Model Fitting Part 1

Joe Martin

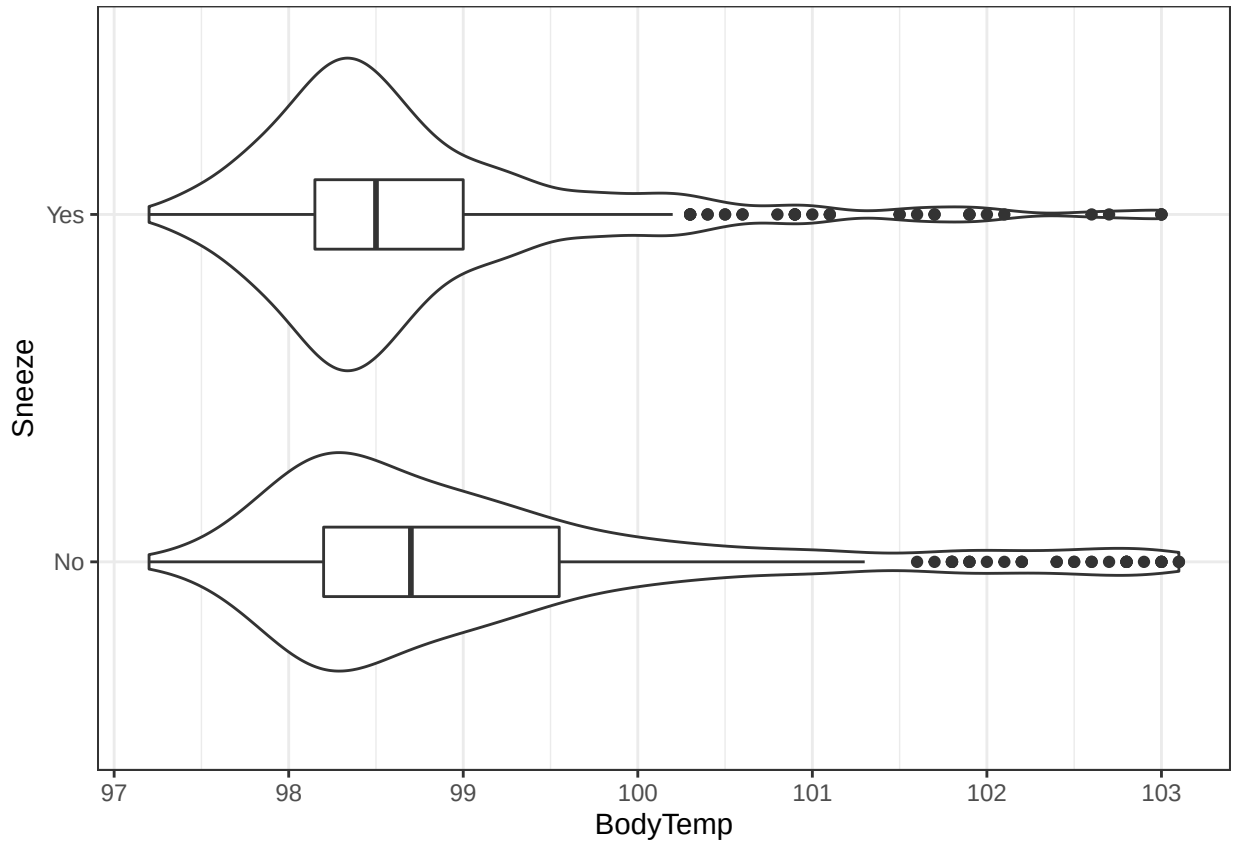
10/14/2021

Introduction

Presence of sneezing is the main predictor variable and has the strongest relationship to the continuous outcome variable, BodyTemp

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

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



p-value of .0000006037

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

```
##
## Call:
## lm(formula = BodyTemp ~ Sneeze, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9490 -0.7496 -0.3490  0.3504  4.2504
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 99.14897    0.06411 1546.478  < 2e-16 ***
## SneezeYes   -0.39935    0.08760   -4.559 6.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.18 on 728 degrees of freedom
## Multiple R-squared:  0.02775,    Adjusted R-squared:  0.02642
## F-statistic: 20.78 on 1 and 728 DF,  p-value: 6.037e-06
```

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.

```
# 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 ~ Sneeze, data = df)

lm_fit
```

```
## parsnip model object
##
## Fit time:  20ms
##
## Call:
## stats::lm(formula = BodyTemp ~ Sneeze, data = data)
##
## Coefficients:
## (Intercept)    SneezeYes
##     99.1490      -0.3994
```

```
tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)   99.1      0.0641   1546.   0
## 2 SneezeYes     -0.399    0.0876     -4.56 0.00000604
```

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.

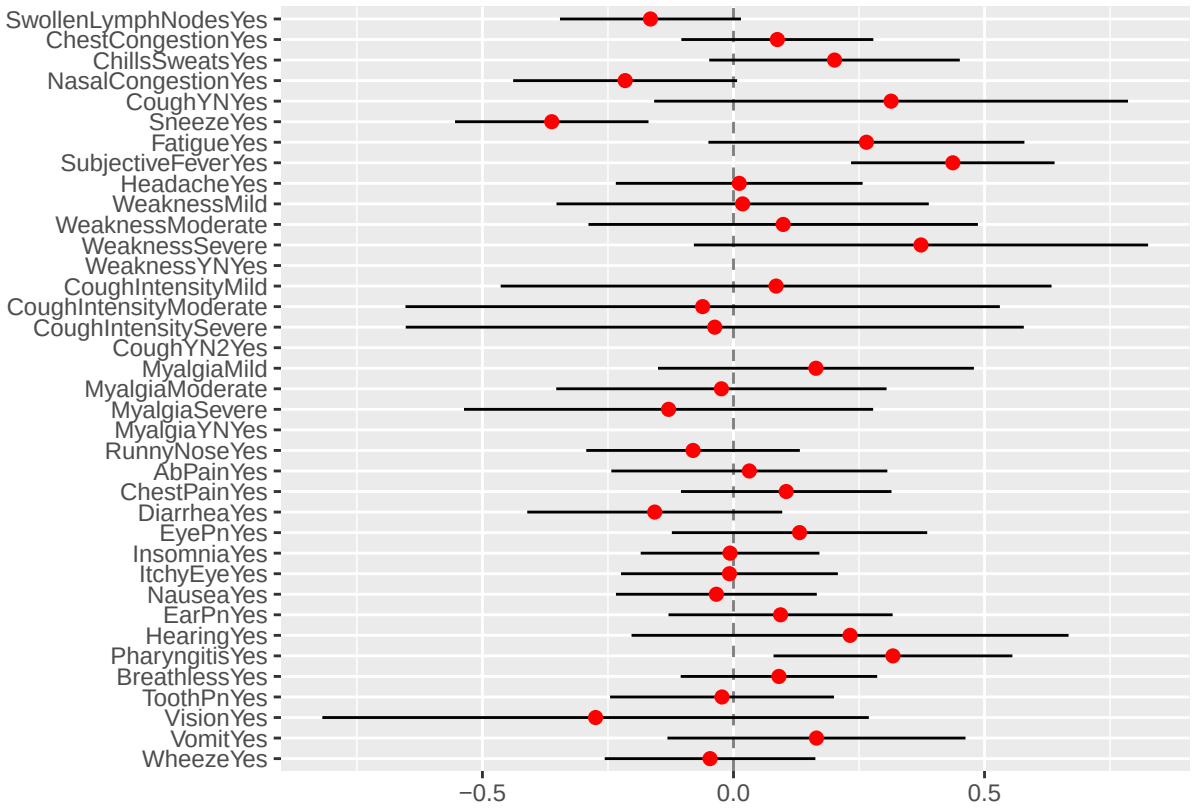
```
# Fits another linear model to the continuous outcome using all (important) predictors of interest.
lm_fit_more <-
  lm_mod %>%
  fit(BodyTemp ~ ., data = df)

lm_fit_more
```

```
## parsnip model object
##
## Fit time:  20ms
##
## 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, I'll use glance to view compare the output between the target variable (Sneeze) and the secondary variables I selected.

```r
# 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    0.0278        0.0264  1.18      20.8 0.00000604     1 -1156. 2318. 2332.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
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

A logistic model will likely be a better choice in this case because the Sneeze variable has values of Yes and No. I'll start by viewing summary statistics again. This time, the outcome is Sneeze (categorical) and the predictor of interest is BodyTemp (continuous).

```r
# 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(Sneeze ~ BodyTemp, data = df)
```

```r
log_fit
```

```
## parsnip model object
##
## Fit time:  20ms
##
## Call:  stats::glm(formula = Sneeze ~ BodyTemp, family = stats::binomial,
##     data = data)
##
## Coefficients:
## (Intercept)     BodyTemp
##     28.6780      -0.2884
##
## Degrees of Freedom: 729 Total (i.e. Null);  728 Residual
## Null Deviance:       1008
## Residual Deviance: 987.8     AIC: 991.8
```

```r
tidy(log_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   28.7      6.50       4.41 0.0000103
## 2 BodyTemp      -0.288    0.0657    -4.39 0.0000114
```

Finally, I'll compare the target variable, Sneeze, with the other variables of interest. I'll do this with a summary statistics table, as well as a dot-and-whisker plot.

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

log_fit_more
```

```
## parsnip model object
##
## Fit time:  51ms
##
## Call:  stats::glm(formula = Sneeze ~ ., family = stats::binomial, data = data)
##
## Coefficients:
##            (Intercept)    SwollenLymphNodesYes      ChestCongestionYes
##              27.789243               -0.009344               -0.069874
##        ChillsSweatsYes      NasalCongestionYes              CoughYNYes
##              -0.170831                0.919425               -0.352298
##             FatigueYes      SubjectiveFeverYes             HeadacheYes
##               0.816214               -0.097430               -0.041171
##           WeaknessMild       WeaknessModerate          WeaknessSevere
##              -0.717244               -0.186253                0.179545
##          WeaknessYNYes      CoughIntensityMild  CoughIntensityModerate
##                     NA                0.855573                0.918641
##   CoughIntensitySevere             CoughYN2Yes             MyalgiaMild
##               0.906606                      NA                0.079327
##        MyalgiaModerate          MyalgiaSevere            MyalgiaYNYes
##              -0.080003                0.068054                      NA
##           RunnyNoseYes               AbPainYes            ChestPainYes
##               1.736425                0.001257                0.229232
##            DiarrheaYes               EyePnYes             InsomniaYes
##               0.292650                0.041775                0.058551
##            ItchyEyeYes              NauseaYes                EarPnYes
##               0.834324                0.195506                0.376147
##             HearingYes         PharyngitisYes           BreathlessYes
##              -0.848845                0.114913               -0.269995
##             ToothPnYes              VisionYes                VomitYes
##               0.237309               -0.109887               -0.406766
##             WheezeYes                BodyTemp
##               0.507365               -0.312743
##
## Degrees of Freedom: 729 Total (i.e. Null);  695 Residual
## Null Deviance:       1008
## Residual Deviance: 784.5     AIC: 854.5
```

```
tidy(log_fit_more)
```

```
## # A tibble: 38 x 5
##    term                 estimate std.error statistic  p.value
##    <chr>                   <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)          27.8         7.97     3.49   0.000485
##  2 SwollenLymphNodesYes -0.00934     0.190   -0.0493 0.961
##  3 ChestCongestionYes   -0.0699      0.200   -0.349  0.727
##  4 ChillsSweatsYes      -0.171       0.268   -0.637  0.524
##  5 NasalCongestionYes    0.919       0.237    3.88   0.000103
##  6 CoughYNYes           -0.352       0.502   -0.701  0.483
##  7 FatigueYes            0.816       0.331    2.46   0.0138
##  8 SubjectiveFeverYes   -0.0974      0.216   -0.452  0.652
##  9 HeadacheYes          -0.0412      0.252   -0.163  0.870
## 10 WeaknessMild         -0.717       0.390   -1.84   0.0656
## # ... 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))
```