Final Project

Section 5

Ella Morrow

Due: Friday, December 18

Step 1 (Load Data)

Use the code chunk provided below to read your data set into R. *You should use the same dataset you used in HW2–HW5.* Then, perform any transformations and/or filtering that you will need for your regression models.

```
#covid <- read_csv("covid_behaviors_US.csv", guess_max = 14000)

covid <- read.csv("covid_behaviors_US.csv")

covid <- covid  %>%
  mutate(across(starts_with("i12_health"), ~as.numeric(factor(.,c('Not at all','Rarel
y','Sometimes','Frequently','Always'))),.names = "{col}_N")) %>%
  mutate(score = i12_health_1_N+i12_health_2_N+i12_health_3_N+i12_health_4_N+i12_heal
th_5_N+i12_health_6_N+i12_health_7_N+i12_health_8_N+i12_health_11_N+i12_health_12_N+i
12_health_13_N+i12_health_14_N+i12_health_15_N+i12_health_16_N+i12_health_17_N+i12_he
alth_18_N+i12_health_19_N+i12_health_20_N)

covid <- covid %>%
  left_join(data.frame(state.name, state.region, state.division),
            by = c("region" = "state.name"))

covid <- covid %>%
  mutate(datetime = lubridate::dmy_hm(endtime)) %>% #R will recognize datetime as a d
ate -- you can summarize with min and max
  mutate(month = factor(lubridate::month(datetime))) %>% #creates month variable that
indicates the month of the survey response
  mutate(week = factor(lubridate::epiweek(datetime))) #creates week variable that ind
icates the week of the year of the survey response

covid <- covid %>% mutate(scared = recode(WCRV_4, `I am very scared that I will contr
act the Coronavirus (COVID-19)` = "Very", `I am fairly scared that I will contract th
e Coronavirus (COVID-19)` = "Fairly", `I am not very scared that I will contract the
Coronavirus (COVID-19)` = "Not very", `I am not at all scared that I will contract th
e Coronavirus (COVID-19)` = "Not at all"))

covid <- covid %>% mutate(scared = factor(scared, levels=c("Not at all", "Not very",
"Fairly", "Very")))

covid <- covid %>% mutate(highscore = score >= 60)

covid <- covid %>% mutate(ageCat = cut(age, 4))
```

Step 2 (Add Variables to Linear Regression Model)

Consider your “best” multiple linear regression model from HW3 and think about at least two additional explanatory variables that you would like to add to that model. Fit this larger model.

```
covid_cc <- covid %>%
  filter(!is.na(scared))

lm.mod.full <- covid_cc %>% with(lm(score ~ age * gender + month + scared))
```

Step 3 (Hypothesis Testing for Linear Regression Coefficients)

Considering this new, larger multiple linear regression model, use hypothesis testing for each individual slope coefficient to consider the evidence you have in support of including those variables in the model. Consider whether some of these variables may not have REAL relationships with the outcome after accounting for the other variables.

```
tidy(lm.mod.full)
```

```
## # A tibble: 11 x 5
##    term           estimate std.error statistic   p.value
##    <chr>             <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     56.6       0.751      75.3  0.
##  2 age              0.0853    0.0109      7.85 4.71e- 15
##  3 genderMale       2.11      0.833       2.54 1.12e-  2
##  4 month5          -1.34      0.476      -2.82 4.77e-  3
##  5 month6          -4.79      0.523      -9.16 6.23e- 20
##  6 month7          -4.67      0.517      -9.05 1.77e- 19
##  7 month8          -5.58      0.582      -9.59 1.12e- 21
##  8 scaredNot very  11.0       0.408      26.9  1.92e-153
##  9 scaredFairly    16.8       0.401      41.9  0.
## 10 scaredVery      20.7       0.460      45.1  0.
## 11 age:genderMale  -0.0909    0.0160     -5.68 1.42e-  8
```

Step 4 (Compare Nested Linear Models)

Now fit a model without some of those variables in `lm.mod.full` that may not have REAL relationships after accounting for the other variables. Use a nested hypothesis test to compare `lm.mod.full` with a smaller model, `lm.mod.sub` . The null hypothesis is that the smaller model is correct. Consider whether you have evidence to reject that hypothesis in favor of the full model (`lm.mod.full`).

```
lm.mod.sub <- covid_cc %>% with(lm(score ~ age+gender+month+scared))

anova(lm.mod.full, lm.mod.sub)
```

```
## Analysis of Variance Table
##
## Model 1: score ~ age * gender + month + scared
## Model 2: score ~ age + gender + month + scared
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1   9091 1569786
## 2   9092 1575350 -1   -5564.3 32.224 1.416e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 5 (Select a Final Linear Regression Model)

Using the tools available to you (residual plots, R-squared, adjusted R-squared, standard deviation of residuals, hypothesis testing, causal diagrams), fit a variety of models and choose one final model. Be systematic in your process as you'll need to describe your model selection process and justify your final model. (Note: for mastery of the Inference > Model Selection objective (see Final Grading Rubric), you must use at least 3 of these model selection tools.)

```
lm.mod.ageCat <- covid %>% with(lm(score ~ ageCat + gender + month + scared))

lm.mod.noscared <- covid %>% with(lm(score ~ age+gender + month))

lm.mod.nomonth <- covid %>% with(lm(score ~ age + gender + scared))

lm.mod.nogender <- covid %>% with(lm(score ~ age + month + scared))

lm.mod.state  <-covid %>% with(lm(score ~ age+gender + state.region + scared))

glance(lm.mod.state)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.226         0.225  13.3      331.       0     8 -36308. 72636. 72707.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance(lm.mod.full)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.243         0.242  13.1      292.       0    10 -36354. 72731. 72817.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
final.lm.mod <- covid %>% with(lm(score ~ age * gender + month + scared))

tidy(final.lm.mod) #you'll need these estimates
```

```
## # A tibble: 11 x 5
##    term           estimate std.error statistic   p.value
##    <chr>             <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     56.6       0.751      75.3  0.
##  2 age              0.0853    0.0109      7.85 4.71e- 15
##  3 genderMale       2.11      0.833       2.54 1.12e-  2
##  4 month5          -1.34      0.476      -2.82 4.77e-  3
##  5 month6          -4.79      0.523      -9.16 6.23e- 20
##  6 month7          -4.67      0.517      -9.05 1.77e- 19
##  7 month8          -5.58      0.582      -9.59 1.12e- 21
##  8 scaredNot very  11.0       0.408      26.9  1.92e-153
##  9 scaredFairly    16.8       0.401      41.9  0.
## 10 scaredVery      20.7       0.460      45.1  0.
## 11 age:genderMale  -0.0909    0.0160     -5.68 1.42e-  8
```

```
confint(final.lm.mod) #you'll need these confidence intervals
```

```
##                       2.5 %     97.5 %
## (Intercept)    55.12669340 58.0722826
## age             0.06395889  0.1065481
## genderMale      0.48159581  3.7479229
## month5         -2.27438659 -0.4100922
## month6         -5.81675129 -3.7663479
## month7         -5.68634926 -3.6610754
## month8         -6.71754106 -4.4373858
## scaredNot very 10.18798791 11.7894191
## scaredFairly   16.02222482 17.5952429
## scaredVery     19.83274807 21.6365337
## age:genderMale -0.12228880 -0.0595109
```

```
glance(final.lm.mod) #you'll need these model evaluation criteria
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.243         0.242  13.1      292.       0    10 -36354. 72731. 72817.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Step 6 (Add Variables to Logistic Regression Model)

Now consider your multiple logistic regression model from HW4 and think about at least two additional explanatory variables that you would like to add to that model. Fit this larger model.

```
glm.mod.full <-  covid_cc %>%
  with(glm(highscore ~ age + gender + month + scared, family = binomial))
```

Step 7 (Hypothesis Testing for Logistic Regression Coefficients)

Considering this new, larger multiple logistic regression model, use hypothesis testing for each individual slope coefficient to consider the evidence you have in support of including those variables in the model.

```
tidy(glm.mod.full)
```

```
## # A tibble: 10 x 5
##    term           estimate std.error statistic   p.value
##    <chr>             <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     0.309     0.139        2.22 2.62e-  2
##  2 age             0.00654   0.00165      3.95 7.75e-  5
##  3 genderMale     -0.302     0.0571      -5.29 1.25e-  7
##  4 month5         -0.304     0.111       -2.74 6.14e-  3
##  5 month6         -0.882     0.116       -7.57 3.72e- 14
##  6 month7         -0.901     0.116       -7.76 8.69e- 15
##  7 month8         -0.946     0.127       -7.44 1.00e- 13
##  8 scaredNot very  1.28      0.0678      18.9  1.51e- 79
##  9 scaredFairly    2.46      0.0815      30.2  5.53e-200
## 10 scaredVery      2.80      0.111       25.1  3.50e-139
```

```
confint(glm.mod.full) %>% exp()
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %     97.5 %
## (Intercept)     1.0388802  1.7898811
## age             1.0033033  1.0098313
## genderMale      0.6610825  0.8269800
## month5          0.5920653  0.9147250
## month6          0.3286413  0.5189720
## month7          0.3227447  0.5088972
## month8          0.3022092  0.4974812
## scaredNot very  3.1539366  4.1149317
## scaredFairly    9.9926404 13.7573778
## scaredVery     13.2270226 20.4726222
```

Question 8 (Compare Nested Logistic Models)

Now fit a model without some of those variables in `glm.mod.full` that may not have REAL relationships after accounting for the other variables. Use a nested hypothesis test to compare `glm.mod.full` with a smaller model, `glm.mod.sub` . The null hypothesis is that the smaller model is correct. Consider whether you have evidence to reject that hypothesis in favor of the full model (`glm.mod.full`).

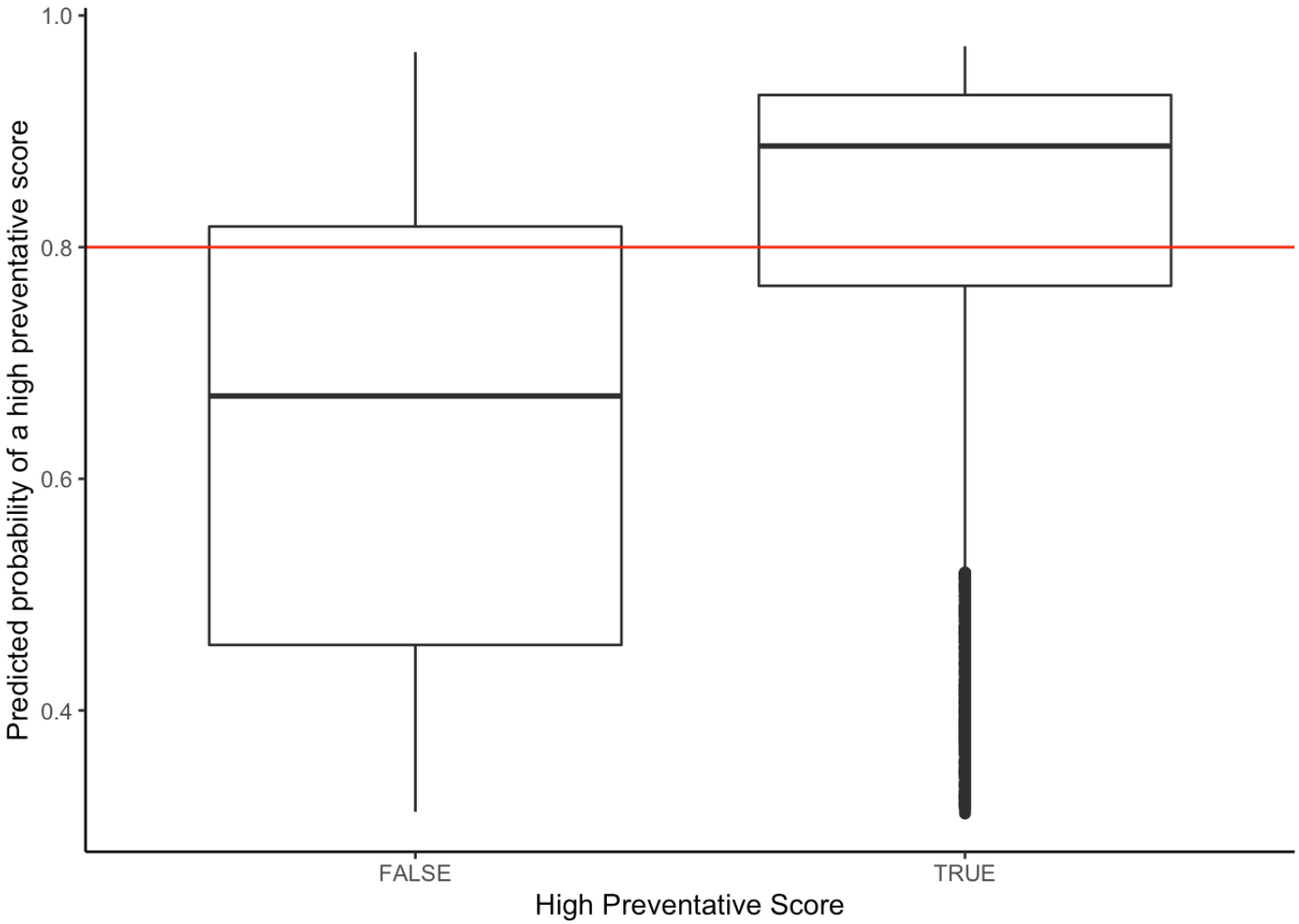
```
glm.mod.sub <- covid_cc %>%
  with(glm(highscore ~  age + gender + scared, family = binomial))
## make sure to filter missing values for all variables included in larger model

anova(glm.mod.full, glm.mod.sub, test='LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: highscore ~ age + gender + month + scared
## Model 2: highscore ~ age + gender + scared
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      9092     7787.2
## 2      9096     7929.2 -4  -141.97 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 9 (Select a Final Logistic Regression Model)

Using the tools available to you (hypothesis testing, causal diagrams, predicted probability boxplots, false positive and false negative rates, accuracy), fit a variety of models and choose one final model. Be systematic in your process as you'll need to describe your model selection process and justify your final model. (Note: for mastery of the Inference > Model Selection objective (see Final Grading Rubric), you must use at least 3 of these model selection tools.)

```
glm.mod.state <- covid %>% with(glm(highscore ~ age + gender + scared + state.region,
family = binomial))

glm.mod.full %>%
    augment(type.predict = "response") %>%
    ggplot(aes(y = .fitted, x = factor(highscore))) +
    geom_boxplot() +
    geom_hline(yintercept = 0.8, color = "red") +
    ylab("Predicted probability of a high preventative score") +
    xlab("High Preventative Score") +
    theme_classic()
```



```
threshold <- 0.8
augment(glm.mod.full, type.predict ='response') %>%
  mutate(predictScore = .fitted > threshold) %>%
  count(highscore, predictScore) %>%
  mutate(correct = predictScore == (highscore == 1)) %>%
  group_by(highscore) %>%
  mutate(relfreq= n/sum(n))
```

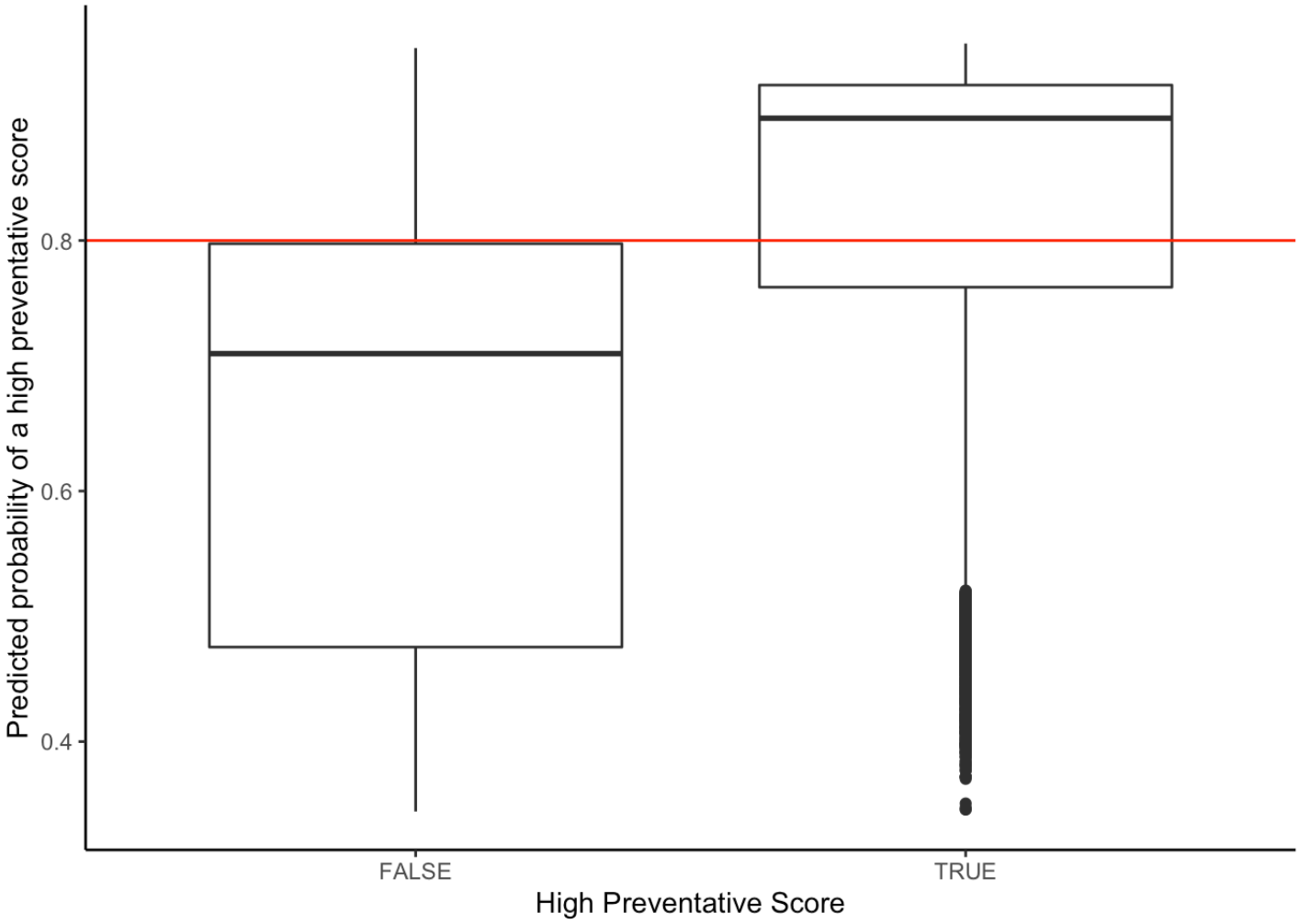
```
## # A tibble: 4 x 5
## # Groups:   highscore [2]
##   highscore predictScore     n correct relfreq
##   <lgl>     <lgl>        <int> <lgl>     <dbl>
## 1 FALSE     FALSE         1366 TRUE      0.713
## 2 FALSE     TRUE           549 FALSE     0.287
## 3 TRUE      FALSE         2018 FALSE     0.281
## 4 TRUE      TRUE          5169 TRUE      0.719
```

```
meanfull <- glm.mod.full %>%
  augment(type.predict = "response")
meanfull  %>%
  group_by(highscore) %>%
  summarize(mean_probability =  mean(.fitted, na.rm = TRUE),
            median_probability = median(.fitted, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   highscore mean_probability median_probability
##   <lgl>                <dbl>              <dbl>
## 1 FALSE                0.644              0.671
## 2 TRUE                 0.828              0.887
```

```
glm.mod.state %>%
    augment(type.predict = "response") %>%
    ggplot(aes(y = .fitted, x = factor(highscore))) +
    geom_boxplot() +
    geom_hline(yintercept = 0.8, color = "red") +
    ylab("Predicted probability of a high preventative score") +
    xlab("High Preventative Score") +
    theme_classic()
```




```
threshold <- 0.8
augment(glm.mod.state, type.predict ='response') %>%
  mutate(predictScore = .fitted > threshold) %>%
  count(highscore, predictScore) %>%
  mutate(correct = predictScore == (highscore == 1)) %>%
  group_by(highscore) %>%
  mutate(relfreq= n/sum(n))
```

```
## # A tibble: 4 x 5
## # Groups:   highscore [2]
##   highscore predictScore     n correct relfreq
##   <lgl>     <lgl>        <int> <lgl>     <dbl>
## 1 FALSE     FALSE         1442 TRUE      0.755
## 2 FALSE     TRUE           469 FALSE     0.245
## 3 TRUE      FALSE         2422 FALSE     0.339
## 4 TRUE      TRUE          4730 TRUE      0.661
```

```
meanstate <- glm.mod.state %>%
  augment(type.predict = "response")
meanstate  %>%
  group_by(highscore) %>%
  summarize(mean_probability =  mean(.fitted, na.rm = TRUE),
            median_probability = median(.fitted, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   highscore mean_probability median_probability
##   <lgl>                <dbl>              <dbl>
## 1 FALSE                0.657              0.710
## 2 TRUE                 0.825              0.898
```

```
final.glm.mod <- glm.mod.full#REPLACE THIS WITH CODE to fit final model

coef(final.glm.mod) %>% exp() #you'll need these estimates
```

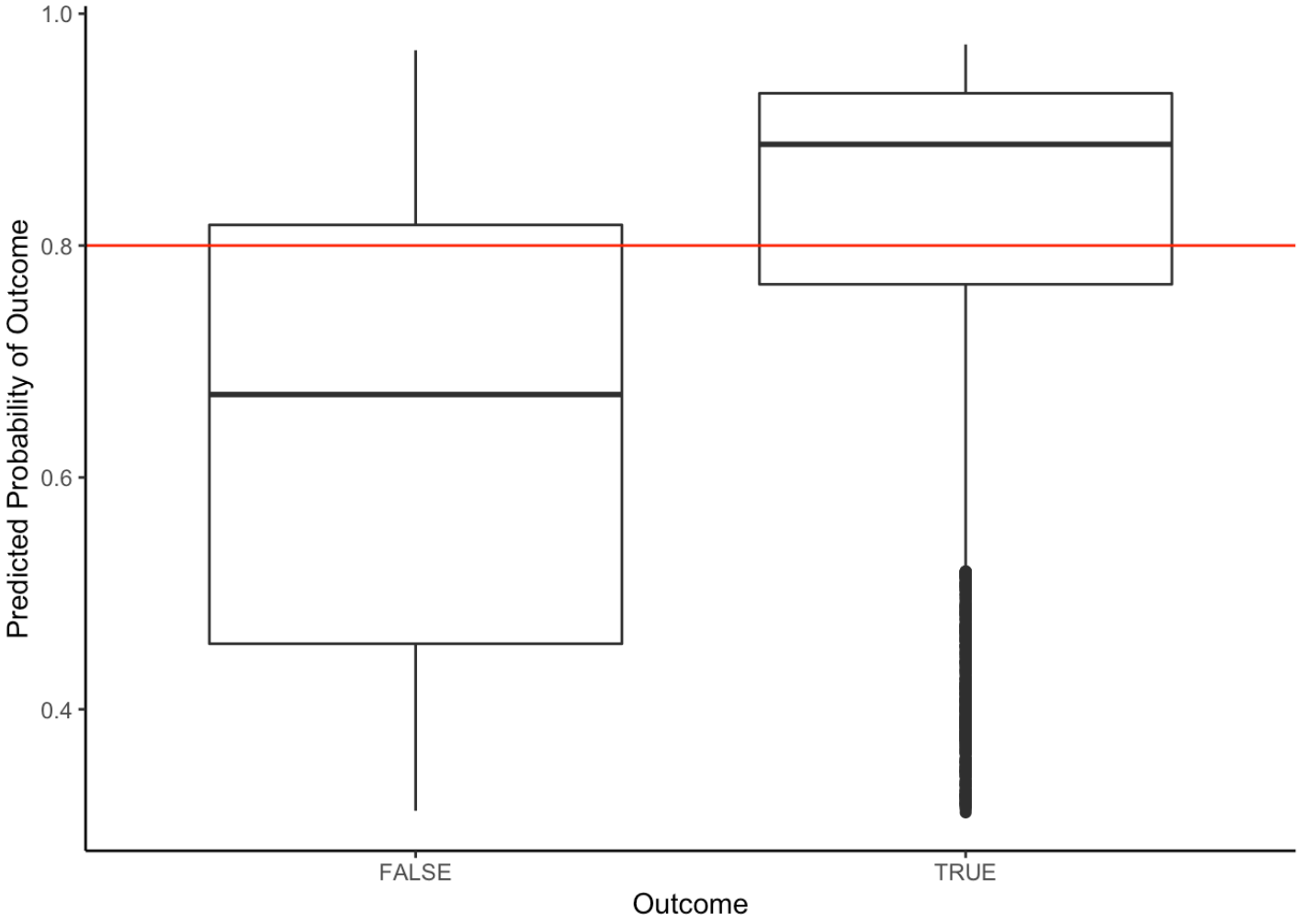
```
##    (Intercept)            age     genderMale         month5         month6
##      1.3613927      1.0065589      0.7394260      0.7379073      0.4139991
##         month7         month8 scaredNot very   scaredFairly     scaredVery
##      0.4062930      0.3884315      3.6011203     11.7088863     16.3869376
```

```
confint(final.glm.mod) %>% exp() #you'll need these confidence intervals
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %     97.5 %
## (Intercept)     1.0388802  1.7898811
## age             1.0033033  1.0098313
## genderMale      0.6610825  0.8269800
## month5          0.5920653  0.9147250
## month6          0.3286413  0.5189720
## month7          0.3227447  0.5088972
## month8          0.3022092  0.4974812
## scaredNot very  3.1539366  4.1149317
## scaredFairly    9.9926404 13.7573778
## scaredVery     13.2270226 20.4726222
```

```
augment(final.glm.mod, type.predict = 'response') %>%
  ggplot(aes(x = factor(highscore), y = .fitted)) + #replace ... with outcome variabl
e name
  geom_boxplot() +
  geom_hline(yintercept = 0.8, color = "red") +
  labs(x = 'Outcome', y = 'Predicted Probability of Outcome') +
  theme_classic()
```



Step 2 (Update Multiple Linear Regression Section)

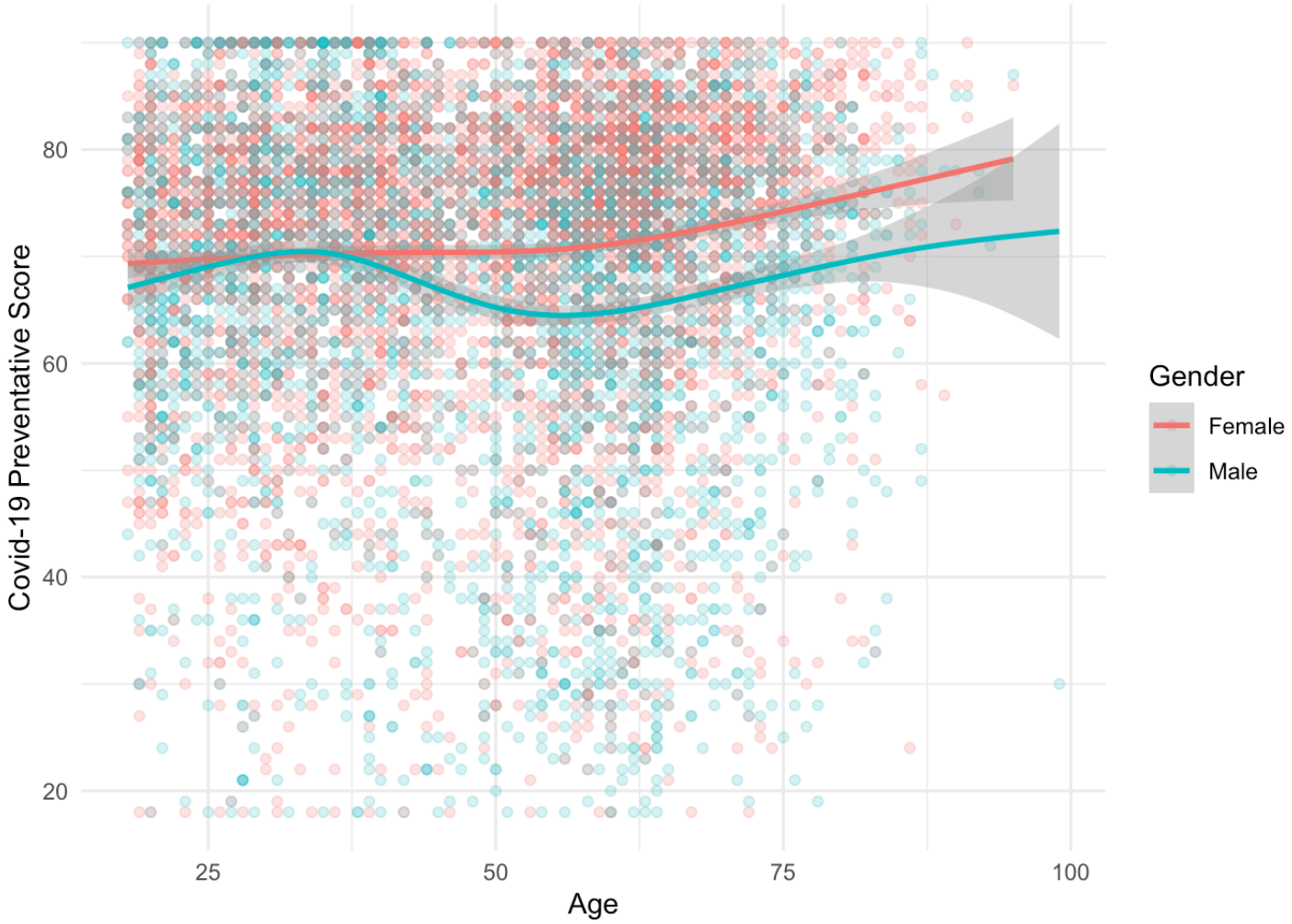
Visualization

Create a visualization that helps address your first research question involving a quantitative outcome. This visualization should include your outcome variable as well as the two explanatory variables that are most relevant to your research question. You do not need to (and should not) include all variables that are involved

in your final linear regression model in this visualization; just focus on the primary variables of interest. (If you feel that two visualizations would be more effective, that is ok too.)

```
covid %>%
  filter(!is.na(scared)) %>%
  ggplot(aes(x = age, y = score, color = gender)) +
  geom_point(alpha = 0.2)+
  geom_smooth()+
  labs(x = "Age", y = "Covid-19 Preventative Score", color = "Gender")+
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
# (and numerical summaries, if desired)

covid %>%
  filter(!is.na(scared)) %>%
  summarize(mean(score), median(score), sd(score), mean(age), median(age), sd(age), c
or(score, age))
```

```
##   mean(score) median(score) sd(score) mean(age) median(age) sd(age)
## 1    69.33751            73  15.09673  49.09372          51  17.308
##   cor(score, age)
## 1      0.01472353
```

```
covid %>%
  filter(!is.na(scared)) %>%
    group_by(gender) %>%
    summarize(
        median_score = median(score),
        mean_score = mean(score),
        sd_score = sd(score))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   gender median_score mean_score sd_score
##   <chr>         <dbl>      <dbl>    <dbl>
## 1 Female           75       71.1     14.1
## 2 Male             71       67.3     16.0
```

Save this visual and upload it (right click - copy and paste) to your **Final Report Google Doc**. Then, in a brief paragraph, thoroughly describe what information you gain from that visualization. You may use numerical summaries in your paragraph to fully describe your visualization.

Fitted Model

Use the code chunk below to print out the estimates, standard errors, p-values, and 95% confidence intervals for each of the coefficients in your final model.

```
#you should have fit final.lm.mod in Part 1

tidy(final.lm.mod) # estimates, standard errors, p-values
```

```
## # A tibble: 11 x 5
##    term           estimate std.error statistic   p.value
##    <chr>             <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     56.6       0.751      75.3  0.
##  2 age              0.0853    0.0109      7.85 4.71e- 15
##  3 genderMale       2.11      0.833       2.54 1.12e-  2
##  4 month5          -1.34      0.476      -2.82 4.77e-  3
##  5 month6          -4.79      0.523      -9.16 6.23e- 20
##  6 month7          -4.67      0.517      -9.05 1.77e- 19
##  7 month8          -5.58      0.582      -9.59 1.12e- 21
##  8 scaredNot very  11.0       0.408      26.9  1.92e-153
##  9 scaredFairly    16.8       0.401      41.9  0.
## 10 scaredVery      20.7       0.460      45.1  0.
## 11 age:genderMale  -0.0909    0.0160     -5.68 1.42e-  8
```

```
confint(final.lm.mod) # confidence intervals
```

```
##                       2.5 %     97.5 %
## (Intercept)    55.12669340 58.0722826
## age             0.06395889  0.1065481
## genderMale      0.48159581  3.7479229
## month5         -2.27438659 -0.4100922
## month6         -5.81675129 -3.7663479
## month7         -5.68634926 -3.6610754
## month8         -6.71754106 -4.4373858
## scaredNot very 10.18798791 11.7894191
## scaredFairly   16.02222482 17.5952429
## scaredVery     19.83274807 21.6365337
## age:genderMale -0.12228880 -0.0595109
```

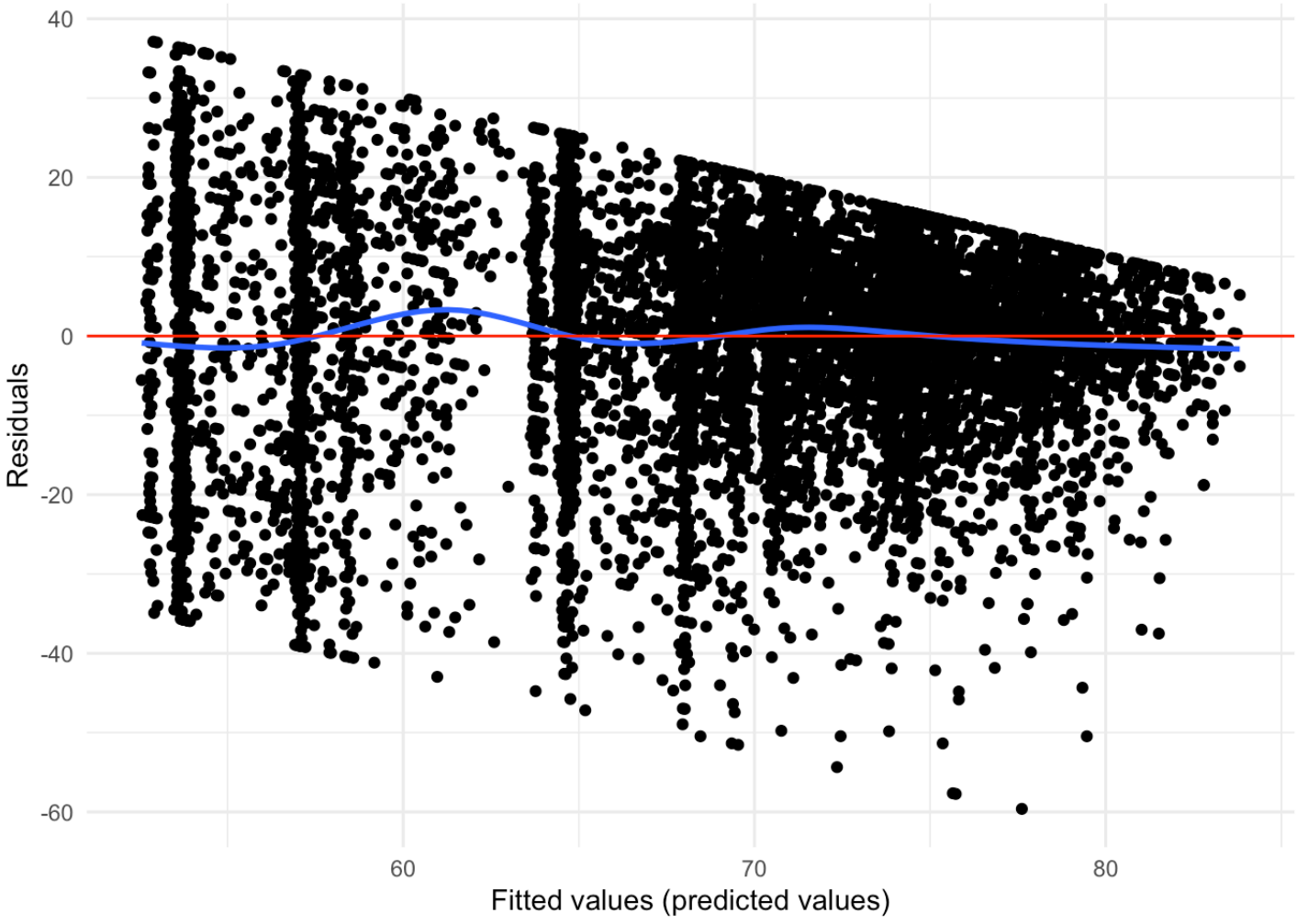
Then, add these estimates, standard errors, and confidence intervals to the table in the *Fitted Model* section of your Final Report Google Doc.

Model Evaluation

Use the code chunk below to check whether your final linear regression model meets all linear model conditions and to assess the “goodness” of your final model.

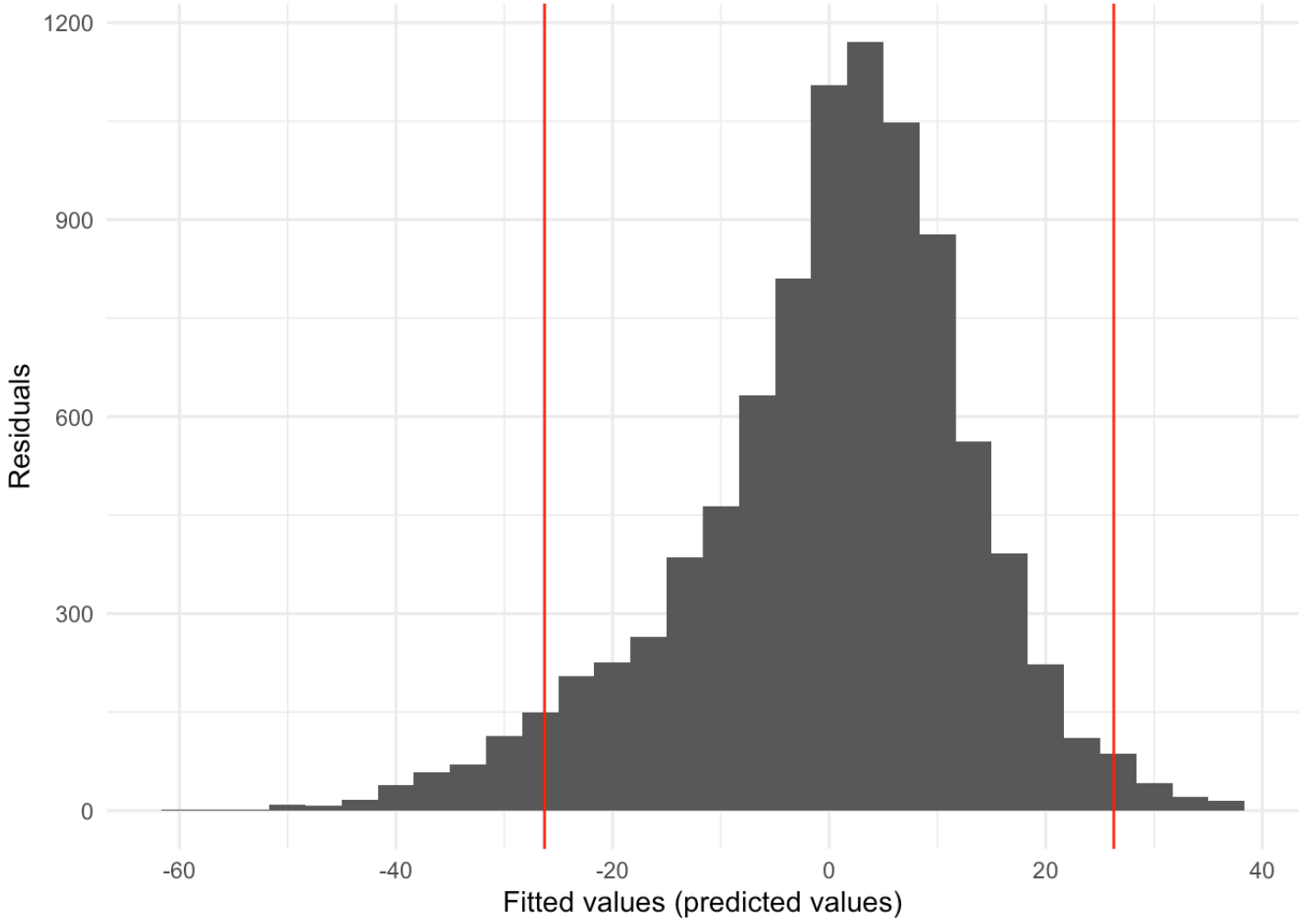
```
# REPLACE THIS WITH CODE to check conditions

augment(final.lm.mod, data = covid_cc) %>%
    ggplot(aes(x = .fitted, y = .resid)) +
    geom_point() +
    geom_smooth(se = FALSE) +
    geom_hline(yintercept = 0, color = "red") +
    labs(x = "Fitted values (predicted values)", y = "Residuals") +
    theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
augment(final.lm.mod, data = covid_cc) %>%
    ggplot(aes(x = .resid)) +
    geom_histogram() +
    geom_vline(xintercept = 26.28, color = "red") +
    geom_vline(xintercept = -26.28, color = "red") +
    labs(x = "Fitted values (predicted values)", y = "Residuals") +
    theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
glance(final.lm.mod) # to evaluate goodness
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.243         0.242  13.1      292.       0    10 -36354. 72731. 72817.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

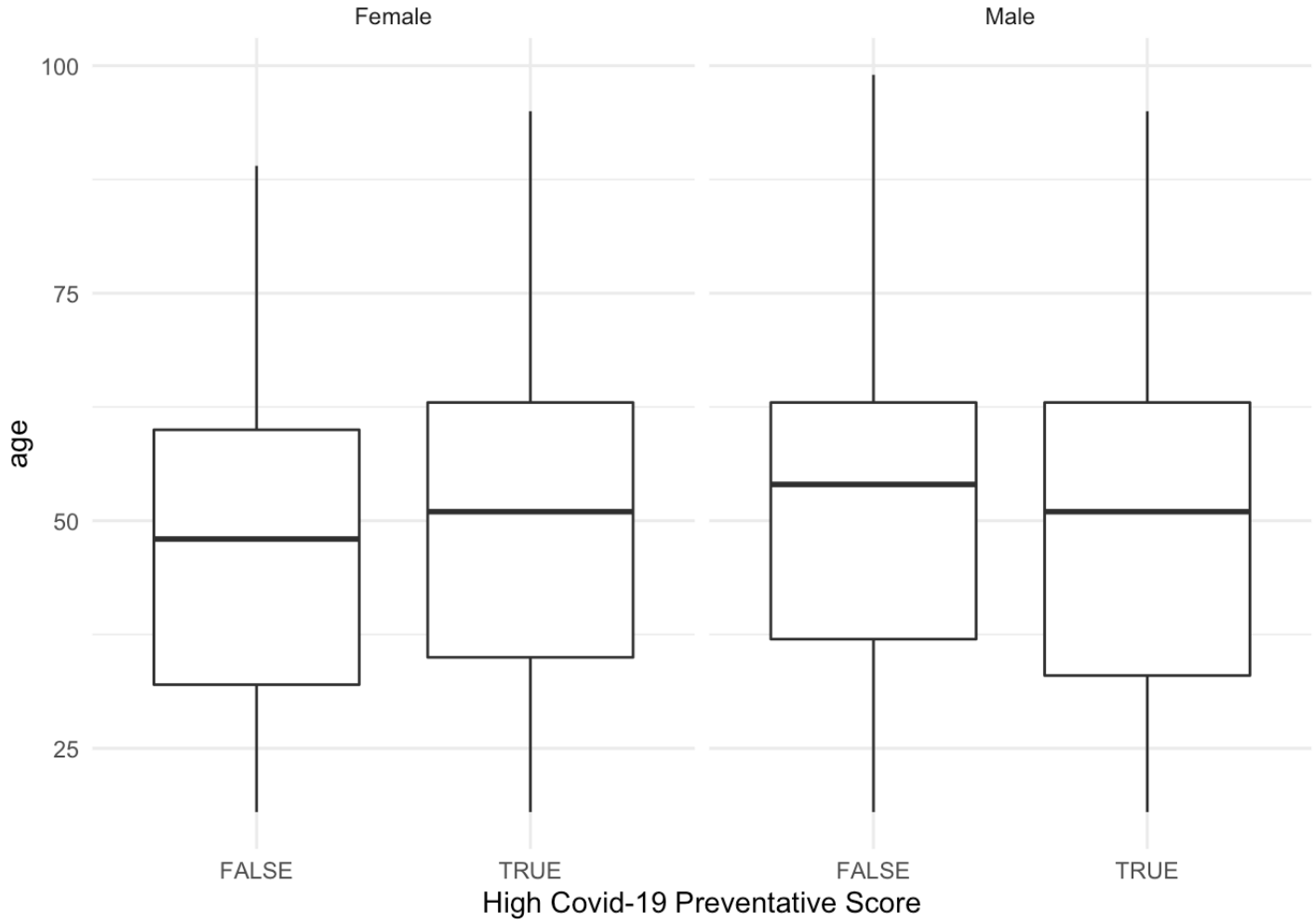
Add any graphical evidence and numerical evidence that you produced above to your Final Report Google Doc. Then, in paragraph form, describe what you've learned about model conditions (straight enough, equal spread, no extreme outliers) and goodness (R-squared, residual standard error, redundancy), putting your conclusions in context.

Step 3 (Update Multiple Logistic Regression Section)

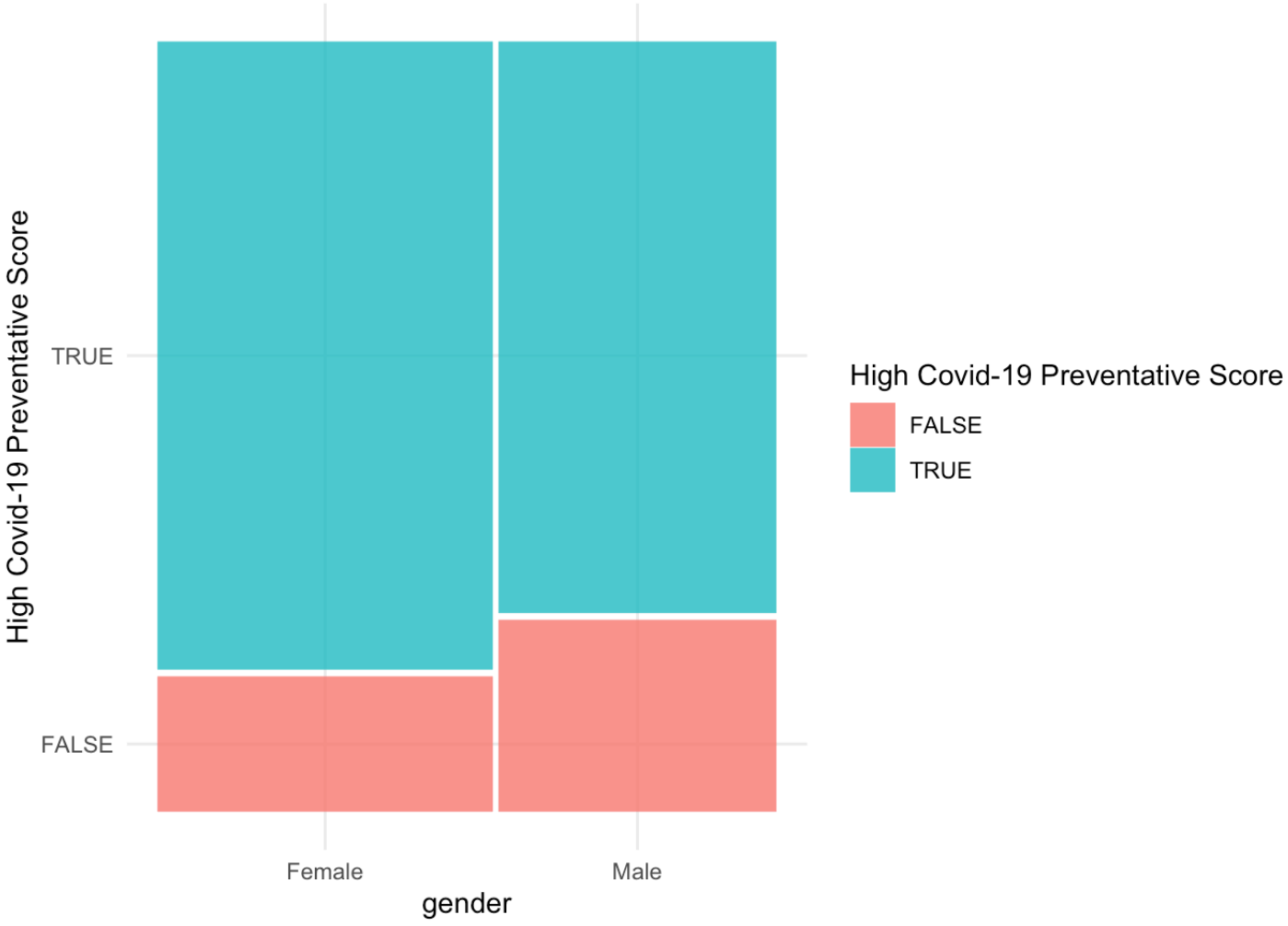
Visualization

Create a visualization that helps address your second research question involving a binary outcome. This visualization should include your outcome variable as well as the two explanatory variables that are most relevant to your research question. As above, you do not need to (and should not) include all variables that are involved in your final logistic regression model in this visualization; just focus on the primary variables of interest. (If you feel that two visualizations would be more effective, that is ok too.)

```
# REPLACE THIS WITH CODE for a plot
covid %>%
  filter(!is.na(scared)) %>%
  mutate(highscore = factor(highscore)) %>%
  ggplot(aes(x = highscore, y = age)) +
  geom_boxplot() +
  facet_wrap(.~gender)+
  xlab("High Covid-19 Preventative Score") +
  theme_minimal()
```




```
covid %>%
  filter(!is.na(scared)) %>%
  mutate(highscore = factor(highscore)) %>%
  ggplot() +
  geom_mosaic(aes(x = product(highscore, gender), fill = highscore))+
  labs(y = "High Covid-19 Preventative Score", fill = "High Covid-19 Preventative Sco
re")+
  theme_minimal()
```




```
# (and numerical summaries, if desired)
```

Save this visual and upload it (right click – copy and paste) to your **Final Report Google Doc**. Then, in a brief paragraph, thoroughly describe what information you gain from that visualization. You may use numerical summaries in your paragraph to fully describe your visualization.

Fitted Model

Use the code chunk below to print out the exponentiated estimates, p-values, and 95% confidence intervals for each of the coefficients in your final model.

```
# should have fit final.glm.mod in Part 1

coef(final.glm.mod) %>% exp() # exp estimates
```

```
##    (Intercept)             age      genderMale          month5          month6
##      1.3613927       1.0065589       0.7394260       0.7379073       0.4139991
##          month7          month8 scaredNot very   scaredFairly     scaredVery
##      0.4062930       0.3884315       3.6011203      11.7088863     16.3869376
```

```
confint(final.glm.mod) %>% exp() # confidence intervals
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %     97.5 %
## (Intercept)     1.0388802  1.7898811
## age             1.0033033  1.0098313
## genderMale      0.6610825  0.8269800
## month5          0.5920653  0.9147250
## month6          0.3286413  0.5189720
## month7          0.3227447  0.5088972
## month8          0.3022092  0.4974812
## scaredNot very  3.1539366  4.1149317
## scaredFairly    9.9926404 13.7573778
## scaredVery     13.2270226 20.4726222
```

```
tidy(final.glm.mod) # p-values
```

```
## # A tibble: 10 x 5
##    term           estimate std.error statistic   p.value
##    <chr>             <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     0.309     0.139        2.22 2.62e-  2
##  2 age             0.00654   0.00165      3.95 7.75e-  5
##  3 genderMale     -0.302     0.0571      -5.29 1.25e-  7
##  4 month5         -0.304     0.111       -2.74 6.14e-  3
##  5 month6         -0.882     0.116       -7.57 3.72e- 14
##  6 month7         -0.901     0.116       -7.76 8.69e- 15
##  7 month8         -0.946     0.127       -7.44 1.00e- 13
##  8 scaredNot very  1.28      0.0678      18.9  1.51e- 79
##  9 scaredFairly    2.46      0.0815      30.2  5.53e-200
## 10 scaredVery      2.80      0.111       25.1  3.50e-139
```

```
levels(state.region)
```

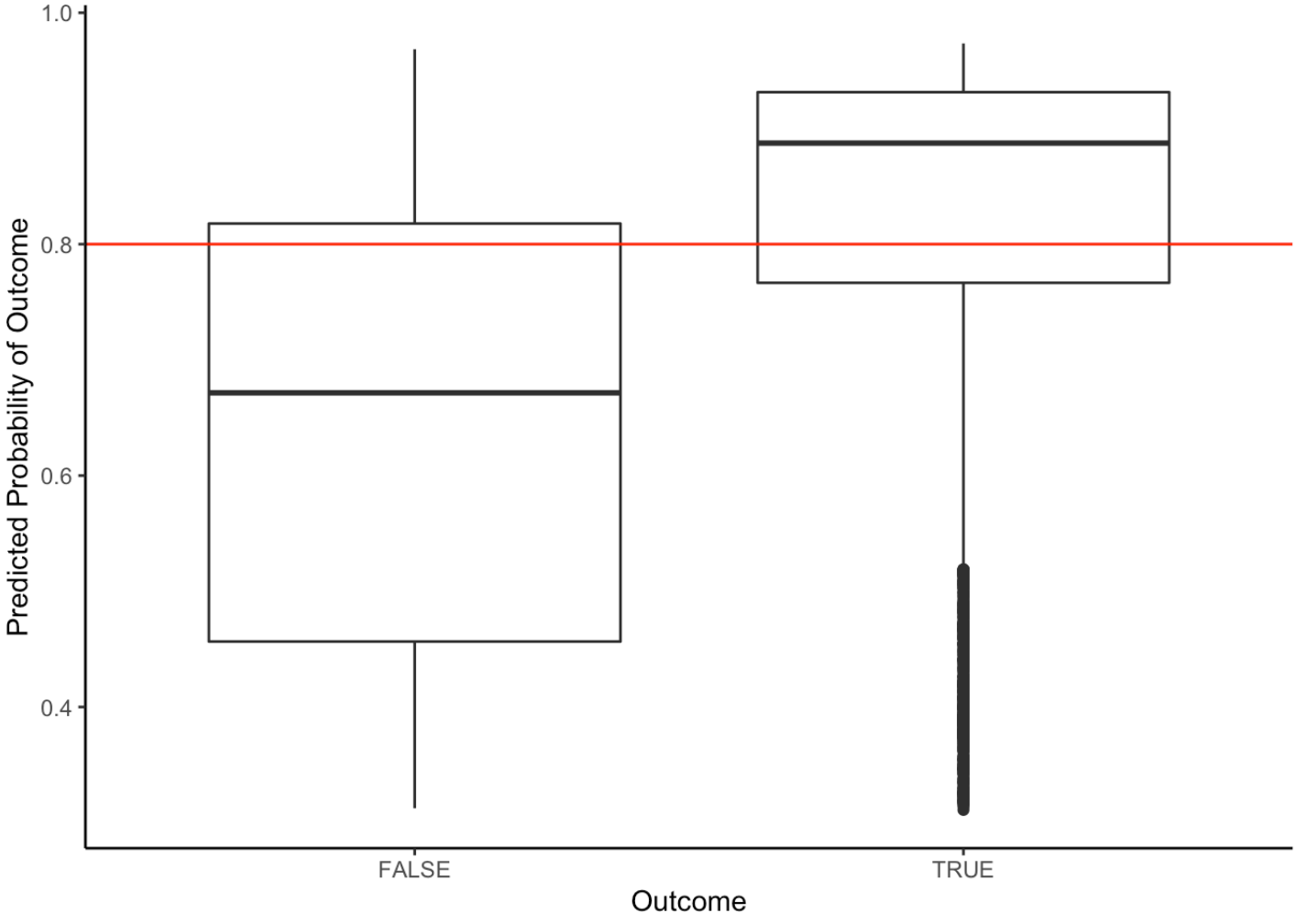
```
## [1] "Northeast"     "South"         "North Central" "West"
```

Then, add these estimates, standard errors, and confidence intervals to the table in the *Fitted Model* section of your Final Report Google Doc.

Model Evaluation

Use the code chunk below to assess the “goodness” of your final model.

```
augment(final.glm.mod, type.predict = 'response') %>%
  ggplot(aes(x = factor(highscore), y = .fitted)) + #replace ... with outcome variable name
  geom_boxplot() +
  geom_hline(yintercept = 0.8, color = "red")+
  labs(x = 'Outcome', y = 'Predicted Probability of Outcome') +
  theme_classic()
```



```
# evaluate goodness
threshold <- 0.8 # REPLACE with chosen threshold

augment(final.glm.mod, type.predict = 'response') %>%
  mutate(PredictOutcome = .fitted > threshold) %>%
  count(highscore, PredictOutcome) %>% #replace ... with outcome variable name
  group_by(highscore) %>% #replace ... with outcome variable name
  mutate(prop = n/sum(n))
```

```
## # A tibble: 4 x 4
## # Groups:   highscore [2]
##   highscore PredictOutcome     n  prop
##   <lgl>     <lgl>          <int> <dbl>
## 1 FALSE     FALSE           1366 0.713
## 2 FALSE     TRUE             549 0.287
## 3 TRUE      FALSE           2018 0.281
## 4 TRUE      TRUE            5169 0.719
```

Add any graphical evidence and numerical evidence that you produced above to your Final Report Google Doc. Then, in paragraph form, describe what you've learned about model goodness (accuracy, sensitivity, specificity, false positive rate, false negative rate), putting your conclusions in context.