Multinomial Logistic Regression

The Basics

Dr. Maria Tackett

11.04.19

STA 210

[**Click for PDF of slides**]

Announcements

- Reading 11 for Monday
- HW 05 **due Wed, Nov 6 at 11:59p**
- Extra credit

HW 03 data analysis

Multinomial Logistic Regression

Generalized Linear Models (GLM)

- In practice, there are many different types of response variables including:
  - **Binary**: Win or Lose
  - **Nominal**: Democrat, Republican or Third Party candidate
  - **Ordered**: Movie rating (1 - 5 stars)
  - and others...
- These are all examples of **generalized linear models**, a broader class of models that generalize the multiple linear regression model
- See *Generalized Linear Models: A Unifying Theory* for more details about GLMs

Binary Response (Logistic)

- Given $P(y_i = 1|x_i) = p_i \quad \text{and} \quad P(y_i = 0|x_i) = 1 - p_i$

$$\log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_i$$

- We can calculate p_i by solving the logit equation:

$$p_i = \frac{\exp\{\beta_0 + \beta_1 x_i\}}{1 + \exp\{\beta_0 + \beta_1 x_i\}}$$

STA 210

Binary Response (Logistic)

- Suppose we consider $y = 0$ the *baseline category* such that

$$P(y_i = 0|x_i) = p_{i0} \text{ and } P(y_i = 1|x_i) = p_{i1}$$

- Then the logit model is

$$\log\left(\frac{p_{i1}}{p_{i0}}\right) = \beta_0 + \beta_1 x_i$$

- **Slope, β_1**: When x increases by one unit, the odds of $Y = 1$ versus the baseline $Y = 0$ are expected to multiply by a factor of $\exp\{\beta_1\}$

- **Intercept, β_0**: When $x = 0$, the odds of $y = 1$ versus the baseline $y = 0$ are expected to be $\exp\{\beta_0\}$

STA 210

Multinomial response variable

- Suppose the response variable y is categorical and can take values $1, 2, \dots, k$ such that $(k > 2)$
- **Multinomial Distribution:**

$$P(Y = 1) = p_1, P(Y = 2) = p_2, \dots, P(Y = k) = p_k$$

such that $\sum\limits_{j=1}^{k} p_j = 1$

STA 210

Multinomial Logistic Regression

- If we have an explanatory variable x, then we want $P(y = j) = p_j$ to be a function of x
- Choose a baseline category. Let's choose $y = 1$. Then,

$$\log\left(\frac{p_{ij}}{p_{i1}}\right) = \beta_{0j} + \beta_{1j}x_i$$

- In the multinomial logistc model, we have a separate equation for each category of the response relative to the baseline cateogry
  - If the response has k possible categories, there will be $k - 1$ equations as part of the multinomial logistic model

Multinomial Logistic Regression

- Suppose we have a response variable Y that can take three possible outcomes that are coded as "1", "2", "3"

- Let "1" be the baseline category. Then

$$\log\left(\frac{p_{i2}}{p_{i1}}\right) = \beta_{02} + \beta_{12}X_i$$

$$\log\left(\frac{p_{i3}}{p_{i1}}\right) = \beta_{03} + \beta_{13}X_i$$

STA 210

Multinomial Regression in R

- Use the `multinom()` function in the nnet package

```
library(nnet)
my.model <- multinom(Y ~ X1 + X2 + ... + XP, data=my.data)
tidy(my.model, exponentiate = FALSE) #display log-odds model
```

```
# calculate predicted probabilities
pred.probs <- predict(my.model, type = "probs")
```

NHANES Data

- National Health and Nutrition Examination Survey is conducted by the National Center for Health Statistics (NCHS)
- The goal is to *"assess the health and nutritional status of adults and children in the United States"*
- This survey includes an interview and a physical examination

NHANES Data

- We will use the data from the **`NHANES`** R package
- Contains 75 variables for the 2009 - 2010 and 2011 - 2012 sample years
- The data in this package is modified for educational purposes and should **not** be used for research
- Original data can be obtained from the NCHS website for research purposes
- Type **`?NHANES`** in console to see list of variables and definitions

NHANES: Health Rating vs. Age & Physical Activity

- **Question**: Can we use a person's age and whether they do regular physical activity to predict their self-reported health rating?
- We will analyze the following variables:
  - **`HealthGen`:** Self-reported rating of participant's health in general. Excellent, Vgood, Good, Fair, or Poor.
  - **`Age`:** Age at time of screening (in years). Participants 80 or older were recorded as 80.
  - **`PhysActive`:** Participant does moderate to vigorous-intensity sports, fitness or recreational activities

The data

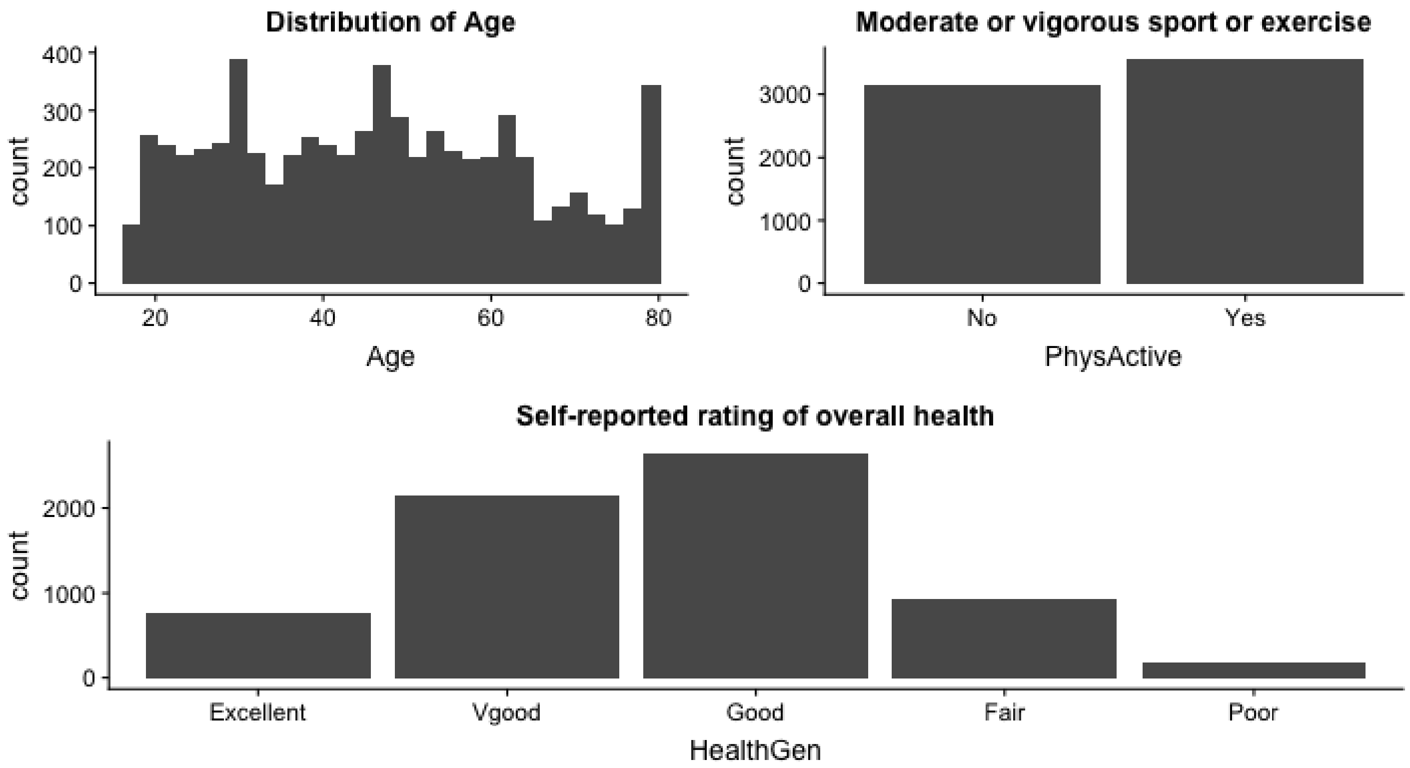
```
library(NHANES)

nhanes_adult <- NHANES %>%
  filter(Age >= 18) %>%
  select(HealthGen, Age, PhysActive) %>%
  drop_na() %>%
  mutate(obs_num = 1:n())
```

```
glimpse(nhanes_adult)
```

```
## Observations: 6,710
## Variables: 4
## $ HealthGen  <fct> Good, Good, Good, Good, Vgood, Vgood, Vgood, Vgood, V…
## $ Age        <int> 34, 34, 34, 49, 45, 45, 45, 66, 58, 54, 50, 33, 60, 5…
## $ PhysActive <fct> No, No, No, No, Yes, Yes, Yes, Yes, Yes, Yes, Yes, No…
## $ obs_num    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16…
```

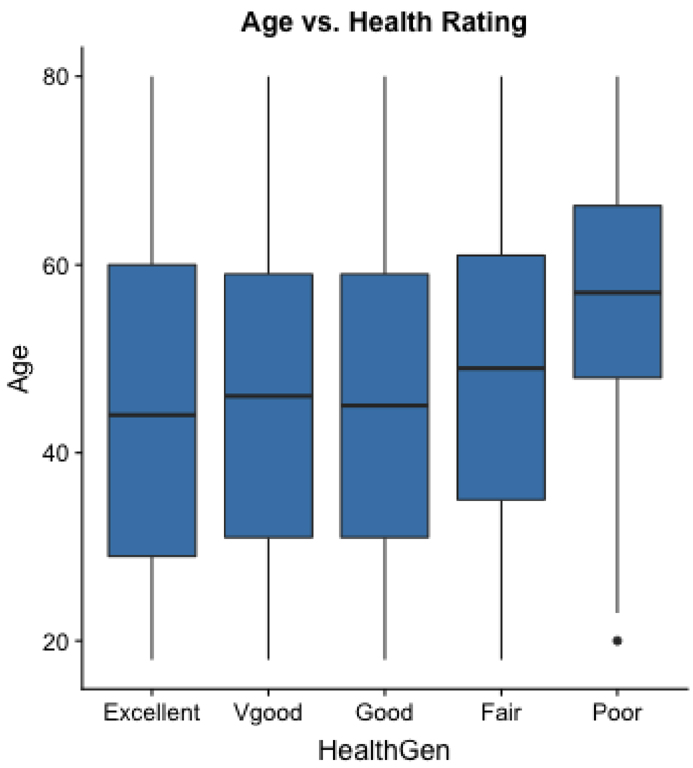
STA 210

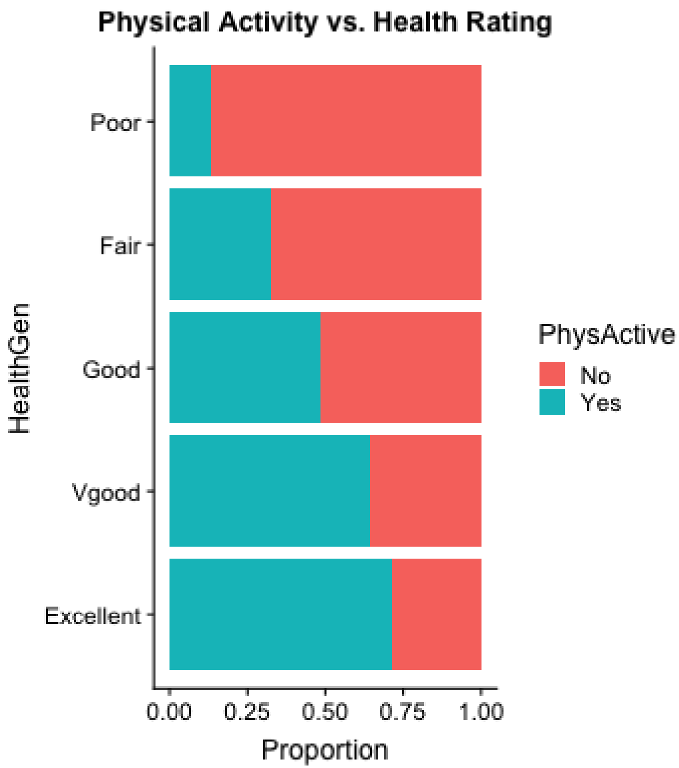
Exploratory data analysis



STA 210

Exploratory data analysis





`HealthGen` vs. `Age` and `PhysActive`

```r
library(nnet)
health_m <- multinom(HealthGen ~ Age + PhysActive,
                     data = nhanes_adult)
```

- Put `results = "hide"` in the code chunk header to suppress convergence output

STA 210

HealthGen vs. Age and PhysActive

```
tidy(health_m, exponentiate = FALSE, conf.int = TRUE) %>%
  kable(digits = 3, format = "markdown")
```

y.level	term	estimate	std.error	statistic	p.value	conf.low	conf.high
Vgood	(Intercept)	1.205	0.145	8.325	0.000	0.922	1.489
Vgood	Age	0.001	0.002	0.369	0.712	-0.004	0.006
Vgood	PhysActiveYes	-0.321	0.093	-3.454	0.001	-0.503	-0.139
Good	(Intercept)	1.948	0.141	13.844	0.000	1.672	2.223
Good	Age	-0.002	0.002	-0.977	0.329	-0.007	0.002
Good	PhysActiveYes	-1.001	0.090	-11.120	0.000	-1.178	-0.825
Fair	(Intercept)	0.915	0.164	5.566	0.000	0.592	1.237
Fair	Age	0.003	0.003	1.058	0.290	-0.003	0.009
Fair	PhysActiveYes	-1.645	0.107	-15.319	0.000	-1.856	-1.435
Poor	(Intercept)	-1.521	0.290	-5.238	0.000	-2.090	-0.952
Poor	Age	0.022	0.005	4.522	0.000	0.013	0.032
Poor	PhysActiveYes	-2.656	0.236	-11.275	0.000	-3.117	-2.194

Interpreting coefficients

1. What is the model baseline category?

2. Write the model for the odds that a person rates themselves as having "Fair" health versus the model baseline category.

3. Interpret the coefficient for `Age` in terms of the odds that a person rates themselves has having "Poor" heath versus the model's baseline category

Calculating probabilities

- For $j = 2, \dots, k$, we calculate the probabilities, p_{ij} as

$$p_{ij} = \frac{\exp\{\beta_{0j} + \beta_{1j}X_i\}}{1 + \sum\limits_{j=2}^{k} \exp\{\beta_{0j} + \beta_{1j}X_i\}}$$

- For the baseline category, $j = 1$, we calculate the probability p_{i1} as

$$p_{i1} = 1 - \sum_{j=2}^{k} p_{ij}$$

STA 210

Model assessment

For each category of the response, j:

- Analyze a plot of the binned residuals vs. predicted probabilities
- Analyze a plot of the binned residuals vs. each continuous predictor variable
- Look for any patterns in the residuals plots
- For each categorical predictor variable, examine the average residuals for each category of the response variable

NHANES: Predicted probabilities

```r
#calculate predicted probabilities
pred_probs <- as.tibble(predict(health_m, type = "probs")) %>%
                        mutate(obs_num = 1:n())
```

```r
pred_probs %>%
  slice(1:10)
```

```
## # A tibble: 10 x 6
##    Excellent Vgood  Good   Fair    Poor obs_num
##        <dbl> <dbl> <dbl>  <dbl>   <dbl>   <int>
##  1    0.0707 0.243 0.457 0.196  0.0328        1
##  2    0.0707 0.243 0.457 0.196  0.0328        2
##  3    0.0707 0.243 0.457 0.196  0.0328        3
##  4    0.0700 0.244 0.437 0.203  0.0454        4
##  5    0.155  0.392 0.360 0.0859 0.00648       5
##  6    0.155  0.392 0.360 0.0859 0.00648       6
##  7    0.155  0.392 0.360 0.0859 0.00648       7
##  8    0.156  0.400 0.343 0.0916 0.0103        8
##  9    0.156  0.397 0.349 0.0894 0.00865       9
## 10    0.156  0.396 0.353 0.0883 0.00791      10
```

STA 210

NHANES: Residuals

```r
#calculate residuals
residuals <- as.tibble(residuals(health_m)) %>%  #calculate residu
  setNames(paste('resid.', names(.), sep = "")) %>% #update colum
  mutate(obs_num = 1:n()) #add obs number
```

```r
residuals %>%
  slice(1:10)
```

```
## # A tibble: 10 x 6
##    resid.Excellent resid.Vgood resid.Good resid.Fair resid.Poor obs_num
##              <dbl>       <dbl>      <dbl>      <dbl>      <dbl>   <int>
##  1         -0.0707      -0.243      0.543    -0.196    -0.0328        1
##  2         -0.0707      -0.243      0.543    -0.196    -0.0328        2
##  3         -0.0707      -0.243      0.543    -0.196    -0.0328        3
##  4         -0.0700      -0.244      0.563    -0.203    -0.0454        4
##  5         -0.155        0.608     -0.360    -0.0859   -0.00648       5
##  6         -0.155        0.608     -0.360    -0.0859   -0.00648       6
##  7         -0.155        0.608     -0.360    -0.0859   -0.00648       7
##  8         -0.156        0.600     -0.343    -0.0916   -0.0103        8
##  9         -0.156        0.603     -0.349    -0.0894   -0.00865       9
## 10         -0.156       -0.396     -0.353     0.912    -0.00791      10
```

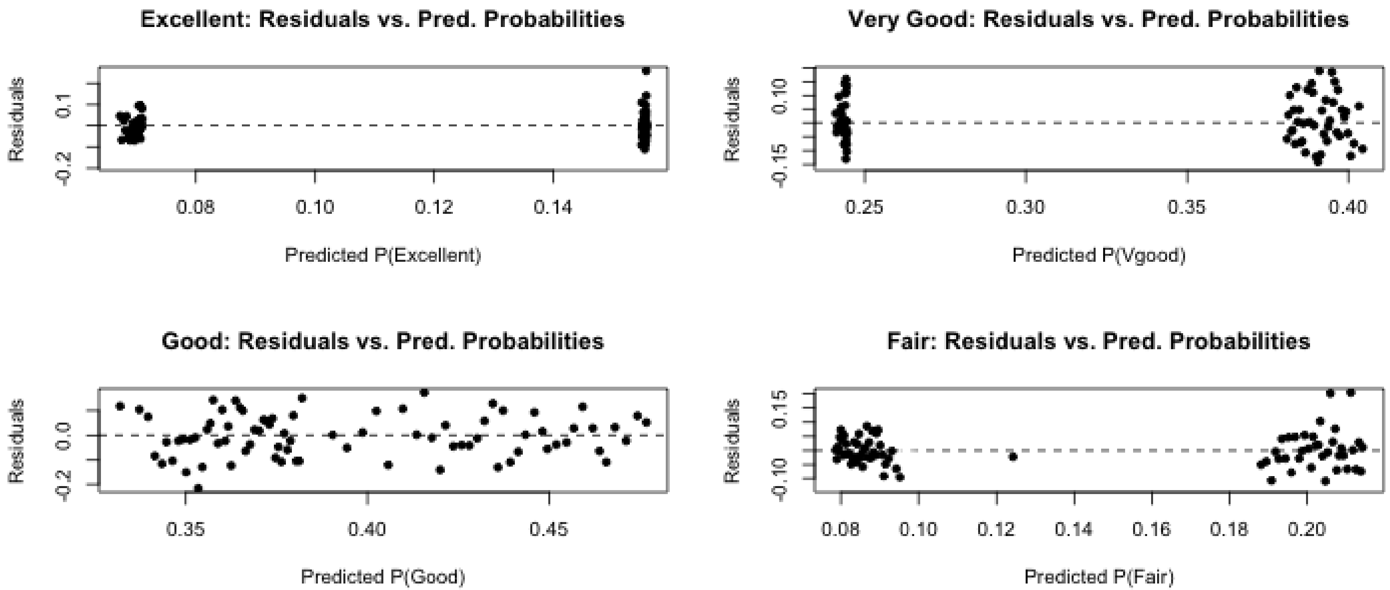
Make "augmented" dataset

```
health_m_aug <- inner_join(nhanes_adult, pred_probs) #add pred pro
health_m_aug <- inner_join(health_m_aug, residuals) #add residuals
```

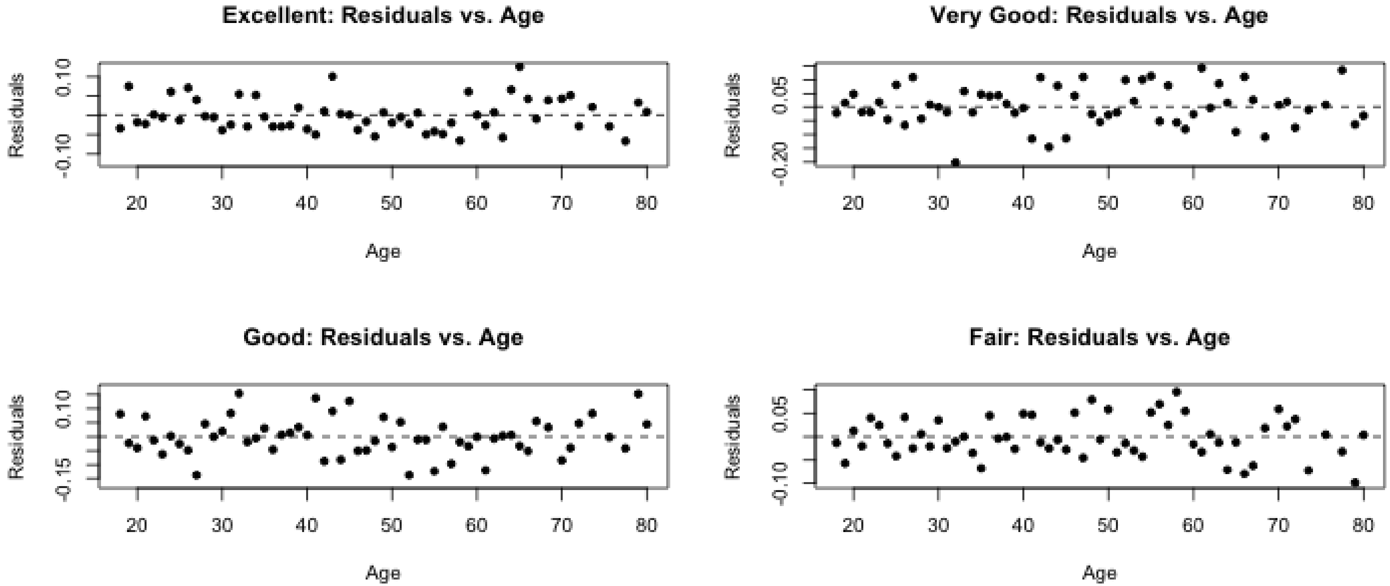
```
health_m_aug %>%
  glimpse()
```

```
## Observations: 6,710
## Variables: 14
## $ HealthGen        <fct> Good, Good, Good, Good, Vgood, Vgood, Vgood, Vgo
## $ Age              <int> 34, 34, 34, 49, 45, 45, 45, 66, 58, 54, 50, 33, .
## $ PhysActive       <fct> No, No, No, No, Yes, Yes, Yes, Yes, Yes, Yes, Ye
## $ obs_num          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## $ Excellent        <dbl> 0.07069715, 0.07069715, 0.07069715, 0.07003173, .
## $ Vgood            <dbl> 0.2433979, 0.2433979, 0.2433979, 0.2444214, 0.39
## $ Good             <dbl> 0.4573727, 0.4573727, 0.4573727, 0.4372533, 0.35
## $ Fair             <dbl> 0.19568909, 0.19568909, 0.19568909, 0.20291032, .
## $ Poor             <dbl> 0.032843150, 0.032843150, 0.032843150, 0.0453833
## $ resid.Excellent  <dbl> -0.07069715, -0.07069715, -0.07069715, -0.070031
## $ resid.Vgood      <dbl> -0.2433979, -0.2433979, -0.2433979, -0.2444214, .
## $ resid.Good       <dbl> 0.5426273, 0.5426273, 0.5426273, 0.5627467, -0.3
## $ resid.Fair       <dbl> -0.19568909, -0.19568909, -0.19568909, -0.202910
```

Binned residuals vs. pred. probabilities



Binned residuals vs. Age



Residuals vs. `PhysActive`

```
health_m_aug %>%
  group_by(PhysActive) %>%
  summarise(mean.Excellent = mean(resid.Excellent),
            mean.Vgood = mean(resid.Vgood),
            mean.Good = mean(resid.Good),
            mean.Fair = mean(resid.Fair),
            mean.Poor = mean(resid.Poor)) %>%
  t()
```

```
##                [,1]            [,2]
## PhysActive     "No"            "Yes"
## mean.Excellent "-1.194022e-07" " 2.106514e-06"
## mean.Vgood     " 1.644794e-06" "-1.871461e-06"
## mean.Good      "-3.227820e-06" " 1.140886e-07"
## mean.Fair      " 1.333924e-06" "-3.860756e-07"
## mean.Poor      "3.685045e-07"  "3.693412e-08"
```

STA 210

Actual vs. Predicted Health Rating

- We can use our model to predict a person's health rating given their age and whether they exercise
- For each observation, the predicted health rating is the one with the highest predicted probability

```
health_m_aug <-
  health_m_aug %>%
  mutate(pred_health = predict(health_m, type = "class"))
```

Actual vs. Predicted Health Rating

```
health_m_aug %>%
  count(HealthGen, pred_health, .drop = FALSE) %>%
  pivot_wider(names_from = pred_health, values_from = n)
```

```
## # A tibble: 5 x 6
##   HealthGen Excellent Vgood  Good  Fair  Poor
##   <fct>         <int> <int> <int> <int> <int>
## 1 Excellent         0   550   223     0     0
## 2 Vgood             0  1376   785     0     0
## 3 Good              0  1255  1399     0     0
## 4 Fair              0   300   642     0     0
## 5 Poor              0    24   156     0     0
```

```
#rows = actual, columns = predicted
```

STA 210

Predictions

```
## # A tibble: 5 x 6
##   Excellent Vgood  Good   Fair    Poor pred_health
##       <dbl> <dbl> <dbl>  <dbl>   <dbl> <fct>
## 1    0.0707 0.243 0.457 0.196  0.0328  Good
## 2    0.0707 0.243 0.457 0.196  0.0328  Good
## 3    0.0707 0.243 0.457 0.196  0.0328  Good
## 4    0.0700 0.244 0.437 0.203  0.0454  Good
## 5    0.155  0.392 0.360 0.0859 0.00648 Vgood
```

STA 210