Model Fitting Part 1

Joe Martin

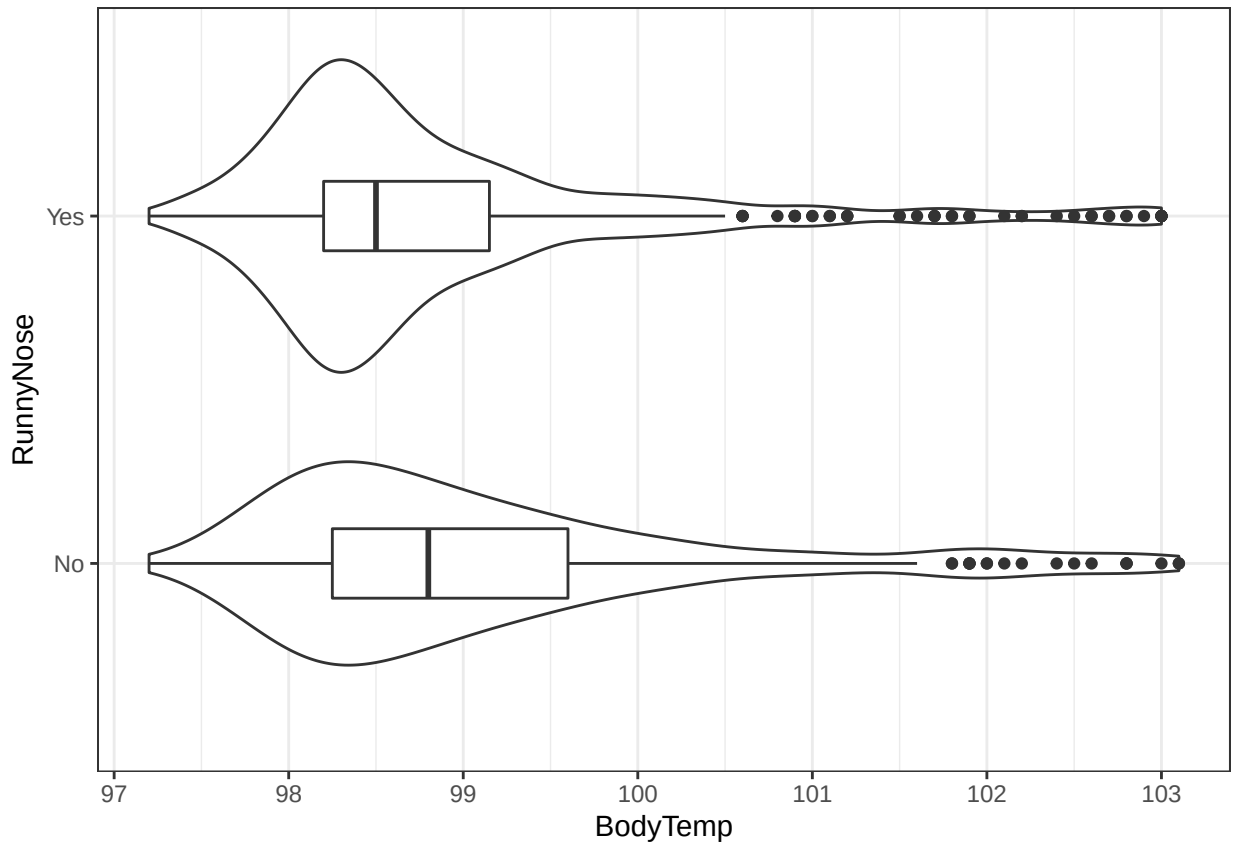
10/14/2021

Introduction

Presence of Runny Nose is the main predictor variable. Body Temperature is the main continuous variable. Nausea is the main categorical variable.

To begin modeling the RunnyNose variable, I'm adding the boxplot and regression read-out from my exploration:

```
rn_boxplot <- df %>% ggplot(aes(x=BodyTemp, y = RunnyNose))+
  geom_violin()+
  geom_boxplot(width = .2)+
  theme_bw()
rn_boxplot
```



p-value of .00268

```r
temp_sneeze <- lm(BodyTemp ~ RunnyNose, data = df)
summary(temp_sneeze)
```

```
##
## Call:
## lm(formula = BodyTemp ~ RunnyNose, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9431 -0.7505 -0.3505  0.3495  4.1495
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  99.14313    0.08191 1210.426  < 2e-16 ***
## RunnyNoseYes -0.29265    0.09714   -3.013  0.00268 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.19 on 728 degrees of freedom
## Multiple R-squared:  0.01231,    Adjusted R-squared:  0.01096
## F-statistic: 9.076 on 1 and 728 DF,  p-value: 0.00268
```

Models

Linear Regression

I'll begin running this as a linear regression. This is not the preferred regression to use here because the variable of interest is categorical. However, this will provide an approximate estimate of probability. The code below generates summary stats for this regression.

```r
# Fits a linear model to the continuous outcome using only the main predictor of interest.
lm_mod <- linear_reg() %>%
  set_engine("lm")

lm_fit <- lm_mod %>%
  fit(BodyTemp ~ RunnyNose, data = df)

lm_fit
```

```
## parsnip model object
##
## Fit time:  0ms
##
## Call:
## stats::lm(formula = BodyTemp ~ RunnyNose, data = data)
##
## Coefficients:
##  (Intercept)  RunnyNoseYes
##      99.1431       -0.2926
```

```
tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic p.value
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)    99.1      0.0819   1210.   0
## 2 RunnyNoseYes   -0.293    0.0971     -3.01 0.00268
```

Now I'll model all variables with BodyTemp. I can review these results and compare models for each variable with summary stats, as well as a dot-and-whisker plot.

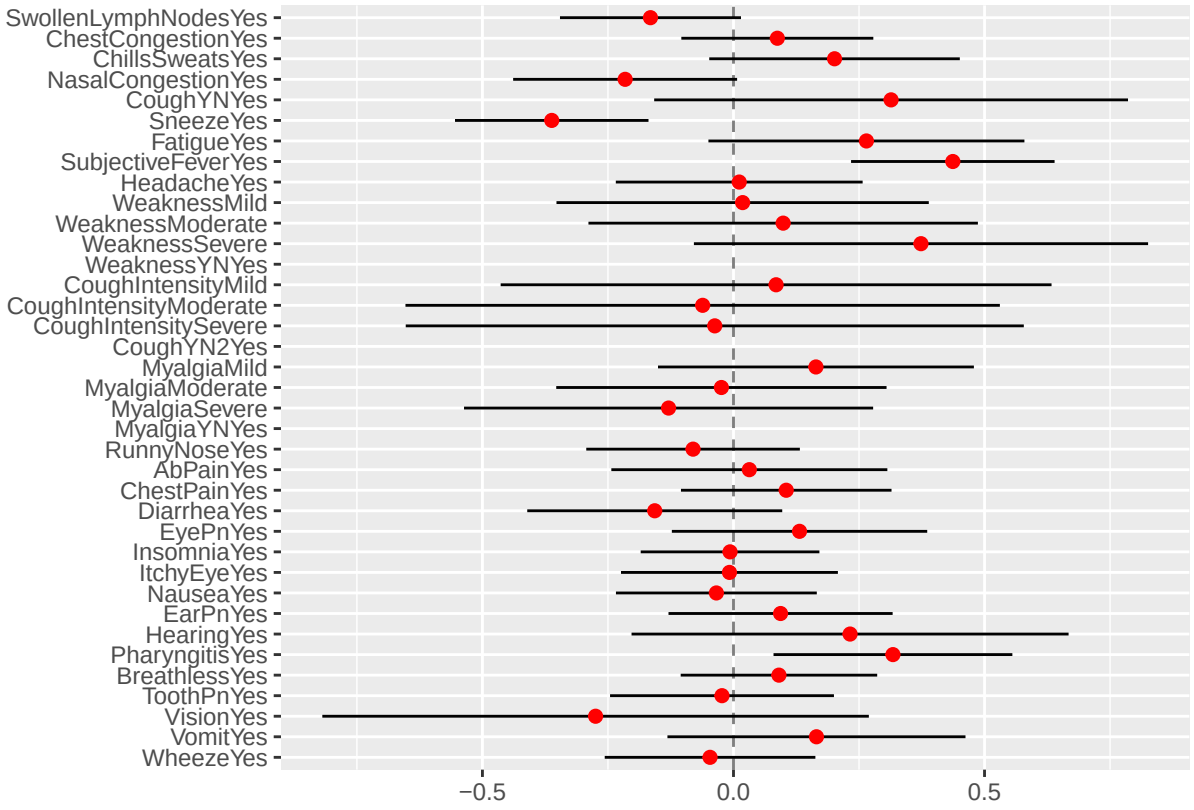
```
# Fits another linear model to the continuous outcome using all (important) predictors of interest.
lm_fit_more <-
  lm_mod %>%
  fit(BodyTemp ~ ., data = df)

lm_fit_more
```

```
## parsnip model object
##
## Fit time:  30ms
##
## Call:
## stats::lm(formula = BodyTemp ~ ., data = data)
##
## Coefficients:
##            (Intercept)    SwollenLymphNodesYes      ChestCongestionYes
##              97.925243               -0.165302                0.087326
##        ChillsSweatsYes      NasalCongestionYes             CoughYNYes
##                0.201266               -0.215771                0.313893
##              SneezeYes             FatigueYes     SubjectiveFeverYes
##              -0.361924                0.264762                0.436837
##            HeadacheYes           WeaknessMild       WeaknessModerate
##                0.011453                0.018229                0.098944
##         WeaknessSevere          WeaknessYNYes     CoughIntensityMild
##                0.373435                      NA                0.084881
## CoughIntensityModerate    CoughIntensitySevere            CoughYN2Yes
##               -0.061384               -0.037272                     NA
##            MyalgiaMild        MyalgiaModerate          MyalgiaSevere
##                0.164242               -0.024064               -0.129263
##           MyalgiaYNYes           RunnyNoseYes              AbPainYes
##                     NA               -0.080485                0.031574
##           ChestPainYes            DiarrheaYes               EyePnYes
##                0.105071               -0.156806                0.131544
##            InsomniaYes            ItchyEyeYes              NauseaYes
##               -0.006824               -0.008016               -0.034066
##              EarPnYes              HearingYes         PharyngitisYes
##                0.093790                0.232203                0.317581
##          BreathlessYes             ToothPnYes              VisionYes
##                0.090526               -0.022876               -0.274625
##              VomitYes              WheezeYes
##                0.165272               -0.046665
```

```
tidy(lm_fit_more)
```

```
## # A tibble: 38 x 5
##    term                 estimate std.error statistic   p.value
##    <chr>                   <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)           97.9       0.304   322.      0
##  2 SwollenLymphNodesYes  -0.165     0.0920   -1.80    0.0727
##  3 ChestCongestionYes     0.0873    0.0975    0.895   0.371
##  4 ChillsSweatsYes        0.201     0.127     1.58    0.114
##  5 NasalCongestionYes    -0.216     0.114    -1.90    0.0584
##  6 CoughYNYes             0.314     0.241     1.30    0.193
##  7 SneezeYes             -0.362     0.0983   -3.68    0.000249
##  8 FatigueYes             0.265     0.161     1.65    0.0996
##  9 SubjectiveFeverYes     0.437     0.103     4.22    0.0000271
## 10 HeadacheYes            0.0115    0.125     0.0913  0.927
## # ... with 28 more rows
```

```
tidy(lm_fit_more) %>%
  dwplot(dot_args = list(size = 2, color = "red"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, color = "grey50", linetype = 2))
```



Next, use the glance function to compare the output between the target variable (RunnyNose) and all other variables. Comparing p-values, it is clear that using all variables is a more robust way to predict body temperature.

```
# Compares the model results for the model with just the main predictor and all predictors.
glance(lm_fit)
```

```
## # 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    0.0123        0.0110  1.19      9.08 0.00268     1 -1162. 2329. 2343.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance(lm_fit_more)
```

```
## # 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     0.129        0.0860  1.14      3.02 0.0000000420    34 -1116. 2304. 2469.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Logistic Regression

This logistic regression model will test the relationship between Nausea, the main outcome of interest, and Runny Nose, the predictor of interest.

```
# Fits a logistic model to the categorical outcome using only the main predictor of interest.
log_mod <- logistic_reg() %>%
  set_engine("glm")

log_fit <-
  log_mod %>%
  fit(Nausea ~ RunnyNose, data = df)
```

```
log_fit
```

```
## parsnip model object
##
## Fit time:  20ms
##
## Call:  stats::glm(formula = Nausea ~ RunnyNose, family = stats::binomial,
##     data = data)
##
## Coefficients:
##  (Intercept)  RunnyNoseYes
##     -0.65781       0.05018
##
## Degrees of Freedom: 729 Total (i.e. Null);  728 Residual
## Null Deviance:       944.7
## Residual Deviance: 944.6     AIC: 948.6
```

```
tidy(log_fit)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic    p.value
##   <chr>           <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)   -0.658      0.145    -4.53  0.00000589
## 2 RunnyNoseYes   0.0502     0.172     0.292 0.770
```

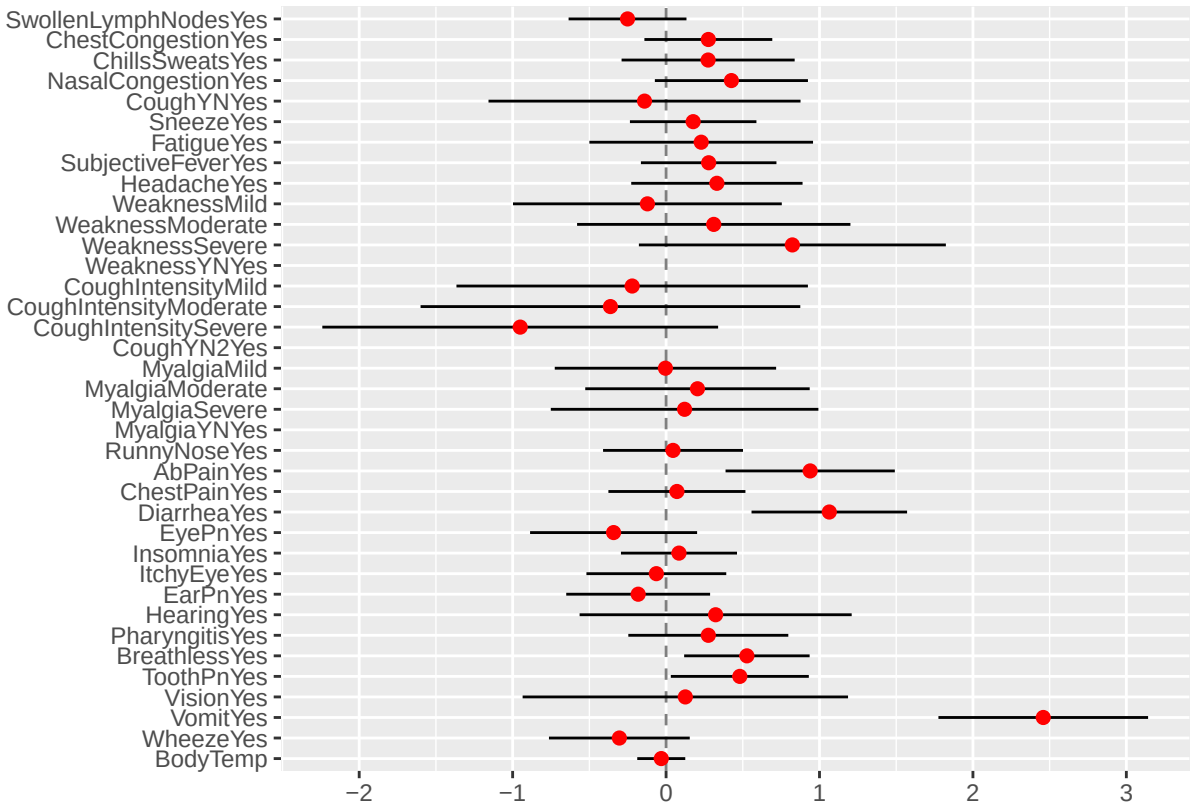
The p-value in this logistic regression is .77, so this isn't a significant relationship.

Finally, I'll compare the target variable, Nausea, with the other variables of interest.

```
# Fits another logistic model to the categorical outcome using all (important) predictors of interest.
log_fit_more <-
  log_mod %>%
  fit(Nausea ~ ., data = df)

tidy(log_fit_more)
```

```
## # A tibble: 38 x 5
##    term                 estimate std.error statistic p.value
##    <chr>                   <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept)             0.223     7.83     0.0285  0.977
##  2 SwollenLymphNodesYes   -0.251     0.196   -1.28    0.200
##  3 ChestCongestionYes      0.276     0.213    1.30    0.195
##  4 ChillsSweatsYes         0.274     0.288    0.952   0.341
##  5 NasalCongestionYes      0.426     0.255    1.67    0.0944
##  6 CoughYNYes             -0.140     0.519   -0.271   0.787
##  7 SneezeYes               0.177     0.210    0.840   0.401
##  8 FatigueYes              0.229     0.372    0.616   0.538
##  9 SubjectiveFeverYes      0.278     0.225    1.23    0.218
## 10 HeadacheYes             0.331     0.285    1.16    0.245
## # ... with 28 more rows
```

```
tidy(log_fit_more) %>%
  dwplot(dot_args = list(size = 2, color = "red"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, color = "grey50", linetype = 2))
```




```
glance(log_fit)
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1          945.     729  -472.  949.  958.     945.         728   730
```

```
glance(log_fit_more)
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1          945.     729  -376.  821.  982.     751.         695   730
```