DS110 Project 1 - Eye Detection

Nathan Gonyo, Paul Ritter, and Elliot Schendel
Professor Rasitha Jayasekare

DS 110

11/04/2020

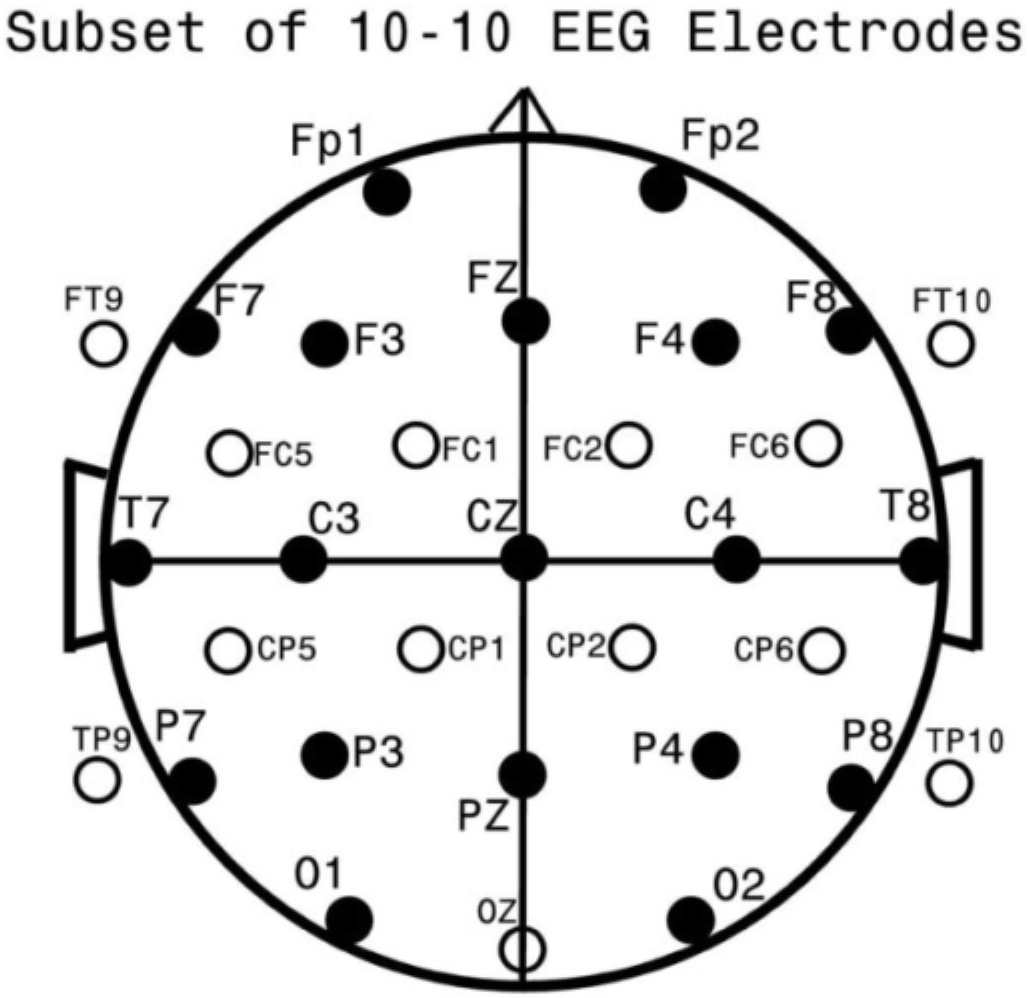


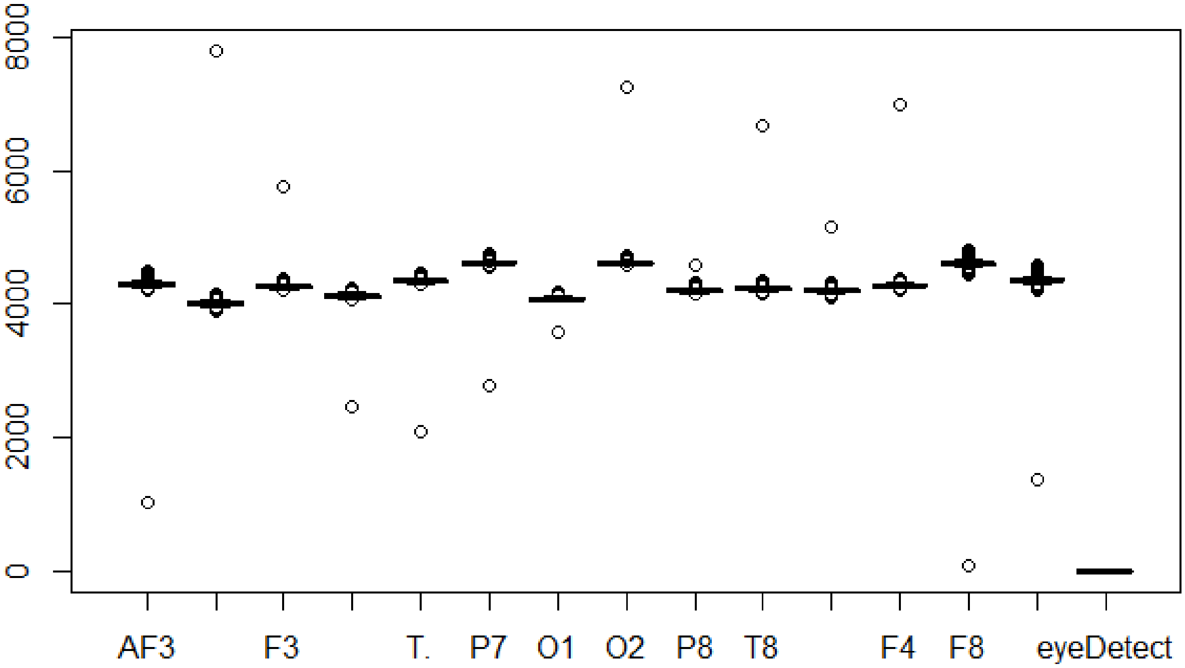
Table of Contents

Pre-Processing

One of the first steps we undertook for this project was preprocessing the data such that it could be used for further analysis. If we didn't do this, our analyses could not be as accurate as possible due to the datasets being generally unorganized, inconsistent due to outliers and missing values for certain records (among other reasons).

Outlier Removal

After we understood the variables present in our original dataset, we took the first step in preprocessing by removing all rows of records which had missing values from the same dataset. Once completed, our group generated a random sample of 5,000 records from the original dataset (which was now reduced). This new data represented a reduced dataset that would be condensed even further by removing all possible outliers. Our strategy for removing outliers was creating a boxplot that exposed all of the outliers present in our 5,000 records; below is the first boxplot of the dataset (note: the y-axis values represent the electrical activity and the x-axis shows the variables used to measure the electrical activity as well as showing the 'eyeDetect' variable).



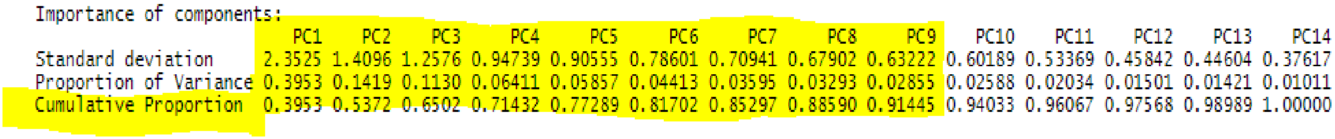
Once we modeled the presence of outliers, we began methodically removing outliers in waves (four [4] rounds of outlier removal were conducted overall), starting with AF3 and ending with F8. This method allowed us to consistently remove outliers as they appeared until no outliers remained; this approach removed more than 33% of the total records from our new dataset which left us with a dataset of 3257 outlier-free records. Below is a table that shows how many outlier records were removed based on each variable in each respective round of outlier removal. For example, 515 records were removed from the dataset in 'Round 1' based on the outliers present in 'AF3'.

	Outlier Removal Table													
	Variables													
	AF3	**F7**	**F3**	**FC 5**	**T.**	**P7.**	**O1**	**O2**	**P8**	**T8**	**FC 6**	**F4**	**F8**	**AF4**
Round 1	515	63	110	62	68	131	23	25	36	89	97	12	87	46
Round 2	77	39	9	28	3	27	21	1	2	21	22	4	13	11
Round 3	19	20	0	0	5	8	0	0	0	8	12	0	0	2
Round 4	12	7	0	0	0	0	1	0	0	0	0	0	7	0
Total	**623**	**129**	**119**	**90**	**76**	**166**	**45**	**26**	**38**	**118**	**131**	**16**	**107**	**59**

Once we developed an outlier-free dataset, we then randomly generated 'test' and 'train' datasets by randomly selecting 80% of the records from our now outlier-free dataset. These records are selected and become our 'train' dataset; we then created the 'test' dataset by subtracting the 'train dataset' from the outlier-free dataset mentioned above (representing 20% of records from the outlier-free dataset). 'Trained' datasets represent the records we use to make models, whereas the 'test' dataset will be implemented to gauge the accuracy of the models shown below.

PCA Analysis

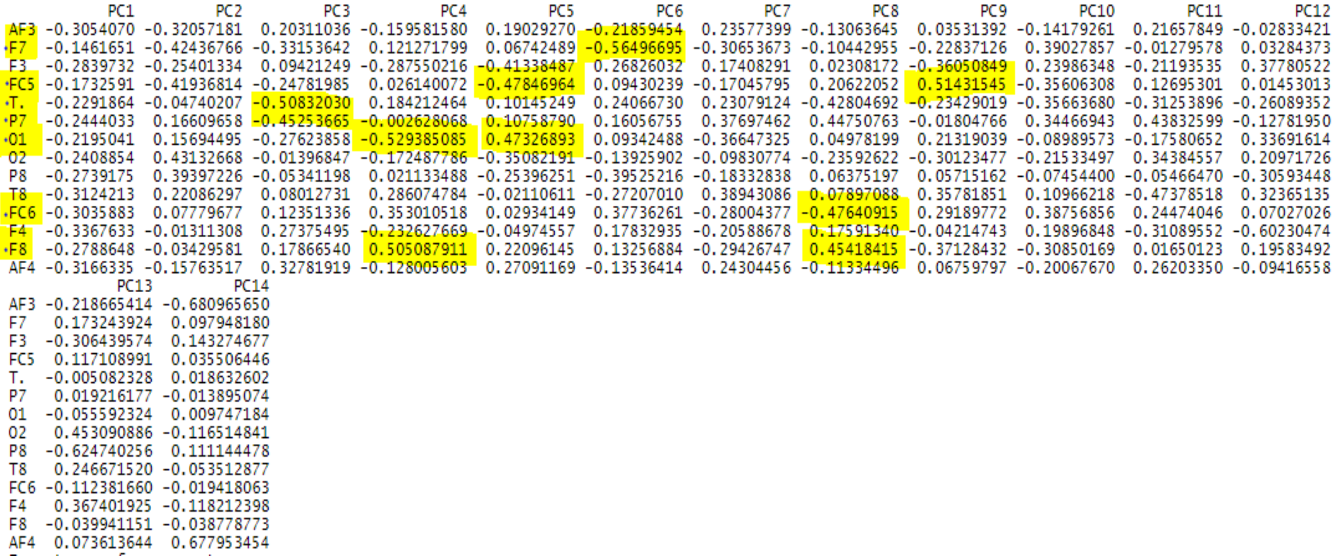
The final step in preprocessing that was undertaken was reducing dimensionality through PCA (principal component analysis). The primary objective of PCA is using the least amount of 'principal components' to illustrate the maximum variance seen in the 'train' dataset and help establish visible patterns in the data. We first began by observing the cumulative proportions of each principal component in our 'train' dataset, looking for the first PCA that has a proportion equal to at least .90 (90%). Below is the output of PCA proportions; highlighted areas represent the principal components our group selected.As seen below, 'PC9' is the first PC that has a cumulative proportion of at least .90.



```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12    PC13    PC14
Standard deviation     2.3525 1.4096 1.2576 0.94739 0.90555 0.78601 0.70941 0.67902 0.63222 0.60189 0.53369 0.45842 0.44604 0.37617
Proportion of Variance 0.3953 0.1419 0.1130 0.06411 0.05857 0.04413 0.03595 0.03293 0.02855 0.02588 0.02034 0.01501 0.01421 0.01011
Cumulative Proportion  0.3953 0.5372 0.6502 0.71432 0.77289 0.81702 0.85297 0.88590 0.91445 0.94033 0.96067 0.97568 0.98989 1.00000
```

Afterwards, our group decided to select variables which had an r value of at least |.49|, as we initially believed that this was a high enough value to remove enough variables from the dataset.

Upon further analysis, we removed ten (10) variables from our dataset, being left with the following variables: F7, FC5, T., O1, and F8. We changed this r value to |.45| so that we did not remove as many variables as previously mentioned while also conserving a reasonable amount of variables from our dataset. Using an r value of at least .45, we selected the following to be retained moving forward: T., P7, O1, FC5, F7, FC6, and F8. Below is the PCA analysis used to identify these variables (the highlighted values represent instances where the r value is >|.45|).



```
            PC1         PC2         PC3          PC4         PC5         PC6         PC7         PC8         PC9        PC10        PC11        PC12
AF3 -0.3054070 -0.32057181  0.20311036 -0.159581580  0.19029270 -0.21859454  0.23577399 -0.13063645  0.03531392 -0.14179261  0.21657849 -0.02833421
F7  -0.1461651 -0.42436766 -0.33153642  0.121271799  0.06742489 -0.56496695 -0.30653673 -0.10442955 -0.22837126  0.39027857 -0.01279578  0.03284373
F3  -0.2839732 -0.25401334  0.09421249 -0.287550216 -0.41338487  0.26826032  0.17408291  0.02308172 -0.36050849  0.23986348 -0.21193535  0.37780522
FC5 -0.1732591 -0.41936814 -0.24781985  0.026140072 -0.47846964  0.09430239 -0.17045795  0.20622052  0.51431545 -0.35606308  0.12695301  0.01453013
T.  -0.2291864 -0.04740207 -0.50832030  0.184212464  0.10145249  0.24066730  0.23079124 -0.42804692 -0.23429019 -0.35663680 -0.31253896 -0.26089352
P7  -0.2444033  0.16609658 -0.45253665 -0.002628068  0.10758790  0.16056755  0.37697462  0.44750763 -0.01804766  0.34466943  0.43832599 -0.12781950
O1  -0.2195041  0.15694495 -0.27623858 -0.529385085  0.47326893  0.09342488 -0.36647325  0.04978199  0.21319039 -0.08989573 -0.17580652  0.33691614
O2  -0.2408854  0.43132668 -0.01396847 -0.172487786 -0.35082191 -0.13925902 -0.09830774 -0.23592622 -0.30123477 -0.21533497  0.34384557  0.20971726
P8  -0.2739175  0.39397226 -0.05341198  0.021133488 -0.25396251 -0.39525216 -0.18332838  0.06375197  0.05715162 -0.07454400 -0.05466470 -0.30593448
T8  -0.3124213  0.22086297  0.08012731  0.286074784 -0.02110611 -0.27207010  0.38943086  0.07897088  0.35781851  0.10966218 -0.47378518  0.32365135
FC6 -0.3035883  0.07779677  0.12351336  0.353010518  0.02934149  0.37736261 -0.28004377 -0.47640915  0.29189772  0.38756856  0.24474046  0.07027026
F4  -0.3367633 -0.01311308  0.27375495 -0.232627669 -0.04974557  0.17832935 -0.20588678  0.17591340 -0.04214743  0.19896848 -0.31089552 -0.60230474
F8  -0.2788648 -0.03429581  0.17866540  0.505087911  0.22096145  0.13256884 -0.29426747  0.45418415 -0.37128432 -0.30850169  0.01650123  0.19583492
AF4 -0.3166335 -0.15763517  0.32781919 -0.128005603  0.27091169 -0.13536414  0.24304456 -0.11334496  0.06759797 -0.20067670  0.26203350 -0.09416558
            PC13         PC14
AF3 -0.218665414 -0.680965650
F7   0.173243924  0.097948180
F3  -0.306439574  0.143274677
FC5  0.117108991  0.035506446
T.  -0.005082328  0.018632602
P7   0.019216177 -0.013895074
O1  -0.055592324  0.009747184
O2   0.453090886 -0.116514841
P8  -0.624740256  0.111144478
T8   0.246671520 -0.053512877
FC6 -0.112381660 -0.019418063
F4   0.367401925 -0.118212398
F8  -0.039941151 -0.038778773
AF4  0.073613644  0.677953454
```

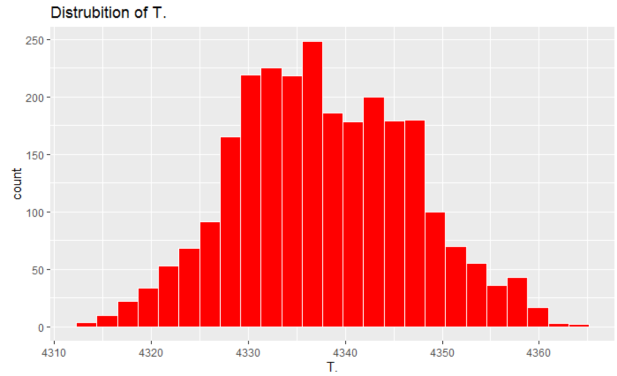
Numerical & Visual Descriptions of Variables

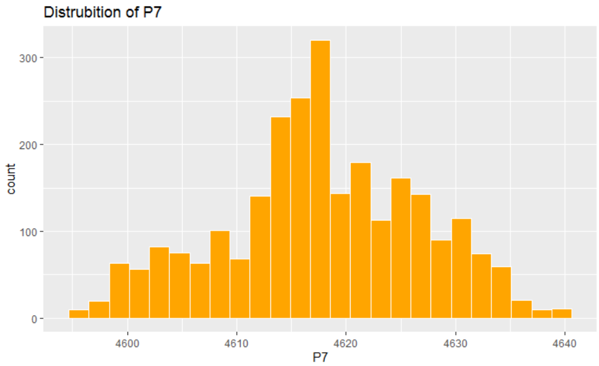
After we completed PCA Analysis and removed the variables which we deemed unnecessary to conduct modeling, we performed described the variables numerically, as well as through some data visualization using ggplot2. This gives us a better understanding of how each variable is distributed and their relationships with each other, especially considering each variable's relationship with the rate of eye detection. Keep in mind that since we are doing this after the PCA analysis, it will only include the seven (7) variables that were not removed from the dataset, as well as only values that were left over after the multiple rounds of outlier removal.

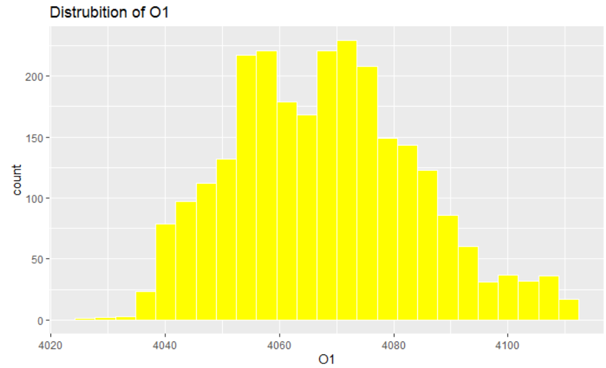
	Numerical Descriptions of Variables							
	Variables							
	T.	P7	O1	FC5	F7	FC6	F8	eyeDetect
Min	4312	4596	4027	4083	3952	4168	4563	0
1st Qrtr	4331	4613	4056	4108	3989	4191	4591	-

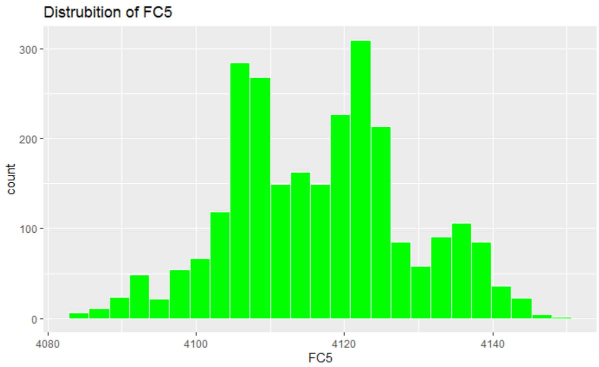
Median	4337	4617	4086	4117	4001	4198	4601	-
Mean	4338	4618	4068	4117	4002	4199	4601	-
3rd Qrtr	4344	4624	4078	4124	4013	4207	4610	-
Max	4363	4640	4112	4148	4051	4231	4638	1

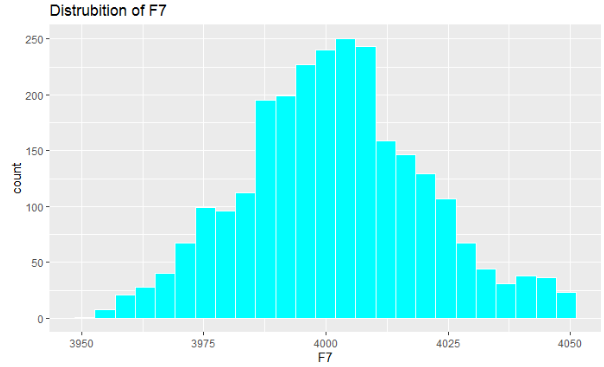
Now that we have a baseline understanding of the numerical summaries of each variable, let's start with visualization. We started the visualization by providing a simple histogram of each of the variables, in order to get a visual idea of how they are distributed. After this we will provide multivariate plots to compare variables. Recall the variables that remain after PCA are T., P7, O1, FC5, F7, FC6, F8

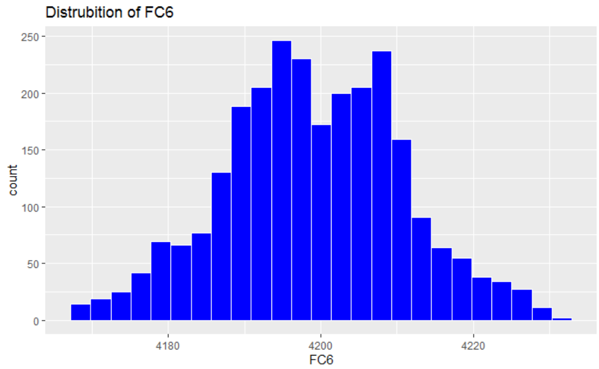

Distrubition of T.
count
0
50
100
150
200
250
4310
4320
4330
4340
4350
4360
T.

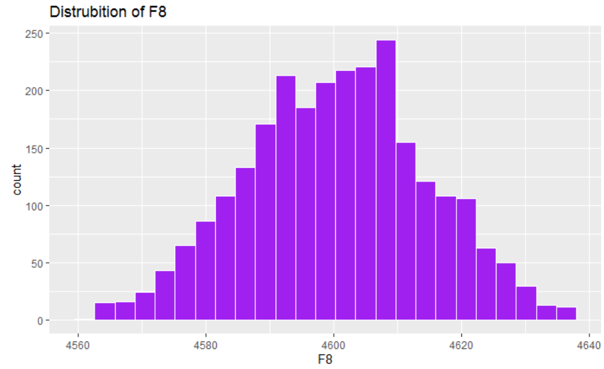

Distrubition of P7
count
0
100
200
300
4600
4610
4620
4630
4640
P7


Distrubition of O1
count
0
50
100
150
200
4020
4040
4060
4080
4100
O1

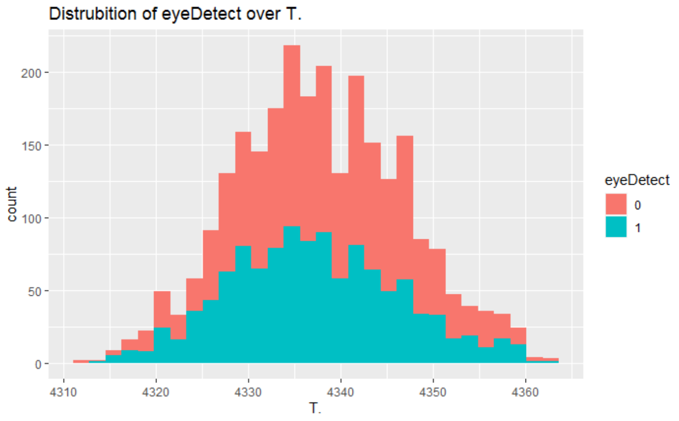

Distrubition of FC5
count
0
100
200
300
4080
4100
4120
4140
FC5

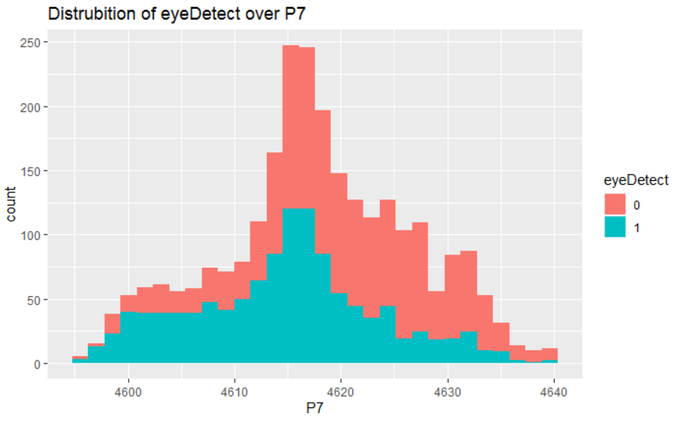

Distrubition of F7
count
0
50
100
150
200
250
3950
3975
4000
4025
4050
F7

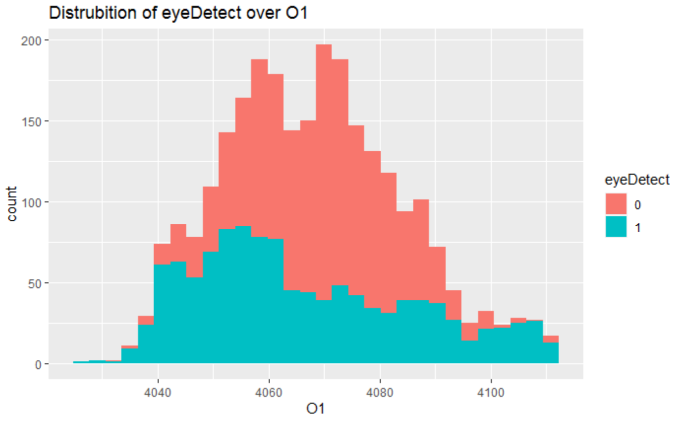

Distrubition of FC6
count
0
50
100
150
200
250
4180
4200
4220
FC6

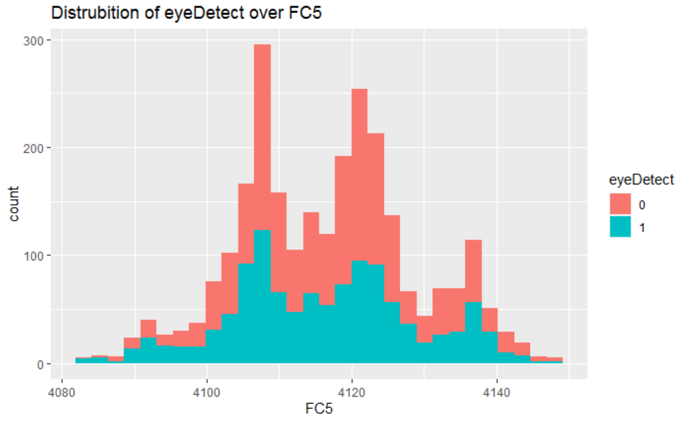

Distrubition of F8
count
0
50
100
150
200
250
4560
4580
4600
4620
4640
F8

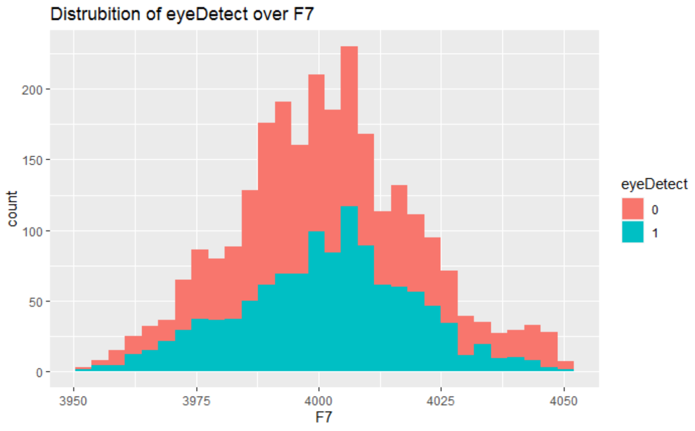
After this, we wanted to see how the rate of eyeDetection changes with the distribution of each variable. In order to do this we created multivariate histograms overlapping eyeDetect and each of the variables.

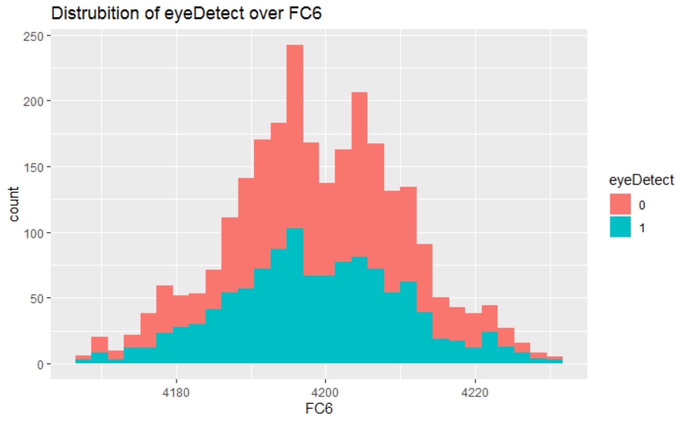


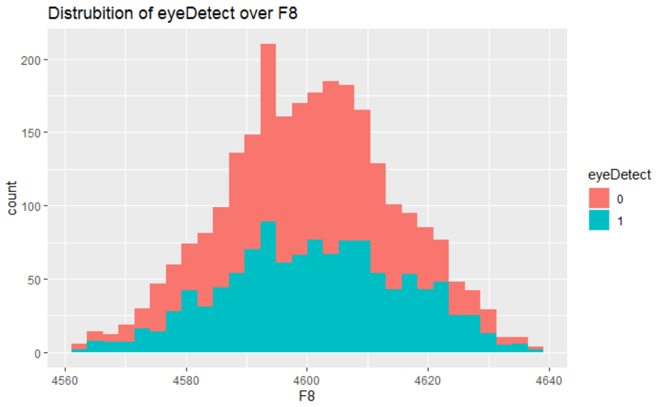




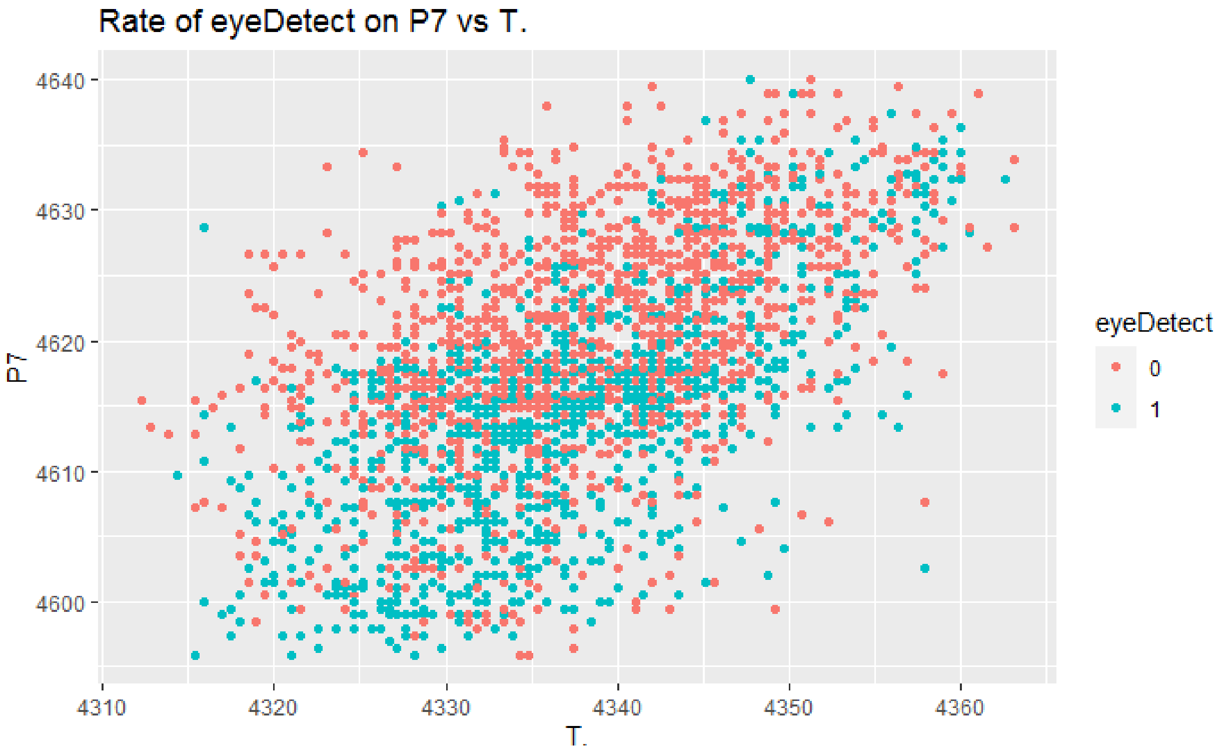






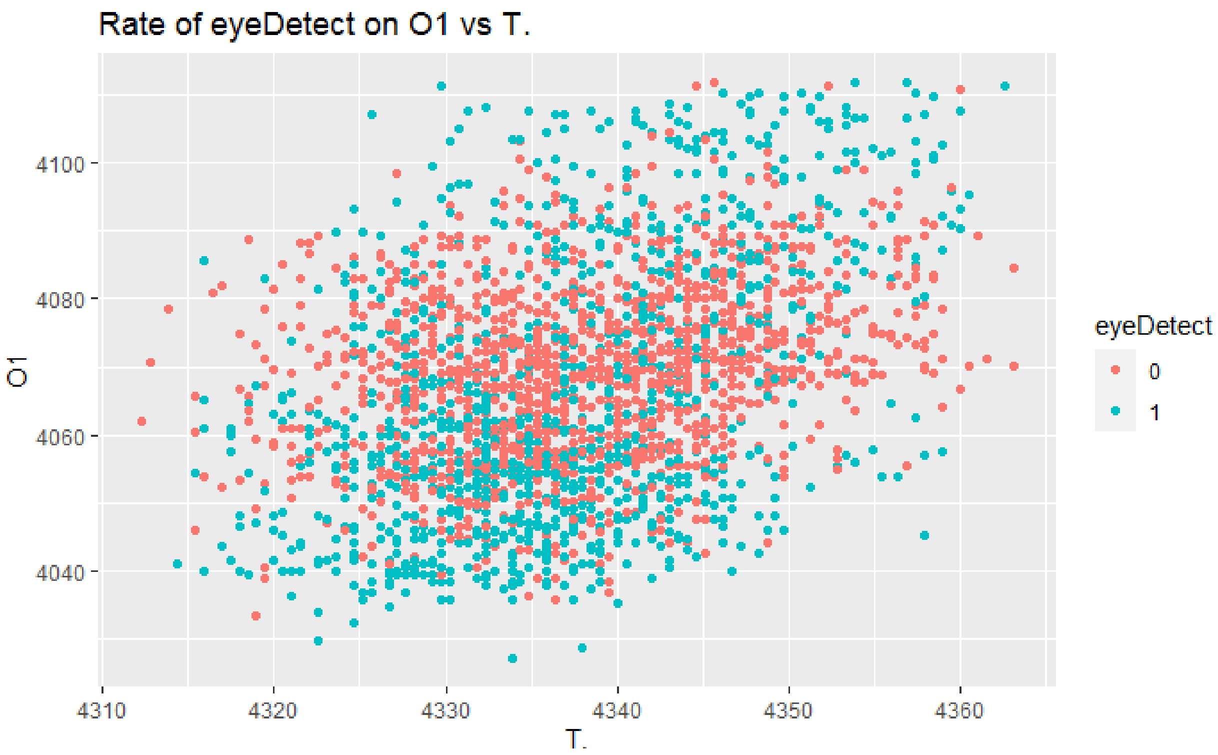


We noticed that from these layered histograms, it seems that for some variables, eyeDetect has a more clear relationship with the given variable (P7, O1). Generally, for the other variables the rate of eye detection seems to follow a relatively normal distribution. After this, we decided to explore the variables which seem to have a more clear relationship with eyeDetect further. These variables are P7 & O1. In order to do this, we created multivariate scatter plots with the variables we wanted to explore further (P7 / O1) plotted against a variable for which eyeDetect follows a normal distribution (we choose T.). To start, let's plot P7 over T., with the color of the dots on the plot dependent on eyeDetect.



From this chart we can see that as P7 increases (y axis), the rate of eyeDetect decreases (goes from blue to red). Most of the observations where eyeDetect is 1 are focused in the lower to middle values of P7.

Now lets try with O1 over T., with the color of the dots on the plot dependent on eyeDetect.



From this chart we can see the relationship between O1 (y axis) and eyeDetect. As O1 nears the lower and upper bounds of its values, eyeDetect is mostly 1 (Blue). As it nears the middle of the distribution, an eyeDetect value of 0 (red) becomes the trend. These two multivariate plots are useful in the analysis of the variables as they show that for some variables, eyeDetect has a clear relationship that becomes evident when shown visually. When working with so many variables on a vast dataset, it is hard to find clear relationships without digging further with visualization techniques, especially multivariate visualization. The goal of this section is to get a better understanding of what is happening with the data "under the hood" before we move onto the modeling and classification techniques.

Applying Classification Techniques

After we had finished pre-processing it was time to begin the modeling process. We wanted to perform 3 classification techniques to our train dataset: Logistic Regression, KNN analysis, and Decision tree.

Logistic Regression

Logistic regression is an appropriate analysis because our response variable (Eye Detection) is binary with the success being when the eye is open. When first running this model, we wanted to see if activity in certain areas of the brain would lead to one's eyes being open or closed. After our PCA analysis we determined that the brain EEG locations of T., P7, O1, FC5, F7, FC6, and F8 would be the most useful. After running our first logistic regression model with these variables, the data showed that at a 10% significance level both F7 and FC6 were not useful in predicting whether a person’s eyes were open. We then re-ran the model using only variables T., P7, O1, FC5, and F8. Below is our estimated logistic regression model.

$$log_e\left(\frac{1}{1-p}\right) = 236.087031 - \; 0.006384(FC5) + \; 0.03567(T.) - .102629(P7) + \; .005691(O1) \\ + \; .0018679(F8)$$

Model interpretation:

Beta1 = 0.006384. Thus for every 1 microvolt increase in location FC5 the log odds of the eye being open decrease by 0.006384, keeping all other variables constant.

Beta2 = 0.03567. Thus for every 1 microvolt increase in location T. the log odds of the eye being open increase by 0.03567, keeping all other variables constant.

Beta3 = -0.102629. Thus for every 1 microvolt increase in location P7 the log odds of the eye being open decrease by 0.102629, keeping all other variables constant.

Beta4 = 0.005691. Thus for every 1 microvolt increase in location O1 the log odds of the eye being open increase by 0.005691, keeping all other variables constant.

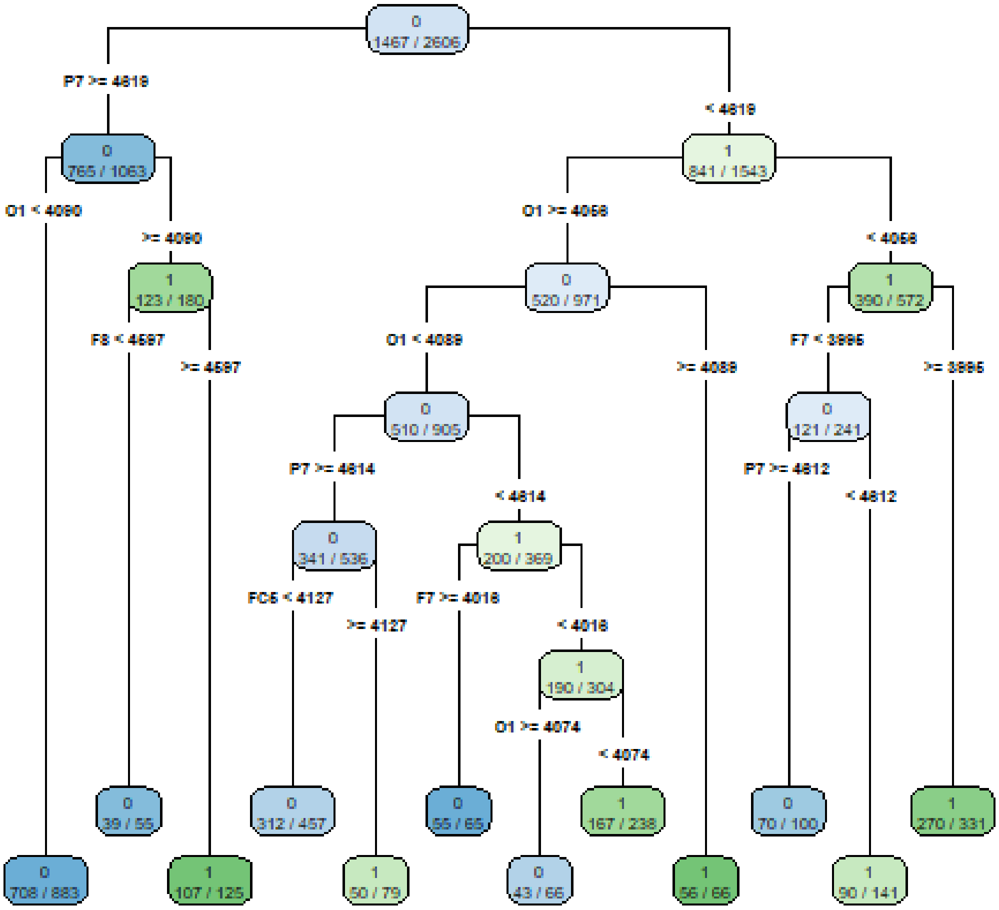
Beta5 = 0.0018679. Thus for every 1 microvolt increase in location F8 the log odds of the eye being open increase by .0018679, keeping all other variables constant.

KNN Analysis

When performing the KNN analysis we first needed to select a correct “k”. To do this I took the square root of the number of our observations and then checked numbers around that number to find which had the best accuracy. We determined that using k = 28 would create the most accurate model. Later in this document we will discuss it’s prediction accuracy further.

Decision Tree

The last classification technique used is the decision tree. We created the decision tree model using the explanatory variables of EEG locations T., P7, O1, FC5, F7, FC6, and F8. Below is the decision tree using these variables:



Predictions and Prediction Accuracy

Next we used the above models on the test dataset to predict eye openness. Below is a table comparing the three models.

Model Type	**Accuracy**	**Sensitivity**	**AUC of ROC**
Logistic Regression	62.67%	46.88%	0.6104
KNN	83.49%	71.53%	0.8160
Decision Tree	68.51%	58.68%	0.6749

Conclusion

We want to select the Model with the highest accuracy, sensitivity, and Area Under the ROC curve. Based on the table above we determined that KNN (Nearest Neighbor classification) is the best model to use when predicting eye openness from an EEG output.

References

Duffy, F. H., McAnulty, G. B., McCreary, M. C., Cuchural, G. J., & Komaroff, A. L. (2011). EEG spectral coherence data distinguish chronic fatigue syndrome patients from healthy controls and depressed patients--a case control study. *BMC neurology*, *11*, 82. Retrieved November 2nd, 2020, from https://doi.org/10.1186/1471-2377-11-82.