Mental Health in the Work Environment

By Anh, Annabel, Kierra, Rafael, Robby

May 9th, 2022

Introduction:

Mental health includes our emotional, psychological, and social well-being. It affects how we think, feel, and act. It also helps determine how we handle stress, relate to others, and make choices. Mental health is important at every stage of life, from childhood and adolescence through adulthood. Mental Health has become an increasingly important issue for people in the workplace as burnout and worker fatigue are rising and more people than ever are quitting their jobs citing toxic workplace culture and to maintain their mental health. The dataset that our group is using is called the Medical Treatment Dataset from Kaggle. It was created by Shadab Hussain and the data set contains various attributes relating to mental health and aims to predict treatment for each patient in the data set with mental health issues. The datasets will include variables such as S.no which is the ID number for each patient, Timestamp to track the time, Age of the patient, Gender of the patient, Country they are from, State they reside in, whether they are self-employed, if they have a family history of mental health issues, and their number of employees. From this data set, the model(s) is/are created to predict the variable Treatment from the test set to see whether treatment is needed with a "yes" or a "no". We will use exploratory data analysis and several machine learning models to answer the following questions:

Questions of Interest:

1. Are employees older than 40 accustomed to stress in the workplace and not seeking mental health treatment compared to employees younger than 40?
2. Is one gender seeking mental health services in the workplace more than the other?
3. Which model is best in classifying whether or not the participant needs treatment?

4. What are the important/significant factors that determine whether or not an employee needs treatment?

5. Given a person's profile, would our best fitting model predict the person seeking help?

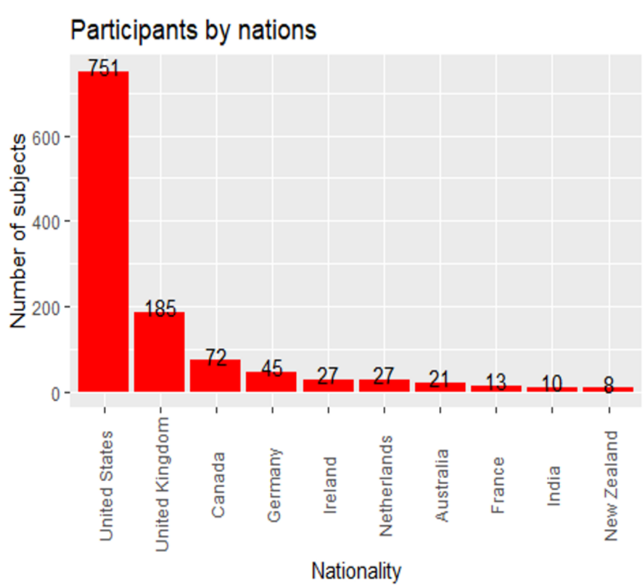
Analysis:

The dataset for this study was downloaded from Kaggle and came in three separate files, which were provided train, test, and a sample file with the treatment column for the test file. We concatenated the 3 files together for the purpose of our study.

Data summary

Name	MH
Number of rows	1259
Number of columns	27
Column type frequency:	
character	25
numeric	1
POSIXct	1
Group variables	None

There are 1259 observations and 27 columns(25 categorical, 1 numeric, and POSIXct variables) in the original dataset.



Our study focuses on the profiles of participants from the US, which accounts for 751 observations in this dataset.

We also removed the country(singular value) and state(too many values). Timestamp was also deleted since our study would not involve time series analysis.

Data summary

Name	MH_US
Number of rows	751
Number of columns	23

751 observations and 23 columns left

Column type frequency:	
character	22
numeric	1

22 character and 1 numeric columns. Character columns would be converted to factor ones.

Group variables	None

skim_variable	n_missing	complete_rate	min	max	empty	n_unique
Gender	0	1.00	1	16	0	30
self_employed	11	0.99	2	3	0	2
family_history	0	1.00	2	3	0	2
treatment	0	1.00	2	3	0	2
work_interfere	144	0.81	5	9	0	4

The skim table displayed that there were several missing values in the self_employed and work_interfere columns. Since these two columns were categorical, we imputed the missing values with the one that had the highest frequency in each column.

```
table(MH_US$self_employed)/nrow(MH_US)
```

```
##
##         No        Yes
## 0.91078562 0.07456724
```

For the self_employed variable, “No” will be the imputed value for the missing ones

```
table(MH_US$work_interfere)/nrow(MH_US)
```

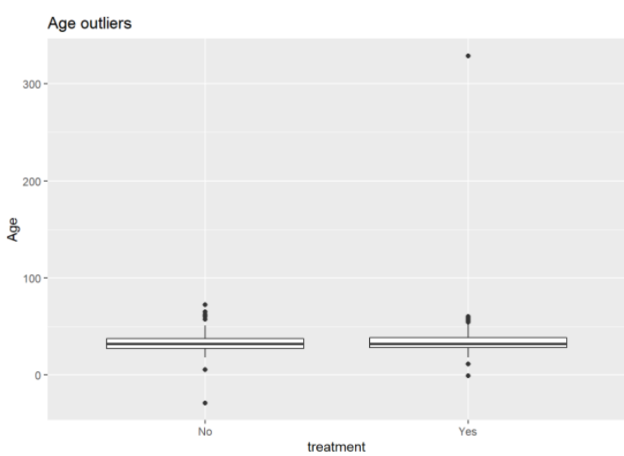
```
##
##     Never     Often    Rarely Sometimes
## 0.1664447 0.1091877 0.1478029 0.3848202
```

For the work_interfere variable, “Sometimes” will be the imputed value for the missing ones. Based on the skim table, we could observe that Gender column had 30 unique values. Therefore, we decided to explore what those values were, instead of the usual male and female values.

```
##
## cis-female/femme      Cis Female        cis male        Cis Male
##                1               1               1               2
##                f               F          femail          Femake
##               12              34               1               1
##           female          Female    Female (cis) Female (trans)
##               43              84               1               2
##      Genderqueer               m               M            Mail
##                1              18              92               1
##            maile            Make            male            Male
##                1               4              90             350
##         Male-ish             Man            msle             Nah
##                1               1               1               1
##       non-binary               p    Trans-female     Trans woman
##                1               1               1               1
##            woman           Woman
##                1               2
```

We can see that male or female values were recorded in several different ways. Some were even misspelled. We will rename the values and group them accordingly into male/female. Also, if neither male(m, msle, make, man,maile..) or female(femail, F, woman..) is specified, we will name the gender as others.

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Age	0	1	33.33	13.53	-29	27.5	32	37.5	329	▇▁▁▁▁



For the only numeric variable, age, we can see that the lowest value is -29 and highest value is 329. We need to check for the outliers in Age.

```
## # A tibble: 5 x 23
##     Age Gender self_employed family_history treatment work_interfere
##   <dbl> <fct>  <fct>         <fct>          <fct>     <fct>
## 1   -29 1      0             0              No        2
## 2   329 1      0             0              Yes       3
## 3     5 1      0             0              No        2
## 4    11 1      1             0              Yes       0
## 5    -1 2      1             1              Yes       3
```

We can see that there are 2 negative values, 1 329 value, and 2 illegal working age value. We will impute these values using the column means of Age.

Our response variable for this study was “treatment”, which consisted of binary values of 0 and 1. Regarding the categorical predictors, firstly, the 2 ordinal columns, which were work_interfere and leave, were encoded using the scale 0/1/2/3.... based on the according increasing level of the values. “Self_employed”, “family_history”, “remote_work”, “tech_company”, “obs_consequece”, “coworkers”, and “supervisors” were the 7 variables with dichotonomous values Yes/No, which were as 1/0. The majority of the categorical variables were nominal with 3 values(Yes/No and the 3rd values), which were “Gender”, "benefits", "care_options", "wellness_program", "seek_help", "anonymity", "mental_health_consequence", "phys_health_consequence", "mental_health_interview", "phys_health_interview", and "mental_vs_physical". We encoded these variables by assigned 1/0 for Yes/No and 2 for the 3rd values.

```
##
##           1-5      100-500       26-100     500-1000         6-25
##            76          113          170           42          134
## More than 1000
##           216
```

There are 6 types of no_employees values, which represents the numbber of employees in the companies of the participants. We encoded 1 to 6 based on the increasing level of the values.

The response and all of the categorical variables were also converted to the factor type.

INFERENCE: Did people who made comments need help?

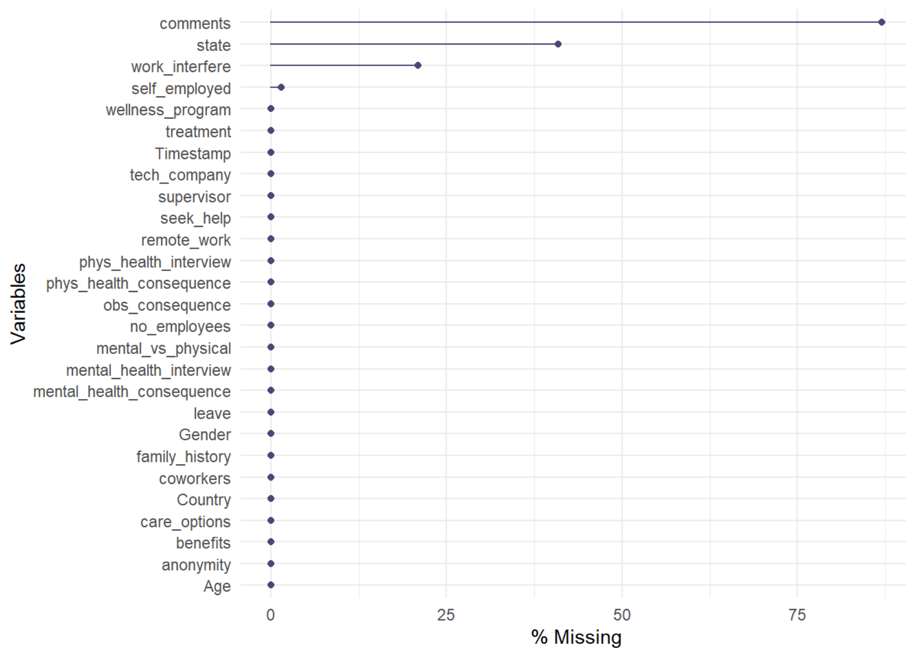


Figure 1. Missing Data

According to Figure 1. There are three variables with a significant amount of mising entries. Work_interfere with a little under 25%, state with about 40% and leading with the largest amount is comments at over 75% missing. This raised the question, did people who make comments need help?

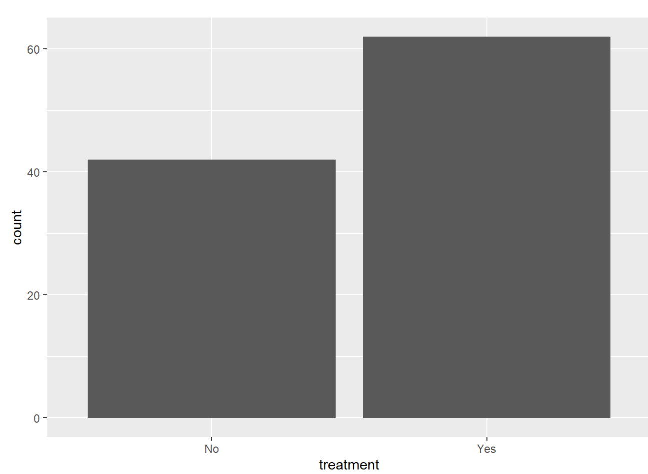


Figure 2. Treatments for Comments made

Figure 2. Gives us the balance for the treatment outcomes for the entries that made comments. We can see that 60% of the people that made comments needed help

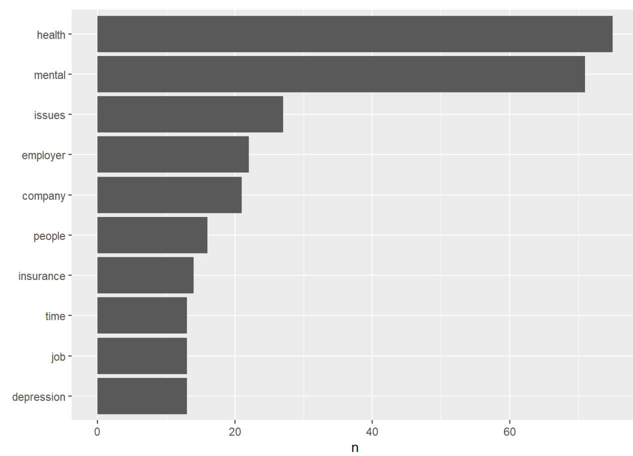
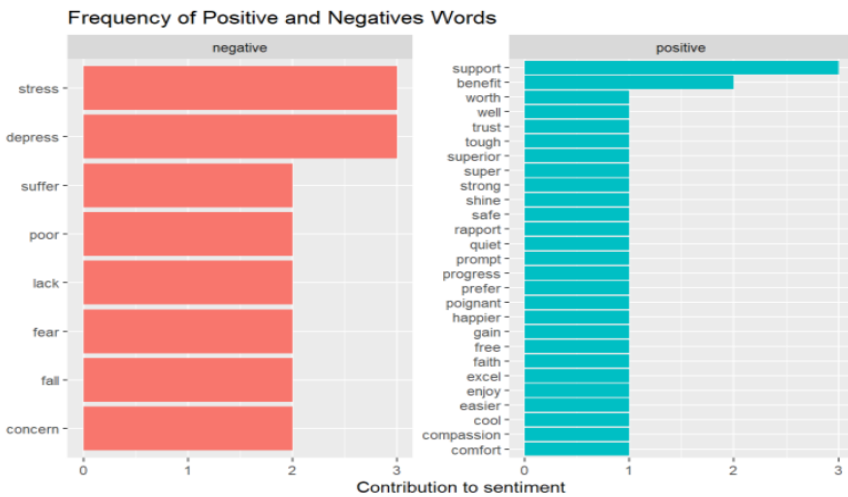


Figure 3. Frequent Word Count

Figure 3. Shows us wich word has the most frequency, we see that metal and health are the top two most frequent words, this could be due to talking about the subject. However, some notable words are insurance, time, job and depression. This means that it could be that people with mental health issues associate their jobs with depression, they are talking about their insurance and time seems to be an issue.

Figure 4. Word Cloud

The above word cloud gives us a bit more insight towards the most frequent word usage. We can see that comapy issues in the tech industry effect our mental health negatively, it gives us depression, anxiety and is affecting our family. We would like help in the form of leave or time off, and perhaps something covered by our insurance.



Many sentiment words have the same frequency. The causes of mental illnesses seem to be stress and depress. Meanwhile, these people need support and health benefit to deal with their mental illnesses.

1. **Are employees older than 40 accustomed to stress in the workplace and not seeking mental health treatment compared to employees younger than 40?**

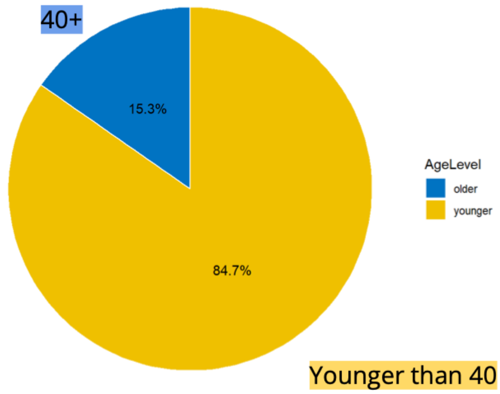


Figure: Distribution of Age

In order to answer the first question, we needed to visualize the data to see if employees older than 40 really are more accustomed to stress in the workplace and not seeking mental health treatment compared to employees younger than 40. In our dataset, we see that most people surveyed fit in the younger demographic with 84.7% of those surveyed being under the age of 40 and only 15.3% surveyed were 40 years old and older.

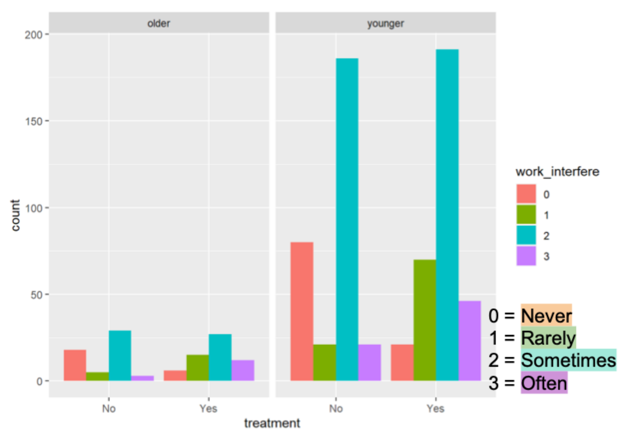
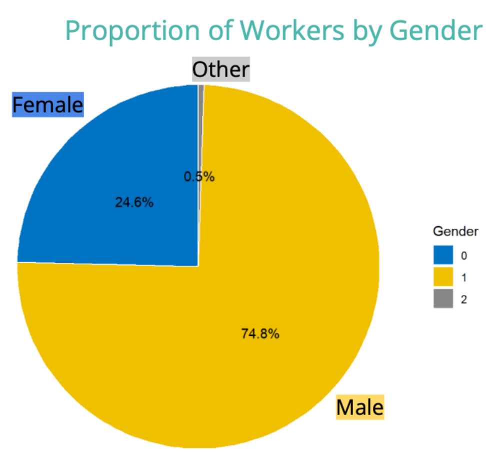


Figure: Treatment and Work interference based on Age

When we view a bar graph, we can compare populations that received treatment and those that did not in each group to see if they reported that their mental health interfered with their work. When this is viewed, we see that age was quite significant in terms of worker interference with younger workers indicating that their mental health interferes with their work sometimes and that younger workers in the treatment group were much less likely to say that mental health never affects their work productivity, and those in the treatment group were more likely to say that their mental health affects their work productivity often than those not in the treatment group. With these observations in mine, models and tests are needed to run in order to see if it is truly significant.

2. **Is one gender seeking mental health services in the workplace more than the other?**

For the second question of interest, we apply these same principles to gender to see if one gender is seeking mental health services in the workplace more than the other. First we need to observe the gender distribution to see the breakdown by gender.



From the pie chart, we can see that in this dataset the vast majority of workers identify as male with the rest being female and a much smaller percentage that did not identify with either. More specifically, 74.8% of respondents identified as male, 24.6% identified as female, and finally 0.5% identified with neither gender. When we view a bar graph comparing gender grouped by their treatment group, we do not see much change.

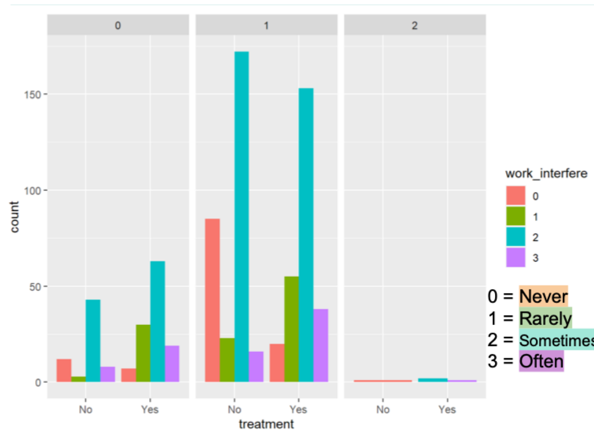
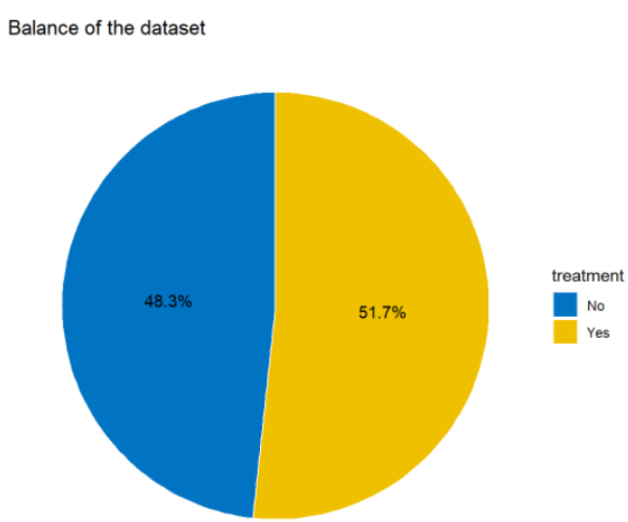


Figure: Treatment and Work Interference based on Gender

From the bar graph, it appears that we cannot conclude the hypothesis we created, as those who need mental health treatment and those who do not, indicate similar levels of work interference. Also, both males and females have similar levels of work interference with all groups regardless of being in the treatment group or not indicate that their mental health interferes with their work sometimes.

3. **Which model is best in classifying whether or not the participant needs treatment?**



Based on this pie chart, the dataset was balanced as there were 51.7% of the number of individuals needing treatment in comparison with 48.3% that did not need treatment. Because of this reason, we would base on the overall accuracy to select the best model(s). Dataset was split into train/test set using 80/20 ratio

Method	Confusion matrix	Overall Accuracy

Logistic Classification 1	`true`
	`pred No Yes`
	`No 57 19`
	`Yes 21 53`
		73.3%

Logistic Classification 2	`true`
	`pred No Yes`
	`No 55 16`
	`Yes 23 56`
		74.0%

LDA	`true`
	`pred No Yes`
	`No 57 20`
	`Yes 21 52`
		72.7%

QDA	`true`
	`pred No Yes`
	`No 51 23`
	`Yes 27 49`
		66.7%

Decision Tree	`true`
	`pred No Yes`
	`No 41 11`
	`Yes 37 61`
		68.0%

Pruned Tree	`pred No Yes`
	`No 45 17`
	`Yes 33 55`
		66.7%

Method	Confusion matrix	Overall Accuracy

Bagging	`true`
	`pred No Yes`
	`No 51 16`
	`Yes 27 56`
		71.3%

Random Forest	true pred No Yes No 53 14 Yes 25 58	74.0%
Boosting N.Tree = 100 Shrinkage = 0.05, I.D = 3	true pred 0 1 0 53 14 1 25 58	74.0%
SVC Linear - best parameters: cost gamma 0.04641589 0.001	truth pred No Yes No 53 20 Yes 25 52	70.0%
SVM Radial - best parameters: cost gamma 7.742637 0.003593814	truth pred No Yes No 54 22 Yes 24 50	69.3%
SVM Poly - best parameters: cost degree 100 3	truth pred No Yes No 53 25 Yes 25 47	66.7%

12 classification/machine learning models were applied on the cleaned dataset. Logistic classification with only significant predictors, random forests, and boosting are the 3 models with the best overall accuracy score of 74%.

4. **What are the important/significant factors that determine whether or not an employee needs treatment?**

+ Logistic Classification

```
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                 -2.300121   1.009421  -2.279  0.02269 *  
## Age                          0.001472   0.014057   0.105  0.91661    
## Gender1                     -0.194866   0.241468  -0.807  0.41966    
## Gender2                     -1.199522   1.445687  -0.830  0.40669    
## self_employed1               0.019208   0.464283   0.041  0.96700    
## family_history1              0.856896   0.202312   4.236 2.28e-05 ***
## work_interfere1              2.531967   0.403016   6.283 3.33e-10 ***
## work_interfere2              1.223338   0.295232   4.144 3.42e-05 ***
## work_interfere3              2.104243   0.424017   4.963 6.95e-07 ***
## no_employees2               -0.413177   0.436256  -0.947  0.34359    
## no_employees3               -0.107164   0.467584  -0.229  0.81872    
## no_employees4                0.043669   0.490957   0.089  0.92912    
## no_employees5               -0.485419   0.625939  -0.776  0.43804    
## no_employees6               -0.457819   0.487862  -0.938  0.34803    
## remote_work1                -0.129795   0.234450  -0.554  0.57984    
## tech_company1                0.149127   0.264029   0.565  0.57220    
## benefits1                   -0.286940   0.365100  -0.786  0.43191    
## benefits2                   -0.646993   0.388890  -1.664  0.09617 .  
## care_options1                0.756571   0.268759   2.815  0.00488 ** 
## care_options2                0.003500   0.264017   0.013  0.98942    
## wellness_program1            0.174989   0.335732   0.521  0.60222    
## wellness_program2            0.397994   0.311501   1.278  0.20137    
## seek_help1                   0.212630   0.347620   0.612  0.54075    
## seek_help2                   0.464986   0.274612   1.693  0.09041 .  
## anonymity1                   0.336043   0.690145   0.487  0.62632    
## anonymity2                   0.155778   0.675650   0.231  0.81766    
## leave1                      -0.165537   0.321539  -0.515  0.60667    
## leave2                       0.231792   0.281106   0.825  0.40961    
## leave3                       0.860605   0.396552   2.170  0.02999 *  
## leave4                      -0.306077   0.426036  -0.718  0.47249    
## mental_health_consequence1  1.027544   0.374187   2.746  0.00603 ** 
## mental_health_consequence2  0.574655   0.285930   2.010  0.04445 *  
## phys_health_consequence1    -0.062298   0.562721  -0.111  0.91185    
## phys_health_consequence2     0.131633   0.278677   0.472  0.63668    
## coworkers1                   0.392483   0.307919   1.275  0.20244    
## supervisor1                 -0.118107   0.264746  -0.446  0.65551    
## mental_health_interview1    -0.068784   0.805096  -0.085  0.93191    
## mental_health_interview2    -0.191185   0.337241  -0.567  0.57078    
## phys_health_interview1       0.176080   0.365387   0.482  0.62988    
## phys_health_interview2      -0.483388   0.224303  -2.155  0.03116 *  
## mental_vs_physical1          0.374232   0.357981   1.045  0.29584    
## mental_vs_physical2          0.093043   0.269887   0.345  0.73029    
## obs_consequence1             0.139963   0.341734   0.410  0.68212    
## ---
```

family_history, work_interfere, care_options, leave, mental_health_consequence, and phys_health_interview are the 6 significant predictors at the 5% significant level.

We refitted the logistic regression model using only significant predictors.

```
Coefficients:
                           Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -2.17624    0.35729  -6.091 1.12e-09 ***
family_history1             0.85899    0.19178   4.479 7.50e-06 ***
work_interfere1             2.44518    0.38283   6.387 1.69e-10 ***
work_interfere2             1.14276    0.28302   4.038 5.40e-05 ***
work_interfere3             1.94505    0.39462   4.929 8.27e-07 ***
care_options1               1.01634    0.22709   4.476 7.62e-06 ***
care_options2               0.12968    0.24321   0.533  0.59389    
leave1                     -0.04309    0.29930  -0.144  0.88553    
leave2                      0.27258    0.26351   1.034  0.30094    
leave3                      0.89488    0.37664   2.376  0.01750 *  
leave4                     -0.19528    0.37874  -0.516  0.60613    
mental_health_consequence1  0.77768    0.26605   2.923  0.00347 ** 
mental_health_consequence2  0.41364    0.22409   1.846  0.06492 .  
phys_health_interview1      0.16140    0.31055   0.520  0.60325    
phys_health_interview2     -0.51218    0.20178  -2.538  0.01114 *  
```

family_history, work_interfere, care_options 1, leave 3, mental_health_consequence 1, and phys_health_interview 2 are the significant predictors at the 5% significant level in this model.

Log odds fitted model

log(p(x)/(1+p(x))) = -2.17624 + 0.85899 * I(family_history=1) + 2.44518 * I(work_interfere=1) + 1.14276* I(work_interfere=2) + 1.94505* I(work_interfere=3) + 1.01634 * I(care_options=1) + 0.12968 * I(care_options=2) - 0.04309 * I(leave=1) + 0.27258 * I(leave=2) + 0.89488 * I(leave=3) - 0.19528 * I(leave=4) + 0.77768 * I(mental_health_consequence=1) + 0.41364 * I(mental_health_consequence=2) + 0.16140 * I(phys_health_interview=1) - 0.51218 * I(phys_health_interview=2)

Given all other predictors are constant, the odds of needing treatment for participants with family history of mental illnesses are 236.0775% of those for participants without family history of mental illnesses.

Given all other predictors are constant, the odds of needing treatment for participants who claimed mental illnesses rarely interfere there works are 1153.263% of those for participants who claimed mental illnesses never interfere there works.

Given all other predictors are constant, the odds of needing treatment for participants who claimed mental illnesses sometimes interfere there works are 313.541% of those for participants who claimed mental illnesses never interfere there works.

Given all other predictors are constant, the odds of needing treatment for participants who claimed mental illnesses often interfere there works are 699.3982% of those for participants who claimed mental illnesses never interfere there works.

Given all other predictors are constant, the odds of needing treatment for participants who know about the health care options for mental illnesses offered by employers are 276.3063% of those for participants don’t know.

Given all other predictors are constant, the odds of needing treatment for participants who felt that it is somewhat difficult to take medical leave for mental health condition are 244.7042% of those for participants who don’t know about this situation.

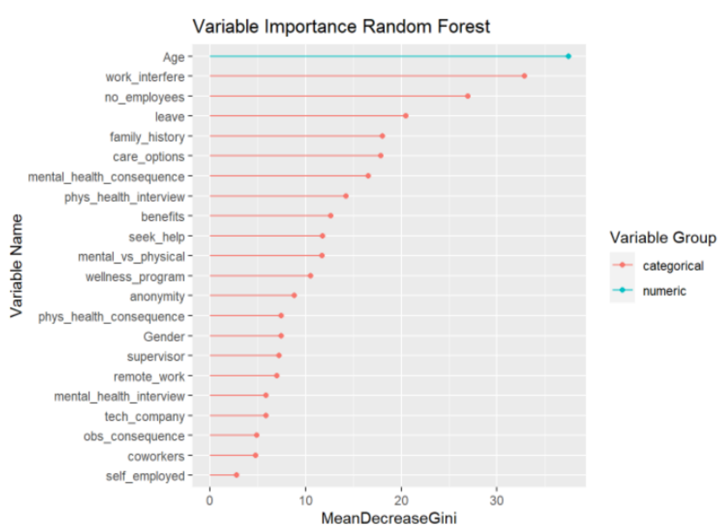
Given all other predictors are constant, the odds of needing treatment for participants who think that discussing the mental health issues with his/her employers will cause consequences are 217.6417% of those for participants don’t think so.

Given all other predictors are constant, the odds of needing treatment for participants may bring up a physical health issue with a potential employer in an interview are 59.91879% of those for participants will not do so.

+ Random Forests

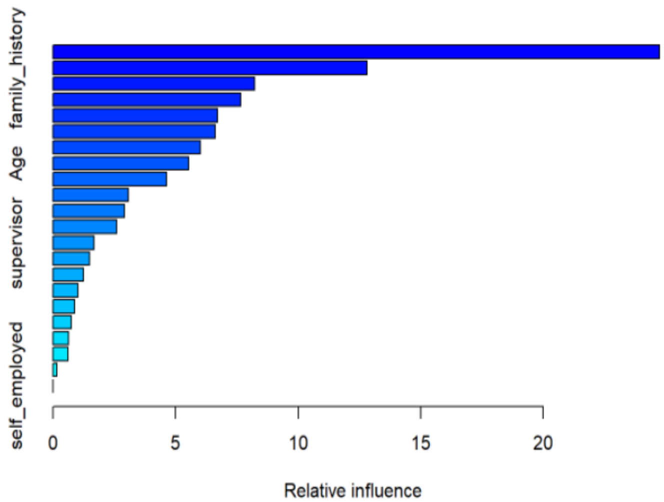
```
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 31.28%
## Confusion matrix:
##      No Yes class.error
## No  185 100   0.3508772
## Yes  88 228   0.2784810
```

The number of trees in this random forest model were 500 with number of variables tried at each split of 5. The out-of-bag estimate of error rate is 31.28%



Age, work_interfere, no_employees, leave, family_history, care_options are the top important variables

Variable Importance Plot




```
##                                                    var    rel.inf
## work_interfere                          work_interfere 24.7297491
## family_history                          family_history 12.8101152
## care_options                              care_options  8.2346478
## no_employees                              no_employees  7.6568696
## mental_health_consequence mental_health_consequence  6.7112185
## leave                                            leave  6.6187399
## phys_health_interview            phys_health_interview  6.0198338
## Age                                                Age  5.5284108
## benefits                                      benefits  4.6387484
## Gender                                          Gender  3.0736276
## anonymity                                    anonymity  2.9185020
## seek_help                                    seek_help  2.6084590
## supervisor                                  supervisor  1.6714032
## mental_vs_physical                  mental_vs_physical  1.4912156
## wellness_program                      wellness_program  1.2456252
## coworkers                                    coworkers  1.0178215
## mental_health_interview        mental_health_interview  0.8733035
## tech_company                              tech_company  0.7544107
## phys_health_consequence        phys_health_consequence  0.6288770
## obs_consequence                        obs_consequence  0.6164023
## remote_work                                remote_work  0.1520191
## self_employed                            self_employed  0.0000000
```

+ Boosting

```
##   n.trees_vec shrinkage_vec interaction.depth_vec miss_classification_rate
## 1         100          0.05                     3                 1.178525
```

n.trees = 100, shrinkage = 0.05 and interaction depth = 3 are the best parameters after tuning.

Work_interfere, family_history, care_options, and number of employees are the 4 most important variables in this boosting model.

Logistic Classification

with only significant predictors

```
     true
pred  No Yes
  No  55  16
  Yes 23  56
```

```
Overall Accuracy is 0.74
Sensitivity is 0.778
Specificity is 0.705
```

Random Forests

```
     true
pred  No Yes
  No  53  14
  Yes 25  58
```

```
Overall Accuracy is 0.74
Sensitivity is 0.806
Specificity is 0.679
```

Boosting

```
    true
pred  0  1
   0 53 14
   1 25 58
```

```
Overall Accuracy is 0.74
Sensitivity is 0.806
Specificity is 0.679
```

Overall, all 3 models share the same overall accuracy of 74%. Random Forests and Boosting show a slightly higher number of people who were predicted to need mental health assistance indeed needed treatment.

Except for Age, which was shown to be the most important factors in the Random Forest model, work_interfere, no_employees, leave, family_history, and care_options are the top 5 important/significant variables for the classification of treatment.

The detailed explanation for all of the important variables are:

- ➢ Family_history: Family with history of mental illness? Yes/No
- ➢ Work_interfere: If the participant feels that mental illness interferes with work? Never/Rarely/Sometimes/Often
- ➢ No_employees: Number of employees at the participant's workplace? 1-5/6-25/26-100/100-500/500-1000/More than 1000
- ➢ Benefits: Does employer provide mental health benefits? Yes/No/Don't Know
- ➢ Care_options: Does participant know the options for mental health care your employer provides? Yes/No/Not Sure
- ➢ Mental_health_consequence: Does participant think that discussing a mental health issue with his/her employer would have negative consequences? Yes/No/Maybe
- ➢ Phys_health_interview: Would participant bring up a physical health issue with a potential employer in an interview? Yes/No/Maybe

5. Given a person's profile, would our best fitting model predict the person seeking help?

Logistic Classification 2

```
data_logit = data.frame(family_history = "1", work_interfere = "3", care_options = "1", leave = "3", mental_health_consequen
ce = "0", phys_health_interview = "1")
```

```
##          1
## 0.9370514
```

Boosting

```
# using boosting model for prediction
data_boosting = data.frame(Age = 34, Gender = "1", self_employed = "0", family_history = "1", work_interfere = "3", no_emplo
yees = "5", remote_work = "0", tech_company = "1", benefits = "1", care_options = "1", wellness_program = "1", seek_help =
"0", anonymity = "1", leave = "3", mental_health_consequence = "0", phys_health_consequence = "0", coworkers = "0", supervis
or = "0", mental_health_interview = "1", phys_health_interview = "1", mental_vs_physical = "0", obs_consequence = "1")
```

```
## [1] 0.8287095
```

Given the profile of an individual, with a more detailed features in Boosting, the predicted probability of a male individual being determined to need a treatment for his mental issues are 0.937 for Logistic Classification with only significant predictors or 0.829 for Boosting.

Conclusion:

In conclusion, we utilized twelve models to decide which model was best. Out of the twelve model tested, Logistic Classification with significant predictors only, Boosting, and Random Forest were the top three model with an accuracy of 74%. When testing these three models to make a prediction using an employee's profile, Logistic Classification and Boosting Models, had a very high accuracy of 94% and 83%. We were also able to determine that employees over the age of 40 are seeking mental health services more than those employees under 40. When testing age, we were unable to conclude if one gender needed treatment more than the other. Number of employees, Family history, Work interference, Care options, Leave , Mental health consequences, and Physical health are significant predictors in determining whether or not a subject needs treatment for the logistic classification, random forest, and boosting models. Age is considered important only in the random forest model. Our sentiment analysis used to analyze the comments in the dataset concluded stress and depression are the top negative opinions of employees in reference to mental health. Meanwhile, the top positive words, support and benefit, mean that employees need better assistances and benefits from employers with regards to the mental health issues. In the future, we would like to search for a dataset with balanced character and numerical variables to get better results. We would also like to research further into a dataset with balanced gender to have more conclusive results regarding gender.

Appendix:

EDA

```r
# Read Original files, which are provided train, test, and sample(with response y for test) data
og_train <- read_csv("train.csv")
```

```r
og_test <- read_csv("test.csv")
```

```r
test_y <- read_csv("sample.csv")
```

```r
# Join original test set and sample(sample contains treatment column for test set)
og_test_complete <- og_test %>%
  right_join(test_y, by = "s.no") %>%
  dplyr::select(-s.no)
```

```r
# Join original train and new test set
MH <- og_train %>%
  dplyr::select(-s.no) %>%
  full_join(og_test_complete)
```

```r
# Check the MH dataset
skim(MH)
```

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Gender	0	1.00	1	46	0	44	0
Country	0	1.00	5	22	0	48	0
state	515	0.59	2	2	0	45	0
self_employed	18	0.99	2	3	0	2	0
family_history	0	1.00	2	3	0	2	0
treatment	0	1.00	2	3	0	2	0
work_interfere	264	0.79	5	9	0	4	0
no_employees	0	1.00	3	14	0	6	0
remote_work	0	1.00	2	3	0	2	0
tech_company	0	1.00	2	3	0	2	0
benefits	0	1.00	2	10	0	3	0
care_options	0	1.00	2	8	0	3	0
wellness_program	0	1.00	2	10	0	3	0
seek_help	0	1.00	2	10	0	3	0
anonymity	0	1.00	2	10	0	3	0
leave	0	1.00	9	18	0	5	0
mental_health_consequence	0	1.00	2	5	0	3	0
phys_health_consequence	0	1.00	2	5	0	3	0
coworkers	0	1.00	2	12	0	3	0
supervisor	0	1.00	2	12	0	3	0
mental_health_interview	0	1.00	2	5	0	3	0
phys_health_interview	0	1.00	2	5	0	3	0
mental_vs_physical	0	1.00	2	10	0	3	0
obs_consequence	0	1.00	2	3	0	2	0
comments	1096	0.13	1	3548	0	159	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Age	0	1	30.79	50.83	-1726	27	31	36	329	▁▁▁▇

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Timestamp	0	1	2014-08-27 11:29:31	2016-02-01 23:04:31	2014-08-28 02:30:00	1246

Checking the number of participants from each country

```
ggplot(bardf, aes(x = reorder(Country, -n), y = n)) +
    geom_bar(fill = "red",stat = "identity") + geom_text(aes(label = n), vjust = 0.2) +
    labs(title = "Participants by nations",y = "Number of subjects", x = "Nationality") + theme(axis.text.x = element_text(a
ngle = 90, vjust = 0.5))
```

```
# Choose only US participants
MH_US <- MH %>%
  filter(Country=="United States") %>%
  dplyr::select(-c(Timestamp,comments, Country, state)) # save comments for text mining
```

Data Wrangling

Gender

```
table(MH_US$Gender)
```

```
# Change values in Gender to 0 for female, 1 for male, 2 for others
female_value = c("cis-female/femme", "Cis Female", "f", "F", "femail", "Femake", "female", "Female", "Female (cis)", "Female (trans)", "Trans-female", "Trans woman", "woman", "Woman")
male_value = c("m", "M", "Mail", "maile", "Make", "male", "Male", "Male-ish", "Man", "msle", "cis male", "Cis Male")


MH_US <- MH_US %>%
  mutate(Gender = as.factor(ifelse(Gender %in% female_value, 0,
                            ifelse(Gender %in% male_value,1,2))), treatment = as.factor(treatment))
```

Impute NA values in self_employed. Encode self_employed, family_history, remote_work, tech_company, obs_consequece, coworkers, and supervisors

```
table(MH_US$self_employed)/nrow(MH_US)
```

```
# Update NA values of self_employed to No
MH_US$self_employed <- MH_US$self_employed %>% replace_na('No')

# Create vector of columns with Yes or No only
Yes_No <- c("self_employed", "family_history", "remote_work", "tech_company", "obs_consequence", "coworkers", "supervisor")

# Function to change Yes/No to 1/0
helperFunction <- function(x){
    as.factor(ifelse(x == "Yes", 1,0))
}

# Apply function to the Yes_No columns
MH_US[,Yes_No] = lapply(MH_US[,Yes_No], helperFunction)
```

Impute NA Values in work_interfere. Encode work interfere and leave

```
table(MH_US$work_interfere)/nrow(MH_US)
```

```r
MH_US$work_interfere <- MH_US$work_interfere %>% replace_na('Sometimes') # Sometimes has the highest frequency

# Encoding work_interfere and leave (ordinal)
MH_US <- MH_US %>%
  mutate(work_interfere = as.factor(ifelse(work_interfere=="Often",3,ifelse(work_interfere=="Sometimes",2,ifelse(work_interf
ere=="Rarely",1,0)))), leave = as.factor(ifelse(leave=="Very difficult",4,ifelse(leave=="Somewhat difficult",3,ifelse(leave=
="Somewhat easy",2,ifelse(leave=="Very easy",1,0))))))
```

Encode “benefits”, “care_options”, “wellness_program”, “seek_help”, “anonymity”,“mental_health_consequence”, “phys_health_consequence”, “mental_health_interview”, “phys_health_interview”, and “mental_vs_physical”

```r
# Create vector of columns with Yes or No or 3rd value
Yes_No_3rd <- c("benefits", "care_options", "wellness_program", "seek_help", "anonymity","mental_health_consequence", "phys_
health_consequence", "mental_health_interview", "phys_health_interview","mental_vs_physical")

# Function to change Yes/No/3rd-value to 1/0/2
helperFunction <- function(x){
    as.factor(ifelse(x == "Yes", 1, ifelse(x == "No",0, 2)))
}

# Apply function to the Yes_No columns
MH_US[,Yes_No_3rd] = lapply(MH_US[,Yes_No_3rd], helperFunction)
```

Encode the number of employees

```r
# Encoding no_employees from 6 to 1 for in accordance with the decreasing values
MH_US <- MH_US %>%
  mutate(no_employees = as.factor(ifelse(no_employees=="More than 1000",6,ifelse(no_employees=="500-1000",5,ifelse(no_employ
ees=="100-500",4,ifelse(no_employees=="26-100",3,ifelse(no_employees=="6-25",2,1)))))))

# Check the values of no_employees after encoding
table(MH_US$no_employees)
```

Outliers in Age

```r
# Check for outliers in age
ggplot(data = MH_US, mapping = aes(x = treatment, y = Age)) +
geom_boxplot() + labs(title = "Age outliers")
```

```r
MH_US %>%
  filter(Age < 18 | Age >72)
```

```r
# Impute the Age outliers with mean of Age column
MH_US <- MH_US  %>%
  mutate(Age = ifelse(Age <18 | Age >72,round(mean(MH_US$Age)),Age))
```

Check the cleaned dataset

```
skim(MH_US)
```

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Gender	0	1	FALSE	3	1: 562, 0: 185, 2: 4
self_employed	0	1	FALSE	2	0: 695, 1: 56
family_history	0	1	FALSE	2	0: 421, 1: 330
treatment	0	1	FALSE	2	Yes: 388, No: 363
work_interfere	0	1	FALSE	4	2: 433, 0: 125, 1: 111, 3: 82
no_employees	0	1	FALSE	6	6: 216, 3: 170, 2: 134, 4: 113
remote_work	0	1	FALSE	2	0: 513, 1: 238
tech_company	0	1	FALSE	2	1: 611, 0: 140
benefits	0	1	FALSE	3	1: 398, 2: 236, 0: 117
care_options	0	1	FALSE	3	1: 311, 0: 239, 2: 201
wellness_program	0	1	FALSE	3	0: 455, 1: 167, 2: 129
seek_help	0	1	FALSE	3	0: 300, 2: 262, 1: 189
anonymity	0	1	FALSE	3	2: 495, 1: 237, 0: 19
leave	0	1	FALSE	5	0: 385, 2: 137, 1: 108, 3: 68
mental_health_consequence	0	1	FALSE	3	2: 300, 0: 280, 1: 171
phys_health_consequence	0	1	FALSE	3	0: 571, 2: 150, 1: 30
coworkers	0	1	FALSE	2	0: 627, 1: 124
supervisor	0	1	FALSE	2	0: 447, 1: 304
mental_health_interview	0	1	FALSE	3	0: 635, 2: 100, 1: 16
phys_health_interview	0	1	FALSE	3	0: 339, 2: 320, 1: 92
mental_vs_physical	0	1	FALSE	3	2: 363, 1: 201, 0: 187
obs_consequence	0	1	FALSE	2	0: 662, 1: 89

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Age	0	1	33.13	7.62	18	28	32	37	72	

Inference

```r
# Extract comment column from MH dataset
MH_com <- MH %>%
  filter(Country == "United States") %>%
  dplyr::select(comments,treatment)
```

```
table(MH_com$treatment)/nrow(MH_com)
```

```
##
##        No       Yes
## 0.4038462 0.5961538
```

```
# remove NA observations
MH_com <- na.omit(MH_com)
```

```
ggplot(MH_com) +
  geom_bar(aes(x=treatment),
           position = "dodge") + labs(title = "Yes/No Commenters")
```

```
# Tokenize and remove stop words in comments
data("stop_words")
text_tidy = MH_com %>%
  unnest_tokens(word, comments) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
```

```
# Rank frequency of words in comments
text_tidy %>%
  slice_max(order_by = n, n = 10) %>%
  ggplot(aes(n, reorder(word, n))) +
  geom_col(show.legend = FALSE) +
  labs(y = NULL) + labs(title = "Frequency of Words")
```

```
# Create wordcloud
set.seed(123)
text_tidy%>%
with(wordcloud(word, n, random.order = FALSE,
colors = brewer.pal(8, "Dark2")))
```

```r
# Stem words in text_tidy
library(SnowballC)
text_tidy <- mutate(text_tidy,
word.stem = wordStem(word, language = "en"))
```

```r
word.freq <- text_tidy %>%
  inner_join(get_sentiments("bing"),by = c("word.stem"="word")) %>%
  count(word.stem, sentiment, sort = TRUE) %>%
  rename(counts = n)
```

```r
# Remove the f-bomb word
word.freq <- word.freq %>%
  filter(word.stem!="fuck")
```

```r
word.freq %>%
group_by(sentiment) %>%
slice_max(order_by = counts, n = 5) %>%
mutate(word.stem = reorder(word.stem, counts)) %>%
ggplot(aes(counts, word.stem, fill = sentiment)) +
geom_col(show.legend = FALSE) +
facet_wrap(~sentiment, scales = "free_y") +
labs(x = "Contribution to sentiment",
y = NULL) + labs(title = "Frequency of Positive and Negatives Words" )
```

First Question

```r
# Create new dataset with AgeLevel column
MH_Age <- MH_US %>%
  mutate(AgeLevel = as.factor(ifelse(Age>=40,"older","younger")))
```

```r
MH_Age %>%
  dplyr::select(AgeLevel) %>%
  count(AgeLevel) %>%
  mutate(prop = round(n*100/sum(n), 1),
         lab.ypos = cumsum(prop) - 0.5*prop) %>%
  ggplot(aes(x = "", y = prop, fill = AgeLevel)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  ggpubr::fill_palette("jco")+
  geom_text(aes(label = paste0(prop, "%")),
            position = position_stack(vjust = 0.5))+
  coord_polar("y", start = 0)+ labs(title = "Distribution of Age") +

  theme_void()
```

```
ggplot(MH_Age) +
  geom_bar(aes(x=treatment, fill=work_interfere),
           position = "dodge") +
  facet_wrap(~AgeLevel) + labs(title = "treatment and work interfere based on Age")
```

Second Question

```
MH_Age %>%
  dplyr::select(Gender) %>%
  count(Gender) %>%
  mutate(prop = round(n*100/sum(n), 1),
         lab.ypos = cumsum(prop) - 0.5*prop) %>%
  ggplot(aes(x = "", y = prop, fill = Gender)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  ggpubr::fill_palette("jco")+
  geom_text(aes(label = paste0(prop, "%")),
            position = position_stack(vjust = 0.5))+
  coord_polar("y", start = 0)+ labs(title = "Distribution of Gender") +

  theme_void()
```

```
ggplot(MH_Age) +
  geom_bar(aes(x=treatment, fill=work_interfere),
           position = "dodge") +
  facet_wrap(~Gender) + labs(title = "Treatment and work interference based on Gender")
```

Third Question

Split the dataset

```
table(MH_US$treatment)/nrow(MH_US)
```

```
MH_US %>%
  dplyr::select(treatment) %>%
  count(treatment) %>%
  mutate(prop = round(n*100/sum(n), 1),
         lab.ypos = cumsum(prop) - 0.5*prop) %>%
  ggplot(aes(x = "", y = prop, fill = treatment)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  ggpubr::fill_palette("jco")+
  geom_text(aes(label = paste0(prop, "%")),
            position = position_stack(vjust = 0.5))+
  coord_polar("y", start = 0)+ labs(title = "Balance of the dataset") +

  theme_void()
```

```r
# Splitting MH_US into train and test sets
set.seed(123)
n = nrow(MH_US)
prop = .8
train_id = sample(1:n, size = round(n*prop), replace = FALSE)
test_id = (1:n)[-which(1:n %in% train_id)]
train_set = MH_US[train_id, ]
test_set = MH_US[test_id, ]
```

Logistic Classification

```r
# Fit logistic regression model
logit_fit <- glm(treatment ~., data = train_set, family = "binomial")
summary(logit_fit)
```

```r
# Confusion matrix and overall accuracy for this model
logit1.probs = predict(logit_fit, test_set, type = "response")
logit1.pred = ifelse(logit1.probs >= 0.5, "Yes", "No")
#considering probabilities >0.5 as Up
tb1 = table(pred = logit1.pred, true = test_set$treatment)
tb1
```

```r
glue("Overall Accuracy is {round((tb1[1,1] + tb1[2,2])/sum(tb1),3)} \nSensitivity is {round(tb1[2,2]/sum(tb1[,2]),3)}\nSpeci
ficity is {round(tb1[1,1]/sum(tb1[,1]),3)}")
```

```r
logit_fit_2 <- glm(treatment ~ family_history + work_interfere + care_options + leave + mental_health_consequence + phys_hea
lth_interview, data = train_set, family = "binomial")
summary(logit_fit_2)
```

```r
logit2.probs = predict(logit_fit_2, test_set, type = "response")
logit2.pred = ifelse(logit2.probs >= 0.5, "Yes", "No")
#considering probabilities >0.5 as Up
tb2 = table(pred = logit2.pred, true = test_set$treatment)
tb2
```

```r
glue("Overall Accuracy is {round((tb2[1,1] + tb2[2,2])/sum(tb2),3)} \nSensitivity is {round(tb2[2,2]/sum(tb2[,2]),3)}\nSpeci
ficity is {round(tb2[1,1]/sum(tb2[,1]),3)}")
```

LDA/QDA

```r
lda_mod <- lda(treatment~., data = train_set)
lda_mod
```

```
lda_prob <- predict(lda_mod, test_set)
lda_class <- lda_prob$class
tb3 = table(pred = lda_class, true = test_set$treatment)
tb3
```

```
glue("Overall Accuracy is {round((tb3[1,1] + tb3[2,2])/sum(tb3),3)} \nSensitivity is {round(tb3[2,2]/sum(tb3[,2]),3)}\nSpeci
ficity is {round(tb3[1,1]/sum(tb3[,1]),3)}")
```

```
qda_mod <- qda(treatment ~., data = train_set)
qda_mod
```

```
qda_prob <- predict(qda_mod, test_set)
qda_class <- qda_prob$class
tb4 = table(pred = qda_class, true = test_set$treatment)
tb4
```

```
glue("Overall Accuracy is {round((tb4[1,1] + tb4[2,2])/sum(tb4),3)} \nSensitivity is {round(tb4[2,2]/sum(tb4[,2]),3)}\nSpeci
ficity is {round(tb4[1,1]/sum(tb4[,1]),3)}")
```

Tree-based model

Decision Tree

```
mod.tree = tree(treatment~. , data = train_set)
summary(mod.tree)
```

```
tree.pred <- predict(mod.tree, test_set, type = "class")
tb5 = table(pred = tree.pred , true = test_set$treatment)
tb5
```

```
glue("Overall Accuracy is {round((tb5[1,1] + tb5[2,2])/sum(tb5),3)} \nSensitivity is {round(tb5[2,2]/sum(tb5[,2]),3)}\nSpeci
ficity is {round(tb5[1,1]/sum(tb5[,1]),3)}")
```

Pruned Decision Tree

```
set.seed(123)
cv.out = cv.tree(mod.tree, K=10, FUN = prune.misclass)
cv.out
```

```
prune.mod = prune.misclass(mod.tree, best = cv.out$size[which.min(cv.out$dev)])
prune.mod
```

```
prune_tree.pred <- predict(prune.mod, test_set, type = "class")
tb6 = table(pred = prune_tree.pred , test_set$treatment)
tb6
```

```
glue("Overall Accuracy is {round((tb6[1,1] + tb6[2,2])/sum(tb6),3)} \nSensitivity is {round(tb6[2,2]/sum(tb6[,2]),3)}\nSpeci
ficity is {round(tb6[1,1]/sum(tb6[,1]),3)}")
```

Bagging

```
set.seed(123)
p = ncol(train_set) - 1
bag_fit = randomForest(treatment~ ., data = train_set, mtry = p, importance = TRUE)
bag_fit
```

```
imp <- varImpPlot(bag_fit)
```

```
imp <- as.data.frame(imp)
imp$varnames <- rownames(imp) # row names to column
rownames(imp) <- NULL
imp$var_categ <- ifelse(imp$varnames == "Age", "numeric", "categorical")
```

```
ggplot(imp, aes(x=reorder(varnames, MeanDecreaseGini), y=MeanDecreaseGini, color=as.factor(var_categ))) +
  geom_point() +
  geom_segment(aes(x=varnames,xend=varnames,y=0,yend=MeanDecreaseGini)) +
  scale_color_discrete(name="Variable Group") +
  ylab("MeanDecreaseGini") +
  xlab("Variable Name") + labs(title = "Variable Importance Bagging") + coord_flip()
```

```
bag_pred = predict(bag_fit, test_set, type = "class")
tb7 = table(pred = bag_pred, true=test_set$treatment)
tb7
```

```
glue("Overall Accuracy is {round((tb7[1,1] + tb7[2,2])/sum(tb7),3)} \nSensitivity is {round(tb7[2,2]/sum(tb7[,2]),3)}\nSpeci
ficity is {round(tb7[1,1]/sum(tb7[,1]),3)}")
```

Random Forests

```
set.seed(123)
rf_fit = randomForest(treatment ~ ., data = train_set, mtry = round(sqrt(p)), importance = TRUE)
rf_fit
```

```
imp <- varImpPlot(rf_fit)
```

```
imp <- as.data.frame(imp)
imp$varnames <- rownames(imp) # row names to column
rownames(imp) <- NULL
imp$var_categ <- ifelse(imp$varnames == "Age", "numeric", "categorical")
```

```
ggplot(imp, aes(x=reorder(varnames, MeanDecreaseGini), y=MeanDecreaseGini, color=as.factor(var_categ))) +
  geom_point() +
  geom_segment(aes(x=varnames,xend=varnames,y=0,yend=MeanDecreaseGini)) +
  scale_color_discrete(name="Variable Group") +
  ylab("MeanDecreaseGini") +
  xlab("Variable Name") + labs(title = "Variable Importance Random Forest") + coord_flip()
```

```
rf_pred = predict(rf_fit, test_set, type = "class")
tb8 = table(pred = rf_pred, true=test_set$treatment)
tb8
```

```
glue("Overall Accuracy is {round((tb8[1,1] + tb8[2,2])/sum(tb8),3)} \nSensitivity is {round(tb8[2,2]/sum(tb8[,2]),3)}\nSpeci
ficity is {round(tb8[1,1]/sum(tb8[,1]),3)}")
```

Boosting

```
grid = expand.grid(
n.trees_vec = c(100, 200),
shrinkage_vec = c(0.2, 0.1, 0.06, 0.05, 0.04, 0.02, 0.01),
interaction.depth_vec = c(1, 2, 3),
miss_classification_rate = NA,
time = NA
)
head(grid, 10)
```

```r
# Train set/test set for boosting
train_set_boost = train_set %>%
  mutate(treatment_numeric = ifelse(treatment == "Yes", 1, 0)) %>%
  dplyr::select(-treatment)

test_set_boost = test_set %>%
  mutate(treatment_numeric = ifelse(treatment == "Yes", 1, 0)) %>%
  dplyr::select(-treatment)

# Tuning parameters for boosting using 5-fold cross validation
library(gbm)
set.seed(123)
for(i in 1:nrow(grid)){
time = system.time({
boost_fit = gbm(treatment_numeric~ ., train_set_boost,
n.trees = grid$n.trees_vec[i],
shrinkage = grid$shrinkage_vec[i],
interaction.depth = grid$interaction.depth_vec[i],
distribution = "bernoulli", cv.folds = 5)
}
)
grid$miss_classification_rate[i] =
boost_fit$cv.error[which.min(boost_fit$cv.error)]
grid
}
```

```r
grid %>% arrange(miss_classification_rate)
```

```r
boost_fit_best = gbm(treatment_numeric~ ., train_set_boost, n.trees = 100,
shrinkage = 0.05, interaction.depth = 3,
distribution = "bernoulli")
```

```r
summary(boost_fit_best)
```

```r
phat.test_boost_best = predict(boost_fit_best, test_set_boost,
type = "response")
```

```
## Using 100 trees...
```

```r
yhat.test_boost_best = ifelse(phat.test_boost_best > 0.5, 1, 0)
tb9 = table(pred = yhat.test_boost_best,
true = test_set_boost$treatment_numeric)
tb9
```

```
glue("Overall Accuracy is {round((tb9[1,1] + tb9[2,2])/sum(tb9),3)} \nSensitivity is {round(tb9[2,2]/sum(tb9[,2]),3)}\nSpeci
ficity is {round(tb9[1,1]/sum(tb9[,1]),3)}")
```

SVM

SVC Linear

```
set.seed(123)
tune_svm_linear = tune(svm, treatment ~., data = train_set, kernel = "linear", ranges = list(cost = 10^seq(-3,2, length.out=
10), gamma =10^seq(-3,2, length.out=10) ))
summary(tune_svm_linear)
```

```
svm_fit_linear = svm(treatment ~., data = train_set, kernel = "linear", gamma = 0.001, cost=0.04641589, scale = FALSE)
```

```
yhat_test_linear = predict(svm_fit_linear, test_set)

tb_svm_linear = table(pred = yhat_test_linear, truth = test_set$treatment)
tb_svm_linear
```

```
glue("Overall Accuracy is {round((tb_svm_linear[1,1] + tb_svm_linear[2,2])/sum(tb_svm_linear),3)} \nSensitivity is {round(tb
_svm_linear[2,2]/sum(tb_svm_linear[,2]),3)}\nSpecificity is {round(tb_svm_linear[1,1]/sum(tb_svm_linear[,1]),3)}")
```

SVM Radial

```
set.seed(123)
tune_svm_radial = tune(svm, treatment ~., data = train_set, kernel = "radial", ranges = list(cost = 10^seq(-3,2, length.out=
10), gamma =10^seq(-3,2, length.out=10) ))
summary(tune_svm_radial)
```

```
svm_fit_radial = svm(treatment ~., data = train_set, kernel = "radial", gamma=0.003593814, cost = 7.742637, scale = FALSE)
```

```
yhat_test_radial = predict(svm_fit_radial, test_set)
tb_svm_radial = table(pred = yhat_test_radial, truth = test_set$treatment)
tb_svm_radial
```

```
glue("Overall Accuracy is {round((tb_svm_radial[1,1] + tb_svm_radial[2,2])/sum(tb_svm_radial),3)} \nSensitivity is {round(tb
_svm_radial[2,2]/sum(tb_svm_radial[,2]),3)}\nSpecificity is {round(tb_svm_radial[1,1]/sum(tb_svm_radial[,1]),3)}")
```

SVM Polynomial

```
set.seed(123)
tune_svm_poly = tune(svm, treatment ~., data = train_set, kernel = "polynomial", ranges = list(cost = 10^seq(-3,2, length.ou
t=10), degree = c(2,3) ))
summary(tune_svm_poly)
```

```
svm_fit_poly = svm(treatment ~., data = train_set, kernel = "polynomial", cost = 100, degree = 3, scale = FALSE)
```

```
yhat_test_poly = predict(svm_fit_poly, test_set)
tb_svm_poly = table(pred = yhat_test_poly, truth = test_set$treatment)
tb_svm_poly
```

```
glue("Overall Accuracy is {round((tb_svm_poly[1,1] + tb_svm_poly[2,2])/sum(tb_svm_poly),3)} \nSensitivity is {round(tb_svm_p
oly[2,2]/sum(tb_svm_poly[,2]),3)}\nSpecificity is {round(tb_svm_poly[1,1]/sum(tb_svm_poly[,1]),3)}")
```

Question 5: Prediction

```
#using logistic model with significant predictors for prediction
data_logit = data.frame(family_history = "1", work_interfere = "3", care_options = "1", leave = "3", mental_health_consequen
ce = "0", phys_health_interview = "1")
print(predict(logit_fit_2,data_logit, type = "response"))
```

```
# using boosting model for prediction
data_boosting = data.frame(Age = 34, Gender = "1", self_employed = "0", family_history = "1", work_interfere = "3", no_emplo
yees = "5", remote_work = "0", tech_company = "1", benefits = "1", care_options = "1", wellness_program = "1", seek_help =
"0", anonymity = "1", leave = "3", mental_health_consequence = "0", phys_health_consequence = "0", coworkers = "0", supervis
or = "0", mental_health_interview = "1", phys_health_interview = "1", mental_vs_physical = "0", obs_consequence = "1")
factor_col = c(2:22)
data_boosting[,factor_col] = lapply(data_boosting[,factor_col], factor)
print(predict(boost_fit_best,data_boosting, type = "response"))
```