Bayesian Sequential Survey Estimator Demo on Yemen mVAM Data

Intro - Data Prep

In this notebook we use the Bayesian Suquential Survey Estimator on an mVAM dataset collected in Yemen starting in March of 2018. We use three datasets:

1. mVAM data - collected daily over the phone with key demographic variables and the Food Consumption Score indicator
2. Baseline data (F2F)- Emergency Food Security and Nutrition Assessment (EFSNA) data collected face-to-face in Yemen during ceasefire in fall of 2016. Also contains key demographic variables and the Food Consumption Score indicator
3. Demographic and Health Survey (DHS) data - 2013 survey of over 17000 households in Yemen collecting key demographic and health indicators. does not have Food Consumption Score, but we use it solely for the purpose of posterior simulation, predicting FCS from the demographic variables, which are presumably as representative as can be without a census

We also use an additional dataset of population by governorate to convert prevalence estimates to absolute values.

```
##################
### Data Prep  ###
##################
library(xlsx)
library(survey)
library(plyr)

#mvam Suvey- data we will use to update baseline model
ymnRspDF <- data.table::fread('ymnData/mVAMYmnObs.csv',sep=';',na.strings='')

#DHS Survey - demographic data to use for posterior simulation
dhsDF <- data.table::fread('ymnData/DHS2013.csv',sep=';',na.strings='')
dhsDF <- as.data.frame(dhsDF)

#EFSNA face-t0-face survey - data to fit baseline model (this dataset is confid
ential and cannot be shared)
efsnaDF <- data.table::fread('ymnData/ymnEFSNA2016.csv',sep=';',na.strings='')

#Population Data
ymnPopDF <- read.xlsx('C:/Users/gaurav.singhal/Desktop/ymnSeqMdl/cso_2017_popul
ation_projection_sexage_disagreggated.xlsx',
                      sheetIndex=1)

#Create table for IPC Phase
svyYmn <- svydesign(ids=~Cluster_No+ID,
                    strata=~ADM1_NAME+NULL,
                    weights=~weight,
                    pps='brewer',
                    data=efsnaDF,
                    nest=TRUE)

ymn.FCG.2016 <- prop.table(svytable(~ADM1_NAME+FCS_Cat,svyYmn),1)
colnames(ymn.FCG.2016) <- c('2016.Severe','2016.Moderate','2016.OK')
ymn.FCG.2016 <- rbind(ymn.FCG.2016,ymn.FCG.2016[c("Hajjah","Hadramaut"),])
rownames(ymn.FCG.2016)[dim(ymn.FCG.2016)[1]-1] <- "Sa'ada"
rownames(ymn.FCG.2016)[dim(ymn.FCG.2016)[1]] <- "Al Maharah"
ymn.FCG.2016 <- as.data.frame(ymn.FCG.2016)

#Assign IPC phase to governorates
ymn.FCG.2016$IPCclass <- 'Emergency'
stressed <- c('Hadramaut','Al Maharah')
ymn.FCG.2016$IPCclass[rownames(ymn.FCG.2016) %in% stressed] <- 'Stressed'
crisis <- c('Al Jawf','Marib','Amran','Amanat Al Asimah','Al Mahwit','Al Hudayd
ah','Dhamar','Raymah','Ibb')
ymn.FCG.2016$IPCclass[rownames(ymn.FCG.2016) %in% crisis] <- 'Crisis'
ymn.FCG.2016$ADM1_NAME <- rownames(ymn.FCG.2016)
rm(svyYmn,stressed,crisis)
```

```
#Filter mVAM dataset for latest questionnaire and Merge
colsmVAM <- c('ObsID','RspID','SvyID','SelectWt','CmbAdjWt','ADM1_NAME','ADM2_N
AME',
              'rCSI','FCS','FCG','FoodInsecure','isUrban','isIDP',
              'HHSizeGrp','HoHSex_is_F','HoHEdu','H2OSrc','FuelSrc','HH_has_Elc
trc','Toilet_is_Flush')

mvamDF <- as.data.frame(ymnRspDF[ymnRspDF$SvyID>=181,])
mvamDF <- mvamDF[,colSums(is.na(mvamDF))<0.9*nrow(mvamDF)]
mvamDF <- mvamDF[complete.cases(mvamDF[,colsmVAM]),]
mvamDF <- mvamDF[,append(colsmVAM,setdiff(colnames(mvamDF),colsmVAM))]
mvamDF <- merge(mvamDF,ymn.FCG.2016,all.x=TRUE,by='ADM1_NAME')
mvamDF$FreeResponse <- NULL
mvamDF$FreeResponseEng <- NULL

#Filter DHS dataset for complete.cases and merge
colsDHS <- c('HHID','ADM1_NAME','wgt','HHS_Class','HHS_Score','isUrban',
             'HHSizeGrp','HoHSex_is_F','HoHEdu','H2OSrc','FuelSrc','HH_has_Elct
rc','Toilet_is_Flush')
dmgDF <- as.data.frame(dhsDF)
dmgDF <- dmgDF[complete.cases(dmgDF[,colsDHS]),]
dmgDF <- dmgDF[,append(colsDHS,setdiff(colnames(dhsDF),colsDHS))]
dmgDF <- merge(dmgDF,ymn.FCG.2016,all.x=TRUE,by='ADM1_NAME')

#Filter EFSNA df for complete.cases and merge
colsEFSNA <- c('ID','ADM1_NAME','wgt','isUrban','FCS','isIDP',
               'HHSizeGrp','HoHSex_is_F','HoHEdu','H2OSrc','FuelSrc','HH_has_El
ctrc','Toilet_is_Flush')
efsnaDF <- as.data.frame(efsnaDF)
baseDF <- efsnaDF[complete.cases(efsnaDF[,colsEFSNA]),]
baseDF <- baseDF[,colsEFSNA]
baseDF <- merge(baseDF,ymn.FCG.2016,all.x=TRUE,by='ADM1_NAME')
baseDF$FCS[baseDF$FCS<12] <- 12
baseDF$logFCS.2016 <- log(baseDF$FCS)

#add rows for Sa'ada and Al Maharah as they are missing the original baseline d
ata
extDF <- baseDF[baseDF$ADM1_NAME %in% c('Hajjah','Hadramaut'),]
extDF$ADM1_NAME <- revalue(extDF$ADM1_NAME,c('Hajjah'="Sa'ada",'Hadramaut'='Al
Maharah'))
baseDF <- rbind(baseDF,extDF)
baseDF$ADM1_NAME <- as.character(baseDF$ADM1_NAME)
rm(extDF)

print('Data Loaded!')
```

baseline dataset

```
##   ADM1_NAME   ID   wgt isUrban  FCS isIDP HHSizeGrp HoHSex_is_F    HoHEdu
## 1     Abyan    1 1.036    TRUE 18.0 FALSE     (5,8]        TRUE    noEduc
## 2     Abyan    2 1.038   FALSE 20.5 FALSE  (16,100]       FALSE   primary
## 3     Abyan  732 0.942   FALSE 22.5 FALSE     (5,8]        TRUE    noEduc
## 4     Abyan 3528 0.930   FALSE 21.0 FALSE     (5,8]       FALSE    noEduc
## 5     Abyan 1464 1.038    TRUE 22.0 FALSE    (8,12]       FALSE    higher
## 6     Abyan 3409 0.942   FALSE 42.0 FALSE     (5,8]       FALSE secondary
##       H2OSrc FuelSrc HH_has_Elctrc Toilet_is_Flush 2016.Severe
## 1      piped    wood         FALSE           FALSE   0.1305556
## 2 protectSrc    wood         FALSE            TRUE   0.1305556
## 3 protectSrc     gas         FALSE            TRUE   0.1305556
## 4  tankTruck    wood         FALSE            TRUE   0.1305556
## 5 protectSrc     gas         FALSE            TRUE   0.1305556
## 6 protectSrc    wood         FALSE            TRUE   0.1305556
##   2016.Moderate   2016.OK  IPCclass logFCS.2016
## 1     0.3527778 0.5166667 Emergency    2.890372
## 2     0.3527778 0.5166667 Emergency    3.020425
## 3     0.3527778 0.5166667 Emergency    3.113515
## 4     0.3527778 0.5166667 Emergency    3.044522
## 5     0.3527778 0.5166667 Emergency    3.091042
## 6     0.3527778 0.5166667 Emergency    3.737670
```

mVAM dataset

```
##   ADM1_NAME isUrban FCS isIDP HHSizeGrp HoHSex_is_F    HoHEdu H2OSrc
## 1     Abyan   FALSE  57 FALSE     (5,8]       FALSE secondary  piped
## 2     Abyan   FALSE  77 FALSE    (8,12]       FALSE secondary  piped
## 3     Abyan   FALSE  94 FALSE     (5,8]       FALSE secondary  piped
## 4     Abyan   FALSE  93 FALSE    (8,12]       FALSE secondary  piped
## 5     Abyan   FALSE  63 FALSE     (2,5]       FALSE    higher  other
## 6     Abyan   FALSE  48 FALSE     (5,8]       FALSE secondary  piped
##   FuelSrc HH_has_Elctrc Toilet_is_Flush 2016.Severe 2016.Moderate
## 1     gas          TRUE            TRUE   0.1305556     0.3527778
## 2     gas          TRUE            TRUE   0.1305556     0.3527778
## 3     gas          TRUE            TRUE   0.1305556     0.3527778
## 4     gas          TRUE            TRUE   0.1305556     0.3527778
## 5     gas          TRUE            TRUE   0.1305556     0.3527778
## 6     gas          TRUE            TRUE   0.1305556     0.3527778
##     2016.OK  IPCclass  SelectWt    ObsDate SvyID
## 1 0.5166667 Emergency 0.3333333 2018-05-12   183
## 2 0.5166667 Emergency 0.5000000 2018-05-10   183
## 3 0.5166667 Emergency 0.3333333 2018-06-11   184
## 4 0.5166667 Emergency 0.5000000 2018-04-16   182
## 5 0.5166667 Emergency 0.5000000 2018-05-08   183
## 6 0.5166667 Emergency 0.3333333 2018-04-01   182
```

DHS dataset

```
##   ADM1_NAME   wgt isUrban HHSizeGrp HoHSex_is_F  HoHEdu H2OSrc FuelSrc
## 1       Ibb 1.026    TRUE    (8,12]       FALSE primary  piped     gas
## 2       Ibb 1.026    TRUE     (2,5]       FALSE primary  piped     gas
## 3       Ibb 1.026    TRUE     (5,8]       FALSE  higher  piped     gas
## 4       Ibb 1.026    TRUE     (5,8]        TRUE  noEduc  piped     gas
## 5       Ibb 1.026    TRUE     (5,8]       FALSE  noEduc  piped     gas
## 6       Ibb 1.026    TRUE     (2,5]       FALSE  noEduc  piped     gas
##   HH_has_Elctrc Toilet_is_Flush
## 1          TRUE            TRUE
## 2          TRUE            TRUE
## 3          TRUE            TRUE
## 4          TRUE            TRUE
## 5          TRUE            TRUE
## 6          TRUE            TRUE
```

Construct initial baseline model

Next we construct a multi-level random effect model on the baseline F2F data, that predicts log(FCS) on the host of available demographic and socioeconomic information. Effectively the models are dividing the population into a myriad of *cells* demograpahically decohering the log(FCS) estimates such that we are estimating the marginal distributions of log(FCS) by demographic covariate. Those covariates that are not independent of each other (such as IPC phase and governorate) are appropriately nested such that the nested variable is conditionally independent on the other). The joint probabilities of the marginal distributions yield the individual demographic-cell value, allowing sparse cells (those with few observations) to *borrow strength* from neighboring cells.

First we use lme4 to fit a non-Bayesian model based on maximizing the Random-Effects Maximum Likelihood criterion. Those estimates are then set as priors to the same model but estimated using rstanarm's Markov Chain Monte Carlo estimator. The results should largely be the same, but the latter model can be Bayes' updated with new information.

```
####################################
### Create initial baseline model ###
####################################
library(lme4)
library(rstanarm)

#create model formula
base.fmla <- as.formula(log(FCS) ~ HoHSex_is_F+HH_has_Elctrc+Toilet_is_Flush+is
Urban+
                  (1|HHSizeGrp)+(1|HoHEdu)+(1|H2OSrc)+(1|FuelSrc)+(1|IPCclass/
ADM1_NAME))

#(A) fit REML model using lme4 on the baseline data
base.mdl <- lmer(base.fmla,data=baseDF,weights=wgt)

#(B) fit Bayiesn model using lme model values as priors. We inflate the estimat
ed variance to give the MCMC sampler more 'search room'
base.mdl.stan <- stan_lmer(base.fmla,data=baseDF,weights=wgt,
                           iter=3000,chains=4,cores=4,thin=2,control=list(adapt
_delta=.965),QR=TRUE,
                           prior_intercept = normal(offset.mdl@beta[1],sqrt(vco
v(base.mdl)[1,1])*10))
rm(base.mdl)
```

Construct mode-effect model

The above model allows us to correct for differences in demographics with our mVAM surveys, as we are using the mVAM information to solely estimate the mean and variances of the demographic marginal distributions with respect to log(FCS). However, as mVAM uses a different modality–phone vs face-to-face–there exists a second form bias whereby respondents' estimates change due to survey mode. We eliminate this bias with the assumption that is is fixed (not dependent on time) and dependent on the respondents' socio-demographic profile as well. Therefore we take an mVAM dataset from the same time period, use the above model to predict FCS (out-of-sample) using the socio-demographic variables in the mVAM dataset, and compute its' errors vis-a-vis the reported mVAM FCS. Those errors are then the respondents' theorized estimate of 'mode-effect.' We regress the same hierarchical random-effects model on the errors to then predict 'mode-effect' on new respondents. Ergo, when new mVAM data arrives. We first run the mode-effect correction model to adjust the respondents' corresponding log(FCS) for mode-bias. As the mode-effect itself is a random variable, we must use the bootstrap technique to compute subsequent variances.

```
########################################
### Create mode-effect (offset) model ###
########################################
##(2) fit offset model to correct for mode effect using the residuals of the ba
seline model simulated on mVAM data

#(A) massage mvam data first because Sa'ada and Al Maharah governorates missing
from baseline survey
###  we assume Sa'ada=Hajjah and Al Maharah="Hadramaut for this purpose
mvamDF$ADM1_NAME <- as.factor(mvamDF$ADM1_NAME)
mvamDF$ADM1_NAME_ACTUAL <- mvamDF$ADM1_NAME
mvamDF$ADM1_NAME <- droplevels(revalue(mvamDF$ADM1_NAME,c("Sa'ada"="Hajjah","Al
Maharah"="Hadramaut")))

#(B) now perform posterior simulations (i.e. predictions) of the previous basel
ine model on the new mVAM data
mvamDF$logFCS.prior <- rowMeans(t(posterior_predict(base.mdl.stan,newdata=mvamD
F[,c(colsmVAM,'IPCclass')],
                                                    draws=500)))

#(C) now as know actual reported log(FCS) from the mVAM data, compute differenc
es vis-a-vis the above predictions mvamDF$logFCS.diff <- with(mvamDF,log(FCS)-l
ogFCS.prior)

#(D) revert governorates
mvamDF$ADM1_NAME <- mvamDF$ADM1_NAME_ACTUAL
mvamDF$ADM1_NAME_ACTUAL <- NULL

### Model Formula
offset.fmla <- as.formula(logFCS.diff ~ HoHSex_is_F+HH_has_Elctrc+Toilet_is_Flu
sh+isUrban+
                      (1|HHSizeGrp)+(1|HoHEdu)+(1|H2OSrc)+(1|FuelSrc)+(1|IPCclas
s/ADM1_NAME))

#NOTE: SvyID==181 is a filter as to use only 1 round of information correspondi
ng to mVAM's closest available time period to when the F2F survey was completed

#(E) first fit lme model as prior for rstanarm model
offset.mdl <- lmer(offset.fmla,data=mvamDF[mvamDF$SvyID==181,],weights=SelectWt
)

#(F) fit rstanarm with lme prior
offset.mdl.stan <- stan_lmer(offset.fmla,weights=SelectWt,
                             iter=3000,chains=4,cores=4,thin=2,control=list(ada
pt_delta=.965),QR=TRUE,
                             prior_intercept = normal(offset.mdl@beta[1],sqrt(v
cov(offset.mdl)[1,1])*10))
```

```
rm(offset.mdl)
```

Perform Bayesian Sequential Update on models with new data

With these two models in place we can now perform Bayesian sequential updates as new mVAM data arrives upon the baseline model that predicts log(FCS) as a function of socio-demographic variables. This is done as a batch process using a rolling window over the data of a specified time length (usually 30 days). The model for the new moving window is computed by performing a Bayesian sequential update, that is using the previous parameter estimates as a prior for the new parameter estimates, on the previous window's model using only the data from the current moving window. This is repeated for the whole sequence moving windows in the dataset.

An important note is that at the first Bayesian sequential update, the model is adjusted to include the estimates from the baseline model, that is we add the term `log(FCS)_baseline` to the right-hand-side of the regression equation, i.e.
`log(FCS)_mvam_t ~ {sociodemographic random effect variables} + log(FCS)_baseline` where `log(FCS)_baseline` are the predictions from the baseline model. Hence the priors for the random-effects come from the offset model, not the baseline model. And the random effects are effectively estimating the difference between the mVAM results and the baseline results, just as the offset model does. We call this set-up the 'cross-sectional' version of the Bayesian sequential update process. Another option is to use the previous time-window's model estimate of log(FCS) in the RHS, i.e. `log(FCS)_mvam_t ~ {sociodemographic random effect variables} + log(FCS)_mvam_t-1` . We call this set-up to be the 'auto-regressive' version.

Furthermore, the round of information used to compute the above offset is discarded. The reason for doing so is obvious; we will be effectively returning the results from the F2F survey.

```
#source code for Bayesian Sequential Survey Estimator
#assume current working directory is repository root
source('./R/seqSvyEst.R')

#Estimate models over time windows (using cross-sectional modelling approach)
seq.mdls <- seq.estimator.XS(mvamDF[,c(colsmVAM,'IPCclass','logFCS.prior')],bas
e.mdl.stan,mvamDF$ObsDate)
```

Simulate posterior on DHS data

Now with the posterior (updated with mvam data) models in hand for each time-step we simulate the results on the DHS dataset, providing the most representative breakdown of the socio-demographic characteristics of our population withstanding the existence of a proper census.

The function returns the probability distribution function and quantile function of FCS by strata by time-window

```
#Return estimates via simulating models on DHS data for each window
rslt.DHS <- mdl.simulator(seq.mdls,offset.mdl.stan,dmgDF[,c(colsDHS,'IPCclass')
],'ADM1_NAME','wgt')
```

Result by prevalence of food consumption categories

```
##   ADM1_NAME Start.Date   End.Date  FCSMean CI.Hi.FCS CI.Lo.FCS   PctPoor
## 1     Abyan 2018-02-26 2018-03-28 39.40743  45.24349  34.14754 0.4789665
## 2     Abyan 2018-03-05 2018-04-04 43.62123  49.97993  38.22887 0.4187277
## 3     Abyan 2018-05-14 2018-06-13 48.04118  53.94819  42.60540 0.3597973
## 4     Abyan 2018-04-23 2018-05-23 42.66908  47.99200  37.94645 0.4219921
## 5     Abyan 2018-03-12 2018-04-11 44.94083  50.48076  39.60920 0.3933568
## 6     Abyan 2018-03-19 2018-04-18 45.40297  50.20391  40.32024 0.3806525
##   CI.Hi.Poor CI.Lo.Poor   PctBrdr CI.Hi.Brdr CI.Lo.Brdr    Pop  NumPoor
## 1  0.5544041  0.4090557 0.6811747  0.7491103  0.6128763 568000 272053.0
## 2  0.4888967  0.3494521 0.6292766  0.6973284  0.5623274 568000 237837.3
## 3  0.4194724  0.3035233 0.5748220  0.6338494  0.5109731 568000 204364.9
## 4  0.4845644  0.3647365 0.6362603  0.6980079  0.5778629 568000 239691.5
## 5  0.4631563  0.3301679 0.6098235  0.6738962  0.5442604 568000 223426.6
## 6  0.4396604  0.3255414 0.6003439  0.6584308  0.5460853 568000 216210.6
##    NumBrdr
## 1 386907.2
## 2 357429.1
## 3 326498.9
## 4 361395.9
## 5 346379.8
## 6 340995.3
```

Result by quantile

```
## # A tibble: 6 x 201
## # Groups:   ADM1_NAME, End.Date [6]
##   ADM1_NAME End.Date   ` 0.5%` ` 1.0%` ` 1.5%` ` 2.0%` ` 2.5%` ` 3.0%`
##   <chr>     <date>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Abyan     2018-03-28    22.4    22.6    22.9    23.1    23.2    23.6
## 2 Abyan     2018-04-04    24.7    25.3    25.5    25.9    26.1    26.3
## 3 Abyan     2018-04-11    25.3    25.9    26.3    26.6    27.0    27.2
## 4 Abyan     2018-04-18    26.1    26.5    27.1    27.3    27.6    27.7
## 5 Abyan     2018-04-25    24.6    25.2    25.6    26.1    26.3    26.6
## 6 Abyan     2018-05-02    24.0    24.3    24.3    24.4    24.5    24.7
## # ... with 193 more variables: ` 3.5%` <dbl>, ` 4.0%` <dbl>, `
## #   4.5%` <dbl>, ` 5.0%` <dbl>, ` 5.5%` <dbl>, ` 6.0%` <dbl>, `
## #   6.5%` <dbl>, ` 7.0%` <dbl>, ` 7.5%` <dbl>, ` 8.0%` <dbl>, `
## #   8.5%` <dbl>, ` 9.0%` <dbl>, ` 9.5%` <dbl>, `10.0%` <dbl>,
## #   `10.5%` <dbl>, `11.0%` <dbl>, `11.5%` <dbl>, `12.0%` <dbl>,
## #   `12.5%` <dbl>, `13.0%` <dbl>, `13.5%` <dbl>, `14.0%` <dbl>,
## #   `14.5%` <dbl>, `15.0%` <dbl>, `15.5%` <dbl>, `16.0%` <dbl>,
## #   `16.5%` <dbl>, `17.0%` <dbl>, `17.5%` <dbl>, `18.0%` <dbl>,
## #   `18.5%` <dbl>, `19.0%` <dbl>, `19.5%` <dbl>, `20.0%` <dbl>,
## #   `20.5%` <dbl>, `21.0%` <dbl>, `21.5%` <dbl>, `22.0%` <dbl>,
## #   `22.5%` <dbl>, `23.0%` <dbl>, `23.5%` <dbl>, `24.0%` <dbl>,
## #   `24.5%` <dbl>, `25.0%` <dbl>, `25.5%` <dbl>, `26.0%` <dbl>,
## #   `26.5%` <dbl>, `27.0%` <dbl>, `27.5%` <dbl>, `28.0%` <dbl>,
## #   `28.5%` <dbl>, `29.0%` <dbl>, `29.5%` <dbl>, `30.0%` <dbl>,
## #   `30.5%` <dbl>, `31.0%` <dbl>, `31.5%` <dbl>, `32.0%` <dbl>,
## #   `32.5%` <dbl>, `33.0%` <dbl>, `33.5%` <dbl>, `34.0%` <dbl>,
## #   `34.5%` <dbl>, `35.0%` <dbl>, `35.5%` <dbl>, `36.0%` <dbl>,
## #   `36.5%` <dbl>, `37.0%` <dbl>, `37.5%` <dbl>, `38.0%` <dbl>,
## #   `38.5%` <dbl>, `39.0%` <dbl>, `39.5%` <dbl>, `40.0%` <dbl>,
## #   `40.5%` <dbl>, `41.0%` <dbl>, `41.5%` <dbl>, `42.0%` <dbl>,
## #   `42.5%` <dbl>, `43.0%` <dbl>, `43.5%` <dbl>, `44.0%` <dbl>,
## #   `44.5%` <dbl>, `45.0%` <dbl>, `45.5%` <dbl>, `46.0%` <dbl>,
## #   `46.5%` <dbl>, `47.0%` <dbl>, `47.5%` <dbl>, `48.0%` <dbl>,
## #   `48.5%` <dbl>, `49.0%` <dbl>, `49.5%` <dbl>, `50.0%` <dbl>,
## #   `50.5%` <dbl>, `51.0%` <dbl>, `51.5%` <dbl>, `52.0%` <dbl>,
## #   `52.5%` <dbl>, `53.0%` <dbl>, …
```

Result by probability distribution

```
## # A tibble: 6 x 72
## # Groups:   ADM1_NAME, End.Date [6]
##   ADM1_NAME End.Date       `11`     `12`     `13`     `14`     `15`
##   <chr>     <date>        <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Abyan     2018-03-28 0.       5.26e-16 1.11e-15 3.62e-16 1.27e-16
## 2 Abyan     2018-04-04 0.       5.02e-16 2.01e-15 0.       0.
## 3 Abyan     2018-04-11 1.84e-15 0.       4.44e-15 1.53e-15 0.
## 4 Abyan     2018-04-18 0.       7.60e-16 5.36e-16 4.65e-15 3.27e-15
## 5 Abyan     2018-04-25 7.54e-16 2.34e-15 1.39e-15 6.25e-15 3.70e-15
## 6 Abyan     2018-05-02 0.       1.11e-15 1.53e-15 0.       9.24e-16
## # ... with 65 more variables: `16` <dbl>, `17` <dbl>, `18` <dbl>,
## #   `19` <dbl>, `20` <dbl>, `21` <dbl>, `22` <dbl>, `23` <dbl>,
## #   `24` <dbl>, `25` <dbl>, `26` <dbl>, `27` <dbl>, `28` <dbl>,
## #   `29` <dbl>, `30` <dbl>, `31` <dbl>, `32` <dbl>, `33` <dbl>,
## #   `34` <dbl>, `35` <dbl>, `36` <dbl>, `37` <dbl>, `38` <dbl>,
## #   `39` <dbl>, `40` <dbl>, `41` <dbl>, `42` <dbl>, `43` <dbl>,
## #   `44` <dbl>, `45` <dbl>, `46` <dbl>, `47` <dbl>, `48` <dbl>,
## #   `49` <dbl>, `50` <dbl>, `51` <dbl>, `52` <dbl>, `53` <dbl>,
## #   `54` <dbl>, `55` <dbl>, `56` <dbl>, `57` <dbl>, `58` <dbl>,
## #   `59` <dbl>, `60` <dbl>, `61` <dbl>, `62` <dbl>, `63` <dbl>,
## #   `64` <dbl>, `65` <dbl>, `66` <dbl>, `67` <dbl>, `68` <dbl>,
## #   `69` <dbl>, `70` <dbl>, `71` <dbl>, `72` <dbl>, `73` <dbl>,
## #   `74` <dbl>, `75` <dbl>, `76` <dbl>, `77` <dbl>, `78` <dbl>,
## #   `79` <dbl>, `80` <dbl>
```

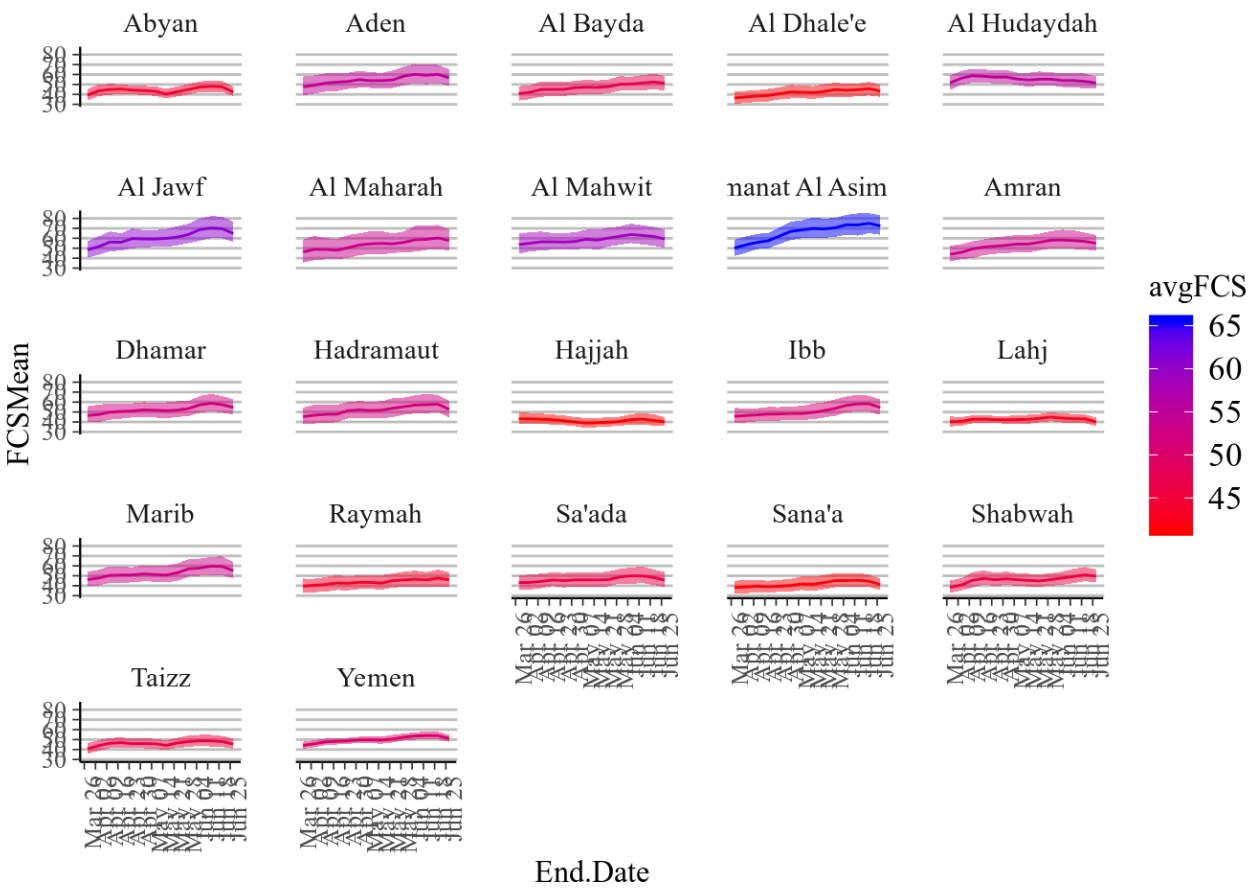
Oooh, it’s plotting time!

Now with results in hand, we can make some pretty plots :)

Plot A: Evolution of mean FCS over time by Governorate

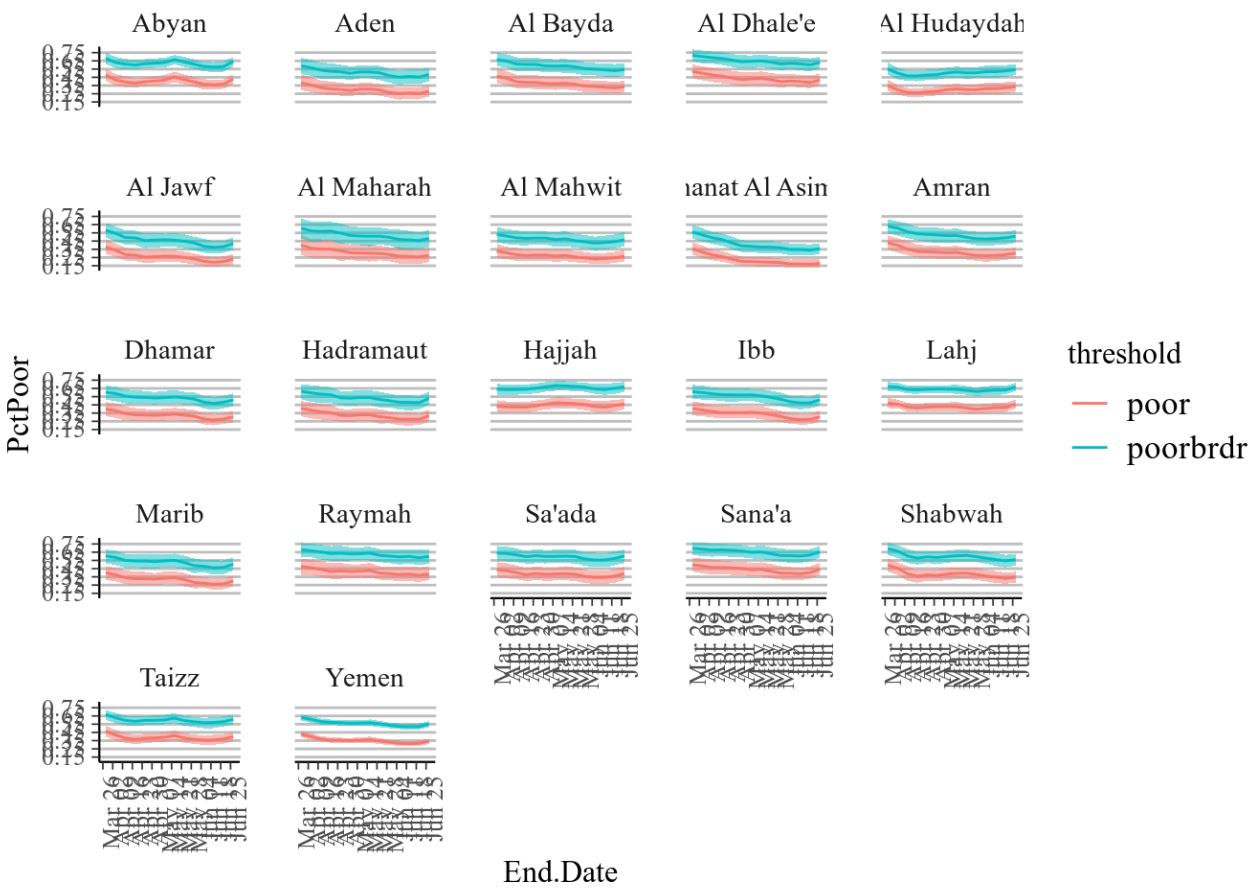
```
library(dplyr)
library(ggplot2)
library(reshape2)
library(scales)
library(stringr)
library(stringi)
#function to rowbind when column names are different
force_bind = function(df1, df2) {
  colnames(df2) = colnames(df1)
  bind_rows(df1, df2)
}

ggplot(data=rslt.DHS$str.est %>% group_by(ADM1_NAME) %>% mutate(avgFCS=mean(FCS
Mean)),
       mapping=aes(x=End.Date,FCSMean,CI.Lo.FCS,CI.Hi.FCS,color=avgFCS,fill=avg
FCS))+
  geom_line(aes(y = FCSMean))+
  geom_ribbon(aes(ymin=CI.Lo.FCS, ymax=CI.Hi.FCS),alpha=0.5,color=NA)+facet_wra
p(~ADM1_NAME)+
  scale_fill_gradient(low='red',high='blue')+scale_colour_gradient(low='red',hi
gh='blue')+
  scale_x_date(breaks = pretty_breaks(14))+
  scale_y_continuous(minor_breaks = seq(80,80,5),breaks = seq(30,80,10))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        panel.grid.major.y = element_line(colour="grey", size=0.5))
```



Plot B: Evolution of prevalence of food consumption groups over time by Governorate

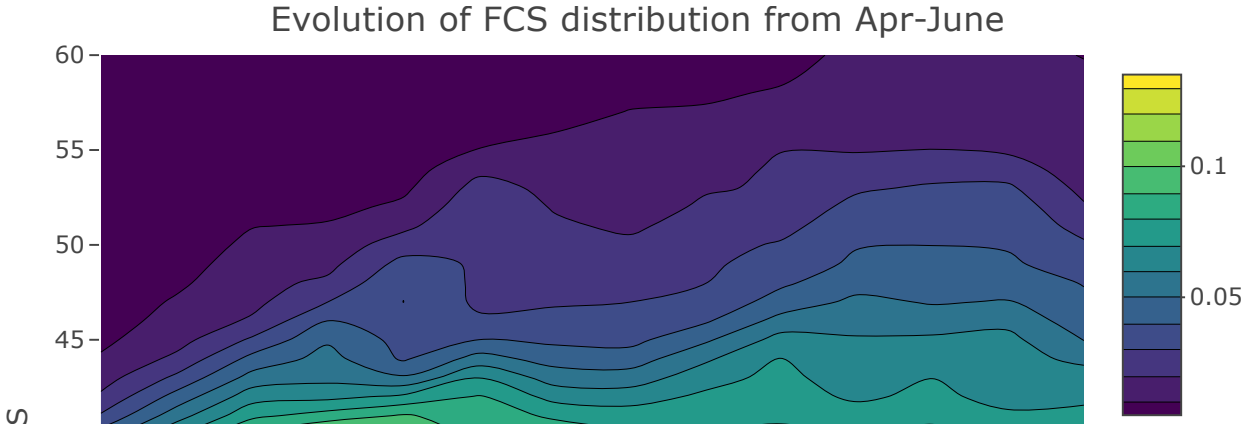
```
ggplot(force_bind(cbind(rslt.DHS$str.est[,c(1,3,7,8,9)],'threshold'='poor'),
                  cbind(rslt.DHS$str.est[,c(1,3,10,11,12)],'threshold'='poorbrd
r')),
       mapping=aes(x=End.Date,PctPoor,CI.Hi.Poor,CI.Lo.Poor,fill=threshold,colo
r=threshold))+
  geom_line(aes(y = PctPoor))+
  geom_ribbon(aes(ymin=CI.Lo.Poor,ymax=CI.Hi.Poor),alpha=0.5,show.legend=FALSE,
colour=NA)+facet_wrap(~ADM1_NAME)+
  scale_x_date(breaks = pretty_breaks(14))+
  scale_y_continuous(minor_breaks = seq(15,75,5)/100,breaks = seq(15,75,10)/100
)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        panel.grid.major.y = element_line(colour="grey", size=0.5))
```



Plot C: Evolution of probability distribution of FCS for all of Yemen over time

```
library(plotly)

df <- rslt.DHS$pdf.est %>% filter(ADM1_NAME=='Yemen') %>% select(-ADM1_NAME)
plot_ly(x=df$End.Date,y=c(20:60),
        z=t(as.matrix(df[,as.character(c(20:60))]))/10000,type='contour',autoco
ntour=TRUE) %>%
  layout(xaxis=list(title='Date'),yaxis=list(title='FCS'),title='Evolution of F
CS distribution from Apr-June')
```



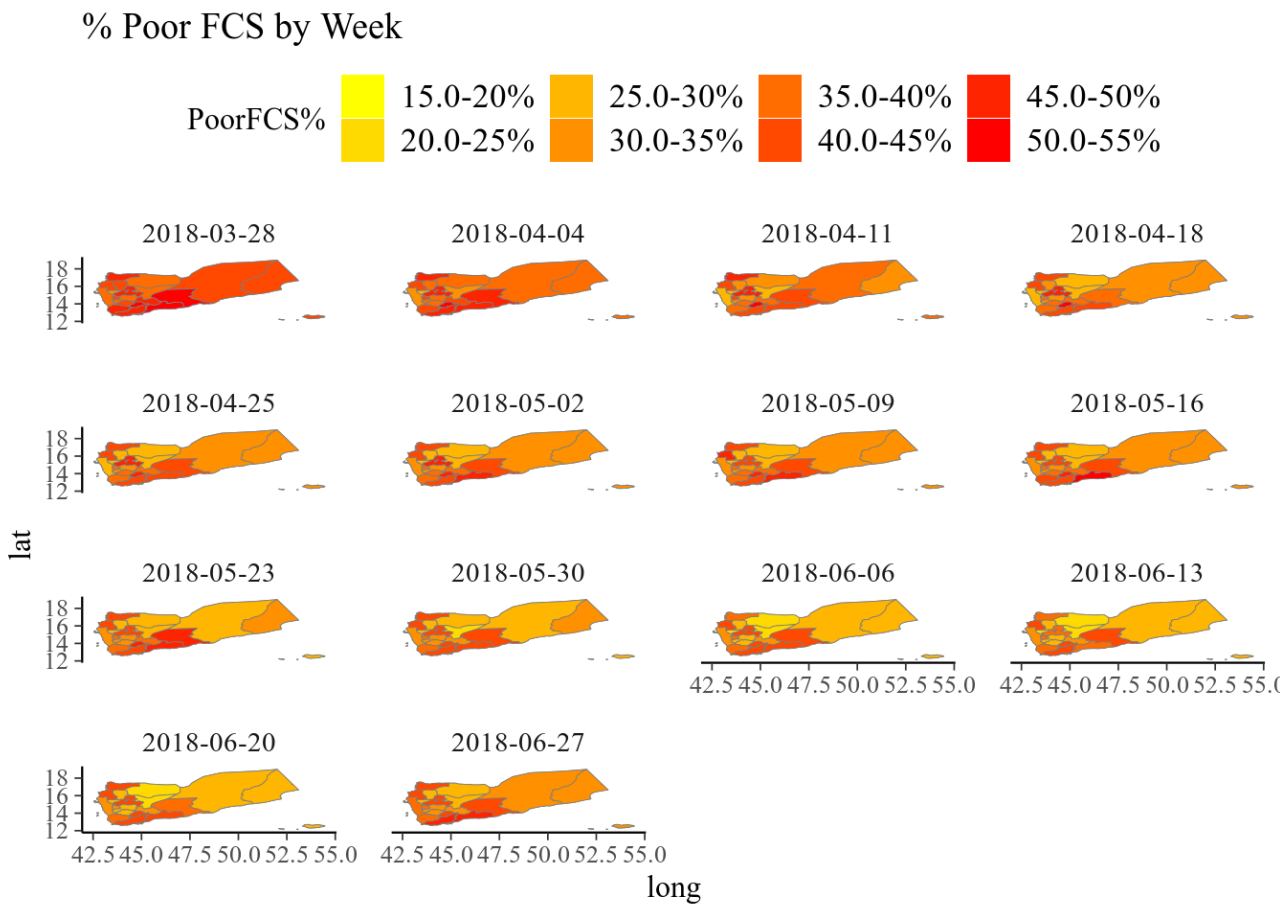
```
rm(df)
```

Plot D: Prevalence map of Poor Food Consumption over time

```
#convert spatial polygons dataframe
shp.DF <- fortify(ymnGDF) %>%
  left_join(data_frame(id=rownames(ymnGDF@data), name=ymnGDF@data$ADM1_NAME)) %
>%
  select(-id) %>% rename(id=name)

#assemble dataframe
dat.DF <- rslt.Base$str.est[rslt.Base$str.est$ADM1_NAME!='Yemen',c('ADM1_NAME',
'End.Date','PctPoor','NumPoor')]
colnames(dat.DF)[1] <- 'id'
#reformat so prevalence is in buckets of 15% to 55% by 5% increments
breaks.prv <- seq(15,55,5)
breaks.lbl <- sprintf("%2.1f-%s", breaks.prv, percent(lead(breaks.prv/100))) %>
%
  stri_replace_all_regex(c("^0.0", "-NA%"), c("0", "%"), vectorize_all=FALSE) %
>% head(-1)
dat.DF <- dat.DF %>% mutate(`PoorFCS%`=cut(PctPoor,breaks.prv/100,breaks.lbl))

#plot
ggplot()+
  geom_polygon(data=shp.DF,aes(x=long,y=lat, group=group),fill='white',color='#
7f7f7f',size=0.15)+
  geom_map(data=dat.DF, map=shp.DF,aes(map_id=id, fill=`PoorFCS%`),color="#7f7f
7f", size=0.15)+
  scale_fill_manual(values=colorRampPalette(c("yellow", "red"))(length(breaks.l
bl)))+
  guides(fill=guide_legend(override.aes=list(colour=NA)))+
  labs(title="% Poor FCS by Week")+
  facet_wrap(~End.Date)+
  theme(legend.position="top")
```

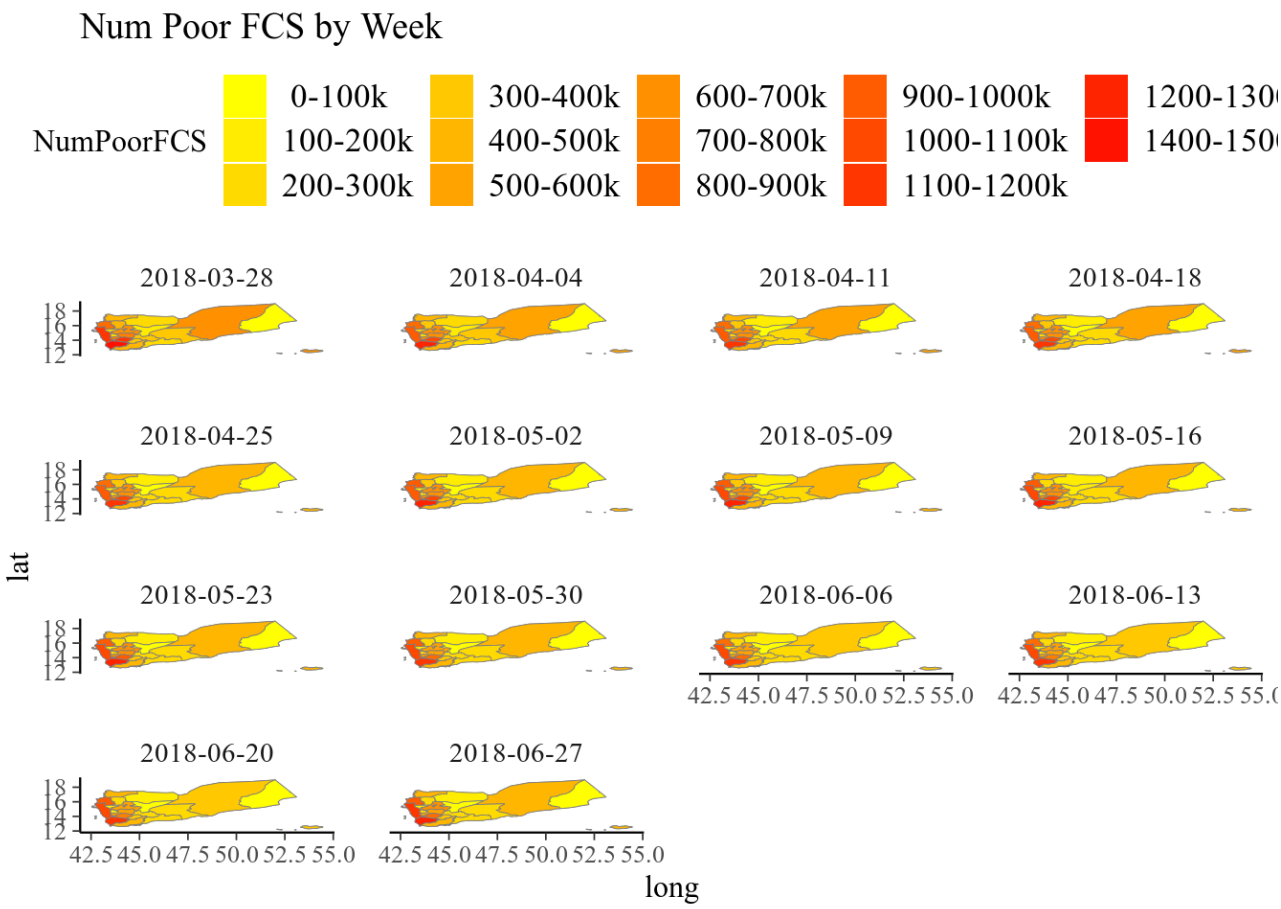



```
rm(breaks.lbl,breaks.prv)
```

Plot E: Convert prevalence to number of people in need

```
breaks.num <- seq(0,1500000,100000)
breaks.lbl <- sprintf("%2.0f-%2.0fk", breaks.num/1000, lead(breaks.num)/1000) %
>% head(-1)
dat.DF <- dat.DF %>% mutate(`NumPoorFCS`=cut(NumPoor,breaks.num,breaks.lbl))

ggplot()+
  geom_polygon(data=shp.DF,aes(x=long,y=lat, group=group),fill='white',color='#
7f7f7f',size=0.15)+
  geom_map(data=dat.DF, map=shp.DF,aes(map_id=id, fill=`NumPoorFCS`),color="#7f
7f7f", size=0.15)+
  scale_fill_manual(values=colorRampPalette(c("yellow", "red"))(length(breaks.l
bl)))+
  guides(fill=guide_legend(override.aes=list(colour=NA)))+
  labs(title="Num Poor FCS by Week")+
  facet_wrap(~End.Date)+
  theme(legend.position="top")
```




```
rm(breaks.lbl,breaks.num)

rm(dat.DF)
```

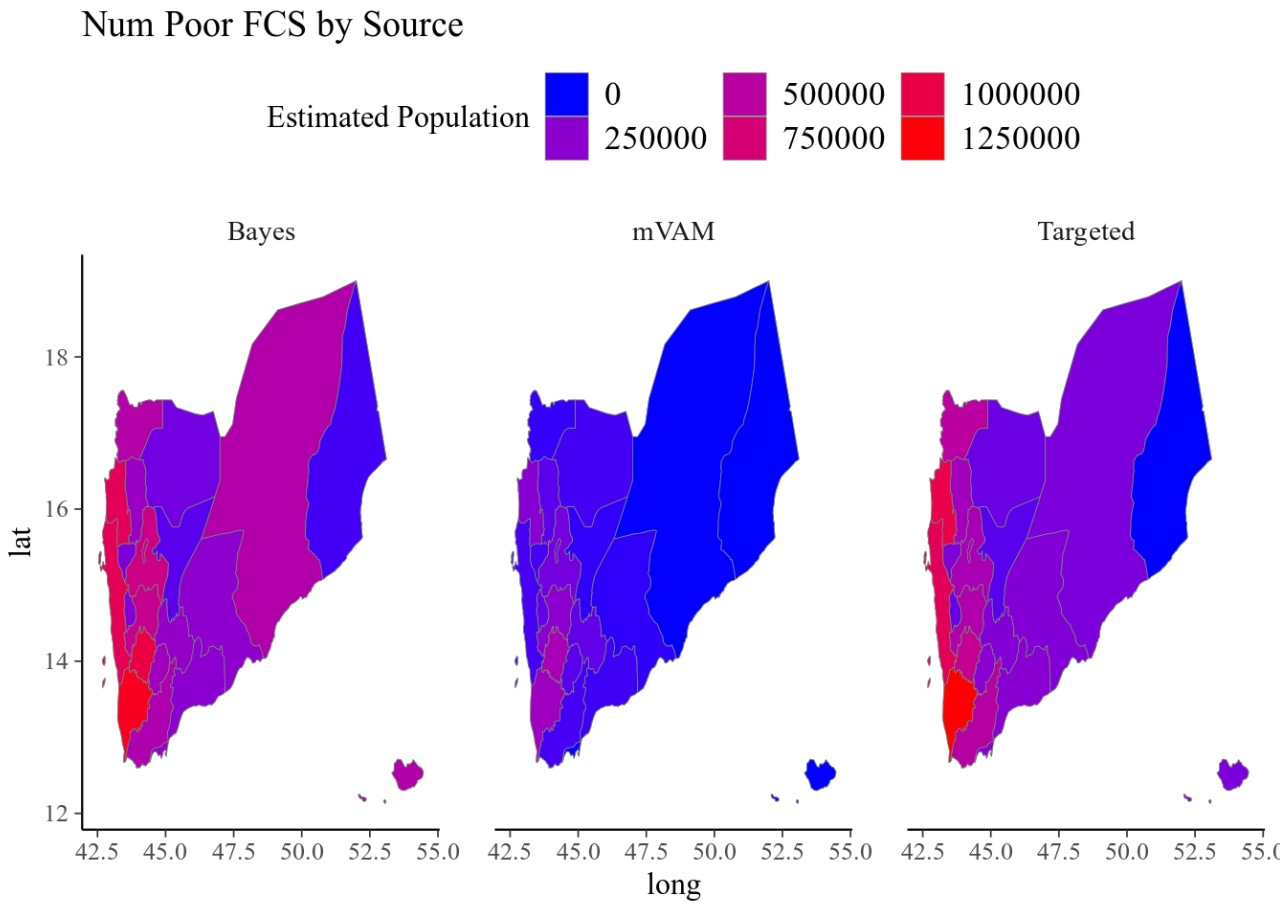
Plot F: Comparison of traditional mVAM estimates with Bayesian Sequential Update

Now we compare the above results to what we get through our traditional mVAM estimates. These estimates use the standard Horowtiz-Thompson estimator (i.e. weighted average) for stratified random sample survey designs with first-stage selection weights determined by number of phones owned by the household and a final post-stratification weighting by IDP status. Each survey round is discrete.

For ground-truth we use the governorate-level food-distribution targets for May 2018. These are typically tabulated by the estimated prevalence rate of poor food consumption coming from ad-hoc face-to-face convenience surveys conducted quarterly. Hence, for comparison we use the April mVAM data round and the Bayesian Sequential Estimates from the period beteween 2nd of April and 2nd of May.

```
##dataset preloaded as cmpr.DF

ggplot()+
  geom_polygon(data=shp.DF,aes(x=long,y=lat, group=group),fill='white',color='#
7f7f7f',size=0.15)+
  geom_map(data=cmpr.DF[cmpr.DF$Type!='EFSA',],
           map=shp.DF,aes(map_id=id, fill=NumPop),color="#7f7f7f", size=0.15)+
  scale_fill_gradient(low='blue',high='red',
                      guide = guide_legend(
                        label.theme = element_text(angle = 90)))+
  guides(fill=guide_legend(title='Estimated Population'))+
  labs(title="Num Poor FCS by Source")+
  facet_wrap(~Type)+
  theme(legend.position="top")
```



Plot G: Percent difference with Targeted amounts for mVAM vs Bayesian Sequential Update

```
ggplot()+
  geom_polygon(data=shp.DF,aes(x=long,y=lat, group=group),fill='white',color='#
7f7f7f',size=0.15)+
  geom_map(data=cmpr.DF[!(cmpr.DF$Type %in% c('Targeted','EFSA')),],
           map=shp.DF,aes(map_id=id, fill=PctDiff*100),color="#7f7f7f", size=0.
15)+
  scale_fill_gradient2(low='red',high='blue',mid='white',
                       guide = guide_legend(
                         label.theme = element_text(angle = 90)))+
  guides(fill=guide_legend(title='%Diff'))+
  labs(title="% Difference with May Food Distribution Targets")+
  facet_wrap(~Type,ncol=2)+
  theme(legend.position="top")
```