```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.72 (7847) x86_64-apple-darwin17.0]

[Workspace restored from /Users/andreasmurr/.RData]
[History restored from /Users/andreasmurr/.Rapp.history]

> setwd('/Users/andreasmurr/Dropbox/PostModernBayes/Replication')
> # Computing quantities of interest and their uncertainty using Bayesian simulation
> # Andreas Murr, Richard Traunmueller, and Jeff Gill
> # Creates Figures 3, 5, and 6 as well as Table 3 based on voter turnout data
>
> # clear working memory
>
>  rm(list=ls())
>
> # load packages
>
>  library(R2jags)
Loading required package: rjags
Loading required package: coda
Linked to JAGS 4.2.0
Loaded modules: basemod,bugs

Attaching package: ‘R2jags’

The following object is masked from ‘package:coda’:

    traceplot

>  library(foreign)
>
> # write functions
>
>  logit = function(x){log(x/(1-x))}
>  invlogit = function(x){exp(x)/(1+exp(x))}
>
> # load data
>
>  turnout = read.dta("./King2000/turnout.dta", convert.factors=FALSE)
>
> # note that in the data set agesqrd = age^2 / 100
>
>  with(turnout, plot(age^2 / 100, agesqrd))
```

```
>  abline(a=0, b=1)
>
> # create college dummy
>
>  turnout$college = ifelse(turnout$educate>=13, 1, 0) # with 12 years or less they have
finished highschool, with 13 years or more they at least started college
>
> # list data
>
>  y = turnout$turnout
>  X1 = cbind(1, turnout$educate, turnout$age, turnout$agesqrd, turnout$income, turnout$white)
>  X2 = cbind(1, turnout$educate, turnout$educate^2 / 10, turnout$college,
turnout$educate*turnout$college, (turnout$educate^2 / 10)*turnout$college, turnout$age,
turnout$agesqrd, turnout$income, turnout$white)
>  K1 = ncol(X1)
>  K2 = ncol(X2)
>  N = length(y)
>
>  jags.dat1 = list(
+    y = y,
+    X = X1,
+    K = K1,
+    N = N
+ )
>
>  jags.dat2 = list(
+    y = y,
+    X = X2,
+    K = K2,
+    N = N
+ )
>
> # write model
>
>  logit.mod = function(){
+    for(i in 1:N){
+       y[i] ~ dbinom(p[i], 1)
+       logit(p[i]) = X[i,]%*%beta
+    }
+    for (k in 1:K){
+       beta[k] ~ dnorm(0, 0.01)
+    }
+ }
>
> # set parameters to monitor
>
>  jags.par = c("beta")
>
> # fit models
>
>  n.iter = 21000
>  n.burn = 20000
>
>  set.seed(123)
>  jags.mod1 = jags(model.file = logit.mod,
+    data = jags.dat1,
+   parameters.to.save = jags.par,
+    n.chains = 5,
+    n.iter = n.iter,
+   n.burnin = n.burn)
module glm loaded
```

```
  |**************************************************| 100%
>
>  set.seed(123)
>  jags.mod2 = jags(model.file = logit.mod,
+     data = jags.dat2,
+    parameters.to.save = jags.par,
+     n.chains = 5,
+     n.iter = n.iter,
+    n.burnin = n.burn)

  |**************************************************| 100%
>
> # get coefficients (s x k matrix)
>
>   B1 = jags.mod1$BUGSoutput$sims.matrix[,colnames(jags.mod1$BUGSoutput$sims.matrix)!
="deviance"]
>   B2 = jags.mod2$BUGSoutput$sims.matrix[,colnames(jags.mod2$BUGSoutput$sims.matrix)!
="deviance"]
>
> # ============
> # = figure 3 =
> # ============
>
> # computation
>
>  # probability of turnout (replication of King et al. (2002: Figure 1 on p. 355), based on
replication stata do-file, King2000/turnout.do)
>  ## set x values for high school (.his) and college (.col)
>  X.his = cbind(1, 12, 18:95, (18:95)^2/100, mean(turnout$income), mean(turnout$white))
>  X.col = cbind(1, 16, 18:95, (18:95)^2/100, mean(turnout$income), mean(turnout$white))
>  # compute probability of turnout (P)
>  P.his = invlogit(apply(B1, 1, function(x){X.his%*%x}))
>  P.col = invlogit(apply(B1, 1, function(x){X.col%*%x}))
>  # compute median value (.me) and 99% credible intervals (.lo, .hi)
>  p.his.lo = apply(P.his, 1, function(x){quantile(x, .005)})
>  p.his.me = apply(P.his, 1, function(x){quantile(x, .5)})
>  p.his.hi = apply(P.his, 1, function(x){quantile(x, .995)})
>  p.col.lo = apply(P.col, 1, function(x){quantile(x, .005)})
>  p.col.me = apply(P.col, 1, function(x){quantile(x, .5)})
>  p.col.hi = apply(P.col, 1, function(x){quantile(x, .995)})
>  # marginal effect
>  ### king et al. use age^2 / 100, which means that b2 * age^2 / 100 is b2/100 * age^2 and so
its derivative with respect to age becomes 2*b2/100 * age
>  ## compute linear predictor
>  Xb = apply(B1, 1, function(x){X1%*%x})
>  # compute marginal effect
>  marginal = invlogit(Xb) * ((B1[,3] + 2 * cbind(turnout$age) %*% (B1[,4] / 100)) / (1 +
exp(Xb)))
>  # select marginal effect (m) for high school (.his) and college (.col)
>  m.his = marginal[turnout$educate==12,]
>  m.col = marginal[turnout$educate==16,]
>  # compute median effect (.me) and 99% credible interval (.lo, .hi)
>  m.his.lo = apply(m.his, 1, function(x){quantile(x, 0.005)})
>  m.his.me = apply(m.his, 1, function(x){quantile(x, 0.5)})
>  m.his.hi = apply(m.his, 1, function(x){quantile(x, 0.995)})
>  m.col.lo = apply(m.col, 1, function(x){quantile(x, 0.005)})
>  m.col.me = apply(m.col, 1, function(x){quantile(x, 0.5)})
>  m.col.hi = apply(m.col, 1, function(x){quantile(x, 0.995)})
>
> # numerical example in estimate level / marginal effects / third paragraph
>
```

```
>  # age with smallest credible interval of predicted probability (high school)
>  (18:95)[which.min(p.his.hi - p.his.lo)]
[1] 34
>  # age with largest variation of marginal effects (high school)
>  names(which.max(tapply(m.his.me, turnout$age[turnout$educate==12], var)))
[1] "34"
>
> # numerical example in estimate level / marginal effects / fourth paragraph
>
>  # age at which credible intervals of marginal effects approach 0 from above
>  ## high school
>  turnout$age[turnout$educate==12][m.his.lo>0][which.min(m.his.lo[m.his.lo>0])]
[1] 52
>  ## college
>  turnout$age[turnout$educate==16][m.col.lo>0][which.min(m.col.lo[m.col.lo>0])]
[1] 53
>  # age at which medians of marginal effects approach 0 from above
>  ## high school
>  turnout$age[turnout$educate==12][m.his.me>0][which.min(m.his.me[m.his.me>0])]
[1] 70
>  ## college
>  turnout$age[turnout$educate==16][m.col.me>0][which.min(m.col.me[m.col.me>0])]
[1] 70
>
> # numerical example in estimate level / marginal effects / fifth paragraph
>
>  # observed range of age with 16 years of education
>  range(turnout$age[turnout$educate==16])
[1] 20 88
>
> # plotting
>
>  # jitter x-values for graphic
>  set.seed(123)
>  x.his = jitter(turnout$age[turnout$educate==12], factor=1.3)
>  x.col = jitter(turnout$age[turnout$educate==16], factor=1.3)
>  # range of x and y values for marginal effect
>  x.ran = range(c(18:95, x.his, x.col))
>  y.ran = range(c(m.his.lo, m.his.hi, m.col.lo, m.col.hi))
>  # open device
>  cairo_pdf("figure-3.pdf", width=6, height=3.5, pointsize=10)
>  # graphical parameters
>  par(mar=c(0,0,0,0), oma=c(3,6,2,0), mgp=c(1,.5,0), tck=-.01, las=1, mfrow=c(2,2),
family="Gill Sans")
>  ### probability of turnout
>  # high school
>  plot(18:95, p.his.me, type="p", ylim=c(.2,1), xlim=x.ran, xlab="", ylab="", axes=F, pch=16,
cex=.5)
>  segments(x0=18:95, x1=18:95, y0=p.his.lo, y1=p.his.hi)
>  axis(2, at=seq(.4, 1, .2),  tick=F)
>  mtext(side=2, "Probability of turnout\nat mean values", las=3, line=3)
>  mtext(side=3, "High school degree (12 years of education)")
>  # college
>  plot(18:95, p.col.me, type="p", ylim=c(.2,1), xlim=x.ran, xlab="", ylab="", axes=F, pch=16,
cex=.5)
>  segments(x0=18:95, x1=18:95, y0=p.col.lo, y1=p.col.hi)
>  mtext(side=3, "College degree (16 years of education)")
>  ### marginal effect
>  # high school
>  plot(x.his, m.his.me, type="p", ylim=y.ran, xlim=x.ran, xlab="", ylab="", axes=F, pch=16,
cex=.5)
```

```
> segments(x0=x.his, x1=x.his, y0=m.his.lo, y1=m.his.hi, lwd=.5)
> lines(x.ran, rep(0, 2), lty=2)
> axis(1, at=seq(20,100, 10), tick=F)
> axis(2, tick=F)
> mtext(side=1, "Age of respondent", line=2)
> mtext(side=2, "Marginal effect of age\nat observed values", las=3, line=3)
> # college
> plot(x.col, m.col.me, type="p", ylim=y.ran, xlim=x.ran, xlab="", ylab="", axes=F, pch=16,
cex=.5)
> segments(x0=x.col, x1=x.col, y0=m.col.lo, y1=m.col.hi, lwd=.5)
> lines(x.ran, rep(0, 2), lty=2)
> axis(1, at=seq(20,100, 10), tick=F)
> mtext(side=1, "Age of respondent", line=2)
> # close device
> dev.off()
quartz
     2
>
> # =============
> # = figure 5 =
> # =============
>
> # posterior predictive checks
>
> # compute predicted probabilities (n x k)(k x s) = (n x s)
>
> P1 = invlogit(X1%*%t(B1))
> P2 = invlogit(X2%*%t(B2))
>
> # compute replications
>
> set.seed(123)
> R1 = apply(P1, 2, function(x){rbinom(n=length(x), 1, p=x)})
> R2 = apply(P2, 2, function(x){rbinom(n=length(x), 1, p=x)})
>
> # select 20 replicates
>
> set.seed(123)
> p.sel = sample(1:ncol(P1), 20)
> P1.sel = P1[,p.sel]
> R1.sel = R1[,p.sel]
> R2.sel = R2[,p.sel]
>
> # plot posterior predictive checks
>
> plot.check = function(var.nam, x.var, y.var, replications, ylim1=rep(NA, 2)){
+   x = sort(unique(x.var))
+   y = tapply(y.var, x.var, mean)
+   z = apply(replications, 2, function(r){tapply(r, x.var, mean)})

+   par(oma=c(0,0,0,0))
+   par(mfrow=c(1,2), mar=c(3,4,1,1), mgp=c(0,.25,0), tck=-.01, las=1, cex=1.5, family="Gill
Sans")
+   plot(x, y, type="n", xlab="", ylab="", ylim=range(cbind(z, y)), axes=F)
+   #grid()
+   axis(1, lty=0)
+   axis(2, lty=0)
+   mtext(text=var.nam, side=1, line=1.5, cex=1.5)
+   mtext(text="Pr(Turnout)", side=2, line=2, las=3, cex=1.5)
+   apply(z, 2, function(p){lines(x, p, col="grey")})
+   lines(x, y, type="l", lwd=2)
```

```
+     d.hat = apply(z, 2, function(p){y - p})
+     plot(x, d.hat[,1], type="n", xlab="", ylab="", ylim=range(d.hat), axes=FALSE)
+     #grid()
+     axis(1, lty=0)
+     axis(2, lty=0)
+     mtext(text=var.nam, side=1, line=1.5, cex=1.5)
+     mtext(text="Actual - Predicted", side=2, line=2, las=3, cex=1.5)
+     apply(d.hat, 2, function(p){lines(x, p, col="grey")})
+     abline(h=0, lty=2)
+ }
>  cairo_pdf("figure-5-age.pdf", width=6, height=3, pointsize=8)
>  plot.check("Age", turnout$age, turnout$turnout, R1.sel)
>  dev.off()
quartz
     2
>  cairo_pdf("figure-5-education.pdf", width=6, height=3, pointsize=8)
>  plot.check("Education", turnout$educate, turnout$turnout, R1.sel)
>  dev.off()
quartz
     2
>  cairo_pdf("figure-5-income.pdf", width=6, height=3, pointsize=8)
>  plot.check("Income", turnout$income, turnout$turnout, R1.sel)
>  dev.off()
quartz
     2
>  cairo_pdf("figure-5-white.pdf", width=6, height=3, pointsize=8)
>  plot.check("White", turnout$white, turnout$turnout, R1.sel)
>  dev.off()
quartz
     2
>
> # ============
> # = figure 6 =
> # ============
>
> # posterior predictive check: education in model 1 and 2
>
>  cairo_pdf("figure-6-education-m1.pdf", width=6, height=3, pointsize=8)
>  plot.check("Education", turnout$educate, turnout$turnout, R1.sel)
>  dev.off()
quartz
     2
>  cairo_pdf("figure-6-education-m2.pdf", width=6, height=3, pointsize=8)
>  plot.check("Education", turnout$educate, turnout$turnout, R2.sel)
>  dev.off()
quartz
     2
>
> # ===========
> # = table 3 =
> # ===========
>
> # aic (first row)
>
>  m1 = glm(turnout ~ educate + age + agesqrd + income + white, family=binomial(link="logit"),
data=turnout)
>  m2 = glm(turnout ~ educate + I(educate^2 / 10) + college + educate:college + I(educate^2 /
10):college + age + I(age^2 / 100) + income + white, family=binomial(link="logit"),
data=turnout)
>
>  aic.tab = round(c(AIC(m1), AIC(m2), AIC(m1) - AIC(m2)))
```

```
>
> # waic (second row)
>
>  # compute densities (n x s): p ( y_i | theta^s )
>  p1 = apply(P1, 2, function(x){
+    jags.dat1$y*x + (1 - jags.dat1$y)*(1 - x)
+ })
>  p2 = apply(P2, 2, function(x){
+    jags.dat2$y*x + (1 - jags.dat2$y)*(1 - x)
+ })
>  # compute log pointwise predictive density: sum log ( 1/S sum p ( y_i | theta^s ) )
>  lpd.hat1 = sum(log(rowMeans(p1)))
>  lpd.hat2 = sum(log(rowMeans(p2)))
>  # check reliability of any of the terms in estimated effective number of parameters
>  if (sum(apply(log(p1), 1, var) > .4)==0 & sum(apply(log(p2), 1, var) > .4)==0){
+    # computed effective number of parameters sum var(log p ( y_i | theta^s ))

+    p.waic.hat1 = sum(apply(log(p1), 1, var))
+    p.waic.hat2 = sum(apply(log(p2), 1, var))
+    # compute expected log pointwise predictive density

+    elpd.waic.hat1 = lpd.hat1 - p.waic.hat1
+    elpd.waic.hat2 = lpd.hat2 - p.waic.hat2
+    # compute expected log pointwise predictive density on devience scale
+    waic1 = -2*elpd.waic.hat1
+    waic2 = -2*elpd.waic.hat2
+    # compute standard error
+    se.waic1 = sqrt(N)*sd(-2*(log(rowMeans(p1)) - apply(log(p1), 1, var)))
+    se.waic2 = sqrt(N)*sd(-2*(log(rowMeans(p2)) - apply(log(p2), 1, var)))
+    # compute standard error of difference

+    se.waic12 = sqrt(N)*sd(-2*(log(rowMeans(p2)) - apply(log(p2), 1, var)) -
-2*(log(rowMeans(p1)) - apply(log(p1), 1, var)))
+    # return results ()
+    waic.tab = round(c(waic1, se.waic1, waic2, se.waic2, waic1 - waic2, se.waic12))

+ }
>
> # complete table
>  aic.tab
[1] 15836 15758    79
>  waic.tab
[1] 15842   132 15763   133    79    19
>
> # ====================
> # = end source code =
> # ====================
>
```