432 Class 18

https://thomaselove.github.io/432-2025/

2025-03-20

## Today’s Agenda

* Two types of Hurdle model (one Poisson, one NB)
* Can we fit a linear model to a count outcome?
* Selecting non-linear terms in light of Spearman
* Fitting a Poisson regression with the rms package
* Checking Assumptions in Logistic Regression Models

## R Setup

knitr::opts\_chunk$set(comment=NA)  
  
library(janitor); library(gt); library(broom)   
library(rsample); library(yardstick)  
library(car); library(here); library(conflicted)  
library(countreg) ## for rootograms  
library(topmodels) ## for rootograms  
library(mosaic)  
library(pscl)   
library(rms)  
library(easystats)  
library(tidyverse)  
  
theme\_set(theme\_bw())  
  
conflicts\_prefer(dplyr::select(), dplyr::filter(), base::max(),   
 base::sum(), rms::Predict(),   
 yardstick::rmse(), yardstick::mae(),  
 pscl::zeroinfl(), pscl::hurdle())

## The medicare data from Class 1

medicare <- read\_csv(here("c17/data/medicare.csv"),   
 show\_col\_types = FALSE) |>   
 mutate(across(where(is\_character), as\_factor),  
 subject = as.character(subject),   
 insurance = fct\_relevel(insurance, "no", "yes"),  
 logvisits = log(visits + 1)) ## needed because some have 0 visits  
  
set.seed(432)  
med\_split <- initial\_split(medicare, prop = 0.75)  
  
med\_train = training(med\_split)  
med\_test = testing(med\_split)

## The medicare data

medicare

# A tibble: 4,406 × 9  
 subject visits hospital health chronic sex school insurance logvisits  
 <chr> <dbl> <dbl> <fct> <dbl> <fct> <dbl> <fct> <dbl>  
 1 1 5 1 average 2 male 6 yes 1.79   
 2 2 1 0 average 2 female 10 yes 0.693  
 3 3 13 3 poor 4 female 10 no 2.64   
 4 4 16 1 poor 2 male 3 yes 2.83   
 5 5 3 0 average 2 female 6 yes 1.39   
 6 6 17 0 poor 5 female 7 no 2.89   
 7 7 9 0 average 0 female 8 yes 2.30   
 8 8 3 0 average 0 female 8 yes 1.39   
 9 9 1 0 average 0 female 8 yes 0.693  
10 10 0 0 average 0 female 8 yes 0   
# ℹ 4,396 more rows

## Reiterating the Goal

Predict visits using these 6 predictors…

| Predictor | Description |
| --- | --- |
| hospital | # of hospital stays |
| health | self-rated health (poor, average, excellent) |
| chronic | # of chronic conditions |
| sex | male or female |
| school | years of education |
| insurance | subject (also) has private insurance? (yes/no) |

## First Four models (fit last class)

mod\_1 <- glm(visits ~ hospital + health + chronic +  
 sex + school + insurance,  
 data = med\_train, family = "poisson")  
mod\_2 <- glm.nb(visits ~ hospital + health + chronic +  
 sex + school + insurance,  
 data = med\_train)  
mod\_3 <- zeroinfl(visits ~ hospital + health +   
 chronic + sex + school + insurance,  
 data = med\_train)  
mod\_4 <- zeroinfl(visits ~ hospital + health + chronic +  
 sex + school + insurance,  
 dist = "negbin", data = med\_train)

## First Four Models, augmented

mod\_1\_aug <- augment(mod\_1, med\_train,   
 type.predict = "response")  
mod\_2\_aug <- augment(mod\_2, med\_train,   
 type.predict = "response")

Warning: The `augment()` method for objects of class `negbin` is not maintained by the broom team, and is only supported through the `glm` tidier method. Please be cautious in interpreting and reporting broom output.  
  
This warning is displayed once per session.

mod\_3\_aug <- med\_train |>  
 mutate(".fitted" = predict(mod\_3, type = "response"),  
 ".resid" = resid(mod\_3, type = "response"))  
mod\_4\_aug <- med\_train |>  
 mutate(".fitted" = predict(mod\_4, type = "response"),  
 ".resid" = resid(mod\_4, type = "response"))

## First Four Model Summaries

mets <- metric\_set(rsq, rmse, mae)  
mod\_1\_summary <-   
 mets(mod\_1\_aug, truth = visits, estimate = .fitted) |>  
 mutate(model = "mod\_1") |> relocate(model)  
mod\_2\_summary <-   
 mets(mod\_2\_aug, truth = visits, estimate = .fitted) |>  
 mutate(model = "mod\_2") |> relocate(model)  
mod\_3\_summary <-   
 mets(mod\_3\_aug, truth = visits, estimate = .fitted) |>  
 mutate(model = "mod\_3") |> relocate(model)  
mod\_4\_summary <-   
 mets(mod\_4\_aug, truth = visits, estimate = .fitted) |>  
 mutate(model = "mod\_4") |> relocate(model)

## Training Sample through mod\_4

bind\_rows(mod\_1\_summary, mod\_2\_summary,   
 mod\_3\_summary, mod\_4\_summary) |>   
 pivot\_wider(names\_from = model,   
 values\_from = .estimate) |>   
 gt() |> fmt\_number(decimals = 3) |>  
 tab\_options(table.font.size = 20)

| .metric | .estimator | mod\_1 | mod\_2 | mod\_3 | mod\_4 |
| --- | --- | --- | --- | --- | --- |
| rsq | standard | 0.100 | 0.078 | 0.108 | 0.094 |
| rmse | standard | 6.594 | 6.941 | 6.560 | 6.709 |
| mae | standard | 4.189 | 4.252 | 4.164 | 4.191 |

# Hurdle Models

## The Hurdle Model

The hurdle model is a two-part model that specifies one process for zero counts and another process for positive counts. The idea is that positive counts occur once a threshold is crossed, or put another way, a hurdle is cleared. If the hurdle is not cleared, then we have a count of 0.

* The first part of the model is typically a **binary logistic regression** model. This models whether an observation takes a positive count or not.
* The second part of the model is usually a truncated Poisson or Negative Binomial model. Truncated means we’re only fitting positive counts, and not zeros.

# mod\_5: Poisson-Logistic Hurdle Model

## Fitting a Hurdle Model / Poisson-Logistic

In fitting a hurdle model to our medicare training data, the interpretation would be that one process governs whether a patient visits a doctor or not, and another process governs how many visits are made.

## The mod\_5 model

mod\_5 <- hurdle(visits ~ hospital + health + chronic +  
 sex + school + insurance,  
 dist = "poisson", zero.dist = "binomial",   
 data = med\_train)  
mod\_5

Call:  
hurdle(formula = visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train, dist = "poisson", zero.dist = "binomial")  
  
Count model coefficients (truncated poisson with log link):  
 (Intercept) hospital healthpoor healthexcellent   
 1.29892 0.16041 0.30243 -0.28116   
 chronic sexfemale school insuranceyes   
 0.09697 0.05611 0.02332 0.09351   
  
Zero hurdle model coefficients (binomial with logit link):  
 (Intercept) hospital healthpoor healthexcellent   
 -0.2998 0.3044 0.1114 -0.3705   
 chronic sexfemale school insuranceyes   
 0.4970 0.3652 0.0637 0.6625

## mod\_5 summary

summary(mod\_5)

Call:  
hurdle(formula = visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train, dist = "poisson", zero.dist = "binomial")  
  
Pearson residuals:  
 Min 1Q Median 3Q Max   
-5.4472 -1.1621 -0.4769 0.5698 24.3403   
  
Count model coefficients (truncated poisson with log link):  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) 1.298920 0.029612 43.865 < 2e-16 \*\*\*  
hospital 0.160410 0.006781 23.656 < 2e-16 \*\*\*  
healthpoor 0.302432 0.020026 15.102 < 2e-16 \*\*\*  
healthexcellent -0.281162 0.035755 -7.864 3.73e-15 \*\*\*  
chronic 0.096971 0.005417 17.901 < 2e-16 \*\*\*  
sexfemale 0.056109 0.014933 3.757 0.000172 \*\*\*  
school 0.023321 0.002138 10.907 < 2e-16 \*\*\*  
insuranceyes 0.093508 0.019789 4.725 2.30e-06 \*\*\*  
Zero hurdle model coefficients (binomial with logit link):  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -0.29978 0.16500 -1.817 0.069240 .   
hospital 0.30445 0.10235 2.975 0.002934 \*\*   
healthpoor 0.11138 0.18993 0.586 0.557605   
healthexcellent -0.37054 0.16200 -2.287 0.022179 \*   
chronic 0.49699 0.05112 9.722 < 2e-16 \*\*\*  
sexfemale 0.36524 0.10116 3.610 0.000306 \*\*\*  
school 0.06370 0.01382 4.611 4.01e-06 \*\*\*  
insuranceyes 0.66251 0.11789 5.620 1.91e-08 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1   
  
Number of iterations in BFGS optimization: 14   
Log-likelihood: -1.222e+04 on 16 Df

## mod\_5 parameters

model\_parameters(mod\_5, ci = 0.90)

# Fixed Effects  
  
Parameter | Log-Mean | SE | 90% CI | z | p  
--------------------------------------------------------------------------  
(Intercept) | 1.30 | 0.03 | [ 1.25, 1.35] | 43.86 | < .001  
hospital | 0.16 | 6.78e-03 | [ 0.15, 0.17] | 23.66 | < .001  
health [poor] | 0.30 | 0.02 | [ 0.27, 0.34] | 15.10 | < .001  
health [excellent] | -0.28 | 0.04 | [-0.34, -0.22] | -7.86 | < .001  
chronic | 0.10 | 5.42e-03 | [ 0.09, 0.11] | 17.90 | < .001  
sex [female] | 0.06 | 0.01 | [ 0.03, 0.08] | 3.76 | < .001  
school | 0.02 | 2.14e-03 | [ 0.02, 0.03] | 10.91 | < .001  
insurance [yes] | 0.09 | 0.02 | [ 0.06, 0.13] | 4.73 | < .001  
  
# Zero-Inflation  
  
Parameter | Log-Odds | SE | 90% CI | z | p  
----------------------------------------------------------------------  
(Intercept) | -0.30 | 0.17 | [-0.57, -0.03] | -1.82 | 0.069   
hospital | 0.30 | 0.10 | [ 0.14, 0.47] | 2.97 | 0.003   
health [poor] | 0.11 | 0.19 | [-0.20, 0.42] | 0.59 | 0.558   
health [excellent] | -0.37 | 0.16 | [-0.64, -0.10] | -2.29 | 0.022   
chronic | 0.50 | 0.05 | [ 0.41, 0.58] | 9.72 | < .001  
sex [female] | 0.37 | 0.10 | [ 0.20, 0.53] | 3.61 | < .001  
school | 0.06 | 0.01 | [ 0.04, 0.09] | 4.61 | < .001  
insurance [yes] | 0.66 | 0.12 | [ 0.47, 0.86] | 5.62 | < .001

The model has a log- or logit-link. Consider using `exponentiate =  
 TRUE` to interpret coefficients as ratios.

## mod\_5 performance summaries

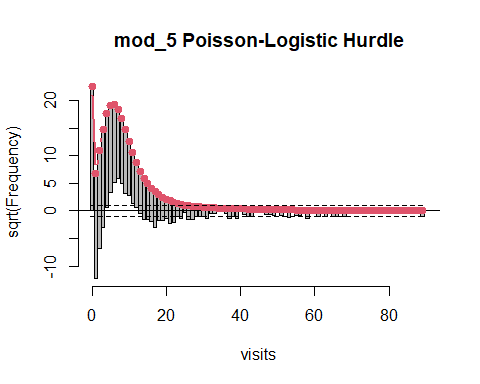
model\_performance(mod\_5)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
---------------------------------------------------------------------  
24468.303 | 24468.468 | 24565.949 | 0.656 | 0.656 | 6.560 | 6.576  
  
AIC | Score\_log | Score\_spherical  
---------------------------------------  
24468.303 | -5.277 | 0.007

* No glance() available.

## Rootogram for Poisson-Logistic Hurdle model

plot(rootogram(mod\_5, plot = FALSE), xlim = c(0, 90),   
 main = "mod\_5 Poisson-Logistic Hurdle")



## Store mod\_5 Predictions

No augment or other broom functions for hurdle models, so …

mod\_5\_aug <- med\_train |>  
 mutate(".fitted" = predict(mod\_5, type = "response"),  
 ".resid" = resid(mod\_5, type = "response"))  
  
mod\_5\_aug |> select(subject, visits, .fitted, .resid) |>  
 head(3)

# A tibble: 3 × 4  
 subject visits .fitted .resid  
 <chr> <dbl> <dbl> <dbl>  
1 355 19 5.32 13.7   
2 2661 3 4.20 -1.20  
3 2895 0 4.59 -4.59

## Training Sample mod\_5 Fit

mod\_5\_summary <-   
 mets(mod\_5\_aug, truth = visits, estimate = .fitted) |>  
 mutate(model = "mod\_5") |> relocate(model)  
mod\_5\_summary |> gt() |> fmt\_number(decimals = 3) |>  
 tab\_options(table.font.size = 20)

| model | .metric | .estimator | .estimate |
| --- | --- | --- | --- |
| mod\_5 | rsq | standard | 0.108 |
| mod\_5 | rmse | standard | 6.560 |
| mod\_5 | mae | standard | 4.164 |

## Training Sample through mod\_5

bind\_rows(mod\_1\_summary, mod\_2\_summary, mod\_3\_summary,   
 mod\_4\_summary, mod\_5\_summary) |>   
 pivot\_wider(names\_from = model,   
 values\_from = .estimate) |>   
 gt() |> fmt\_number(decimals = 3) |>   
 tab\_options(table.font.size = 20)

| .metric | .estimator | mod\_1 | mod\_2 | mod\_3 | mod\_4 | mod\_5 |
| --- | --- | --- | --- | --- | --- | --- |
| rsq | standard | 0.100 | 0.078 | 0.108 | 0.094 | 0.108 |
| rmse | standard | 6.594 | 6.941 | 6.560 | 6.709 | 6.560 |
| mae | standard | 4.189 | 4.252 | 4.164 | 4.191 | 4.164 |

What do you think?

## Are ZIP and Poisson-Logistic Hurdle the Same?

temp\_check <- tibble(  
 subject = mod\_3\_aug$subject,  
 visits = mod\_3\_aug$visits,  
 pred\_zip = mod\_3\_aug$.fitted,  
 pred\_hur = mod\_5\_aug$.fitted,  
 diff = pred\_hur - pred\_zip)  
  
favstats(~ diff, data = temp\_check)

min Q1 median Q3 max mean  
 -0.02412644 -0.0004091297 0.0003033201 0.0009282682 0.03270714 0.0003299622  
 sd n missing  
 0.00304993 3304 0

## Vuong test: mod\_3 vs. mod\_5

vuong(mod\_3, mod\_5)

Vuong Non-Nested Hypothesis Test-Statistic:   
(test-statistic is asymptotically distributed N(0,1) under the  
 null that the models are indistinguishible)  
-------------------------------------------------------------  
 Vuong z-statistic H\_A p-value  
Raw 1.913291 model1 > model2 0.027855  
AIC-corrected 1.913291 model1 > model2 0.027855  
BIC-corrected 1.913291 model1 > model2 0.027855

There’s some evidence mod\_3 (ZIP) fits a bit better than mod\_5 (Hurdle) in our training sample, though the p value (barely) exceeds 0.05.

# mod\_6: Negative Binomial-Logistic Hurdle Model

## Hurdle Model / NB-Logistic

mod\_6 <- hurdle(visits ~ hospital + health + chronic +  
 sex + school + insurance,  
 dist = "negbin", zero.dist = "binomial",   
 data = med\_train)  
mod\_6

Call:  
hurdle(formula = visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train, dist = "negbin", zero.dist = "binomial")  
  
Count model coefficients (truncated negbin with log link):  
 (Intercept) hospital healthpoor healthexcellent   
 1.07299 0.21828 0.36531 -0.30074   
 chronic sexfemale school insuranceyes   
 0.12372 0.05795 0.02429 0.13122   
Theta = 1.406   
  
Zero hurdle model coefficients (binomial with logit link):  
 (Intercept) hospital healthpoor healthexcellent   
 -0.2998 0.3044 0.1114 -0.3705   
 chronic sexfemale school insuranceyes   
 0.4970 0.3652 0.0637 0.6625

## mod\_6 Summary

summary(mod\_6)

Call:  
hurdle(formula = visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train, dist = "negbin", zero.dist = "binomial")  
  
Pearson residuals:  
 Min 1Q Median 3Q Max   
-1.1756 -0.7080 -0.2771 0.3371 17.4791   
  
Count model coefficients (truncated negbin with log link):  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) 1.072988 0.072095 14.883 < 2e-16 \*\*\*  
hospital 0.218281 0.024288 8.987 < 2e-16 \*\*\*  
healthpoor 0.365310 0.054624 6.688 2.27e-11 \*\*\*  
healthexcellent -0.300745 0.076626 -3.925 8.68e-05 \*\*\*  
chronic 0.123717 0.014253 8.680 < 2e-16 \*\*\*  
sexfemale 0.057948 0.037229 1.557 0.11958   
school 0.024290 0.005156 4.711 2.46e-06 \*\*\*  
insuranceyes 0.131216 0.049295 2.662 0.00777 \*\*   
Log(theta) 0.340775 0.049007 6.954 3.56e-12 \*\*\*  
Zero hurdle model coefficients (binomial with logit link):  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -0.29978 0.16500 -1.817 0.069240 .   
hospital 0.30445 0.10235 2.975 0.002934 \*\*   
healthpoor 0.11138 0.18993 0.586 0.557605   
healthexcellent -0.37054 0.16200 -2.287 0.022179 \*   
chronic 0.49699 0.05112 9.722 < 2e-16 \*\*\*  
sexfemale 0.36524 0.10116 3.610 0.000306 \*\*\*  
school 0.06370 0.01382 4.611 4.01e-06 \*\*\*  
insuranceyes 0.66251 0.11789 5.620 1.91e-08 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1   
  
Theta: count = 1.406  
Number of iterations in BFGS optimization: 16   
Log-likelihood: -9102 on 17 Df

## mod\_6 parameters

model\_parameters(mod\_6, ci = 0.90)

# Fixed Effects  
  
Parameter | Log-Mean | SE | 90% CI | z | p  
--------------------------------------------------------------------------  
(Intercept) | 1.07 | 0.07 | [ 0.95, 1.19] | 14.88 | < .001  
hospital | 0.22 | 0.02 | [ 0.18, 0.26] | 8.99 | < .001  
health [poor] | 0.37 | 0.05 | [ 0.28, 0.46] | 6.69 | < .001  
health [excellent] | -0.30 | 0.08 | [-0.43, -0.17] | -3.92 | < .001  
chronic | 0.12 | 0.01 | [ 0.10, 0.15] | 8.68 | < .001  
sex [female] | 0.06 | 0.04 | [ 0.00, 0.12] | 1.56 | 0.120   
school | 0.02 | 5.16e-03 | [ 0.02, 0.03] | 4.71 | < .001  
insurance [yes] | 0.13 | 0.05 | [ 0.05, 0.21] | 2.66 | 0.008   
  
# Zero-Inflation  
  
Parameter | Log-Odds | SE | 90% CI | z | p  
----------------------------------------------------------------------  
(Intercept) | -0.30 | 0.17 | [-0.57, -0.03] | -1.82 | 0.069   
hospital | 0.30 | 0.10 | [ 0.14, 0.47] | 2.97 | 0.003   
health [poor] | 0.11 | 0.19 | [-0.20, 0.42] | 0.59 | 0.558   
health [excellent] | -0.37 | 0.16 | [-0.64, -0.10] | -2.29 | 0.022   
chronic | 0.50 | 0.05 | [ 0.41, 0.58] | 9.72 | < .001  
sex [female] | 0.37 | 0.10 | [ 0.20, 0.53] | 3.61 | < .001  
school | 0.06 | 0.01 | [ 0.04, 0.09] | 4.61 | < .001  
insurance [yes] | 0.66 | 0.12 | [ 0.47, 0.86] | 5.62 | < .001

## mod\_6 performance summaries

model\_performance(mod\_6)

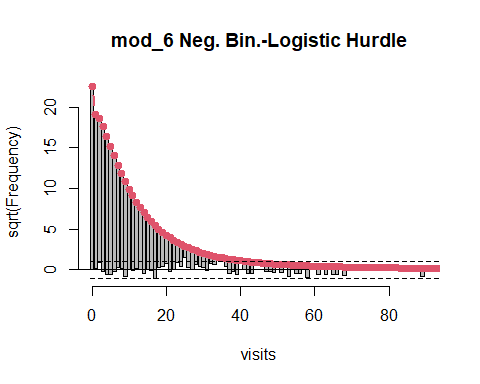
Warning in log(p\_y): NaNs produced

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma | Score\_spherical  
---------------------------------------------------------------------------------------  
18238.228 | 18238.414 | 18341.977 | 0.905 | 0.905 | 6.772 | 6.790 | 0.007

* No glance() available.

## Rootogram for NB-Logistic Hurdle model

plot(rootogram(mod\_6, plot = FALSE), xlim = c(0, 90),   
 main = "mod\_6 Neg. Bin.-Logistic Hurdle")



## Store mod\_6 Predictions

mod\_6\_aug <- med\_train |>  
 mutate(".fitted" = predict(mod\_6, type = "response"),  
 ".resid" = resid(mod\_6, type = "response"))  
  
mod\_6\_aug |> select(subject, visits, .fitted, .resid) |>  
 head(3)

# A tibble: 3 × 4  
 subject visits .fitted .resid  
 <chr> <dbl> <dbl> <dbl>  
1 355 19 5.35 13.6   
2 2661 3 4.16 -1.16  
3 2895 0 4.49 -4.49

## Training Sample mod\_6 Fit

mod\_6\_summary <-   
 mets(mod\_6\_aug, truth = visits, estimate = .fitted) |>  
 mutate(model = "mod\_6") |> relocate(model)  
mod\_6\_summary |>   
 gt() |> fmt\_number(decimals = 3) |>  
 tab\_options(table.font.size = 20)

| model | .metric | .estimator | .estimate |
| --- | --- | --- | --- |
| mod\_6 | rsq | standard | 0.089 |
| mod\_6 | rmse | standard | 6.772 |
| mod\_6 | mae | standard | 4.209 |

## Training Sample through mod\_6

bind\_rows(mod\_1\_summary, mod\_2\_summary, mod\_3\_summary,   
 mod\_4\_summary, mod\_5\_summary, mod\_6\_summary) |>   
 pivot\_wider(names\_from = model, values\_from = .estimate) |>   
 select(-.estimator) |>   
 gt() |> fmt\_number(decimals = 3) |>  
 tab\_options(table.font.size = 20)

| .metric | mod\_1 | mod\_2 | mod\_3 | mod\_4 | mod\_5 | mod\_6 |
| --- | --- | --- | --- | --- | --- | --- |
| rsq | 0.100 | 0.078 | 0.108 | 0.094 | 0.108 | 0.089 |
| rmse | 6.594 | 6.941 | 6.560 | 6.709 | 6.560 | 6.772 |
| mae | 4.189 | 4.252 | 4.164 | 4.191 | 4.164 | 4.209 |

## Vuong test: mod\_4 vs. mod\_6

vuong(mod\_4, mod\_6)

Vuong Non-Nested Hypothesis Test-Statistic:   
(test-statistic is asymptotically distributed N(0,1) under the  
 null that the models are indistinguishible)  
-------------------------------------------------------------  
 Vuong z-statistic H\_A p-value  
Raw 0.0299491 model1 > model2 0.48805  
AIC-corrected 0.0299491 model1 > model2 0.48805  
BIC-corrected 0.0299491 model1 > model2 0.48805

There’s some evidence mod\_4 (ZINB) fits better than mod\_6 (NB Hurdle) in our training sample, but not much, based on the large *p* value.

# Validation including Hurdle Models

## Validation: Test Sample Predictions

Predict the visit counts for each subject in our test sample.

test\_1 <- predict(mod\_1, newdata = med\_test,  
 type.predict = "response")  
test\_2 <- predict(mod\_2, newdata = med\_test,  
 type.predict = "response")  
test\_3 <- predict(mod\_3, newdata = med\_test,  
 type.predict = "response")  
test\_4 <- predict(mod\_4, newdata = med\_test,  
 type.predict = "response")  
test\_5 <- predict(mod\_5, newdata = med\_test,  
 type.predict = "response")  
test\_6 <- predict(mod\_6, newdata = med\_test,  
 type.predict = "response")

## Create a Tibble with Predictions

Combine the various predictions into a tibble with the original data.

test\_res6 <- bind\_cols(med\_test,   
 pre\_m1 = test\_1, pre\_m2 = test\_2,   
 pre\_m3 = test\_3, pre\_m4 = test\_4,   
 pre\_m5 = test\_5, pre\_m6 = test\_6)  
  
names(test\_res6)

[1] "subject" "visits" "hospital" "health" "chronic" "sex"   
 [7] "school" "insurance" "logvisits" "pre\_m1" "pre\_m2" "pre\_m3"   
[13] "pre\_m4" "pre\_m5" "pre\_m6"

## Summarize fit in test sample for each model

m1\_sum <- mets(test\_res6, truth = visits, estimate = pre\_m1) |>  
 mutate(model = "mod\_1")   
m2\_sum <- mets(test\_res6, truth = visits, estimate = pre\_m2) |>  
 mutate(model = "mod\_2")   
m3\_sum <- mets(test\_res6, truth = visits, estimate = pre\_m3) |>  
 mutate(model = "mod\_3")  
m4\_sum <- mets(test\_res6, truth = visits, estimate = pre\_m4) |>  
 mutate(model = "mod\_4")  
m5\_sum <- mets(test\_res6, truth = visits, estimate = pre\_m5) |>  
 mutate(model = "mod\_5")  
m6\_sum <- mets(test\_res6, truth = visits, estimate = pre\_m6) |>  
 mutate(model = "mod\_6")  
  
test\_sum6 <- bind\_rows(m1\_sum, m2\_sum, m3\_sum, m4\_sum,  
 m5\_sum, m6\_sum)

## Validation Results in Test Sample

test\_sum6 <- bind\_rows(m1\_sum, m2\_sum, m3\_sum, m4\_sum,  
 m5\_sum, m6\_sum) |>  
 pivot\_wider(names\_from = model,   
 values\_from = .estimate)  
  
test\_sum6 |>  
 select(-.estimator) |>   
 gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 20)

| .metric | mod\_1 | mod\_2 | mod\_3 | mod\_4 | mod\_5 | mod\_6 |
| --- | --- | --- | --- | --- | --- | --- |
| rsq | 0.1032 | 0.1082 | 0.0993 | 0.0970 | 0.0992 | 0.0937 |
| rmse | 7.2122 | 7.2051 | 5.9069 | 5.9674 | 5.9072 | 6.0009 |
| mae | 4.4550 | 4.4496 | 3.9943 | 4.0095 | 3.9946 | 4.0265 |

* Now which model would you choose?

# Could we fit a linear model for a count outcome?

## Linear Model for our Count Outcome

Let’s fit a **linear regression** (mod\_0: note *log* transformation) to go along with the Poisson regression (mod\_1) we fit last time.

mod\_0 <- lm(log(visits+1) ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train)  
  
mod\_1 <- glm(visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train, family = "poisson")

## Linear Model Coefficients?

## linear model  
tidy(mod\_0) |> gt() |> fmt\_number(decimals = 3) |>   
 tab\_options(table.font.size = 20)

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 0.678 | 0.054 | 12.661 | 0.000 |
| hospital | 0.174 | 0.020 | 8.678 | 0.000 |
| healthpoor | 0.234 | 0.048 | 4.872 | 0.000 |
| healthexcellent | -0.247 | 0.056 | -4.417 | 0.000 |
| chronic | 0.175 | 0.012 | 14.855 | 0.000 |
| sexfemale | 0.113 | 0.030 | 3.761 | 0.000 |
| school | 0.025 | 0.004 | 6.083 | 0.000 |
| insuranceyes | 0.234 | 0.038 | 6.189 | 0.000 |

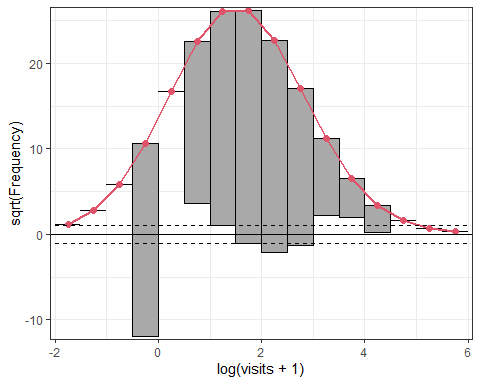
## Poisson Model Coefficients?

## Poisson model  
tidy(mod\_1) |> gt() |> fmt\_number(decimals = 3) |>   
 tab\_options(table.font.size = 20)

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 0.887 | 0.029 | 30.770 | 0.000 |
| hospital | 0.164 | 0.007 | 24.374 | 0.000 |
| healthpoor | 0.310 | 0.020 | 15.294 | 0.000 |
| healthexcellent | -0.359 | 0.035 | -10.287 | 0.000 |
| chronic | 0.137 | 0.005 | 26.082 | 0.000 |
| sexfemale | 0.098 | 0.015 | 6.641 | 0.000 |
| school | 0.031 | 0.002 | 14.808 | 0.000 |
| insuranceyes | 0.200 | 0.019 | 10.278 | 0.000 |

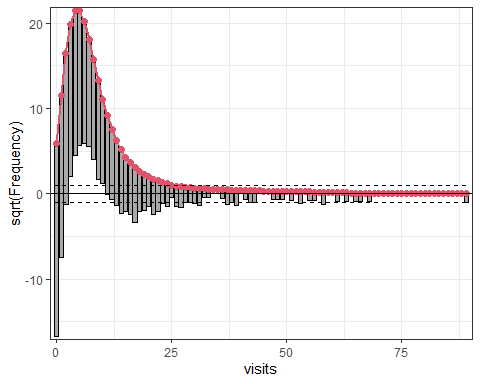
## Rootogram for Linear Model

rootogram(mod\_0)



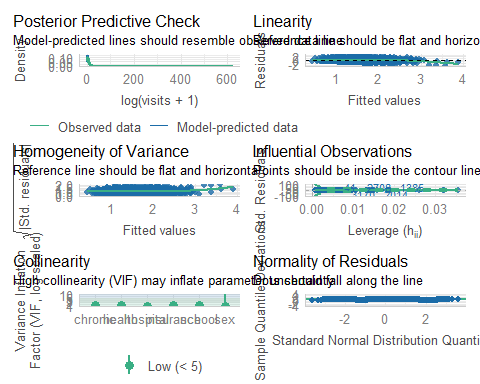
## Rootogram for Poisson Model

rootogram(mod\_1)



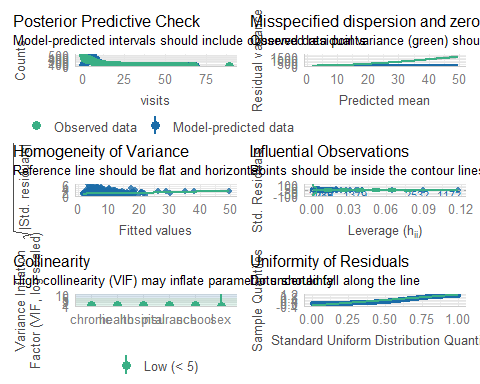
## Linear Regression Assumptions?

check\_model(mod\_0)



## Poisson Regression Plots?

check\_model(mod\_1)



## Test Sample Results (1st 6 subjects)

Actual visits seen in the test sample:

head(med\_test$visits)

[1] 1 16 9 3 0 44

Predicted visits From our linear model (mod\_0):

test\_0 <-   
 exp(predict(mod\_0, newdata = med\_test, type.predict = "response")) - 1  
  
head(test\_0)

1 2 3 4 5 6   
 4.098644 4.730367 2.412072 2.412072 2.412072 10.658053

Predicted visits from our Poisson model (mod\_1):

test\_1 <- predict(mod\_1, newdata = med\_test, type = "response")  
  
head(test\_1)

1 2 3 4 5 6   
 5.885106 6.878921 4.200529 4.200529 4.200529 12.235198

## Test Sample Predictions

No negative predictions with either model.

describe(test\_0) ## predictions from Linear fit

test\_0   
 n missing distinct Info Mean pMedian Gmd .05   
 1102 0 489 1 3.835 3.477 2.037 1.702   
 .10 .25 .50 .75 .90 .95   
 1.972 2.554 3.330 4.365 6.228 8.058   
  
lowest : 0.832359 0.836974 0.969082 1.05604 1.07177   
highest: 15.9109 16.8962 17.5769 24.3302 24.658

describe(test\_1) ## predictions from Poisson fit

test\_1   
 n missing distinct Info Mean pMedian Gmd .05   
 1102 0 489 1 5.71 5.363 2.225 3.270   
 .10 .25 .50 .75 .90 .95   
 3.591 4.334 5.228 6.385 8.429 9.839   
  
lowest : 1.94482 2.10986 2.32785 2.37599 2.40177  
highest: 17.5383 19.1728 19.3223 26.5918 26.5953

## Validation Results: These Two Models

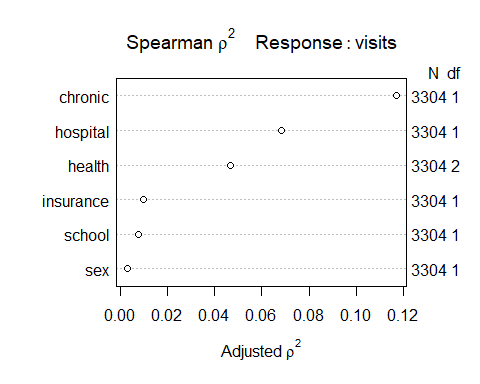
mets <- metric\_set(rsq, rmse, mae)  
test\_res <- bind\_cols(med\_test, pre\_m0 = test\_0, pre\_m1 = test\_1)  
m0\_sum <- mets(test\_res, truth = visits, estimate = pre\_m0) |>  
 mutate(model = "Linear")  
m1\_sum <- mets(test\_res, truth = visits, estimate = pre\_m1) |>  
 mutate(model = "Poisson")   
test\_sum <- bind\_rows(m0\_sum, m1\_sum) |>  
 pivot\_wider(names\_from = model, values\_from = .estimate)  
test\_sum |> select(-.estimator) |>   
 gt() |> fmt\_number(decimals = 3) |>   
 tab\_options(table.font.size = 20)

| .metric | Linear | Poisson |
| --- | --- | --- |
| rsq | 0.100 | 0.095 |
| rmse | 6.125 | 5.915 |
| mae | 3.793 | 4.021 |

# Selecting non-linear terms after Spearman

## Spearman plot

plot(spearman2(visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train))



## Reiterating the Goal

This is the order of the predictors (chronic highest) on the Spearman plot from the previous slide.

| Predictor | Description |
| --- | --- |
| chronic | # of chronic conditions (all values 0-8) |
| hospital | # of hospital stays (all values 0-8) |
| health | self-rated health (poor, average, excellent) |
| insurance | subject (also) has private insurance? (yes/no) |
| school | years of education |
| sex | male or female |

## What might we do?

* chronic is a count (all values 0-8), then a gap to…
* hospital also quantitative, also a count (0-8)
* health is a 3-category factor

We might:

* include a restricted cubic spline with 4-5 knots in chronic
* include a rcs with fewer knots in hospital
* include an interaction between health and chronic or perhaps health and hospital

## Could we build an ols() fit?

Splines sometimes crash with discrete predictors (like counts.)

* For these data, it turns out that even a 3-knot spline in hospital fails (if we already have the four-knot spline in chronic), but the ols() function will let us add both interactions we’re considering.

d <- datadist(medicare); options(datadist = "d")  
  
mod\_toobig <- ols(log(visits + 1) ~   
 rcs(chronic, 4) + hospital \* health +   
 chronic %ia% health +  
 sex + school + insurance, data = med\_train)

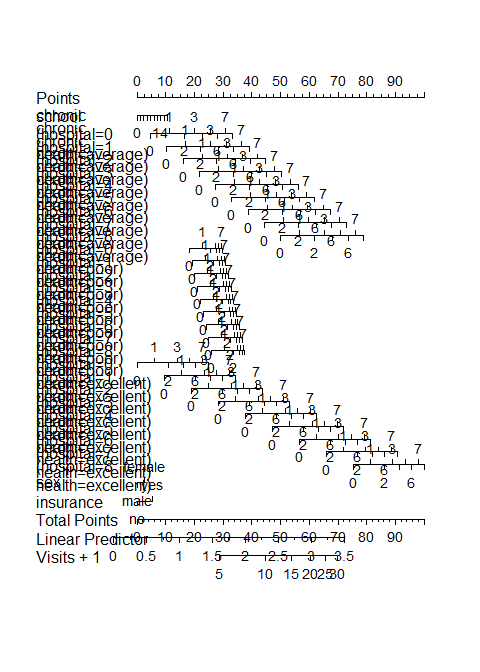
## Why is this model “too big”?

mod\_toobig

Linear Regression Model  
  
ols(formula = log(visits + 1) ~ rcs(chronic, 4) + hospital \*   
 health + chronic %ia% health + sex + school + insurance,   
 data = med\_train)  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 3304 LR chi2 664.03 R2 0.182   
sigma0.8363 d.f. 13 R2 adj 0.179   
d.f. 3290 Pr(> chi2) 0.0000 g 0.444   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-2.42109 -0.55490 0.08359 0.56662 3.07394   
  
 Coef S.E. t Pr(>|t|)  
Intercept 0.5590 0.0575 9.73 <0.0001   
chronic 0.3011 0.0546 5.52 <0.0001   
chronic' -0.2051 0.2579 -0.80 0.4264   
chronic'' 0.2159 0.6311 0.34 0.7323   
hospital 0.2475 0.0249 9.95 <0.0001   
health=poor 0.5914 0.0952 6.21 <0.0001   
health=excellent -0.2022 0.0730 -2.77 0.0057   
chronic \* health=poor -0.0931 0.0335 -2.78 0.0054   
chronic \* health=excellent -0.0213 0.0565 -0.38 0.7058   
sex=female 0.1088 0.0297 3.66 0.0003   
school 0.0257 0.0041 6.20 <0.0001   
insurance=yes 0.2353 0.0375 6.28 <0.0001   
hospital \* health=poor -0.2053 0.0421 -4.88 <0.0001   
hospital \* health=excellent 0.1623 0.1493 1.09 0.2769

## Uh, oh.

plot(nomogram(mod\_toobig, fun = exp, funlabel = "Visits + 1"))



## A more reasonable option?

d <- datadist(medicare); options(datadist = "d")  
  
mod\_new <- ols(log(visits + 1) ~   
 rcs(chronic, 4) + hospital + health +   
 chronic %ia% health +  
 sex + school + insurance, data = med\_train)

## What does this mod\_new show?

mod\_new

Linear Regression Model  
  
ols(formula = log(visits + 1) ~ rcs(chronic, 4) + hospital +   
 health + chronic %ia% health + sex + school + insurance,   
 data = med\_train)  
  
 Model Likelihood Discrimination   
 Ratio Test Indexes   
Obs 3304 LR chi2 637.75 R2 0.176   
sigma0.8393 d.f. 11 R2 adj 0.173   
d.f. 3292 Pr(> chi2) 0.0000 g 0.435   
  
Residuals  
  
 Min 1Q Median 3Q Max   
-3.06891 -0.54950 0.08035 0.56946 3.04632   
  
 Coef S.E. t Pr(>|t|)  
Intercept 0.5743 0.0576 9.97 <0.0001   
chronic 0.3036 0.0548 5.55 <0.0001   
chronic' -0.1710 0.2588 -0.66 0.5089   
chronic'' 0.1165 0.6331 0.18 0.8540   
hospital 0.1799 0.0199 9.02 <0.0001   
health=poor 0.5437 0.0951 5.72 <0.0001   
health=excellent -0.1940 0.0724 -2.68 0.0074   
chronic \* health=poor -0.1199 0.0331 -3.62 0.0003   
chronic \* health=excellent -0.0163 0.0563 -0.29 0.7718   
sex=female 0.1051 0.0298 3.53 0.0004   
school 0.0256 0.0042 6.16 <0.0001   
insurance=yes 0.2307 0.0376 6.14 <0.0001

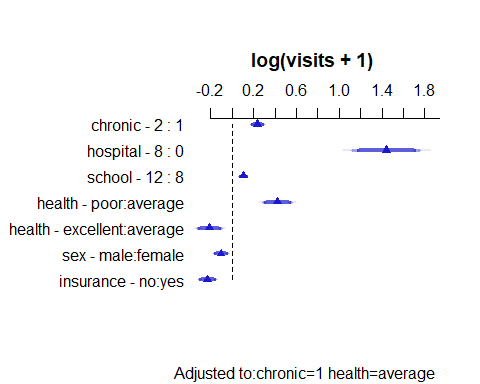
## How many df did we add here?

anova(mod\_new)

Analysis of Variance Response: log(visits + 1)   
  
 Factor d.f. Partial SS MS   
 chronic (Factor+Higher Order Factors) 5 189.349521 37.8699042  
 All Interactions 2 9.251314 4.6256570  
 Nonlinear 2 9.483346 4.7416730  
 hospital 1 57.342322 57.3423218  
 health (Factor+Higher Order Factors) 4 39.675076 9.9187689  
 All Interactions 2 9.251314 4.6256570  
 chronic \* health (Factor+Higher Order Factors) 2 9.251314 4.6256570  
 sex 1 8.763370 8.7633703  
 school 1 26.694549 26.6945488  
 insurance 1 26.545793 26.5457929  
 TOTAL NONLINEAR + INTERACTION 4 31.942728 7.9856821  
 REGRESSION 11 493.787139 44.8897399  
 ERROR 3292 2319.211257 0.7044992  
 F P   
 53.75 <.0001  
 6.57 0.0014  
 6.73 0.0012  
 81.39 <.0001  
 14.08 <.0001  
 6.57 0.0014  
 6.57 0.0014  
 12.44 0.0004  
 37.89 <.0001  
 37.68 <.0001  
 11.34 <.0001  
 63.72 <.0001

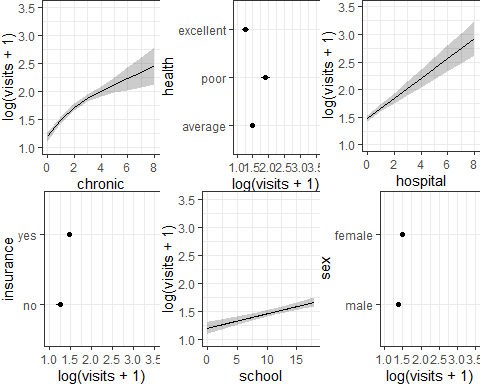
## What does this ols() fit look like?

plot(summary(mod\_new))



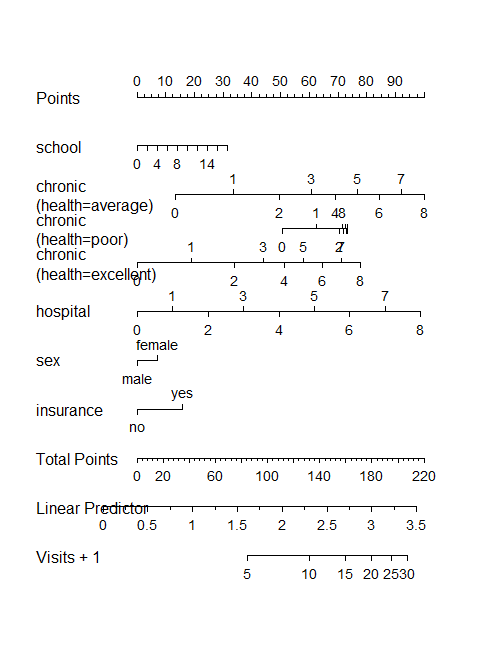
## What does this ols() fit look like?

ggplot(Predict(mod\_new))



## How’s the nomogram?

plot(nomogram(mod\_new, fun = exp, funlabel = "Visits + 1"))



# Can we fit a Poisson model with a function from rms?

## The Glm() function in rms

d <- datadist(medicare); options(datadist = "d")  
  
mod\_1\_Glm <- Glm(visits ~ hospital + health + chronic + sex + school +   
 insurance, data = med\_train, family = poisson())

and we could have used rcs() or polynomials or interactions if we wanted to do so.

Complete and updated documentation for the rms package is found at <https://hbiostat.org/r/rms/>.

### Does a Glm() fit do everything we are used to?

* Nope. No validate() or calibrate() methods exist.

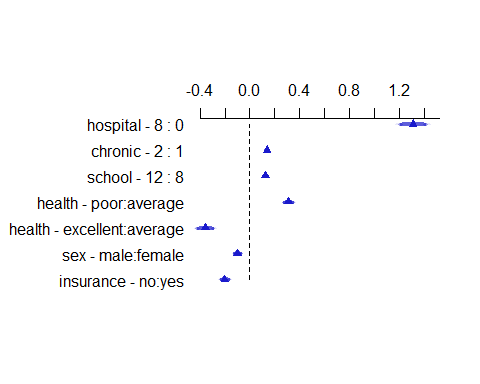
## What’s in mod\_1\_Glm?

mod\_1\_Glm

General Linear Model  
  
Glm(formula = visits ~ hospital + health + chronic + sex + school +   
 insurance, family = poisson(), data = med\_train)  
  
 Model Likelihood   
 Ratio Test   
 Obs3304 LR chi2 3019.53   
Residual d.f.3296 d.f. 7   
 g 0.386 Pr(> chi2) <0.0001   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept 0.8866 0.0288 30.77 <0.0001   
hospital 0.1636 0.0067 24.37 <0.0001   
health=poor 0.3096 0.0202 15.29 <0.0001   
health=excellent -0.3588 0.0349 -10.29 <0.0001   
chronic 0.1373 0.0053 26.08 <0.0001   
sex=female 0.0983 0.0148 6.64 <0.0001   
school 0.0313 0.0021 14.81 <0.0001   
insurance=yes 0.2002 0.0195 10.28 <0.0001

## What can we do: mod\_1\_Glm?

plot(summary(mod\_1\_Glm))



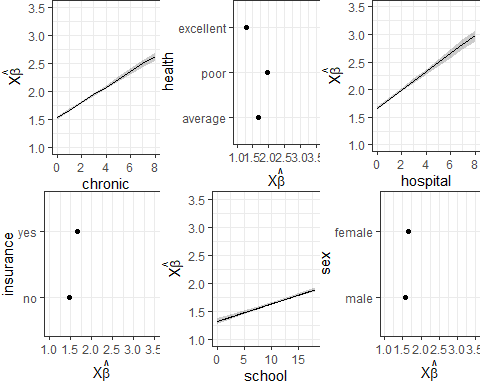
## What can we do: mod\_1\_Glm?

summary(mod\_1\_Glm)

Effects Response : visits   
  
 Factor Low High Diff. Effect S.E. Lower 0.95  
 hospital 0 8 8 1.308400 0.0536810 1.20320   
 chronic 1 2 1 0.137350 0.0052661 0.12702   
 school 8 12 4 0.125030 0.0084433 0.10848   
 health - poor:average 1 2 NA 0.309610 0.0202440 0.26992   
 health - excellent:average 1 3 NA -0.358760 0.0348750 -0.42714   
 sex - male:female 2 1 NA -0.098325 0.0148050 -0.12735   
 insurance - no:yes 2 1 NA -0.200250 0.0194840 -0.23845   
 Upper 0.95  
 1.413700   
 0.147670   
 0.141590   
 0.349300   
 -0.290380   
 -0.069297   
 -0.162050

## What can we do: mod\_1\_Glm?

ggplot(Predict(mod\_1\_Glm))



plot(nomogram(mod\_1\_Glm, fun = exp, funlabel = "Visits",  
 fun.at = c(1, 2, 3, 4, 5, 10, 15, 20, 25, 30)))

