Replication: Gendered Citation…

Mio Hienstorfer-Heitmann

2 12 2021

```
df_articles %>%
  group_by(newjnlid, authorteam) %>%
  summarise(N = n()) %>%
  mutate(Percent = 100 * N / sum(N)) %>%
  flextable()
```

newjnlid	authorteam	N	Percent
APSR	Male only	324	69.8275862
APSR	Female only	67	14.4396552
APSR	Mixed	73	15.7327586
Politics & Gender	Male only	27	7.9178886
Politics & Gender	Female only	266	78.0058651
Politics & Gender	Mixed	47	13.7829912
Politics & Gender		1	0.2932551
Political Analysis	Male only	220	74.5762712
Political Analysis	Female only	8	2.7118644
Political Analysis	Mixed	67	22.7118644
Econometrica	Male only	465	76.9867550
Econometrica	Female only	25	4.1390728
Econometrica	Mixed	114	18.8741722
Soc. Methods & Res.	Male only	153	65.1063830
Soc. Methods & Res.	Female only	19	8.0851064

newjnlid	authorteam	N	Percent
Soc. Methods & Res.	Mixed	63	26.8085106

```r
df %>%
  filter(refauthcomplete == 1 & !is.na(refteam)) %>%
  group_by(newjnlid, refteam) %>%
  summarise(N = n()) %>%
  mutate(Percent = 100 * N / sum (N)) %>%
  flextable()
```

newjnlid	refteam	N	Percent
APSR	Male	11,617	74.239519
APSR	Female	2,203	14.078476
APSR	Mixed	1,828	11.682004
Politics & Gender	Male	1,649	27.977604
Politics & Gender	Female	3,405	57.770614
Politics & Gender	Mixed	840	14.251781
Political Analysis	Male	4,650	78.933967
Political Analysis	Female	322	5.465965
Political Analysis	Mixed	919	15.600068
Econometrica	Male	9,226	84.883614
Econometrica	Female	475	4.370227
Econometrica	Mixed	1,168	10.746159
Soc. Methods & Res.	Male	2,937	72.464841
Soc. Methods & Res.	Female	347	8.561559
Soc. Methods & Res.	Mixed	769	18.973600

```r
logit_fun <- function(y, X, theta){

    if(!is.null(ncol(X))){
```

```r
      beta <- theta[1:ncol(X)]

      mu <- X %*% beta


    } else {

      beta <- theta[1]

      mu <- X * beta


    }

    p <- 1 / ( 1 + exp(-mu) )

    logll <-  sum( y * log(p) + (1 - y) * log ( 1- p) )

    return(logll)

  }

logistic_per_journal <- function(journal, vcovcoef = FALSE){

  require(tidyverse)
  require(MASS)
  require(rms)
  require(optimx)


  if(journal != "Pooled"){
    df_anal <- df %>%
      filter(newjnlid %in% journal & refauthcomplete == 1) %>%
      dplyr::select(newjnlid, authorteam, reffemonly, newartid) %>%
      na.omit() %>%
      mutate(Female = ifelse(authorteam == "Female", 1, 0),
             Mixed = ifelse(authorteam == "Mixed", 1, 0)) %>%
      dplyr::select(-authorteam)


      y <- df_anal$reffemonly

  X <- cbind(1,
             df_anal$Female,
             df_anal$Mixed)



    # start values
  startvals <- rep(0, ncol(X))

  # optimize
  res <- optim(
```

```r
    par = startvals,
    fn = logit_fun,
    y = y,
    X = X,
    control = list(fnscale = -1),
    hessian = TRUE,
    method = "BFGS"
  )

startvals2 <- c(0, 0) # Why three this time?

restricted <- optim(
  startvals2,
  logit_fun,
  y = y,
  X = X[, 1],
  # restricted model
  control = list(fnscale = -1),
  method = "BFGS"
)

  coef <- res$par
  # vcov <- solve(-res$hessian)
  # se <- sqrt(diag(vcov))

  # Unfortunately, I am not yet able to compute robust standard errors
  # clustered at article level by hand. But I am on it.
  fit=lrm(data = df_anal, reffemonly ~ Female + Mixed, x=T, y=T)
  vcov <- vcov(robcov(lrm(data = df_anal, reffemonly ~ Female + Mixed, x=T, y=T),
               cluster = df_anal$newartid)
               )
  se <- sqrt(diag(vcov))
  ##robust standard error

  } else {

    df_anal <- df %>%
      filter(refauthcomplete == 1) %>%
      dplyr::select(newjnlid, authorteam, reffemonly, newartid) %>%
      na.omit() %>%
      mutate(Female = ifelse(authorteam == "Female", 1, 0),
             Mixed = ifelse(authorteam == "Mixed", 1, 0),
             APSR = ifelse(newjnlid == "APSR", 1, 0),
             PG = ifelse(newjnlid == "Politics & Gender", 1, 0),
             PA = ifelse(newjnlid == "Political Analysis", 1, 0),
             Econ. = ifelse(newjnlid == "Econometrica", 1, 0),
             SMR = ifelse(newjnlid == "Soc. Methods & Res.", 1, 0)) %>%
      dplyr::select(-authorteam)

      y <- df_anal$reffemonly

  X <- cbind(1,
             df_anal$Female,
```

```
             df_anal$Mixed,
             df_anal$PG,
             df_anal$PA,
             df_anal$Econ.,
             df_anal$SMR)



    # start values
  startvals <- rep(0, ncol(X))

  # optimize
  res <- optim(
    par = startvals,
    fn = logit_fun,
    y = y,
    X = X,
    control = list(fnscale = -1),
    hessian = TRUE,
    method = "BFGS"
  )

startvals2 <- c(0, 0) # Why three this time?

restricted <- optim(
  startvals2,
  logit_fun,
  y = y,
  X = X[, 1],
  # restricted model
  control = list(fnscale = -1),
  method = "BFGS"
)

  coef <- res$par
  # vcov <- solve(-res$hessian)
  # se <- sqrt(diag(vcov))

  # Unfortunately, I am not yet able to compute robust standard errors
  # clustered at article level by hand. But I am on it.
  vcov <- vcov(robcov(lrm(data = df_anal, reffemonly ~ Female + Mixed + PG + PA + Econ. + SMR, x=T, y=T
               cluster = df_anal$newartid))
  #               )
  se <- sqrt(diag(vcov))
  ##robust standard error
  }

  Names = c("Intercept", "Female", "Mixed", "P&G", "PA", "Econ", "SMR", "Pseudo R2", "NullLL",  "LL", "
  if(journal == "Pooled"){

    ModelTable <- data.frame(Name = c(paste0(round(coef, 2), " (", round(se, 2), ")"),
                                          round( 1- (restricted$value / res$value), 4),
                                          round(restricted$value, 0),
```

```r
                                              round(res$value,0),
                                              length(unique(df_anal$newartid)),
                                               nrow(df_anal))
                              )

  } else {

    ModelTable <- data.frame(Name = c(paste0(round(coef, 2), " (", round(se, 2), ")"),
                                              rep("", 4),
                                              round( 1- (restricted$value / res$value), 4),
                                              round(restricted$value, 0),
                                              round(res$value,0),
                                              length(unique(df_anal$newartid)),
                                              nrow(df_anal)))
  }

  rownames(ModelTable) <- Names
  colnames(ModelTable) <- journal


  if(vcovcoef == FALSE){

     return(ModelTable)

  } else {

    return(list(coef = coef,
                vcov = vcov))
  }

}
```

```r
models <- do.call("cbind", lapply(unique(df$newjnlid), logistic_per_journal) )
pooled <- logistic_per_journal("Pooled")

flextable(cbind.data.frame(models, pooled) %>% rownames_to_column(" "))
```

	APSR	Politics & Gender	Political Analysis	Econometrics	Soc. Methods & Res.	Pooled
Intercept	-2.07 (0.05)	-0.01 (0.11)	-2.84 (0.09)	-3.18 (0.06)	-2.46 (0.1)	-2.02 (0.05)
Female	0.99 (0.16)	0.53 (0.12)	0.42 (0.38)	1.14 (0.22)	0.76 (0.28)	0.86 (0.1)
Mixed	0.21 (0.13)	-0.15 (0.16)	-0.08 (0.16)	0.07 (0.14)	0.06 (0.18)	0.11 (0.08)
P&G						1.73 (0.1)
PA						-0.89 (0.09)
Econ						-1.14 (0.07)
SMR						-0.47 (0.1)
Pseudo R2	-0.026	-0.0165	-7e-04	-0.0106	-0.0078	-0.2796
NullLL	-6359	-4007	-1249	-1951	-1185	-18566

	APSR	Politics & Gender	Political Analysis	Econometrica	Soc. Methods & Res.	Pooled
LL	-6198	-3942	-1248	-1931	-1175	-14509
Clusters	464	332	295	604	232	1927
Observations	15648	5883	5891	10869	4053	42344

```
models_coef <- lapply(unique(df$newjnlid), logistic_per_journal, vcovcoef = TRUE)

nsim <- 1000

coef <- models_coef[[1]]$coef
vcov <- models_coef[[1]]$vcov

scenarios <-data.frame <- c(1,0,0)
scenario_female <- c(1,1,0)
scenario_mixed <- c(1,0,1)


sim_fun <- function(coef, vcov, nsim, scenarios){

  S <- mvrnorm(n = nsim, mu = coef, Sigma = vcov)

  scenarios_df <- data.frame()

  for( scenario in scenarios){

    mu <- S %*% scenario_male

    p <- 1 / (1 + exp(-mu))

    df <- data.frame(ev = mean(p),
                     lwr = quantile(p, 0.025),
                     upr = quantile(p, 0.975),
                     )

    scenarios_df <- rbind(scenarios_df, df)

  }

  plot <- ggplot(data =df,
                 x)


}
```

A different approach: counts of citations

```
n <- 1000

female <- sample(size = n, x = c(0,1), replace = TRUE)
```

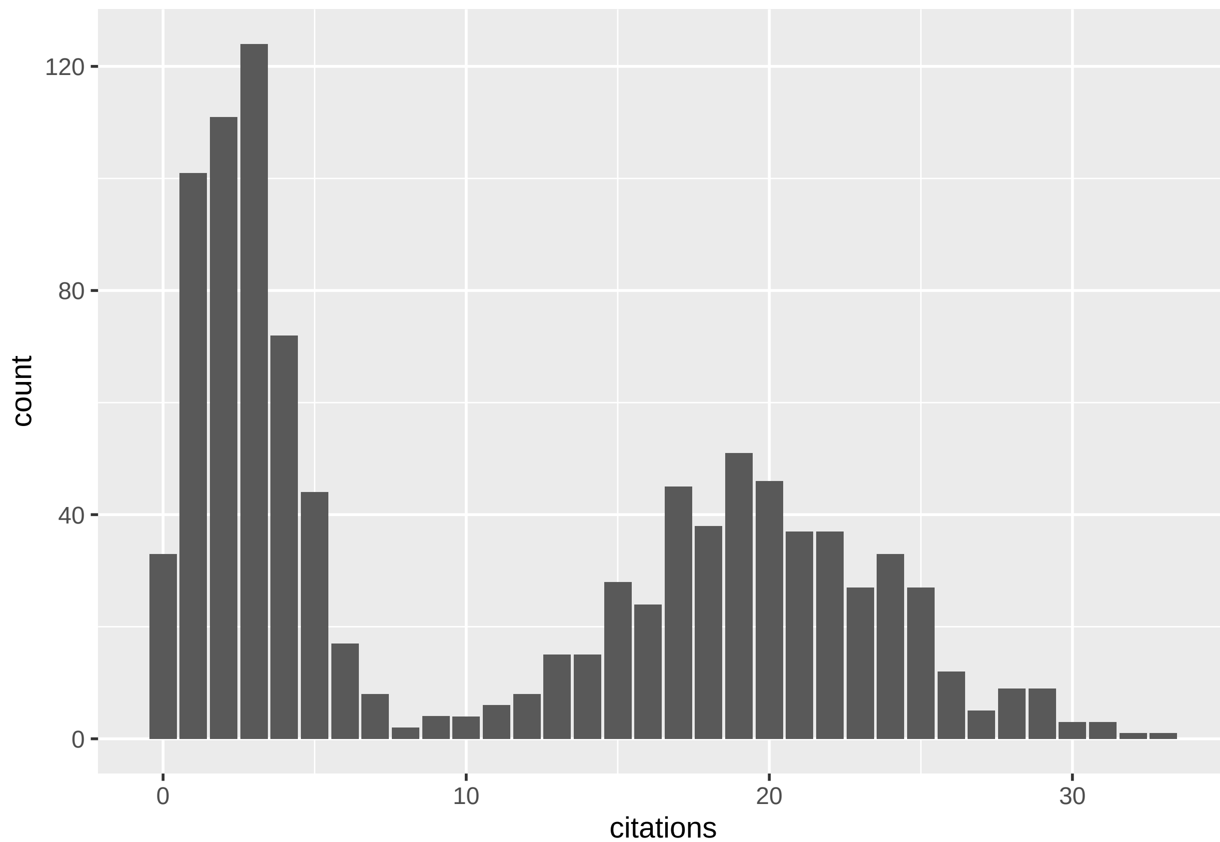
```r
beta0 <- 3
beta1 <- -2

lambda <- exp(beta0 + beta1 * female)

citations <- rpois(n, lambda)

df <- data.frame(citations = citations,
                 female = female)


ggplot(data = df,
       aes(x = citations,
           group = as.factor(female)))+
  geom_bar(stat = "count")
```




```r
df %>%
  group_by(female) %>%
  summarise(mean = mean(citations))
```

```
## # A tibble: 2 x 2
##   female  mean
##    <dbl> <dbl>
## 1      0 19.9
## 2      1  2.76
```

```
pois_ll <- function(y, X, theta){

  beta <- theta[1:ncol(X)]

  #beta <- c(beta0, beta1)

  #y = df$citations
  #X = cbind(rep(1, nrow(df)), df$female)

  # logl = sum(Y* (beta %*% t(X)) - exp(beta %*% t(X)))

  logll <- -sum(exp(beta %*% t(X))) + sum(y * (beta%*% t(X))) - sum(log(factorial(y)))

  #ll <- prod( ( exp(-exp(beta %*% t(X))) * exp(beta %*% t(X)) ^ y ) / ( factorial(y) ) )

  return(logll)

}

stval <- c(0, 0)
res <-
  optim(
    stval,
    pois_ll,
    y = df$citations,
    X = cbind(rep(1, nrow(df)), df$female),
    control = list(fnscale = -1), # this is important
    hessian = TRUE,
    method = "BFGS"
    # and tell optimx to maximize rather than to minimize.
  )
coef <- res$par
varcov <- solve(-res$hessian)


## simulate the data


scenario <- c(0,1)
nsim <- 1000
S <- mvrnorm(nsim, coef, solve(-res$hessian))


sim_pois <- function(scenario, nsim, S){

  scenarios <- c(1, scenario)

  mu <- S %*% scenarios

  # response function
  lambda <- exp(mu)

  pois <- rpois(nsim, lambda)
```

```r
  #pois <- lambda

  mean <- mean(pois)
  lower <- quantile(pois, 0.025)
  upper <- quantile(pois, 0.975)

  return(data.frame(mean = mean,
                    lower = lower,
                    upper = upper,
                    scenario = scenario))

}

sim_df <- do.call("rbind", lapply(scenario, sim_pois, S = S, nsim = nsim))

sim_df$scenario <- ifelse(sim_df$scenario == 1, "female", "male")

ggplot(data = sim_df, aes(x = scenario, ymin = lower, max = upper))+
  geom_errorbar()
```