Multinomial Logistic Regression

Predictions & Drop-in Deviance Test

Dr. Maria Tackett

11.04.19

STA 210

[Click for PDF of slides]

Announcements

- Multinomial Logistic Regression: Reading 10 and Reading 11
- HW 05 **due TODAY at 11:59p**

STA 210

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)

- 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 $\exp\{\beta_0\}$

Multinomial response variable

- Suppose the response variable y is categorical and can take values $1, 2, \ldots, k$ such that $(k > 2)$

- Multinomial Distribution:

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

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

STA 210

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

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

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

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")
```

STA 210

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

STA 210

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

STA 210

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

STA 210

The data

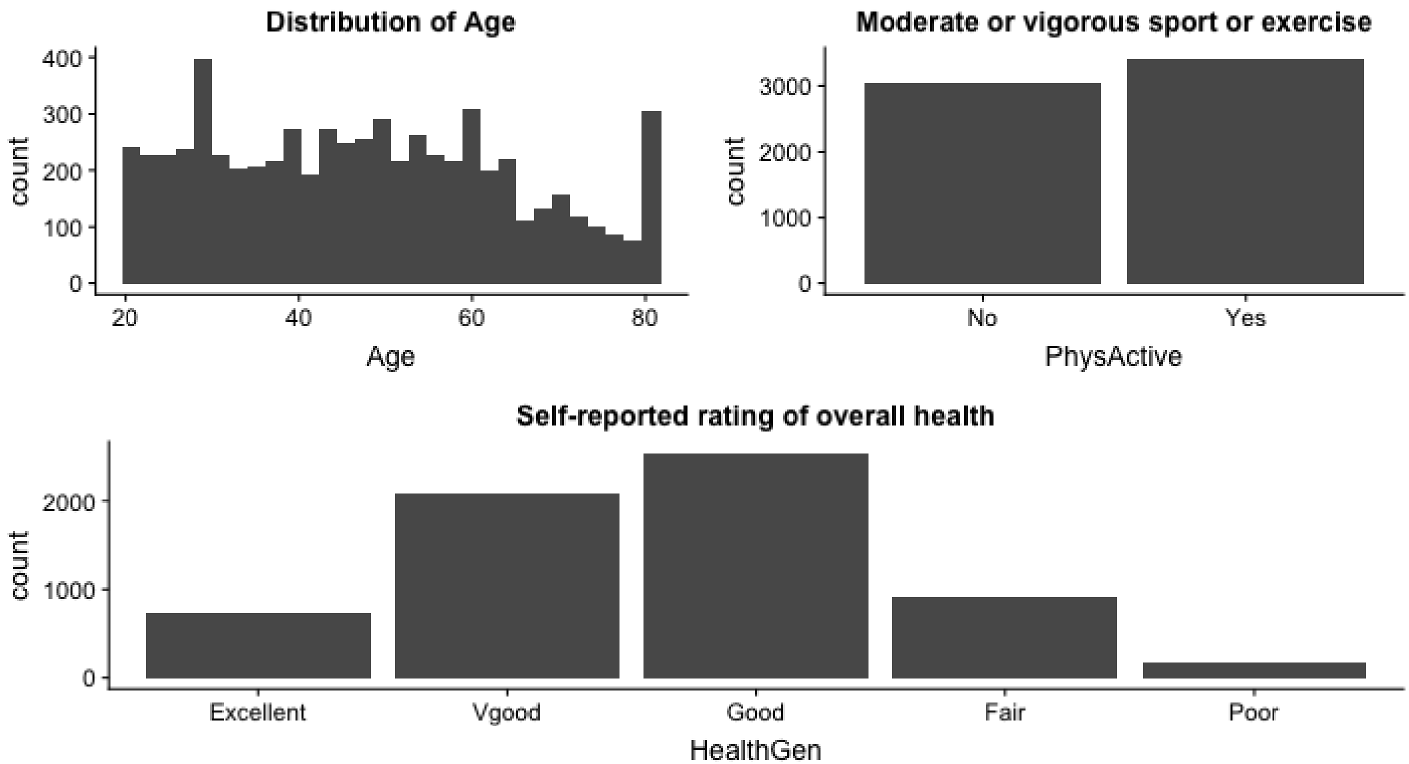
```r
library(NHANES)

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

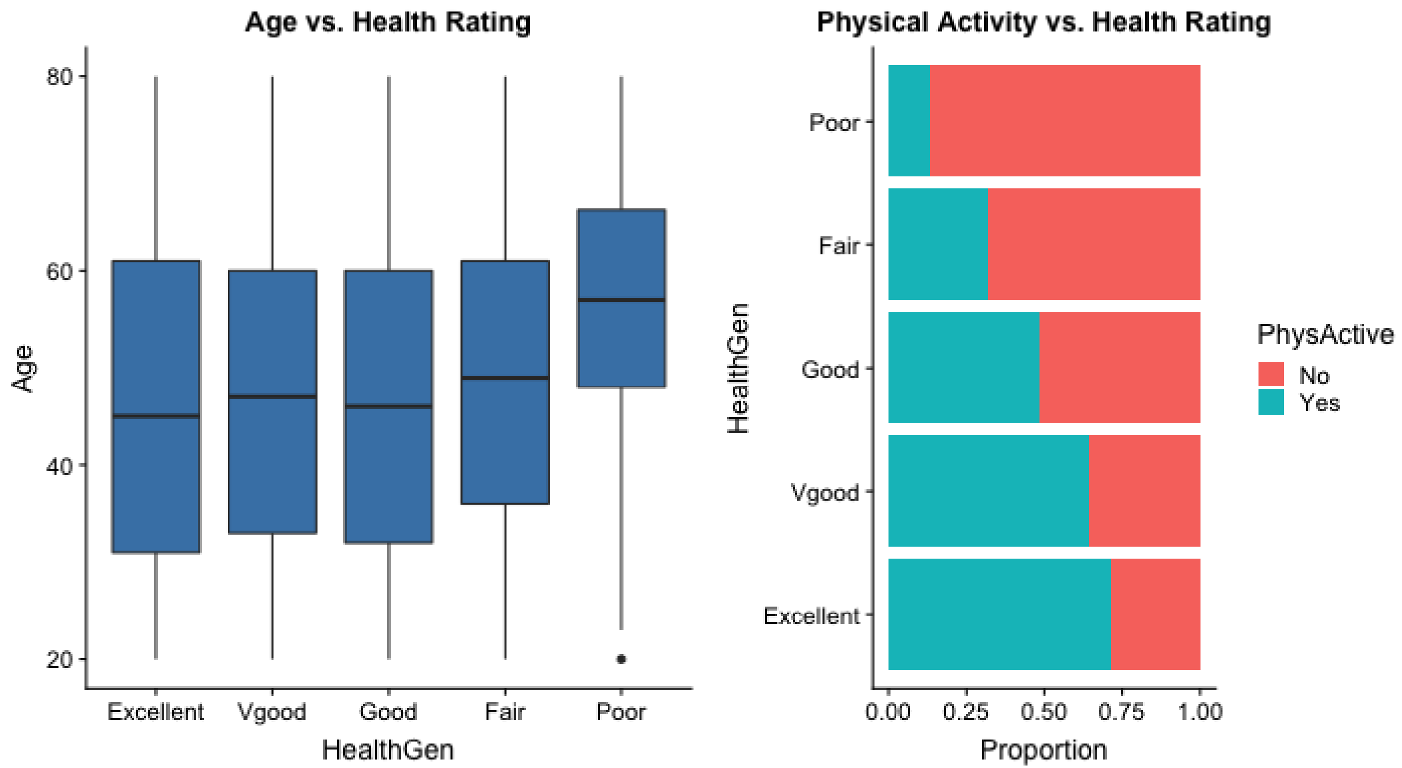
```r
glimpse(nhanes_adult)
```

```
## Observations: 6,465
## Variables: 5
## $ 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…
## $ Education  <fct> High School, High School, High School, Some College, …
## $ obs_num    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16…
```

Exploratory data analysis



Exploratory data analysis



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

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

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

HealthGen vs. Age and PhysActive

```r
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)	1.265	0.154	8.235	0.000	0.964	1.567
Vgood	Age	0.000	0.003	-0.014	0.989	-0.005	0.005
Vgood	PhysActiveYes	-0.332	0.095	-3.496	0.000	-0.518	-0.146
Good	(Intercept)	1.989	0.150	13.285	0.000	1.695	2.282
Good	Age	-0.003	0.003	-1.187	0.235	-0.008	0.002
Good	PhysActiveYes	-1.011	0.092	-10.979	0.000	-1.192	-0.831
Fair	(Intercept)	1.033	0.174	5.938	0.000	0.692	1.374
Fair	Age	0.001	0.003	0.373	0.709	-0.005	0.007
Fair	PhysActiveYes	-1.662	0.109	-15.190	0.000	-1.877	-1.448
Poor	(Intercept)	-1.338	0.299	-4.475	0.000	-1.924	-0.752
Poor	Age	0.019	0.005	3.827	0.000	0.009	0.029
Poor	PhysActiveYes	-2.670	0.236	-11.308	0.000	-3.133	-2.208

Interpreting coefficients

1. What is the model baseline category, i.e. the baseline category of the response variable?

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

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    0.0687 0.243 0.453 0.201  0.0348        1
##  2    0.0687 0.243 0.453 0.201  0.0348        2
##  3    0.0687 0.243 0.453 0.201  0.0348        3
##  4    0.0691 0.244 0.435 0.205  0.0467        4
##  5    0.155  0.393 0.359 0.0868 0.00671       5
##  6    0.155  0.393 0.359 0.0868 0.00671       6
##  7    0.155  0.393 0.359 0.0868 0.00671       7
##  8    0.157  0.400 0.342 0.0904 0.0102        8
##  9    0.156  0.397 0.349 0.0890 0.00872       9
## 10    0.156  0.396 0.352 0.0883 0.00804      10
```

NHANES: Residuals

```r
#calculate residuals
residuals <- as_tibble(residuals(health_m)) %>%  #calculate resid
  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         -0.0687      -0.243      0.547    -0.201    -0.0348        1
##  2         -0.0687      -0.243      0.547    -0.201    -0.0348        2
##  3         -0.0687      -0.243      0.547    -0.201    -0.0348        3
##  4         -0.0691      -0.244      0.565    -0.205    -0.0467        4
##  5         -0.155        0.607     -0.359    -0.0868   -0.00671       5
##  6         -0.155        0.607     -0.359    -0.0868   -0.00671       6
##  7         -0.155        0.607     -0.359    -0.0868   -0.00671       7
##  8         -0.157        0.600     -0.342    -0.0904   -0.0102        8
##  9         -0.156        0.603     -0.349    -0.0890   -0.00872       9
## 10         -0.156       -0.396     -0.352     0.912    -0.00804      10
```

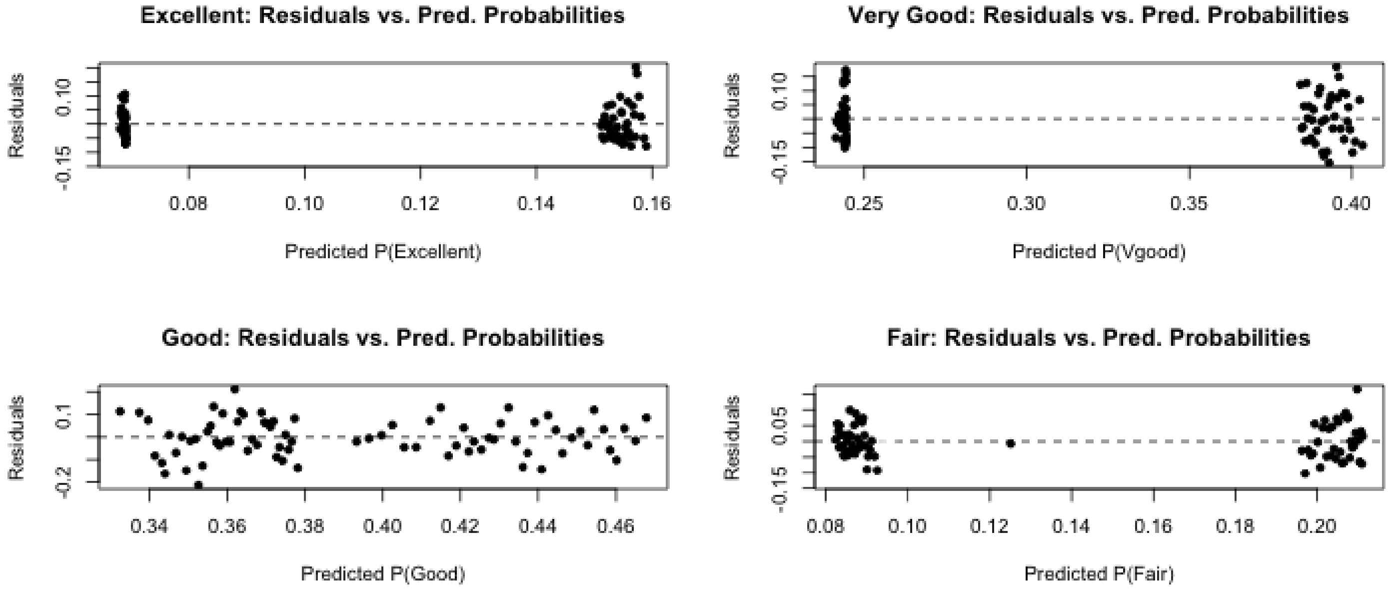
Make "augmented" dataset

```
health_m_aug <- inner_join(nhanes_adult, pred_probs) #add probs
health_m_aug <- inner_join(health_m_aug, residuals) #add resid
```

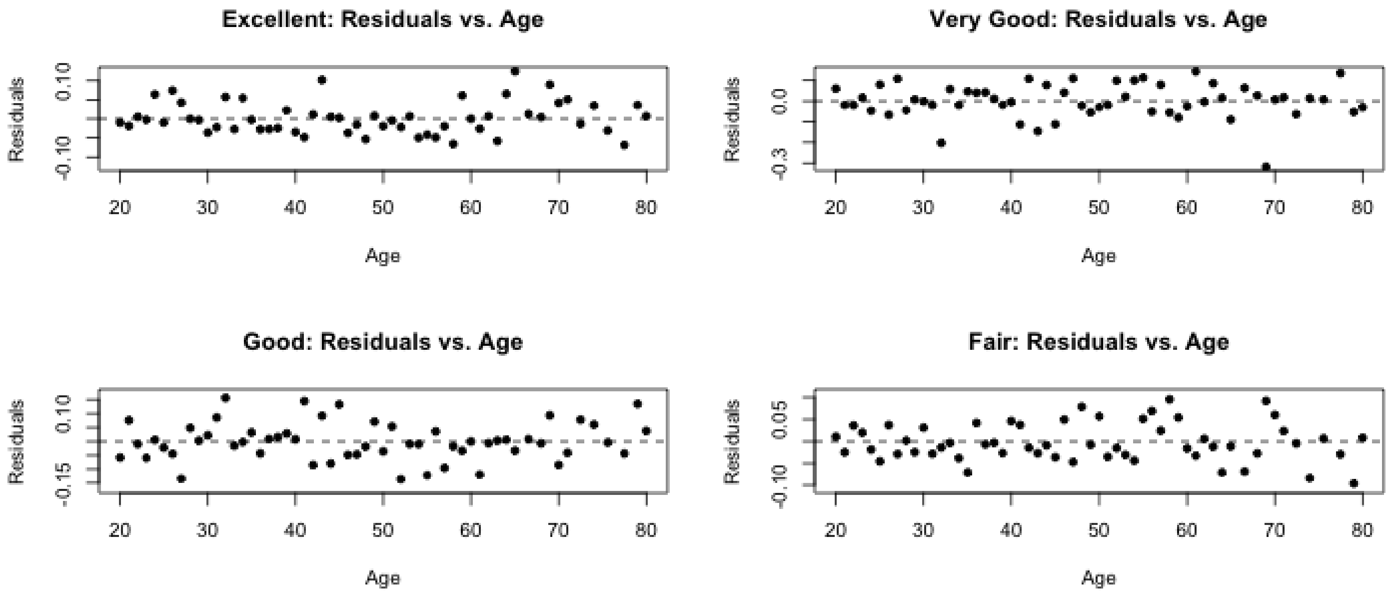
```
health_m_aug %>%
  glimpse()
```

```
## Observations: 6,465
## Variables: 15
## $ 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
## $ Education        <fct> High School, High School, High School, Some Coll
## $ obs_num          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## $ Excellent        <dbl> 0.06870508, 0.06870508, 0.06870508, 0.06906126, 
## $ Vgood            <dbl> 0.2432327, 0.2432327, 0.2432327, 0.2443614, 0.39
## $ Good             <dbl> 0.4527247, 0.4527247, 0.4527247, 0.4348186, 0.35
## $ Fair             <dbl> 0.20055763, 0.20055763, 0.20055763, 0.20503866, 
## $ Poor             <dbl> 0.034779881, 0.034779881, 0.034779881, 0.0467200
## $ resid.Excellent  <dbl> -0.06870508, -0.06870508, -0.06870508, -0.069061
## $ resid.Vgood      <dbl> -0.2432327, -0.2432327, -0.2432327, -0.2443614, 
## $ resid.Good       <dbl> 0.5472753, 0.5472753, 0.5472753, 0.5651814, -0.3
```

Binned residuals vs. pred. probabilities



Binned residuals vs. `Age`



STA 210

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 "-2.683227e-07" " 1.732639e-06"
## mean.Vgood     " 4.866088e-07" "-1.193899e-06"
## mean.Good      " 7.868508e-07" "-1.241316e-06"
## mean.Fair      "-1.081921e-06" " 6.788893e-07"
## mean.Poor      "7.678444e-08"  "2.368658e-08"
```

Calculating probabilities

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

$$p_{ij} = \frac{\exp\{\beta_{0j} + \beta_{1j}x_i\}}{1 + \sum_{j=2}^{k}\exp\{\beta_{0j} + \beta_{1j}x_i\}}$$

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

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

We will use these probabilities to assign a category of the response for each observation

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"))
```

STA 210

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   528   210     0     0
## 2 Vgood             0  1341   743     0     0
## 3 Good              0  1226  1316     0     0
## 4 Fair              0   296   625     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    0.0687 0.243 0.453 0.201  0.0348  Good
## 2    0.0687 0.243 0.453 0.201  0.0348  Good
## 3    0.0687 0.243 0.453 0.201  0.0348  Good
## 4    0.0691 0.244 0.435 0.205  0.0467  Good
## 5    0.155  0.393 0.359 0.0868 0.00671 Vgood
```

STA 210

Drop-in-deviance Test

- Suppose there are two models:
    - Model 1 includes predictors $x_1, \ldots, x_q$
    - Model 2 includes predictors $x_1, \ldots, x_q, x_{q+1}, \ldots, x_p$
- We want to test the hypotheses

$$H_0 : \beta_{q+1} = \cdots = \beta_p = 0$$
$$H_a : \text{ at least 1 } \beta_j \text{ is not} 0$$

- Use the **drop-in-deviance test** to compare models (similar to logistic regression)

STA 210

Add `Education` to the model?

- We consider adding the participants' `Education` level to the model.
    - Education takes values `8thGrade`, `9-11thGrade`, `HighSchool`, `SomeCollege`, and `CollegeGrad`
- Models we're testing:
    - Model 1: `Age`, `PhysActive`
    - Model 2: `Age`, `PhysActive`, `Education`

$$
\begin{aligned}
&H_0 : \beta_{9-11thGrade} = \beta_{HighSchool} = \beta_{SomeCollege} = \beta_{CollegeGrad}\\
&H_a : \text{ at least one } \beta_j \text{ is not equal to } 0
\end{aligned}
$$

STA 210

Add `Education` to the model?

$$H_0 : \beta_{9-11thGrade} = \beta_{HighSchool} = \beta_{SomeCollege} = \beta_{CollegeGrad}$$
$$H_a : \text{ at least one } \beta_j \text{ is not equal to 0}$$

```
m1 <- multinom(HealthGen ~ Age + PhysActive,
               data = nhanes_adult)
m2 <- multinom(HealthGen ~ Age + PhysActive + Education,
               data = nhanes_adult)
```

Add `Education` to the model?

```
m1 <- multinom(HealthGen ~ Age + PhysActive,
               data = nhanes_adult)
m2 <- multinom(HealthGen ~ Age + PhysActive + Education,
               data = nhanes_adult)
```

```
kable(anova(m1, m2, test = "Chisq"), format = "markdown")
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
Age + PhysActive	25848	16994.23		NA	NA	NA
Age + PhysActive + Education	25832	16505.10	1 vs 2	16	489.1319	0

At least one coefficient associated with `Education` is non-zero. Therefore, `Education` is a statistically significant predictor for `HealthGen`.