CSSS 510: Lab 3

Logistic Regression

2017-10-13

# 0. Agenda

1. Deriving a likelihood function for the logistic regression model
2. Fitting a logit model using and
3. Simulating predicted values and confidence intervals
4. Simulating first differences

# 1. Deriving a likelihood function for the logistic regression model

Recall from lecture the logit model:

In the simple case, this stems from the latent variable model:

where the relationship between latent variable and the explanatory variable is modeled using simple linear regression, and the binary outcome is a function of the sign of :

The logistic regression model is obtained if we assume the errors of this latent variable model follow a standard logistic distribution. Recall that the pdf and cdf of the standard logisitic distribution are as follows:

We therefore have the following:

Since we assume the errors follow a standard logistic distribution, we have

and E=0 and Var.

The logit function is the inverse of the logistic function:

or

We therefore have the following

or

or

$\newline$

Recall from lecture that a Bernoulli distribution has the following pdf:

And the likelihood function can be derived from the joint probability:

# 2. Fitting a logit model using and

rm(list = ls()) # clear up the memory  
  
#install and load the packages needed  
#from CRAN: install.packages("MASS", dependencies = TRUE)  
library(MASS)  
library(RColorBrewer)  
  
#download simcf and tile packages from- http://faculty.washington.edu/cadolph/software  
# don't unzip the archive (tar) file   
library(simcf)  
library(tile)

## Loading required package: grid

# Load data  
file <- "nes00a.csv"  
data <- read.csv(file, header=TRUE)  
  
# attach(data)   
  
# Estimate logit model using optim()  
# Construct variables and model objects  
y <- data$vote00  
x <- cbind(data$age,data$hsdeg,data$coldeg)  
  
# Likelihood function for logit  
llk.logit <- function(param,y,x) {  
 os <- rep(1,length(x[,1])) # constant   
 x <- cbind(os,x) # constant+covariates  
 b <- param[ 1 : ncol(x) ]   
 # number of parameters to be estimated equals number of columns in x   
 # (i.e, one for constant and one for each covariates : total 4)  
 xb <- x%\*%b   
 sum( y\*log(1+exp(-xb)) + (1-y)\*log(1+exp(xb))) # log-likelihood function for logit model   
 # (based on our choice of standard logistic cdf as the systematic component)  
 # optim is a minimizer, so use -lnL here  
}  
  
# Fit logit model using optim  
ls.result <- lm(y~x) # use ls estimates as starting values (for convenience)  
stval <- ls.result$coefficients # initial guesses  
logit.result.opt <- optim(stval,llk.logit,method="BFGS",hessian=T,y=y,x=x)  
 # call minimizer procedure or max by adding control=list(fnscale=-1)  
pe.opt <- logit.result.opt$par # point estimates  
vc.opt <- solve(logit.result.opt$hessian) # var-cov matrix  
se.opt <- sqrt(diag(vc.opt)) # standard errors  
ll.opt <- -logit.result.opt$value # likelihood at maximum  
  
logit.optim<-data.frame(cbind(round(pe.opt,3), round(se.opt,3)))  
rownames(logit.optim)<-c("intercept", "age", "highschool" , "college")  
colnames(logit.optim)<-c("pe", "std.err")  
logit.optim

## pe std.err  
## intercept -2.149 0.257  
## age 0.031 0.003  
## highschool 1.213 0.179  
## college 1.102 0.130

#p-value based on t-statistics  
2\*pt(abs(logit.optim$pe/logit.optim$std.err), df=length(y)-length(pe.opt) , lower.tail = FALSE)

## [1] 1.229175e-16 2.393414e-24 1.668155e-11 4.780627e-17

# Estimate logit model using glm()  
  
# Run logit & extract results using glm.  
# GLM solves the likelihood equations with a common numeric algorithm   
# called iteratively re-weighted least squares (IRWLS).  
  
logit.result <- glm(vote00~age +hsdeg+coldeg, family=binomial, data=data)   
# family "binomial" calls logit transformation (log(pi/1-pi)) as a "link function"   
# corresponding to logistic distribution.   
# Link function transforms pi so that it follows a linear model.   
# (So although pi itself is dependent on covariates in a non-linear way,   
# logit transformed pi is dependent on covariates in a linear way.)   
  
summary(logit.result)

##   
## Call:  
## glm(formula = vote00 ~ age + hsdeg + coldeg, family = binomial,   
## data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4487 -1.1347 0.6360 0.8973 1.8854   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.147997 0.256608 -8.371 < 2e-16 \*\*\*  
## age 0.030885 0.003386 9.121 < 2e-16 \*\*\*  
## hsdeg 1.212882 0.179447 6.759 1.39e-11 \*\*\*  
## coldeg 1.102465 0.130426 8.453 < 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: 2293.5 on 1782 degrees of freedom  
## Residual deviance: 2076.0 on 1779 degrees of freedom  
## AIC: 2084  
##   
## Number of Fisher Scoring iterations: 4

# now a new model adding age^2  
model <- vote00 ~ age + I(age^2) + hsdeg + coldeg  
mdata <- extractdata(model, data, na.rm=TRUE) # needs library(simcf)  
  
logit.result <- glm(model, family=binomial, data=mdata)   
summary(logit.result)

##   
## Call:  
## glm(formula = model, family = binomial, data = mdata)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2045 -1.1145 0.6335 0.8743 1.9841   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.0193891 0.4181899 -7.220 5.19e-13 \*\*\*  
## age 0.0747252 0.0168440 4.436 9.15e-06 \*\*\*  
## I(age^2) -0.0004427 0.0001655 -2.674 0.00749 \*\*   
## hsdeg 1.1243908 0.1800069 6.246 4.20e-10 \*\*\*  
## coldeg 1.0795702 0.1312113 8.228 < 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: 2293.5 on 1782 degrees of freedom  
## Residual deviance: 2069.0 on 1778 degrees of freedom  
## AIC: 2079  
##   
## Number of Fisher Scoring iterations: 4

pe <- logit.result$coefficients # point estimates  
vc <- vcov(logit.result) # var-cov matrix  
  
pe

## (Intercept) age I(age^2) hsdeg coldeg   
## -3.0193890720 0.0747251922 -0.0004427014 1.1243907749 1.0795702357

sqrt(diag(vc))

## (Intercept) age I(age^2) hsdeg coldeg   
## 0.4181899367 0.0168440337 0.0001655466 0.1800068893 0.1312113265

# 3. Simulating predicted values and confidence intervals

# Simulate parameter distributions  
sims <- 10000  
simbetas <- mvrnorm(sims, pe, vc) #needs library(MASS) # draw 10000 sets of simulated   
# parameter (beta) estimates from a multivariate normal distribution with   
# mean pe and variance-covariance vc  
  
# Now let's plan counterfactuals: We will have three sets of counterfactuals based   
# on education lavel (less than hs edu, hs edu, college or higher edu), and for each set   
# we will make age varies between 18 years old and 97 years old.   
  
# Set up counterfactuals: all ages, each of three educations  
xhyp <- seq(18,97,1) # create age vector   
nscen <- length(xhyp) # we will have total 80 different age scenarios for each education level  
  
nohsScen <- hsScen <- collScen <- cfMake(model, mdata, nscen) #this is just to initialize   
# 80 scenarios for each education level. As default, all covariate values are set at the mean.  
  
# Create three sets of education counterfactuals  
  
  
for (i in 1:nscen) {  
 # No High school scenarios (loop over each age, total 80 scenarios)  
 nohsScen <- cfChange(nohsScen, "age", x = xhyp[i], scen = i)  
 nohsScen <- cfChange(nohsScen, "hsdeg", x = 0, scen = i) # no hs degree  
 nohsScen <- cfChange(nohsScen, "coldeg", x = 0, scen = i) # no college degree  
  
 # HS grad scenarios (loop over each age, total 80 scenarios)  
 hsScen <- cfChange(hsScen, "age", x = xhyp[i], scen = i)  
 hsScen <- cfChange(hsScen, "hsdeg", x = 1, scen = i) # has hs degree  
 hsScen <- cfChange(hsScen, "coldeg", x = 0, scen = i) # no college degree  
  
 # College grad scenarios (loop over each age, total 80 scenarios)  
 collScen <- cfChange(collScen, "age", x = xhyp[i], scen = i)  
 collScen <- cfChange(collScen, "hsdeg", x = 1, scen = i) # has hs degree  
 collScen <- cfChange(collScen, "coldeg", x = 1, scen = i) # has college degree  
}  
  
# # Now given the counterfactual covariates (nohsScen/hsScen/collScen) and simulated   
# parameters (simbetas), we can calculate expected value of the response.   
# In this case, expected probability of voting!  
  
head(nohsScen$x) #we will fit the counterfactual data

## vote00 age hsdeg coldeg  
## 1 0.6567583 18 0 0  
## 2 0.6567583 19 0 0  
## 3 0.6567583 20 0 0  
## 4 0.6567583 21 0 0  
## 5 0.6567583 22 0 0  
## 6 0.6567583 23 0 0

nohsScen$model # in the model specification

## vote00 ~ age + I(age^2) + hsdeg + coldeg

nohsSims <- logitsimev(nohsScen, simbetas, ci=0.95) # using simulated betas to get expected values.   
# Built-in function "logitsimev" calculates the expected value   
# for every individual scenario you created.  
  
nohsSims #reports lower and upper confidence intervals as well as expected probabilities

## $pe  
## [1] 0.1423245 0.1494199 0.1567040 0.1641689 0.1718058 0.1796049 0.1875560  
## [8] 0.1956476 0.2038680 0.2122046 0.2206444 0.2291739 0.2377791 0.2464461  
## [15] 0.2551605 0.2639078 0.2726736 0.2814437 0.2902038 0.2989400 0.3076387  
## [22] 0.3162864 0.3248704 0.3333783 0.3417980 0.3501183 0.3583283 0.3664178  
## [29] 0.3743771 0.3821972 0.3898696 0.3973864 0.4047404 0.4119249 0.4189337  
## [36] 0.4257612 0.4324023 0.4388524 0.4451072 0.4511632 0.4570169 0.4626654  
## [43] 0.4681062 0.4733369 0.4783556 0.4831606 0.4877504 0.4921238 0.4962797  
## [50] 0.5002172 0.5039356 0.5074344 0.5107130 0.5137711 0.5166083 0.5192246  
## [57] 0.5216196 0.5237933 0.5257456 0.5274765 0.5289861 0.5302743 0.5313412  
## [64] 0.5321871 0.5328119 0.5332160 0.5333995 0.5333627 0.5331061 0.5326300  
## [71] 0.5319350 0.5310218 0.5298910 0.5285436 0.5269807 0.5252033 0.5232131  
## [78] 0.5210114 0.5186002 0.5159816  
##   
## $lower  
## [1] 0.09414506 0.10021998 0.10645964 0.11299442 0.11964225 0.12641217  
## [7] 0.13318537 0.14037238 0.14759789 0.15463468 0.16178487 0.16903749  
## [13] 0.17661431 0.18416165 0.19171287 0.19898681 0.20620553 0.21373229  
## [19] 0.22154273 0.22896588 0.23668305 0.24407347 0.25131200 0.25882911  
## [25] 0.26617232 0.27339496 0.28071315 0.28831646 0.29557257 0.30285314  
## [31] 0.30995949 0.31660137 0.32361783 0.33056986 0.33746379 0.34410282  
## [37] 0.35080651 0.35732269 0.36354360 0.36992516 0.37593593 0.38175969  
## [43] 0.38727974 0.39292402 0.39835389 0.40348000 0.40842797 0.41322212  
## [49] 0.41715425 0.42115274 0.42548609 0.42916366 0.43284656 0.43565919  
## [55] 0.43809931 0.43973268 0.44122840 0.44217206 0.44275234 0.44329355  
## [61] 0.44317706 0.44245166 0.44127716 0.43971638 0.43719524 0.43382790  
## [67] 0.43063755 0.42749658 0.42350855 0.41872785 0.41332281 0.40741735  
## [73] 0.40083489 0.39423607 0.38712280 0.37972611 0.37297617 0.36556420  
## [79] 0.35700231 0.34813122  
##   
## $upper  
## [1] 0.2031163 0.2104908 0.2183312 0.2266098 0.2348842 0.2435910 0.2519094  
## [8] 0.2608193 0.2695737 0.2787268 0.2888808 0.2985746 0.3084208 0.3181555  
## [15] 0.3279108 0.3374475 0.3474061 0.3574278 0.3667764 0.3767922 0.3866669  
## [22] 0.3964421 0.4063612 0.4154110 0.4244396 0.4335266 0.4417965 0.4504563  
## [29] 0.4585957 0.4667611 0.4747415 0.4825236 0.4896639 0.4970609 0.5040653  
## [36] 0.5104027 0.5170776 0.5231332 0.5288539 0.5343227 0.5403786 0.5449157  
## [43] 0.5502528 0.5550381 0.5592327 0.5632922 0.5671909 0.5705736 0.5744022  
## [50] 0.5783203 0.5816298 0.5850227 0.5881974 0.5914378 0.5944166 0.5973487  
## [57] 0.6001473 0.6031980 0.6067339 0.6096288 0.6130170 0.6163833 0.6194670  
## [64] 0.6221588 0.6247982 0.6288187 0.6321189 0.6355289 0.6387887 0.6426457  
## [71] 0.6463310 0.6500314 0.6530057 0.6571359 0.6609766 0.6640637 0.6683856  
## [78] 0.6721334 0.6755513 0.6793763

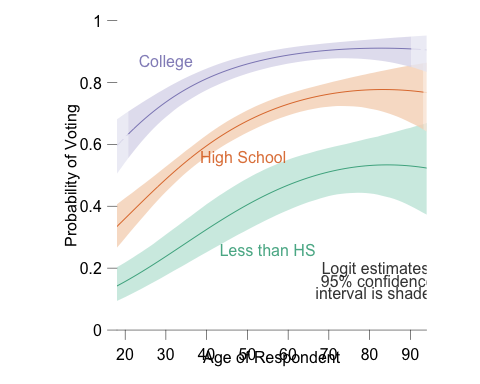
# same thing for two other sets of sceneraios  
hsSims <- logitsimev(hsScen, simbetas, ci=0.95)  
collSims <- logitsimev(collScen, simbetas, ci=0.95)  
  
# Get 3 nice colors for traces  
col <- brewer.pal(3,"Dark2")  
  
# Set up lineplot traces of expected probabilities  
  
#Traces are elements of tile : lines, labels, legends...  
  
# no hs   
nohsTrace <- lineplot(x=xhyp, # age on x-axis  
 y=nohsSims$pe, #expected probability on y-axis  
 lower=nohsSims$lower, # lower confidence interval  
 upper=nohsSims$upper, #upper confidence interval  
 col=col[1], #color choice   
 extrapolate=list(data=mdata[,2:ncol(mdata)],   
 #actual covariates (i.e., values in your data)  
 cfact=nohsScen$x[,2:ncol(hsScen$x)],#counterfactual covariates  
 omit.extrapolated=FALSE), #don't show extrapolated values   
 plot=1)  
  
# hs but no college  
hsTrace <- lineplot(x=xhyp,  
 y=hsSims$pe,  
 lower=hsSims$lower,  
 upper=hsSims$upper,  
 col=col[2],  
 extrapolate=list(data=mdata[,2:ncol(mdata)],  
 cfact=hsScen$x[,2:ncol(hsScen$x)],  
 omit.extrapolated=FALSE),  
 plot=1)  
  
#college   
collTrace <- lineplot(x=xhyp,  
 y=collSims$pe,  
 lower=collSims$lower,  
 upper=collSims$upper,  
 col=col[3],  
 extrapolate=list(data=mdata[,2:ncol(mdata)],  
 cfact=collScen$x[,2:ncol(hsScen$x)],  
 omit.extrapolated=FALSE),  
 plot=1)  
  
# Set up traces with labels  
  
labelTrace <- textTile(labels=c("Less than HS", "High School", "College"),  
 x=c( 55, 49, 30),  
 y=c( 0.26, 0.56, 0.87),  
 col=col,  
 plot=1)  
  
# For legend  
  
legendTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),  
 x=c(82, 82, 82),  
 y=c(0.2, 0.16, 0.12),  
 plot=1)  
  
  
#options(device="quartz")  
  
# Plot traces using tile  
voting<-tile(nohsTrace,  
 hsTrace,  
 collTrace,  
 labelTrace,  
 legendTrace,  
 limits=c(18,94,0,1),  
 xaxis=list(at=c(20,30,40,50,60,70,80,90)),  
 yaxis=list(label.loc=-0.5, major=FALSE),  
 xaxistitle=list(labels="Age of Respondent"),  
 yaxistitle=list(labels="Probability of Voting"),  
 width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5)  
 #,output=list(file="educationEV",width=5.5)  
)

## Loading required package: WhatIf

## Loading required package: lpSolve

## #######################################################  
## ##  
## ## WhatIf (Version 1.5-6, built 2014-01-06)  
## ## Complete documentation available from http://gking.harvard.edu/whatif   
## ##  
## #######################################################

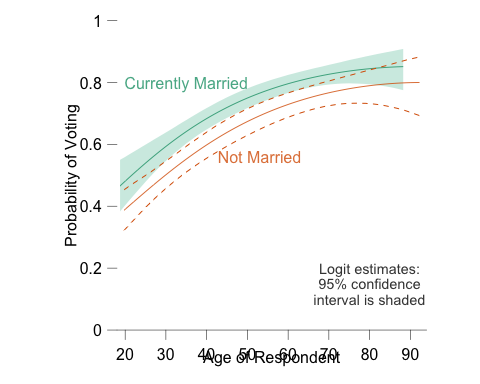
## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"  
## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"  
## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"



# 4. Simulating first differences

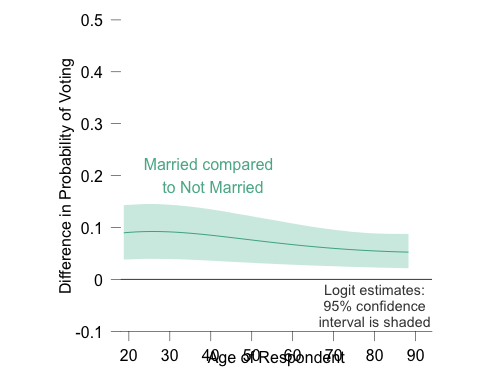
################################################################  
#  
# Now consider a new specification adding the variable  
# "ever married", or marriedo  
#  
# We will estimate this new model with glm(), then  
# simulate new scenarios for marrieds and non-marrieds  
  
  
# Estimate logit model using glm()  
  
  
# Set up a new model formula and model specific data frame  
  
model2 <- vote00 ~ age + I(age^2) + hsdeg + coldeg + marriedo  
mdata2 <- extractdata(model2, data, na.rm=TRUE)  
  
# Run logit & extract results  
logit.m2 <- glm(model2, family=binomial, data=mdata2)  
pe.m2 <- logit.m2$coefficients # point estimates  
vc.m2 <- vcov(logit.m2) # var-cov matrix  
  
  
# Simulate parameter distributions  
sims <- 10000  
simbetas.m2 <- mvrnorm(sims, pe.m2, vc.m2)  
  
  
# Set up counterfactuals: all ages  
  
xhyp <- seq(18,97,1)  
nscen <- length(xhyp)  
marriedScen <- notmarrScen <- cfMake(model2, mdata2, nscen)  
for (i in 1:nscen) {  
   
  
 # - we will use the marriedScen counterfactuals in FDs and RRs as well as EVs  
 # Note below the careful use of before scenarios (xpre) and after scenarios (x)   
 # :i.e., use of the same age range (18-97) for both x and xpre, only marriedo values differ.  
   
 # Married (loop over each age)  
 marriedScen <- cfChange(marriedScen, "age", x = xhyp[i], xpre= xhyp[i], scen = i)  
 marriedScen <- cfChange(marriedScen, "marriedo", x = 1, xpre= 0, scen = i)  
   
 # Not Married (loop over each age)  
 notmarrScen <- cfChange(notmarrScen, "age", x = xhyp[i], scen = i)  
 notmarrScen <- cfChange(notmarrScen, "marriedo", x = 0, scen = i)  
}  
  
# Simulate expected probabilities for all age scenarios for married and not married respectively  
marriedSims <- logitsimev(marriedScen, simbetas.m2, ci=0.95)   
notmarrSims <- logitsimev(notmarrScen, simbetas.m2, ci=0.95)   
  
# Simulate first difference of voting wrt marriage: E(y|married)-E(y|notmarried)  
marriedFD <- logitsimfd(marriedScen, simbetas.m2, ci=0.95)  
  
# Simulate relative risk of voting wrt marriage: E(y|married)/E(y|notmarried)  
marriedRR <- logitsimrr(marriedScen, simbetas.m2, ci=0.95)   
  
  
## Make plots using tile  
  
# Get 3 nice colors for traces  
col <- brewer.pal(3,"Dark2")  
  
# Set up lineplot traces of expected probabilities  
marriedTrace <- lineplot(x=xhyp,  
 y=marriedSims$pe,  
 lower=marriedSims$lower,  
 upper=marriedSims$upper,  
 col=col[1],  
 extrapolate=list(data=mdata2[,2:ncol(mdata2)],  
 cfact=marriedScen$x[,2:ncol(marriedScen$x)],  
 omit.extrapolated=TRUE),  
 plot=1)  
  
notmarrTrace <- lineplot(x=xhyp,  
 y=notmarrSims$pe,  
 lower=notmarrSims$lower,  
 upper=notmarrSims$upper,  
 col=col[2],  
 ci = list(mark="dashed"),  
 extrapolate=list(data=mdata2[,2:ncol(mdata2)],  
 cfact=notmarrScen$x[,2:ncol(notmarrScen$x)],  
 omit.extrapolated=TRUE),  
 plot=1)  
  
  
# Set up traces with labels and legend  
labelTrace <- textTile(labels=c("Currently Married", "Not Married"),  
 x=c( 35, 53),  
 y=c( 0.8, 0.56),  
 col=col,  
 plot=1)  
  
legendTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),  
 x=c(80, 80, 80),  
 y=c(0.2, 0.15, 0.10),  
 cex=0.9,  
 plot=1)  
  
# Plot traces using tile  
tile(marriedTrace,  
 notmarrTrace,  
 labelTrace,  
 legendTrace,  
 limits=c(18,94,0,1),  
 xaxis=list(at=c(20,30,40,50,60,70,80,90)),  
 yaxis=list(label.loc=-0.5, major=FALSE),  
 xaxistitle=list(labels="Age of Respondent"),  
 yaxistitle=list(labels="Probability of Voting"),  
 width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5)  
 #,output=list(file="marriedEV",width=5.5)  
)

## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"  
## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"



# Plot First Difference  
  
# Set up lineplot trace of first difference  
  
marriedFDTrace <- lineplot(x=xhyp,  
 y=marriedFD$pe,  
 lower=marriedFD$lower,  
 upper=marriedFD$upper,  
 col=col[1],  
 extrapolate=list(data=mdata2[,2:ncol(mdata2)],  
 cfact=marriedScen$x[,2:ncol(marriedScen$x)],  
 omit.extrapolated=TRUE),  
 plot=1)  
  
  
# Set up baseline: for first difference, this is 0  
baseline <- linesTile(x=c(18,94),  
 y=c(0,0),  
 plot=1)  
  
# Set up traces with labels and legend  
labelFDTrace <- textTile(labels=c("Married compared \n to Not Married"),  
 x=c( 40),  
 y=c( 0.20),  
 col=col[1],  
 plot=1)  
  
legendFDTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),  
 x=c(80, 80, 80),  
 y=c(-0.02, -0.05, -0.08),  
 cex=0.9,  
 plot=1)  
  
# Plot traces using tile  
tile(marriedFDTrace,  
 labelFDTrace,  
 legendFDTrace,  
 baseline,  
 limits=c(18,94,-0.1,0.5),  
 xaxis=list(at=c(20,30,40,50,60,70,80,90)),  
 yaxis=list(label.loc=-0.5, major=FALSE),  
 xaxistitle=list(labels="Age of Respondent"),  
 yaxistitle=list(labels="Difference in Probability of Voting"),  
 width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5)  
 #,output=list(file="marriedFD",width=5.5)  
)

## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"



# Plot Relative Risk  
  
# Set up lineplot trace of relative risk  
marriedRRTrace <- lineplot(x=xhyp,  
 y=marriedRR$pe,  
 lower=marriedRR$lower,  
 upper=marriedRR$upper,  
 col=col[1],  
 extrapolate=list(data=mdata2[,2:ncol(mdata2)],  
 cfact=marriedScen$x[,2:ncol(marriedScen$x)],  
 omit.extrapolated=TRUE),  
 plot=1)  
  
  
# Set up baseline: for relative risk, this is 1  
  
baseline <- linesTile(x=c(18,94),  
 y=c(1,1),  
 plot=1)  
  
# Set up traces with labels and legend  
labelRRTrace <- textTile(labels=c("Married compared \n to Not Married"),  
 x=c( 55),  
 y=c( 1.25),  
 col=col[1],  
 plot=1)  
  
legendRRTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),  
 x=c(80, 80, 80),  
 y=c(0.98, 0.95, 0.92),  
 cex=0.9,  
 plot=1)  
  
# Plot traces using tile  
tile(marriedRRTrace,  
 labelRRTrace,  
 legendRRTrace,  
 baseline,  
 limits=c(18,94,0.9,1.5),  
 xaxis=list(at=c(20,30,40,50,60,70,80,90)),  
 yaxis=list(label.loc=-0.5, major=FALSE),  
 xaxistitle=list(labels="Age of Respondent"),  
 yaxistitle=list(labels="Relative Risk of Voting"),  
 width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5)  
 #,output=list(file="marriedRR",width=5.5)  
)

## [1] "Running whatif"  
## [1] "Preprocessing data ..."  
## [1] "Performing convex hull test ..."  
## [1] "Calculating distances ...."  
## [1] "Calculating the geometric variance..."  
## [1] "Calculating cumulative frequencies ..."  
## [1] "Finishing up ..."  
## [1] "Whatif finished; returning to tile"

