Logistic (Logit) Regression

CJ 702: Advanced Criminal Justice Statistics

Thomas Bryan Smith*

February 12, 2025

Contents

1 Setting up your environment

```
# Load Packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
```

*University of Mississippi, tbsmit10@olemiss.edu

```
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(ggpubr)

# ============================================================================ #

# Load the NCVS dataset we have been working with:
person <- readRDS("../Data/person.rds")

# Check your data:
head(person)
```

```
## # A tibble: 6 x 18
##    YEAR YEARQ IDPER      IDHH    AGE   SEX   V3020 WGTPER   YIH WGTVIC_V VIOLENT
##   <dbl> <fct> <fct>      <fct>   <fct> <fct> <fct>  <dbl> <dbl>    <dbl>   <dbl>
## 1  2000 001   2000966984 200099~ 40-49 Male  Coll~  1063.    10       NA       0
## 2  2000 001   2000951294 200099~ 40-49 Fema~ Coll~   894.     9       NA       0
## 3  2000 001   2000470356 200099~ 12-17 Male  Elem~  1317.     9       NA       0
## 4  2000 001   2000205990 200016~ 35-39 Male  Coll~  1093.     4       NA       0
## 5  2000 001   2000361146 200016~ 30-34 Fema~ Coll~  1101.     4    2202.       1
## 6  2000 001   2000879996 200073~ 40-49 Male  High~  1063.     6       NA       0
## # i 7 more variables: WGTVIC_NV <dbl>, NONVIOLENT <dbl>, ADJINC_WT_V <dbl>,
## #   VLNT_WGT <dbl>, ADJINC_WT_NV <dbl>, NVLNT_WGT <dbl>, EDUC <fct>
```

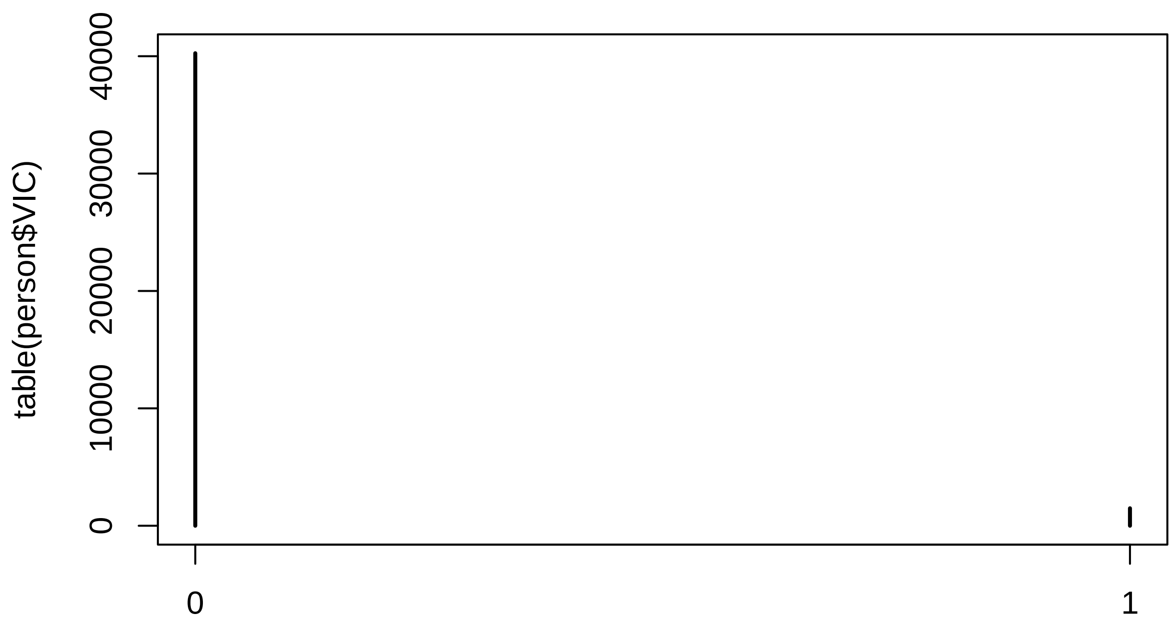
```r
# Take note of the variables:
## ID: Person ID                                  (numeric)
## IDHH: Household ID                             (numeric)
## PER_WGT: Person Weight                         (numeric)
## VIOLENT: Violent victimization count           (numeric, count, ratio)
## VLNT_WGT: Violent victimization weight         (numeric)
## NONVIOLENT: Nonviolent victimization count     (numeric, count, ratio)
## NVLNT_WGT: Nonviolent victimization weight     (numeric)
# YIH: Years in household                        (numeric, years, interval)
# EDUC: Education level                          (factor, ordinal)
# AGE: Age                                       (factor, years, ordinal)
# SEX: Sex                                       (factor, nominal)

# Check for missingness:
missing <- person %>%
  filter(!complete.cases(VIOLENT, NONVIOLENT,
                         YIH, EDUC, AGE, SEX)) %>%
  nrow()
n <- person %>% nrow()
missing / n
```

```
## [1] 0.08812478
```

```r
# Satisfied with sufficiently low missingness,
# you can perform listwise deletion:
person <- person %>%
  filter(complete.cases(VIOLENT, NONVIOLENT,
                        YIH, EDUC, AGE, SEX))

# Create your binary dependent variable (victimization)
person <- person %>%
  mutate(VIC = as.numeric((VIOLENT > 0) | (NONVIOLENT > 0)))
```

2 Descriptives and visualizing binary variables

```r
# You can generate a simple plot of the Bernoulli distribution of your
# dependent variable with the plot() and table() functions:
person$VIC |> table() |> plot()
```




```r
# This is useful for your own diagnostics, and understanding what proportion
# of respondents in your data were victimized. However, it's not analytically
# interesting and best described with the mean() function.

# Remember, the mean() of a Bernoulli random variable is the proportion, 'p',
# of observations with the affirmative / TRUE / "1" response:
person$VIC |> mean()
```

```
## [1] 0.03567151
```

```r
# You can find the variance, which is defined as p * (1 - p),
# with the var() function. However, like the plot, you typically
# wouldn't include this in a publication (it doesn't tell you much!)
person$VIC |> var()
```

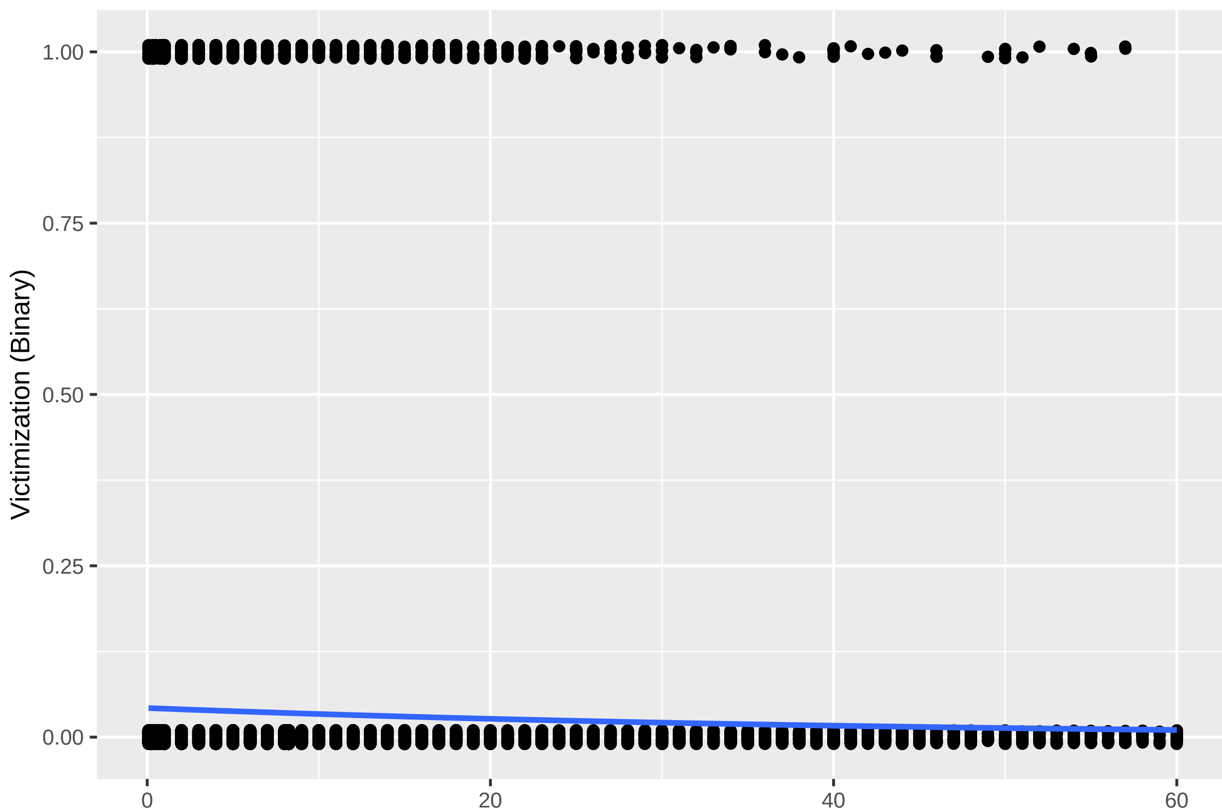
```
## [1] 0.03439987
```

```r
# The table() function by itself will provide you the frequencies for
# the variable:
person$VIC |> table()
```

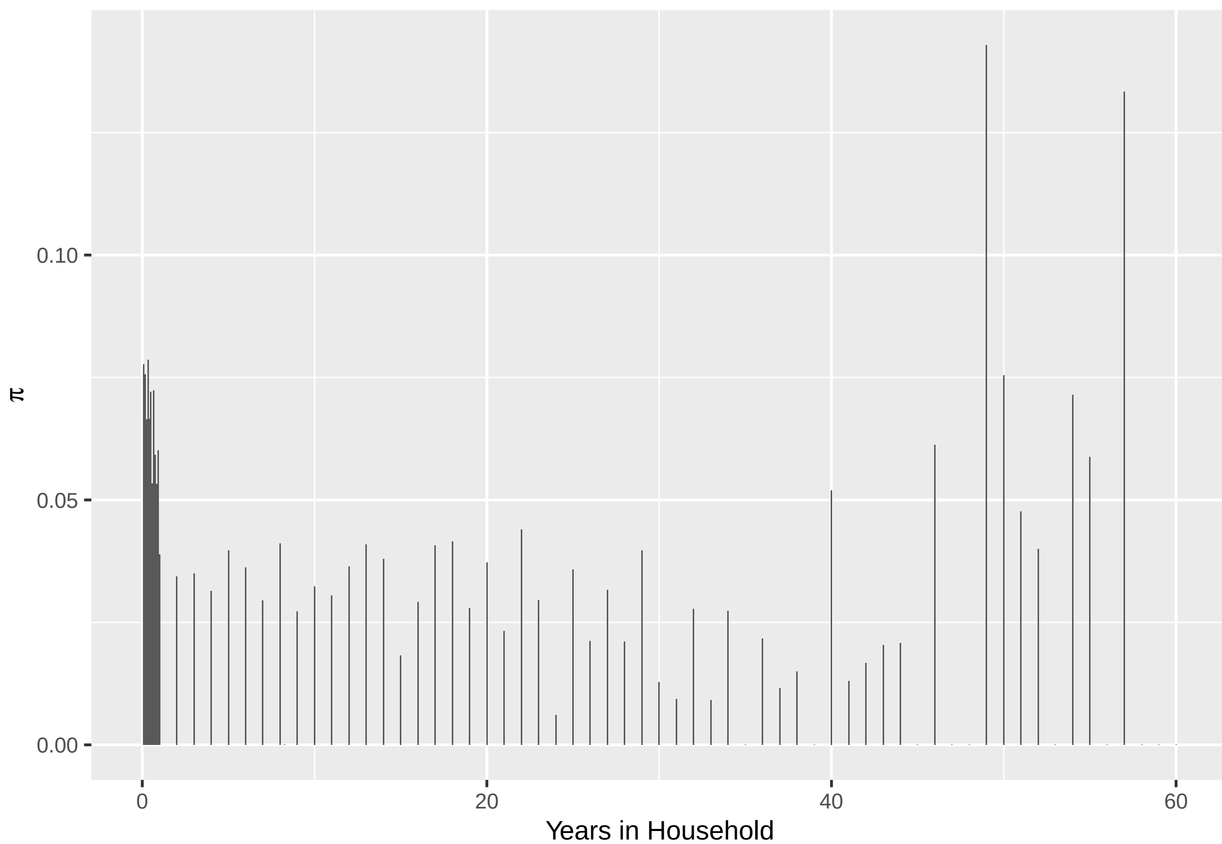
```
##
##     0     1
## 40253  1489
```

```r
# Bivariate Graphs
## Visualizing the relationship between two variables as a scatter plot:
ggplot(person, aes(x = YIH, y = VIC)) +
  geom_jitter(width = 0.01, height = 0.01) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = TRUE) +
  labs(x = " ",
       y = "Victimization (Binary)") +
  theme(text = element_text(size = 10))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

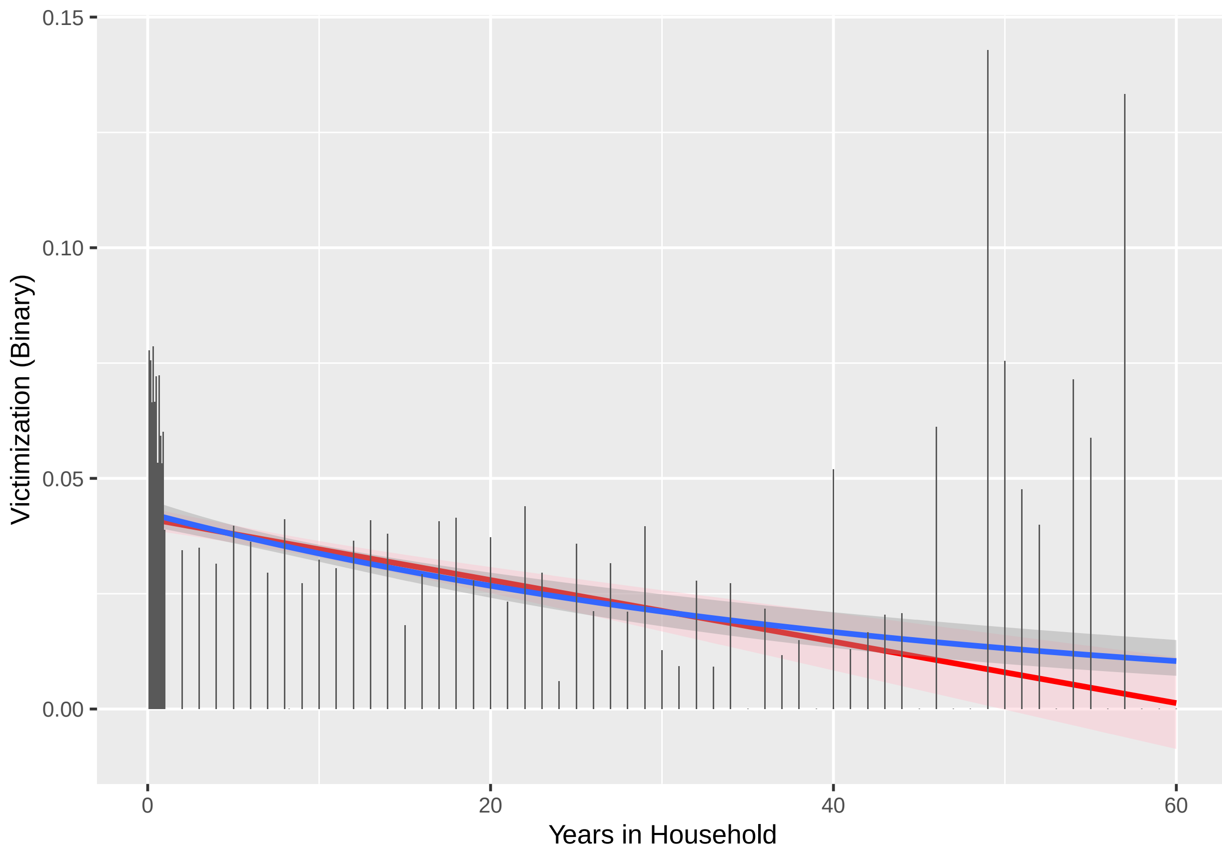



```
## Visualizing the relationship between two variables as a bar chart:
person %>%
  select(YIH, VIC) %>%
  group_by(YIH) %>%
  summarise(`pi` = mean(VIC)) %>%
  ggplot(aes(x = YIH, y = `pi`)) +
    geom_bar(stat = "identity") +
    labs(x = "Years in Household",
         y = expression(pi)) +
    theme(text = element_text(size = 10))
```



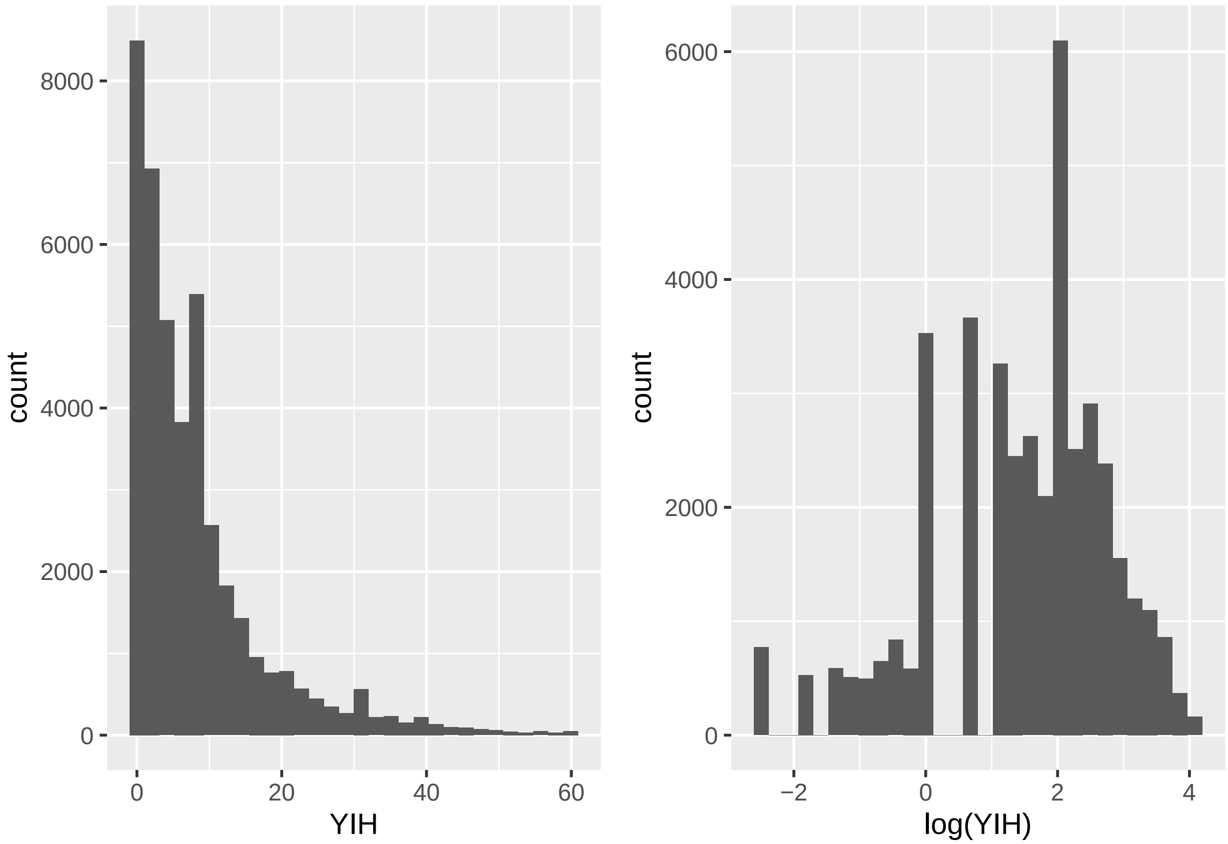

```
## Combining both approaches:
### Prepare the bar data using what we know about pi = f(x):
person <- person %>%
  group_by(YIH) %>%
  mutate(n = n(),
         `pi` = mean(VIC),
         `pi/n` = `pi` / n) %>%
  ungroup()

### Build the combined plot:
ggplot(person, aes(x = YIH, y = VIC)) +
  geom_smooth(method = "lm",
              se = TRUE,
              color = "red",
              fill = "pink") +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = TRUE) +
  geom_bar(aes(y = `pi/n`), stat = "identity") +
  labs(x = "Years in Household",
       y = "Victimization (Binary)") +
  theme(text = element_text(size = 10))
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



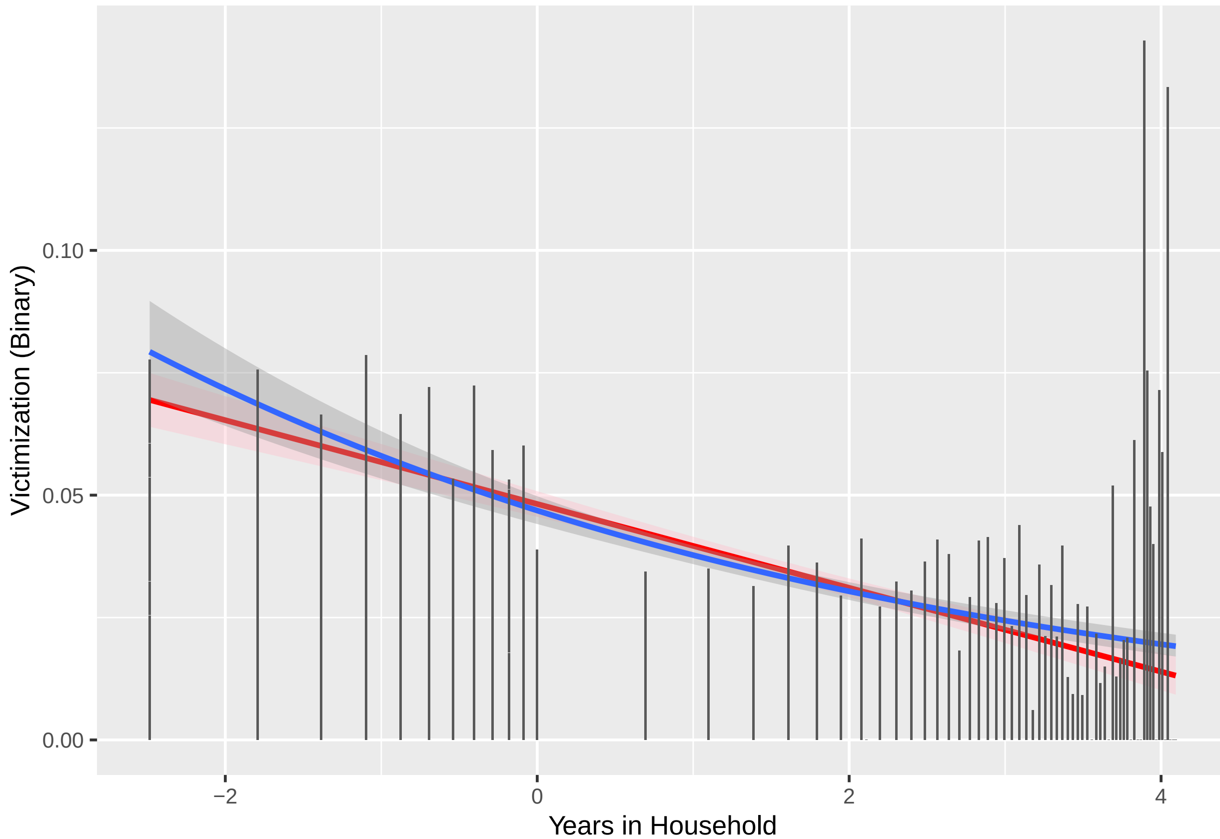

```
## Looking at the previous plots, you may notice that the
## "years in household" variable is right-skewed.
## To 'normalize' the variable, we can log-transform it:
ggarrange(ggplot(person, aes(x = YIH)) + geom_histogram(),
          ggplot(person, aes(x = log(YIH))) + geom_histogram())
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
## We can apply this log transformation to the code for the
## above combined plot to examine how it affects bivariate
## model fit:
person <- person %>%
  group_by(log(YIH)) %>%
  mutate(n = n(),
         `pi` = mean(VIC),
         `pi/n` = `pi` / n) %>%
  ungroup()

ggplot(person, aes(x = log(YIH), y = VIC)) +
  geom_smooth(method = "lm",
              se = TRUE,
              color = "red",
              fill = "pink") +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = TRUE) +
  geom_bar(aes(y = `pi/n`), stat = "identity") +
  labs(x = "Years in Household",
       y = "Victimization (Binary)") +
  theme(text = element_text(size = 10))
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



3 Estimating binomial generalized linear models

```
# Let's start by fitting a linear probability model using the lm() function:
summary(m1 <- lm(VIC ~ log(YIH) +                    # Log Years in Household
                       I(log(YIH)^2) +               # Log Years in Household^2
                       scale(as.numeric(AGE)) +      # Age (Ordinal)
                       scale(as.numeric(EDUC)) +     # Education (Ordinal)
                       SEX,                          # Sex (Binary)
                 data = person))
```

```
##
## Call:
## lm(formula = VIC ~ log(YIH) + I(log(YIH)^2) + scale(as.numeric(AGE)) +
##     scale(as.numeric(EDUC)) + SEX, data = person)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09932 -0.03856 -0.03157 -0.02626  0.98197
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.0422185  0.0017513  24.107  < 2e-16 ***
## log(YIH)                 -0.0119817  0.0009929 -12.068  < 2e-16 ***
```

```
## I(log(YIH)^2)            0.0023538  0.0003871   6.081 1.20e-09 ***
## scale(as.numeric(AGE))  -0.0048463  0.0010320  -4.696 2.66e-06 ***
## scale(as.numeric(EDUC))  0.0014889  0.0009607   1.550   0.1212
## SEXFemale                0.0031691  0.0018149   1.746   0.0808 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.185 on 41736 degrees of freedom
## Multiple R-squared:  0.00511,    Adjusted R-squared:  0.004991
## F-statistic: 42.88 on 5 and 41736 DF,  p-value: < 2.2e-16
```

```
## Model Specification:
### Pr(VIC = 1) ~ b0 + b1(YIH) + b2(YIH^2) + b3(AGE) + b4(EDUC) + b5(SEX) + e

## Interpretation (same as OLS, but the DV is a probability):
### b0: the intercept, the average value of the DV when all IVs are 0.
### bk: the average change in the probability of the DV (1),
###     for each interval increase in the IV, controlling for the other IVs.
### Pr(>|t|): P-value, probability of observing the current (or a more extreme)
###           effect size under the assumption that the null hypothesis is true.
### Degrees of freedom: n - (k + 1); n = # obs; k = # IVs.
### R-squared: Proportion of variance in the DV explained by the IVs.
### F-test: Overall model significance.

## For all of the reasons discussed in class, and demonstrated in the figures
## you generated in the previous section of this R module, it is typically
## ill-advised to fit a linear probability model.

# ============================================================================ #

# Now, let's fit a logit model with the same specification, but
# Pr(VIC = 1) becomes log(VIC / (1 - VIC)):

summary(m2 <- glm(VIC ~ log(YIH) +                # Log Years in Household
                        I(log(YIH)^2) +           # Log Years in Household^2
                        scale(as.numeric(AGE)) +  # Age (Ordinal)
                        scale(as.numeric(EDUC)) + # Education (Ordinal)
                        SEX,                      # Sex (Binary)
                  data = person,
                  family = binomial(logit)))
```

```
##
## Call:
## glm(formula = VIC ~ log(YIH) + I(log(YIH)^2) + scale(as.numeric(AGE)) +
##     scale(as.numeric(EDUC)) + SEX, family = binomial(logit),
##     data = person)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -3.17674    0.05053 -62.865  < 2e-16 ***
## log(YIH)                -0.24488    0.02133 -11.481  < 2e-16 ***
## I(log(YIH)^2)            0.03492    0.01007   3.466 0.000528 ***
## scale(as.numeric(AGE))  -0.13625    0.03003  -4.537  5.7e-06 ***
## scale(as.numeric(EDUC))  0.04197    0.02822   1.488 0.136857
```

```
## SEXFemale                 0.09214    0.05318   1.733 0.083134 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12851  on 41741  degrees of freedom
## Residual deviance: 12670  on 41736  degrees of freedom
## AIC: 12682
##
## Number of Fisher Scoring iterations: 6
```

```
## Note that the only changes to the code are: (1) the function, which
## changes from lm() [linear model] to glm() [generalized linear model],
## and we introduce the "family =" option with the "binomial(logit)"
## link function.

# Interpreting the results
## Log Odds
summary(m2)
```

```
##
## Call:
## glm(formula = VIC ~ log(YIH) + I(log(YIH)^2) + scale(as.numeric(AGE)) +
##     scale(as.numeric(EDUC)) + SEX, family = binomial(logit),
##     data = person)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -3.17674    0.05053 -62.865  < 2e-16 ***
## log(YIH)                 -0.24488    0.02133 -11.481  < 2e-16 ***
## I(log(YIH)^2)             0.03492    0.01007   3.466 0.000528 ***
## scale(as.numeric(AGE))   -0.13625    0.03003  -4.537  5.7e-06 ***
## scale(as.numeric(EDUC))   0.04197    0.02822   1.488 0.136857
## SEXFemale                 0.09214    0.05318   1.733 0.083134 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12851  on 41741  degrees of freedom
## Residual deviance: 12670  on 41736  degrees of freedom
## AIC: 12682
##
## Number of Fisher Scoring iterations: 6
```

```
### Intercept: -3.18      When all IVs are 0, we expect the average
###                       log odds of victimization to be -3.18.

### AGE: -0.14            For each standard deviation increase in age, we expect
###                       an average reduction of 0.14 in the log odds
###                       of victimization, net of control variables.
```

```
### SEX (Female): 0.09    On average, women are expected to score 0.09
###                       greater than men on the log-odds of victimization.

### Note that these interpretations are all a little clunky. This is because
### there is no real 'meaningful' interpretation for the log-odds.
### It is an unintuitive transformation.


## Odds Ratios
### Conveniently, you can use this handy line of code to simultaneously
### exponentiate your coefficients AND confidence intervals!
exp(cbind(coef(m2), confint(m2)))
```

```
## Waiting for profiling to be done...
```

```
##                                      2.5 %     97.5 %
## (Intercept)              0.04172157 0.03775226 0.04602326
## log(YIH)                 0.78279586 0.75118786 0.81672662
## I(log(YIH)^2)            1.03553652 1.01509692 1.05599809
## scale(as.numeric(AGE))   0.87262350 0.82273087 0.92551763
## scale(as.numeric(EDUC))  1.04286828 0.98671379 1.10211919
## SEXFemale                1.09652154 0.98812791 1.21719159
```

```
### Intercept: 0.04       When all IVs are 0, we expect the average
###                       odds of victimization to be 0.04.

### AGE: 0.87             For each standard deviation increase in age, we expect
###                       a 0.87 factor change in victimization likelihood/
###                       a 13 percent reduction in the odds of victimization.

### SEX (Female): 1.10    On average, women are expected to report at
###                       least one victimization 10% more frequently than men.
```

4 Post-estimation functions and visualization

```
# Predicted Probabilities
## Whole sample:
head(predict(m2, type = "response"))
```

```
##          1          2          3          4          5          6
## 0.02703490 0.02983363 0.03291636 0.03176214 0.03677757 0.02702775
```

```
## Typical / interesting individuals:
pred_prob <- function(y){
  exp(y) / (1 + exp(y))
}

### Keep in mind the order of your variables / coefficients:
#### 1. Intercept
```

```r
#### 2. Years in Household
#### 3. Years in Household (Squared)
#### 4. Age (Centered, Z-Score)
#### 5. Education (Centered, Z-Score)
#### 6. Sex (Female = 1, Binary)

### Men (0 years in home, average age and education level):
sum(coef(m2) * c(1, 0, 0, 0, 0, 0)) %>%
  pred_prob()
```

```
## [1] 0.04005059
```

```r
### Women (0 years in home, average age and education level):
sum(coef(m2) * c(1, 0, 0, 0, 0, 1)) %>%
  pred_prob()
```

```
## [1] 0.04374722
```

```r
### Women w. a PhD (0 years in home, average age):
sum(coef(m2) * c(1,
                 0,
                 0,
                 0,
                 person$EDUC %>% as.numeric() %>% scale() %>% max(),
                 1)) %>%
  pred_prob()
```

```
## [1] 0.04948469
```

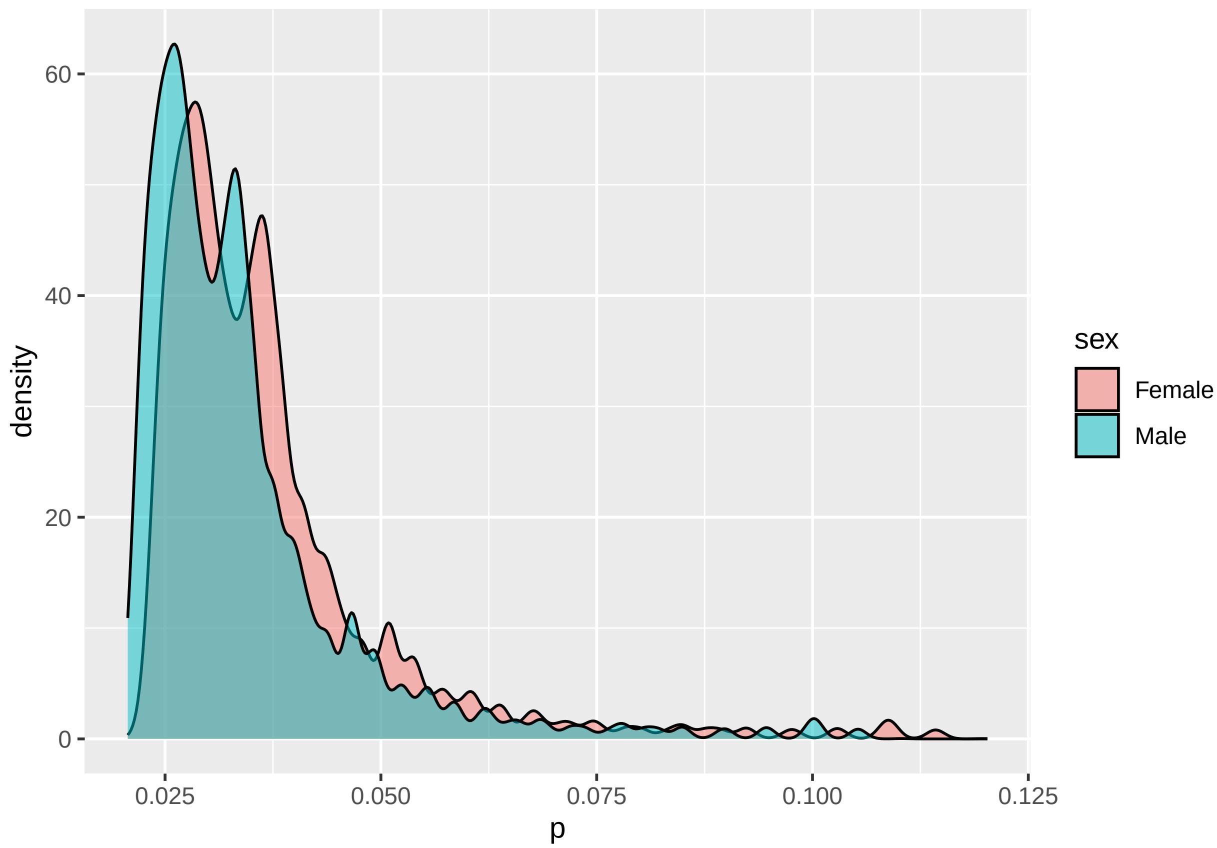
```r
## Testing the effect of specific parameters on the sample:
### The following code will give you the predicted probabilities
### for the whole sample (maintaining their observed scores for
### most variables), but treat all observations as FEMALE.
### This is achieved by "forcing" the "SEX" variable to be "Female".
### If you View() the ppf object, you can verify that all observations
### are treated as "Female" for the purpose of generating predictions.
### You can do this for any regression, and any variable!
ppf <- data.frame("YIH" = person$YIH,
                  "AGE" = person$AGE,
                  "EDUC" = person$EDUC,
                  "SEX" = as.factor("Female"))
head(fm_pp <- predict(m2, newdata = ppf, type = "response"))
```

```
##          1          2          3          4          5          6
## 0.02956720 0.02983363 0.03597919 0.03472142 0.03677757 0.02955940
```

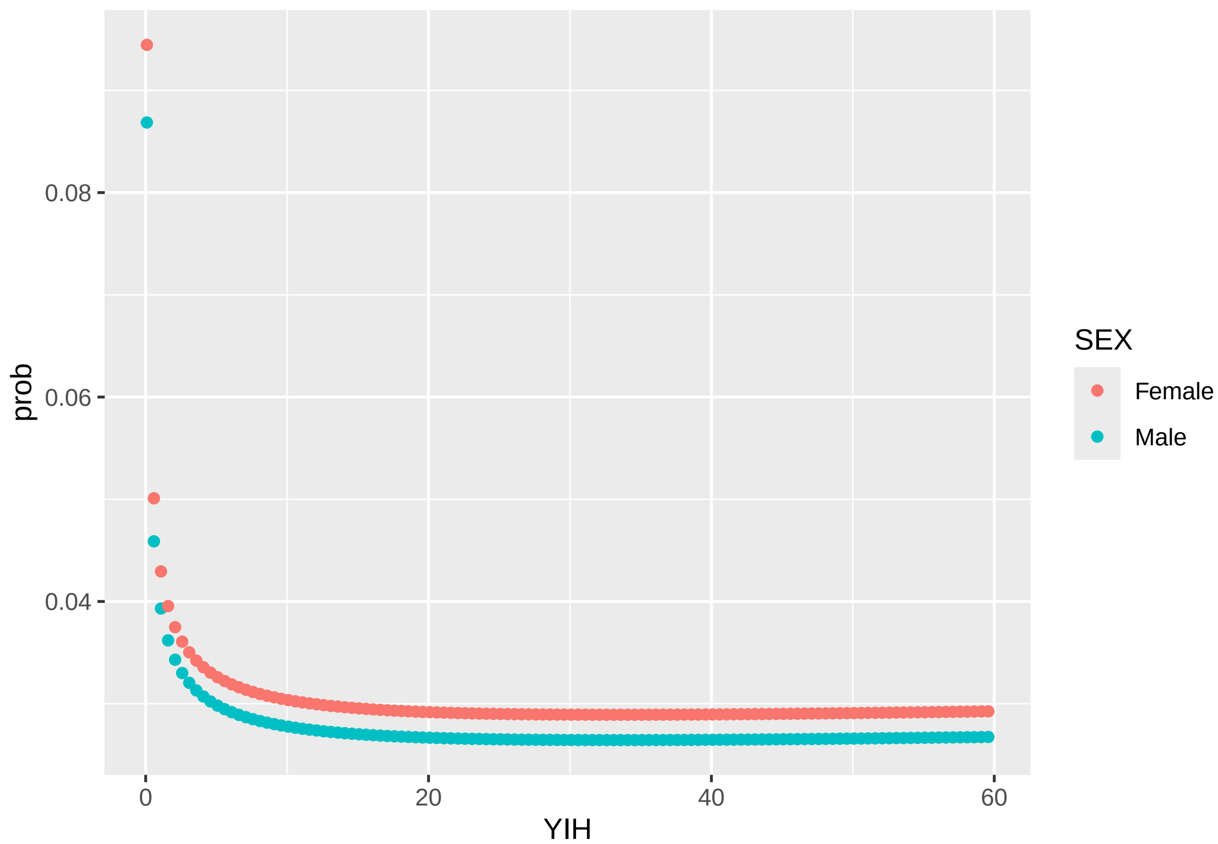
```r
### If we generate the same for men, we could look at how the distribution
### changes when comparing men and women:
ppm <- data.frame("YIH" = person$YIH,
                  "AGE" = person$AGE,
                  "EDUC" = person$EDUC,
                  "SEX" = as.factor("Male"))
head(m_pp <- predict(m2, newdata = ppm, type = "response"))
```

```
##          1          2          3          4          5          6
## 0.02703490 0.02727916 0.03291636 0.03176214 0.03364915 0.02702775
```

```
### Now we can plot the distributions for males v. females side by side.
### Note that, because females a predicted to report victimization
### more frequently, their distribution has shifted slightly to the right.
ggplot(data.frame("p" = c(m_pp, fm_pp),
                  "sex" = c(rep("Male", length(m_pp)),
                            rep("Female", length(fm_pp)))),
      aes(x = p, fill = sex)) +
  geom_density(alpha = 0.5)
```




```
## Calculating predicted probabilities for a range of values:
pp <- data.frame("YIH" = seq(min(person$YIH), max(person$YIH), by = 0.5) %>%
                   rep(each = 2),
                 "AGE" = mean(as.numeric(person$AGE)),
                 "EDUC" = mean(as.numeric(person$EDUC)),
                 "SEX" = as.factor(c("Male", "Female")))
pp$prob <- predict(m2, newdata = pp, type = "response")

ggplot(pp, aes(x = YIH, y = prob, color = SEX)) +
  geom_point()
```




```r
### Note that this particular approach to visualizing your
### model predictions is particularly helpful for visualizing
### non-linear model paramters (and interactions).

### In general, generating model predicitons are a very good
### way of understanding what a complex model might be telling
### you about your sample!

# ============================================================================= #

# Post-estimation functions
## Most of the same post-estimation functions that we used for OLS
## also apply to Logit (you should see some of them in the above code!)
## As a reminder:

### Akaike's Information Criterion (AIC)
AIC(m2)
```

```
## [1] 12682.24
```

```r
### Bayesian Information CRiterion (BIC)
BIC(m2)
```

```
## [1] 12734.07
```

```r
### You can extract the coefficients as a named numeric vector:
coef(m2)
```

```
##             (Intercept)                log(YIH)           I(log(YIH)^2)
##             -3.17673709             -0.24488333              0.03491967
##  scale(as.numeric(AGE)) scale(as.numeric(EDUC))               SEXFemale
##             -0.13625108              0.04197488              0.09214293
```

```r
### You can generate a named matrix of confidence intervals:
confint(m2, level = 0.95)
```

```
## Waiting for profiling to be done...
```

```
##                               2.5 %      97.5 %
## (Intercept)             -3.27671001 -3.07860829
## log(YIH)                -0.28609951 -0.20245085
## I(log(YIH)^2)            0.01498409  0.05448637
## scale(as.numeric(AGE))  -0.19512614 -0.07740210
## scale(as.numeric(EDUC)) -0.01337526  0.09723486
## SEXFemale               -0.01194312  0.19654623
```

```r
### You can generated a named numeric vector of predicted marginal scores:
fitted(m2) %>% head()
```

```
##          1          2          3          4          5          6
## 0.02703490 0.02983363 0.03291636 0.03176214 0.03677757 0.02702775
```

```r
predict(m2) %>% head()
```

```
##         1         2         3         4         5         6
## -3.583220 -3.481831 -3.380315 -3.417203 -3.265396 -3.583491
```

```r
### Analysis of Deviance:
anova(m2)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: VIC
##
## Terms added sequentially (first to last)
##
##
##                         Df Deviance Resid. Df Resid. Dev
## NULL                                    41741      12851
## log(YIH)                 1  150.711     41740      12700
## I(log(YIH)^2)            1    6.817     41739      12694
## scale(as.numeric(AGE))   1   18.035     41738      12676
## scale(as.numeric(EDUC))  1    2.305     41737      12673
## SEX                      1    3.010     41736      12670
```

```r
#### You can use this to compare model fit:
anova(m1, m2)
```

```
## Analysis of Variance Table
##
## Model 1: VIC ~ log(YIH) + I(log(YIH)^2) + scale(as.numeric(AGE)) + scale(as.numeric(EDUC)) +
##     SEX
## Model 2: VIC ~ log(YIH) + I(log(YIH)^2) + scale(as.numeric(AGE)) + scale(as.numeric(EDUC)) +
##     SEX
##   Res.Df     RSS Df Sum of Sq F Pr(>F)
## 1  41736  1428.5
## 2  41736 12670.2  0    -11242
```

```r
### You can print your variance-covariance matrix:
vcov(m2)
```

```
##                          (Intercept)      log(YIH) I(log(YIH)^2)
## (Intercept)             0.0025535186 -1.040693e-04 -2.384825e-04
## log(YIH)               -0.0001040693  4.549539e-04 -1.136054e-04
## I(log(YIH)^2)          -0.0002384825 -1.136054e-04  1.014978e-04
## scale(as.numeric(AGE))  0.0004702201 -8.699884e-05 -6.141607e-05
## scale(as.numeric(EDUC)) -0.0001099118 -1.334927e-06  2.497381e-05
## SEXFemale              -0.0015376571  3.380997e-06 -4.622765e-06
##                        scale(as.numeric(AGE)) scale(as.numeric(EDUC))
## (Intercept)                      4.702201e-04           -1.099118e-04
## log(YIH)                        -8.699884e-05           -1.334927e-06
## I(log(YIH)^2)                   -6.141607e-05            2.497381e-05
## scale(as.numeric(AGE))           9.017117e-04           -2.977837e-04
## scale(as.numeric(EDUC))         -2.977837e-04            7.961737e-04
## SEXFemale                       -2.488157e-05           -2.618781e-05
##                            SEXFemale
## (Intercept)            -1.537657e-03
## log(YIH)                3.380997e-06
## I(log(YIH)^2)          -4.622765e-06
## scale(as.numeric(AGE)) -2.488157e-05
## scale(as.numeric(EDUC)) -2.618781e-05
## SEXFemale               2.827707e-03
```

```r
### The 'car' package will let you find the variance inflation factor (VIF):
vif(m2)
```

```
##                log(YIH)           I(log(YIH)^2)  scale(as.numeric(AGE))
##                1.522236                1.554570                1.292735
## scale(as.numeric(EDUC))                     SEX
##                1.145225                1.001054
```