# Chapter 11.4–12

DJM

27 March 2018

## Generalized linear models

- Think back to the beginning of this class: we set out to model the **regression function**

$$\mu(x) = \mathbb{E}\left[Y \mid X = x\right]$$

- We then showed how Ordinary Least Squares sets

$$\mu(x) = \mathbb{E}\left[Y \mid X = x\right] = \beta_0 + \beta^\top x$$

- Generalized linear models "generalize" this idea

## Transforming the response

- A generalized linear model starts by trying to **transform** the response $Y$
- You've done this before to handle skewed distributions or other issues, but let's push through some math
- Suppose we transform to $g(Y)$
- Now, follow me along without justification (because I know where this goes), take a Taylor expansion to one term around $\mu(x)$:

$$g(Y) \approx g(\mu(x)) + (Y - \mu(x))g'(\mu(x))$$

- Now we've written $g(Y)$ in terms of our regression function $\mu(x) = \mathbb{E}\left[Y \mid X = x\right]$
- Let's make a new random variable $Z = g(\mu(x)) + (Y - \mu(x))g'(\mu(x))$

## Let's generalize

$$g(Y) \approx g(\mu(x)) + (Y - \mu(x))g'(\mu(x)) =: Z$$

- Let $g$ be any function and define

$$\begin{aligned}\mu(x) &= \mathbb{E}\left[Y \mid X = x\right]\\ \eta(x) &= g(\mu(x))\\ \epsilon(x) &= Y - \mu(x)\end{aligned}$$

- Rather than assume $\mu(x) = \beta_0 + \beta^\top x$, we assume

$$\eta(x) = g(\mu(x)) = \beta_0 + \beta^\top x$$

- Terms

- $\eta$ is the **linear predictor**
- $g$ is called the **link function**

- Other bits
  - $\epsilon(x) = Y - \mu(x)$ has mean 0 conditional on $X = x$
  - $\mathbb{E}[Z \mid X = x] = \mathbb{E}[g(\mu(x))] + 0 = g(\mu(x))$
  - $\mathbb{V}[Z \mid X = x] = \mathbb{V}[\eta(x) \mid X = x] + \mathbb{V}[(Y - \mu(x))g'(\mu(x)) \mid X = x] = 0 + (g'(\mu(x)))^2 \mathbb{V}[Y \mid X = x]$

## Why do this?

- If $Y$ is binary (is either 0 or 1), than transforming $g(Y) = \log \frac{Y}{1-Y}$ doesn't help
- For that $g$, $g(Y) = \pm\infty$.
- So, often, if we just look at $g(Y)$, we have issues
- Instead we look at the Taylor expansion because it doesn't depend on $g(Y)$

## Example: OLS

$$g(Y) \approx g(\mu(x)) + (Y - \mu(x))g'(\mu(x)) =: Z$$

$$\begin{aligned}
\mu(x) &= \mathbb{E}[Y \mid X = x] \\
g(m) &= m \\
g^{-1}(m) &= m \\
g(\mu(x)) &= \mathbb{E}[Y \mid X = x] \\
\mathbb{V}[Y \mid X = x] &= \sigma^2 \\
(g'(m))^2 &= 1
\end{aligned}$$

## Example: logistic regression

$$g(Y) \approx g(\mu(x)) + (Y - \mu(x))g'(\mu(x)) =: Z$$

$$\begin{aligned}
\mu(x) &= \mathbb{E}[Y \mid X = x] = P(Y = 1 \mid X = x) \\
g(m) &= \log \frac{m}{1-m} \\
g^{-1}(m) &= \frac{\exp(m)}{1 + \exp(m)} \\
g(\mu(x)) &= \log \frac{P(Y = 1 \mid X = x)}{1 - P(Y = 1 \mid X = x)} \\
\mathbb{V}[Y \mid X = x] &= \mu(x)(1 - \mu(x)) \\
(g'(m))^2 &= \left(\frac{d}{dm} \log \frac{m}{1-m}\right)^2 = \left(\frac{1}{m(1-m)}\right)^2
\end{aligned}$$

## Estimation

1. Get some data $(y_1, x_1), \ldots, (y_n, x_n)$, figure out the **link function** $g$, and calculate $g^{-1}$, $g'$ and $\mathbb{V}[Y \mid X = x]$. Now give some initial guesses for $\beta$ (say $\beta = \mathbf{0}$)
2. Iterate until convergence:
   a. Calculate $\eta(x_i) = \beta^\top x_i$ and $\hat{\mu}(x_i) = g^{-1}(\eta(x_i))$.
   b. Find $z_i = \eta(x_i) + (y_i - \hat{\mu}(x_i))g'(\hat{\mu}(x_i))$.
   c. Calculate the weights $w_i = [(g'(\hat{\mu}(x_i)))^2 \mathbb{V}[\hat{\mu}(x_i)]$.
   d. Do weighted least squares of $z_i$ on $x_i$ with weights $w_i$. This gives a new $\beta$.

## R code

```r
irwls <- function(y, x, # input data, first column is intercept if desired
                  invlink=function(m) m, # g^{-1}, defaults to "identity" (lm)
                  linkPrime = function(m) rep(1,length(m)), # g', defaults to "identity"
                  V = function(m) 1, # Variance function, defaults to "identity"
                  maxit = 100, tol=1e-6) # control parameters
{
  n = length(y)
  x = as.matrix(x) # make sure this is a matrix
  p = ncol(x)
  beta = double(p) # initialize coefficients
  conv = FALSE # hasn't converged
  iter = 1 # first iteration
  while(!conv && (iter<maxit)){ # check loops
    iter = iter + 1 # update first thing (so as not to forget)
    eta = x %*% beta # eta
    mu = invlink(eta) # mu
    gp = linkPrime(mu) # evaluate g'(mu)
    z = eta + (y - mu) * gp # effective transformed response
    w = gp^2 * V(mu) # variance parameter
    betaNew = coef(lm(z~x-1, weights=1/w)) # do the regression
    conv = (mean((beta-betaNew)^2)<tol) # check if the betas are "moving"
    beta = betaNew # update betas
  }
  return(beta)
}
```

## Testing, testing…

```r
set.seed(04042017)
n = 100
b = c(2,-2)
b0 = 0
x = matrix(runif(n*2,-1,1), nrow=n)
y.lm = b0 + x %*% b + rnorm(n)
c(b0,b)
```

```
## [1]  0  2 -2
```

```r
coef(lm(y.lm~x))
```

```
## (Intercept)          x1          x2
## -0.01195974  2.04150931 -1.80045591
```

```r
irwls(y.lm,cbind(1,x))
```

```
##          x1          x2          x3
## -0.01195974  2.04150931 -1.80045591
```

### Testing, testing, testing…

```r
ilogit <- function(z) exp(z)/(1+exp(z))
y.logit = rbinom(n, 1, prob = ilogit(b0 + x %*% b))
(true.logit <- c(b0,b))
```

```
## [1]  0  2 -2
```

```r
(glm.logit <- coef(glm(y.logit~x,family = 'binomial')))
```

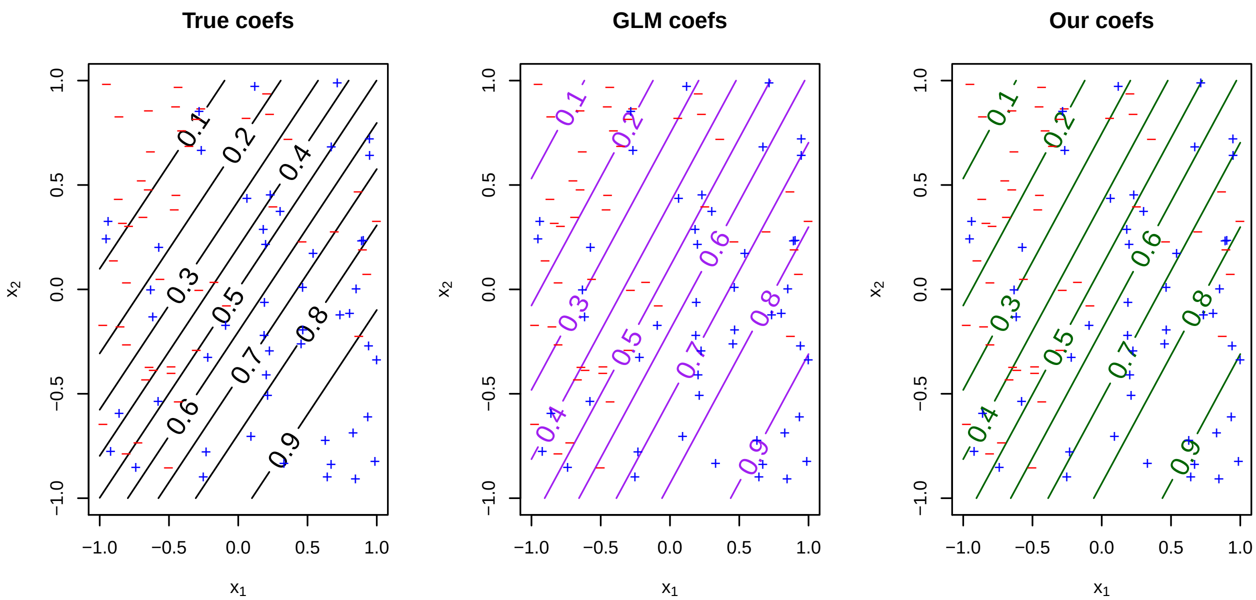
```
## (Intercept)          x1          x2
##   0.1473591   1.6369813  -1.3332978
```

```r
(our.logit <- irwls(y.logit, cbind(1,x), invlink=ilogit,
                    linkPrime=function(m) 1/(m*(1-m)), V=function(m) m*(1-m)))
```

```
##         x1         x2         x3
##  0.1473591  1.6369812 -1.3332977
```

### Visualizing

```r
plot.logistic <- function(x, y, coeff, n.grid=50, labcex=1, col="purple", ...) {
  grid.seq <- seq(from=-1,to=1,length.out=n.grid)
  plot.grid <- as.matrix(expand.grid(grid.seq,grid.seq))
  p <- matrix(ilogit(coeff[1] + (plot.grid %*% coeff[2:3] )),nrow=n.grid)
  contour(x=grid.seq,y=grid.seq,z=p, xlab=expression(x[1]), zlim=c(0,1),
          ylab=expression(x[2]),labcex=labcex,col=col,...)
  points(x[,1],x[,2],pch=ifelse(y==1,"+","-"),col=ifelse(y==1,"blue","red"))
}
par(mfrow=c(1,3))
plot.logistic(x,y.logit,true.logit,main='True coefs',col=1)
plot.logistic(x,y.logit,glm.logit,main='GLM coefs')
plot.logistic(x,y.logit,our.logit,main='Our coefs',col='darkgreen')
```




## O-logit

- A study looks at factors that influence the decision of whether to apply to graduate school.
- College juniors are asked if they are unlikely, somewhat likely, or very likely to apply to graduate school. Hence, our outcome variable has three categories.
- Data on parental educational status, whether the undergraduate institution is public or private, and current GPA is also collected.
- The researchers have reason to believe that the "distances" between these three points are not equal.
- For example, the "distance" between "unlikely" and "somewhat likely" may be shorter than the distance between "somewhat likely" and "very likely".

## Ordinal logistic regression

```
load('gfx/ologit.Rdata')
lapply(dat[, c("apply", "pared", "public")], table)
```

```
## $apply
##
##        unlikely somewhat likely     very likely
##             220             140              40
##
## $pared
##
##   No grad degree One+ grad degree
##              337               63
##
## $public
##
## Yes  No
## 343  57
```

```r
ftable(xtabs(~ public + apply + pared, data = dat)) # what does this information tell you?
```

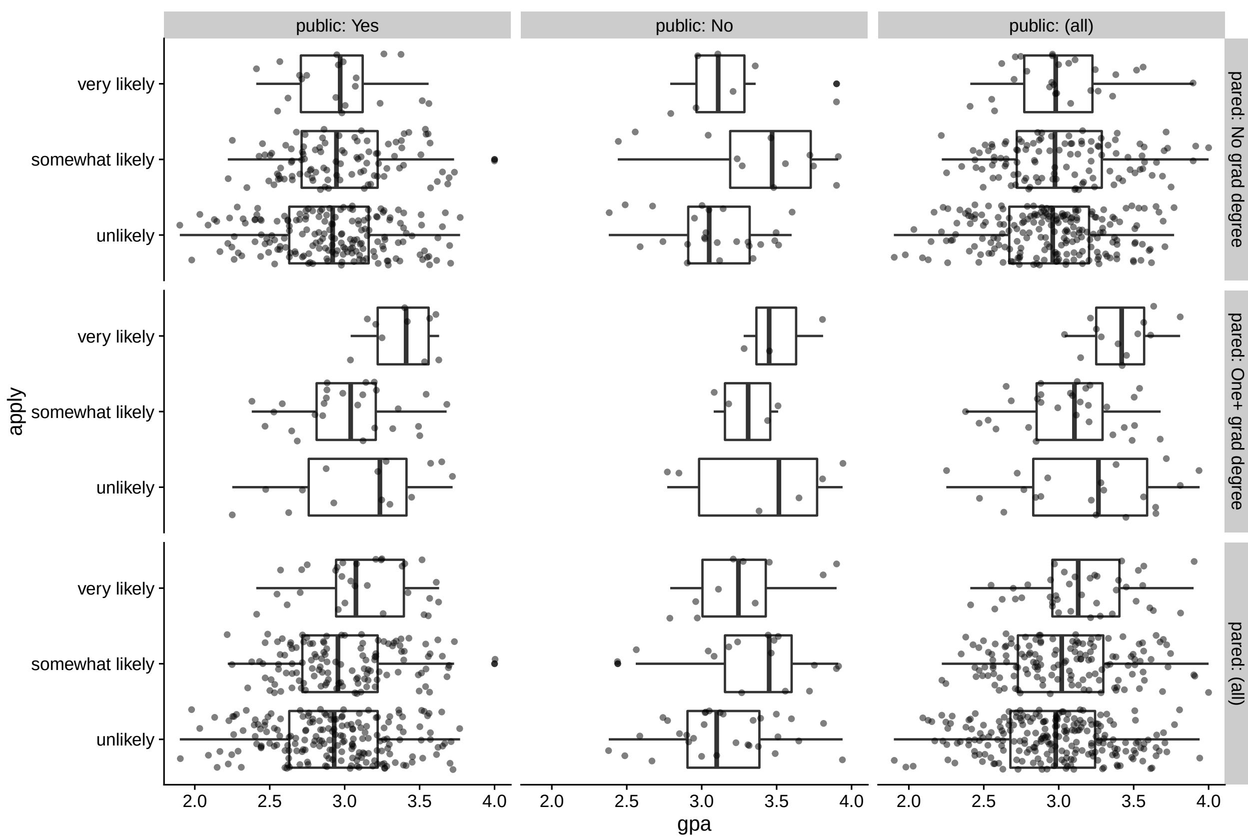
```
##                        pared No grad degree One+ grad degree
## public apply
## Yes    unlikely                         175               14
##        somewhat likely                   98               26
##        very likely                       20               10
## No     unlikely                          25                6
##        somewhat likely                   12                4
##        very likely                        7                3
```

```r
c(summary(dat$gpa), sd=sd(dat$gpa))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.        sd
## 1.9000000 2.7200000 2.9900000 2.9989250 3.2700000 4.0000000 0.3979409
```

## Plotting

```r
library(cowplot)
ggplot(dat, aes(x = apply, y = gpa)) +
  geom_boxplot(size = .75) +
  geom_jitter(alpha = .5) +
  facet_grid(pared ~ public, margins = TRUE, labeller = label_both) + theme_cowplot() +
  coord_flip()
```



## Do the estimation

```
library(MASS) # this one is already installed
m <- polr(apply ~ pared + public + gpa, data = dat, Hess=TRUE)
summary(m)
```

```
## Call:
## polr(formula = apply ~ pared + public + gpa, data = dat, Hess = TRUE)
##
## Coefficients:
##                          Value Std. Error t value
## paredOne+ grad degree  1.04769     0.2658  3.9418
## publicNo              -0.05879     0.2979 -0.1974
## gpa                    0.61594     0.2606  2.3632
##
## Intercepts:
##                             Value   Std. Error t value
## unlikely|somewhat likely     2.2039  0.7795     2.8272
## somewhat likely|very likely  4.2994  0.8043     5.3453
##
## Residual Deviance: 717.0249
## AIC: 727.0249
```

- There are estimates for two intercepts, which are sometimes called cutpoints.
- The intercepts indicate where the latent variable is cut to make the three groups that we observe in our data. Note that this latent variable is continuous.
- note that there are no p-values in the summary output
- what R code would you use to calculate them?

```
t.values = summary(m)$coef[1:3,3]
signif(2*pt(abs(t.values), df.residual(m), lower.tail = FALSE),3)
```

```
## paredOne+ grad degree              publicNo                   gpa
##              9.56e-05              8.44e-01              1.86e-02
```

## Confidence intervals

```
confint(m) # this does "profiled" confidence intervals using numerical information from polr
```

```
##                            2.5 %    97.5 %
## paredOne+ grad degree  0.5281768 1.5721750
## publicNo              -0.6522060 0.5191384
## gpa                    0.1076202 1.1309148
```

```
confint.default(m) # this assumes normality, are they very different? (the first is preferred)
```

```
##                            2.5 %    97.5 %
## paredOne+ grad degree  0.5267524 1.5686278
## publicNo              -0.6425833 0.5250119
## gpa                    0.1051074 1.1267737
```

- Interpret the coefficient for `pared` in context (this is a logistic regression, it uses log odds)

### Odds ratios

```r
exp(coef(m))
```

```
## paredOne+ grad degree              publicNo                   gpa
##             2.8510579             0.9429088             1.8513972
```

```r
exp(cbind(odds.ratio=coef(m), confint(m))) # how would you interpret these?
```

```
##                       odds.ratio     2.5 %   97.5 %
## paredOne+ grad degree  2.8510579 1.6958376 4.817114
## publicNo               0.9429088 0.5208954 1.680579
## gpa                    1.8513972 1.1136247 3.098490
```

- Called **proportional odds ratios**
- We would say that for a one unit increase in parental education, i.e., going from 0 (Low) to 1 (High), the odds of "very likely" applying versus "somewhat likely" or "unlikely" applying combined are 2.85 greater, given that all of the other variables in the model are held constant.
- When a student's gpa moves 1 unit, the odds of moving from "unlikely" applying to "somewhat likely" or "very likley" applying (or from the lower and middle categories to the high category) are multiplied by 1.85.

### Assumptions

- The relationship between each pair of outcome groups is the same
- I.e., the coefficients that describe the relationship between, say, the lowest versus all higher categories of the response variable are the same as those that describe the relationship between the next lowest category and all higher categories, etc.
- Called the "proportional odds assumption" or the "parallel regression assumption."
- Because the relationship between all pairs of groups is the same, there is only one set of coefficients.
- If this were not the case, we would need different sets of coefficients in the model to describe the relationship between each pair of outcome groups.
- To asses the appropriateness of our model, we need to evaluate whether the proportional odds assumption is tenable.

### Use methods from before

```r
require(nnet)
full = multinom(apply~pared+public+gpa,
                data=dat, trace=FALSE) # This fits the unconstrained model (not ordinal)
# The problem is that it doesn't play well with other things (like anova)
# It also likes to print out garbage (trace=FALSE makes it go away)
simulate.from.ologit <- function(df, mdl) {
  probs <- predict(mdl, type="prob") # changed from ch 10 code
  newy <- factor(apply(probs, 1, function(x) sample(colnames(probs), 1, prob=x)))
  df[[names(mdl$model)[1]]] <- newy
  return(df)
}
```

```
# Simulate from an estimated ordinal logistic model, and refit both the ordered logistic
  # regression and a multinomial logit
# Inputs: data frame with covariates (df), fitted ologistic model (logr)
# Output: difference in deviances
delta.deviance.sim <- function (df, logr) {
  sim.df <- simulate.from.ologit(df, logr)
  form = formula(logr)[1:3] # not sure if 1:3 will always work??
  ologit.dev <- polr(form, data=sim.df)$deviance
  multi.dev <- multinom(form, data=sim.df,trace=FALSE)$deviance
  return(ologit.dev - multi.dev)
}
```

### Results

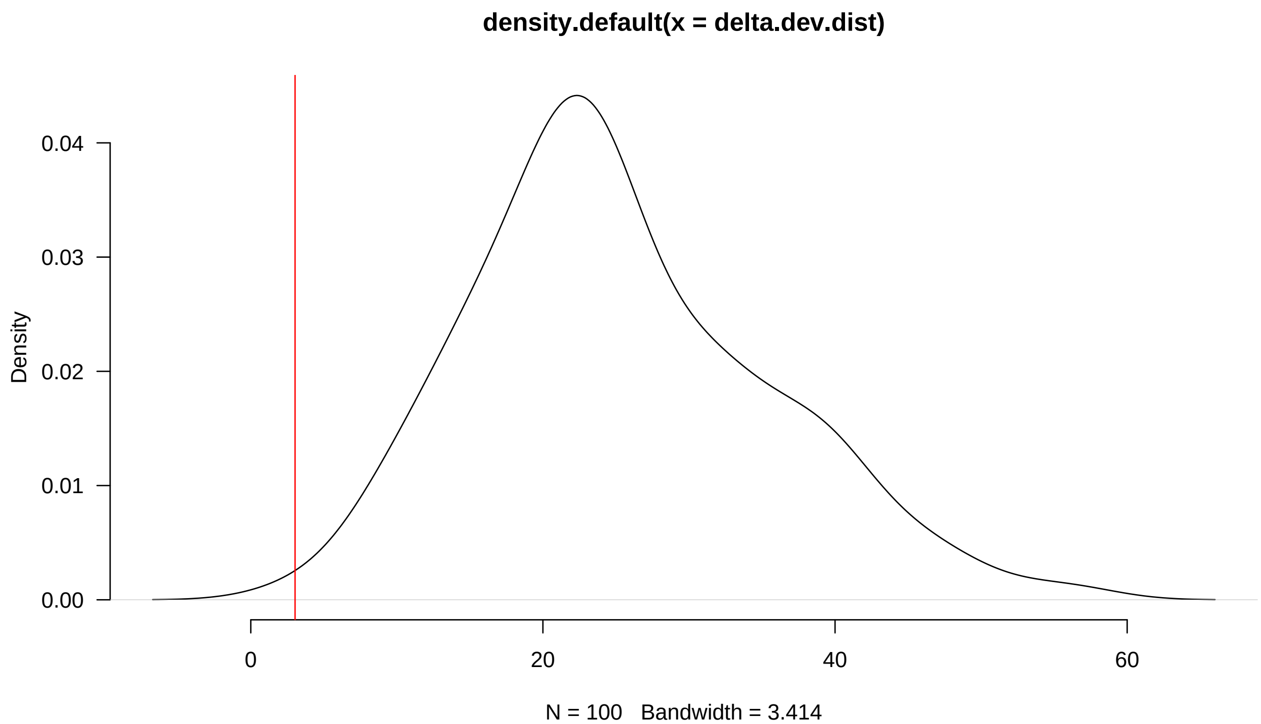
```
(delta.observed = m$deviance - full$deviance)
```

```
## [1] 3.030909
```

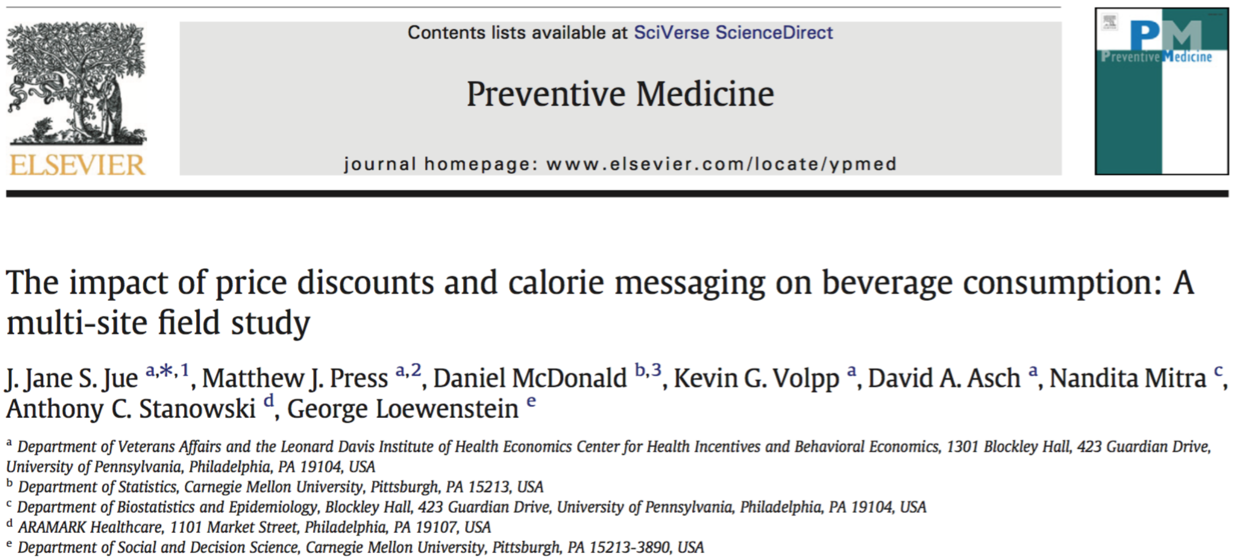
```
delta.dev.dist = replicate(100, delta.deviance.sim(dat, m))
mean(delta.observed <= delta.dev.dist) # % of sims that fit better than what we saw
```

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

```
plot(density(delta.dev.dist), las=1, bty='n')
abline(v=delta.observed, col=2)
```



## Poisson regression



Contents lists available at SciVerse ScienceDirect

Preventive Medicine

journal homepage: www.elsevier.com/locate/ypmed

The impact of price discounts and calorie messaging on beverage consumption: A multi-site field study

J. Jane S. Jue [a,*,1], Matthew J. Press [a,2], Daniel McDonald [b,3], Kevin G. Volpp [a], David A. Asch [a], Nandita Mitra [c], Anthony C. Stanowski [d], George Loewenstein [e]

[a] Department of Veterans Affairs and the Leonard Davis Institute of Health Economics Center for Health Incentives and Behavioral Economics, 1301 Blockley Hall, 423 Guardian Drive, University of Pennsylvania, Philadelphia, PA 19104, USA
[b] Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[c] Department of Biostatistics and Epidemiology, Blockley Hall, 423 Guardian Drive, University of Pennsylvania, Philadelphia, PA 19104, USA
[d] ARAMARK Healthcare, 1101 Market Street, Philadelphia, PA 19107, USA
[e] Department of Social and Decision Science, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA

## The study

- Examining how many sugary drinks were sold in hospital cafeterias
- Looked at 0 different hospitals
- Some sites had different cafeterias in them
- Recorded number of sugary beverages and number of total customers daily
- Each hospital had an intervention partway through:
  1. Posters saying that there were 200 calories in that yucky drink.
  2. Posters saying that it would take 1 hour of exercise to work off
  3. A 10% discount on zero calorie beverages
  4. Some combinations

## Data processing

```
soda = soda %>% mutate(
  DofW = factor(DofW, labels = c('Mon','Tue','Wed','Thur','Fri','Sat','Sun')),
  cont = fct_collapse(factor(Intervention), control = c('follow','preint')) %>%
                      fct_relevel('control') # makes 'control' the baseline
)
soda = filter(soda, DofW %in% c('Mon','Tue','Wed','Thur','Fri')) # remove weekends
```

## The model

- Want to know whether the interventions changed purchases of (a) **sugary drinks** and/or (b) **zero calorie drinks**.

- Control for day of the week effects
- Estimate for each hospital separately
- Use log of total customers as an 'offset': means that the coefficient is known to be 1
- Basic idea is that we are modelling $\log(bev/total) = \log(bev) - \log(total)$
- Use Poisson regression: good for count or percentage data

## As a glm

- The poisson distribution

$$p(y \mid \lambda) = \frac{\lambda^y \exp(-\lambda)}{y!} I(y \in \mathbb{Z}^+)$$

- $\mathbb{E}[Y] = \lambda$
- Take $\mu(x) = \mathbb{E}[Y \mid X = x] = \lambda$
- So redefine $\lambda = \lambda(x) = x^\top \beta$

$$p(y \mid X = x) = \frac{\lambda(x)^y \exp(-\lambda(x))}{y!} I(y \in \mathbb{Z}^+) = \frac{(x^\top \beta)^y \exp(-x^\top \beta)}{y!} I(y \in \mathbb{Z}^+)$$

- The right choice of $g$ turns out to be log
- The variance of Poisson is also $\lambda$
- Therefore, when we run the glm, we always use "overdispersion". This lets the variance scale differently from the mean (in OLS, the mean was $x^\top \beta$ and the variance was $\sigma^2$)

## Estimating the thing

- Do it separately for each site

```
sugary = soda %>% group_by(Site) %>%
  do(mod=glm(Regular~DofW + cont + offset(log(CosTot)), family='quasipoisson', data=.))
zerocal = soda %>% group_by(Site) %>%
  do(mod=glm(ZeroCal~DofW + cont + offset(log(CosTot)), family='quasipoisson', data=.))
sugary_coef = sapply(sugary$mod, coef)
colnames(sugary_coef) = sugary$Site
sugary_coef = as_tibble(sugary_coef, rownames='coef')
zerocal_coef = sapply(zerocal$mod, coef)
colnames(zerocal_coef) = zerocal$Site
zerocal_coef = as_tibble(zerocal_coef, rownames='coef')
kable(sugary_coef, digits = 2)
```

| coef | chop | HF | NS |
|---|---|---|---|
| (Intercept) | -1.11 | -0.66 | -1.00 |
| DofWTue | -0.02 | 0.05 | 0.02 |
| DofWWed | -0.01 | -0.03 | 0.03 |
| DofWThur | -0.02 | 0.01 | -0.02 |
| DofWFri | 0.01 | 0.02 | 0.06 |
| contboth | 0.00 | 0.19 | -0.11 |
| contcal | 0.02 | 0.07 | -0.04 |

| coef | chop | HF | NS |
|---|---|---|---|
| contdis | 0.05 | -0.23 | -0.03 |
| contdismes | 0.03 | -0.20 | -0.04 |
| contexcer | -0.02 | 0.08 | 0.00 |

```
kable(zerocal_coef, digits = 2)
```

| coef | chop | HF | NS |
|---|---|---|---|
| (Intercept) | -0.78 | -0.79 | -0.70 |
| DofWTue | 0.01 | -0.05 | -0.02 |
| DofWWed | -0.02 | 0.00 | -0.03 |
| DofWThur | 0.00 | -0.01 | -0.01 |
| DofWFri | -0.04 | -0.02 | -0.08 |
| contboth | 0.01 | -0.28 | 0.05 |
| contcal | -0.03 | -0.09 | 0.01 |
| contdis | 0.00 | 0.20 | 0.04 |
| contdismes | 0.00 | 0.18 | 0.03 |
| contexcer | 0.00 | -0.17 | -0.01 |

### Transformed back (as percentage changes)

```
to_perc <- function(x) (exp(x)-1)*100
mutate_if(sugary_coef,is.numeric,to_perc) %>% kable(digits=2)
```

| coef | chop | HF | NS |
|---|---|---|---|
| (Intercept) | -66.92 | -48.44 | -63.25 |
| DofWTue | -1.56 | 4.61 | 1.63 |
| DofWWed | -0.71 | -2.60 | 2.95 |
| DofWThur | -2.45 | 0.79 | -2.27 |
| DofWFri | 0.65 | 1.66 | 5.81 |
| contboth | -0.30 | 21.03 | -10.74 |
| contcal | 2.01 | 7.26 | -4.11 |
| contdis | 4.79 | -20.67 | -3.20 |
| contdismes | 2.83 | -17.93 | -3.99 |
| contexcer | -1.49 | 7.80 | 0.18 |

```
mutate_if(zerocal_coef,is.numeric,to_perc) %>% kable(digits=2)
```

| coef | chop | HF | NS |
|---|---|---|---|
| (Intercept) | -54.08 | -54.64 | -50.12 |
| DofWTue | 0.58 | -5.11 | -1.98 |
| DofWWed | -1.86 | 0.19 | -2.92 |
| DofWThur | -0.14 | -1.46 | -1.14 |
| DofWFri | -4.13 | -1.83 | -7.87 |
| contboth | 0.80 | -24.72 | 5.44 |
| contcal | -2.88 | -8.50 | 0.52 |
| contdis | -0.49 | 22.74 | 4.05 |
| contdismes | -0.30 | 20.11 | 2.89 |
| contexcer | 0.47 | -16.02 | -0.97 |

## Numbers are bad, make plots

```
df = bind_rows("sugary" = sugary_coef, "zerocal" = zerocal_coef, .id='drink_type') %>%
  pivot_longer(chop:NS) %>% arrange(name)
ncoef = nrow(sugary_coef)
df$low = c(rbind(sapply(sugary$mod, confint)[1:ncoef,],
            sapply(zerocal$mod,confint)[1:ncoef,]))
df$high = c(rbind(sapply(sugary$mod,confint)[(ncoef+1):(2*ncoef),],
             sapply(zerocal$mod,confint)[(ncoef+1):(2*ncoef),]))
df = filter(df, grepl('cont',coef)) %>% mutate(coef = substring(coef, 5)) %>%
  mutate_if(is.numeric, to_perc)
ggplot(df, aes(coef, value, color=drink_type)) +
  geom_errorbar(aes(ymax=high,ymin=low)) +
  geom_point(size=2) + facet_wrap(~name) +
  xlab('treatment') + ylab('% increase in sales') +
  scale_color_viridis_d() + theme_cowplot() +
  theme(legend.position='bottom', legend.title=element_blank())
```