FINAL PROJECT

DSC 424

ADVANCED DATA ANALYSIS

EDUCATION DATASET COVID-19

ARTEMIY YALOVENKO

ALEXANDRA MISCHOU

ARELI MUÑOZ

HARNAIN KAUR SARDARNI

Overview of Data:

The COVID-19 Pandemic has had a significant impact on our lives. Especially the education system. Due to which, online platform's have become a common medium for students worldwide. This project aims to analyze the responses of a survey conducted among student's to understand their experiences and their perspectives related to online classes during pandemic.

Dataset - https://www.kaggle.com/datasets/kunal28chaturvedi/covid19-and-its-impact-on-students

It contains 1,182 rows and 19 columns, The columns include online class, self-study, fitness, sleep, social media, and TV as the continuous variables, rating of class experience and change in weight as ordinal variables, region, medium, and social media platform as categorical variables, and several binary variables, as well as two open-response variables that were manually reclassed to categories.

Non-Technical Summary of Analyses:

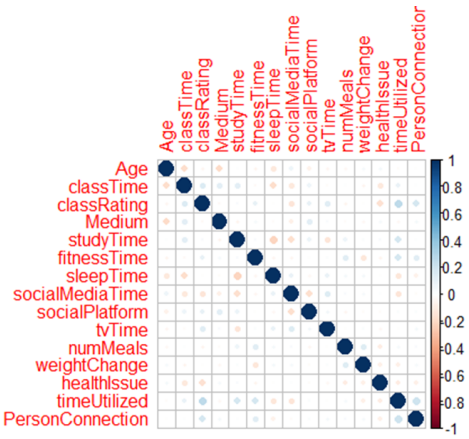
Background Analysis:

During the analysis of the “covidSurvey '' dataset, we renamed the columns, added tables for the categorical values for better understanding, and removed all the missing values. We ensured that the ordinal variables were correctly recast to numeric while retaining the response levels. We ensured all of our time variables were measured in hours, so no further transformation was needed.

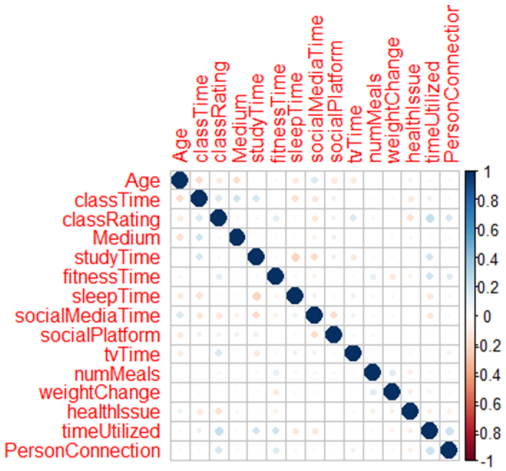
Looking at the initial histogram’s, we could see a right skew was present for many variables. We even performed visualization using scatterplot to check the trend between variables, but it was not useful as there was no clear trend in the pattern. Therefore, to combat this, we explored the option of log transforming our variables to scale our data.

While performing log transformation, we performed additional log transformations for “classTime” and “tvTime” and then calculated the skewness of those values. With that we got to know that the distribution of “classTime” is moderately left-skewed, whereas the distribution of “tvTime” is approximately symmetric. We then performed visualizations using histograms for data with log transformation. They showed a very significant right skew across the majority of the data variables. The initial correlation plots for our variables are very weak. In order to improve inter-variable correlation we performed binning, log transformation, and binning on log transformed data. Log transformed data can be seen improving correlation by a small amount while also transforming the variable skewness (as seen in table of fig.101 and correlation plots from fig.102 and 103). These notions lead us to use the log transformed data across all our analyses with exception of Bayesian Networks analysis.

```
> print(merged_df)
           variable skewness Original skewness LogTranform
1               Age                1.94252404   0.42767309
2       classRating                0.02787885  -0.29528349
3         classTime                0.36564768  -0.77718770
4       fitnessTime                0.96694383   0.06932941
5       healthIssue                2.12115616   2.12115616
6            Medium               -0.19440806  -2.27812551
7          numMeals                0.46638141  -0.60309189
8  PersonConnection               -0.88876546  -0.88876546
9         sleepTime                0.73456569   0.10902314
10  socialMediaTime                1.69714958   0.20554856
11   socialPlatform               -0.28638654  -0.39582603
12        studyTime                1.73051421  -0.25570939
13     timeUtilized                0.05755343   0.05755343
14           tvTime                2.68166174   0.53478075
15     weightChange               -0.30178296  -0.64531580
```



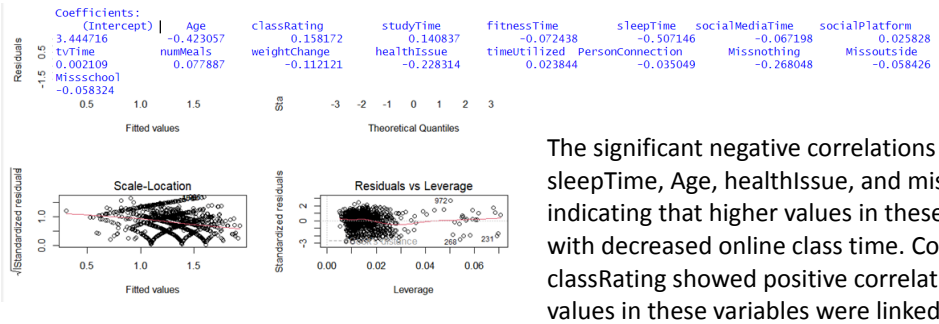
(original dataset)



(log transformed)

Linear Regression Analysis

The results of the linear regression analysis revealed several important findings. Initially, the fit of the model indicated potential violations of linearity and normality, as well as heteroscedasticity, suggesting varying residual dispersion across predictor values.



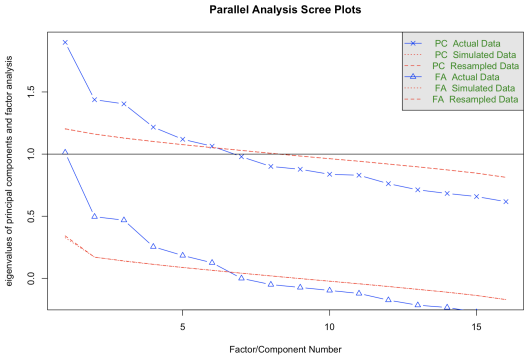
The significant negative correlations were observed for sleepTime, Age, healthIssue, and missing-nothing variables, indicating that higher values in these variables were associated with decreased online class time. Conversely, studyTime and classRating showed positive correlations, indicating that higher values in these variables were linked to increased online class attendance.

These findings were validated through cross-validation, which yielded consistent results. However, the overall accuracy of the model was found to be quite low at 10.37%, suggesting that it was not reliable. Despite attempts to address data skewness and preprocess variables, the limitations of linear regression in handling categorical and ordinal variables were evident. Therefore, it is recommended to explore alternative classification algorithms better suited for the dataset, considering the presence of such variables and the skewed nature of the data.

Polychoric PCA Factor Analysis

Principal Component Analysis is a procedure used to convert a set of observations that are possibly correlated variables into a value set of linearly uncorrelated variables. We are trying to simplify the data and to reduce noise. We need to standardize the dataset by normalizing the data. As seen in our dataset, we have a low correlation in this plot. Anything above a .60 are positively average and great to have in the majority of datasets. But in our dataset, we had positively low correlations or low negative correlations. A way to improve that; was to use a log transformation.

Principal was the next function to view the returns of a subset of the best nfactors. From the parallel analysis, factor 6 with a variance of at 69%. Each column represents a factor and the loadings method used in principal are extracting the correlations between what the input variables are and the new components.



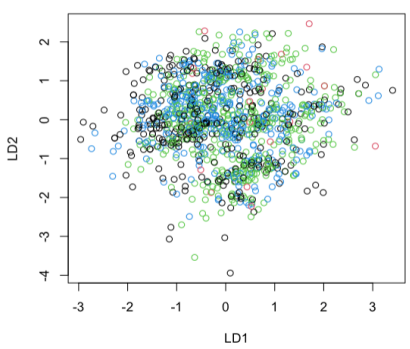
Further looking into polychoric, it suggests any stronger association and a higher variance among the factor analysis. Using the function polychor(), looked into healthIssue and Age; there is a 0.15 polychoric correlation and standard error between the two ordinal variables.

A reduced amount of variables were performed for PCA to best display these PCA relationships. I have redone the PCA for age, studyTime, fitnessTime, sleepTime and socialMediaTime. There are high PCs with studyTime being near perfect then sleepTime. There are similarities and differences in relationships by viewing the vector loadings. The variables sleepTime and Age are similar but have negative vector loadings. While fitnessTime and studyTime have positive vector loadings but are still far apart.

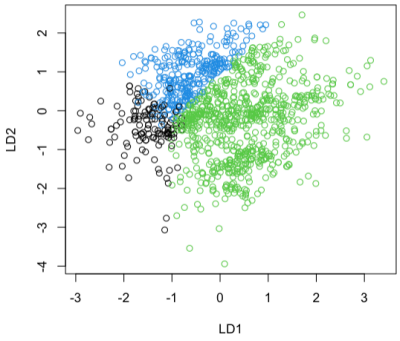
Linear Discriminant Analysis

Linear discriminant analysis (LDA) uses continuous or ordinal variables to predict a class variable. The function of LDA aims to take jumbled class data and find the directions of maximum class separability in order to accurately predict which category an instance falls into. We used the variables classRating (ordinal), studyTime, fitnessTime, sleepTime, and socialMediaTime to predict which category of activity students missed the most during

the covid19 shutdowns. A demonstration of our data separation before and after LDA is shown below. While overall accuracy was 42%, by looking at the coefficients of both discriminants we found that there is an inverse relationship between FitnessTime, socialMediaTime, and classTime with classRating, and a positive relationship between sleepTime and studyTime with class rating. In this way, while LDA was not very successful at predicting the category of activity students missed most, it helped us better understand the relationships between our predictors.



Before LDA

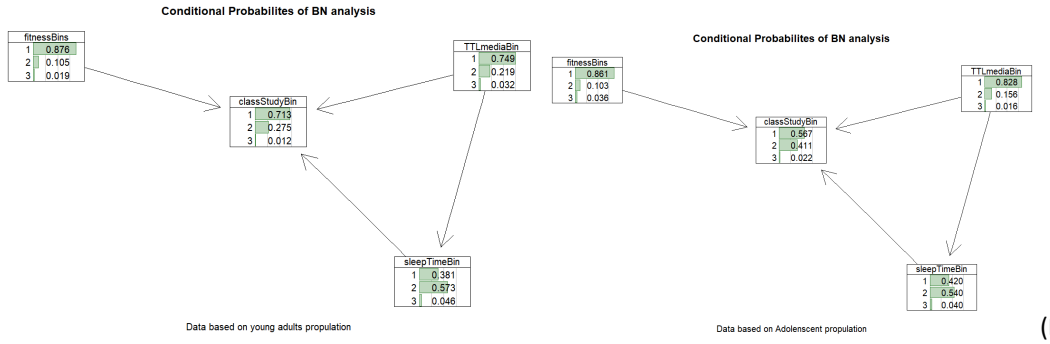


After LDA

Bayesian Network Analysis

Bayesian networks are probabilistic graphical models that use Bayesian probability theory to represent and analyze complex relationships between variables. They consist of a directed acyclic graph (DAG), where nodes represent variables and directed edges represent probabilistic dependencies. By explicitly modeling conditional probabilities and incorporating a priori knowledge, Bayesian networks enable inference and prediction under uncertainty. Networks make it easy to update probabilities based on observed data, allowing beliefs to be iteratively refined.

The developed Bayesian belief network examined the relationship between wellness behavior, commitment to class, and distraction-based behavior in age groups 7-17 and 18-27. The findings from conditional probability tables provided valuable insights. For ages 7-17, increased sleep time correlated with higher fitness levels but lower class commitment. Surprisingly, as media time decreased, teenagers were more likely to sleep less, contrary to existing beliefs. Among adults below 30, increased sleep time combined with limited social media usage led to higher fitness time, while class/study time decreased. The highest chances of spending the most time on class and fitness were observed when sleep time ranged from 4.5-7 hours and media time was 0-4.5 hours. Higher media time was associated with decreased online class attendance. Young adults were most likely to sleep between 7.8-11 hours, regardless of media time. Notably, individuals who spent the least time on social media had the highest chance of having the lowest sleep time. These insights demonstrate the intricate connections between wellness behavior, class commitment, and distraction-based behavior across different age groups, emphasizing the need for further research and expert input.



(Fig 436,Fig.437)

Technical Analysis:

Background Analysis

We conducted our final project on a dataset of student responses to a survey about habits and feelings during the covid-19 pandemic shutdown in and around the Delhi region of India. It contains 1,182 instances of 18 variables including age, time spent on: online class, self-study, fitness, sleep, social media, and TV as the continuous variables, rating of class experience and change in weight as ordinal variables, region, medium, and social media platform as categorical variables, and several binary variables, as well as two open-response variables that were manually reclassed to categories.

In our initial cleaning of the data, we renamed our columns and removed any data points with null values. We ensured ordinal variables were correctly recast to numeric while retaining the response levels. We ensured all of our time variables were measured in hours, so no further transformation was needed. Looking at the initial histograms, we could see a right skew was present for many variables (*Figures 101-111 in the appendix show this).* To combat this, we explored the option of removing all outliers by including a function to remove instances outside of each numeric variable's interquartile range. This did not improve our skew so we instead decided to include outliers and log transform our variables to both center and scale our data (*Figures 101-111 and Fig 113 show an example of histograms before and after log transformation).*

The initial correlation plots for our variables were very weak (*Figure 200).* The correlations did not improve much after log transformation. We also tried a few combinations of variables without much success in improving our variable correlations. This gives us the opportunity to explore latent variables within our data.

A significant part of the background analysis was data exploration besides assessment of intervariable correlations. We performed a number of scatter plot visualizations which allowed us to see that some variables have a relationship which may influence further analyses. For instance, higher social media times are observed in lower age groups; a similar trend to sleep, class, and study times. We can also observe that class time and Social Media time were rather evenly distributed, but still had a slightly increased concentration around lower values of both variables. We also took note of binary variables: health issue variables were greatly unbalanced while Personal Connection and Time Utilized variables were more even. (Fig 112) In case we wanted to perform certain classification algorithms it is useful to know if these variables are about even in their distributions because class imbalance may violate certain assumptions.

Additionally, frequency tables were created for categorical variables. We learned from them that the majority of people attend class via desktop, but this medium is closely followed by smartphones. We additionally can be certain that Facebook products (Instagram, Facebook, WhatsApp) are combined leaders in terms of most used media platforms with. Instagram and WhatsApp being the certain leaders, but followed closely also by YouTube. Class ratings were distributed into 5 categories which we recognized made them a good candidate for polychoric analysis due to them having a likert scale ranking. The class ratings had majority values in very poor, followed by average, followed by good, excellent and then poor. Our dataset also had a text variable $Miss that needed to be grouped by text themes since it was a "free-for-all-text-entry" within the survey. After performing the binning of the values we found that the majority of people within the dataset missed the outside (presumably because of the Covid-19 lockdown measures), yet surprisingly followed by 382 people who missed school, followed by friends and family. These relationship influences were further explored within regression, factor analysis, and linear discriminant analyses that are explained within this paper.

Linear regression and suggestion of other techniques

Ordinary Least Squares (OLS) regression is a widely used technique for estimating the relationships between variables. It provides insights into the linear relationships between the predictor variables and the target variable by minimizing the sum of squared residuals. OLS regression offers valuable information about the direction, magnitude, and significance of the relationships. However, OLS may encounter limitations when dealing with high-dimensional data or when the predictor variables are highly correlated, leading to overfitting or unstable coefficient estimates.

Looking at the initial fit of the OLS model using classTime as the response variable we can derive certain information from residual plots (Fig 123). We can see that linearity could be violated as there is a slight negative slope to the distribution of data points in the residuals vs fitted values plot. From the Normal Q-Q plot we can derive that normality could also be violated since about only half of the data points lie on the positively sloped line. From the Scale-Location plot we can say that the model's heteroscedasticity assumption is potentially violated as there is a clear systemic pattern within the plot. The plot's data is cone-shaped and also has a checkered pattern.

A checkered pattern in the Scale-Location plot typically occurs when the variability of the residuals changes systematically across the range of the predictor variable(s). It indicates heteroscedasticity, where the spread or dispersion of the residuals is not constant.

The checkered pattern typically manifests as alternating bands or clusters of points that have different spreads or variances. This can indicate that the residuals have different levels of variability for different values or groups of the predictor variable(s). In other words, the residuals may have a different spread or dispersion for different subsets of the data. This will greatly impact the reliability and accuracy of the OLS model; this notion is supported by our initial fit R^2 and coefficient results. To check for multicollinearity of the model we used Variance Inflation Factor values: $VIF_i = \frac{1}{1 - R_i^2}$. The test showed that no variables were excessively collinear with all VIF values below 2 (see fig. 125)

The results of the OLS initial fit (Figure 124) tell us that sleepTime, Age, healthIssue, and binary variable of missing-nothing had the highest significant negative correlations. Meaning that with any increase of values in these variables, time spent in the online class went down. Conversely, studyTime and classRating were the highest positive values, meaning that if the person was taking the higher-rated class and studied more there was a higher chance of the person attending the online class. We confirmed these coefficients and the significance of the variables using 10-fold cross validation using the caret library in R. The validated regression summaries as seen in figure 126 had the same values as the results in figure 124.

The OLS regression model reported a mere accuracy of 10.37% with each of 10-folds R squared ranging between 3.89% and 20.33%(Fig 124). We also know that since assumptions were violated this model can not be accurate or reliable. To check for overfitting within the OLS we computed RMSE values for each cross validated fold; the RMSEs varied between 0.52 and 0.61 (Fig 126). Additionally, when separated for a testing and training set (fig 126), RMSE for training and testing sets were 0.62879 and 0.64783 respectively. This indicates that there was no evidence of overfitting since the test set error is slightly higher than the training set error.

It is critical to note that the model does not adequately fit the data as it violates all assumptions of OLS . We have taken number preprocessing steps (see code preprocessing code p.41-47), as well as additional transformations such as binning and log transformations to address the skewness of the data model that fits adequately ourdata (fig 102-111,fig 112, fig 113). However, the bane of the problem is the fact that

linear regression algorithms do not handle categorical and ordinal variables as well as other methods. Additionally, the data we use has a significant right skew if not transformed. These factors point us that decision-tree based algorithms such as Random Forest or Gradient Boosting Trees are ought to be a better fit. Decision tree-based algorithms handle categorical and ordinal features naturally, without the need for encoding or transformation. They are robust to outliers and can effectively incorporate them without significant disruption to the model. On the contrary, Random Forest and other multiple Decision.Tree Models are rather black boxes; the complete relationship between variables cannot be explained without extensive research and heavy computation of how the algorithm discovers information gain metric.

Regularized regression was also performed through an algorithm which iteratively applies a range of alpha values for an Elastic net technique. However, the results pointed to lack of overfitting within the OLS model. Based on this notion the best analysis option chosen by the algorithm was OLS, but since we know all of assumptions are violated within this technique given this data

Polychoric PCA Factor Analysis

For our final project, we went on using the algorithms on Principal Component, Polychoric and Factor Analysis within our dataset. Below is a breakdown of each section discovered throughout the analysis.

PCA

Principal Component Analysis, measures correlation between two unobserved continuous variables and has bivariate normal distribution. Each unobserved variable, you can get from an observed ordinal variable. First step is to normalize the data so that no one variable overpowers the other with their values. We complete this step by improving the log transformation. (Fig. 200) If the variables each have their own scale then it will lead to a biased result. So, it is important to normalize the data to prevent this. Each variable is subtracted by its mean then divided by its standard deviation. The next we would need to do is find out the covariance matrix, to see how the covariance between the variables interact. The third step is to see geometrically the eigenvector and eigenvalue. The eigenvector represents a direction while the eigenvalue shows the variance in the given direction. When you see the summary of principal components, this will show the pairs you've selected to analyze; the highest eigenvalue and eigenvector is shown in the first principal component. As you go down the list, it will be the least highest eigenvalues and eigenvectors. In our project, it is important to set the dataset as numeric when setting up for PCA and having it as a data frame. Continued with the variable named ployPCA.

Principal

This returns a subset of the best nfactors. I chose to have 6 factors by the parallel analysis which shows a variance at 69%.(Fig. 201) Found something interesting when trying to rotate the data. There was no change but looked into why it may be. By rotating, this isn't the principal components with the axes associating with the eigen value decomposition by the components. It is believed, the axes aren't associated and remain the same. In Figure 202, there is no difference when rotating. RC2 = 0.610studyTime -0.698sleepTime + classTime-.408-tvTime0.404. This shows that study time and sleep time are oppositely correlated. If you want to give way by not having enough sleep; class time and tvTime are taking up that slot.

Parallel Analysis

Parallel Analysis compares the eigenvalues obtained from the correlation matrix. This technique will compare the scree factors of the observed data. Interestingly, after using the log transformation,

there was no use to scale the PCA function. The scree plot doesn't confirm the amount of factors, 6 as the variance level 1 isn't shown. Parallel Analysis confirms the amount of factors mentioned. (Fig. 203)

Polychoric Correlation

Polychoric correlation is between two observed binary variables; also known as tetrachoric correlation. It computes the polychoric correlation and its standard error between two ordinal variables. Choose heath and age to get .15 between two observed binary variables. (Fig.204) Further looking at classTime, studytime and socialMediaTime by their age. Social media time was the highest score out of the 5 tested. (Fig. 205) Also, use the hetcor() which computes the heterogenous correlation matrix between numeric variables. This gives the maximum Likelihood Estimates and it came to be Pearson correlation for this dataset. (Fig. 206)

Plot Analysis

Choosing a reduced amount of variables to best display these PCA plots. I have redone the PCA for age, studyTime, fitnessTime, sleepTime and socialMediaTime. Here are the new PCs with studyTime being near perfect in correlation then sleepTime. (Fig. 207)

Using the autoplot function this shows the PC1 and PC2 for classRating. The range is given as .75 to 1.75. From the original scale this is showing 1 through 5. PC1 on the x-axis shows a range of clustered with PC1 holding 53.55% loadings and PC2 with 30.41% loadings. (Fig. 208)

Using the biplot function, this shows in a 2D graph the points. The variables sleepTime and Age are similar but have negative vector loadings. While fitnessTime and studyTime have positive vector loadings but are far apart. This plot shows the similarities and differences between each variable. (Fig. 209) Then using the library factoextra, this shows the clear similarities and differences between age, studyTime, fitnessTime, sleepTime and socialMediaTime. (Fig. 210)

Linear Discriminant Analysis

For our final line of analysis for our final project, we conducted linear discriminant analysis (LDA). LDA uses continuous or ordinal independent variables to predict a dependent class variable. Thus, unlike principal component analysis (PCA) which aims to find the directions of maximum variance, LDA aims to find directions of maximum class separability. LDA accomplishes this by maximizing the variance between class labels while minimizing the variance within each class. Therefore the overall equation for LDA computes the scatter of the means and 'divides out' the scatter within each class to compute the matrix $S_B^{1/2} S_I^{-1} S_B^{1/2}$. We then find eigenvalues and eigenvectors of this matrix to separate our classes. This offers dimensionality reduction, but unlike PCA, is a supervised method.

There are several assumptions about the input data for LDA that we made sure were satisfied beforehand, namely: that each independent variable is normally distributed, and co-variance of the measurements are identical across different classes. These were satisfied by centering and scaling the data through log transformation as well as the removal of points lying outside the one and half standard deviations of the mean, which we determined were outliers and would skew the analysis. *Figure 301 of the appendix shows an example histogram after centering and scaling one of our predictor variables.*

A difficulty with LDA on our dataset is that before transformation, for non-numeric variables we only had binary variables or open responses to questions such as "What was your go-to stress reliever during covid-19" and "What do you miss the most while being in lockdown". I first attempted using LDA to predict classMedium, socialMediaPlatform, or Region, which were not successful because of the specificity of our sample population. Then, I binned numeric variables as a pseudo-categorical variable. For example, I transformed the numeric variable 'number of hours slept' into the three bins "Not enough" (sleep 0-5 hrs), "Average" (sleep 6-8 hrs), and "Too much" (sleep 9-12 hours). This was also wildly unsuccessful and had an accuracy rate of less than 10%. I then decided to manually reclass the open response variable "What do you miss the most while being in lockdown" ("MISS") to 4 different

categories of answers, being “Friends and family”, “Outside”, “School”, or “Nothing”. I ran the lda function from the MASS library using “MISS” as my categorical dependent variable and classTime, classRating (ordinal), studyTime, fitnessTime, sleepTime, and socialMediaTime as my continuous independent variables. This was much more successful, with the first linear discriminant proportion of trace at 77%.

As shown in *figure 302* of the appendix, the first discriminant is heavily dependent on sleepTime, classTime, and classRating. The second discriminant is dependent on classRating, fitnessTime, and sleepTime. These are understandable predictors for the outside and school classes, as well as friends and family considering the survey consisted of students who likely used to see the majority of their friends at school. We can see by looking at the coefficients of both discriminants that there is an inverse relationship between FitnessTime, socialMediaTime, and classTime with classRating, and a positive relationship between sleepTime and studyTime with class rating. In this way, LDA helped us understand the relationship of other predictors with classRating, despite it not being our target variable.

The confusion matrix shows there is still more to be desired before using this as a true predictive classification model. Namely, because the prior probability of the ‘Nothing’ category was only 2%, the model did not correctly classify any of these results. It disproportionately predicted many values as ‘outside’ as well. Overall accuracy was 42%, and I believe this is largely due to my manual reclassing of fifty variables to four categories. To improve, a future project could perform cluster analysis on the response values to get a better idea of the amount and type of categories to reclass the ‘MISS’ variable to. *Figures 303 and 304 of the appendix show the linear discriminant projection and confusion matrices, respectively.*

A 3d MDS plot shows the misclassification in further detail. MDS plots use a distance matrix to represent the relationship between objects. In the plots, the points are arranged so that the distance between each pair correlates as best as possible to the dissimilarity between them. The values of the points themselves does not mean anything. Many groups had similar means *shown in Figure 305 of the appendix*, meaning the model had a tougher time accurately predicting class, which can be seen as we rotate the plots in *Figure 306 of the appendix.*

Extra credit:

Bayesian Network Analysis

Bayesian networks (BN), or Bayesian Belief Network, are graphical models used in both supervised and unsupervised machine learning. They capture probabilistic relationships among variables and are commonly employed for modeling complex systems with uncertainty. Bayesian networks find application in various domains such as healthcare, finance, natural language processing, and computer vision, aiding in decision-making, risk assessment, pattern recognition, and prediction tasks.

Bayesian Network is represented by a mathematical object called a network structure, denoted as G = (V, A). V refers to the set of nodes, such as v1, v2, ..., vN, where N is the total number of nodes in the network. A represents the set of edges or arcs, which depict connections between nodes. The graph is uniquely defined by the combination of V and A. In a directed acyclic graph (DAG), the edges are directed from one node to another, denoted as, $(v_i, v_j) \neq (v_j, v_i), v_i \rightarrow v_j,$ with the constraint that each pair of nodes is connected by only one edge.

In a DAG, vi is referred to as the "parent" node, and vj as the "child" node. DAGs are acyclic, meaning they do not contain any loops or cycles. Loops refer to edges from a node to itself (vi → vi), while cycles represent sequences of edges that form closed loops, such as vi → vj → ... → vk → vi. The DAG plays a crucial role in a BN by expressing the relationships between variables in terms of conditional independence. It represents the dependencies and independence assumptions among variables, where variables do not directly cause each other.

Therefore, the DAG structure in a BN allows for the modeling of conditional independence relationships and aids in understanding the causal dependencies and dependencies between variables within the network.

In a Bayesian network (BN), if two nodes are not connected by an edge, they are either independent or conditionally independent given some other nodes. This property, known as the local Markov property, states that the absence of a direct edge implies independence or conditional independence. The global Markov property is a stronger version, extending the local property to subsets of variables, where any two subsets are conditionally independent given a separating subset. The graphical separation in a BN implies probabilistic independence, represented by the formula $v_i \perp\!\!\!\perp_G v_j \,|\, v_k \Rightarrow v_i \perp\!\!\!\perp_P v_j \,|\, v_k$ where $\mathbb{H}_G$ means graphical separation and $\perp\!\!\!\perp_P$ probabilistic independence. Hence, a BN's directed acyclic graph (DAG) serves as an independence map, illustrating the conditional independence relationships between nodes. *(for further explanation of how Markov Property allows for decomposition of a large model into smaller subsets given absence of cycles see the cited Briganti page 3-4).*

Assumptions:

When using structure learning algorithms for Bayesian networks, certain key assumptions are made:

- The underlying structure being learned is a directed acyclic graph (DAG).
- There are no selection biases, latent variables, or confounding variables. All the common causes of measured variables are accounted for, known as causal sufficiency.
- Only variables that are d-separated in a DAG exhibit independence, while others are dependent, following the principle of causal faithfulness, which is the converse of the Markov property.
- Bayesian networks encode only conditional independencies, representing the sole type of relationships between random variables in X.
- Each node in the network represents a distinct random variable in X, ensuring that there are no multiple nodes that serve as deterministic functions of the same random variable, like a sum score of two variables.
- Observations should be independent realizations, allowing for unbiased estimation of probabilities.

- The global probability distribution P(X) must have strictly positive values, ensuring that all possible combinations of variable values in X represent observable events. This requirement guarantees the presence of uniquely determined Markov blankets and an identifiable model structure.

Note on the **goal of the analysis** and pre-processing :

Our goal of the analysis was to analyze the relationship between wellness/healthy behavior (sleeping and fitness variable related), commitment to class (class time and study time variables), and distraction-based behavior (Time spent on Social Media and TV) between age groups 7-17 and 18-27.
To achieve this we used a reduced number of variables(that the original 16 variable model) which were also merged to assess our objective question. Variables of time spent on class attendance and studying time were binned to represent the overall commitment to education during Covid lockdown. Variables of time spent on tv and on social media were binned to represent the overall destructive behavior via entertainment.The variables had to be binned and then casted as numeric factors.
The factor distributions were the following (age in years, while sleep time, fitness, media time, and class time were in hours).

```
> summary(CovNumBin)
ageBins      sleepTimeBin   fitnessBins    TTLmediaBin    classStudyBin
7-17 :302    4-7.5 :484     0-1:1035       0-4.75 :914    0-7.3 :799
18-27:790    7.8-11:645     2  : 118       5.0-9.5:233    7.5-14:364
             12-15 : 50     4-5:  26       10-14.5: 32    15-22 : 16
```

It could be said that that increasing bin values within sleep time, fitness, and class time are beneficial to wellness while increasing time spent on media does reverse according to study by Walsh of MIT. (Walsh)

Discovering the DAG and measuring node stability:

Briganti of Harvard University writes:
“For cross-sectional studies, it is desirable to study a "stable" network structure, that is, a set of edges and directions that is unlikely to vary. [...] For partial correlation networks, one can use bootstrapping methods to evaluate the stability of network estimates (Epskamp et al., 2018). It is possible to directly account for stability in structure learning in a similar way: the same algorithm learns a sufficiently large number of BNs from bootstrap samples and we only consider edges that appear in a proportion of BNs higher than some threshold. Previous empirical papers (Briganti, Scutari, et al., 2020, 2021) only included edges that appeared in more than 85% of networks (this is called strength), and whose direction appeared in more than 50% of networks (this is called minimum direction). We recommend that researchers only report stable BNs obtained in this way, and that they should use 100-200 bootstrap samples to ensure the proportion of BNs in which each edge appears are estimated accurately.” (Briganti, 7)

Network resulting using The IC algorithm which is implemented by the Peter & Clark (PC) algorithm described by Briganti above:

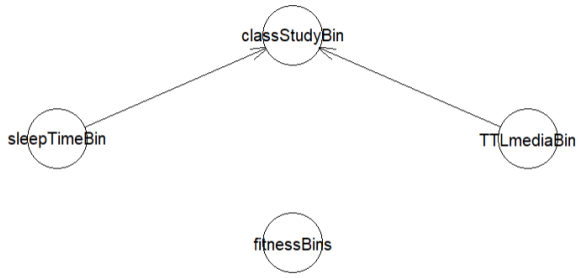


Fig.433

In addition to using the method described by Brigatti for ensuring goodness of fit we leverages a few other techniques for the assessment:

The Hill-Climbing (HC) algorithm, as described, is a greedy search approach for exploring Directed Acyclic Graphs (DAGs) by making single-edge additions, removals, and reversals. (Britanti, 9)

Network resulting using Hill Climbing Algorithm:

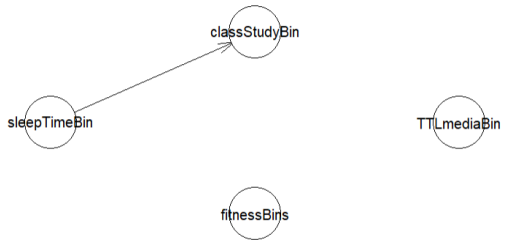


Fig.432

Network resulting using Incremental Association:

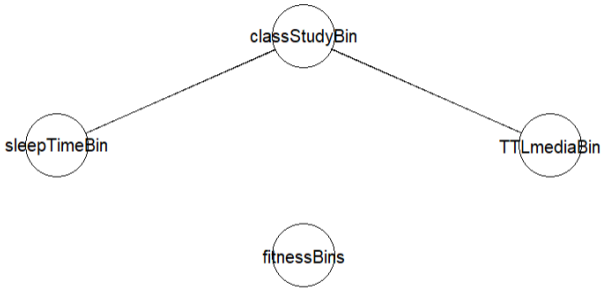


fig.434

According to Briganti: *"A comprehensive simulation study showed that there were no systematic differences in performance or sensitivity to error in real-world data when comparing the three classes of learning algorithms (Scutari et al., 2018)"(Briganti, 10)*

However, even though the stability algorithms pointed towards eliminating the assessment of the fitness variable, it was included in the final model. We leveraged the … as the backbone to support our inclusion of fitness into the model as the multitude of studies concluded: *"The strongest relationships have been found between aerobic fitness and performance in mathematics, reading, and English. For children in a school setting, regular participation in physical activity is particularly beneficial with respect to tasks that require working memory and problem solving. These findings are corroborated by the results of both authentic correlational studies and experimental randomized controlled trials." (Kohl III, Cook).*
Additionally, supported by studies published within National Library of Medicine regarding and concluding an overall negative effect of sleep within adolescents, young adult, and college students, we created a relationship between Media time and Sleep time (Garett, Liu,Young) (Levenson,Shensa, Sidani ,Colditz , Primack) (Pirdehghan, Khezmeh, Panahi).

Network that was analyzed using Bayesian Belief network:

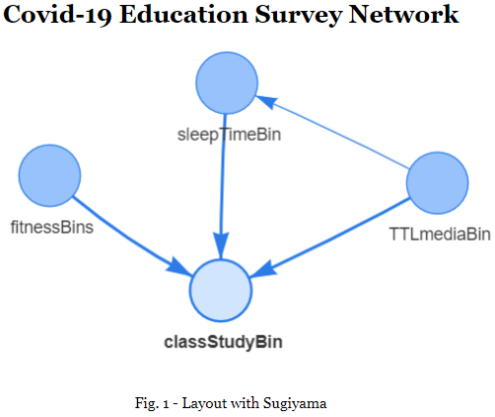


Fig.435

Evaluation of Results:

As mentioned earlier, the goal of the developed bayesian belief network is to assess the relationship of wellness behavior (sleeping and fitness related), commitment to class (class time and study time), and distraction-based behavior (Time spent on Social Media and TV) between age groups 7-17 and 18-27. Figure 321 provides conditional probability tables based on multinomial distribution for assessing the behavioral factors for ages of 7-17. Figure 322 provides conditional probability tables for age group of 18-27.

Looking at results in Fig.321 we can derive the following insights:

- As sleep time increases the chances of increased fitness given any level of class commitment time generally increase for the age group of 7-17. At the same time as sleep time increases the chance of higher class commitment given any level of fitness lowers. Low sleep and increasing media time had mixed results due to progressively decreasing observations within high class attendance and high fitness time groups.
- From parameters of node sleepTimeBin, it is statistically more likely that teenagers sleep between 7.5-14 hours. Interestingly there is a sign that as media time decreases there is a higher chance of sleeping less among the teenage population. This is contrary to the general belief and contrary to the conclusion in studies such as one done by Pirdehghan on "Social Media Use and Sleep Disturbance among Adolescents". Further research and assistance of subject-matter experts is required.

Looking at Fig.322 we can derive the following insights:

- Assessing the parameters of node explaining class/study time we can derive that as as sleep increased among adults before 30, given that social media time stayed at 0-4.75 hours a week level, time spent on fitness progressively increased. Meanwhile, as sleep time increased, time spent on the class decreased. Within this age group, the highest cumulative chance of a person spending most time on class (between 7.5-14 hours) and most time on fitness (between 4-5 hours) was given that the person spends between 4.5-7 hours on sleep and 0-4.5 hours on media. We can also derive that as time on media increased chances of spending more time on the online class decreased. While those that spent a small amount sleeping and most amount on media, as well as those that spend most amount of time sleeping and on social media were guaranteed to be in the lowest groups in class and fitness attendance.
- Looking at the parameters of node explaining time spent on sleep we can derive that given any media time young adults were most likely to sleep between 7.8 - 11 hours. Interestingly, highest chance of lowest sleep time was given when persons spent least time on social media (40.2%); this could potentially indicate that they could've been occupied with other tasks or suffered a health issue.

Bayesian Networks citations:

- *Briganti, Giovanni, et al. "A Tutorial on Bayesian Networks for Psychopathology Researchers." Psychological Methods, 2022, https://doi.org/10.1037/met0000479.*
- *Levenson, Jessica C., et al. "The Association between Social Media Use and Sleep Disturbance among Young Adults." Preventive Medicine, vol. 85, 2016, pp. 36–41, https://doi.org/10.1016/j.ypmed.2016.01.001.*
- *Garett, Renee, et al. "The Relationship between Social Media Use and Sleep Quality among Undergraduate Students." Information, Communication & Society, vol. 21, no. 2, 2016, pp. 163–173, https://doi.org/10.1080/1369118x.2016.1266374.*
- *Pirdehghan, A., et al. "[PDF] Social Media Use and Sleep Disturbance among Adolescents: A Cross-Sectional Study: Semantic Scholar." Iranian Journal of Psychiatry, 1 Jan. 1970, www.semanticscholar.org/paper/Social-Media-Use-and-Sleep-Disturbance-among-A-Pirdehghan-Khezmeh/96b2850cb488cfb0164825f0577125d2538271ff.*
- *Yuan, Ying, and Valen E Johnson. "Goodness-of-Fit Diagnostics for Bayesian Hierarchical Models." Biometrics, Mar. 2012, www.ncbi.nlm.nih.gov/pmc/articles/PMC3276744/.*
- *Hackerman, David. At - University of Pennsylvania, www.cis.upenn.edu/~mkearns/papers/barbados/heckerman.pdf. Accessed 8 June 2023.*
- *Murphy, Kevin. "A Brief Introduction to Graphical Models and Bayesian Networks." Graphical Models, 1998, www.cs.ubc.ca/~murphyk/Bayes/bnintro.html.*
- *Roweis, Sam. A Unifying Review of Linear Gaussian Models - New York University, cs.nyu.edu/~roweis/papers/NC110201.pdf. Accessed 8 June 2023.*
- *Walsh, Dylan. "Study: Social Media Use Linked to Decline in Mental Health." MIT Sloan, 14 Sept. 2022,mitsloan.mit.edu/ideas-made-to-matter/study-social-media-use-linked-to-decline-mental-health*
- *Goodyear, Victoria A., et al. "The Effect of Social Media Interventions on Physical Activity and Dietary Behaviours in Young People and Adults: A Systematic Review - International Journal of Behavioral Nutrition and Physical Activity." BioMed Central, 5 June 2021, ijbnpa.biomedcentral.com/articles/10.1186/s12966-021-01138-3.*
- *Rohmer, Jeremy. "Uncertainties in Conditional Probability Tables of Discrete Bayesian Belief Networks: A Comprehensive Review." Engineering Applications of Artificial Intelligence, vol. 88, 2020, p. 103384, https://doi.org/10.1016/j.engappai.2019.103384.*
- *Kohl III , Harold W., and Heather D. Cook, editors. "Educating the Student Body: Taking Physical Activity and Physical Education to School." Committee on Physical Activity and Physical Education in the School Environment; Food and Nutrition Board; Institute of Medicine, 2013, https://doi.org/10.17226/18314.*
- *Rohmer, Jeremy. "Uncertainties in Conditional Probability Tables of Discrete Bayesian Belief Networks: A Comprehensive Review." Engineering Applications of Artificial Intelligence, vol. 88, 2020, p. 103384, https://doi.org/10.1016/j.engappai.2019.103384.*

Individual Contributions - Appendix:

Alexandra Mischou - data cleaning, background before transformations, LDA

I was tasked with the initial cleaning of the data, for this section I renamed our columns and reclassed categorical variables to numeric so we could use them in further analysis. I replaced null values with either the median or mode for their respective category. I ensured ordinal variables were correctly recast to numeric while retaining the response levels, specifically for the class rating variable. I created the initial histograms and tables to look at the distribution of the data before transformation. I also created an initial correlation plot before any further transformations were done.

For my line of analysis for the final project, I conducted linear discriminant analysis (LDA). There are several assumptions about the input data for LDA that I made sure were satisfied beforehand, namely: that each independent variable is normally distributed, and co-variance of the measurements are identical across different classes. These were primarily satisfied by centering and scaling the data through log transformation.

A difficulty with LDA on our dataset was that before transformation, for non-numeric variables we only had binary variables or open responses to questions such as “What was your go-to stress reliever during covid-19” and “What do you miss the most while being in lockdown”. I tried LDA using our categorical variables as the target class first, namely, class medium, social media platform, and region. These were not successful because the sample was very cohesive and thus the class means were way too similar, even after transformation. I then attempted LDA using binned numeric variables as a pseudo-categorical variable. For example, I transformed the numeric variable ‘number of hours slept’ into the three bins “Not enough” (sleep 0-5 hrs), “Average” (sleep 6-8 hrs), and “Too much” (sleep 9-12 hours). This was also wildly unsuccessful and had an accuracy rate of less than 10%. I then decided to manually reclass the open response variable “What do you miss the most while being in lockdown” (“MISS”) to 4 different categories of answers, being “Friends and family”, “Outside”, “School”, or “Nothing”. With initially over 50 different responses to this question, it was pretty tedious, however, it helped build a LDA with 32% improved accuracy. While the LDA was not good enough as a predictive classification model, it did help me better understand some of the relationships between predictors that I couldn’t see as easily in the initial plots.

I made the powerpoint slides and wrote the non-technical and technical summaries for the sections above. I included MDS plots in the analysis per the feedback given on the presentation.

Artemiy Yalovenko- Data transformation, Linear Regression, Bayesian Networks

This report presents a comprehensive analysis of the COVID-19 Impact Survey data, focusing on factors influencing sleep time, study time, social media time, and efficient time utilization during the pandemic. The analysis includes data preprocessing, exploratory data analysis, and modeling techniques such as descriptive statistics, correlation analysis, factor analysis, and regression analysis. Additionally, an extra credit initiative explores Bayesian Belief Networks, a model with interpretable inter-variable inferences. The findings shed light on the relationships between variables and provide insights into coping mechanisms reported by survey respondents.

Data Preprocessing:

The initial steps I was directly involved in were data preprocessing, including handling missing values and transforming variables. Descriptive statistics and visualizations are used to understand variable distributions and characteristics. Log transformations are applied to address skewed distributions, but no significant improvement in inter variable correlations is observed. Binning variables are also attempted to address correlations, but no significant impact is achieved. Granted, Alex did the absolute most of the data cleaning.

Exploratory Data Analysis:

Correlation analysis is conducted to examine relationships between variables. Heatmaps and correlation matrices are visualized to identify strong and weak correlations. Factor analysis is employed to assess variable contributions to overall variance. Scree plots and parallel analysis aid in determining the appropriate number of factors. Within the final document, Alex and I wrote the entirety of the background analysis and provided additional visualizations on extremely short notice. Areli, Alex, I had to set up Appendices in a similar manner.

Regression Modeling:

Ordinary Least Squares (OLS) regression is employed to predict class time based on predictor variables. Model performance, overfitting, and multicollinearity are evaluated using summary statistics and variance inflation factors. Elastic Net regression is attempted, but results show poor accuracy due to the dataset's characteristics. Most importantly, due to the violation of all assumptions I assessed the decision tree-based models such as random forest would be a much better fit for our low correlate data particularly when using a balanced binary response variable such as Personal Connection or Time Utilized.

Extra Credit: Bayesian Network analysis

An extra credit initiative explores the technique of the Bayesian Network in examining the relationship between wellness/healthy behavior, commitment to class, and distraction-based behavior between the age groups of 7-17 and 18-27 within our datasets. It is important to note that the initial study was done using log-transformed variables and using Gaussian Distribution assumption. The initial attempts did provide useful information and it was thus included in the video presentation. However, after discussions with the group, we concluded that it is drawing more specific inferences is required if we were to hypothetically use this model in the real world. An additional reason for this was the assessment of the network with multiple validation algorithms which pointed to the multidirectional relationship of certain important arcs within the first model. The model presented in the analysis is backed by 3 different validation algorithms as well as research papers that provide expert knowledge which is so needed in Bayesian networks.

Conclusion:

The analysis of the COVID-19 Impact Survey data provides valuable insights into time utilization factors during the pandemic. The correlations and factors identified contribute to a deeper understanding of the dataset. Regression models, although limited in accuracy, offer initial predictions. The extra credit initiative further explores a relationship between binned ordinal data providing an interesting insight into the tendencies of wellness and distractions of the young population during Covid-19! Overall, this analysis enhances our understanding of student behavior during a challenging period and opens avenues for further research in this area.

Areli Muñoz - Data transformation, Visuals, PCA, Principal, Polychoric Analysis

For our education dataset, our team took on the Covid-19 Survey dataset. I played a role to help organize the team. By communicating with each member to secure our meetings, send out zoom invites, send out emails, keep track of meeting notes and keep track of the milestones deadlines. With each meeting we came out with a game plan of work to do based on our strengths and have a set of tasks for each teammate. Worked on the outline of the final presentation and created a layout to where we can present top down analysis.

I helped to further clean up the explanatory analysis section with Art after Alex's first go at it. There was little to no correlation to the variables in the dataset. To understand the different relationships in pairs of the covid-19 survey. I did ggplots of a variety of variables I thought may be interesting to look further. Especially in the visuals, there is something interesting about their age and if they had health issues. The number 0 is no and 1 is yes; the ages up to 40, only 19 plotted data. Majority are plotted in the 20s – mid20s

– mid 30s and the max 40s with yes they had health issues. One would think that the people over the age of 40 would have more health problems but covid affected the younger generations health.
Supported Art with removing the outliers and elastic net. This was needed to clean up the data because if not, the scaling and moving forward with the dataset for other functions will become inaccurate.

Worked with Harnain on the tasks for Polychoric PCA factor analysis. Figured out how to use the log transform that Art worked on into our PCA analysis. Realized after working with PCA functions, the log transformed dataset wasn't being used. Ran the principal function with and without rotation; noticed there was no difference when applying the rotation.

I took the PCA/Polychoric further and looked into why the rotation wasn't applying to further the team's analysis point. Also, looked into detail to understand PCA, Principal and Polychoric Correlation algorithms; to understand their purpose. PCA measures correlation between two unobserved continuous variables, have bivariate normal distribution. Principal returns a subset of the best nfactors. Polychoric Correlations is between two observed binary variables. And Plots to best describe the relationships. Learned new libraries such as ggfortify to use autoplot(), fviz_pca_var() and biplot().

In this paper, I worked on the nontechnical and technical portions of PCA Factor Analysis, Principal and Polychoric Correlation. Also worked on organizing the paper format.

HARNAIN KAUR SARDARNI - PCA, BACKGROUND ANALYSIS PPT

For the project, Areli and I worked on PCA. We performed polychoricPCA analysis with the log transformation performed by Art. We started by converting the log dataset to a numeric format, then performed the polychoric correlation matrix using the hector(). Then captured the correlations between the variables in the matrix. We then performed PCA on the dataset using the principal function. Then created two PCA models, one without rotation and one with rotation using the varimax method. Performing the analysis we realized that there was no difference when applying the rotation. The purpose here to perform the analysis was to identify the underlying factors and to understand the relationship between the variables.

I then worked on the Powerpoint of background analysis explaining what all steps were taken to get to the further analysis, renaming columns, using unclass() to convert categorical variables to numerical and then even working on missing values. Then looked into the histograms and scatter plot for a better understanding of the data, then to look into the relation between variable's performed visualization using correlation. Then we performed variable transformation using log to see if it could improve the correlation. In this paper, I worked on the introduction and non - technical analysis.

Appendix-Technical Summary

Background Pre-processing and analysis

Fig.101 (skewness before and after log transformation using skewness())

```
> print(merged_df)
          variable skewness Original skewness LogTranform
1              Age        1.94252404          0.42767309
2      classRating        0.02787885         -0.29528349
3        classTime        0.36564768         -0.77718770
4      fitnessTime        0.96694383          0.06932941
5      healthIssue        2.12115616          2.12115616
6           Medium       -0.19440806         -2.27812551
7         numMeals        0.46638141         -0.60309189
8  PersonConnection       -0.88876546         -0.88876546
9        sleepTime        0.73456569          0.10902314
10 socialMediaTime        1.69714958          0.20554856
11  socialPlatform       -0.28638654         -0.39582603
12       studyTime        1.73051421         -0.25570939
13    timeUtilized        0.05755343          0.05755343
14          tvTime        2.68166174          0.53478075
15    weightChange       -0.30178296         -0.64531580
```

Histograms (before transformations)

Fig 102

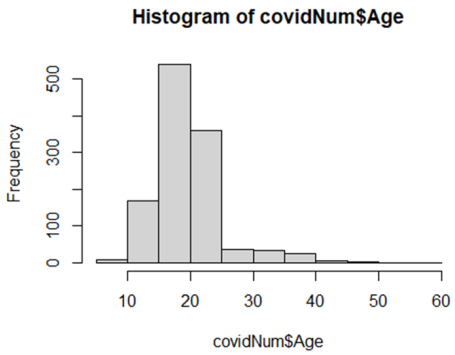


Fig 103

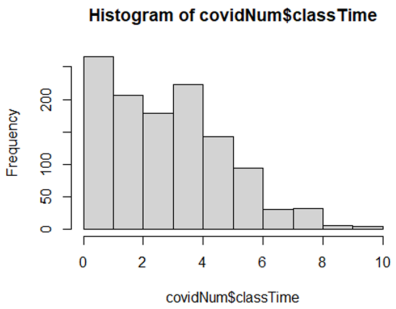


Fig 104

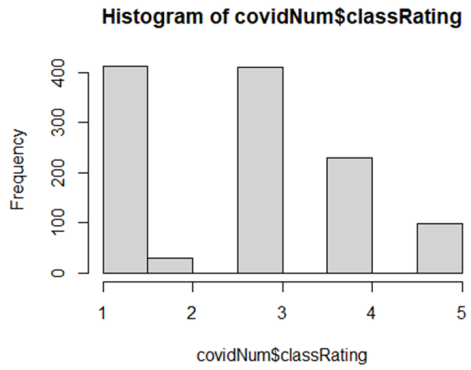


Fig 105

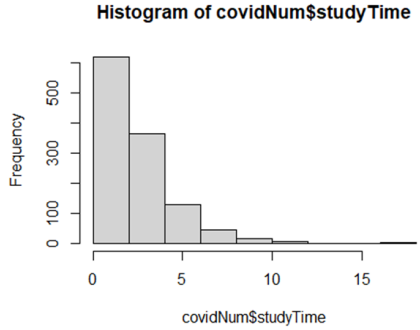


Fig 106

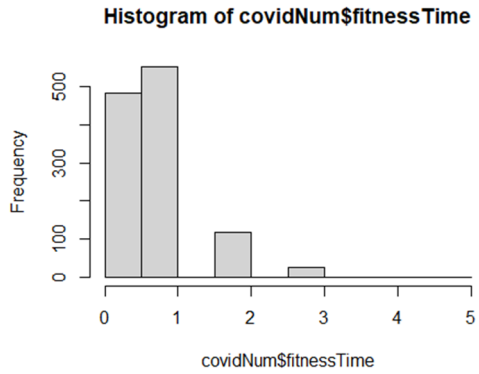


Fig 107

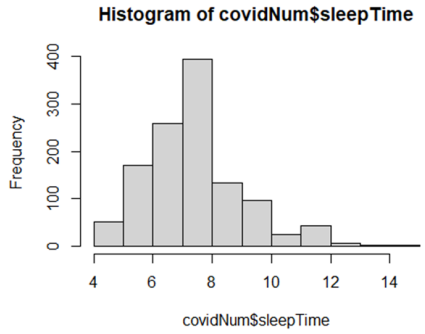


Fig 108

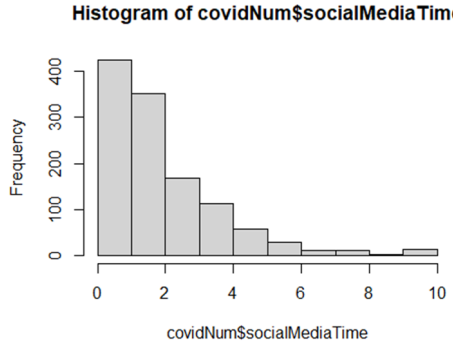


Fig 109

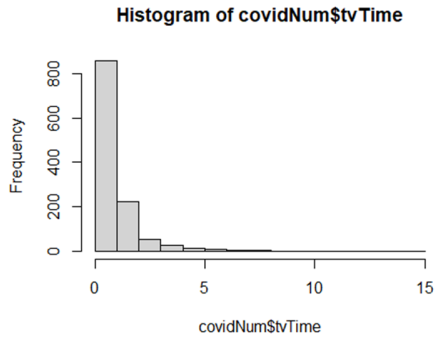


Fig 110:

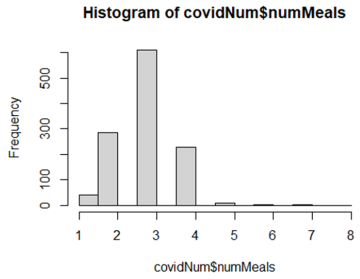


Fig 111.

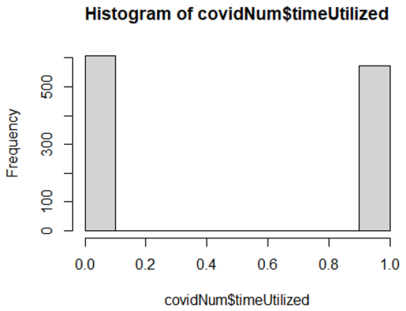
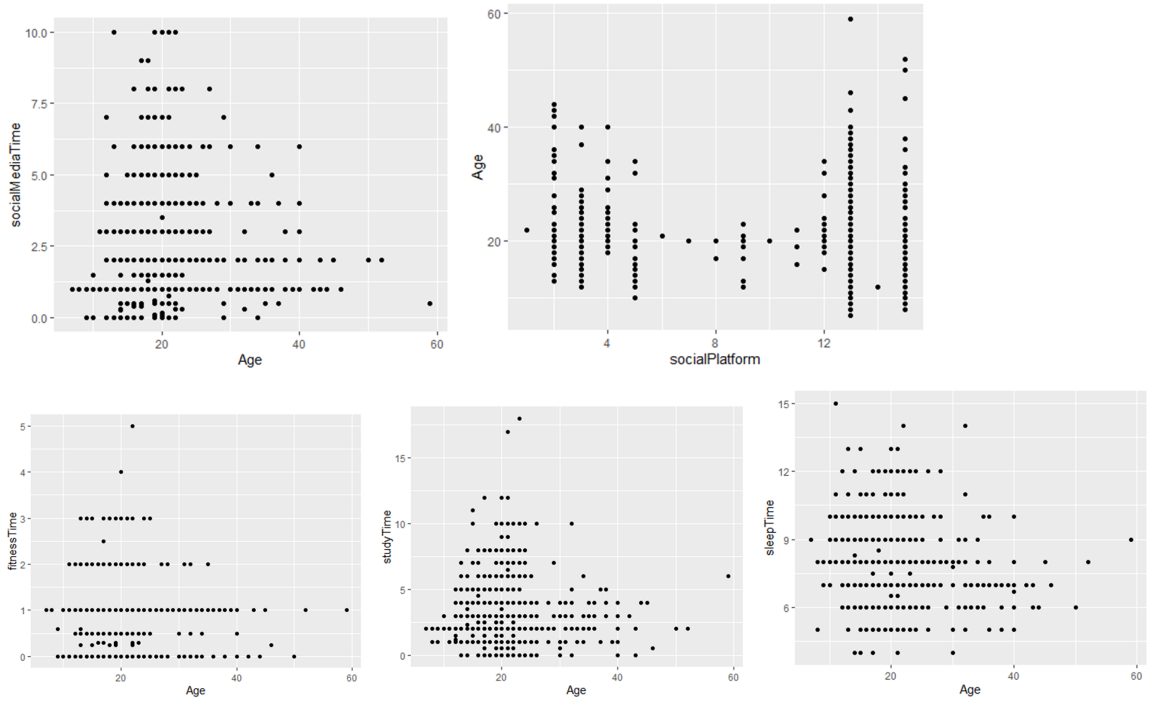
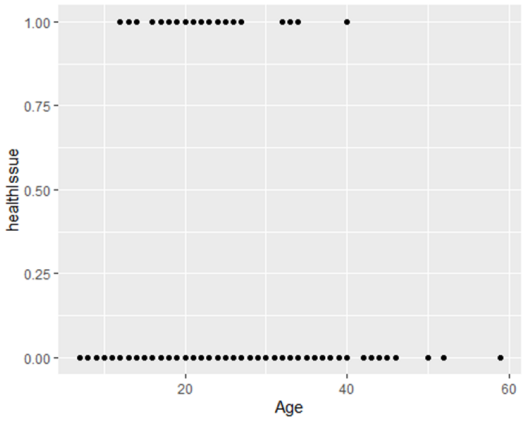
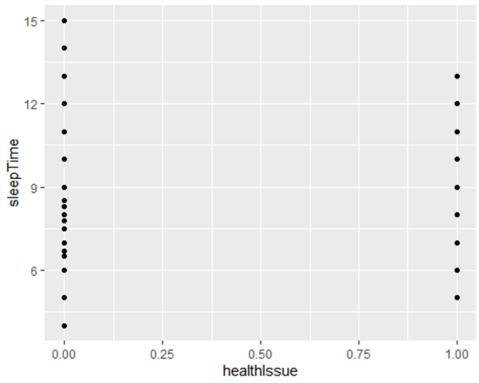
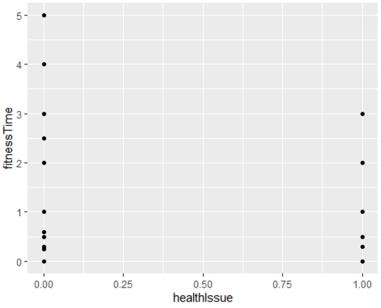


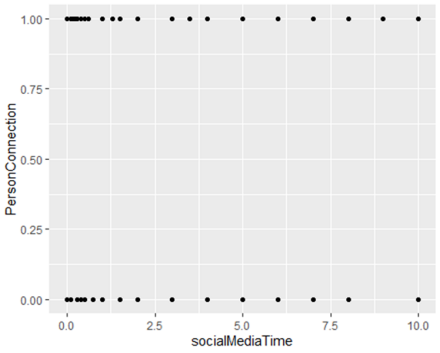
Fig 112 (visualizations of original dataset scatterplots):

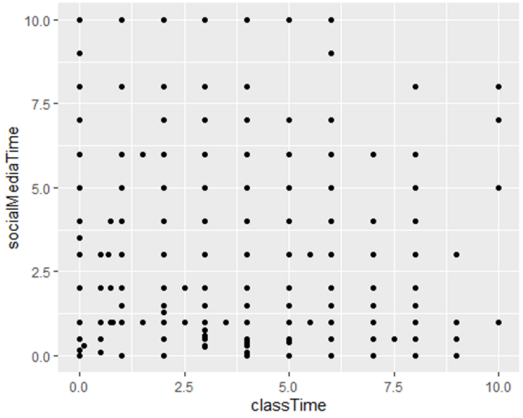


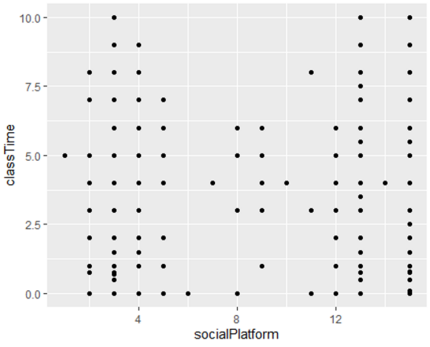

healthIssue
Age

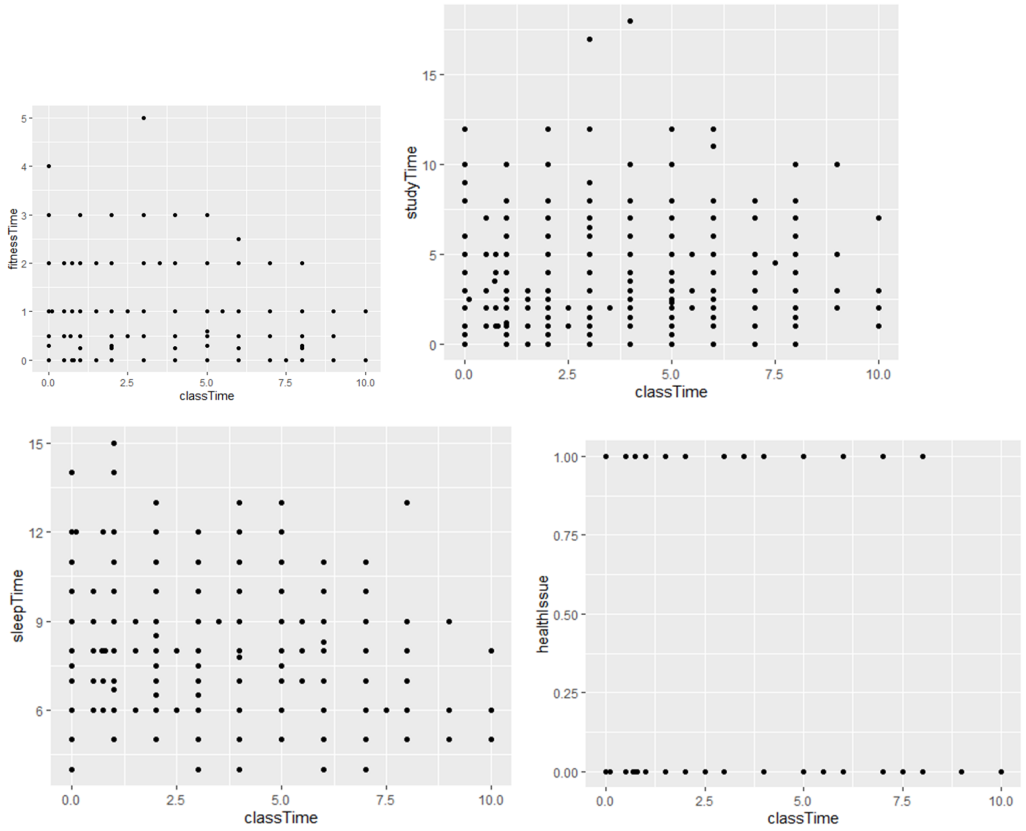

sleepTime
healthIssue


fitnessTime
healthIssue


PersonConnection
socialMediaTime

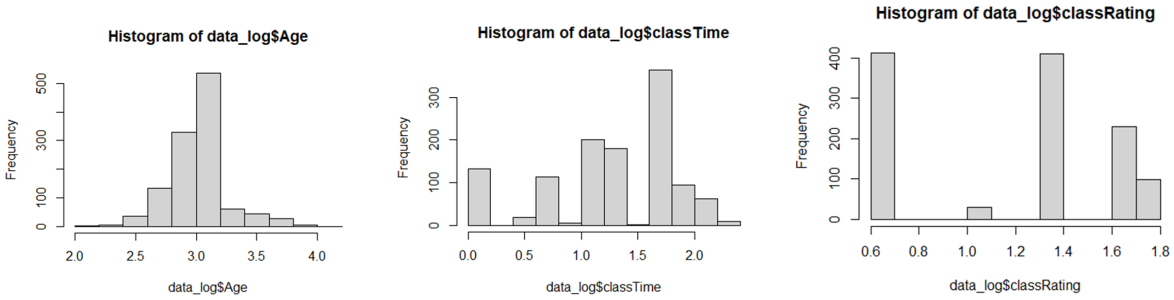

socialMediaTime
classTime


classTime
socialPlatform



Histograms (after transformations)

Fig 113 (Data log variable distributions)



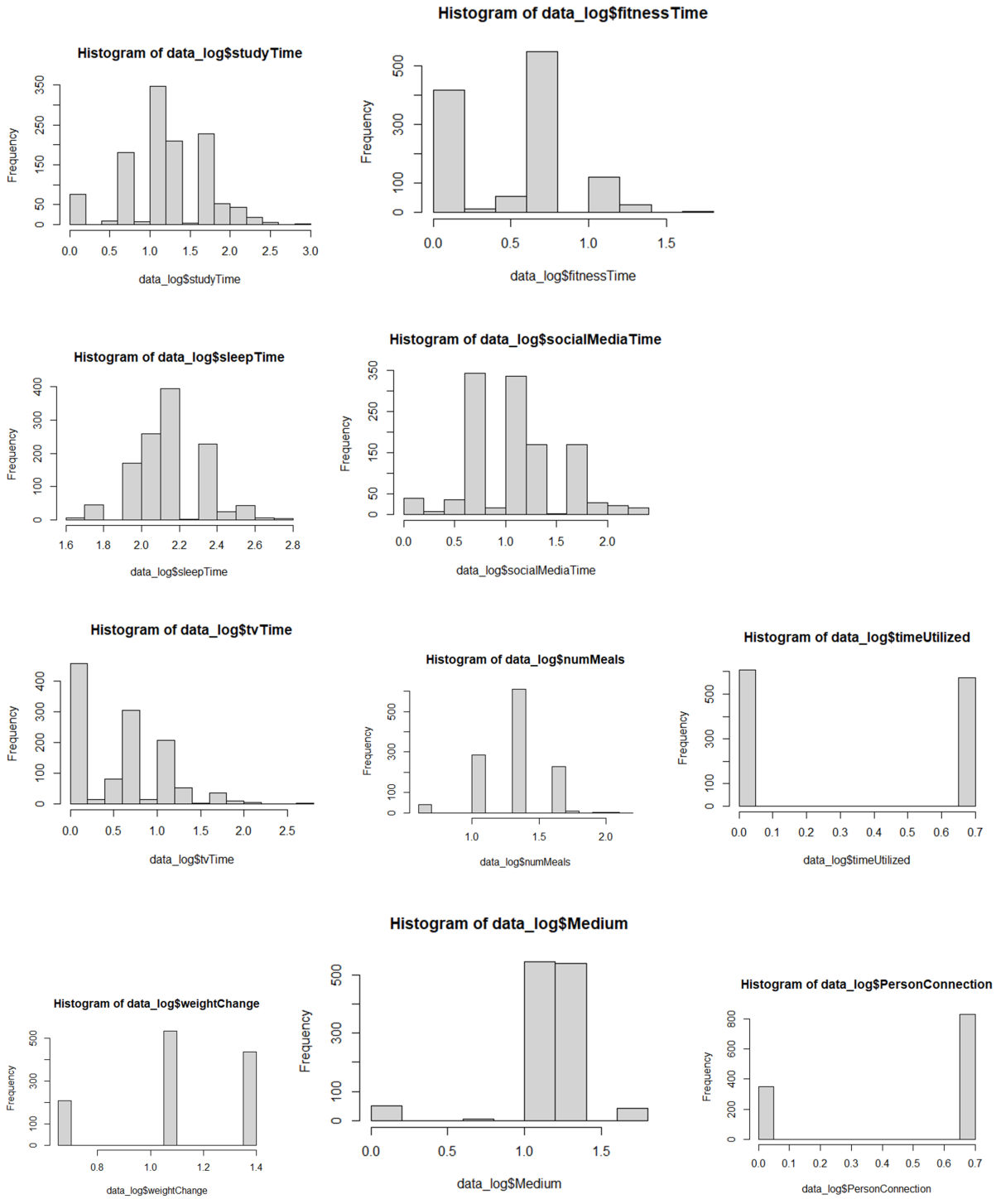


Fig. 114 (Tables of variables: medium, social platform, weight change, health issue, class rating, Miss)

```
##
##          Any Gadget      Laptop/Desktop
##           5               545
```

```
##              Smartphone Smartphone or Laptop/Desktop
##                     539                           5
##                  Tablet
##                      37
```

```
table(covidSurvey$socialPlatform)
```

```
##
##  Elyment  Facebook Instagram  Linkedin      None              Omegle       Quora    Reddit
##        1        52       352        61        18                   1           1         5
##  Snapchat  Talklife  Telegram   Twitter  Whatsapp  WhatsApp   Youtube
##        8         1         3        28       336                   1         314
```

```
table(covidSurvey$weightChange)
```

```
##
##      Decreased          Increased Remain Constant
##            209               438            535
```

```
table(covidSurvey$healthIssue)
```

```
##
##   NO  YES
## 1021  161
```

```
table(covidSurvey$timeUtilized)
```

```
##
##  NO YES
## 608 574
```

```
table(covidNum$classRating)
```

```
##
##   Average             Excellent          Good      Poor      Very poor
##       387                    98           230            30       413
```

```
table(covidNum$weightChange)
```

```
##
##      Decreased          Increased Remain Constant
##            209               438            535
```

```
table(covidNum$Miss)
```

```
##
## friends/family          nothing       outside        school
##            330               23           447           382
```

Art- Linear regression plots:

Fig 123. (trained OLS residuals violating all assumptions)

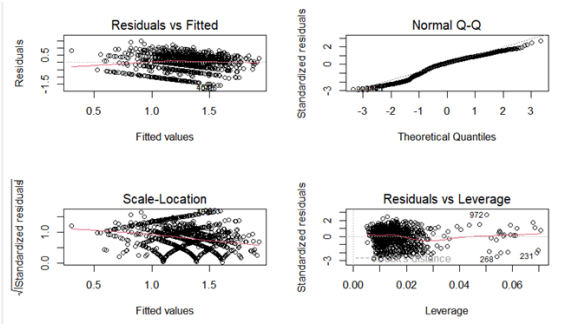


Fig 124 (OLS initfit coefficient, std.err. And p-values)

```
Call:
lm(formula = classTime ~ ., data = data_log[, -4])

Residuals:
    Min      1Q  Median      3Q     Max
-1.6582 -0.2876  0.1104  0.4167  1.5092

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       3.444716   0.367311   9.378  < 2e-16 ***
Age              -0.423057   0.075403  -5.611 2.52e-08 ***
classRating       0.158172   0.044691   3.539 0.000417 ***
studyTime         0.140837   0.033609   4.190 2.99e-05 ***
fitnessTime      -0.072438   0.043522  -1.664 0.096302 .
sleepTime        -0.507146   0.098249  -5.162 2.87e-07 ***
socialMediaTime  -0.067198   0.037483  -1.793 0.073271 .
socialPlatform    0.025828   0.026690   0.968 0.333408
tvTime            0.002109   0.033105   0.064 0.949204
numMeals          0.077887   0.078703   0.990 0.322556
weightChange     -0.112121   0.071451  -1.569 0.116872
healthIssue      -0.228314   0.072477  -3.150 0.001673 **
timeUtilized      0.023844   0.052653   0.453 0.650740
PersonConnection -0.035049   0.055141  -0.636 0.525150
Missnothing      -0.268048   0.124964  -2.145 0.032159 *
Missoutside      -0.058426   0.043137  -1.354 0.175861
Missschool       -0.058324   0.043495  -1.341 0.180202
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5759 on 1165 degrees of freedom
Multiple R-squared:  0.1182,    Adjusted R-squared:  0.106
F-statistic: 9.756 on 16 and 1165 DF,  p-value: < 2.2e-16
```

Fig 125 (VIFs of OLS)

```
                    GVIF Df GVIF^(1/(2*Df))
Age             1.129725  1        1.062885
classRating     1.164252  1        1.079005
studyTime       1.140669  1        1.068021
fitnessTime     1.096902  1        1.047331
sleepTime       1.094308  1        1.046092
socialMediaTime 1.129360  1        1.062713
socialPlatform  1.058572  1        1.028869
tvTime          1.076876  1        1.037726
numMeals        1.066334  1        1.032635
weightChange    1.064782  1        1.031883
healthIssue     1.058315  1        1.028744
timeUtilized    1.185865  1        1.088974
PersonConnection 1.086987  1        1.042587
Miss            1.103146  3        1.016496
```

Fig 126 (CV model Coefficients, R^2, RMSEs and R^2 of each fold, and variables’ p-values)

```
Coefficients:
    (Intercept) |    Age       classRating         studyTime       fitnessTime         sleepTime   socialMediaTime    socialPlatform
3.444716          -0.423057          0.158172          0.140837         -0.072438         -0.507146         -0.067198          0.025828
tvTime          numMeals      weightChange      healthIssue      timeUtilized  PersonConnection       Missnothing       Missoutside
0.002109          0.077887         -0.112121         -0.228314          0.023844         -0.035049         -0.268048         -0.058426
Missschool
-0.058324

Linear Regression

1182 samples
  14 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1064, 1063, 1064, 1062, 1064, 1064, ...
Resampling results:

  RMSE     Rsquared   MAE
  0.57958  0.1037443  0.4569616

Tuning parameter 'intercept' was held constant at a value of TRUE
```

RMSE <dbl>	Rsquared <dbl>	MAE <dbl>	Resample <chr>
0.6056610	0.07520167	0.4830943	Fold01
0.5655132	0.07098350	0.4445869	Fold02
0.5751878	0.06351936	0.4547146	Fold03
0.6232505	0.04364033	0.4809615	Fold04
0.5690730	0.18069923	0.4500603	Fold05
0.5546676	0.20335279	0.4410174	Fold06
0.6082366	0.03868981	0.4832286	Fold07
0.5598345	0.13327489	0.4345692	Fold08
0.5239043	0.14052244	0.4227937	Fold09
0.6104714	0.08755909	0.4745899	Fold10

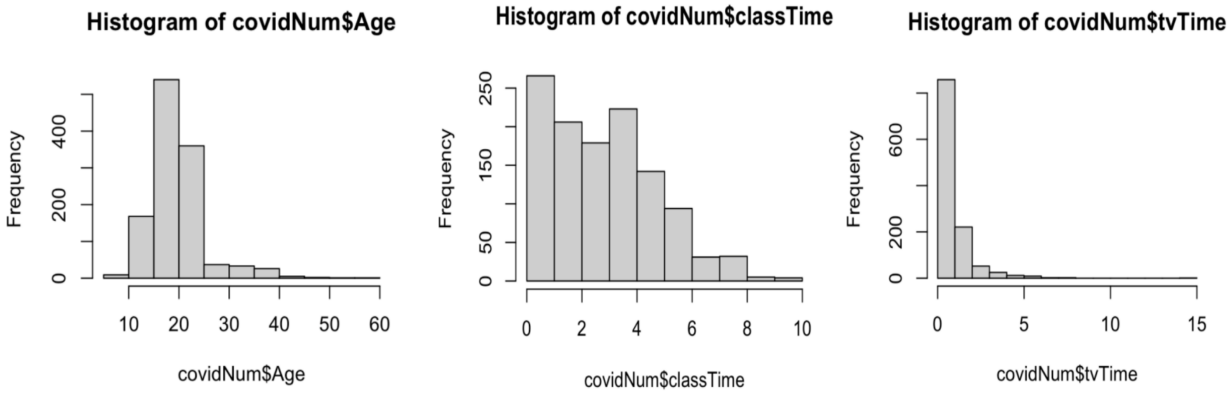
	Coefficient <dbl>	p.value <dbl>
(Intercept)	3.444715893	3.380713e-20
Age	-0.423057058	2.517168e-08
classRating	0.158172185	4.170687e-04
studyTime	0.140836693	2.994953e-05
fitnessTime	-0.072437893	9.630240e-02
sleepTime	-0.507146026	2.872707e-07
socialMediaTime	-0.067198384	7.327090e-02
socialPlatform	0.025827654	3.334077e-01
tvTime	0.002109472	9.492039e-01
numMeals	0.077887426	3.225563e-01

	Coefficient <dbl>	p.value <dbl>
weightChange	-0.112121115	1.168721e-01
healthIssue	-0.228313662	1.673307e-03
timeUtilized	0.023844206	6.507396e-01
PersonConnection	-0.035048682	5.251496e-01
Missnothing	-0.268047701	3.215861e-02
Missoutside	-0.058425592	1.758607e-01
Missschool	-0.058323679	1.802024e-01

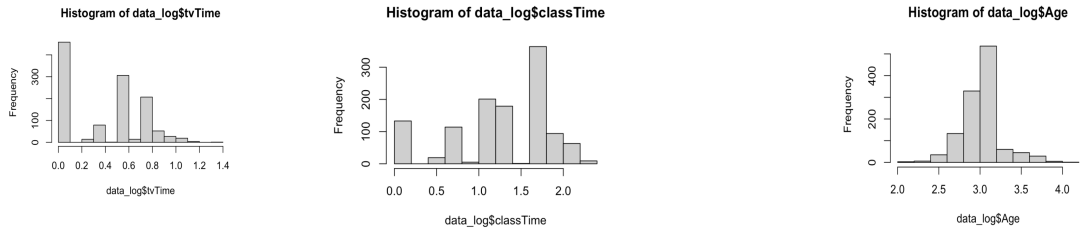
```
> cat("Training set RMSE:", train_rmse, "\n")
Training set RMSE: 0.6478398
> cat("Testing set RMSE:", test_rmse, "\n")
Testing set RMSE: 0.6287978
```

Areli - PCA Factor Analysis

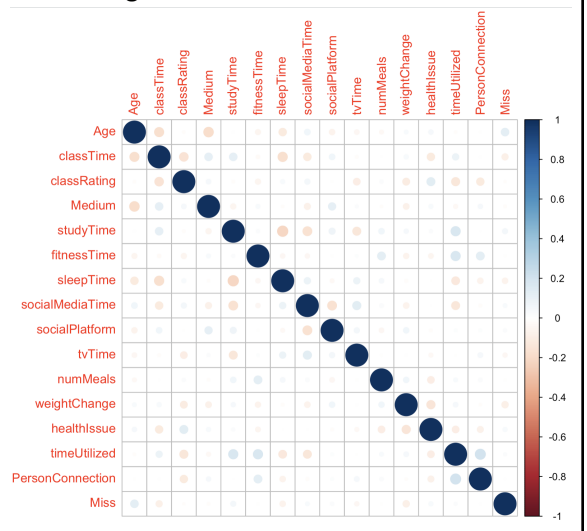
Fig. 200:



After



Before Log



After Log

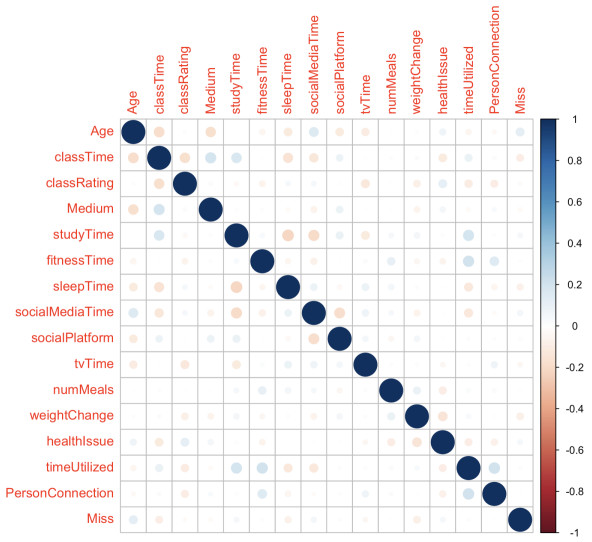


Fig. 201:

```
> summary(pca)
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7     PC8
Standard deviation     0.7033 0.6200 0.5781 0.5420 0.4954 0.44971 0.42577 0.40912
Proportion of Variance 0.1751 0.1361 0.1183 0.1040 0.0869 0.07161 0.06419 0.05926
Cumulative Proportion  0.1751 0.3112 0.4296 0.5336 0.6205 0.69209 0.75628 0.81554
                           PC9    PC10    PC11    PC12    PC13    PC14    PC15
Standard deviation     0.33037 0.30567 0.28425 0.25853 0.22997 0.21451 0.2112
Proportion of Variance 0.03864 0.03308 0.02861 0.02367 0.01873 0.01629 0.0158
Cumulative Proportion  0.85419 0.88727 0.91588 0.93954 0.95827 0.97456 0.9904
                          PC16
Standard deviation     0.16500
Proportion of Variance 0.00964
Cumulative Proportion  1.00000
```

Fig. 202:

Without rotation

```
> print(p1$loadings, cutoff = 0.4, sort=T)

Loadings:
                RC2    RC3    RC6    RC4    RC1    RC5
studyTime        0.610
sleepTime       -0.698
fitnessTime             0.707
timeUtilized            0.618
PersonConnection        0.598
Age                            -0.559
classTime        0.408         0.626
Medium                         0.714
weightChange                          0.658
healthIssue                          -0.574
socialMediaTime                              -0.651
socialPlatform                                0.719
classRating                                          0.662
tvTime          -0.404                               -0.586
numMeals                              0.499
Miss

                RC2   RC3   RC6   RC4   RC1   RC5
SS loadings   1.489 1.426 1.381 1.326 1.306 1.213
Proportion Var 0.093 0.089 0.086 0.083 0.082 0.076
Cumulative Var 0.093 0.182 0.268 0.351 0.433 0.509
```

With rotation

```
> print(p2$loadings, cutoff = 0.4, sort=T)

Loadings:
                RC2    RC3    RC6    RC4    RC1    RC5
studyTime        0.610
sleepTime       -0.698
fitnessTime             0.707
timeUtilized            0.618
PersonConnection        0.598
Age                            -0.559
classTime        0.408         0.626
Medium                         0.714
weightChange                          0.658
healthIssue                          -0.574
socialMediaTime                              -0.651
socialPlatform                                0.719
classRating                                          0.662
tvTime          -0.404                               -0.586
numMeals                              0.499
Miss

                RC2   RC3   RC6   RC4   RC1   RC5
SS loadings   1.489 1.426 1.381 1.326 1.306 1.213
Proportion Var 0.093 0.089 0.086 0.083 0.082 0.076
Cumulative Var 0.093 0.182 0.268 0.351 0.433 0.509
```

Fig. 203:

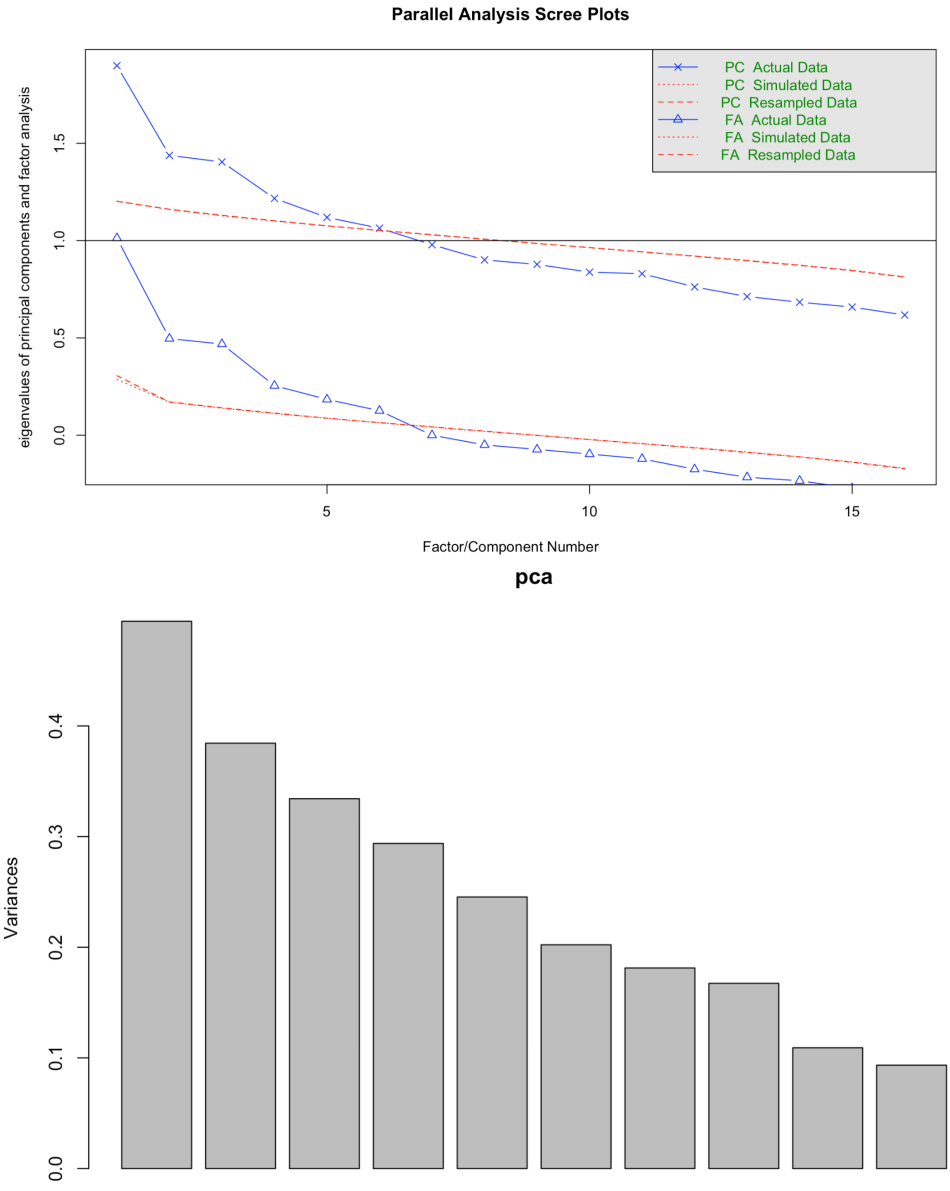


Fig. 204:

```
> polychor(polyPCA$healthIssue, polyPCA$Age, ML = FALSE, control = list(),
+          std.err = FALSE, maxcor=.9999, start, thresholds=FALSE)
[1] 0.1520573
```

Fig. 205:

```
> ## function polychor
> polychor(polyPCA$classTime, polyPCA$Age, ML = FALSE, control = list(),
+          std.err = FALSE, maxcor=.9999, start, thresholds=FALSE)
[1] -0.2252136
> ## function polychor
> polychor(polyPCA$studyTime, polyPCA$Age, ML = FALSE, control = list(),
+          std.err = FALSE, maxcor=.9999, start, thresholds=FALSE)
[1] 0.02531156
> ## function polychor
> polychor(polyPCA$socialMediaTime, polyPCA$Age, ML = FALSE, control = list(),
+          std.err = FALSE, maxcor=.9999, start, thresholds=FALSE)
[1] 0.1727742
```

Fig. 206:

```
> corr_matrix

Maximum-Likelihood Estimates

Correlations/Type of Correlation:
                    Age classTime classRating    Medium studyTime fitnessTime sleepTime socialMediaTime socialPlatform   tvTime  numMeals weightChange healthIssue timeUtilized
Age                   1   Pearson     Pearson   Pearson   Pearson     Pearson   Pearson         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
classTime       -0.1816         1     Pearson   Pearson   Pearson     Pearson   Pearson         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
classRating      0.01439   -0.1716           1   Pearson   Pearson     Pearson   Pearson         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
Medium          -0.1713    0.1956     0.02504         1   Pearson     Pearson   Pearson         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
studyTime      0.0008232    0.1731   -0.03131  -0.01276         1     Pearson   Pearson         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
fitnessTime     -0.05719  0.007776   -0.06429   0.01841   0.03963           1   Pearson         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
sleepTime       -0.1137   -0.1593    0.04167  -0.02456   -0.2176   -0.05234         1         Pearson        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
socialMediaTime  0.1505   -0.1336    0.04274  -0.06861   -0.1962   -0.07531   0.08348               1        Pearson  Pearson   Pearson      Pearson     Pearson      Pearson
socialPlatform  -0.1105    0.08121   0.008771   0.08553    0.0949  -0.005073   -0.0364         -0.1869              1  Pearson   Pearson      Pearson     Pearson      Pearson
tvTime          -0.1033  0.009488    -0.1219 -0.003341   -0.1184    0.02511   0.08966         0.07937         0.0587        1  Pearson      Pearson     Pearson      Pearson
numMeals       -0.002532    0.01439   0.008619 0.0008683   0.04276     0.1077   0.05873         0.04145        -0.0491 -0.01341        1  Pearson      Pearson      Pearson
weightChange     0.01282    0.02629   -0.08838  -0.06077   0.05963   -0.04953   -0.0126         -0.06674        0.05776   0.0175  0.09444            1  Pearson      Pearson
healthIssue       0.0737   -0.1133     0.1188   0.05438    0.0143   -0.0603  0.001208         0.006637        -0.0086 -0.04203 -0.09635      -0.1447            1  Pearson
timeUtilized    -0.05218    0.09847    -0.1073  -0.02807    0.1968     0.202   -0.1246          -0.1284        0.01897  0.01712  0.02947      0.03732     -0.09467            1
PersonConnection -0.03149    0.02306   -0.09446   0.03185   0.01363     0.1426  -0.05991         -0.02188        0.01149   0.0746  0.02463    -0.008462      -0.0766       0.2091
Miss              0.1201   -0.09059   -0.02291  -0.02927   0.04639   -0.01049    -0.076          0.05443        0.01898 -0.04014 -0.007502      -0.08816      0.04739      0.01691
                PersonConnection    Miss
Age                      Pearson Pearson
classTime                Pearson Pearson
classRating              Pearson Pearson
Medium                   Pearson Pearson
studyTime                Pearson Pearson
fitnessTime              Pearson Pearson
sleepTime                Pearson Pearson
socialMediaTime          Pearson Pearson
socialPlatform           Pearson Pearson
tvTime                   Pearson Pearson
numMeals                 Pearson Pearson
weightChange             Pearson Pearson
healthIssue              Pearson Pearson
timeUtilized             Pearson Pearson
PersonConnection               1 Pearson
Miss                    -0.001766       1
```

```
Standard Errors:
                    Age classTime classRating  Medium studyTime fitnessTime sleepTime socialMediaTime socialPlatform  tvTime numMeals weightChange healthIssue timeUtilized
Age
classTime        0.02814
classRating      0.02909   0.02824
Medium           0.02825   0.02799     0.02908
studyTime         0.0291   0.02823     0.02907 0.02909
fitnessTime        0.029    0.0291     0.02898 0.02909   0.02905
sleepTime        0.02872   0.02836     0.02905 0.02908   0.02772     0.02902
socialMediaTime  0.02844   0.02858     0.02905 0.02896   0.02798     0.02893    0.0289
socialPlatform   0.02874   0.02891      0.0291 0.02889   0.02884      0.0291   0.02906        0.02808
tvTime           0.02879    0.0291     0.02867  0.0291   0.02869     0.02908   0.02886        0.02892         0.029
numMeals          0.0291   0.02909      0.0291  0.0291   0.02905     0.02876     0.029        0.02905       0.02903 0.02909
weightChange     0.02909   0.02908     0.02887 0.02899   0.02899     0.02903   0.02909        0.02897         0.029 0.02909  0.02884
healthIssue      0.02894   0.02873     0.02869 0.02901   0.02909     0.02899    0.0291         0.0291        0.0291 0.02905  0.02883      0.02849
timeUtilized     0.02902   0.02882     0.02876 0.02908   0.02797     0.02791   0.02865        0.02862       0.02909 0.02909  0.02907      0.02906     0.02884
PersonConnection 0.02907   0.02908     0.02884 0.02907   0.02909     0.02851   0.02899        0.02908       0.02909 0.02894  0.02908       0.0291     0.02893      0.02783
Miss             0.02868   0.02886     0.02908 0.02907   0.02904      0.0291   0.02893        0.02901       0.02909 0.02905   0.0291      0.02887     0.02903      0.02909
                 PersonConnection
Age
classTime
classRating
Medium
studyTime
fitnessTime
sleepTime
socialMediaTime
socialPlatform
tvTime
numMeals
weightChange
healthIssue
timeUtilized
PersonConnection
Miss                       0.0291

n = 1182

n = 1182

P-values for Tests of Bivariate Normality:
                                  Age            classTime           classRating               Medium  studyTime fitnessTime  sleepTime       socialMediaTime        socialPlatform
Age
classTime                  1.114e-74
classRating               1.415e-143           6.034e-68
Medium                    3.473e-272 8.97999999999999e-223           4.636e-269
studyTime                  1.759e-63           1.638e-21            1.643e-62          1.779e-211
fitnessTime               1.193e-177          1.924e-130           9.441e-178 2.025669147941e-322 1.344e-125
sleepTime                  2.862e-64           2.438e-20            3.924e-63          3.994e-211  3.891e-19   6.66e-126
socialMediaTime           8.261e-200          2.334e-146 8.99699999999999e-195                   0 2.297e-146  1.536e-257 7.161e-144
socialPlatform   1.4020000042645 3e-316          2.178e-257           4.726e-309                   0 4.022e-255           0 6.942e-255                   0
tvTime                    6.627e-116           5.071e-67            1.084e-111          7.488e-257  6.247e-64  1.573e-172  5.317e-61          5.115e-191 7.04599999999999e-303
numMeals                   4.23e-236          2.565e-189           7.587e-236                   0 4.017e-186  4.905e-301 1.343e-186 9.45799999120294e-316                   0
weightChange               1.58e-230           1.49e-184           5.452e-231                   0 2.623e-180   1.35e-297 2.467e-181          1.136e-309                   0
healthIssue                        0                   0                    0                   0         0           0         0                   0                   0
timeUtilized                       0                   0                    0                   0         0           0         0                   0                   0
PersonConnection                   0                   0                    0                   0         0           0         0                   0                   0
Miss                      1.762e-226          2.766e-178           1.231e-233                   0 3.304e-172  5.455e-284 1.225e-175          1.792e-301                   0
                  tvTime numMeals weightChange healthIssue timeUtilized PersonConnection
Age
classTime
classRating
Medium
studyTime
fitnessTime
sleepTime
socialMediaTime
socialPlatform
tvTime
numMeals        1.435e-232
weightChange     1.41e-227       0
healthIssue              0       0          0
timeUtilized             0       0          0         0
PersonConnection         0       0          0         0          0
Miss             1.034e-218       0          0         0          0              0
```

Fig. 207:

```
> print(p3)
Standard deviations (1, .., p=4):
[1] 0.5346785 0.4029032 0.2375580 0.1707963

Rotation (n x k) = (4 x 4):
                    PC1          PC2         PC3         PC4
Age          0.0004704301 -0.05030392  0.98198620 0.18213290
studyTime    0.9940952605 -0.07249571 -0.01875552 0.07853163
fitnessTime  0.0711142456  0.99598014  0.04540910 0.03007221
sleepTime   -0.0819582562 -0.01540937 -0.18245368 0.97967140
```

Fig. 208

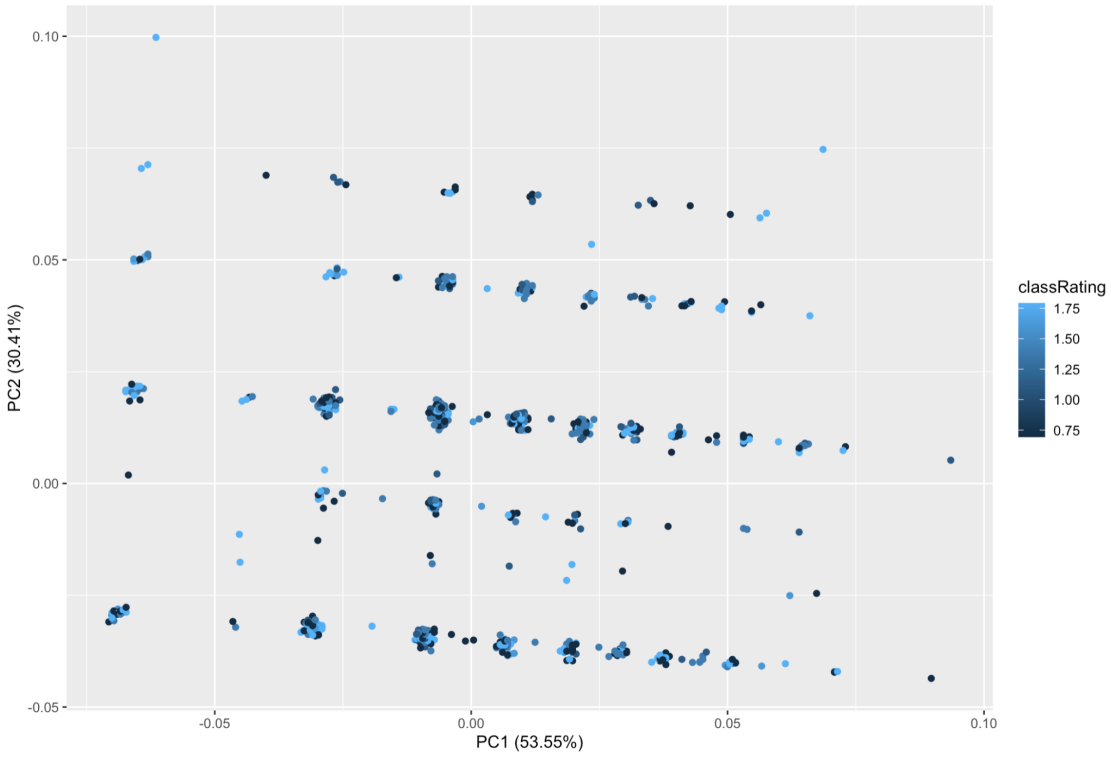


Fig. 209:

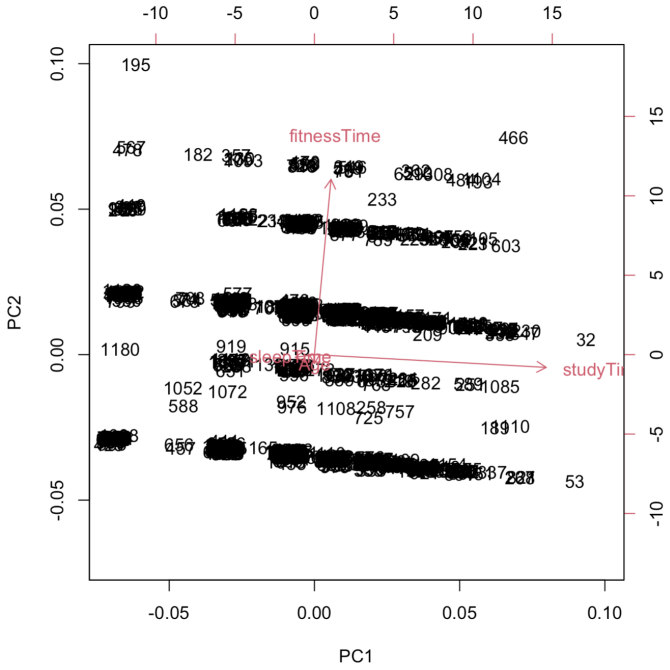


Fig. 210

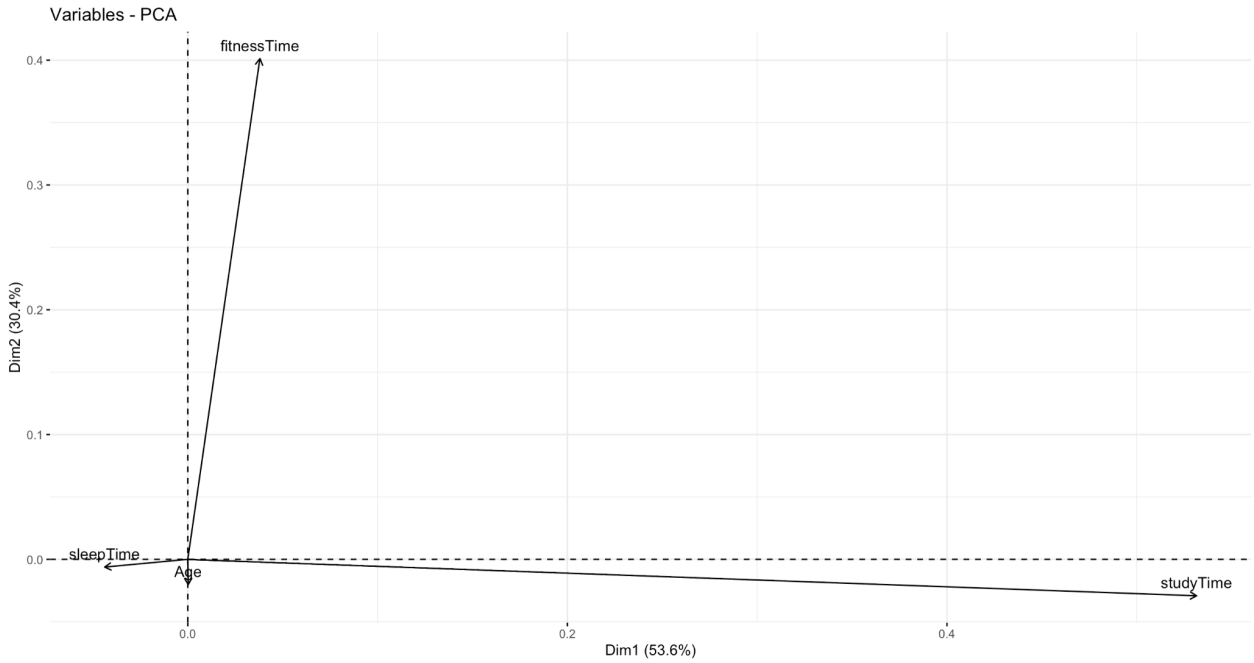
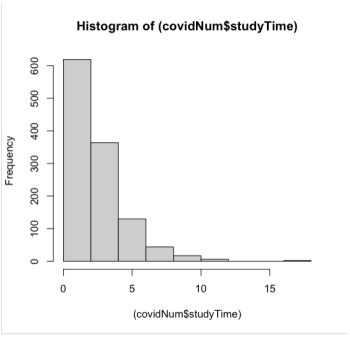


Fig. 301: Left image shows studyTime before transformation, right shows data after transformation.



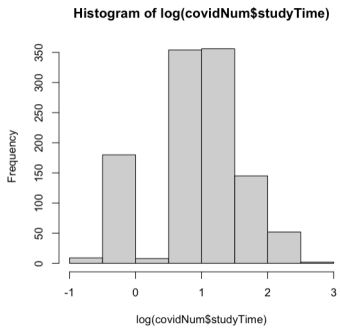


Fig. 302: Prior Probabilities, coefficients of discriminants, proportion of trace

```
Call:
lda(Miss ~ ., data = c)

Prior probabilities of groups:
friends/family        nothing        outside         school
    0.27918782     0.01945854     0.37817259     0.32318105

Group means:
              classTime classRating studyTime fitnessTime sleepTime socialMediaTime
friends/family  1.352315    1.252564  1.217006   0.5233069  2.177875        1.064929
nothing         1.083179    1.119153  1.189027   0.3542551  2.114617        1.121684
outside         1.231134    1.159949  1.221026   0.4860425  2.159827        1.123286
school          1.287269    1.249148  1.246304   0.4666422  2.168512        1.093343

Coefficients of linear discriminants:
                      LD1          LD2         LD3
classTime       -0.93248643 -0.472515993 -0.16550684
classRating     -1.35507812  1.678731705  0.03740727
studyTime        0.05661164  0.429614033  1.48325185
fitnessTime     -0.65454725 -1.884586147  0.34334496
sleepTime       -2.72818866 -0.894906573  2.71424392
socialMediaTime  0.37936682 -0.004741383  1.39082470

Proportion of trace:
   LD1    LD2    LD3
0.7669 0.1997 0.0333
```

Fig. 303: LDA projection with color

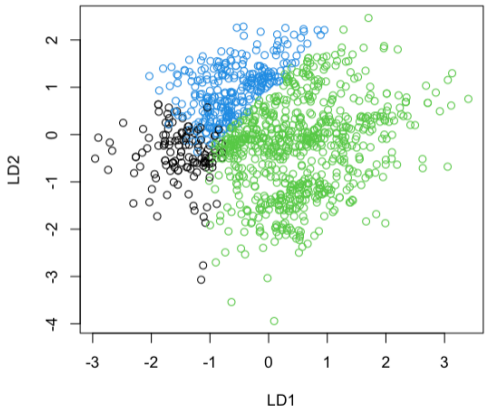


Fig. 304: LDA Confusion Matrices

```
> table(ldaResult$class, c$Miss)

                friends/family nothing outside school
  friends/family             67       0      28     32
  nothing                     0       0       0      0
  outside                   183      19     319    235
  school                     80       4     100    115
```

```
> confusion(c$Miss, ldaResult$class)
Overall accuracy = 0.424

Confusion matrix
      Predicted (cv)
Actual  [,1] [,2]  [,3]  [,4]
  [1,] 0.203    0 0.555 0.242
  [2,] 0.000    0 0.826 0.174
  [3,] 0.063    0 0.714 0.224
  [4,] 0.084    0 0.615 0.301
```

Fig. 305: MDS plot in 3D with rotation to show similarity of classes (closeness between points) before LDA.

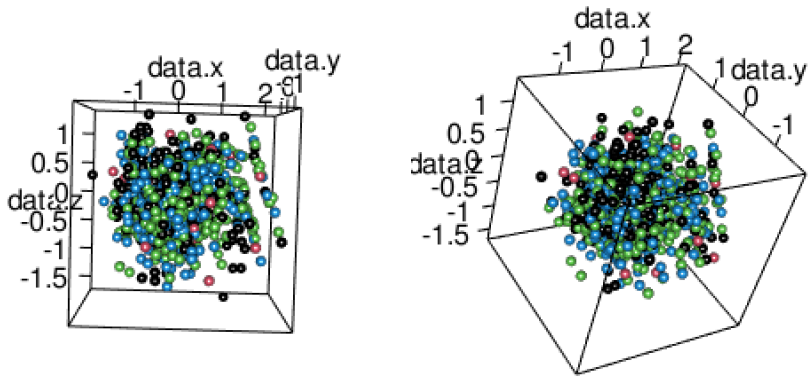
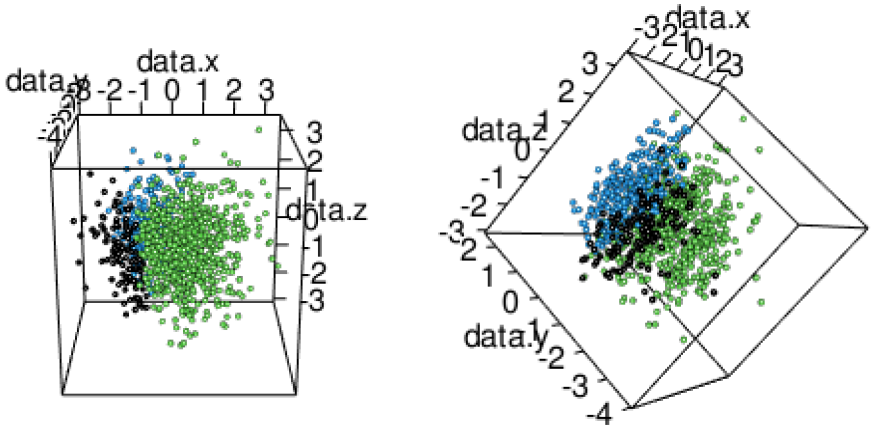


Fig. 306: MDS plot in 3D with rotation to show overlap present even after LDA.



Bayesian Network Appendix- Art

Fig 421: (cond. Prob, Tables of ages 7-17)

```
fittedBN1 <- bn.fit(res5, filtered_dataset1[,c(8,9,10,11)])
> fittedBN1

  Bayesian network parameters

  Parameters of node classStudyBin (multinomial distribution)

Conditional probability table:

, , sleepTimeBin = 1, TTLmediaBin = 1

            fitnessBins
classStudyBin         1         2         3
            1 0.42391304 0.30769231 0.60000000
            2 0.53260870 0.69230769 0.40000000
            3 0.04347826 0.00000000 0.00000000

, , sleepTimeBin = 2, TTLmediaBin = 1
```

```
          fitnessBins
classStudyBin        1         2         3
         1 0.61538462 0.90000000 0.66666667
         2 0.38461538 0.10000000 0.33333333
         3 0.00000000 0.00000000 0.00000000

, , sleepTimeBin = 3, TTLmediaBin = 1

          fitnessBins
classStudyBin        1         2 3
         1 0.75000000 0.50000000
         2 0.25000000 0.50000000
         3 0.00000000 0.00000000

, , sleepTimeBin = 1, TTLmediaBin = 2

          fitnessBins
classStudyBin        1         2         3
         1 0.45454545 1.00000000 1.00000000
         2 0.45454545 0.00000000 0.00000000
         3 0.09090909 0.00000000 0.00000000

, , sleepTimeBin = 2, TTLmediaBin = 2

          fitnessBins
classStudyBin        1         2         3
         1 0.80000000 0.33333333 0.50000000
         2 0.16000000 0.66666667 0.50000000
         3 0.04000000 0.00000000 0.00000000

, , sleepTimeBin = 3, TTLmediaBin = 2

          fitnessBins
classStudyBin        1 2 3
         1 0.50000000
         2 0.50000000
         3 0.00000000

, , sleepTimeBin = 1, TTLmediaBin = 3

          fitnessBins
classStudyBin        1 2 3
         1 0.50000000
         2 0.50000000
         3 0.00000000

, , sleepTimeBin = 2, TTLmediaBin = 3

          fitnessBins
classStudyBin        1 2 3
         1 0.66666667
         2 0.33333333
         3 0.00000000

, , sleepTimeBin = 3, TTLmediaBin = 3

          fitnessBins
classStudyBin 1 2 3
         1
         2
         3

 Parameters of node sleepTimeBin (multinomial distribution)

Conditional probability table:

          TTLmediaBin
sleepTimeBin         1         2         3
        1 0.44000000 0.31914894 0.40000000
        2 0.52000000 0.63829787 0.60000000
        3 0.04000000 0.04255319 0.00000000

 Parameters of node TTLmediaBin (multinomial distribution)
```

```
Conditional probability table:
       1        2        3
0.82781457 0.15562914 0.01655629
```

Fig 422: (cond.prob.tables of ages 18-27)

```
> fittedBN2 <- bn.fit(res5, filtered_dataset2[,c(8,9,10,11)])
> fittedBN2

  Bayesian network parameters

  Parameters of node classStudyBin (multinomial distribution)

Conditional probability table:

, , sleepTimeBin = 1, TTLmediaBin = 1

            fitnessBins
classStudyBin         1         2         3
            1 0.536231884 0.535714286 0.333333333
            2 0.429951691 0.464285714 0.666666667
            3 0.033816425 0.000000000 0.000000000

, , sleepTimeBin = 2, TTLmediaBin = 1

            fitnessBins
classStudyBin         1         2         3
            1 0.757679181 0.793103448 0.833333333
            2 0.235494881 0.206896552 0.166666667
            3 0.006825939 0.000000000 0.000000000

, , sleepTimeBin = 3, TTLmediaBin = 1

            fitnessBins
classStudyBin         1         2         3
            1 0.952380952 1.000000000 1.000000000
            2 0.047619048 0.000000000 0.000000000
            3 0.000000000 0.000000000 0.000000000

, , sleepTimeBin = 1, TTLmediaBin = 2

            fitnessBins
classStudyBin         1         2         3
            1 0.822222222 0.875000000 1.000000000
            2 0.177777778 0.125000000 0.000000000
            3 0.000000000 0.000000000 0.000000000

, , sleepTimeBin = 2, TTLmediaBin = 2

            fitnessBins
classStudyBin         1         2         3
            1 0.795918367 0.750000000 1.000000000
            2 0.204081633 0.250000000 0.000000000
            3 0.000000000 0.000000000 0.000000000

, , sleepTimeBin = 3, TTLmediaBin = 2

            fitnessBins
classStudyBin         1         2 3
            1 1.000000000 1.000000000
            2 0.000000000 0.000000000
            3 0.000000000 0.000000000

, , sleepTimeBin = 1, TTLmediaBin = 3

            fitnessBins
classStudyBin         1 2 3
            1 1.000000000
            2 0.000000000
            3 0.000000000

, , sleepTimeBin = 2, TTLmediaBin = 3
```

```
          fitnessBins
classStudyBin         1         2         3
         1 0.818181818 0.000000000 1.000000000
         2 0.181818182 1.000000000 0.000000000
         3 0.000000000 0.000000000 0.000000000

, , sleepTimeBin = 3, TTLmediaBin = 3

          fitnessBins
classStudyBin         1 2 3
         1 1.000000000
         2 0.000000000
         3 0.000000000


 Parameters of node sleepTimeBin (multinomial distribution)

Conditional probability table:

          TTLmediaBin
sleepTimeBin         1         2         3
         1 0.40202703 0.31791908 0.32000000
         2 0.55405405 0.64739884 0.52000000
         3 0.04391892 0.03468208 0.16000000

 Parameters of node TTLmediaBin (multinomial distribution)

Conditional probability table:
         1         2         3
0.74936709 0.21898734 0.03164557
```

Fig. 432 (network validated by HC)

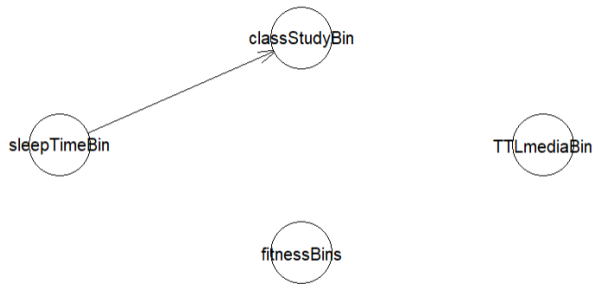


Fig.433 (Network validated by IC)

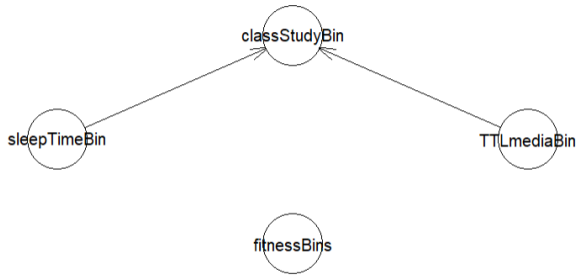


Fig.434 (Network validated by *Incremental Association)*

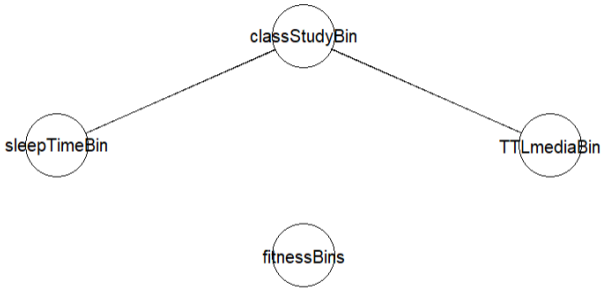


Fig.435 (Network used for probabilistic inferences)

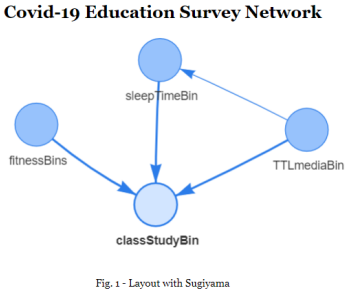


Fig.436 (network and conditional probabilities of variables for ages 18-27)

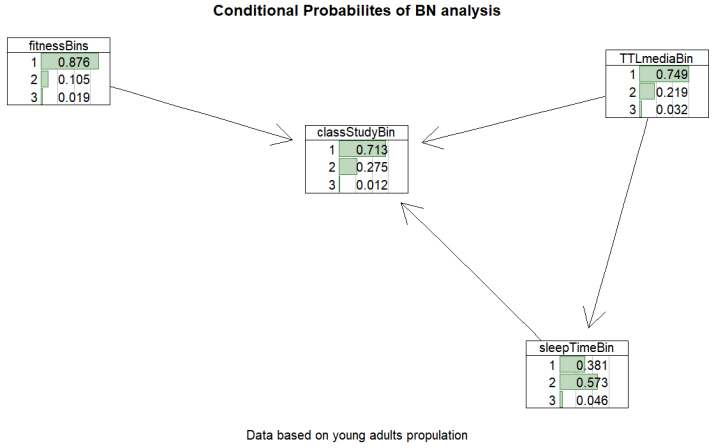
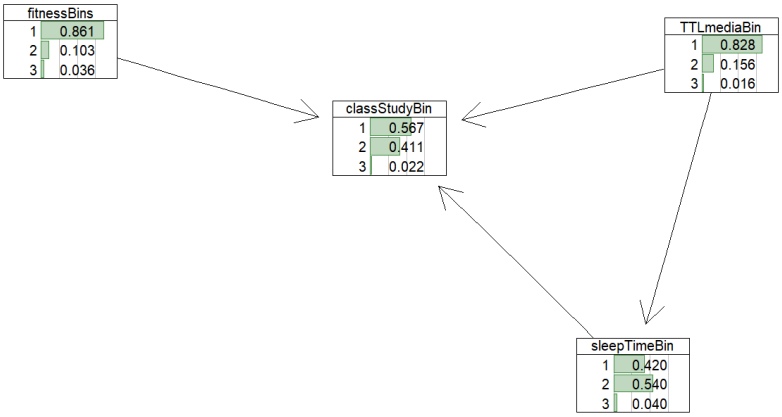


Fig. 437(network and conditional probabilities of variables for ages 7-17)

Conditional Probabilites of BN analysis



Data based on Adolenscent propulation

Code added:

```
---
title: "Final Project"
author: "Teammates"
date: "2023-05-05"
output:
  word_document: default
  html_document: default
---
Final Project Data

``` {r warning = FALSE, message = FALSE}
library(caret)
library(readr)
library(QuantPsyc)
library(psych)
library(corrplot)
library(ggplot2)
library(leaps)
library(car)
library(tidyverse)
library(glmnet)
library(MASS)
library(stringdist)
library(tm)
library(randomForest)
library(caret)
library(rattle)
library(bnlearn)
library(rgl)
library(Rgraphviz)
library(gRain)
```
#CLEANING AND PRE PROCESSING
<u>*Importing and overview of data*</u>
``` {r echo=TRUE}
covidSurvey <- read_csv("COVID19_Student.csv")
#overview of data
head(covidSurvey)
summary(covidSurvey)
```
cleaning and renaming columns - not removing yet
```{r echo=TRUE}
covidSurvey <- as.data.frame(covidSurvey)
colNames <- list("ID", "Region", "Age", "classTime", "classRating", "Medium",
                 "studyTime", "fitnessTime", "sleepTime", "socialMediaTime",
                 "socialPlatform", "tvTime", "numMeals","weightChange",
                 "healthIssue", "copingMech", "timeUtilized", "PersonConnection",
                 "Miss")
names(covidSurvey) <- c(colNames)
head(covidSurvey)
```
Adding some tables for the categoricals to understand better FOR COVIDSURVEY
```{r echo=TRUE}
#some tables for the categoricals to understand better
table(covidSurvey$Medium)
table(covidSurvey$socialPlatform)
table(covidSurvey$weightChange)
table(covidSurvey$healthIssue)
table(covidSurvey$timeUtilized)
table(covidSurvey$Connection)
```
<u>*Casting the categorical to numeric - placing this in a separate data set for now.*</u>
``` {r echo=TRUE}
covidNum <- covidSurvey
# removing certain columns due to lack of correlation or use
covidNum <- covidNum[,-c(1)] #took out ID
covidNum <- covidNum[,-c(1)] #took out the region variable
#covidNum <- covidNum[,-c("Medium")]
```
```

#likert scale conversion to ordinals
#I am ensuring class rating is being classed ordinally instead of to random values

```
table(covidNum$classRating)
recode <- list("Very poor"=1, "Poor"=2, "Average"=3, "Good"=4, "Excellent"=5)
covidNum$classRating[is.na(covidNum$classRating)] <- 'Average'
covidNum$classRating = unlist(recode[as.character(covidNum$classRating)])
table(covidNum$classRating)
```

#I am ensuring weight is being classed ordinarily instead of to random values

```
table(covidNum$weightChange)
recode <- list("Decreased"=1, "Remain Constant"=2, "Increased"=3)
covidNum$weightChange[is.na(covidNum$weightChange)] <- 'Remain Constant'
covidNum$weightChange = unlist(recode[as.character(covidNum$weightChange)])
table(covidNum$weightChange)

#covidNum$weightChange <- factor(covidNum$weightChange, levels = c(1, 2, 3), ordered = T)
#covidNum$classRating <- factor(covidNum$classRating, levels = c(1, 2, 3, 4,), ordered = T)
```

#binary conversion

```
covidNum<- covidNum %>%
  mutate(healthIssue = ifelse(tolower(healthIssue) == "yes",1,0)) %>%
  mutate(timeUtilized = ifelse(tolower(timeUtilized) == "yes",1,0)) %>%
  mutate(PersonConnection = ifelse(tolower(PersonConnection) == "yes",1,0))
covidNum$tvTime <- as.numeric(covidNum$tvTime) #changed the tvTime from char to double. still has NAs to remove
table(covidNum$Miss)

#I see here the main categories are school, friends/family, nothing, being outside (travelling, eating outside)

recode2 <- list('.' = 'friends/family',
            'all' = 'friends/family',
            'All' = 'friends/family',
            'All ' = 'friends/family',
            'ALL' = 'friends/family',
            'All above' = 'friends/family',
            'all of the above' = 'friends/family',
            'All of the above' = 'friends/family',
            'All of them' = 'friends/family',
            'All the above' = 'friends/family',
            'Badminton in court' = 'outside',
            'Being social' = 'friends/family',
            'Colleagues' = 'friends/family',
            'Eating outside' = 'outside',
            'Eating outside and friends.' = 'friends/family',
            'everything' = 'friends/family',
            'Family' = 'friends/family',
            'Family ' = 'friends/family',
            'Football' = 'outside',
            'Friends , relatives' = 'friends/family',
            'Friends and roaming around freely' = 'friends/family',
            'Friends and School' = 'friends/family',
            'Friends, relatives & travelling' = 'friends/family',
            "Friends,Romaing and traveling" = 'friends/family',
            "Going to the movies" = 'friends/family',
            "Gym" = 'outside',
            "I have missed nothing" = 'nothing',
            "Internet"= 'nothing',
            "Job"= 'outside',
            "Metro"= 'outside',
            "My normal routine"= 'nothing',
            "Nah, this is my usual lifestyle anyway, just being lazy...."= 'nothing',
            "Normal life"= 'nothing',
            "nothing"= 'nothing',
            "Nothing"= 'nothing',
            "NOTHING"= 'nothing',
            "Nothing " = 'nothing',
            'Nothing this is my usual life'= 'nothing',
            'Only friends'= 'friends/family',
            'Playing'= 'friends/family',
            'Previous mistakes'= 'outside',
            'Roaming around freely'= 'outside',
            'School and friends.'= 'school',
            'School and my school friends'= 'school',
            'school, relatives and friends'= 'school',
```

```
        'School/college'= 'school',
        'Taking kids to park'= 'outside',
        'The idea of being around fun loving people but this time has certainly made us all to reconnect (and fill the gap if any) with our families and relatives so it is fun but certainly we do miss hanging out with friends'='friends/family',
        'To stay alone.'= 'nothing',
        'Travelling'= 'outside',
        'Travelling & Friends' = 'outside',
        ' ' = 'nothing')

covidNum$Miss[is.na(covidNum$Miss)] <- 'nothing' #replacing nulls with nothing
covidNum$Miss = unlist(recode2[as.character(covidNum$Miss)])
table(covidNum$Miss)



#covidNum$Region <- unclass(as.factor(covidNum$Region))

#if you want to unclass the categoricals de-comment this:
#dont forget to remove copingMech:
covidNum$Medium <- unclass(as.factor(covidNum$Medium))
covidNum$socialPlatform <- unclass(as.factor(covidNum$socialPlatform))
#covidNum$Miss <- unclass(as.factor(covidNum$Miss))
covidNum <- covidNum[,!(names(covidNum) == "copingMech")]


head(covidNum)
```

Looking at summary and distributions of each variable
we found some NAs in classRating,Medium, and TvTime so we modified the columns

```
summary(covidNum)

covidNum$classRating[is.na(covidNum$classRating)] <- 3 #replacing missing class rating with median value
covidNum$Medium[is.na(covidNum$Medium)] <- 0 #replacing missing medium (social media) with 0 / no medium
head(covidNum)

covidNum$tvTime[is.na(covidNum$tvTime)] <- 0 #replacing the NAs with 0 due to NA being "no TV" in char
head(covidNum)
```

#EXPLORATORY - Visualizations, histograms, tables

checking skew with histograms of the variables

```
hist(covidNum$Age) #right skew, makes sense for students
hist(covidNum$classTime) #right skew
hist(covidNum$classRating) #three spikes . there must be a way to make this hist look better.
hist(covidNum$studyTime) #right skew  also an outlier
hist(covidNum$fitnessTime) #heavy right skew
hist(covidNum$sleepTime) #pretty even!
hist(covidNum$socialMediaTime) #right skew
hist(covidNum$tvTime) #massive right skew with an outlier
hist(covidNum$numMeals) #kind of normal
hist(covidNum$timeUtilized) #even split
```

checking out the intervariable correlation using corrplot

```
#classic correlation plot and bubble-corr plot - removing ID, copingMech, and MISS
corrCovid <- covidNum
head(corrCovid)
summary(corrCovid)
corrCovid <- sapply(corrCovid[,-16],as.numeric)
corrCovid<- as.data.frame(corrCovid)


#sapply(corrCovid,class)
#plot(corrCovid)

corrplot(cor(corrCovid))
pairs.panels(corrCovid)
corrplot(cor(corrCovid[,-c(12,4,15,9)]))
```

We notice that vast majority of variable have little to no correlation. This means that there will be very few variables that could explain the dataset variance and will likely issue an inaccurate model.

```
``` {r include=FALSE}
# these actions have already been done. kept this for all to confirm that its implemented in prior chunks
#otherwise the PCA or corr table wouldn't work

# df <- mutate_all(corrCovid, function(x) as.numeric(as.character(x))) #wasn't reading the factors as nums so had to convert
#df2 <- df[,-c(10)] #remove tv time because it was being weird - idk why !!! fixed it
# corrplot(cor(df), method= "circle")
# corrplot(cor(df), method = "color", type = "upper", order = "hclust")
# covidNum <- df
```
```

Scatter plot vizualization

```
```{r echo=TRUE}
summary(corrCovid)

ggplot(corrCovid, aes(Age,socialMediaTime)) + geom_point()

ggplot(corrCovid, aes(socialPlatform, Age)) + geom_point()

ggplot(corrCovid, aes(Age, fitnessTime)) + geom_point()

ggplot(corrCovid, aes(Age, studyTime)) + geom_point()

ggplot(corrCovid, aes(Age, sleepTime)) + geom_point()

ggplot(corrCovid, aes(Age, healthIssue)) + geom_point()

ggplot(corrCovid, aes(healthIssue, sleepTime)) + geom_point()

ggplot(corrCovid, aes(healthIssue, fitnessTime)) + geom_point()

ggplot(corrCovid, aes(socialMediaTime, PersonConnection)) + geom_point()

ggplot(corrCovid, aes(classTime,socialMediaTime)) + geom_point()

ggplot(corrCovid, aes(socialPlatform, classTime)) + geom_point()

ggplot(corrCovid, aes(classTime, fitnessTime)) + geom_point()

ggplot(corrCovid, aes(classTime, studyTime)) + geom_point()

ggplot(corrCovid, aes(classTime, sleepTime)) + geom_point()

ggplot(corrCovid, aes(classTime, healthIssue)) + geom_point()

```
```

Exploration summary:

 # VARIABLE TRANSFORMATIONS

We tried multiple data tranformations in hopes of increasing our dataset correlations. We removed outliars, we log transformed continuous variables, and we attempted to bucket some continuous variables.

removing outliers outside of IQR

```
``` {r echo=TRUE}
#removing classtime outliers
quartiles<- quantile(covidNum[,c('classTime')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('classTime')])

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(covidNum,
                          covidNum[,c('classTime')] > Lower & covidNum[,c('classTime')] < Upper)

#removing studyTime outliers

quartiles<- quantile(covidNum[,c('studyTime')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('studyTime')])
```
```

```
Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(data_wo_outliers,
                  data_wo_outliers[,c('studyTime')] > Lower & data_wo_outliers[,c('studyTime')] < Upper)


#removing fitnessTime outliers
quartiles<- quantile(covidNum[,c('fitnessTime')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('fitnessTime')])

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(data_wo_outliers,
                  data_wo_outliers[,c('fitnessTime')] > Lower & data_wo_outliers[,c('fitnessTime')] < Upper)


#removing sleepTime outliers
quartiles<- quantile(covidNum[,c('sleepTime')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('sleepTime')])

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(data_wo_outliers,
                  data_wo_outliers[,c('sleepTime')] > Lower & data_wo_outliers[,c('sleepTime')] < Upper)


#removing socMedTIme outliers
quartiles<- quantile(covidNum[,c('socialMediaTime')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('socialMediaTime')])

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(data_wo_outliers,
                  data_wo_outliers[,c('socialMediaTime')] > Lower & data_wo_outliers[,c('socialMediaTime')] < Upper)


#removing tvtime outliers
quartiles<- quantile(covidNum[,c('tvTime')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('tvTime')])

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(data_wo_outliers,
                  data_wo_outliers[,c('tvTime')] > Lower & data_wo_outliers[,c('tvTime')] < Upper)

#removing numMeals outliers
quartiles<- quantile(covidNum[,c('numMeals')], probs= c(.25, .75))
IQR <- IQR(covidNum[,c('numMeals')])

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

data_wo_outliers<- subset(data_wo_outliers,
                  data_wo_outliers[,c('numMeals')] > Lower & data_wo_outliers[,c('numMeals')] < Upper)

```

#Log transforming highly skewed variables

```{r echo=TRUE}
data_log <- covidNum

logvariables <- names(covidNum)


for (variable in logvariables) {
  if (is.numeric(data_log[[variable]]) && !anyNA(data_log[[variable]])) {
    data_log[[variable]] <- log(data_log[[variable]] + 1)
  } else {
    # Handle non-numeric or missing values
    # You can choose to skip, impute, or handle them based on your specific requirements
    print(paste("Skipping variable:", variable))
```

```
 }
}

head(data_log)
```

#checking distributions without outliers

```{r echo=TRUE}

hist(data_wo_outliers$Age)
hist(data_wo_outliers$classTime)
hist(data_wo_outliers$classRating)
hist(data_wo_outliers$studyTime)
hist(data_wo_outliers$fitnessTime)
hist(data_wo_outliers$sleepTime)
hist(data_wo_outliers$socialMediaTime)
hist(data_wo_outliers$tvTime)
hist(data_wo_outliers$numMeals)
hist(data_wo_outliers$timeUtilized)
hist(data_wo_outliers$weightChange)
hist(data_wo_outliers$Medium)
hist(data_wo_outliers$PersonConnection)

```

#checking distributions with log distibution

```{r echo=TRUE}
hist(data_log$Age)
hist(data_log$classTime)
hist(data_log$classRating)
hist(data_log$studyTime)
hist(data_log$fitnessTime)
hist(data_log$sleepTime)
hist(data_log$socialMediaTime)
hist(data_log$tvTime)
hist(data_log$numMeals)
hist(data_log$timeUtilized)
hist(data_log$weightChange)
hist(data_log$Medium)
hist(data_log$PersonConnection)

```

binning variables to check to tackle lack of correlation

```{r echo=TRUE}

CovNumBin <- covidNum

CovNumBin$ageBins <- cut(CovNumBin$Age, breaks = 5, labels = FALSE, include.lowest = TRUE)

CovNumBin$sleepTimeBin <- cut(CovNumBin$sleepTime, breaks = 5, labels = FALSE, include.lowest = TRUE)

CovNumBin$fitnessBins <- cut(CovNumBin$fitnessTime, breaks = 3, labels = FALSE, include.lowest = TRUE)

CovNumBin$SMbin <- cut(CovNumBin$socialMediaTime, breaks = 3, labels = FALSE, include.lowest = TRUE)

CovNumBin$tvBin <- cut(CovNumBin$tvTime, breaks = 3, labels = FALSE, include.lowest = TRUE)

CovNumBin$classBin <- cut(CovNumBin$classTime, breaks = 4, labels = FALSE, include.lowest = TRUE)

colNonBin <- c("Age","sleepTime","fitnessTime", "socialMediaTime","tvTime","classTime")

CovNumBin <- CovNumBin[ , !(names(CovNumBin) %in% colNonBin)]


#CovNumBin <- sapply(CovNumBin,as.numeric)
#CovNumBin<- as.data.frame(CovNumBin)

summary(CovNumBin)

```

checking log transformed variables

```{r echo=TRUE}

CovNumLogBin <- data_log
```

```
CovNumLogBin$ageBins <- cut(CovNumLogBin$Age, breaks = 5, labels = FALSE, include.lowest = TRUE)

CovNumLogBin$sleepTimeBin <- cut(CovNumLogBin$sleepTime, breaks = 5, labels = FALSE, include.lowest = TRUE)

CovNumLogBin$fitnessBins <- cut(CovNumLogBin$fitnessTime, breaks = 3, labels = FALSE, include.lowest = TRUE)

CovNumLogBin$SMbin <- cut(CovNumLogBin$socialMediaTime, breaks = 3, labels = FALSE, include.lowest = TRUE)

CovNumLogBin$tvBin <- cut(CovNumLogBin$tvTime, breaks = 3, labels = FALSE, include.lowest = TRUE)

CovNumLogBin$classBin <- cut(CovNumLogBin$classTime, breaks = 4, labels = FALSE, include.lowest = TRUE)


colNonLogBin <- c("Age","sleepTime","fitnessTime", "socialMediaTime","tvTime","classTime")


CovNumLogBin <- CovNumLogBin[ , !(names(CovNumLogBin) %in% colNonLogBin)]


#CovNumLogBin <- sapply(CovNumLogBin,as.numeric)
#CovNumLogBin<- as.data.frame(CovNumLogBin)

summary(CovNumLogBin)

```
```{r echo=TRUE}

corrplot(cor(as.data.frame(sapply(covidNum[,-16],as.numeric))))

corrplot(cor(as.data.frame(sapply(CovNumBin[,-10],as.numeric))))

corrplot(cor(as.data.frame(sapply(data_log[,-16],as.numeric))))

corrplot(cor(as.data.frame(sapply(CovNumLogBin[,-10],as.numeric))))
```

Transformation summary:

The tranformations had limited improvement in correlation.

#FACTOR ANALYSIS

```
``` {r echo=TRUE, warning = FALSE}
############################
## Polychoric PCA factor analysis  (Areli) -Further Analysis
##
## Polychoric Correlation / PCA definition
## Define: Measures correlation between two unobserved continuous variables, have bivariate normal distribution.
## Each unobserved variable, you can get from an observed ordinal variable. Polychoric correlation is between two
## observed binary variables; also know as tetrachoric correlation.
## in group chat, mentioned doesn't make sense to do a log transform and bin. So will use just the log transform
## using log transform and creating as numeric dataframe

##Set up using data log transform, setting as numeric and data frame
polyPCA <- data_log
polyPCA <- sapply(polyPCA,as.numeric)
polyPCA <- as.data.frame(polyPCA)
###################### PCA ######################
pca = prcomp(polyPCA) # removed scaling
pca
summary(pca)
### Plotting PCA
plot(pca)
abline(1, 0, col="red")
### Parallel Analysis###
parallel_pca = fa.parallel(polyPCA, n.iter=250)
parallel_pca
## eigenvlaues
pp = princomp(polyPCA)
### Plotting PCA
plot(pp)
abline(1, 0, col="red")
print(pp)
plot(pp)
summary(pp)
###################### PolyChor ######################
```
```

```
library(polycor)
corr_matrix <- hetcor(polyPCA, ML=TRUE)
corr_matrix
summary(pca)
plot(polyPCA)
## function polychor
polychor(polyPCA$healthIssue, polyPCA$Age, ML = FALSE, control = list(),
      std.err = FALSE, maxcor=.9999, start, thresholds=FALSE)
###################### Principal ######################
library(MASS)
## Run principal() with the corr matrix with and w/o rotation
# without rotation
p1 <- principal(polyPCA, nfactors=6)
# with rotation
p2 <- principal(polyPCA, rotate = "varimax", nfactors=6)
# print
print(p1$loadings, cutoff = 0.4, sort=T)
print(p2$loadings, cutoff = 0.4, sort=T)
plot(p1)
plot(p2)
###################### Plots ######################
## Age, studyTime, fitnessTime, sleepTime, socialMediaTime
p3 <- prcomp(polyPCA[,c(1,5,6,7)], center = TRUE) ## not scaling
plot(p3)
print(p3)
##head(p2$x,20)
summary(p3)

## Plot 1 using autoplot()
install.packages("ggfortify")
library(ggfortify)
covid.pca.plot <- autoplot(p3,
                    data = polyPCA,
                    colour = 'classRating')
covid.pca.plot
## Plot 2 using biplot()
biplot.covid.pca <- biplot(p3)
biplot.covid.pca
## Plot 3 using overall plots of columns
plot.covid.pca <- plot(p3)
plot.covid.pca
## Plot 4 using fviz_pca_var()
fviz_pca_var(p3, col.var = "black")

```

REGRESSIONS (OLS, and Elastic Net)

plotting initial OLS fit

```
``` {r echo=TRUE}
OLS.init <- lm(classTime ~ ., data = covidNum)
summary(OLS.init)

OLS.init2 <- lm(timeUtilized ~ ., data = covidNum)
summary(OLS.init2)

vif(OLS.init) #no multicollinearity confirmed
#0.1 R2 with 8 significant variables.

par(mfrow = c(2,2))
plot(OLS.init)
plot(OLS.init2)
par(mfrow = c(1,1))
```
```

OLS with data_log

```
``` {r echo=TRUE}
OLS.init3 <- lm(classTime ~ ., data = data_log[,-4])
summary(OLS.init3)
vif(OLS.init3)

OLS.init4 <- lm(timeUtilized ~ ., data = data_log)
summary(OLS.init4)

vif(OLS.init3) #no multicollinearity confirmed
```
```

```
#0.1 R2 with 8 significant variables.
par(mfrow = c(2,2))
plot(OLS.init3)
plot(OLS.init4)
par(mfrow = c(1,1))
```

Doing an all subsets regression

```{r echo=TRUE}
library(caret)
ctrl <- trainControl(method = "cv", number = 10)

#fit a regression model and use k-fold CV to evaluate performance
model <- train(classTime ~. , data = data_log[,-4], method = "lm", trControl = ctrl)
model$finalModel
model$resample

trainCovid <- data_log[sample,-c(4)]
testCovid <- data_log[!sample,-c(4)]

train_preds <- predict(model, trainCovid)
test_preds <- predict(model, testCovid)

# Calculate the training and testing set RMSEs
train_rmse <- sqrt(mean((data_log$classTime - train_preds)^2))
test_rmse <- sqrt(mean((data_log$classTime - test_preds)^2))

# Print the results
cat("Training set RMSE:", train_rmse, "\n")
cat("Testing set RMSE:", test_rmse, "\n")

coefficients <- coef(model$finalModel)
p_values <- summary(model$finalModel)$coefficients[, "Pr(>|t|)"]

# Combine the coefficient estimates and p-values into a data frame
results <- data.frame(Coefficient = coefficients, `p-value` = p_values)

# Print the results
print(results)

print(model)
```

fitting an alpha to an elastic plot.

```{ echo= TRUE}
sample <- sample(c(TRUE,FALSE), nrow(data_log), replace = TRUE, prob=c(0.7,0.3))
trainCovid <- data_log[sample,-c(4,16)]
testCovid <- data_log[!sample,-c(4,16)]

xTrain.d1<- as.matrix(trainCovid[,c(-2)])
yTrain.d1<- as.matrix(trainCovid[,c(2)])

xTest.d1 <- as.matrix(testCovid[,c(-2)])
yTest.d1 <- as.matrix(testCovid[,c(2)])


set.seed(17289)

alphaBest = 0
bestError = 9999999    # Start out with a huge error
for (alpha in seq(0, 1, .1))
{
  meanError = 0
  for (i in 1:50)
  {
    # Grab test and training sets
    fitElastic.vid = cv.glmnet(xTrain.d1, yTrain.d1, alpha=alpha, nfolds=10,
                               grouped = FALSE)
    elasticPred = predict(fitElastic.vid, xTest.d1, s="lambda.1se")
    meanError = meanError + sqrt(mean((elasticPred - yTest.d1)^2))
  }
  meanError = meanError / 100
```

```
 if (meanError < bestError)
 {
   alphaBest = alpha
   bestError = meanError
 }
}
print("Best alpha is: ")
print(alphaBest) #gave it as 0
print("Gives mean test error: ")
print(bestError) #very little overfitting


#running with best alpha

lamRange = seq(0,3,0.1)

Best.eNet <- cv.glmnet(xTrain.d1, yTrain.d1, alpha=0.0, nfolds = 7, grouped = FALSE)

elasticPred.test = predict(Best.eNet, xTest.d1, s="lambda.1se")
elasticPred.train =  predict(Best.eNet, xTrain.d1, s="lambda.1se")

rmse.elasticTest = sqrt(mean((elasticPred.test - yTest.d1)^2))
rmse.elasticTrain = sqrt(mean((elasticPred.train - yTrain.d1)^2))

rmse.elasticTrain
rmse.elasticTest

elasticPred.test = predict(model, xTest.d1)
elasticPred.train =  predict(model, xTrain.d1)

rmse.elasticTest = sqrt(mean((elasticPred.test - yTest.d1)^2))
rmse.elasticTrain = sqrt(mean((elasticPred.train - yTrain.d1)^2))

rmse.elasticTrain
rmse.elasticTest

Best.eNet
#very little overfitting! but HOW???


#checking lasso to confirm we cant have penalized regression
BestELasso <- cv.glmnet(xTrain.d1, yTrain.d1, alpha=1, lambda = lamRange)
plot(BestELasso)
BestELasso

plot(Best.eNet)

Best.eNet$lambda.1se
```

LDA Analysis

```
```{r include= FALSE}
##this code block is a function provided by Prof to calculate contingency matrix accuracy for LDA

confusion = function(actual, predicted, names = NULL, printit = TRUE, prior = NULL)
{
 if (is.null(names))
   names = levels(actual)
 tab = table(actual, predicted)
 acctab = t(apply(tab, 1, function(x) x/sum(x)))
 dimnames(acctab) = list(Actual = names, "Predicted (cv)" = names)
 if (is.null(prior))
 {
   relnum = table(actual)
   prior = relnum/sum(relnum)
   acc = sum(tab[row(tab) == col(tab)])/sum(tab)
 }
 else
 {
   acc = sum(prior * diag(acctab))
   names(prior) = names
 }

 print(round(c("Accuracy" = acc, "Prior Frequency" = prior), 4))
```
```

```
 cat("\nConfusion Matrix", "\n")
 print(round(acctab, 4))
}
```

```

``` {r echo = TRUE}
c <- covidNum[,c(1,2,3,5,6,7,8,11)] #is there a reason you didn't include tvTime aka 11th column
head(c)

c <- log((c[,])+1)
head(c)

#starting with PCA for these
p=prcomp(c)
summary(p)
#I think an issue here is that we need at least 5 PCs

#adding class back in
cp <- as.data.frame(p$x)
cp$Miss <- covidNum$Miss

cp$Miss <- as.factor(cp$Miss)

plot(cp$PC1, cp$PC2, col=cp$Miss, pch=1, cex=.5, xlab="PC1", ylab="PC2")

#LDA - commenting out this as it didn't work, leaving it to show the attempt.
# fit = lda(sleepBin ~ studyTime, data=c)
# print(fit)
# plot(fit)
#I have tried: social media and study time to predict ageBin, studytime and fitnessTime to predict sleep Bin,
#and fitnessTime, studyTime, socialMediaTime to predict timeUtilized. none have been successful.


#now I am trying to use all the continuous variables to predict the
#categories Miss (reclassed earlier) --> was very promising, until my computer shut down :( ) or Medium (of the course)
#LDA session two!! #woohoo it works

c$Miss <- as.factor(covidNum$Miss)
fit = lda(Miss ~ ., data=c)
print(fit)
plot(fit)



#plotting the fit using the LDA
ldaResult = predict(fit)
plot(ldaResult$x, col=ldaResult$class)
#ldahist(ldaResult$x[,1], g=class)
ldaResult$class

#confusion matrix - room for improvement
table(c$Miss, ldaResult$class)

#computing accuracy of the confusion matrix
confusion(c$Miss, ldaResult$class)


#Caclulate Distance matrix
covid.dist <- dist(c[,1:8])

#Calculate MDS - k here is the number of dimensions
covid.mds <- cmdscale(covid.dist, k=3)

#Create x,y,z axes
data.x <- covid.mds[,1]
data.y <- covid.mds[,2]
data.z <- covid.mds[,3]

#Plotting in 3-d and coloring based on 'Miss' category
#You can click on the plot and drag to rotate it
plot3d(
  x=data.x, y=data.y, z=data.z,
  col = unclass(as.factor(c$Miss)),
```
```

```
  type = 's',
  radius = .1,
)

#let's show it again with LDA!!!!
#Create x,y,z axes
data.x <- ldaResult$x[,1]
data.y <- ldaResult$x[,2]
data.z <- ldaResult$x[,3]

#You can see the true classes jumble in the middle. I'll comment on this in the techinical summary
plot3d(
  x=data.x, y=data.y, z=data.z,
  col = unclass(as.factor(c$Miss)), #can sub out c$Miss with ldaResult$class to see lda plot predictions in 3d
  type = 's',
  radius = .1,
)

```
```{r include=FALSE}
c <- covidNum[,c(2,3,5,6,7,8)] #is there a reason you didn't include tvTime aka 11th column
head(c)

c <- log((c[,])+1)
head(c)

#starting with PCA for these
p=prcomp(c)
summary(p)
#I think an issue here is that we need at least 5 PCs

#adding class back in
cp <- as.data.frame(p$x)
cp$Miss <- covidNum$Miss

cp$Miss <- as.factor(cp$Miss)

plot(cp$PC1, cp$PC2, col=cp$Miss, pch=1, cex=.5, xlab="PC1", ylab="PC2")

#LDA - commenting out this as it didn't work, leaving it to show the attempt.
# fit = lda(sleepBin ~ studyTime, data=c)
# print(fit)
# plot(fit)
#I have tried: social media and study time to predict ageBin, studytime and fitnessTime to predict sleep Bin,
#and fitnessTime, studyTime, socialMediaTime to predict timeUtilized. none have been successful.


#now I am trying to use all the continuous variables to predict the
#categories Miss (reclassed earlier) --> was very promising, until my computer shut down :( ) or Medium (of the course)
#LDA session two!! #woohoo it works

c$Miss <- as.factor(covidNum$Miss)
fit = lda(Miss ~ ., data=c)
print(fit)
plot(fit)

#plotting the fit using the LDA
ldaResult = predict(fit)
plot(ldaResult$x, col=ldaResult$class)
#ldahist(ldaResult$x[,1], g=class)
ldaResult$class

#confusion matrix - room for improvement
table(c$Miss, ldaResult$class)

#computing accuracy of the confusion matrix
confusion(c$Miss, ldaResult$class)

covid.dist <- dist(c[,1:6])

#Calculate MDS - k here is the number of dimensions
covid.mds <- cmdscale(covid.dist, k=3)

#Create x,y,z axes
data.x <- covid.mds[,1]
```

```
data.y <- covid.mds[,2]
data.z <- covid.mds[,3]

#Plotting in 3-d and coloring based on 'Miss' category
#You can click on the plot and drag to rotate it
plot3d(
  x=data.x, y=data.y, z=data.z,
  col = unclass(as.factor(c$Miss)),
  type = 's',
  radius = .1,
)

#let's show it again with LDA!!!!
#Create x,y,z axes
data.x <- ldaResult$x[,1]
data.y <- ldaResult$x[,2]
data.z <- ldaResult$x[,3]

#You can see the true classes jumble in the middle. I'll comment on this in the techinical summary
plot3d(
  x=data.x, y=data.y, z=data.z,
  col = unclass(as.factor(c$Miss)), #can sub out c$Miss with ldaResult$class to see lda plot predictions in 3d
  type = 's',
  radius = .1,
)
```

Extra Credit: Bayesian Network

```{r}
library(bnviewer)
library(bnlearn)
```

```{r echo = TRUE}
CovNumBin <- covidNum[-c(515,195,466),-c(4,9,16)] #those rows skewed the binning of media time and fitnesstime

#CovNum_logged <- log((CovNumBinXX[,])+1)

age_bin_labels <- c("7-17", "18-27", "28-38", "39-46", "50-59")
```

Binning the continuous variables used in analysis

```
CovNumBin$ageBins <- cut(CovNumBin$Age, breaks = 5, labels = age_bin_labels, include.lowest = TRUE)

# #View the values in each bin
# bins <- split(CovNumBin$Age, CovNumBin$ageBins)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

sleepTime_bins <- c("4-7.5","7.8-11", "12-15")

CovNumBin$sleepTimeBin <- cut(CovNumBin$sleepTime, breaks = 3, labels = sleepTime_bins, include.lowest = TRUE)

# View the values in each bin
# bins <- split(CovNumBin$sleepTime, CovNumBin$sleepTimeBin)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

fit_bins <- c("0-1","2","4-5")
```

```
CovNumBin$fitnessBins <- cut(CovNumBin$fitnessTime, breaks = 3, labels = fit_bins, include.lowest = TRUE)

# # View the values in each bin
# bins <- split(CovNumBin$fitnessTime, CovNumBin$fitnessBins)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

# bin_thresholds <- quantile(CovNumBin$fitnessTime, probs = seq(0, 1, 1/2))
#
# # Create a new variable with custom bins
# CovNumBin$fitnessBins2<- cut(CovNumBin$fitnessTime, breaks = bin_thresholds, labels = FALSE, include.lowest = TRUE)

#bin_thresholds <- quantile(CovNumBin$socialMediaTime+CovNumBin$tvTime, probs = seq(0, 1, 1/3))

# Create a new variable with custom bins
# CovNumBin$TTLmediaBin2 <- cut(CovNumBin$socialMediaTime+CovNumBin$tvTime, breaks = bin_thresholds, labels = FALSE, include.lowest = TRUE)

# bins <- split(CovNumBin$socialMediaTime+CovNumBin$tvTime, CovNumBin$TTLmediaBin2)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

smTime_bins <- c("0-4.75","5.0-9.5","10-14.5")

CovNumBin$TTLmediaBin <- cut(CovNumBin$socialMediaTime+CovNumBin$tvTime, breaks = 3, labels = smTime_bins, include.lowest = TRUE)

# View the values in each bin
# bins <- split(CovNumBin$socialMediaTime+CovNumBin$tvTime, CovNumBin$TTLmediaBin)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

classTBins <- c("0-7.3","7.5-14", "15-22")

CovNumBin$classStudyBin <- cut(CovNumBin$classTime+CovNumBin$studyTime, breaks = 3, labels = classTBins, include.lowest = TRUE)

# View the values in each bin
# bins <- split(CovNumBin$classTime+CovNumBin$studyTime, CovNumBin$classStudyBin)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

# studyBins <- c("0-6", "6.5-12", "17-18")
# CovNumBin$studyBin <- cut(CovNumBin$studyTime, breaks = 3, labels = studyBins , include.lowest = TRUE)

# # View the values in each bin
# bins <- split(CovNumBin$studyTime, CovNumBin$studyBin)
#
# # Calculate and print the minimum and maximum values within each bin
# bin_ranges <- sapply(bins, function(bin) {
```

```
#   min_value <- min(bin)
#   max_value <- max(bin)
#   c(min_value, max_value)
# })
#
# print(bin_ranges)

colNonBin <- c("Age","studyTime","sleepTime","fitnessTime", "socialMediaTime","tvTime","classTime")

CovNumBin <- CovNumBin[ , !(names(CovNumBin) %in% colNonBin)]

#CovNumBin <- sapply(CovNumBin,as.numeric)
#CovNumBin<- as.data.frame(CovNumBin)

#remove Medium as this is not relative
#CovNumBin <- CovNumBin[,-2]
#head(CovNumBin)

#for Bayesian Network I need to transform the categorial variables into ordinals so we use unclass

#CovNumBin$socialPlatform <- unclass(as.factor(CovNumBin$socialPlatform))
#CovNumBin$Miss <- unclass(as.factor(CovNumBin$Miss))

#CovNumBin$socialPlatform <- as.numeric(CovNumBin$socialPlatform)
#CovNumBin$Miss<- as.numeric(CovNumBin$Miss)

head(CovNumBin)
summary(CovNumBin)
sapply(CovNumBin,class)
```

#data is ready to be used for Bayesian Network

```
#lets use hc(): Hill Climbing
#generates a model of non-character variable relationships for fitting Bayesian network

CovNumBinBN<- CovNumBin

for (var in names(CovNumBinBN[,c(7,8,9,10,11)])) {
  CovNumBinBN[[var]] <- unclass(as.factor(CovNumBin[[var]]))
}

# for (var in names(CovNumBinBN)) {
#   CovNumBinBN[[var]] <- as.numeric(CovNumBinBN[[var]])
# }


for (var in names(CovNumBinBN)) {
  CovNumBinBN[[var]] <- as.factor(CovNumBinBN[[var]])
}

sapply(CovNumBinBN,class)

summary(CovNumBinBN)

VarRel = hc(CovNumBinBN)
plot(VarRel)
```

#after we transformed the data via binning we see that that the relationships are very incorrect
#for example, ageBins should not be able to be influenced wheres classRating should be dependent on other factors.
#numMeals variable relationships seem to be plausible, as well as the fact that time utilized variable is influenced by many factors.

#lets fit an initial BNetwork and see the created relationship coefficients.

```
# if (!require("BiocManager", quietly = TRUE))
#   install.packages("BiocManager")
#
# BiocManager::install("Rgraphviz")
#
#
# install.packages("gRain")

fittedBN <- bn.fit(VarRel, CovNumBinBN)
fittedBN
```

```
# graphviz.chart(
#   fittedBN,
#   type = "barprob",
#   layout = "fdp",
#   scale = c(0.75,0.9),
#   grid = TRUE,
#   bar.col = "red",
#   strip.bg = "lightskyblue"
# )

#Thankfully we can rewrite the model manually.
#This is one of the most useful capabilites of Bayesian Networks:
#because we can manually adjust the network's view of relationships, we can create
#a much more precise model that is the most interpertable for this dataset.

#using model2network we adjust the variable relationships
#the variables that are alone in the square brackets are independent, they are not influencable
#other variables combinations are and are specificied with a  |. left of the | is the child right is the parent.
#the parent influences the child.
#using insights we found in Polychoric analysis we can focus on the specific relationships that interest us.

# varRelations<-paste("[ageBins][numMeals][socialPlatform|ageBins]
# [sleepTimeBin|TTLmediaBin:ageBins]
# [studyBin|TTLmediaBin:fitnessBins:ageBins]
# [classBin|TTLmediaBin:fitnessBins:ageBins]
# [TTLmediaBin|socialPlatform:ageBins]
# [fitnessBins|numMeals:healthIssue]
# [classRating|TTLmediaBin:studyBin:fitnessBins:classBin:PersonConnection:Miss]
# [Miss|ageBins:socialPlatform:classBin:fitnessBin:studyBin]
# [timeUtilized|PersonConnection:TTLmediaBin:sleepTimeBin:fitnessBins:classBin:classRating:healthIssue:studyBin:socialPlatform]
# [healthIssue|ageBins:numMeals:fitnessBins:sleepTimeBin:TTLmediaBin:weightChange:PersonConnection]
# [weightChange|numMeals:fitnessBins]
# [sleepTimeBin|fitnessBins:SocialPlatform]
# [PersonConnection|ageBins:TTLmediaBin:healthIssue]")
#


#i failed to include a model for healthIssue because it cause cyclicity, whereas Bayes Network must be acyclic
res = model2network("[ageBins]
            [numMeals]
            [fitnessBins]
            [healthIssue|fitnessBins:numMeals:PersonConnection:TTLmediaBin:sleepTimeBin][sleepTimeBin|TTLmediaBin:ageBins:fitnessBins]
            [studyBin|TTLmediaBin:fitnessBins:ageBins:sleepTimeBin][classBin|TTLmediaBin:fitnessBins:ageBins][TTLmediaBin|ageBins]
            [classRating|TTLmediaBin:sleepTimeBin:studyBin:fitnessBins:classBin:PersonConnection]
            [timeUtilized|PersonConnection:TTLmediaBin:sleepTimeBin:fitnessBins:classBin:classRating:healthIssue:studyBin]
            [weightChange|numMeals:fitnessBins]
            [PersonConnection|ageBins:TTLmediaBin]", debug=TRUE)


res2 =
model2network("[ageBins][numMeals][fitnessBins][healthIssue|fitnessBins:numMeals:PersonConnection:TTLmediaBin:sleepTimeBin][sleepTimeBin|TTLmediaBin:a
geBins:fitnessBins][studyBin|TTLmediaBin:fitnessBins:ageBins:sleepTimeBin][classBin|TTLmediaBin:fitnessBins:ageBins][TTLmediaBin|ageBins][weightChange|num
Meals:fitnessBins][PersonConnection|ageBins:TTLmediaBin]", debug=TRUE)
```

Code used to generate the model in this paper:

```
res3 = model2network("[ageBins]
            [fitnessBins]
            [TTLmediaBin]
            [PersonConnection]
            [healthIssue]
            [classRating]
            [sleepTimeBin]
            [studyBin][classBin|TTLmediaBin:fitnessBins:ageBins:healthIssue:PersonConnection:classRating:sleepTimeBin:studyBin]")

res4 =
model2network("[fitnessBins][TTLmediaBin][healthIssue|sleepTimeBin:fitnessBins:TTLmediaBin][sleepTimeBin|classBin:fitnessBins][classBin|TTLmediaBin]")

plot(res)
fittedBN <- bn.fit(res, CovNumBinBN)
fittedBN

plot(res2)
fittedBN <- bn.fit(res2, CovNumBinBN[,-c(1,5)])
```

```
fittedBN

plot(res3)
fittedBN <- bn.fit(res3, CovNumBinBN[,-c(2,3,5)])
fittedBN


#network setup / Goodness of fit confirmation

VarRelX <- hc(CovNumBinBN[,c(8,9,10,11)])
plot(VarRelX)


BST <- boot.strength(CovNumBinBN[,c(8,9,10,11)], R = 200,algorithm = "pc.stable",debug = TRUE,cpdag = TRUE)
# compute the edge strengths
avgnet1 <- averaged.network(BST,threshold = 0.85)
# compute the average network

plot(avgnet1)

BNpc <- pc.stable(CovNumBinBN[,c(8,9,10,11)])

plot(BNpc)


#res6<- model2network("[sleepTimeBin][fitnessBins][TTLmediaBin][classStudyBin|sleepTimeBin:TTLmediaBin]")


BNgs <- gs(CovNumBinBN[,c(8,9,10,11)])
# this performs the GS algorithm
BNiamb <- iamb(CovNumBinBN[,c(8,9,10,11)])
# this performs the IAMB algorithm




   #set network directions manually
res5 = model2network("[fitnessBins|TTLmediaBin][TTLmediaBin][sleepTimeBin|TTLmediaBin][classStudyBin|TTLmediaBin:fitnessBins:sleepTimeBin]")

plot(res5)

#visualization of Bayesian Network

bnviewer::viewer(res5,
     bayesianNetwork.width = "100%",
     bayesianNetwork.height = "80vh",
     bayesianNetwork.layout = "layout_with_sugiyama",
     bayesianNetwork.title="Covid-19 Education Survey Network",
     bayesianNetwork.footer = "Fig. 1 - Layout with Sugiyama"
)

#the conditional probabilities and overview of filterered data

# plot(res4)
# fittedBN <- bn.fit(res4, CovNumBinBN[,-c(1,2,3,5,6,7,10)])
# fittedBN

filtered_dataset1 <- CovNumBinBN %>% filter(ageBins == "1")
summary(filtered_dataset1[,c(8,9,10,11)])
plot(res5)
fittedBN1 <- bn.fit(res5, filtered_dataset1[,c(8,9,10,11)])
fittedBN1

filtered_dataset2 <- CovNumBinBN %>% filter(ageBins == "2")
summary(filtered_dataset2[,c(8,9,10,11)])
#plot(res5)
fittedBN2 <- bn.fit(res5, filtered_dataset2[,c(8,9,10,11)])
fittedBN2



#goodness of fit

BST <- boot.strength(CovNumBinBN[,c(8,9,10,11)], R = 200,algorithm = "pc.stable",debug = TRUE,cpdag = TRUE)
# compute the edge strengths
```

```
avgnet1 <- averaged.network(BST,threshold = 0.85)
# compute the average network
astr1 <- arc.strength(res5,CovNumBinBN[,c(8,9,10,11)],"bic-g")
# compute edge strengths



graphviz.chart(
  fittedBN1,
  type = "barprob" ,
  layout = "neato",
  grid = TRUE,
  bar.col = "darkgreen",
  main = "Conditional Probabilites of BN analysis",
  sub = "Data based on Adolenscent propulation"
)



graphviz.chart(
  fittedBN2,
  type = "barprob" ,
  layout = "neato",
  grid = TRUE,
  bar.col = "darkgreen",
  main = "Conditional Probabilites of BN analysis",
  sub = "Data based on young adults propulation"
)


```