Model Validation - Code for Analyses and Plots

Leah R. Johnson and Erin Mordecai

Dec. 12, 2016

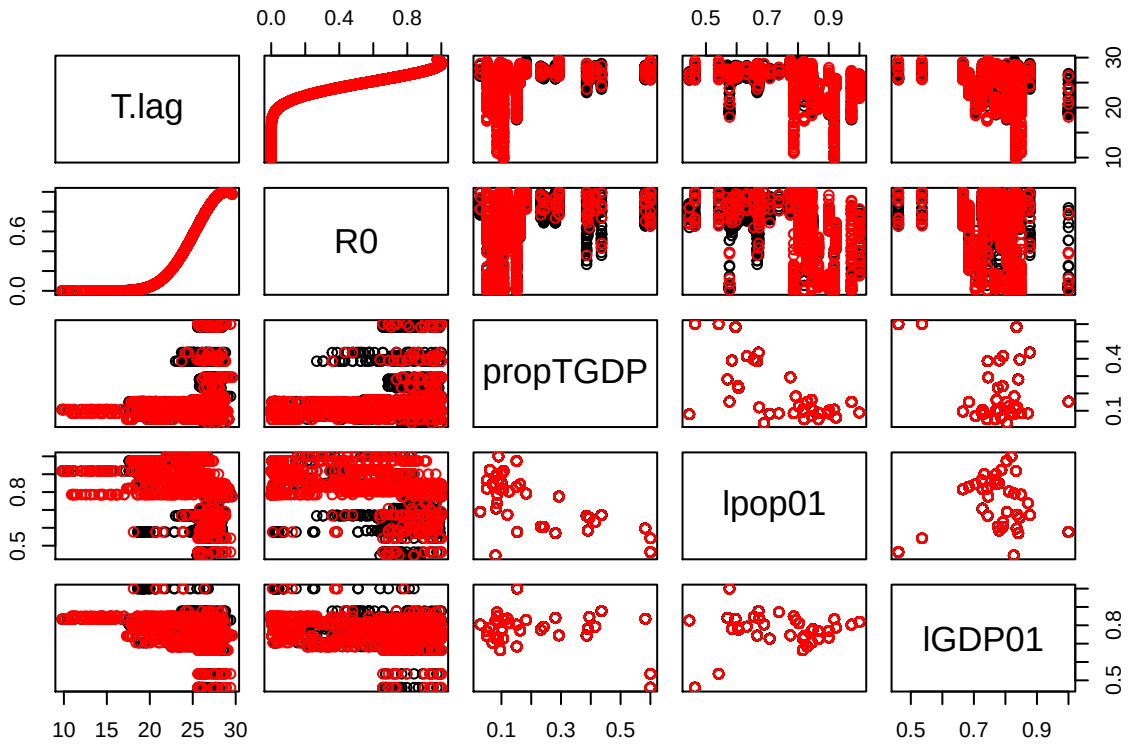
```
library("statmod")
library("visreg")

source("inc_R0_validation_dataprep.R")

dat<-read.csv("inc_dat_clean.csv", header=TRUE)
cc.dat<-read.csv("combinedGDP.csv", header=TRUE)

w.lag<- c(5, 10, 15)+1

## Processing the lags. Let's start with an interval centered around
## some number of weeks back, and smoothed with 2 weeks on either
## side. So this is the filter.

f<-c(1,2,3,2,1)/9

## Now build the data based using the make.lag.data in the dataprep
## file sourced above.
#d.l5<-make.lag.data(dat, f=f, w.lag=w.lag[1], cc.dat=cc.dat)
d.l10<-make.lag.data(dat, f=f, w.lag=w.lag[2], cc.dat=cc.dat)
#d.l15<-make.lag.data(dat, f=f, w.lag=w.lag[3], cc.dat=cc.dat)
```

```
d<-d.l10
d$DEN<-d$virus=="DENV"
d$lpTGDP<-log(d$percTGDP)
d$propTGDP<-d$percTGDP/100
d$lpop<-log(d$population)
d$l.inc<-log(d$pa*d$inc+1)

d$lpop01<-d$lpop/max(d$lpop)
d$lGDP01<-d$lGDP/max(d$lGDP)

## we consider a few combinations of R0 measures and population
d$lR0.alt<-log(d$R0*d$population)
d$lR0.GR0.alt<-log(d$R0.GR0*d$population)

d$R0.alt<-d$R0*log(d$population)
d$R0.GR0.alt<-d$R0.GR0*log(d$population)
```

Preliminary data visualization

First we visually explore our covariates. Notice there is quite a bit of correlation (colinearity) between our environmental (T.lag, R0) and socio-economic variables, and between the socioeconomic variables, especally tourism GDP% and log population. This isn't causal, but this means we have to take care when interpretting the coefficients in the regressions

```
pairs(d[,c(7,12,15, 18,19)], col=d$DEN+1)
```




```
round(cor(d[,c(7,12,15, 18,19)]), digits=3)
```

```
##           T.lag     R0 propTGDP lpop01 lGDP01
## T.lag     1.000  0.926    0.325 -0.379 -0.257
## R0        0.926  1.000    0.368 -0.442 -0.229
## propTGDP  0.325  0.368    1.000 -0.683 -0.140
## lpop01   -0.379 -0.442   -0.683  1.000  0.034
## lGDP01   -0.257 -0.229   -0.140  0.034  1.000
```

Models for the Presence/Absence Data

```
## population only
reg<-glm(pa ~ lpop*DEN - DEN, family=binomial, data=d)

## socio factors only
reg0<-glm(pa ~ (propTGDP + lGDP +lpop)*DEN - DEN, family=binomial, data=d)

## prob R0>0
reg0A<-glm(pa ~ (R0.GR0+lpop)*DEN - DEN, family=binomial, data=d)

## alt prob R0>0 + socio
reg1<-glm(pa ~ (R0.GR0 + propTGDP+ lGDP+lpop)*DEN - DEN,
          family=binomial, data=d)

## model considering a subset of the predictors and combos from reg1
```

```
reg1.sub<-glm(pa ~ R0.GR0 + propTGDP + (lGDP+lpop)*DEN - DEN,
          family=binomial, data=d)

## Combines R0.GR0 with log pop
reg0A.alt<-glm(pa ~ R0.GR0.alt*DEN - DEN,
          family=binomial, data=d)


bics <- c(BIC(reg), BIC(reg0), BIC(reg0A), BIC(reg1), BIC(reg1.sub), BIC(reg0A.alt))
```

Model Comparisons (within sample)

```
bics
```

```
## [1] 1844.424 1619.950 1628.474 1528.249 1518.631 1698.811
```

```
ebics<-exp(-0.5*(bics-min(bics)))
probs <- round(ebics/sum(ebics), 5)
probs
```

```
## [1] 0.00000 0.00000 0.00000 0.00809 0.99191 0.00000
```

```
## Proportion of Deviance  explained, compared to null model
D2s<-c(D.sqr(reg), D.sqr(reg0), D.sqr(reg0A), D.sqr(reg1), D.sqr(reg1.sub), D.sqr(reg0A.alt))
round(D2s, 3)
```

```
## [1] 0.382 0.469 0.461 0.505 0.503 0.432
```

```
## Plots residual diagnostics
models<-list(reg=reg, reg0=reg0, reg0A=reg0A, reg1=reg1, reg1.sub=reg1.sub)
## randomized quantile residuals
for(i in 1:length(models)){
    r<-models[[i]]
    qr<-qresiduals(r)
    par(mfrow=c(1,2))
    plot(r$fitted, qr, col=as.numeric(d$pa)+1, main=names(models)[i], xlab="fitted")
    abline(h=0)
    qqnorm(qr)
    abline(0,1, col=2)
}
```

Reg1.sub is the top model, by BIC, for the full data set. We re-fit with all the predictors scaled, so we can see the relative size of the parameters for important parameters. Note that due to multi-colinearity in the predictors, we can't reliably interpret these parameters as relating to the relative influence of various predictors (or even as the marginal impact of a predictor).

```
summary(glm(pa ~ R0.GR0 + propTGDP + (lGDP01+lpop01)*DEN - DEN,
          family=binomial, data=d))
```

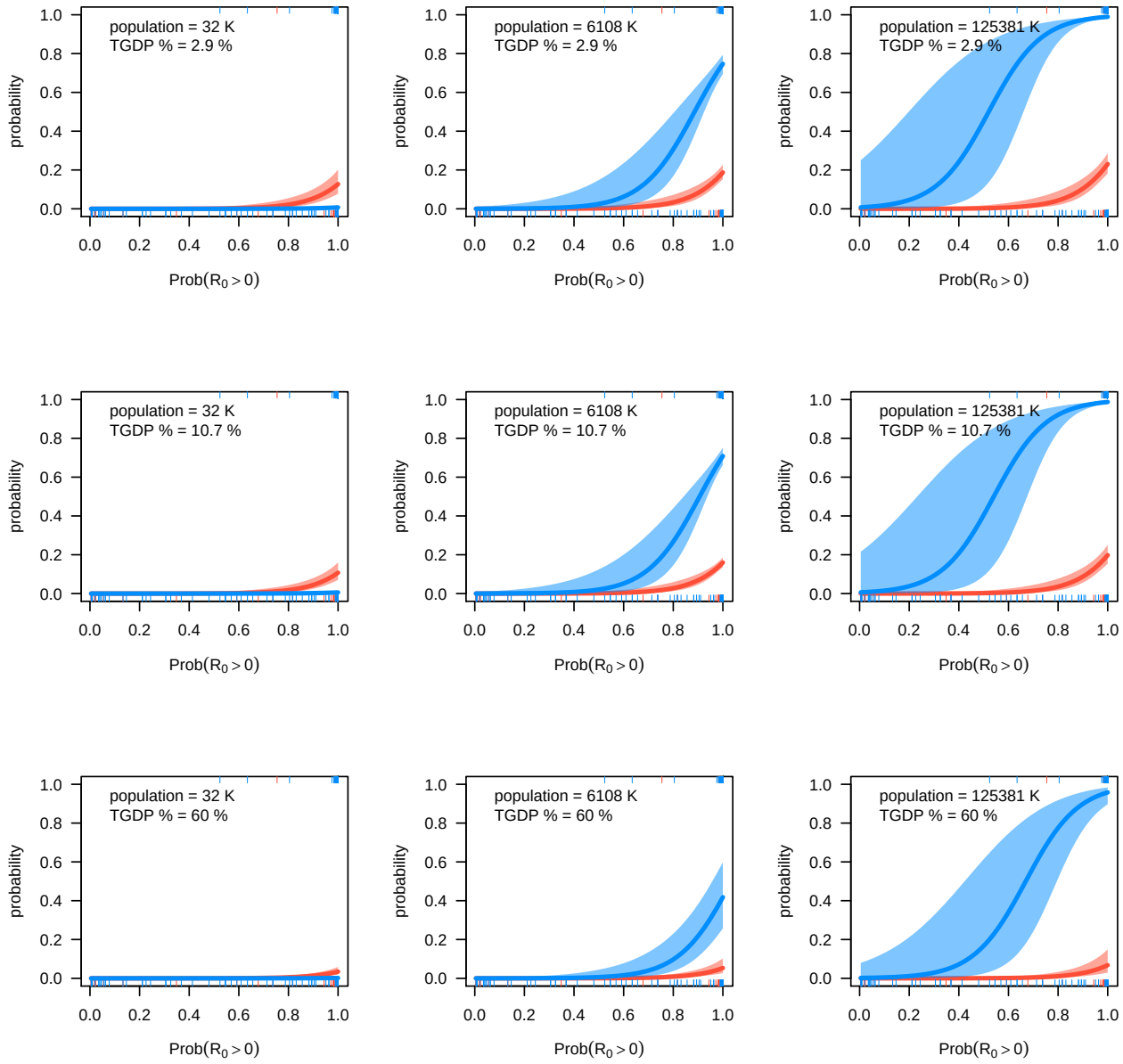
```
##
## Call:
## glm(formula = pa ~ R0.GR0 + propTGDP + (lGDP01 + lpop01) * DEN -
##     DEN, family = binomial, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7807  -0.5524  -0.3528   0.3382   2.9790
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -13.3739     2.3963  -5.581 2.39e-08 ***
## R0.GR0           9.4811     1.9640   4.827 1.38e-06 ***
## propTGDP        -2.4763     0.7648  -3.238   0.0012 **
## lGDP01           1.4617     1.0834   1.349   0.1773
## lpop01           1.6628     0.7867   2.114   0.0346 *
## lGDP01:DENTRUE -17.9643     1.5560 -11.545  < 2e-16 ***
## lpop01:DENTRUE  20.2850     1.4755  13.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2948.4  on 2205  degrees of freedom
## Residual deviance: 1464.7  on 2199  degrees of freedom
## AIC: 1478.7
##
## Number of Fisher Scoring iterations: 7
```

Visualizations of the model fit.

Here are some visualizations of the model fit. First I plot the predicted response at the median log percent of GDP, across a subset of the values of proportion of GDP in toursim and across log population from the data set. The two sets of lines indicate the fit for Dengue (DEN: 1) and for the other 2 diseases (DEN: 0)

```
##png(file="PA_reg1_cond_resp_GDP7274.png", height=900, width=900, pointsize=18)
pops<-unique(d$lpop)
pops<-pops[order(pops)]

my.GDP<-c(min(d$lGDP), median(d$lGDP), max(d$lGDP))
my.TGDP<-c(min(d$propTGDP), median(d$propTGDP), max(d$propTGDP))
my.lpop<-c(pops[3], median(d$lpop), pops[35])

### Plot the model predictions with data
par(mfrow=c(3,3))
for(j in 1:3){
  for(i in 1:3){
  visreg(reg1.sub, "R0.GR0", scale="response", rug=2, by="DEN", overlay=TRUE,
         cond=list(lGDP=my.GDP[2], propTGDP=my.TGDP[j], lpop=my.lpop[i]), ylim=c(0,1),
         xlab=expression(Prob(R[0]>0)), ylab="probability", legend=FALSE)
  legend("topleft",
         c(paste("population = ", round(exp(my.lpop[i])/1000), " K",  sep=""),
         paste("TGDP % = ", round(my.TGDP[j]*100, 3), " %", sep="")),
```

```
          #paste("GDP = ", round(exp(my.GDP[2])), " USD", sep="")),
          bty="n", y.intersp=1)
  }
}
```

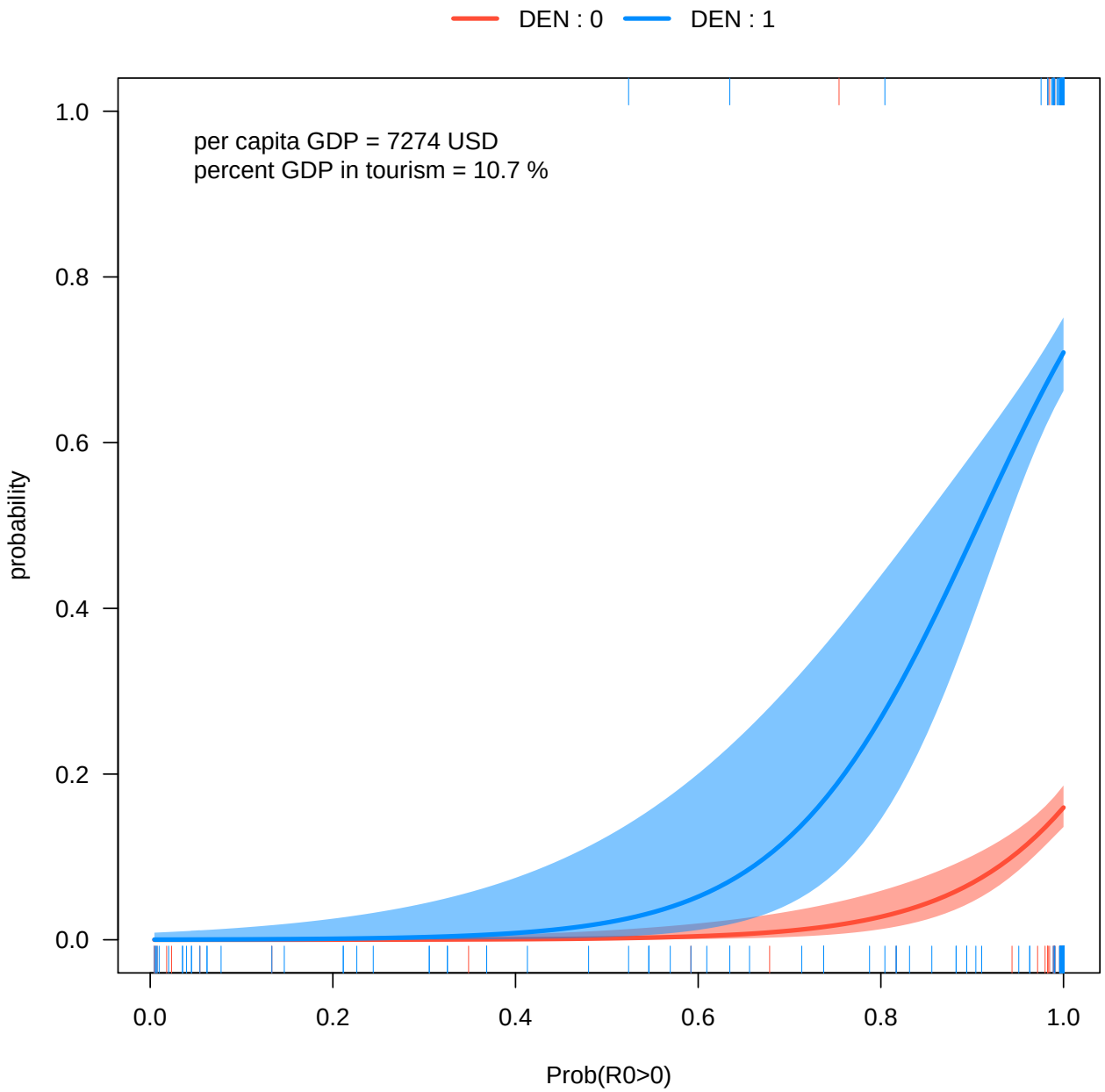



```
##dev.off()
```

These are the response surface for the Dengue predictions, at median prop GDP in tourism and median GDP.

Here is a single plot at median GDP and median percent of tourism in GDP, averaging across population sizes. The data are included in this figure as tick marks on the top an bottom of the figure. Red is for chik/zika, blue for dengue. Ticks on the top represent presence of transmission, on the botton absence.

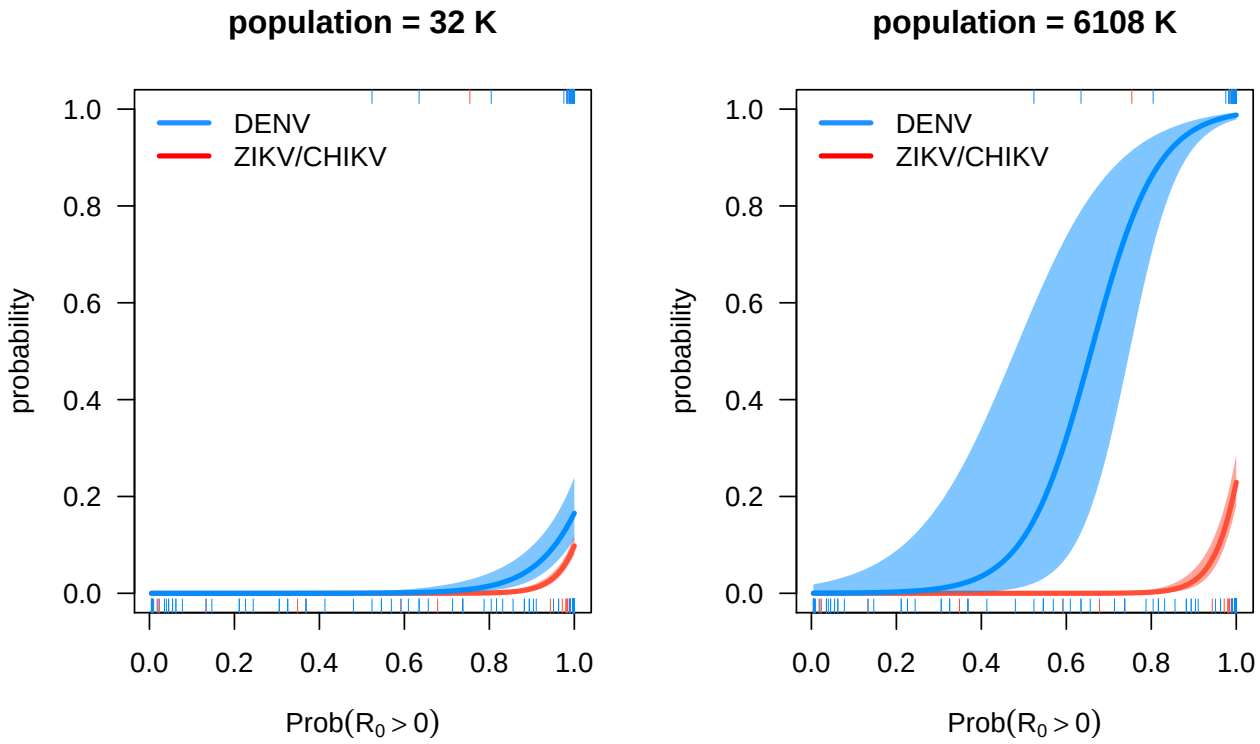
```
##png(file="PA_reg1_medians_resp.png", height=600, width=600, pointsize=16)
m.GDPS<-median(d$lGDP)
m.TGDP<-median(d$propTGDP)
par(mfrow=c(1,1))
visreg(reg1.sub, "R0.GR0", scale="response", rug=2, by="DEN", overlay=TRUE,
       cond=list(lGDP=m.GDPS, propTGDP=m.TGDP), ylim=c(0,1),
       xlab="Prob(R0>0)", ylab="probability")
legend(x=0, y=1, #"topleft",
       c(paste("per capita GDP = ", round(exp(m.GDPS)), " USD",  sep=""),
         paste("percent GDP in tourism = ", round(m.TGDP*100,1), " %",  sep="")),
        bty="n", y.intersp=1)
```



```
##dev.off()
```

We plot reg0A instead, at two levels of the log population size, one middle low and one high.

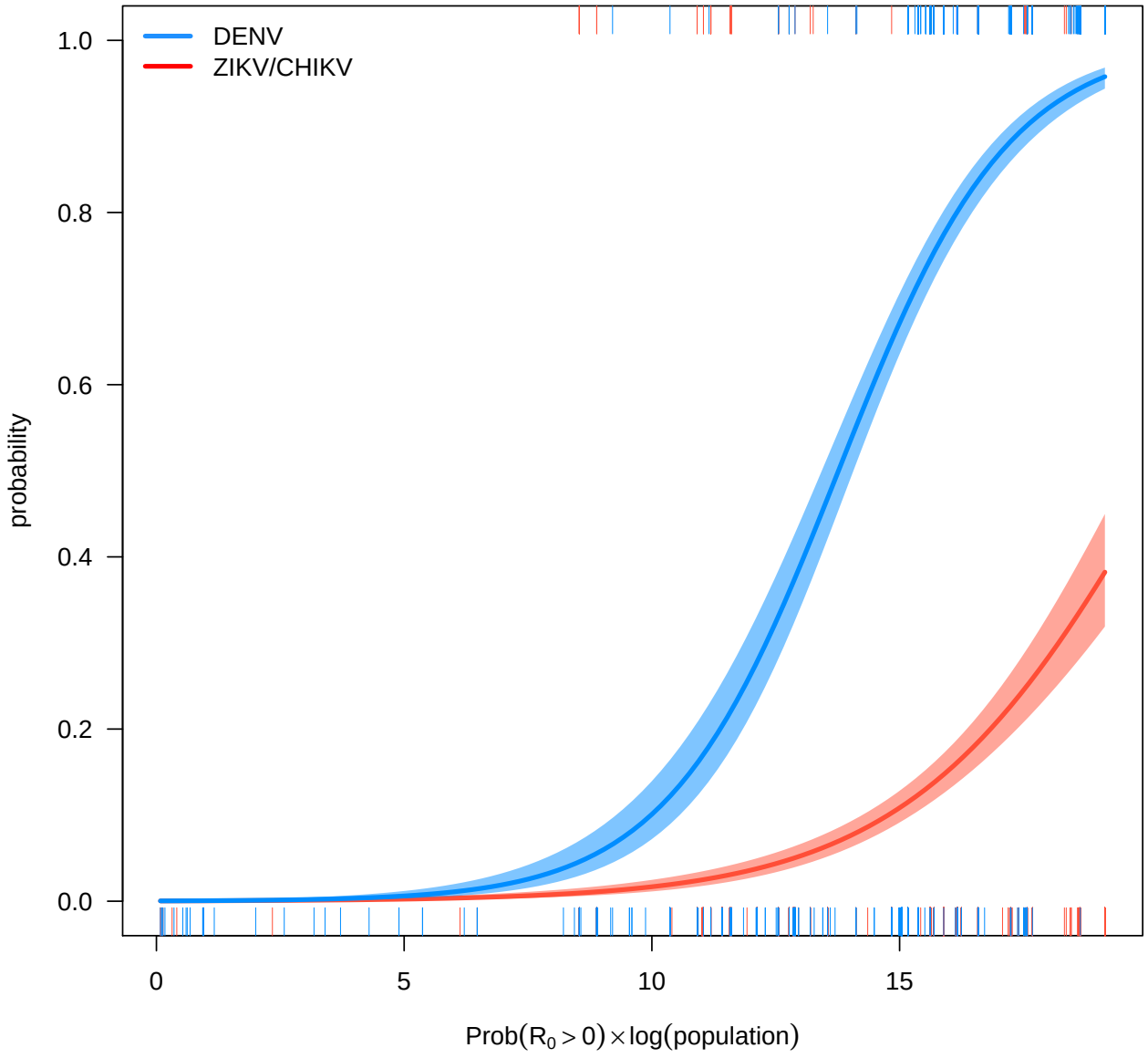
```
##png(file="PA_reg0A_resp.png", height=600, width=1000, pointsize=16)
my.lpop2<-pops[c(15,35)]

par(mfrow=c(1,2))
for(i in 1:2){
  visreg(reg0A, "R0.GR0", scale="response", rug=2, by="DEN", overlay=TRUE,
         ylim=c(0,1), cond=list(lpop=my.lpop2[i]),
         xlab=expression(Prob(R[0]>0)), ylab="probability", legend=FALSE,
         main=paste("population = ", round(exp(my.lpop[i])/1000), " K",  sep=""))
  legend("topleft",
         c("DENV", "ZIKV/CHIKV"),
         col=c("dodgerblue", "red"), lwd=c(3,3),
         bty="n", y.intersp=1)

}
```




```
par(mfrow=c(1,1))
##dev.off()
```

Since reg0A.alt uses Prob($R_0 > 0$)*log(population) as the predictor instead, we can plot the model output more simply.

```
##png(file="PA_reg0A_alt_resp.png", height=600, width=600, pointsize=16)
visreg(reg0A.alt, "R0.GR0.alt", scale="response", rug=2, by="DEN", overlay=TRUE,
       ylim=c(0,1),
       xlab=expression(Prob(R[0]>0)%*%log(population)), ylab="probability", legend=FALSE)
  legend("topleft",
         c("DENV", "ZIKV/CHIKV"),
         col=c("dodgerblue", "red"), lwd=c(3,3),
         bty="n", y.intersp=1)
```




```
##dev.off()
```

Out of sample prediction

```r
nsamps<-1000
thresh.D<-c(0.6, 0.65, 0.65, 0.75, 0.65)

thresh.CZ<-0.15

MSE.boot.D<-MSE.boot.CZ<-BIC.modprobs<-data.frame(matrix(NA, ncol=6, nrow=nsamps))
names(MSE.boot.D)<-names(MSE.boot.CZ)<-names(BIC.modprobs)<-c("m", "m0", "m0A",  "m1", "m1.sub", "m0A.a

## I'll be doing a sample stratefied based on presence/absence
PAs<-as.factor(dat$pa)
n0s<-round(0.1*summary(PAs)[1])
n1s<-round(0.1*summary(PAs)[2])

d0<-subset(d, pa==0)
d1<-subset(d, pa==1)

ndat<-dim(d)[1]-(n1s+n0s)

set.seed(123)
for(i in 1:nsamps){

    if(i%%50==0) print(paste("Bootstrap sample ", i, " of ", nsamps, sep=""))
    test<-train<-i0<-i1<-NULL
    i0<-sample(dim(d0)[1], n0s)
    i1<-sample(dim(d1)[1], n1s)

    train<-rbind(d0[-i0,], d1[-i1,])
    test<-rbind(d0[i0,], d1[i1,])

    ## only population
    reg<-glm(pa ~ lpop*DEN - DEN, family=binomial, data=train)
    pred<-predict(reg, newdata=test, type="response")
    error <- c(test$pa[test$DEN] - (pred[test$DEN] >= thresh.D[1]),
               test$pa[!test$DEN] - (pred[!test$DEN] >= thresh.CZ))

    ## only socio factors
    reg0<-glm(pa ~ (propTGDP + lGDP +lpop)*DEN - DEN, family=binomial, data=train)
    pred0<-predict(reg0, newdata=test, type="response")
    error0 <- c(test$pa[test$DEN] - (pred0[test$DEN] >= thresh.D[1]),
               test$pa[!test$DEN] - (pred0[!test$DEN] >= thresh.CZ))

    ## using Prob R0>0
    reg0A<-glm(pa ~ (R0.GR0+lpop)*DEN, family=binomial, data=train)
    pred0A<-predict(reg0A, newdata=test, type="response")
    error0A <- c(test$pa[test$DEN] - (pred0A[test$DEN] >= thresh.D[2]),
                 test$pa[!test$DEN] - (pred0A[!test$DEN] >= thresh.CZ))

    ## using alt Prob R0>0
    reg0A.alt<-glm(pa ~ R0.GR0.alt*DEN, family=binomial, data=train)
    pred0A.alt<-predict(reg0A.alt, newdata=test, type="response")
    error0A.alt <- c(test$pa[test$DEN] - (pred0A.alt[test$DEN] >= thresh.D[2]),
                 test$pa[!test$DEN] - (pred0A.alt[!test$DEN] >= thresh.CZ))
```

```
    ## Prob R0>0 + socio
    reg1<-glm(pa ~ (R0.GR0 + propTGDP+ lGDP+lpop)*DEN - DEN,
          family=binomial, data=train)
    pred1<-predict(reg1, newdata=test, type="response")
    error1 <- c(test$pa[test$DEN] - (pred1[test$DEN] >= thresh.D[4]) ,
                test$pa[!test$DEN] - (pred1[!test$DEN] >= thresh.CZ))



    reg1.sub<-glm(pa ~ R0.GR0 + propTGDP + (lGDP+lpop)*DEN - DEN,
          family=binomial, data=train)
    pred1.sub<-predict(reg1.sub, newdata=test, type="response")
    error1.sub <- c(test$pa[test$DEN] - (pred1.sub[test$DEN] >= thresh.D[5]) ,
                test$pa[!test$DEN] - (pred1.sub[!test$DEN] >= thresh.CZ))


    Dlims<-length(test$pa[test$DEN])

    MSE.boot.D[i,]<-c(mean(abs(error[1:Dlims])),
                      mean(abs(error0[1:Dlims])),
                      mean(abs(error0A[1:Dlims])),
                      mean(abs(error1[1:Dlims])),
                      mean(abs(error1.sub[1:Dlims])),
                      mean(abs(error0A.alt[1:Dlims])) )

    MSE.boot.CZ[i,]<-c(mean(abs(error[(Dlims+1):(n0s+n1s)])),
                       mean(abs(error0[(Dlims+1):(n0s+n1s)])),
                       mean(abs(error0A[(Dlims+1):(n0s+n1s)])),
                       mean(abs(error1[(Dlims+1):(n0s+n1s)])),
                       mean(abs(error1.sub[(Dlims+1):(n0s+n1s)])),
                       mean(abs(error0A.alt[(Dlims+1):(n0s+n1s)])))

    ## model probs based on BIC
    bics <- c(BIC(reg0), BIC(reg), BIC(reg0A),
              BIC(reg1), BIC(reg1.sub), BIC(reg0A.alt))
    ebics<-exp(-0.5*(bics-min(bics)))
    BIC.modprobs[i,] <- ebics/sum(ebics)

}
```

```
round(colMeans(BIC.modprobs), digits=5)
```

```
##       m      m0     m0A      m1  m1.sub m0A.alt
## 0.00000 0.00000 0.00000 0.01008 0.98992 0.00000
```

What this tells us: Looking within sample, the models that include both a measure of R_0 and the socio economic factors do better than the models that include only one or the other. If we were to look at the histograms of the BIC probs for these 1000 runs we'd see that about 30% of the time m1 comes out with model probability close to 1 and m2 has model probability of near 1 about 2/3 of the time.

Misclassification rates by the models split between the dengue and chik/zika cases

```
abs.errors.D<-colMeans(MSE.boot.D)
sd.errors.D<-apply(MSE.boot.D, MARGIN=2, sd)

# o<-order(abs.errors.D)
# abs.errors.D<-abs.errors.D[o]
# sd.errors.D<-sd.errors.D[o]


abs.errors.CZ<-colMeans(MSE.boot.CZ)
sd.errors.CZ<-apply(MSE.boot.CZ, MARGIN=2, sd)

# o<-order(abs.errors.CZ)
# abs.errors.CZ<-abs.errors.CZ[o]
# sd.errors.CZ<-sd.errors.CZ[o]

# percent accuracy (= 1 - misclassification rate)
round(1 - abs.errors.D, digits=3)
```

```
##       m      m0     m0A      m1  m1.sub m0A.alt
##   0.776   0.858   0.907   0.882   0.880   0.904
```

```
round(1 - abs.errors.CZ, digits=3)
```

```
##       m      m0     m0A      m1  m1.sub m0A.alt
##   0.691   0.566   0.662   0.658   0.649   0.663
```

So we're classifying correctly between 77 - 91% of the time for dengue and about 56 - 70% of the time for chik/zika.

Are there significant differences between these error rates? First dengue:

```
abs.errors<-abs.errors.D
sd.errors<-sd.errors.D
c( abs((abs.errors[1]-abs.errors[2])/sqrt(sd.errors[1]^2+sd.errors[2]^2)),
  abs((abs.errors[1]-abs.errors[3])/sqrt(sd.errors[1]^2+sd.errors[3]^2)),
  abs((abs.errors[1]-abs.errors[4])/sqrt(sd.errors[1]^2+sd.errors[4]^2)),
  abs((abs.errors[1]-abs.errors[5])/sqrt(sd.errors[1]^2+sd.errors[5]^2))
)
```

```
##        m        m        m        m
## 1.843848 3.118363 2.465142 2.410672
```

Then zika/chik

```
abs.errors<-abs.errors.CZ
sd.errors<-sd.errors.CZ
c( abs((abs.errors[1]-abs.errors[2])/sqrt(sd.errors[1]^2+sd.errors[2]^2)),
  abs((abs.errors[1]-abs.errors[3])/sqrt(sd.errors[1]^2+sd.errors[3]^2)),
  abs((abs.errors[1]-abs.errors[4])/sqrt(sd.errors[1]^2+sd.errors[4]^2)),
  abs((abs.errors[1]-abs.errors[5])/sqrt(sd.errors[1]^2+sd.errors[5]^2))
)
```

```
##         m         m         m         m
## 1.9596151 0.3820691 0.4801915 0.6178650
```

There aren't significant differences between the top 4 models in terms of how well they are predicting out of sample. However, the best model is significantly better then the worst in both cases. It is interesting to note that different models come out on top for predicting dengue than chik/zika, and that the top model for chik/zika (pop only) is the worst model for dengue.

Create a summary table to show the results

```
outs = data.frame('model' = c(1:6), 'bics' = round(bics, 0), 'probs' = round(probs, 3), 'D2s' = round(D
outs
```

```
##         model bics probs   D2s oos.probs abs.errors.D abs.errors.CZ
## m           1 1444 0.000 0.382      0.00        0.224         0.309
## m0          2 1644 0.000 0.469      0.00        0.142         0.434
## m0A         3 1465 0.000 0.461      0.00        0.093         0.338
## m1          4 1375 0.008 0.505      0.01        0.118         0.342
## m1.sub      5 1365 0.992 0.503      0.99        0.120         0.351
## m0A.alt     6 1455 0.000 0.432      0.00        0.096         0.337
```

```
write.csv(outs, file = "PA_outs.csv")
```

Models for the Incidence Data

```
dd<-subset(d, l.inc!=0)

m<-glm(l.inc ~ lpop*DEN-DEN,
        data=dd, family="Gamma")


m0<-glm(l.inc ~ (propTGDP + lGDP +lpop)*DEN-DEN,
        data=dd, family="Gamma")

m0B<-glm(l.inc ~ (R0 +lpop)*DEN-DEN, data=dd, family="Gamma")

m0B.alt<-glm(l.inc ~ (lR0.alt)*DEN-DEN, data=dd, family="Gamma")

m2<-glm(l.inc ~ (R0 + propTGDP + lGDP + lpop)*DEN-DEN,
        data=dd, family="Gamma")

## dropping the one non-significant interaction term

m2.sub<-glm(l.inc ~ (R0 + propTGDP + lpop)*DEN + lGDP -DEN,
        data=dd, family=Gamma())

bics<-c(BIC(m), BIC(m0), BIC(m0B), BIC(m2), BIC(m2.sub), BIC(m0B.alt))
```

```
## Plots residual diagnostics
models<-list(m0=m0, m0B=m0B, m2.sub=m2.sub)
## randomized quantile residuals
```

```
for(i in 1:length(models)){
    r<-models[[i]]
    qr<-qresiduals(r)
    par(mfrow=c(1,2))
    plot(r$fitted, qr, col=as.numeric(dd$DEN)+1, main=names(models)[i], xlab="fitted")
    abline(h=0)
    qqnorm(qr)
    abline(0,1, col=2)
}
```

And here are the BICs, etc, for the incidence models.

```
bics
```

```
## [1] 2859.518 2818.669 2724.617 2660.232 2654.406 2746.738
```

```
eBIC <- exp(-0.5*(bics-min(bics)))
round(probs <- eBIC/sum(eBIC), 5)
```

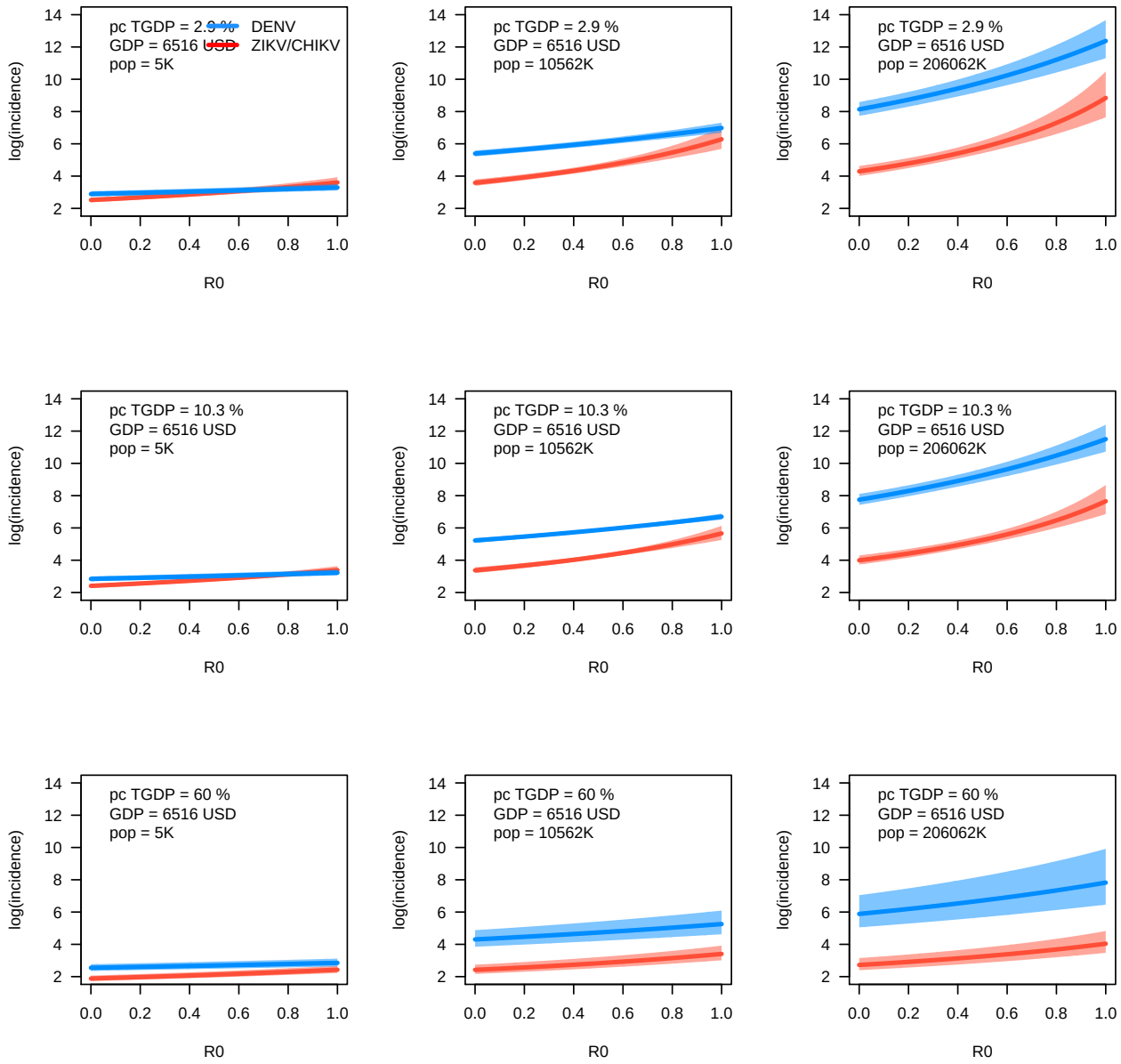
```
## [1] 0.00000 0.00000 0.00000 0.05151 0.94849 0.00000
```

```
## Proportion of Deviance  explained, compared to null model
D2s<-c(D.sqr(m), D.sqr(m0), D.sqr(m0B), D.sqr(m2), D.sqr(m2.sub), D.sqr(m0B.alt))
round(D2s, 3)
```

```
## [1] 0.465 0.506 0.550 0.595 0.595 0.531
```

```
##png(file="INC_m2sub_conditional_resp.png", height=1000, width=1000, pointsize=18)

my.GDP<-c(min(dd$lGDP), median(dd$lGDP), max(dd$lGDP))
my.TGDP<-c(min(dd$propTGDP), median(dd$propTGDP), max(dd$propTGDP))
#my.lpop<-c(min(d$lpop), median(d$lpop), max(d$lpop))
my.lpop<-c(min(dd$lpop), median(dd$lpop), max(dd$lpop))


my.i<-c(2, 16, 26)##w[c(seq(1,14,by=2), 14, 17)]## floor(seq(1, 31, length=9))
dat.cols<-rep(2, length(dd$l.inc))+2*as.numeric(dd$DEN)
par(mfrow=c(3,3))
for(j in 1:3){
  for(i in 1:3){
      visreg(m2.sub, "R0", scale="response", by="DEN",
             overlay=TRUE,
             cond=list(lGDP=my.GDP[2], propTGDP=my.TGDP[j],
                       lpop=my.lpop[i]),
             xlab="R0",
             ylab="log(incidence)", ylim=c(2, 14), rug=0, legend=FALSE)
        ##points(dd$lR0.alt[dd$lpTGDP==TGDPS[i]], dd$l.inc[dd$lpTGDP==TGDPS[i]], col=dat.cols[dd$lpTGDP=
      leg.txt<-c(paste("pc TGDP = ", round(my.TGDP[j]*100, 1), " %",
                       sep=""),
                 paste("GDP = ", round(exp(my.GDP[2])), " USD",
```

```
                    sep=""),
              paste("pop = ", round(exp(my.lpop[i])/1000), "K",
                    sep=""))
      legend("topleft", leg.txt, bty="n")
      if(i==1 && j==1){
          legend("topright",
                 c("DENV", "ZIKV/CHIKV"),
                 col=c("dodgerblue", "red"), lwd=c(3,3),
                 bty="n", y.intersp=1)

      }
  }
}
```



```
##dev.off()
```

Out of sample prediction for the Incidence data

```
## Now the bootstrap bit

nsamps<-1000

MSE.boot<-BIC.modprobs<-data.frame(matrix(NA, ncol=6, nrow=nsamps))
names(MSE.boot)<-names(BIC.modprobs)<-c("m", "m0", "m0B", "m2", "m2.sub", "m0B.alt")

MAPE.boot<-MSE.boot

## I'll stratefy by DEN
Vs<-as.factor(dd$DEN)
nCZs<-signif(0.1*summary(Vs)[1], digits=1)
nDs<-signif(0.1*summary(Vs)[2], digits=1)

dCZ<-subset(dd, DEN==FALSE)
dD<-subset(dd, DEN==TRUE)

ns<-as.numeric(nCZs+nDs)
ndat<-dim(dd)[1]-ns

set.seed(123)
for(i in 1:nsamps){

    if(i%%50==0) print(paste("Bootstrap sample ", i, " of ", nsamps, sep=""))

    test<-train<-iC<-iD<-iZ<-NULL
    iCZ<-sample(dim(dCZ)[1], nCZs)
    iD<-sample(dim(dD)[1], nDs)

    train<-rbind(dCZ[-iCZ,], dD[-iD,])
    test<-rbind(dCZ[iCZ,], dD[iD,])

    ## only pop
    reg<-glm(l.inc ~ lpop*DEN,
             family="Gamma", data=train)
    pred<-predict(reg, newdata=test, type="response")
    error<- test$l.inc - pred


    ## only socio factors
    reg0<-glm(l.inc ~ (propTGDP + lGDP +lpop)*DEN-DEN,
             family="Gamma", data=train)
    pred0<-predict(reg0, newdata=test, type="response")
    error0 <- test$l.inc - pred0


    ## using R0
    reg0B<-glm(l.inc ~ (R0 +lpop)*DEN-DEN, family="Gamma", data=train)
```

```
    pred0B<-predict(reg0B, newdata=test, type="response")
    error0B <- test$l.inc - pred0B

    ## using lR0.alt
    reg0B.alt<-glm(l.inc ~ lR0.alt*DEN-DEN, family="Gamma", data=train)
    pred0B.alt<-predict(reg0B.alt, newdata=test, type="response")
    error0B.alt <- test$l.inc - pred0B.alt


    ## using R0 + socio-economic bits
    reg2<-glm(l.inc ~ (R0 + propTGDP + lGDP + lpop)*DEN-DEN,
              family="Gamma", data=train)
    pred2<-predict(reg2, newdata=test, type="response")
    error2 <- test$l.inc - pred2

    ## using R0 + socio-economic bits
    reg2.sub<-glm(l.inc ~ (R0 + propTGDP + lpop)*DEN + lGDP - DEN,
              family="Gamma", data=train)
    pred2.sub<-predict(reg2.sub, newdata=test, type="response")
    error2.sub <- test$l.inc - pred2.sub

    MSE.boot[i,]<-c(mean(error^2),
                    mean(error0^2),
                    mean(error0B^2),
                    mean(error2^2),
                    mean(error2.sub^2),
                    mean(error0B.alt^2))

    MAPE.boot[i,]<-100*c(mean(abs(error/test$l.inc)),
                    mean(abs(error/test$l.inc)),
                    mean(abs(error0B/test$l.inc)),
                    mean(abs(error2/test$l.inc)),
                    mean(abs(error2.sub/test$l.inc)),
                    mean(abs(error0B.alt/test$l.inc)))

    ## model probs based on BIC
    bics <- c(BIC(reg), BIC(reg0),
              BIC(reg0B), BIC(reg2),
              BIC(reg2.sub), BIC(reg0B.alt))
    ebics<-exp(-0.5*(bics-min(bics)))
    BIC.modprobs[i,] <- ebics/sum(ebics)

}
```

```
round(colMeans(BIC.modprobs), digits=3)
```

```
##       m      m0     m0B      m2  m2.sub m0B.alt
##   0.000   0.000   0.000   0.058   0.942   0.000
```

The model that includes both the measure of R_0 and the socio-economic bits comes out on top via BIC.

Now look at how well these do based on Mean Absolute Percentage Error (this measure has some problems, but it's a bit easier to interpret).

```
abs.errors<-colMeans(MAPE.boot)
sd.errors<-apply(MAPE.boot, MARGIN=2, sd)

# o<-order(abs.errors)
# abs.errors<-abs.errors[o]
# sd.errors<-sd.errors[o]

# accuracy (= 1 - MAPE/100)
round(1 - abs.errors/100, 3)
```

```
##       m      m0     m0B      m2  m2.sub m0B.alt
##   0.831   0.831   0.852   0.860   0.860   0.847
```

Overall, we are off in our predictions of the incidence by about 14 - 17%.

All of the models perform similarly out of sample. To confirm, I test the (pairwise) hypothesis that these statistics are the same (skipping m2, and only using m2.sub)

```
c(abs((abs.errors[1]-abs.errors[3])/sqrt(sd.errors[1]^2+sd.errors[3]^2)),
  abs((abs.errors[1]-abs.errors[4])/sqrt(sd.errors[1]^2+sd.errors[4]^2)),
  abs((abs.errors[1]-abs.errors[5])/sqrt(sd.errors[1]^2+sd.errors[5]^2)),
  abs((abs.errors[1]-abs.errors[6])/sqrt(sd.errors[1]^2+sd.errors[6]^2))
  )
```

```
##         m         m         m         m
## 1.0470339 1.4253416 1.4278956 0.7917135
```

None of these test statistics are > 2 (although some are getting close), so we cannot reject the hypothesis that these statitics are the same between the models.

Just to confirm that we get similar results using MSE as a metric:

```
abs.errors.MSE<-colMeans(MSE.boot)
sd.errors.MSE<-apply(MSE.boot, MARGIN=2, sd)

o<-order(abs.errors.MSE)
abs.errors.MSE<-abs.errors.MSE[o]
sd.errors.MSE<-sd.errors.MSE[o]
abs.errors.MSE
```

```
##    m2.sub        m2       m0B   m0B.alt        m0         m
## 0.9435378 0.9435438 1.0626615 1.1128131 1.2360847 1.3500761
```

```
c(abs((abs.errors[1]-abs.errors[3])/sqrt(sd.errors[1]^2+sd.errors[3]^2)),
  abs((abs.errors[1]-abs.errors[4])/sqrt(sd.errors[1]^2+sd.errors[4]^2)),
  abs((abs.errors[1]-abs.errors[5])/sqrt(sd.errors[1]^2+sd.errors[5]^2))#,
  #abs((abs.errors[3]-abs.errors[4])/sqrt(sd.errors[3]^2+sd.errors[4]^2))
  )
```

```
##        m        m        m
## 1.047034 1.425342 1.427896
```

This agrees with the MAPE results. So m2.sub is definitely best by BIC, but performs about the same as the others out of sample.

Create a table of model outputs.

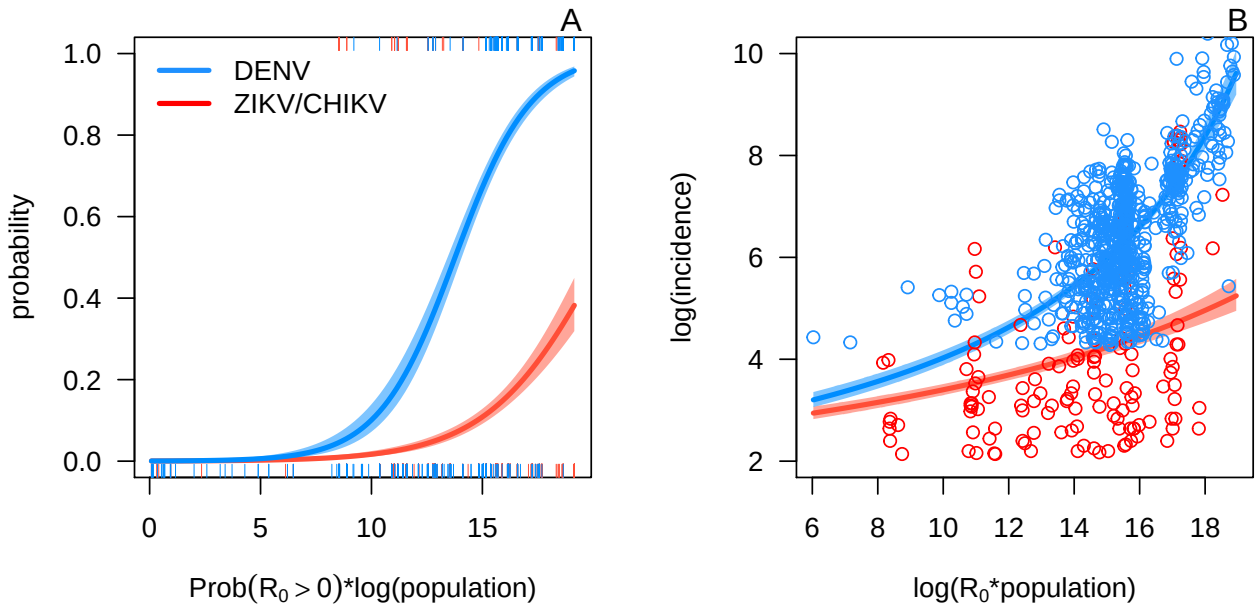
```
outs = data.frame('model' = c(1:6), 'bics' = round(bics, 0), 'probs' = round(probs, 3), 'D2s' = round(D
outs
```

```
##         model bics probs   D2s oos.probs abs.errors
## m           1 2593 0.000 0.465     0.000      0.169
## m0          2 2559 0.000 0.506     0.000      0.169
## m0B         3 2484 0.000 0.550     0.000      0.148
## m2          4 2433 0.052 0.595     0.058      0.140
## m2.sub      5 2427 0.948 0.595     0.942      0.140
## m0B.alt     6 2501 0.000 0.531     0.000      0.153
```

```
write.csv(outs, file = "inc_outs.csv")
```

Figures for main manuscript

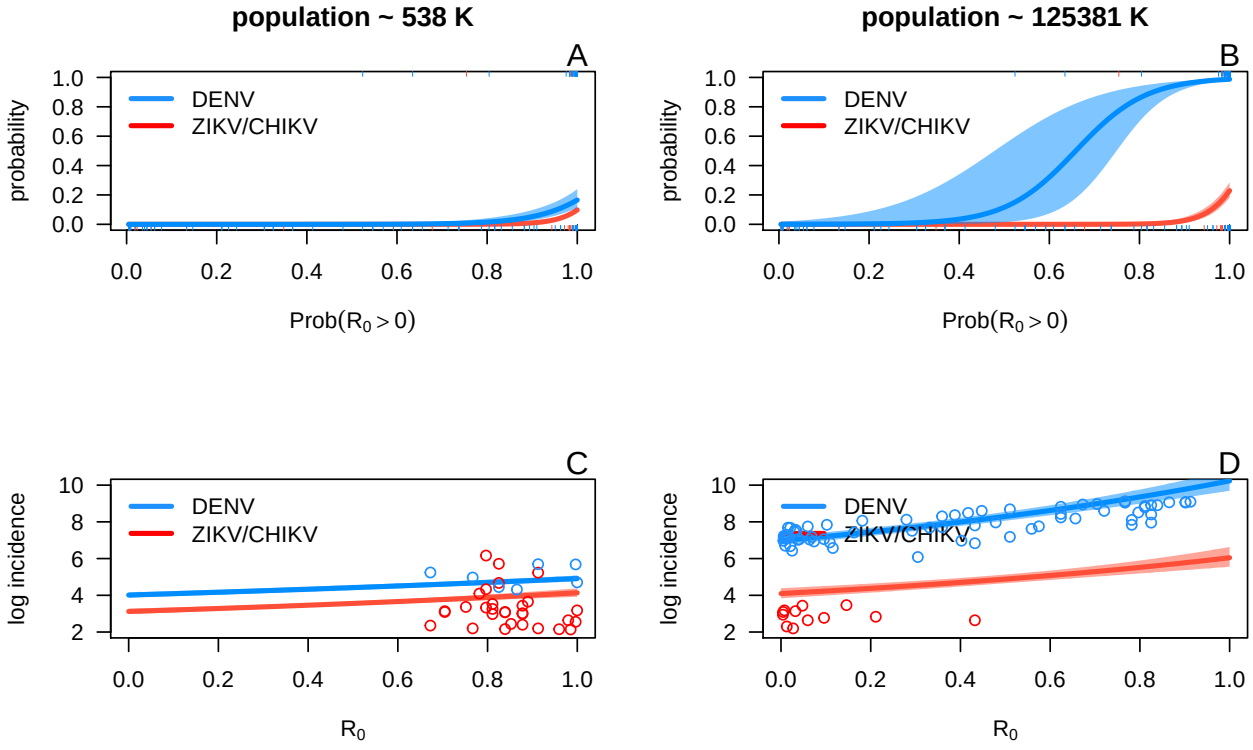
```
# png(file = "all_reg0A.alt_m0B.alt_resp.png", height = 500, width = 1000, pointsize = 16)

reg0A.alt<-glm(pa ~ R0.GR0.alt*DEN - DEN,
          family=binomial, data=d)

m0B.alt<-glm(l.inc ~ (lR0.alt)*DEN-DEN, data=dd, family="Gamma")

par(mfrow=c(1,2))
visreg(reg0A.alt, "R0.GR0.alt", scale="response", rug=2, by="DEN", overlay=TRUE,
         ylim=c(0,1), #cond=list(lpop=my.lpop2[i]),
         xlab=expression(paste(Prob(R[0]>0), "*log(population)", sep = "")), ylab="probability", legend=
         #main=paste("population ~ ", round(exp(my.lpop2[i])/1000), " K",  sep="")
       )
legend("topleft",
         c("DENV", "ZIKV/CHIKV"),
         col=c("dodgerblue", "red"), lwd=c(3,3),
         bty="n", y.intersp=1)
mtext("A", 3, at = 19, cex = 1.2)

visreg(m0B.alt, "lR0.alt", scale="response", by="DEN", overlay=TRUE,
         #cond=list(lpop=my.lpop2[i]),
         ylim=c(2,10), rug=0,
         xlab=expression(paste("log(", R[0], "*population)", sep = "")),
         ylab="log(incidence)", legend=FALSE,
         main="")
mycols<-(2+as.numeric(dd$DEN)*2)
mycols[mycols==2]<-"red"
mycols[mycols==4]<-"dodgerblue"

points(dd$lR0.alt, dd$l.inc, col=mycols)

mtext("B", 3, at = 19, cex = 1.2)
```




```
# dev.off()
```

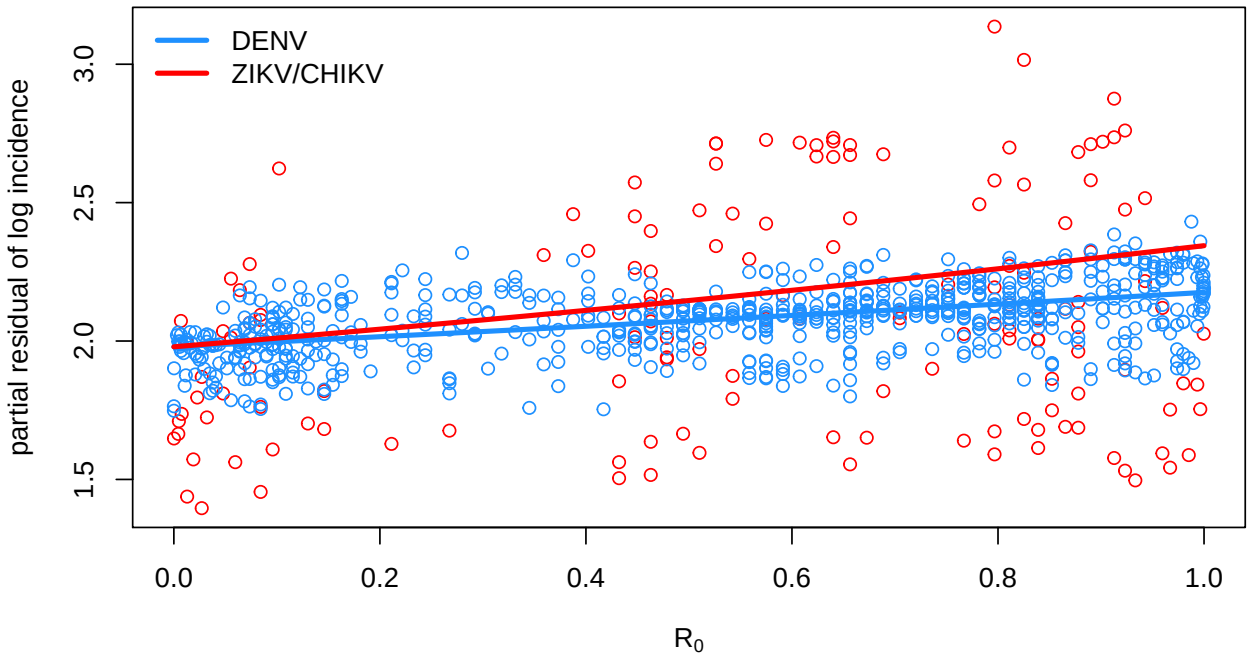
```
##png(file="all_reg0A_reg0B_resp.png", height=1000, width=1000, pointsize=16)


reg0A<-glm(pa ~ (R0.GR0+lpop)*DEN - DEN, family=binomial, data=d)

pops<-unique(d$lpop)
pops<-pops[order(pops)]

my.GDP<-c(min(d$lGDP), median(d$lGDP), max(d$lGDP))
my.TGDP<-c(min(d$propTGDP), median(d$propTGDP), max(d$propTGDP))
#my.lpop<-c(min(d$lpop), median(d$lpop), max(d$lpop))
my.lpop<-c(pops[3], median(d$lpop), pops[35])

my.lpop2<-pops[c(15,35)]

par(mfrow=c(2,2))
for(i in 1:2){
  visreg(reg0A, "R0.GR0", scale="response", rug=2, by="DEN", overlay=TRUE,
         ylim=c(0,1), cond=list(lpop=my.lpop2[i]),
         xlab=expression(Prob(R[0]>0)), ylab="probability", legend=FALSE,
         main=paste("population ~ ", round(exp(my.lpop2[i])/1000), " K",  sep=""))
  legend("topleft",
         c("DENV", "ZIKV/CHIKV"),
         col=c("dodgerblue", "red"), lwd=c(3,3),
         bty="n", y.intersp=1)
  if(i==1) mtext("A", 3, at = 1, cex = 1.2)
  if(i==2) mtext("B", 3, at = 1, cex = 1.2)

}

mycols<-(2+as.numeric(dd$DEN)*2)
mycols[mycols==2]<-"red"
```

```
mycols[mycols==4]<-"dodgerblue"

for(i in 1:2){
  visreg(m0B, "R0", scale="response", by="DEN", overlay=TRUE,
         cond=list(lpop=my.lpop2[i]), ylim=c(2,10), rug=0,
         xlab=expression(R[0]),
         ylab="log incidence", legend=FALSE,
         main="")


  legend("topleft",
         c("DENV", "ZIKV/CHIKV"),
         col=c("dodgerblue", "red"), lwd=c(3,3),
         bty="n", y.intersp=1)
  if(i==1){
    w<-intersect(which(dd$lpop>11), which(dd$lpop<14))
    mtext("C", 3, at = 1, cex = 1.2)
    points(dd$R0[w], dd$l.inc[w], col=mycols[w])

  }
  if(i==2){
    w<-intersect(which(dd$lpop>17.8), which(dd$lpop<19))
    mtext("D", 3, at = 1, cex = 1.2)
   points(dd$R0[w], dd$l.inc[w], col=mycols[w])

    }

}
```



```
##dev.off()
```

```
##png(file="inc_partial_resid_R0.png", height=600, width=600, pointsize=16)

m0B.2<-glm(l.inc ~ (R0 + lpop01) * DEN - DEN, family = "Gamma",
    data = dd)
p.resid<-function(dat, mod){

  b0<-mod$coeff[1]
  b2<-mod$coeff[3]
  b4<-mod$coeff[5]

  ## because I fit a gamma glm with an inverse link function, I transform the data onto the scale on wh
  pr<-(1/dat$l.inc)-(b2*dat$lpop+b4*dat$lpop*as.numeric(dat$DEN))
  return(pr)
}

dd$pr<-p.resid(dd, m0B)

mycols<-(2+as.numeric(dd$DEN)*2)
mycols[mycols==2]<-"red"
mycols[mycols==4]<-"dodgerblue"

par(mfrow=c(1,1))
## transform the partial residual back onto the original scale
plot(dd$R0, 1/dd$pr, col=mycols, ylab="partial residual of log incidence", xlab=expression(R[0]))

## add in the transformed lines for R0
rr<-seq(0, 1, length=100)
lines(rr, 1/(m0B$coeff[1]+rr*m0B$coeff[2]+rr*m0B$coeff[4]), col="dodgerblue", lwd=3)
lines(rr,  1/(m0B$coeff[1]+rr*m0B$coeff[2]), col="red", lwd=3)

  legend("topleft",
         c("DENV", "ZIKV/CHIKV"),
         col=c("dodgerblue", "red"), lwd=c(3,3),
         bty="n", y.intersp=1)
```


DENV
ZIKV/CHIKV
partial residual of log incidence
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
R0

```
##dev.off()
```