NHISSynthesisPresentation

Sarah Boese

4/5/2020

Introduction to the National Health Interview Survey (NHIS)

The National Health Interview Survey

-The National Health Interview Survey (NHIS) is a yearly survey conducted by the US Census Bureau conserning a broad range of health topics.

-The Sample Adult dataset is one of the datasets released within the survey. Some of the variables are topcoded but I have not found any info on how they are otherwise synthesized.

-The Sample Adult dataset has 742 variables, however, I am only synthesizing three of them: FLA1AR, DOINGLWA and WKDAYR. To that end, I am also containing the "known" or unsynthesized variable I am using for analysis to SEX, RACERPI2, AGE_P.

Variables of Interest

FLA1AR Variable

-Description: Any functional limitation, all conditions.

-Outcome (Catagorical):

1	Limited in any way
2	Not limited in any way
3	Unknown if limited

-Model: $y_i \sim Bin(p)$ such that $p \sim Beta(57, 43)$.

DOINGLWA Variable

-Description: Corrected employment status last week

-Outcome (Catagorical)

1	Working for pay at a job or business
2	With a job or business but not at work
3	Looking for work
4	Working, but not for pay, at a family-owned job or business
5	Not working at a job or business and not looking for work
7	Refused
8	Not ascertained
9	Don't know

-Model: $y_i \sim Multinomial(p[i, 1:C])$ such that $p[i, 1:C] \sim Dirichlet(\alpha[1:C])$ where $\alpha[c] = 1$. Here $C = 5$ as I do not consider nonrecoreded values.

WKDAYR Variable

-Description: Number of work loss days, past 12 months

-Outcome (Continuous)

000	None
001 – 366	1 - 366 days
997	Refused
998	Not ascertained
999	Don't know

WKDAYR Model

-I am using a Poisson Model to count the number of days of the year the number of work loss days per month with linear predictor determined by the synthesized values of FLA1AR and DOINGLWA variables.

-Model: $y_i \sim Poisson(\lambda_i)$ where $log(\lambda_i) \sim \beta_0 + \beta_1 \cdot x_{lim} + \beta_2 \cdot x_{stat_1} + \beta_3 \cdot x_{stat_2} + \beta_4 \cdot x_{stat_3} + \beta_5 \cdot x_{stat_4} + \beta_6 \cdot x_{stat_5}$ such that $\beta_i \sim Normal(0, 10)$.

-Here x_lim denotes $FLA1AR = 1$, and x_{stat_i} denotes $DOINGLWA = i$.

Synthesis

FLA1AR Synthesis

```
synthesize_lim <- function(index, n, post_lim){
  synthetic_lim <- rbinom(n, 1, post_lim[index,"p"])
  return(data.frame(synthetic_lim))
}

n <- nrow(NHISdata)
post_lim <- as.mcmc(posterior_lim)
syn_lim <- synthesize_lim(1, n, post_lim)
names(syn_lim)=c("synthesized_func_lim")
```

DOINGLWA Synthesis

```
synthesize_stat<-function(index, n_syn, p){
  synthetic_stat <- rmultinom(n=n_syn, size=1, prob = c(p[
  return(data.frame(t(synthetic_stat)))
}

post_stat <- as.mcmc(posterior_stat)
p<-as.matrix(post_stat)
n<-nrow(NHISdata)
syn_stat <- synthesize_stat(1000, n, p)
names(syn_stat)=c("syn_emp_stat_1", "syn_emp_stat_2", "syn_
```

WKDAYR Synthesis

```
synthesize_wrk <- function(X, index, n, post_wrk){
  lambda <- exp(post_wrk[index, "beta0"] + X$x_lim * post_w
  + X$x_stat_4 * post_wrk[index, "beta5"]
  + X$x_stat_5 * post_wrk[index, "beta6"])
  synthetic_Y <- rpois(n, lambda)
  data.frame(synthetic_Y)
}

post_wrk <- as.mcmc(posterior_wrk)
params<-data.frame(x_lim, x_stat_1, x_stat_2, x_stat_3, x_s
n <- nrow(NHISdata)
syn_wrk <- synthesize_wrk(params, 1, n, post_wrk)
names(syn_wrk)=c("synthesized_lost_wrk")
```

Utility Measures

Propensity Score: Forming Dataframes

```
create_T<- function(org,syn,n, variable){
  original_T<-data.frame(org, integer(length=n))
  names(original_T)= c(variable, "T")

  synthetic_T<-data.frame(syn, integer(length=n)+1)
  names(synthetic_T)= c(variable, "T")
  merged_T<- bind_rows(original_T, synthetic_T)
}
n<-nrow(NHISdata)
merged_data_lim<-create_T(SyntheticData_lim$org_func_lim, S
merged_data_stat <- create_T(SyntheticData_stat$org_emp_sta
merged_data_wrk <- create_T(SyntheticData_wrk$org_lost_work
```

Function for calculating Propensity Score

-Here I wrote a function to calculate Propensity Score and ran it on my synthesized variables.

```
calc_Up<-function(var, merged_data){
  log_reg<-glm(T ~ eval(parse(text = var)), data = merged_d
  pred <- predict(log_reg, data = merged_data)
  probs <- exp(pred)/(1+exp(pred))
  Up <- 1/(2*n)*sum((probs - 1/2)^2)
  return(Up)
}
```

Calculating Propensity Score

```
Up_lim<-calc_Up("FLA1AR", merged_data_lim)
Up_stat<-calc_Up("DOINGLWA", merged_data_stat)
Up_wrk<-calc_Up("WKDAYR", merged_data_wrk)
```

```
Up_lim
```

```
## [1] 2.381706e-05
```

```
Up_stat
```

```
## [1] 0.1003169
```

```
Up_wrk
```

```
## [1] 1.911354e-06
```

Function using clustering algorithim to calculate cluster analysis utility measure.

```
calc_Uc<-function(merged_data){
  clusters <- hclust(dist(merged_data[, 1:2]), method = 'av
  G <- 5
  clusterCut <- cutree(clusters, G)
  cluster_S <- as.data.frame(cbind(clusterCut, merged_dataS
  names(cluster_S) <- c("cluster", "S")
  table(cluster_S)

  n_gS <- table(cluster_S)[, 1]
  n_g <- rowSums(table(cluster_S))
  w_g <- n_g / (2*n)

  Uc <- (1/G) * sum(w_g * (n_gS/n_g - 1/2)^2)
  return(Uc)
}
```

Calculating cluster analysis measure Uc

```
Uc_lim<-calc_Uc(merged_data_lim)
Uc_stat<-calc_Uc(merged_data_stat)
Uc_wrk<-calc_Uc(merged_data_wrk)
```

```
Uc_lim
```

```
## [1] 0.05
```

```
Uc_stat
```

```
## [1] 0.03778899
```

```
Uc_wrk
```

```
## [1] 0.0005050505
```