library(tidyverse)

library(haven)

library(survey)

library(scales)

options( survey.lonely.psu = "adjust" )

#read in PR data

pr = read\_dta("CDPR61FL.zip")

pr = pr %>% mutate(across(where(is.labelled),as\_factor))

#calculate number of children in each HH, in each age band

#used to estimate # under-15 in the household

children\_table = pr %>%

group\_by(hv001,hv002) %>%

summarise(children\_0\_5\_calc = sum(hv105 %in% 0:5,na.rm=T),

children\_0\_15\_calc = sum(hv105 %in% 0:14,na.rm=T)

)

pr = left\_join(pr,children\_table)

#subset to eligible adults:

pr\_sub = pr %>% filter(hv027 =="men's survey", #male survey

hv117 == 'eligible' | #eligible for women's or mens' survey

hv118 == 'eligible'

)

#read in serology

sero = read\_csv("adult\_polio\_serology.csv")

#na samples without sufficient sample, and calculate seropositive:

sero = sero %>% mutate(

Sabin\_1 = if\_else(!is.na(S1\_comment),NA\_real\_,Sabin\_1),

Sabin\_2 = if\_else(!is.na(S2\_comment),NA\_real\_,Sabin\_2),

Sabin\_3 = if\_else(!is.na(S3\_comment),NA\_real\_,Sabin\_3),

p1 = as.numeric(Sabin\_1 >=3),

p2 = as.numeric(Sabin\_2 >=3),

p3 = as.numeric(Sabin\_3 >=3)

)

#merge data together

data = left\_join(pr\_sub,sero,by = c('hv001','hv002','hvidx') )

#rename data for convenience

data = data %>%

rename(cluster = hv001,

hhnum = hv002,

psu = hv021,

strata= hv022,

age = hv105,

sex = hv104,

residence = hv025,

province = hv024,

education = hv106,

wealth = hv270,

weight = hv028,

children05 = hv014,

province\_new = shnprovin)

#note calculated under-5 != reported under-5 children in HH for some houses

data %>% filter(children05 != children\_0\_5\_calc) %>% distinct(cluster,hhnum,.keep\_all = T)

data = data %>% mutate(weight = weight/1e6,

age = as.numeric(age),

age\_cat = cut(age,c(0,19,24,29,34,39,44,100),labels = c('15-19','20-24','25-29','30-34','35-39','40-44','44+')),

child\_cat = cut(children05,c(-1,0,1,2,3,Inf),labels=c('None','1','2','3','4+')),

child15\_cat = cut(children\_0\_15\_calc,c(-1,0,2,5,Inf),labels=c('None','1-2','3-5','5+')))

# Tell R about the survey design, and subset to complete observations:

des = svydesign(ids = ~psu+hhnum, strata = ~strata, weight = ~weight, nest = TRUE, data = data)

des\_complete = subset(des,!is.na(p1),!is.na(p2),is.na(p3))

#### raking to match province, rural/urban, and age weighting of the full sample #################

pop.province = data %>%

group\_by(province\_new) %>% summarise(Freq= sum(weight))

pop.residence = data %>%

group\_by(residence) %>% summarise(Freq = sum(weight))

pop.age = data %>%

group\_by(age\_cat) %>% summarise(Freq= sum(weight))

pop.sex = data %>%

group\_by(sex) %>% summarise(Freq= sum(weight))

des\_ps = rake(des\_complete,sample.margins = list(~province\_new,~residence,~age\_cat,~sex),population.margins = list(pop.province,pop.residence,pop.age,pop.sex))

#helper function to format svyby() output:

reshape\_svyby = function(x){

x %>% unite(sabin1,matches('p1')) %>% unite(sabin2,matches('p2')) %>% unite(sabin3,matches('p3')) %>%

gather(serotype,tmp, matches('sabin')) %>%

separate(tmp,c('seroprevalence','lower','upper'),'\_') %>%

mutate\_at(vars(seroprevalence,lower,upper), funs(as.numeric)) %>%

mutate(serotype = recode(serotype, 'sabin1' = 'Type 1', 'sabin2' = 'Type 2', 'sabin3' = 'Type 3'))

}

#overall seroprevalence

overall =svymean(~p1+p2+p3,des\_ps,na.rm=T,na.rm.all = T)

overall = as\_tibble(overall) %>% mutate(serotype = c('Type 1','Type 2','Type 3')) %>%

bind\_cols(confint(overall) %>% as\_tibble %>% setNames(c('lower','upper'))) %>%

rename(seroprevalence=mean)

#seroprevalece by province

tab = svyby(~p1+p2+p3,

~province\_new,

des,

FUN = svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci')

tab = reshape\_svyby(tab)

tab %>% bind\_rows(overall %>% mutate(province\_new = 'overall')) %>%

mutate(Province = reorder(str\_to\_title(province\_new),-seroprevalence\*as.numeric(serotype=='Type 1') - 1000\*as.numeric(province\_new == 'overall'))) %>%

ggplot(aes(x=Province,y=seroprevalence,ymin = lower,ymax=upper, colour = serotype)) +

geom\_pointrange(position = position\_dodge(width = 0.5)) +

scale\_y\_continuous('Seroprevalence', labels=percent\_format(), limits = c(0,1),breaks =(0:10)/10,oob = squish) +

scale\_colour\_discrete("Serotype") +

scale\_x\_discrete(NULL)+

theme\_bw() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1),panel.grid.major.y = element\_line(), panel.grid.minor.y = element\_line(linetype = 2))

ggsave("Figure\_1.png",width = 7,height=6,units='in',dpi=600)

# seroprevalence by sex and age group

tab\_age\_sex = svyby(~p1+p2+p3,~age\_cat+sex,des\_ps,FUN = svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci')

tab\_age\_sex = reshape\_svyby(tab\_age\_sex)

tab\_age\_sex= tab\_age\_sex %>% mutate(sex = recode(sex,'male'='M','female'='F'))

tab\_age\_sex %>%

ggplot(aes(x=sex,y=seroprevalence,ymin = lower,ymax=upper, colour= serotype)) +

geom\_pointrange(position = position\_dodge(width = 0.3)) +

scale\_y\_continuous('Seroprevalence', labels=percent\_format(), limits = c(0,1),breaks = c(0,.2,.4,.6,.8,1)) +

scale\_colour\_discrete("Serotype") +

scale\_x\_discrete(NULL)+

theme\_bw() +

facet\_wrap(~age\_cat,nrow=1,scales = 'free\_x')+

theme(panel.grid.major.y = element\_line(), panel.grid.minor.y = element\_line(linetype = 2))

ggsave("Figure\_2.png",width = 6,height=4,units='in',dpi=600)

# Pattern of seroprevalence, among all adults, and by sex:

mysvymean = function(formula,design,...){

m = svymean(formula,design,na.rm=T)

ci = confint(m)

tibble(...,seroprevalence = as.numeric(m),lower=ci[,1],upper=ci[,2])

}

tab\_combo = bind\_rows(

svyby(~p1+p2+p3, ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby,

svyby(~I(p1\*p2\*p3), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Types 1, 2, and 3'),

svyby(~I(p1\*p2), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Types 1 and 2'),

svyby(~I(p1\*p3), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Types 1 and 3'),

svyby(~I(p2\*p3), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Types 2 and 3'),

svyby(~I(p1\*(1-p2)\*(1-p3)), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Type 1 only'),

svyby(~I(p2\*(1-p1)\*(1-p3)), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Type 2 only'),

svyby(~I(p3\*(1-p1)\*(1-p2)), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci')%>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'Type 3 only'),

svyby(~I((1-p3)\*(1-p1)\*(1-p2)), ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci')%>%

setNames(c('sex','seroprevalence','lower','upper')) %>% mutate(serotype = 'All negative'),

mysvymean(~p1+p2+p3, des\_ps,sex='All adults',serotype = c('Type 1','Type 2','Type 3')),

mysvymean(~I(p1\*p2\*p3), des\_ps,sex='All adults',serotype = 'Types 1, 2, and 3'),

mysvymean(~I(p1\*p2), des\_ps,sex='All adults',serotype = 'Types 1 and 2'),

mysvymean(~I(p1\*p3), des\_ps,sex='All adults',serotype = 'Types 1 and 3'),

mysvymean(~I(p2\*p3), des\_ps,sex='All adults',serotype = 'Types 2 and 3'),

mysvymean(~I(p1\*(1-p2)\*(1-p3)), des\_ps,sex='All adults',serotype = 'Type 1 only'),

mysvymean(~I(p2\*(1-p1)\*(1-p3)), des\_ps,sex='All adults',serotype = 'Type 2 only'),

mysvymean(~I(p3\*(1-p1)\*(1-p2)), des\_ps,sex='All adults',serotype = 'Type 3 only'),

mysvymean(~I((1-p3)\*(1-p1)\*(1-p2)), des\_ps,sex='All adults',serotype = 'All negative')

)

tab\_combo\_display = tab\_combo %>% mutate(across(c(seroprevalence,lower,upper),~100\*round(.,2))) %>%

transmute(sex,serotype,values = str\_c(seroprevalence, " (",lower,'-',upper,')'))

tab\_combo\_display = tab\_combo\_display %>% pivot\_wider(values\_from=values,names\_from = sex) %>%

select(serotype,`All adults`,Male = male,Female = female)

tab\_combo\_display

tab\_combo\_display %>% write\_csv('table\_1.csv')

#seroprevalence by demographics:

tab\_demos = bind\_rows(

svyby(~p1+p2+p3, ~sex,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=sex) %>% mutate(variable = 'Sex'),

svyby(~p1+p2+p3, ~age\_cat,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=age\_cat) %>% mutate(variable = 'Age'),

svyby(~p1+p2+p3, ~education,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=education) %>% mutate(variable = 'Education'),

svyby(~p1+p2+p3, ~wealth,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=wealth) %>% mutate(variable = 'Wealth'),

svyby(~p1+p2+p3, ~child\_cat,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=child\_cat) %>% mutate(variable = 'Children 0-5'),

svyby(~p1+p2+p3, ~child15\_cat,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=child15\_cat) %>% mutate(variable = 'Children 0-14'),

svyby(~p1+p2+p3, ~residence,des\_ps,svymean,na.rm.by = T,na.rm=T,na.rm.all = T,vartype = 'ci') %>% reshape\_svyby %>%

rename(level=residence) %>% mutate(variable = 'Type of Residence')

)

#reformat for pretty printing:

tab\_demos\_display = tab\_demos %>% mutate(across(c(seroprevalence,lower,upper),~100\*round(.,2))) %>%

transmute(variable,level,serotype,values = str\_c(seroprevalence, " (",lower,'-',upper,')'))

tab\_demos\_display = tab\_demos\_display %>% pivot\_wider(names\_from=serotype,values\_from=values)

#calculate population demographics for reference:

demos\_proportion = svymean(~sex+age\_cat+wealth+education+child\_cat+child15\_cat+residence,des,na.rm=T) %>%

as\_tibble(rownames = 'level') %>%

mutate(variable = str\_extract(level,'sex|age\_cat|wealth|education|child\_cat|child15\_cat|residence'),

level = str\_replace(level,'sex|age\_cat|wealth|education|child\_cat|child15\_cat|residence',''),

variable = recode(variable,sex='Sex',wealth = 'Wealth',age\_cat='Age',education = 'Education',child\_cat='Children 0-5',child15\_cat='Children 0-14',residence='Type of Residence')) %>%

transmute(variable,level,Proportion = 100\*round(mean,2))

tab\_demos\_display = left\_join(tab\_demos\_display,demos\_proportion) %>%

relocate(variable,level,Proportion)

#perform tests:

## updating design to fix error in svychisq() when empty factor levels in education

des\_ps = update(des\_ps,education2 = as.character(education))

test\_df = expand\_grid(serotype = c('p1','p2','p3'),

variable = c('sex','age\_cat','child\_cat','child15\_cat','wealth','education2','residence')) %>%

mutate(formula = str\_c('~',serotype,'+',variable))

test\_df = test\_df %>% rowwise() %>%

mutate(p.value = svychisq(as.formula(formula),des\_ps,na.rm=T)$p.value)

test\_df = test\_df %>% mutate(serotype = recode(serotype,'p1'='Type 1','p2' = 'Type 2','p3' = 'Type 3'),

variable = recode(variable,sex='Sex',wealth = 'Wealth',age\_cat='Age',education2 = 'Education',child15\_cat='Children 0-14',child\_cat='Children 0-5',residence='Type of Residence'))

test\_df = test\_df %>% mutate(p.value = str\_c('p=',format(p.value,digits=2))) %>% select(-formula) %>%

pivot\_wider(names\_from = serotype,values\_from = p.value)

tab\_demos\_display = bind\_rows(tab\_demos\_display,test\_df)

tab\_demos\_display = tab\_demos\_display %>%

arrange(variable)

tab\_demos\_display

write\_csv(tab\_demos\_display,'table\_2.csv')