Professional Models – Model Report

Jack Weyer

2022-12-2

We begin by loading in the required packages and the initial project CSV.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## Warning: package 'tibble' was built under R version 4.2.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(usmap)
```

```
## Warning: package 'usmap' was built under R version 4.2.2
```

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
library(tidymodels)
```

```
## -- Attaching packages ---------------------------------- tidymodels 0.2.0.9000 --

## v broom        0.8.0     v rsample      0.1.1
## v dials        1.0.0     v tune         0.2.0
## v infer        1.0.2     v workflows    0.2.6
## v modeldata    0.1.1     v workflowsets 0.2.1
## v parsnip      1.0.0     v yardstick    1.0.0
## v recipes      0.2.0

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
df <- read_csv("Indicators_of_Anxiety_or_Depression_Based_on_Reported_Frequency_of_Symptoms_During_Last_
```

```
## Rows: 11484 Columns: 14

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): Indicator, Group, State, Subgroup, Phase, Time Period Label, Time ...
## dbl  (4): Time Period, Value, Low CI, High CI
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
#df$Indicator <- df$Indicator %>% as.factor()
```

We have 9 different “Group” variables. We will split these up when modeling because we do not have information beyond the subgroup within groups.

```
df$Group %>% unique()
```

```
## [1] "National Estimate"          "By Age"
## [3] "By Sex"                     "By Race/Hispanic ethnicity"
## [5] "By Education"               "By State"
## [7] "By Disability status"       "By Gender identity"
## [9] "By Sexual orientation"
```

By filtering to state values, we can group variables by state and visualize the mean anxiety or depressive disorder symptoms. We see a possible trend that the symptoms are more present in the South.

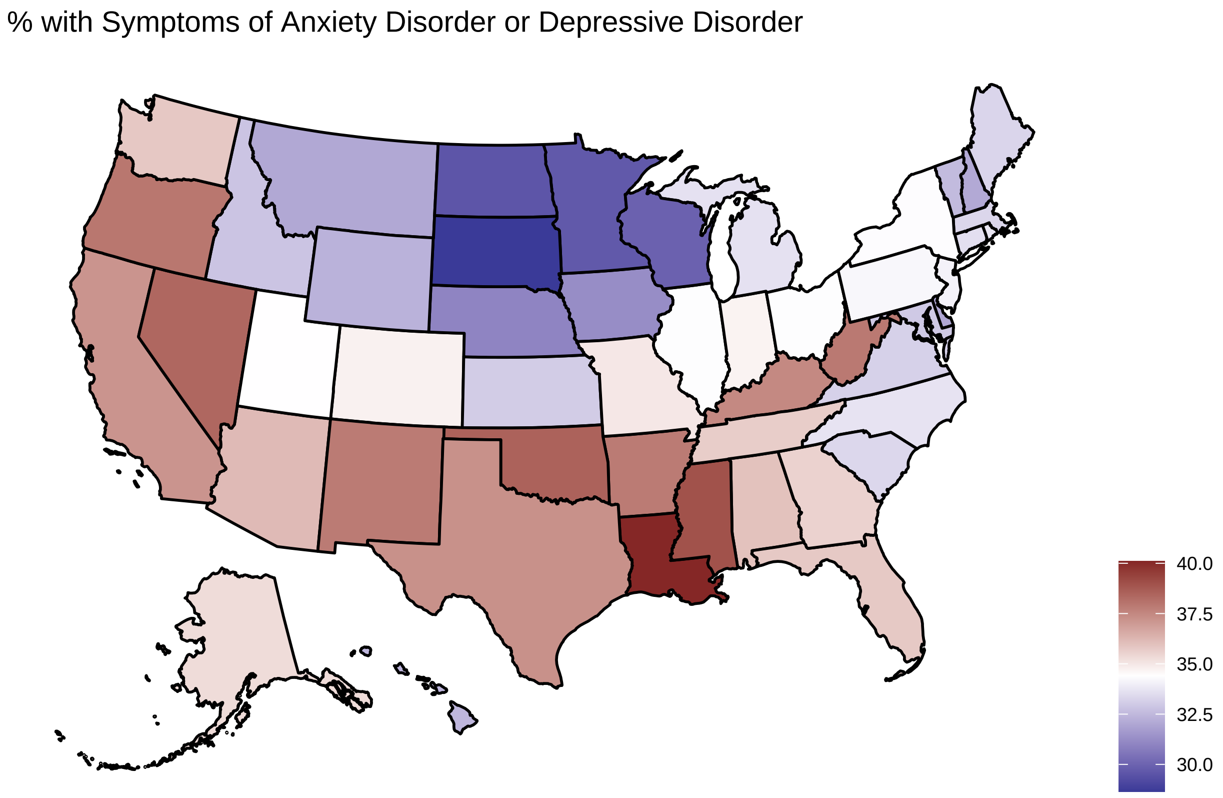
```
state_data <- df %>%
  filter(Group == "By State", Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  group_by(State) %>%
  summarise(value = mean(Value))

state_data <- state_data %>% rename(state = State)

state_data$state <- state_data$state %>% as.factor()

plot_usmap(data = state_data, values = "value", color = "black") +
  scale_fill_gradient2(
    midpoint = mean(state_data$value), mid = "white", low = muted("blue"), high =  muted("red"), name =
  )  + theme(legend.position = "right") + labs(title = "% with Symptoms of Anxiety Disorder or Depressi
```

```
## Warning: Ignoring unknown parameters: linewidth
```



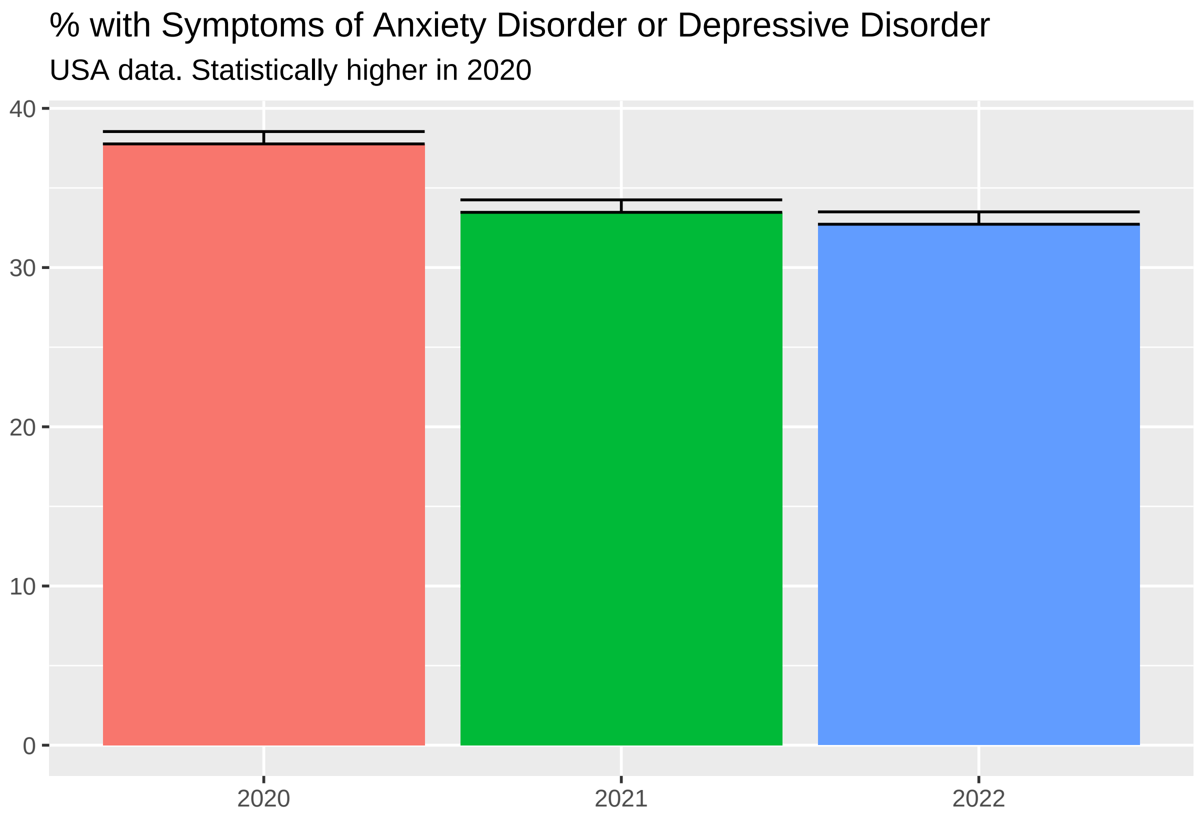
Cleaning up the data a bit, we change the necessary variables to factors. We also add a “middleDay” variable which is the middle of the weekly survey window. From that, we pull the month and year.

```
df$State <- df$State %>% as.factor()
df$Subgroup <- df$Subgroup %>% as.factor()
df$start <- as.Date(df$`Time Period Start Date`, format = '%m/%d/%Y')
df$end <- as.Date(df$`Time Period End Date`, format = '%m/%d/%Y')
df$middleDay <- df$start + floor((df$end - df$start)/2)
```

```
df <- df %>%
  select(-Phase, -`Time Period`, -`Time Period Label`, -`Time Period Start Date`, -`Time Period End Dat
            -start, -end) %>%
  mutate(Month = month(df$middleDay),
         Year = year(df$middleDay)) %>%
  filter(!is.na(Value))
```

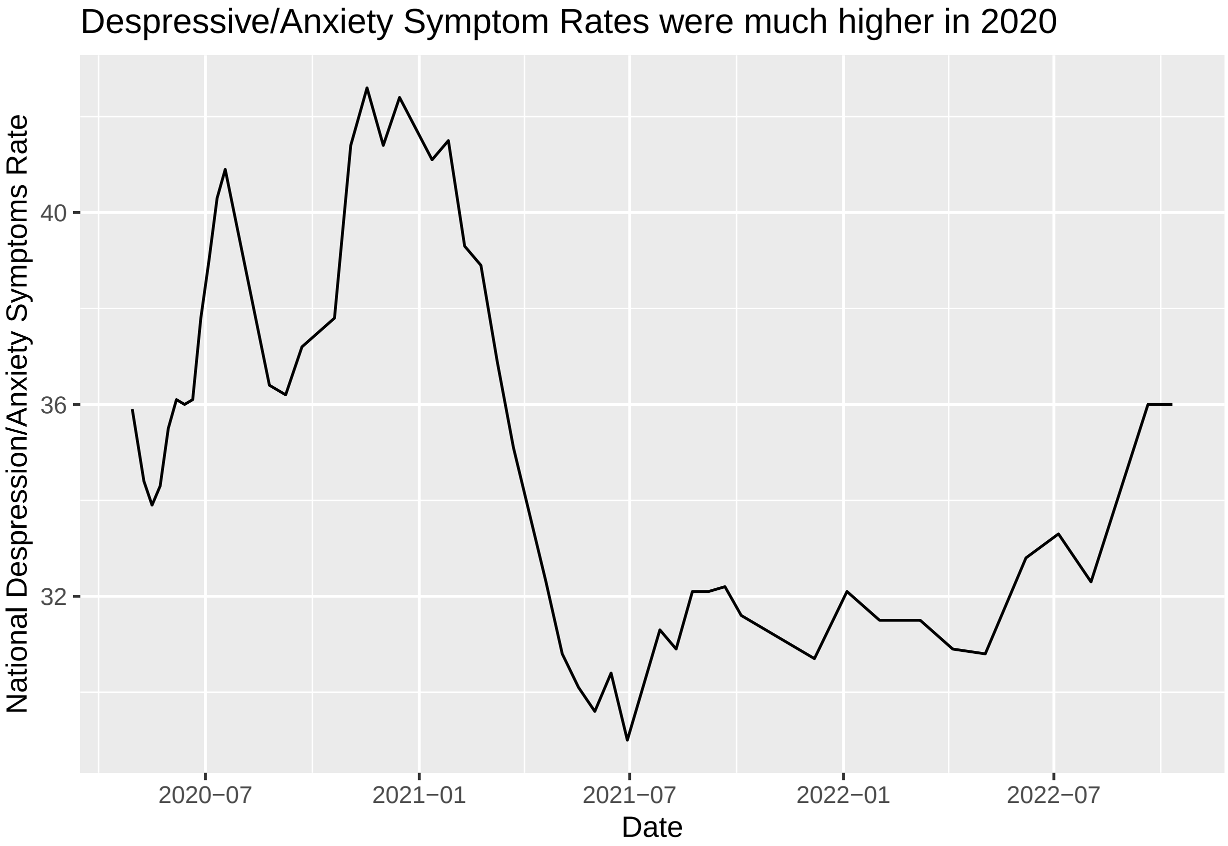
Splitting by year, 2020 is a clear outlier in anxiety and depressive symptom rates, which is more clear when adding the error-bars from the confidence intervals. 2021 and 2022 are very similar. This leads us to wonder whether the emergence of Covid in 2020 had an impact on these increased rates.

```
df %>%
  filter(Group == "National Estimate") %>%
  group_by(Year) %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  summarise(avg = mean(Value), LowCI = mean(`Low CI`), HighCI = mean(`High CI`)) %>%
  ggplot(aes(x = as.factor(Year), y = avg, fill = as.factor(Year))) +
  geom_bar(stat = 'identity') +
  geom_errorbar(aes(ymin = avg, ymax = HighCI)) +
  labs(title = "% with Symptoms of Anxiety Disorder or Depressive Disorder",
       subtitle = "USA data. Statistically higher in 2020",
       x = "",
       y = "") +
  theme(legend.position = "none")
```



This can also be visualized as a line chart, where we again see that anxiety and depressive symptom rates were highest in 2020, but also that they peaked in the winter of 2020.

```
df %>%
  filter(Group == "National Estimate",
         Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  ggplot(aes(x = middleDay, y = Value)) +
  geom_line() +
  labs(x = "Date",
       y = "National Despression/Anxiety Symptoms Rate",
       title = "Despressive/Anxiety Symptom Rates were much higher in 2020")
```



We add in national Covid case counts from the CDC.

```
covid <- read_csv("data_table_for_weekly_case_trends__the_united_states.csv")
```

```
## Rows: 146 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Geography, Date
## dbl (2): Weekly Cases, Historic Cases
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
write.csv(df, file="cleaned_data.csv")
```

Filter the data for national estimates of anxiety and depressive disorder.

```
cleanUS <- df %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  filter(Group == "National Estimate") %>%
  select(Indicator, Value, middleDay)
```

We must do some cleaning on the Covid variable to reach the same format as the middleDay value from the original data.

```
covid <- covid %>%
  mutate(Date_ = gsub(" ", "/", Date)) %>%
   mutate(Date_ = gsub("//", "/", Date_)) %>%
  mutate(Date = as.Date(Date_, "%b/%d/%Y")) %>%
  select(-Geography, -`Historic Cases`,-Date_)
```

Because some of the dates don't line-up due to polling holes in the original data, we manually find the closest dates in the Covid data to append their counts

```
ordered <- covid %>% arrange(Date)

Case_counts <- ordered$`Weekly Cases`[c(14:25, 31,33,35,37,39,41,43,45,47,51,53,55, 57,59,61,65,67,69,7
```

Now our data has the rough amount of Covid counts for the time-periods of our study.

```
cleanUS <- cleanUS %>%
  arrange(middleDay) %>%
  mutate(Weekly_Cases = Case_counts)
```

We add in an "increase" variable which could capture some signal in anxiety and depressive rates by accounting for how much Covid rates have increased from the previous time period.

```
cleanUS <- cleanUS %>% mutate(increase = (Weekly_Cases - lag(Weekly_Cases)) / lag(Weekly_Cases))
```

#Modeling We randomly split our national estimate data into training and testing sets of 70% and 30% respectively, stratified on anxiety and depressive symptom rate value.

```
split <- initial_split(cleanUS, seed = 509, prop=.7, strata = Value)
```

```
## Warning: The number of observations in each quantile is below the recommended threshold of 20.
## * Stratification will use 2 breaks instead.
```

```
train <- training(split)
test <- testing(split)
```

We set up our model workflow as a linear regression model and fit it to the training value.

```
lm_model <- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")

lm_fit <- lm_model %>%
  fit(Value ~ Weekly_Cases + increase + middleDay, data = train)
```

```
summary(lm_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value ~ Weekly_Cases + increase + middleDay,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6702 -2.1986  0.1807  1.8583  6.1815
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.829e+02  3.595e+01   5.089 1.81e-05 ***
## Weekly_Cases  5.039e-06  1.174e-06   4.294 0.000169 ***
## increase     -7.560e-01  6.637e-01  -1.139 0.263667
## middleDay    -8.025e-03  1.923e-03  -4.173 0.000237 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.904 on 30 degrees of freedom
## Multiple R-squared:  0.5216, Adjusted R-squared:  0.4738
## F-statistic:  10.9 on 3 and 30 DF,  p-value: 5.228e-05
```

```
lm_fit <- lm_model %>%
  fit(Value ~ Weekly_Cases + middleDay, data = train)
```

We get a RMSE of 3.1 on the test data with an R^2 of 0.44.

```
results <- predict(lm_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        6.06
```

```
rsq(results, truth = Value, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard     0.00591
```

Next we make a model for the Age group. We filter to age, select the proper variables for modeling, and join with the Covid counts data. We will do this same process of filtering, selecting, and joining for each Grouped variable set.

```
ages <- df %>% filter(Group == "By Age") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

We also do a random stratified split, model fit, and scatterplot for each Grouped variable set.

```
split <- initial_split(ages, seed = 509, prop=.7, strata = Value.x)
train <- training(split)
test <- testing(split)


age_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(age_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8004 -2.6827 -0.1829  2.4724  8.2092
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.981e+02  1.633e+01  12.128  < 2e-16 ***
## Weekly_Cases                1.792e-06  3.646e-07   4.914  1.7e-06 ***
## middleDay                  -7.958e-03  8.730e-04  -9.116  < 2e-16 ***
## Subgroup30 - 39 years      -8.155e+00  8.380e-01  -9.732  < 2e-16 ***
## Subgroup40 - 49 years      -1.243e+01  8.034e-01 -15.470  < 2e-16 ***
## Subgroup50 - 59 years      -1.654e+01  8.124e-01 -20.364  < 2e-16 ***
## Subgroup60 - 69 years      -2.301e+01  8.016e-01 -28.711  < 2e-16 ***
## Subgroup70 - 79 years      -3.006e+01  8.182e-01 -36.732  < 2e-16 ***
## Subgroup80 years and above -3.208e+01  8.461e-01 -37.919  < 2e-16 ***
## increase                   -1.095e+00  2.893e-01  -3.785 0.000197 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.371 on 228 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.9156, Adjusted R-squared:  0.9122
## F-statistic: 274.7 on 9 and 228 DF,  p-value: < 2.2e-16
```

```
results <- predict(age_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

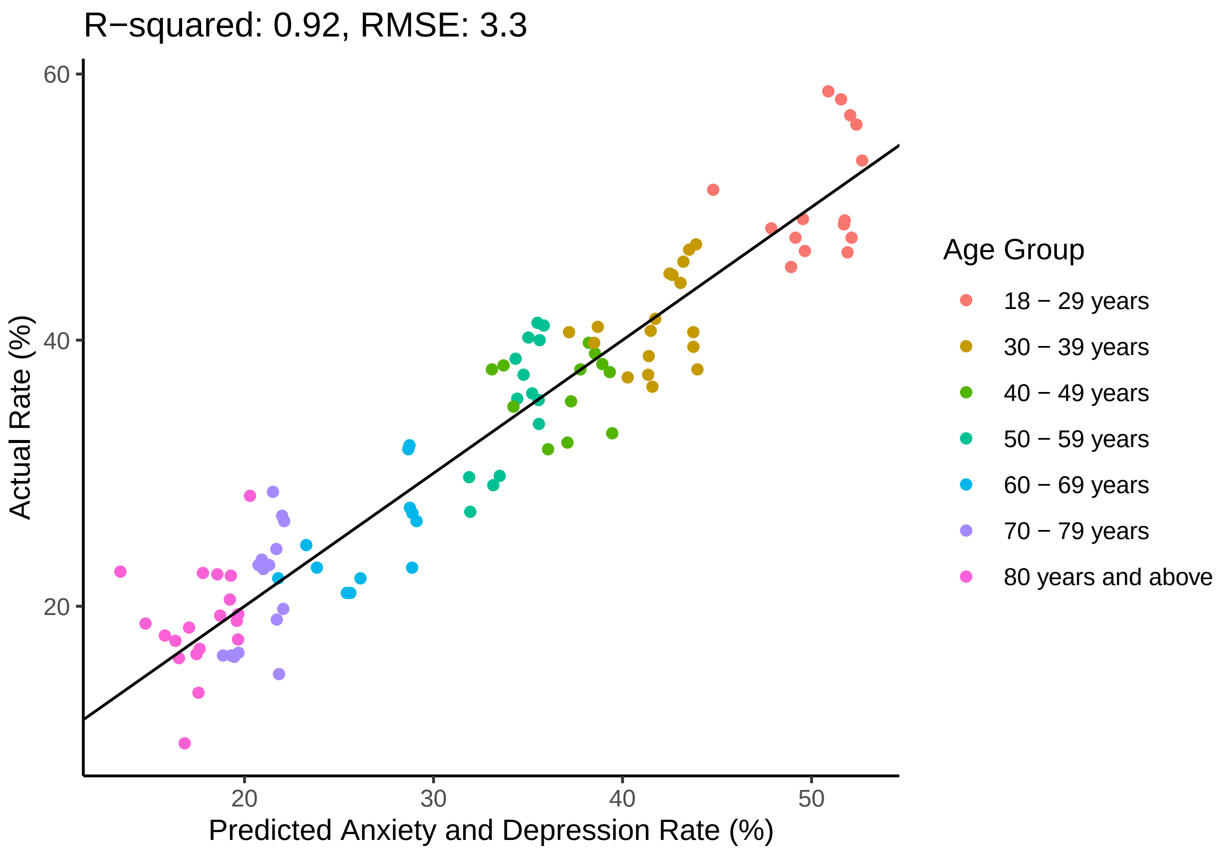
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        3.68
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.902
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.92, RMSE: 3.3",
       color = "Age Group") +
  theme_classic()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



```
age_preds <- results %>%
  filter(!is.na(.pred)) %>%
  select(Subgroup, .pred, Value.x)
```

We get very strong fit values for age, especially considering that we have 7 different subgroups. There is a strong negative trend of age and anxiety and depressive symptom rates during this time period.

We next split by Race/Hispanic ethnicity.

```
races <- df %>% filter(Group == "By Race/Hispanic ethnicity") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

…and fit the linear model.

```
split <- initial_split(races, seed = 509, prop=.7, strata = Value.x)
train <- training(split)
test <- testing(split)


race_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(race_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.736  -2.603  -0.274   2.428   8.501
##
## Coefficients:
##                                                         Estimate Std. Error
## (Intercept)                                            2.224e+02  2.023e+01
## Weekly_Cases                                           1.152e-06  4.087e-07
## middleDay                                             -9.790e-03  1.080e-03
## SubgroupNon-Hispanic Asian, single race               -1.156e+01  8.560e-01
## SubgroupNon-Hispanic Black, single race               -2.241e+00  8.362e-01
## SubgroupNon-Hispanic White, single race               -5.783e+00  8.504e-01
## SubgroupNon-Hispanic, other races and multiple races  5.326e+00  8.454e-01
## increase                                              -7.928e-01  3.309e-01
##                                                       t value Pr(>|t|)
## (Intercept)                                            10.994  < 2e-16 ***
## Weekly_Cases                                            2.819  0.00543 **
## middleDay                                              -9.062 4.17e-16 ***
## SubgroupNon-Hispanic Asian, single race               -13.504  < 2e-16 ***
## SubgroupNon-Hispanic Black, single race                -2.680  0.00813 **
## SubgroupNon-Hispanic White, single race                -6.800 1.94e-10 ***
## SubgroupNon-Hispanic, other races and multiple races   6.300 2.72e-09 ***
## increase                                              -2.396  0.01773 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.49 on 161 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.7636, Adjusted R-squared:  0.7533
## F-statistic: 74.29 on 7 and 161 DF,  p-value: < 2.2e-16
```

```
results <- predict(race_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

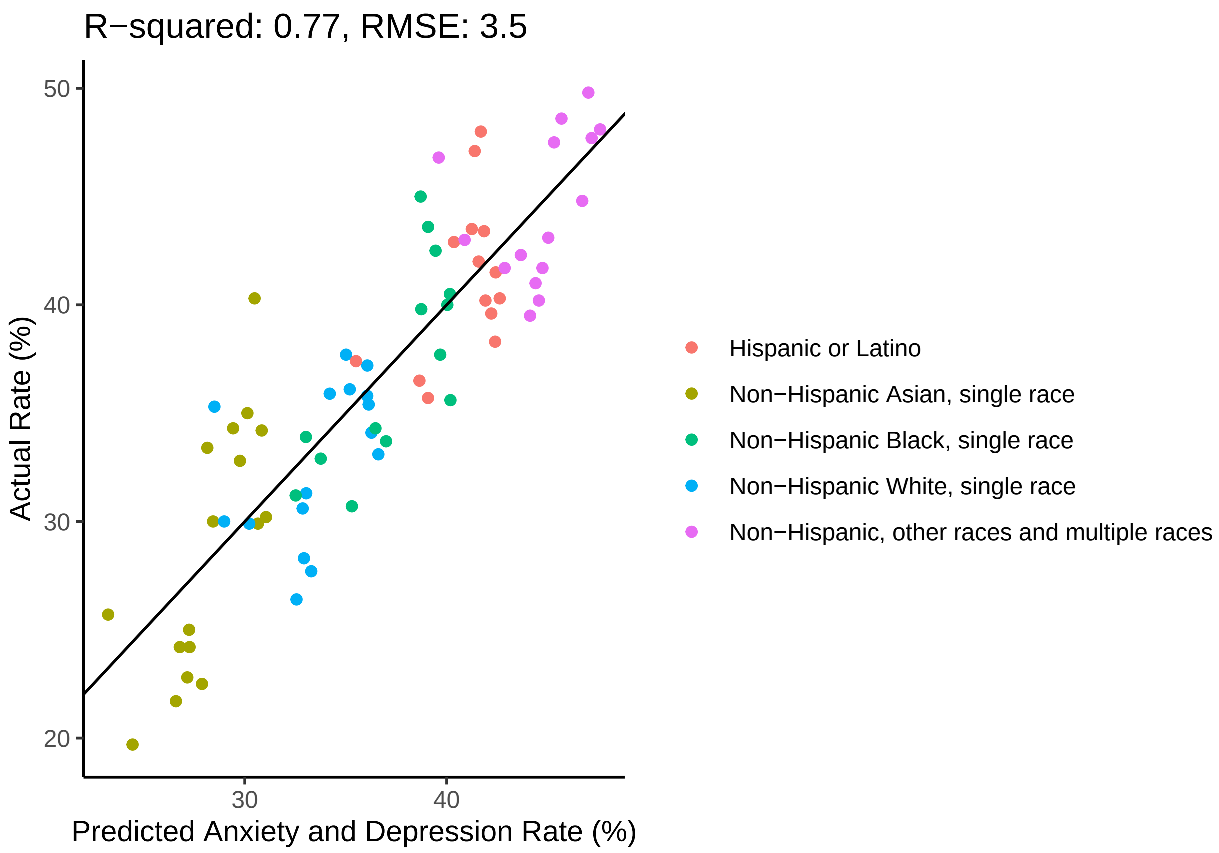
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        3.47
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.765
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.77, RMSE: 3.5",
       color = "") +
  theme_classic() +
  theme(legend.position = "right")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



The predictions are less strong here.

Next we split by education.

```
educations <- df %>% filter(Group == "By Education") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

. . . and fit the model.

```
split <- initial_split(educations, seed = 509, prop=.7, strata = Value.x)
train <- training(split)
test <- testing(split)


edu_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(edu_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9.6219 -2.1522 -0.4456  2.0716  8.9586 
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             1.865e+02  2.318e+01   8.047 4.48e-13
## Weekly_Cases                            1.176e-06  5.017e-07   2.345   0.0205
## middleDay                              -8.442e-03  1.241e-03  -6.802 3.32e-10
## SubgroupHigh school diploma or GED      6.658e+00  9.050e-01   7.357 1.84e-11
## SubgroupLess than a high school diploma 1.320e+01  8.734e-01  15.108  < 2e-16
## SubgroupSome college/Associate's degree 9.882e+00  9.347e-01  10.573  < 2e-16
## increase                               -8.088e-01  3.786e-01  -2.136   0.0345
##
## (Intercept)                            ***
## Weekly_Cases                           *  
## middleDay                              ***
## SubgroupHigh school diploma or GED     ***
## SubgroupLess than a high school diploma ***
## SubgroupSome college/Associate's degree ***
## increase                               *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.659 on 131 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.6846, Adjusted R-squared:  0.6701 
## F-statistic: 47.38 on 6 and 131 DF,  p-value: < 2.2e-16
```

```
results <- predict(edu_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

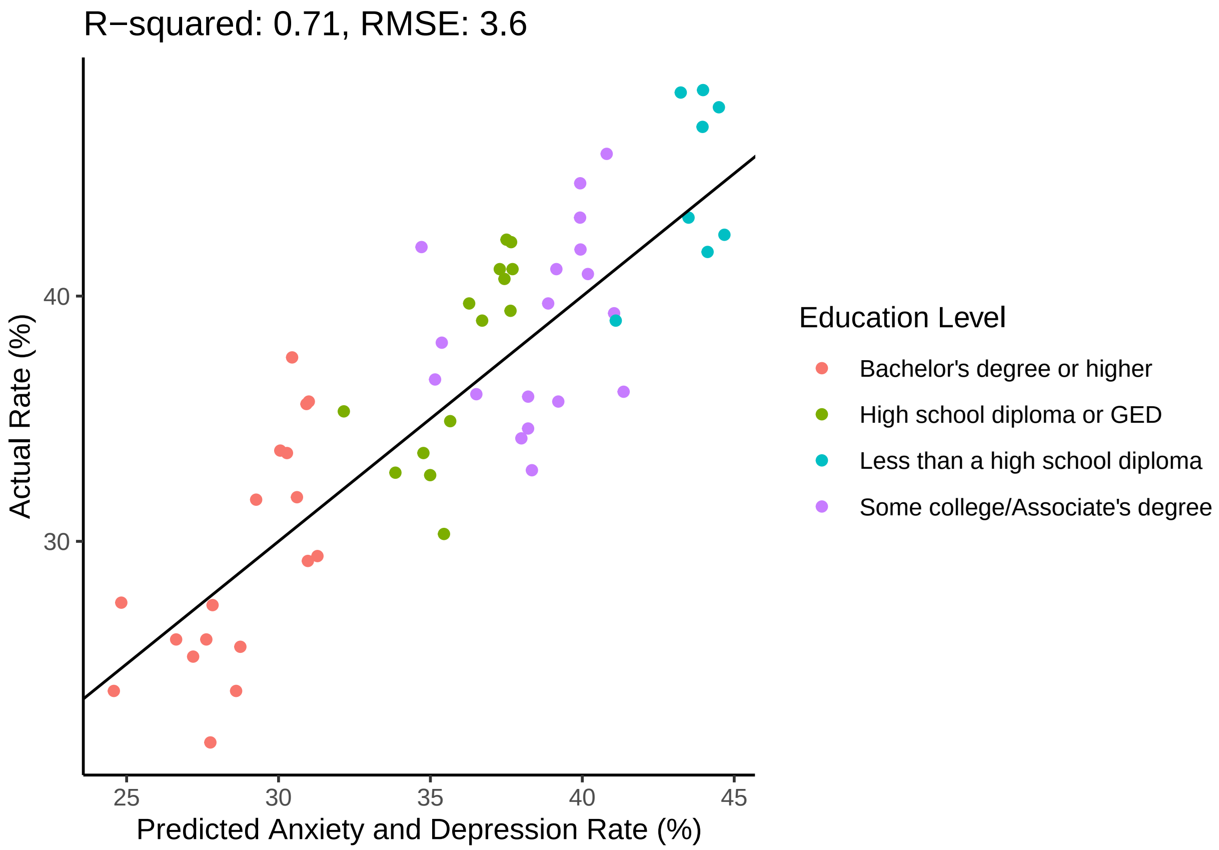
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        3.40
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.747
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.71, RMSE: 3.6",
       color = "Education Level") +
  theme_classic()
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



It appears the more education one has, the less they feel anxiety and depressive symptoms.

Next we split by state.

```
states <- df %>% filter(Group == "By State") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

. . . and model

```
split <- initial_split(states, seed = 509, prop=.7, strata = Value.x)
train <- training(split)
test <- testing(split)


states_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(states_fit$fit)
```

```
##
## Call:
```

```
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup + 
##     increase, data = data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -13.2655  -3.1170  -0.1483   2.8367  13.9150 
## 
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                   1.860e+02  7.810e+00  23.821  < 2e-16 ***
## Weekly_Cases                  2.033e-06  1.773e-07  11.464  < 2e-16 ***
## middleDay                    -8.064e-03  4.162e-04 -19.373  < 2e-16 ***
## SubgroupAlaska               -1.005e+00  1.032e+00  -0.974  0.33028    
## SubgroupArizona              -4.551e-01  1.032e+00  -0.441  0.65917    
## SubgroupArkansas              1.966e+00  1.005e+00   1.955  0.05073 .  
## SubgroupCalifornia            1.084e+00  1.063e+00   1.019  0.30814    
## SubgroupColorado             -1.115e+00  1.039e+00  -1.073  0.28336    
## SubgroupConnecticut          -3.199e+00  1.024e+00  -3.123  0.00182 ** 
## SubgroupDelaware             -4.371e+00  1.018e+00  -4.294 1.85e-05 ***
## SubgroupDistrict of Columbia -1.021e+00  1.039e+00  -0.983  0.32580    
## SubgroupFlorida              -1.120e-01  1.031e+00  -0.109  0.91354    
## SubgroupGeorgia              -7.674e-01  1.055e+00  -0.728  0.46689    
## SubgroupHawaii               -4.137e+00  1.005e+00  -4.115 4.06e-05 ***
## SubgroupIdaho                -3.088e+00  1.031e+00  -2.994  0.00280 ** 
## SubgroupIllinois             -1.805e+00  1.005e+00  -1.795  0.07284 .  
## SubgroupIndiana              -9.230e-01  1.012e+00  -0.912  0.36171    
## SubgroupIowa                 -5.072e+00  1.063e+00  -4.770 2.00e-06 ***
## SubgroupKansas               -2.513e+00  1.005e+00  -2.500  0.01252 *  
## SubgroupKentucky              1.212e+00  1.025e+00   1.183  0.23713    
## SubgroupLouisiana             3.900e+00  1.039e+00   3.754  0.00018 ***
## SubgroupMaine                -2.929e+00  1.005e+00  -2.914  0.00362 ** 
## SubgroupMaryland             -2.560e+00  1.082e+00  -2.366  0.01807 *  
## SubgroupMassachusetts        -3.068e+00  1.031e+00  -2.974  0.00298 ** 
## SubgroupMichigan             -2.584e+00  1.031e+00  -2.505  0.01234 *  
## SubgroupMinnesota            -6.127e+00  1.127e+00  -5.436 6.24e-08 ***
## SubgroupMississippi           3.107e+00  1.055e+00   2.946  0.00326 ** 
## SubgroupMissouri             -1.428e+00  1.047e+00  -1.364  0.17260    
## SubgroupMontana              -4.238e+00  1.018e+00  -4.164 3.29e-05 ***
## SubgroupNebraska             -6.002e+00  1.093e+00  -5.493 4.54e-08 ***
## SubgroupNevada                2.284e+00  1.031e+00   2.214  0.02694 *  
## SubgroupNew Hampshire        -4.797e+00  1.047e+00  -4.584 4.90e-06 ***
## SubgroupNew Jersey           -2.698e+00  1.046e+00  -2.578  0.01001 *  
## SubgroupNew Mexico            1.012e+00  1.039e+00   0.974  0.33012    
## SubgroupNew York             -2.164e+00  1.073e+00  -2.018  0.04378 *  
## SubgroupNorth Carolina       -1.970e+00  1.063e+00  -1.853  0.06404 .  
## SubgroupNorth Dakota         -7.201e+00  1.039e+00  -6.932 5.86e-12 ***
## SubgroupOhio                 -1.104e+00  1.072e+00  -1.030  0.30326    
## SubgroupOklahoma              2.806e+00  1.032e+00   2.719  0.00661 ** 
## SubgroupOregon                1.697e+00  1.072e+00   1.582  0.11385    
## SubgroupPennsylvania         -1.805e+00  1.039e+00  -1.738  0.08239 .  
## SubgroupRhode Island         -2.651e+00  1.012e+00  -2.621  0.00885 ** 
## SubgroupSouth Carolina       -2.822e+00  1.018e+00  -2.772  0.00563 ** 
## SubgroupSouth Dakota         -7.215e+00  1.031e+00  -6.995 3.80e-12 ***
## SubgroupTennessee            -6.395e-01  1.055e+00  -0.606  0.54442    
## SubgroupTexas                 1.434e+00  1.063e+00   1.348  0.17775    
## SubgroupUtah                 -1.881e+00  9.998e-01  -1.881  0.06011 .  
## SubgroupVermont              -3.154e+00  1.047e+00  -3.013  0.00262 ** 
## SubgroupVirginia             -3.127e+00  1.031e+00  -3.032  0.00247 ** 
## SubgroupWashington            4.197e-02  1.011e+00   0.041  0.96690    
## SubgroupWest Virginia         1.974e+00  1.011e+00   1.952  0.05113 .  
## SubgroupWisconsin            -6.582e+00  1.005e+00  -6.546 7.80e-11 ***
## SubgroupWyoming              -2.889e+00  1.039e+00  -2.781  0.00547 ** 
## increase                     -8.434e-01  1.428e-01  -5.905 4.24e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.25 on 1700 degrees of freedom
##   (29 observations deleted due to missingness)
## Multiple R-squared:  0.3929, Adjusted R-squared:  0.374 
## F-statistic: 20.76 on 53 and 1700 DF,  p-value: < 2.2e-16
```

```
results <- predict(states_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

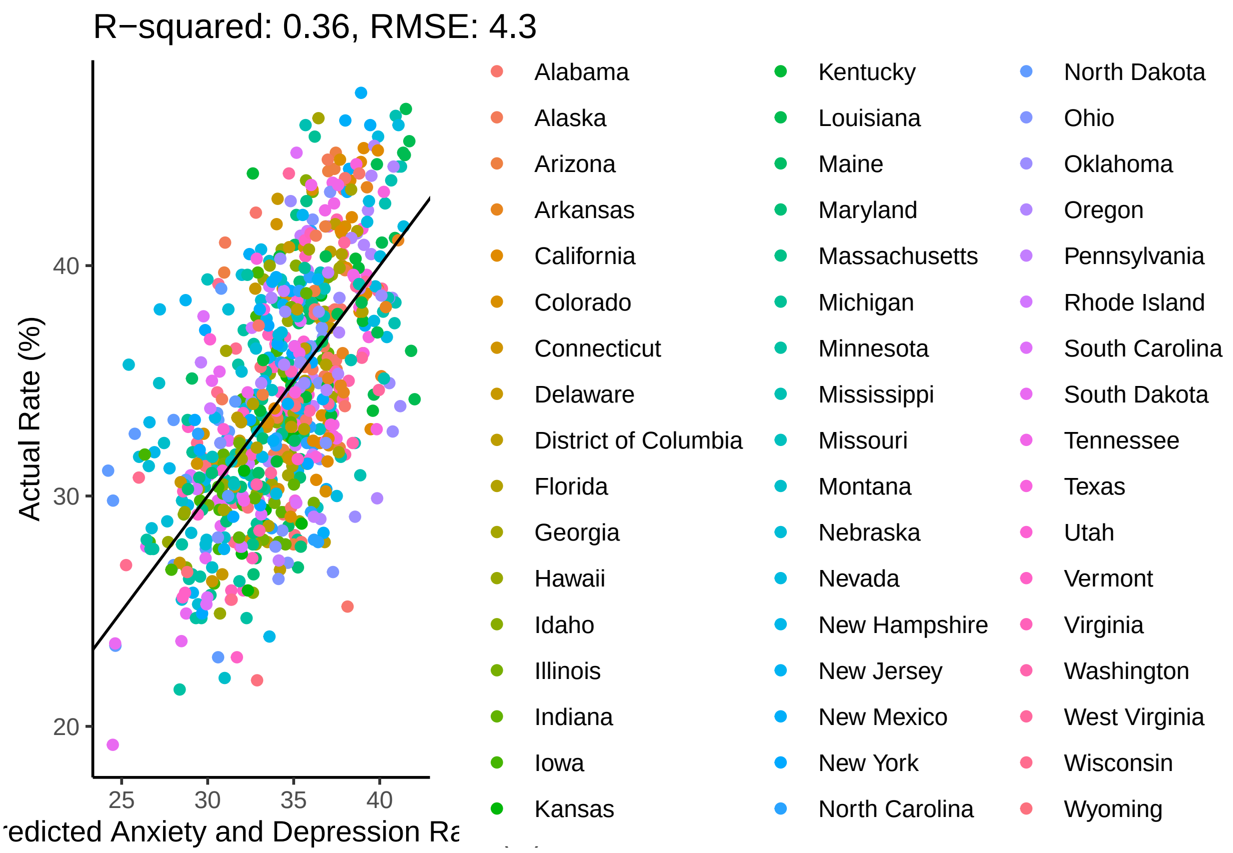
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        4.16
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.353
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.36, RMSE: 4.3",
       color = "") +
  theme_classic()
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```



```
state_preds <- results %>% filter(!is.na(.pred)) %>%
  select(Subgroup, Value.x, .pred)
```

The model is not very strong but that is to be expected with so many different states to capture effects for. Next we split by sex.

```
sex <- df %>% filter(Group == "By Sex") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

. . . and fit the model.

```
split <- initial_split(sex, seed = 509, prop=.7, strata = Value.x)
train <- training(split)
test <- testing(split)


sex_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(sex_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4396 -2.4949  0.1576  2.3164  6.1863
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.324e+02  2.853e+01   8.145 2.48e-11 ***
## Weekly_Cases  3.876e-06  8.428e-07   4.599 2.19e-05 ***
## middleDay    -1.044e-02  1.529e-03  -6.829 4.55e-09 ***
## SubgroupMale -8.100e+00  7.737e-01 -10.470 2.99e-15 ***
## increase     -3.246e-01  6.190e-01  -0.524    0.602
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.061 on 61 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7134, Adjusted R-squared:  0.6946
## F-statistic: 37.96 on 4 and 61 DF,  p-value: 6.353e-16
```

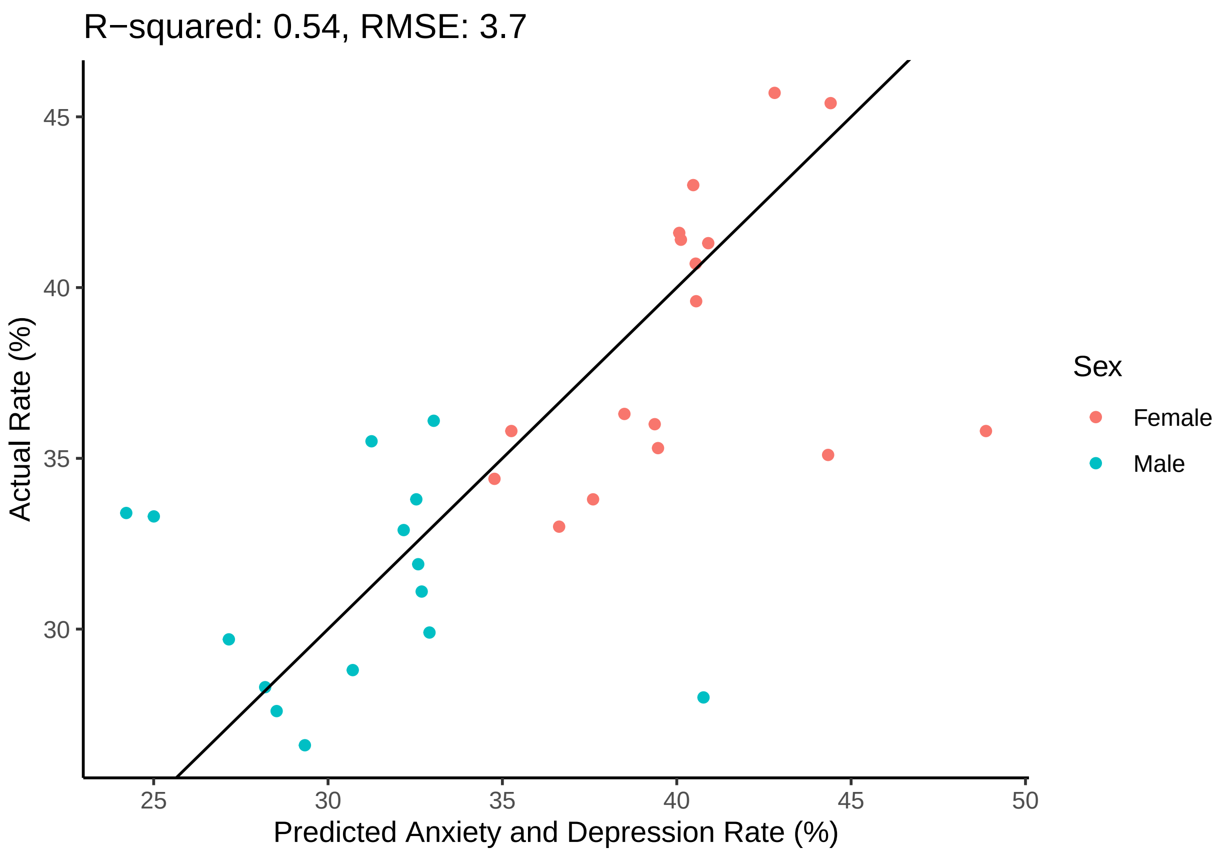
```
results <- predict(sex_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        4.73
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.429
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.54, RMSE: 3.7",
       color = "Sex") +
  theme_classic()
```



Females have higher levels of anxiety and depresssive symptoms than males but the fit is not very strong.

Now we split by disability status.

```
disability <- df %>% filter(Group == "By Disability status") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

```
split <- initial_split(disability, seed = 509, prop=.7, strata = Value.x)
```

```
## Warning: The number of observations in each quantile is below the recommended threshold of 20.
## * Stratification will use 2 breaks instead.
```

```
train <- training(split)
test <- testing(split)

disability_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(disability_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6548 -0.7586  0.1255  0.9160  2.4618
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2.978e+01  3.064e+01  -0.972  0.33985
## Weekly_Cases                 8.455e-08  3.107e-07   0.272  0.78760
## middleDay                    4.835e-03  1.619e-03   2.987  0.00594 **
## SubgroupWithout disability -3.535e+01  5.462e-01 -64.723  < 2e-16 ***
## increase                    -1.734e-01  2.358e-01  -0.736  0.46831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.536 on 27 degrees of freedom
## Multiple R-squared:  0.9936, Adjusted R-squared:  0.9927
## F-statistic:  1054 on 4 and 27 DF,  p-value: < 2.2e-16
```

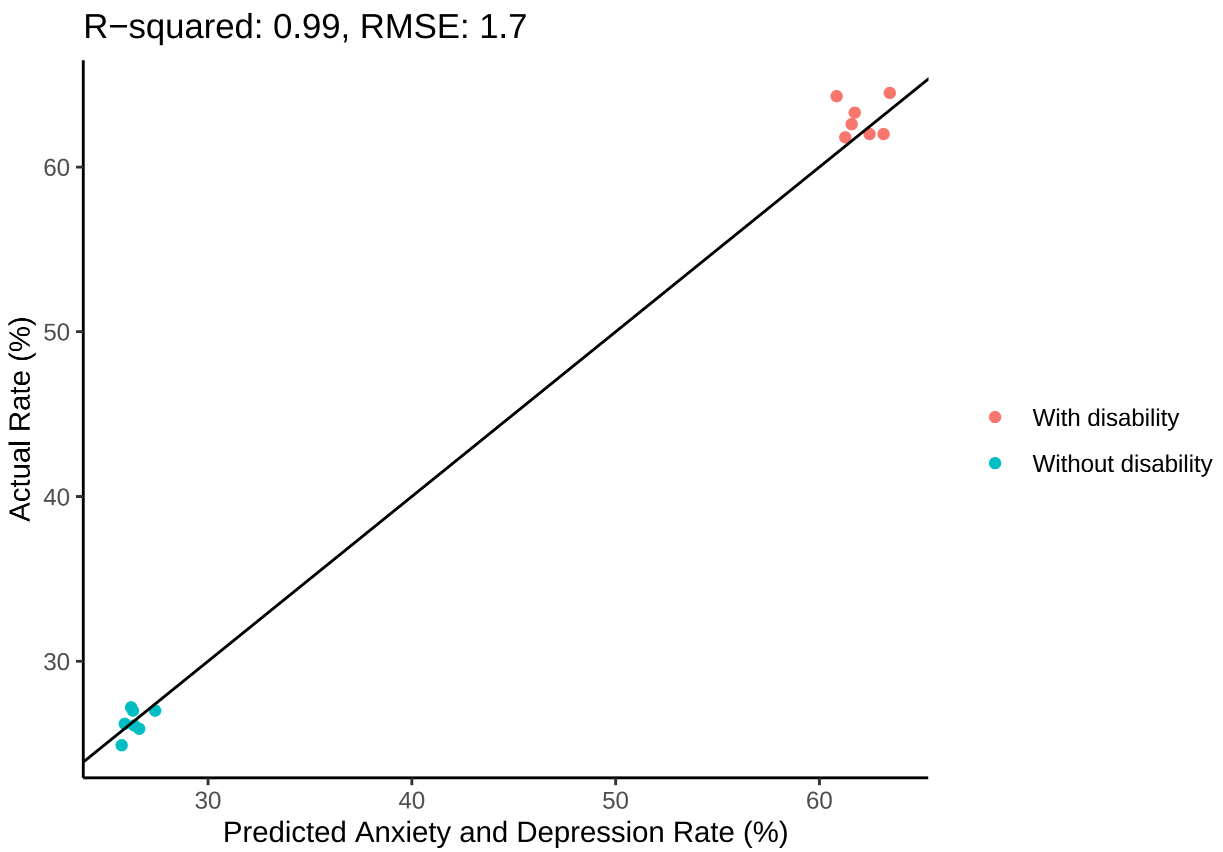
```
results <- predict(disability_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        1.24
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.996
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.99, RMSE: 1.7",
       color = "") +
  theme_classic()
```



This gives us our strongest fit, with people with disabilities with noticebaly higher rates of anxiety and depressive sympoms at around 60%.

Next, we split by gender identity.

```
gender <- df %>% filter(Group == "By Gender identity") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

. . . and model

```
split <- initial_split(gender, seed = 509, prop=.7, strata = Value.x)
```

```
## Warning: The number of observations in each quantile is below the recommended threshold of 20.
## * Stratification will use 2 breaks instead.
```

```
train <- training(split)
test <- testing(split)

gender_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(gender_fit$fit)
```

```
##
## Call:
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1597  -1.5550  -0.0343   1.4756  10.1951
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.119e+02  9.197e+01  -1.217 0.233526
## Weekly_Cases           -1.160e-06  7.925e-07  -1.464 0.154055
## middleDay               7.756e-03  4.822e-03   1.608 0.118602
## SubgroupCis-gender male -6.776e+00  1.542e+00  -4.395 0.000136 ***
## SubgroupTransgender     3.126e+01  1.674e+00  18.674  < 2e-16 ***
## increase                4.444e-01  6.009e-01   0.740 0.465495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.894 on 29 degrees of freedom
## Multiple R-squared:  0.9494, Adjusted R-squared:  0.9406
## F-statistic: 108.8 on 5 and 29 DF,  p-value: < 2.2e-16
```

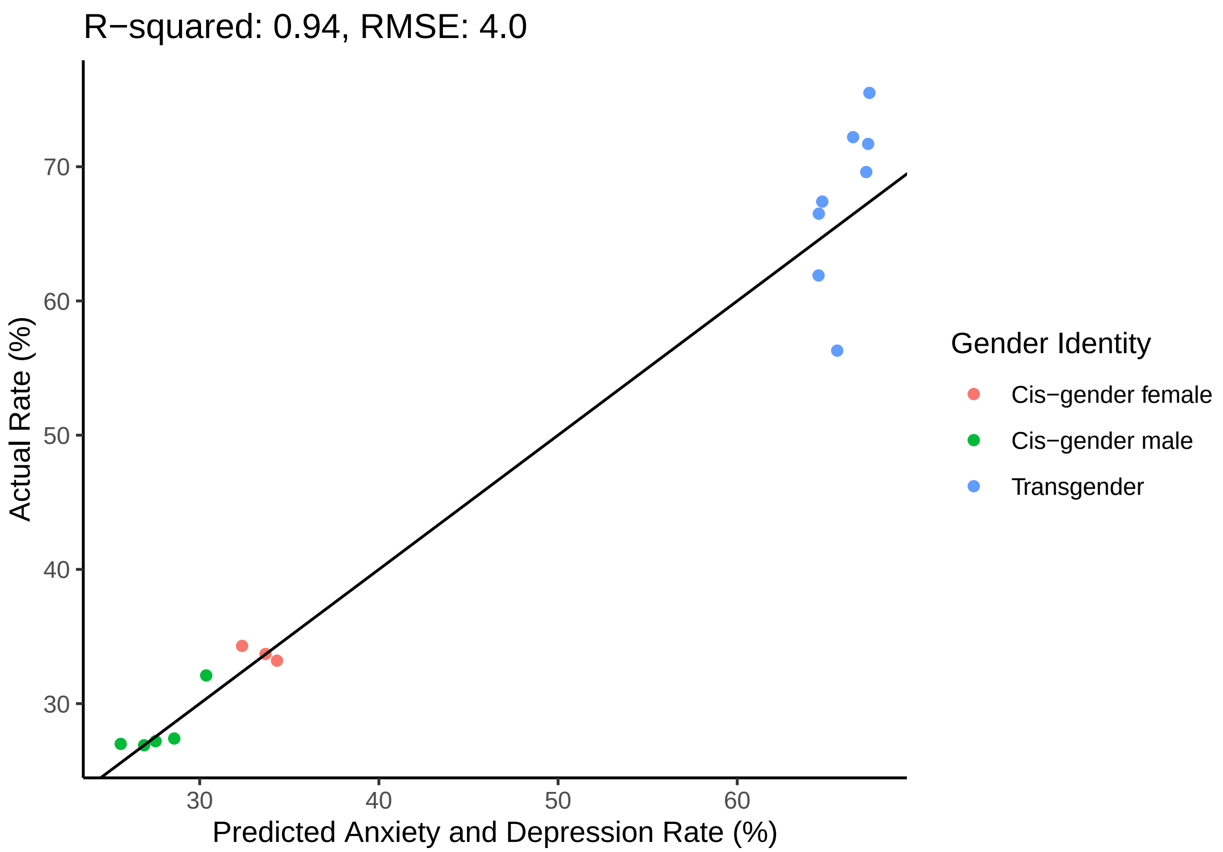
```
results <- predict(gender_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        3.87
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.964
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.94, RMSE: 4.0",
       color = "Gender Identity") +
  theme_classic()
```



Again, we have a very strong fit, with transgender people having anxiety and depressive symptom rates much higher than other individuals.

We split by sexual orientation.

```
orientation <- df %>% filter(Group == "By Sexual orientation") %>%
  filter(Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
  select(Subgroup, Value, middleDay, Year, Month) %>%
  left_join(cleanUS, by = 'middleDay') %>%
  select(-Indicator, - Value.y)
```

```
split <- initial_split(orientation, seed = 509, prop=.7, strata = Value.x)
```

```
## Warning: The number of observations in each quantile is below the recommended threshold of 20.
## * Stratification will use 2 breaks instead.
```

```
train <- training(split)
test <- testing(split)

orientation_fit <- lm_model %>%
  fit(Value.x ~ Weekly_Cases + middleDay + Subgroup + increase, data = train)

summary(orientation_fit$fit)
```

```
##
## Call:
```

```
## stats::lm(formula = Value.x ~ Weekly_Cases + middleDay + Subgroup +
##     increase, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0084 -1.8310 -0.9004  2.6234  5.3630
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              9.000e+01  6.991e+01   1.287    0.208
## Weekly_Cases             1.081e-07  5.769e-07   0.187    0.853
## middleDay               -1.499e-03  3.666e-03  -0.409    0.686
## SubgroupGay or lesbian -1.537e+01  1.256e+00 -12.242 5.57e-13 ***
## SubgroupStraight       -3.170e+01  1.332e+00 -23.801  < 2e-16 ***
## increase               -2.219e-01  4.481e-01  -0.495    0.624
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.971 on 29 degrees of freedom
## Multiple R-squared:  0.9535, Adjusted R-squared:  0.9455
## F-statistic:   119 on 5 and 29 DF,  p-value: < 2.2e-16
```

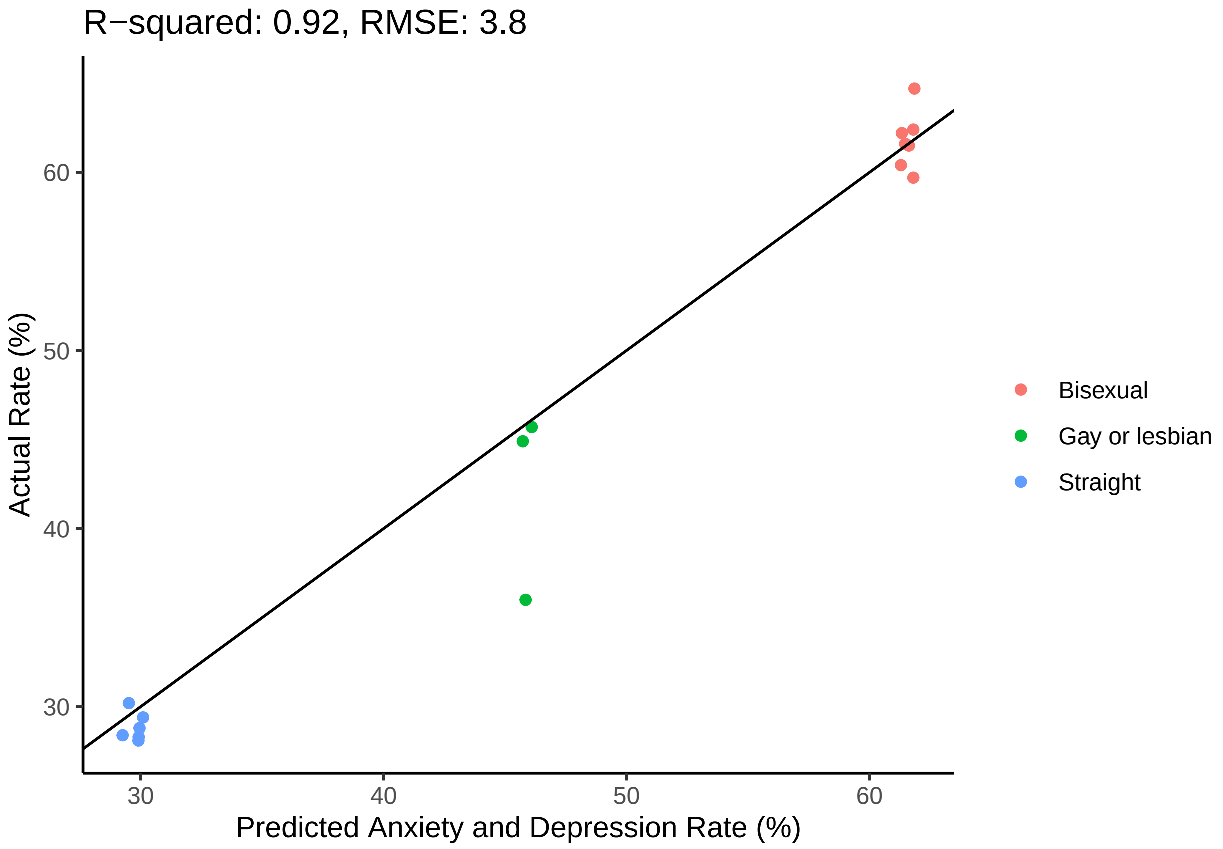
```
results <- predict(orientation_fit, new_data = test) %>% bind_cols(test)
rmse(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        2.75
```

```
rsq(results, truth = Value.x, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.972
```

```
ggplot(data = results, mapping = aes(.pred, Value.x, color = Subgroup)) +
  geom_point() +
  geom_abline(intercept = 0, slope=1) +
  labs(x = "Predicted Anxiety and Depression Rate (%)",
       y = "Actual Rate (%)",
       title = "R-squared: 0.92, RMSE: 3.8",
       color = "") +
  theme_classic()
```



We get a strong fit, especially considering we have three subgroups, with bisexual people having higher symptom rates than gay/lesbian people than straight people.

#Model Comparison Now we want to compare the effectiveness of the models.

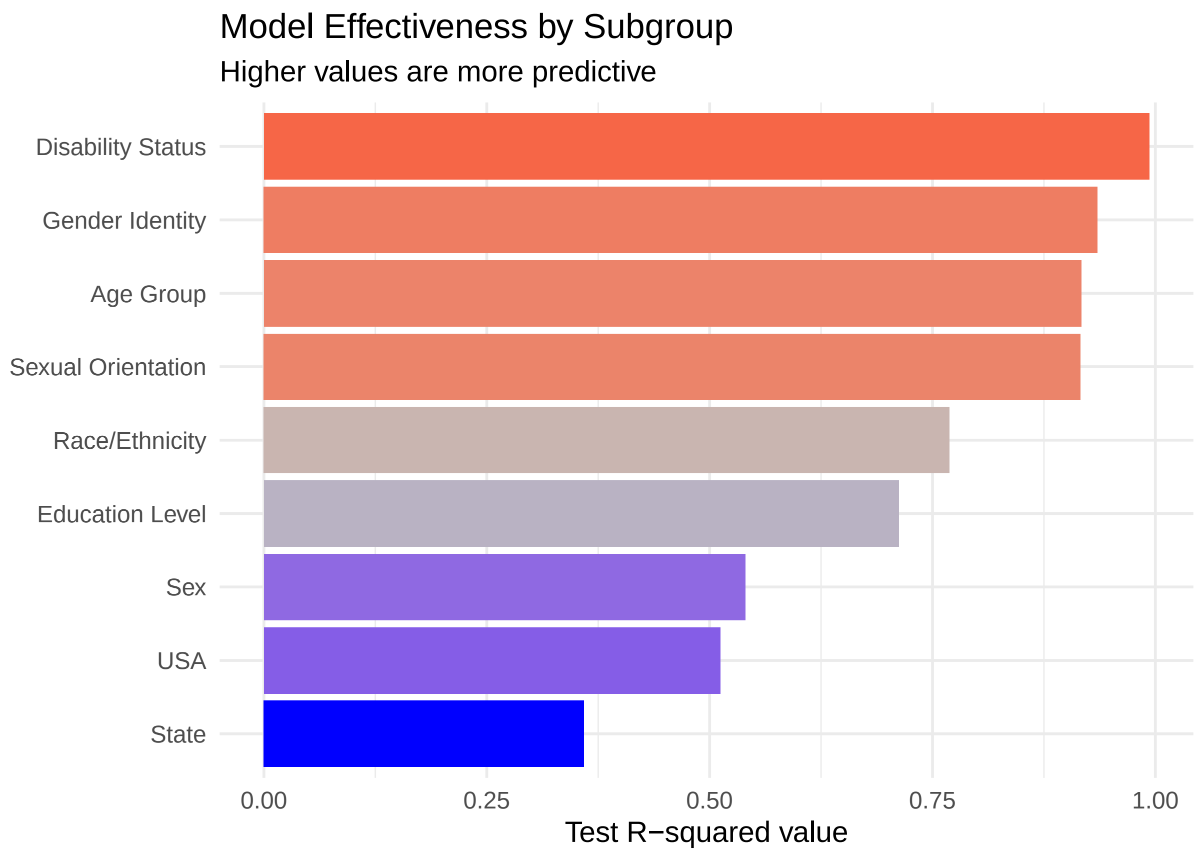
```
subgroups <- c("Disability Status", "Gender Identity", "Age Group", "Sexual Orientation", "Race/Ethnici

R2 <- c(0.993, 0.935, 0.917, 0.916, 0.769, 0.712, 0.540, 0.512, 0.359)

models <- data.frame(Subgroup = subgroups,
                     R2 = R2)
```

The chart below shows the test Rˆ2 value of each model which was split by Group. Higher Rˆ2 values indicate anxiety and depressive symptom rates which were easier to predict, which implies more "divided" rates within the group. Disability status, gender identity, and age group were the most predictable, while state, national rates, and sex were the least predictive.

```
models %>%
  ggplot(aes(fill = R2, x = R2, y = reorder(Subgroup, R2))) +
  geom_bar(stat = "identity") +
  labs(x = "Test R-squared value",
       y = "",
       title = "Model Effectiveness by Subgroup",
       subtitle = "Higher values are more predictive") +
  scale_fill_gradient2(low = "blue", high ="red", midpoint = mean(models$R2), mid = "gray") +
  theme_minimal() +
  theme(legend.position = "none")
```



Here, we write out several files for other analyses.

```
write.csv(state_data, file="state_data.csv")
state_ts <-  df %>%
  filter(Group == "By State", Indicator == "Symptoms of Anxiety Disorder or Depressive Disorder") %>%
   select(middleDay, State, Value)

write.csv(state_ts, file="state_ts.csv")

write.csv(state_preds, file = "state_preds.csv")

write.csv(age_preds, file = "age_preds.csv")

write.csv(models, file = "model_scores.csv")
```