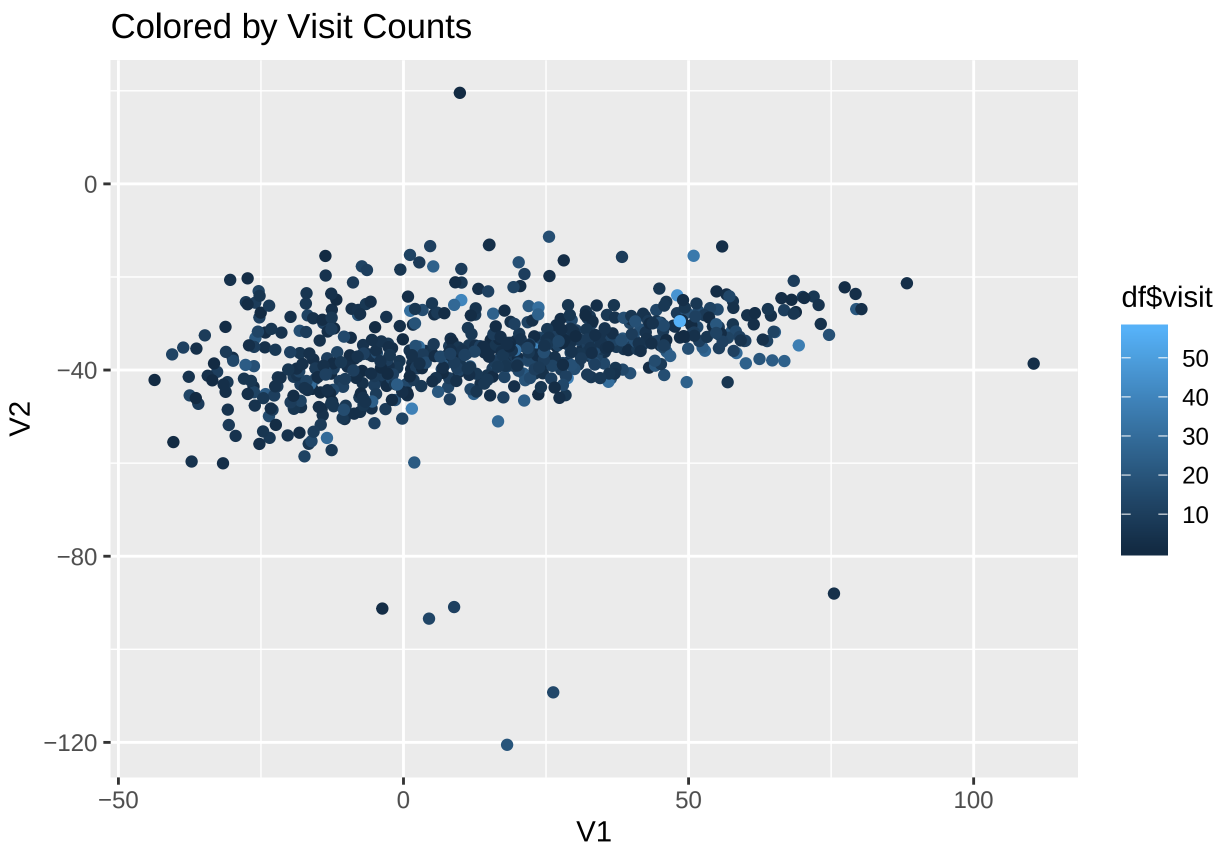
Interpretations

Now that we have the encoded data clustered, we need to get creative and to try and interpret the results. We need to see if we can somewhat mirror the clustered data by visualizing the data with respect to variables from the original PT data set. If we are able to decipher a clear way in which the PT patient groups divide themselves among clusters with respect to variables from the data's original dimensions, then we would then be able to interpret the meaning of the new arbitrary 'V1' and 'V2' variables that were created from the nueral network autoencoder.

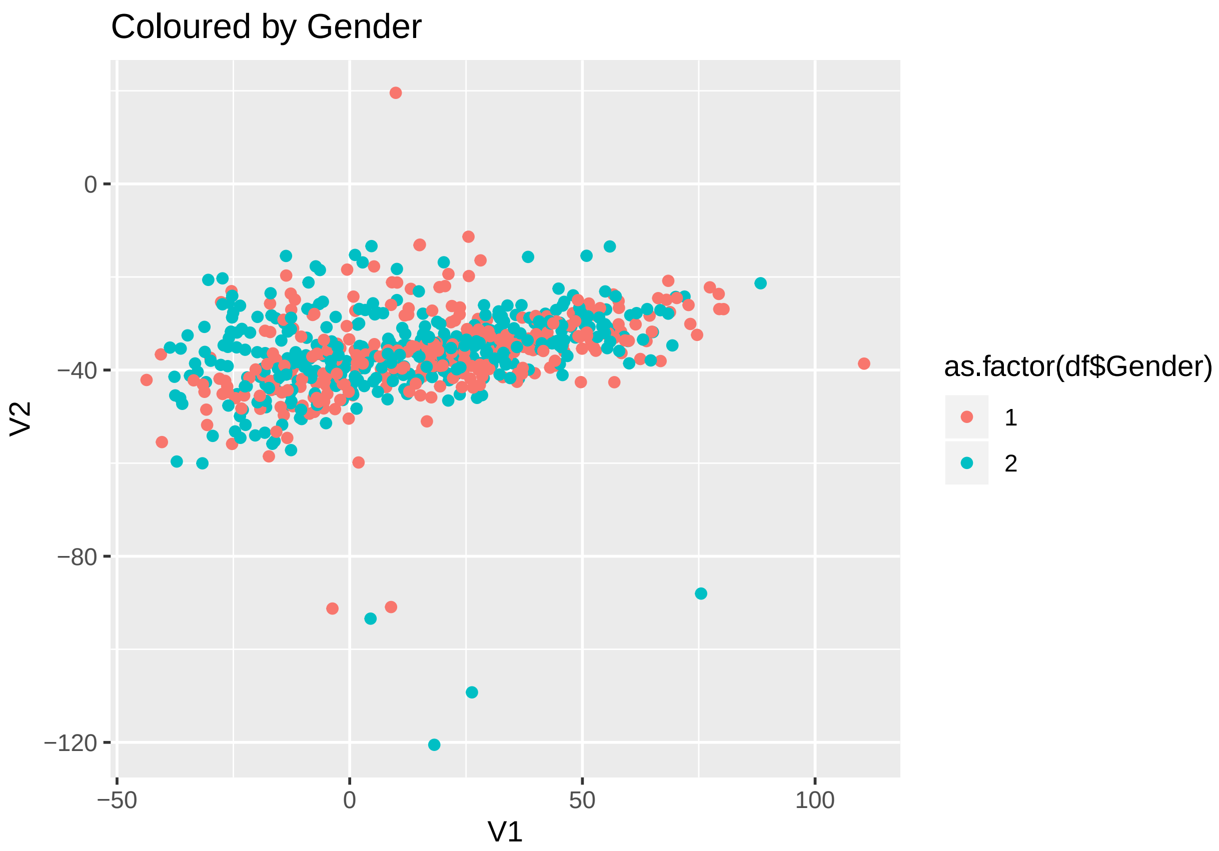
Therefore, what we need to now do is visualize the current encoded data with respect to original variables in the PT patient data.



However, here we can see that there seems to be a strong mix overall here with respect to Body Region, and there is not any trend we can see here that resembles the clusters from above.



Again, here coloring the data by Visit counts taken by the patients we see an overall mix amongst the data, without any clear trend and no decisive groupings like that of the clustering.



Finally we see here that when splitting the encoded data by gender, there also is no clear trend similar to the clustering groups.

For the sake of report length, the rest of the data grouping visualizations can be found in the appendix.

Conclusion

Once we had the data clustered, we needed to get creative to try and derive the representations of the new 2 dimensions for the encoded data, with respect to variables in the original PT data set. However, in our case the results of the k-means clustering didn't seem to mirror any other dividing of the data, based on PT data variables including Body Region Injuries, Chronic Pain, Surgery, Gender, and so on.

The result of our clustering algorithm may seem unexpected, but it is possible. According to the EDA plot "Distributions of treatment duration given admission pain scores", the probability density function curves of treatment duration show similar patterns with each other, which means that Admission Pain Score is not a factor of significant impact on the overall patient clustering. In addition to Admission Pain Score, the EDA plot "Patient Visit Numbers by Injury Body Region" indicates that the distributions of visit times corresponding to each body region are similar, which means that body region may not be an important factor for patient clustering. The EDA plot "Distribution of Body Region Injuries, Chronic vs. Not" agrees the opinion that body regions may not be an important factor for patient clustering. The chronic pain symptom doesn't show specific focus on particular body regions. Instead, the chronic pain keeps proportional to the total counts of treatments.

Therefore, concluding from our analysis, running our autoencoder did not decipher a clear way in which the PT patient groups divide themselves among clusters, when the dimensions are brought down to 2-Dimension. The output of the autoencoder is dependent on the data inputted, and no clear factors within the data that divide the current data.

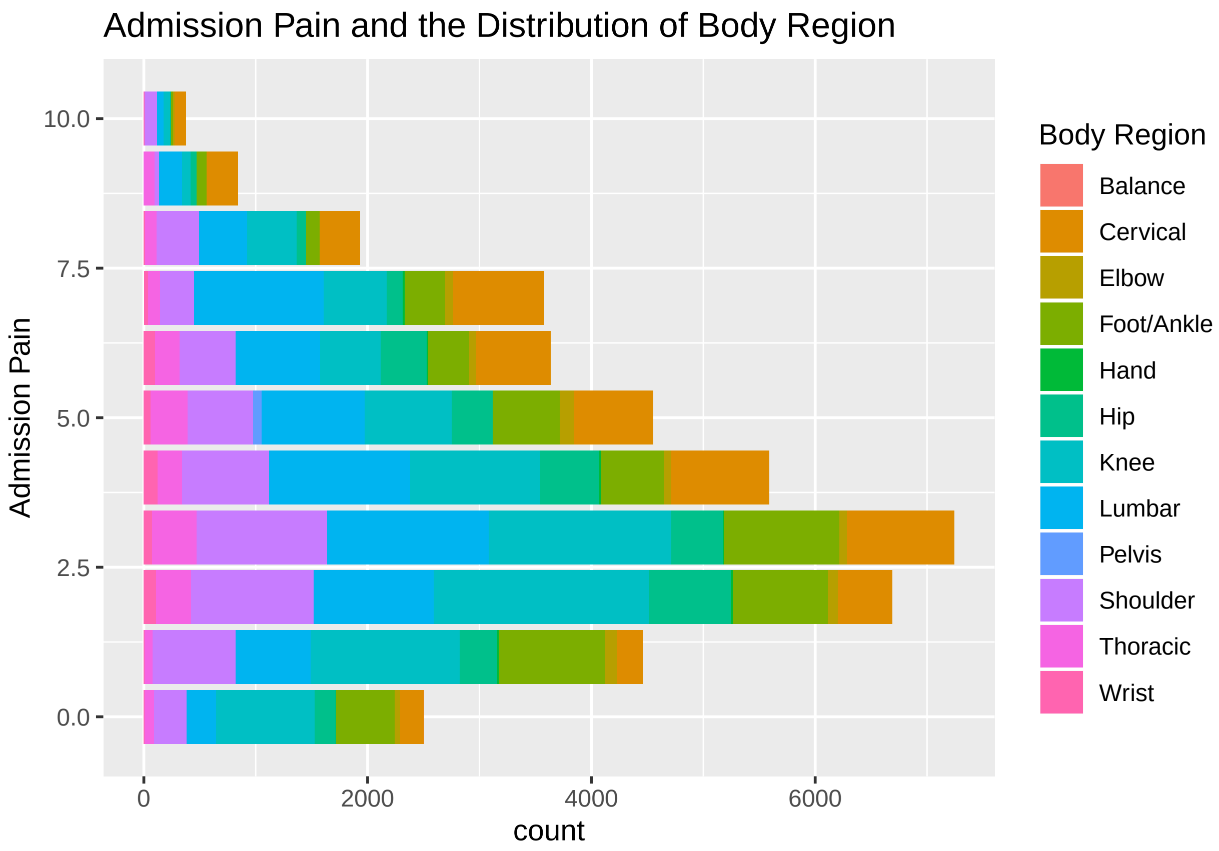
Limitations and Discussion

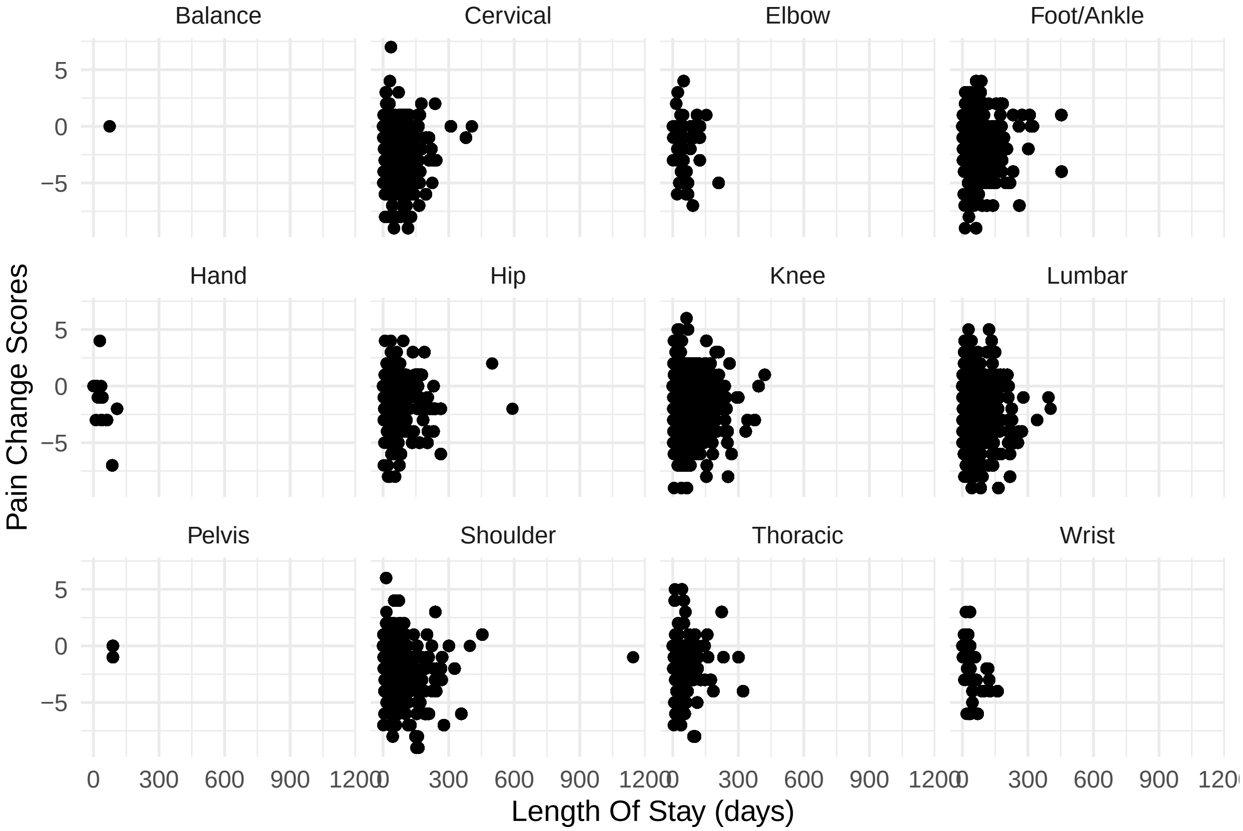
Aside from the unscaled variables in our data set, another potential data issue is that our data set contains both numeric information and text information. Text data needs to be preprocessed prior to building an autoencoder, becasue the Keras package is typically applied to numeric data, as well as the K-means algorithm. One possible solution for handling text data in our case is to create dummy variables and let them represent different text data. However, K-means algorithm performs poorly on categorical/dummy variable data. The reason for this is that the cost function of K-means algorithm usually computes the Euclidean distance between two numeric values. However, it is not possible to define such distance between categorical values.

In order to further optimize the clustering, we also need to confirm the applicability of different activating functions and tune hyperparameters in the future. A better set of hyperparameters and activiting functions may generate better clustering results.

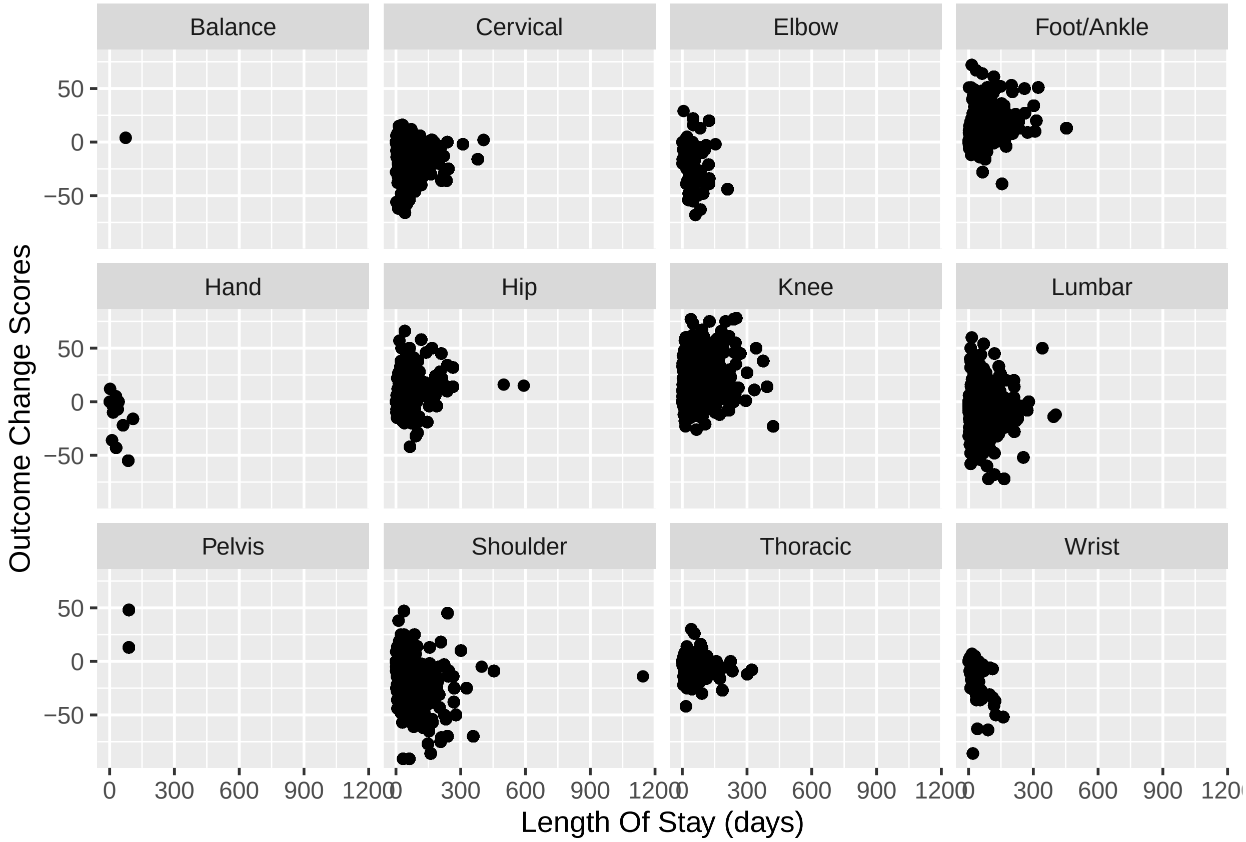
Appendix

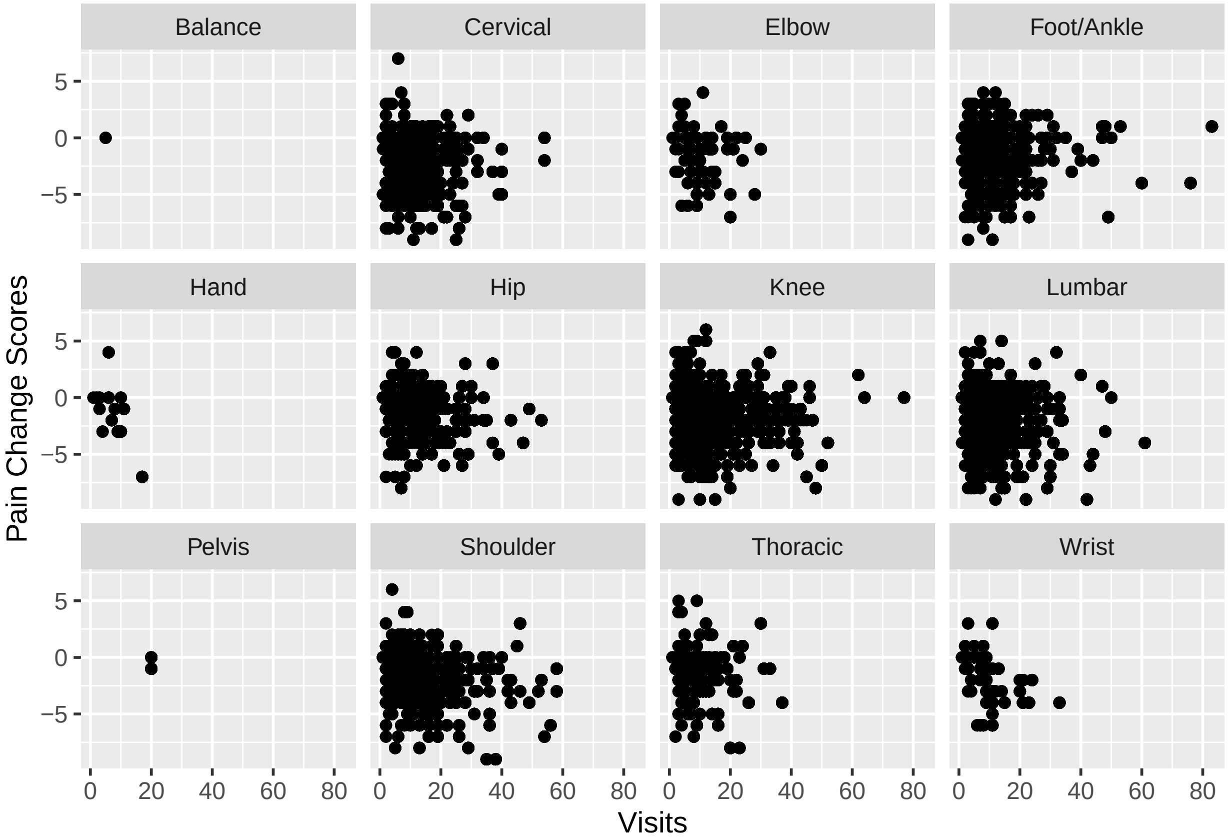
EDA

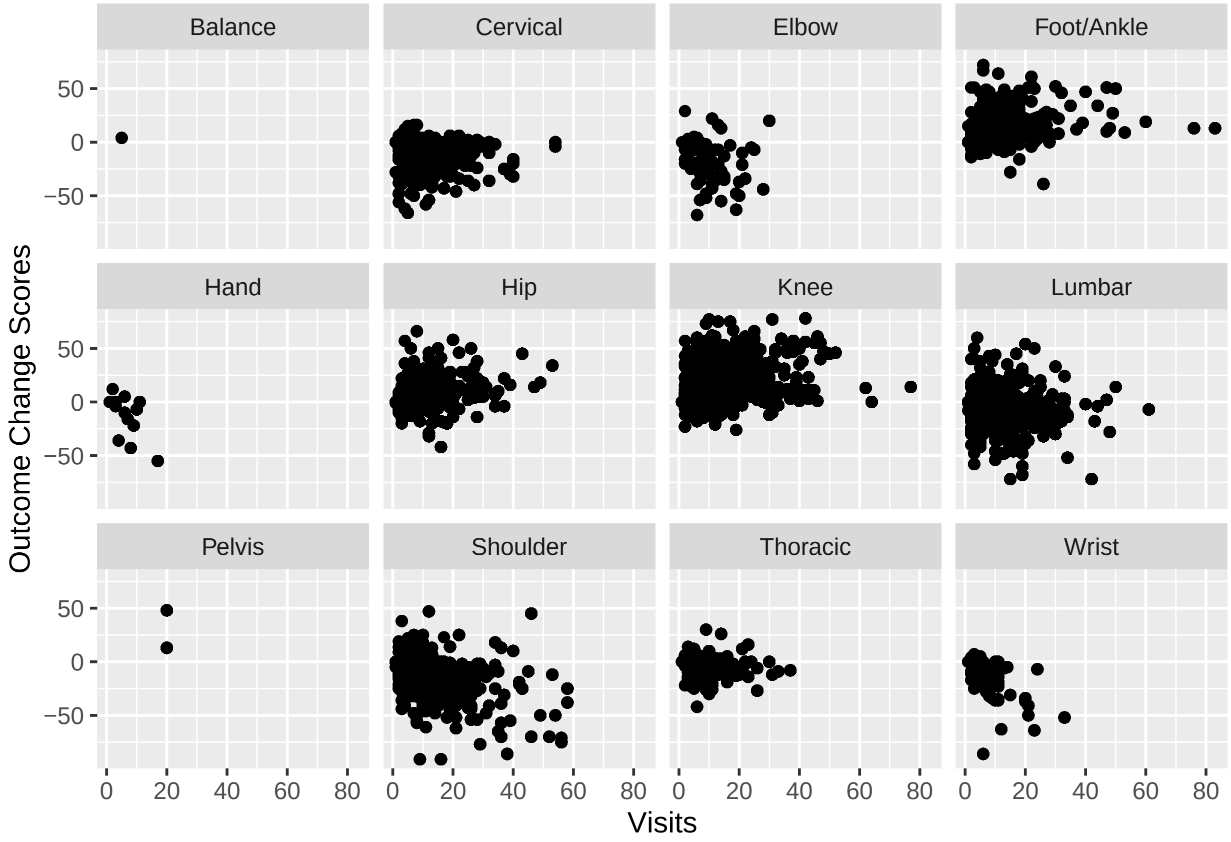


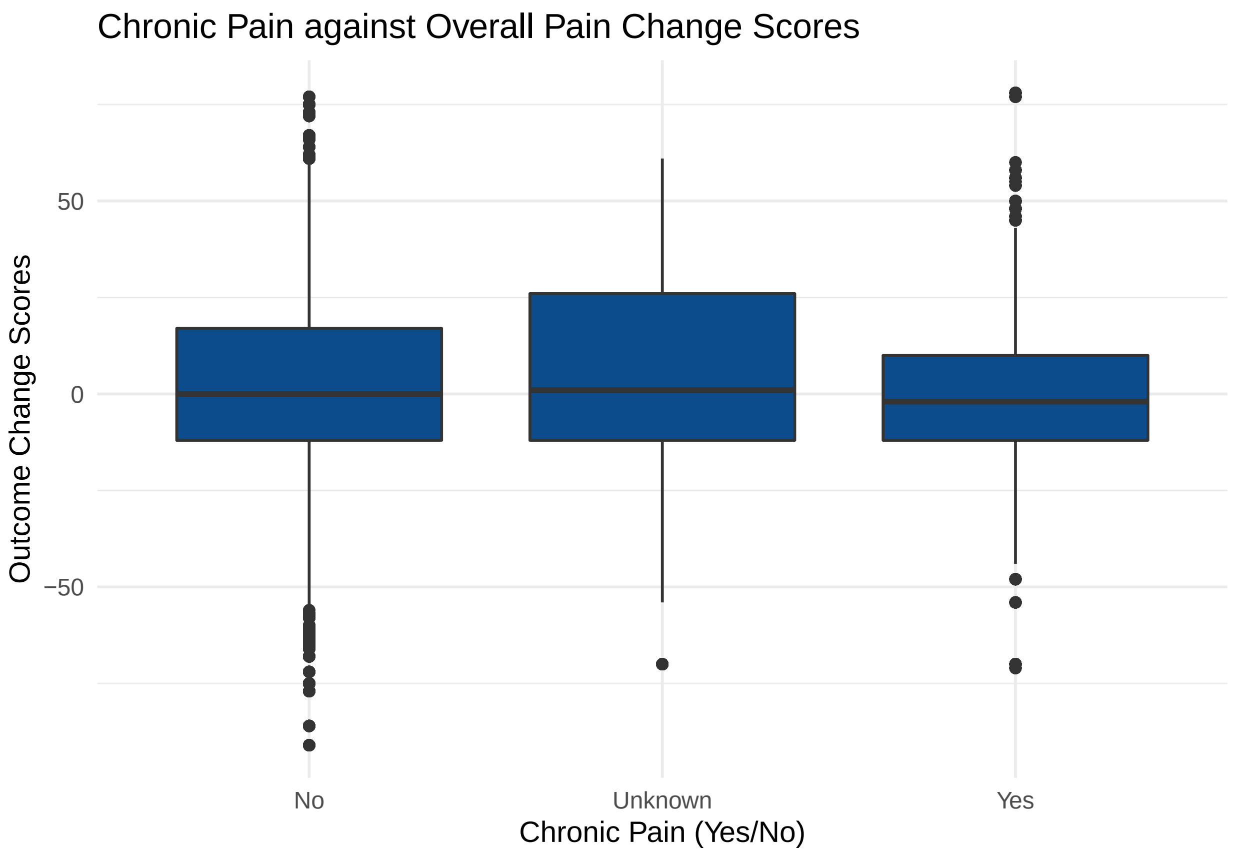


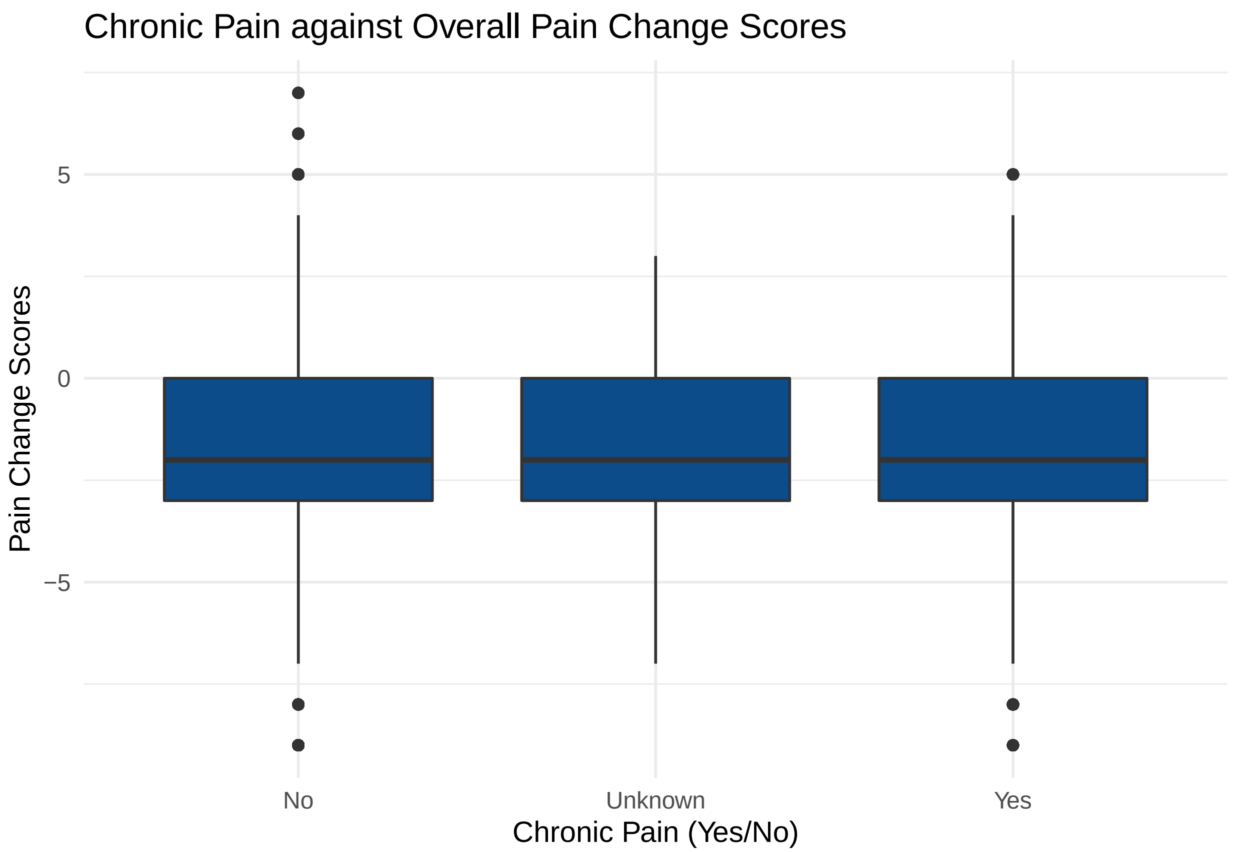
Balance
Cervical
Elbow
Foot/Ankle
Hand
Hip
Knee
Lumbar
Pelvis
Shoulder
Thoracic
Wrist
Pain Change Scores
5
0
−5
0
300
600
900
1200
120
Length Of Stay (days)


Balance
Cervical
Elbow
Foot/Ankle
Hand
Hip
Knee
Lumbar
Pelvis
Shoulder
Thoracic
Wrist
Outcome Change Scores
50
0
−50
0
300
600
900
1200
Length Of Stay (days)

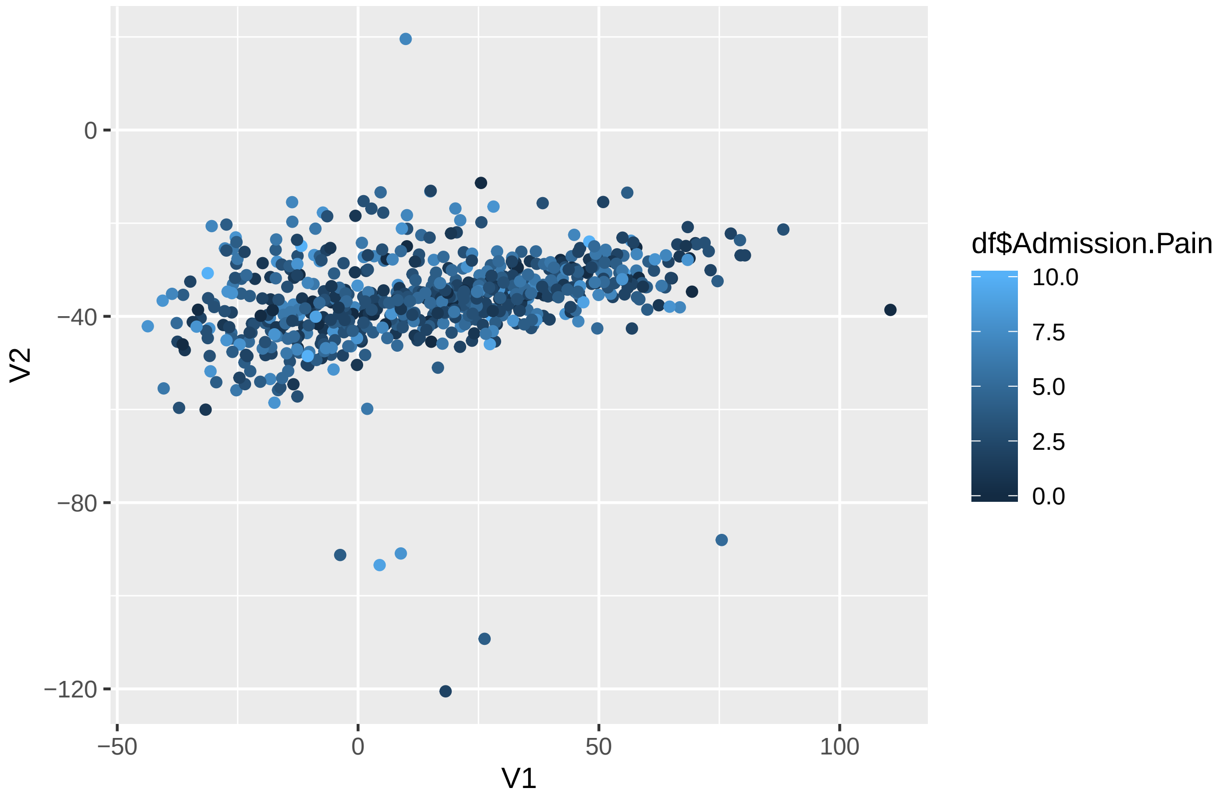

Balance
Cervical
Elbow
Foot/Ankle
Hand
Hip
Knee
Lumbar
Pelvis
Shoulder
Thoracic
Wrist
Pain Change Scores
5
0
−5
Visits
0
20
40
60
80

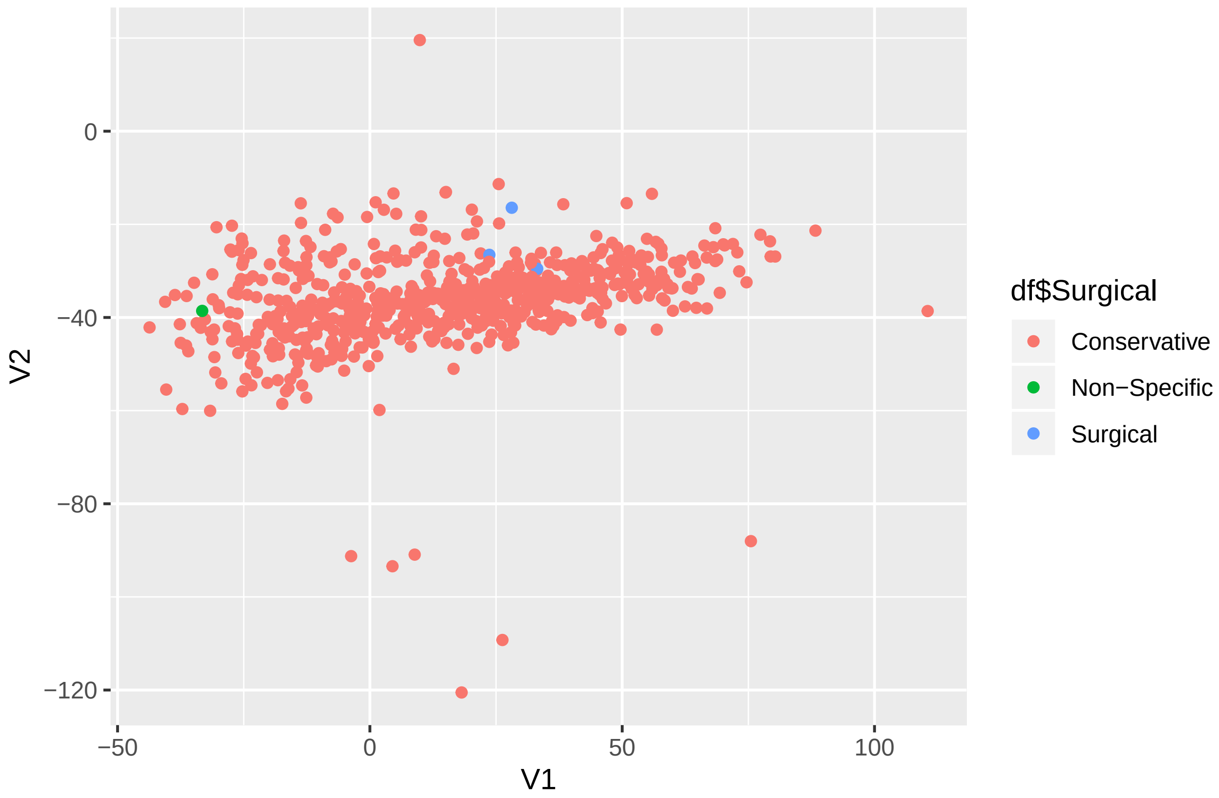

Balance
Cervical
Elbow
Foot/Ankle
Hand
Hip
Knee
Lumbar
Pelvis
Shoulder
Thoracic
Wrist
Outcome Change Scores
50
0
−50
Visits
0
20
40
60
80

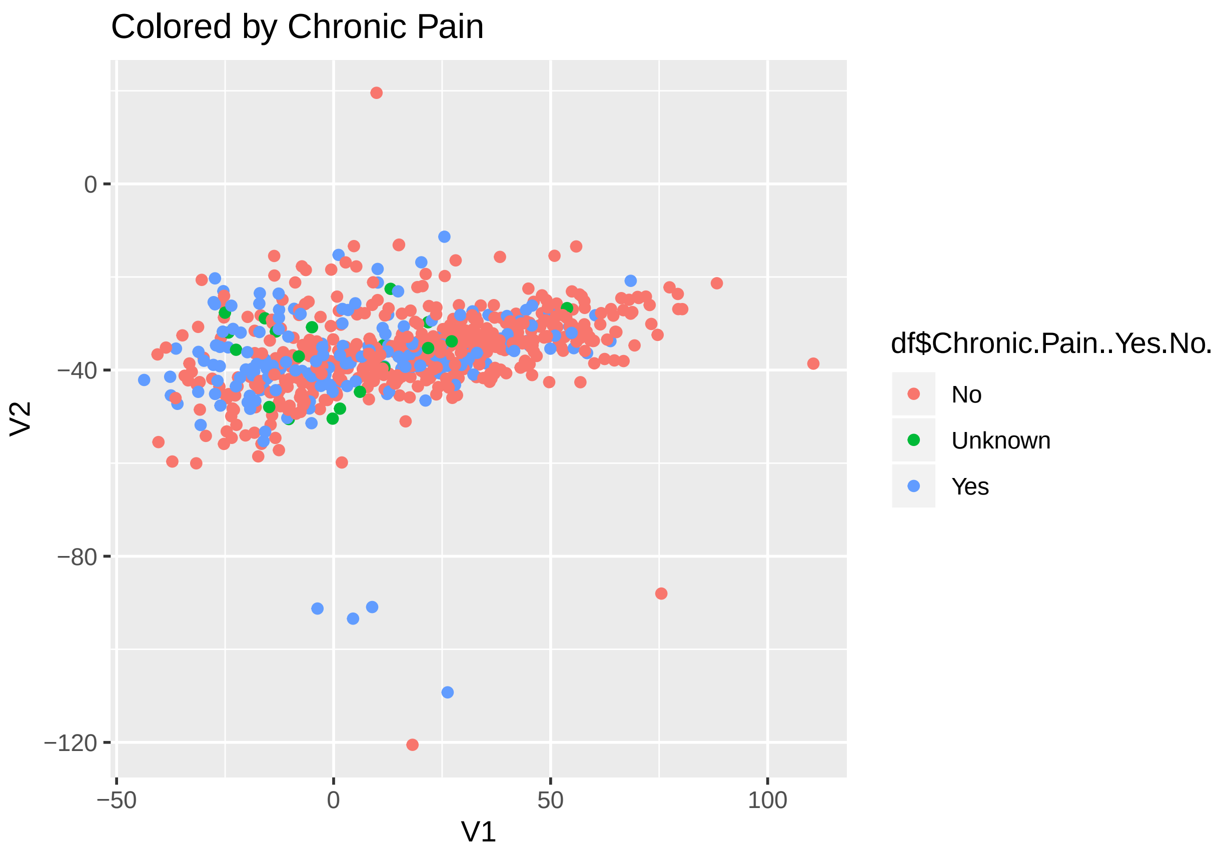

Chronic Pain against Overall Pain Change Scores
50
0
−50
Outcome Change Scores
No
Unknown
Yes
Chronic Pain (Yes/No)


Chronic Pain against Overall Pain Change Scores
Pain Change Scores
5
0
−5
No
Unknown
Yes
Chronic Pain (Yes/No)

Interpretation Groupings


Colored by Admission Pain


Coloured by Surgical



Explanation of Data Cleaning

Our data cleaning process is basically divided into 3 steps. Firstly, we looked into each column, detected and dealt with unusual values. Secondly, we determined the effectiveness of each therapeutic process based on the evaluation criteria of each outcome type. Specifically, we evaluated the effectiveness of pain reduction and the overall improvement. Then, we aggregated some therapy summaries which were logically duplicated, and reevaluated them by taking average.

i)

In the first step, we mainly worked on fixing typos, removing repeated rows, detected and fixed (or removed) problematic values. Typos are general problems caused by inconsistent upper/lower case spelling. And there are many special problems for each column, we analyzed and determined how to smooth the potential errors:

(a) We deleted the record with discharge date in June 2020.

(b) We deleted a 7-digit ROMS ID.

(c) We deleted the negative age and adjusted the inconsistent age based on earlier therapy record.

(d) We adjusted chronic pain as NA if values for chronic pain conflicted with itself.

(e) We fixed inconsistent admission date and discharge date. Particularly, holding other criteria constant (ID, classification, body region, outcome and admission date), if the discharge date varied, then we adjusted all the discharge dates to be the same as the last discharge date of this therapy. Similarly, holding other criteria constant (ID, classification, body region, outcome and discharge date), if the admission date varied, then we adjusted all the admission dates to be the same as the first admission date of this therapy.

(f) After the data cleaning steps mentioned above, we grouped the data by conditions including ROMS ID, Body Region, Outcome, Classification, Admission Date and Discharge Date. We defined each group of data as a combination of therapy records during one entire treatment process. Then we recounted visit times and the time(days) during one entire therapy.

(g) We removed the outcome scores that were out of range. The maximum possible value for LOWER EXTREMITY FUNC SCALE is 80, we eliminated the records for those LOWER EXTREMITY FUNC SCALE values larger than 80 (applied to either admission scores or discharge scores).

ii)

In the second step, we evaluated the effectiveness of pain reduction and the overall improvement of each treatment and gave binary scores to them.

(a) For pain result, if pain scores were reduced by at least 2 points, or reduced from less than 2 to 0, then pain reduction effectiveness was good, we assigned 1 to the result. Otherwise, we considered the therapy's effectiveness of pain reduction was not good, and we assigned 0 to the result.

(b) For overall outcome result, we needed to specify different criteria for different outcome types:

   (1) When outcome type was LOWER EXTREMITY FUNC SCALE, if the score increased at least 9 points or increased from larger than 71 to 80 (max), then we considered the treatment is effective overall, and we assigned the overall effect with score 1. Otherwise, we gave the overall effect with score 0.

   (2) when outcome type was KNEE OUTCOME SURVEY, if the score increased at least 9 points or increased from larger than 91 to 100 (max), then we considered the treatment is effective overall, and we assigned the overall effect with score 1. Otherwise, we gave the overall effect with score 0.

   (3) When outcome belongs to other 3 types, if the score decreased at least 10 points or decreased from less than 10 to 0 (minimum), then we considered the treatment is effective overall, and we assigned the overall effect with score 1. Otherwise, we gave the overall effect with score 0.

iii)

After those 2 steps, then we aggregated some therapy summaries and evaluated them by taking average. For the treatments with identical same ID, body region, classification and treatment time, if the categories of outcome are different, we took the average of the pain reduction effectiveness scores (1 or 0) and took the average of the effect for overall improvement (1 or 0). Since we aggregated some rows, some columns became invalid logically, so they are assigned NA.