Logistic Regression

Model assumptions & diagnostics

Jenine Harris
Brown School

Importing and cleaning the data

```
# import the libraries cleaned file
libraries <- read.csv("/Users/harrisj/Box/teaching/Teaching/Fall2020/dat

# change data types
library(package = "tidyverse")
libraries.cleaned <- libraries %>%
  mutate(age = as.numeric(age))
```

Larger model

```
# run the odds.n.ends code again
lib.model <- glm(formula = uses.lib ~ age + sex + educ + parent + disabl
                 data = libraries.cleaned,
                 na.action = na.exclude,
                 family = binomial("logit"))
```

Checking logistic regression assumptions

- There are three assumptions for logistic regression:
  - independence of observations
  - linearity
  - no perfect multicollinearity.
- The generalized variance inflation factor (GVIF) work to check for multicollinearity.
  - The GVIF is similar to the VIF used for linear regression, but modified to account for the categorical outcome.
- Linearity could be checked by graphing the log-odds of the outcome against each continuous predictor to see if the relationship is linear (i.e., falling along a line).

Assumption: Independence of observations

- Independence of observations is about whether there are observations in the data that are dependent on each other.
- For example, siblings, close friends, or spouses are more likely to share some behaviors or characteristics than unrelated people and would therefore influence the amount of variation in the data and violate the independence of observations assumptions.
- The Pew Research Center conducted a phone survey where they selected a single person in a randomly selected household.

*This data collection strategy is likely to result in independent observations.

- The assumption is met.

Assumption: Linearity

- In linear regression the linearity assumption is checked by examining the relationship between each continuous predictor and the outcome variable.
- For logistic regression, the outcome variable is binary, so its relationship with another variable will never be linear.
- Instead of plotting the relationship of the outcome with each continuous predictor, linearity is tested by plotting the log-odds of the predicted probabilities for the outcome against each of the continuous predictors in the model.
- By examining the relationship between the predicted probabilities and a continuous predictor, the graph for checking linearity shows whether the predictions are equally accurate along the range of the values of the predictor.
- For example, are the predicted values equally accurate for people with a younger age compared to people with an older age.

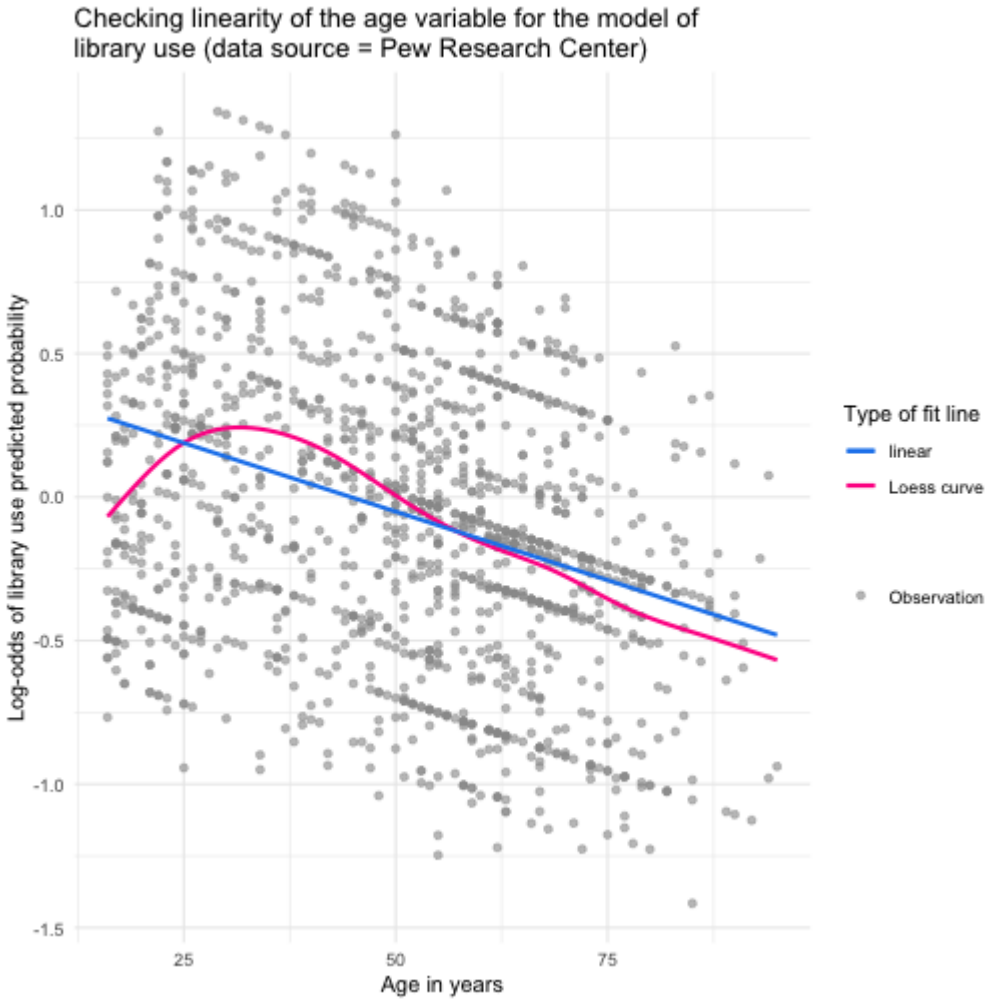
Checking linearity

```
# make a variable of the logit of the predicted values
logit.use <- log(lib.model$fitted.values/(1-lib.model$fitted.values))

# make a small data frame with the logit variable and the age predictor
linearity.data <- data.frame(logit.use, age = lib.model$model$age)

# create a plot with linear and actual relationships shown
linearity.data %>%
  ggplot(aes(x = age, y = logit.use))+
  geom_point(aes(size = "Observation"), color = "gray60", alpha = .6) +
  geom_smooth(se = FALSE, aes(color = "Loess curve")) +
  geom_smooth(method = lm, se = FALSE, aes(color = "linear")) +
  theme_minimal() +
  labs(x = "Age in years", y = "Log-odds of library use predicted probab
       title = "Checking linearity of the age variable for the model of\
  scale_color_manual(name="Type of fit line", values=c("dodgerblue2", "d
  scale_size_manual(values = 1.5, name = "")
```

Checking linearity



Assumption: No perfect multicollinearity

- The GVIF is similar to the VIF in linear regression.
- GVIF examines how well each predictor variable in the model is explained by the group of other predictor variables.
- If a predictor is well explained by the others, it is redundant and unnecessary. For the GVIF, often a threshold of $GVIF^{\frac{1}{2*Df}} < 2$ is used as a cutoff with values of 2 or higher indicating a failed multicollinearity assumption. The **`car`** package is used and the same `vif()` command as was used for the linear model can be used here:

```
# compute GVIF
car::vif(lib.model)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## age        1.254322  1        1.119965
## sex        1.051221  1        1.025291
## educ       1.309506  2        1.069737
## parent     1.101618  1        1.049580
## disabled   1.153173  1        1.073859
## rurality   1.118617  2        1.028420
## raceth     1.212126  2        1.049269
## ses        1.249162  2        1.057194
```

Model diagnostics

- In addition to checking assumptions, there are model diagnostics for determining whether there are any observations that are having an unusual impact on the model.
- An outlier is an observation with unusual values, regression outliers have unusual values of the outcome given the value(s) of predictor(s), and influential observations change the regression coefficients.
- The same measures from linear regression can be used to help identify outliers and influential values.

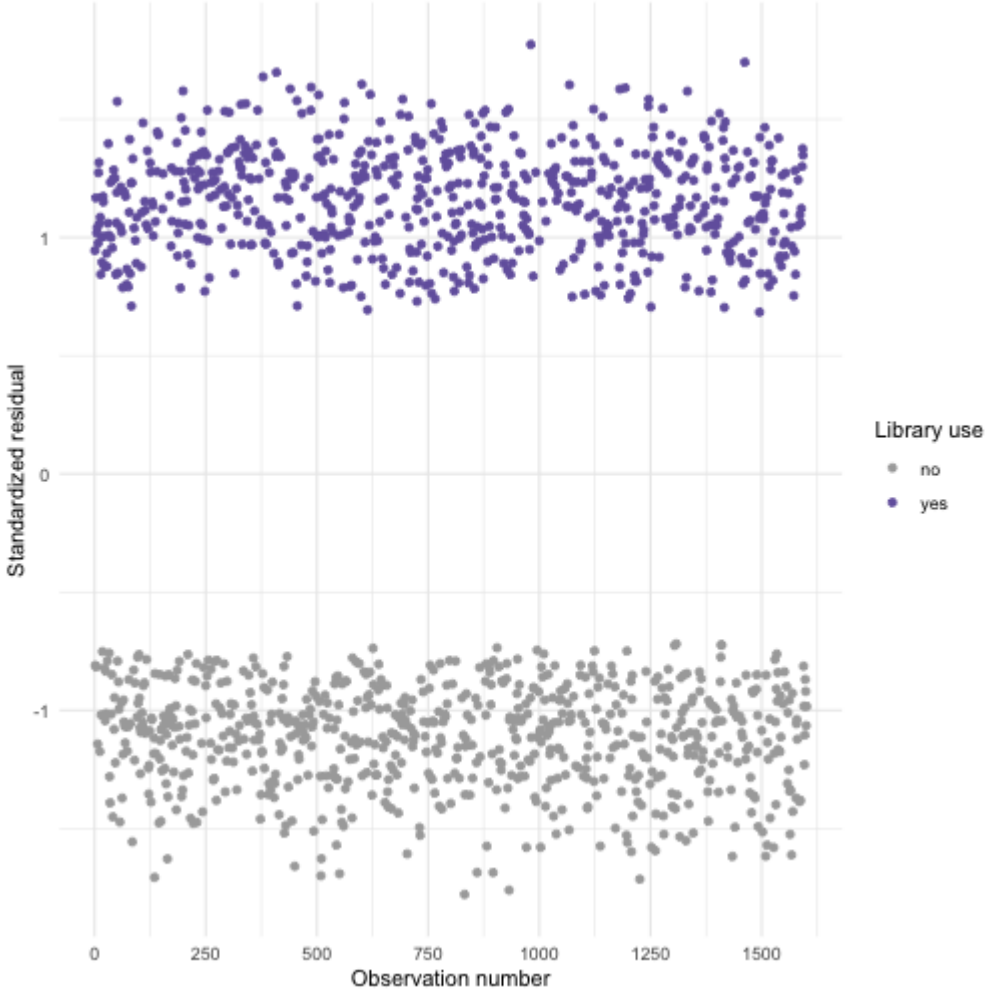
Using standardized residuals to find outliers

- Residuals are the distances between the predicted value of the outcome and the true value of the outcome for each person or observation in the data set.
- These values are standardized by computing z-scores for each one so that they follow a z-distribution.
- Z-scores that are greater than 1.96 or less than -1.96 are about two standard deviations or more away from the mean of a measure.
- In this case, they are more than two standard deviations away from the mean residual value.
- Very large values of standardized residuals can indicate that the predicted value for an observation is far from the true value for that observation, indicating that an examination of that observation could be useful.

```
# get standardized residuals and add to data frame
libraries.cleaned <- libraries.cleaned %>%
  mutate(standardized = rstandard(lib.model))

# check the residuals for large values > 2
libraries.cleaned %>%
  drop_na(standardized) %>%
  summarize(max.resid = max(abs(standardized)))
```

Examining the standardized residuals



Using df-betas to find influential values

- Observations with high df-betas (more than 2) may be influencing the model, causing large differences in the intercept or coefficients.

```
# get influence statistics
influence.lib.mod <- influence.measures(model = lib.model)

# summarize data frame with dfbetas, cooks, leverage
summary(object = influence.lib.mod$infmat)
```

```
##      dfb.1_              dfb.age              dfb.sxml        
##  Min.   :-0.1051922   Min.   :-0.0767940   Min.   :-0.039027  
##  1st Qu.:-0.0129510   1st Qu.:-0.0160893   1st Qu.:-0.023107  
##  Median : 0.0000000   Median : 0.0000000   Median :-0.015882  
##  Mean   : 0.0001392   Mean   :-0.0005355   Mean   :-0.001121  
##  3rd Qu.: 0.0125725   3rd Qu.: 0.0142736   3rd Qu.: 0.024395  
##  Max.   : 0.1142539   Max.   : 0.0825243   Max.   : 0.055434  
##                                                               
##     dfb.edom             dfb.et2d             dfb.prnt        
##  Min.   :-0.1161072   Min.   :-0.1179326   Min.   :-0.073641  
##  1st Qu.:-0.0106307   1st Qu.:-0.0069438   1st Qu.:-0.013862  
##  Median : 0.0000000   Median : 0.0000000   Median : 0.000000  
##  Mean   : 0.0005464   Mean   : 0.0001351   Mean   : 0.000333  
##  3rd Qu.: 0.0113543   3rd Qu.: 0.0068441   3rd Qu.: 0.014662
```

Using Cook's Distance to find influential values

- Cook's D is computed in a similar way to df-beta.
- For Cook's D, each observation is removed and the model is re-estimated without it.
- Cook's D then combines the differences between the models with and without an observation for *all the parameters* together instead of one at a time like the df-betas.
- A high Cook's D would indicate that removing the observation made a big difference and therefore it might be considered influential.
- The cutoff for a high Cook's D value is usually 4/n, the same as in linear regression.
- With 1427 observations in this model, a Cook's D greater than 0.0028031 will be problematic.

Examining Cook's D values

```
# save the data frame
influence.lib <- data.frame(influence.lib.mod$infmat)

# observations with high Cook's D
influence.lib %>%
  filter(cook.d > 4/1427)
```

```
##            dfb.1_     dfb.age    dfb.sxml    dfb.edom     dfb.et2d      dfb.p
## 135  -0.089968791 0.06811733  0.05291068 -0.01482971 -0.003299788  0.0396513
## 204   0.030721307 0.06698698 -0.02050929 -0.09820409 -0.108857230  0.0007253
## 329   0.002832481 0.08252428  0.04199633 -0.09143730 -0.105390995 -0.0109348
## 832   0.010410729 0.02204048  0.04365804 -0.02292003  0.001809933 -0.0558308
## 981   0.014767212 0.05758856  0.05008263 -0.10059417 -0.106892895  0.0025424
## 1067 -0.028273550 0.02666186  0.04253825  0.08914222  0.090112886 -0.0516576
##           dfb.dsbl   dfb.rrltys  dfb.rrltyr     dfb.rN.B     dfb.rN.W
## 135  -0.004729808  0.021942935 -0.05627540 -0.004880216 -0.027773586
## 204   0.040622780 -0.011490805 -0.02882916  0.080510047  0.002088386
## 329  -0.055368373 -0.010223876 -0.02738762  0.084766010  0.014049614
## 832   0.009065273  0.004802848 -0.05369690 -0.076283416 -0.010206138
## 981   0.056500556 -0.015055642 -0.01620049  0.010835404  0.023317516
## 1067  0.027851495 -0.073218857 -0.00850814 -0.027064355 -0.061144528
##           dfb.sslw      dfb.ssmd      dffit     cov.r      cook.d          hat
## 135   0.090412815  0.1094683737 -0.1621125 1.0010074 0.003005208 0.011863095
## 204  -0.032365340 -0.0027014760  0.1821590 1.0154358 0.002976640 0.020492118
## 329   0.046463660  0.0008830421  0.1889008 1.0116403 0.003526649 0.019068983
## 832  -0.001115642 -0.0225046580 -0.1505078 0.9968121 0.002804872 0.009432757
## 981  -0.019984909  0.0023303909  0.1591076 0.9965610 0.003265925 0.010104224
```

Using Leverage to find influential values

- Leverage is the influence that the observed value of the outcome has on the predicted value of the outcome.
- Leverage values range between 0 and 1.
- To determine which leverage values indicate influential observations, a cutoff of $\frac{2(k+1)}{n}$ is often used. In this case, the cutoff is $\frac{2(8+1)}{1427} = 0.0126139$.

```
# observations with high Leverage
influence.lib %>%
  filter(hat > 2*(12+1)/1427)
```

```
##            dfb.1_       dfb.age      dfb.sxml      dfb.edom      dfb.et2d
## 11    0.079409735  0.0107229420 -0.029561280  0.0256295349  0.012018677
## 108  -0.039393366 -0.0111769006  0.019748087  0.0678941555  0.076642077
## 123  -0.021281313  0.0355064851  0.019975575 -0.0510898854 -0.056242988
## 133  -0.077183444  0.0747610090  0.030241098  0.0684679590  0.075757975
## 183  -0.057559813 -0.0546836579  0.022616742  0.0032325745 -0.015737131
## 204   0.030721307  0.0669869812 -0.020509287 -0.0982040887 -0.108857230
## 226  -0.075334091 -0.0315274353  0.028120064 -0.0002796066 -0.020493355
## 329   0.002832481  0.0825242803  0.041996330 -0.0914372961 -0.105390995
## 480  -0.064404816  0.0724579739  0.032636544  0.0779878220  0.081721885
```

Problematic observations

- It was hard to tell from this output if any of the observations were outlying or influential by more than one metric.

```
# problematic observations
influence.lib %>%
  filter(hat > 2*(12+1)/1427 & cook.d > 4/1427)
```

```
##          dfb.1_     dfb.age    dfb.sxml    dfb.edom   dfb.et2d      dfb.prnt
## 204 0.030721307 0.06698698 -0.02050929 -0.09820409 -0.1088572  0.0007253862
## 329 0.002832481 0.08252428  0.04199633 -0.09143730 -0.1053910 -0.0109348717
##        dfb.dsbl  dfb.rrltys  dfb.rrltyr   dfb.rN.B    dfb.rN.W    dfb.sslw
## 204  0.04062278 -0.01149080 -0.02882916 0.08051005 0.002088386 -0.03236534
## 329 -0.05536837 -0.01022388 -0.02738762 0.08476601 0.014049614  0.04646366
##          dfb.ssmd     dffit    cov.r      cook.d        hat
## 204 -0.0027014760 0.1821590 1.015436 0.002976640 0.02049212
## 329  0.0008830421 0.1889008 1.011640 0.003526649 0.01906898
```

- It looks like two of the observations were problematic by more than one measure, which is a small number for such a large data set.

- To review these two cases, merge the `influence.lib` object with the `libraries.cleaned` data frame.

Merging the data and influence measures

- Add the observation numbers for each row to each of the data frames and used the observation numbers to merge.
- Add the predicted probabilities to the data frame as well to compare to the observed values.
- Once the data frames are merged and the predicted probabilities added, filter the two cases to review to see if there is anything suspicious.

```
# make row names as a variable
influence.lib <- influence.lib %>%
  rownames_to_column()

# merge data frame with diagnostic stats
libraries.cleaned.diag <- libraries.cleaned %>%
  rownames_to_column() %>%
  merge(influence.lib, by = 'rowname') %>%
  mutate(pred.prob = predict(lib.model, type = "response"))

# review influential observations
libraries.cleaned.diag %>%
  filter(hat > 2*(12+1)/1427 & cook.d > 4/1427)
```

```
##   rowname age    sex    parent disabled uses.lib    ses                raceth
```