MA678 Homework 4

JingjianGao

10/4/2022

13.5 Interpreting logistic regression coefficients

Here is a fitted model from the Bangladesh analysis predicting whether a person with high-arsenic drinking water will switch wells, given the arsenic level in their existing well and the distance to the nearest safe well:

```
stan_glm(formula = switch ~ dist100 + arsenic, family=binomial(link="logit"), data=wells)
             Median MAD_SD
(Intercept)   0.00   0.08
dist100      -0.90   0.10
arsenic       0.46   0.04
```

Compare two people who live the same distance from the nearest well but whose arsenic levels differ, with one person having an arsenic level of 0.5 and the other person having a level of 1.0. You will estimate how much more likely this second person is to switch wells. Give an approximate estimate, standard error, 50% interval, and 95% interval, using two different methods:

(a)

Use the divide-by-4 rule, based on the information from this regression output.

```
# The approximate estimate is 0.46/4=11.5%
# The standard Error is 0.04, as shown
# The 50% interval is 0.115 +- 0.67*0.04 =[0.0882,0.142]
# The 95% interval is 0.115 +- 1.96*0.04 =[0.0366,0.193]
```

(b)

Use predictive simulation from the fitted model in R, under the assumption that these two people each live 50 meters from the nearest safe well.

```
library(rstanarm)
```

```
## Loading required package: Rcpp

## This is rstanarm version 2.21.3

## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

##   options(mc.cores = parallel::detectCores())
```

```
wells <- read.csv("/Users/billg/Desktop/MA-678-Homework/MA678-HW4/wells.csv")
Reg13.5 <- stan_glm(formula = switch ~ dist100 + arsenic, family=binomial(link="logit"), data=wells,ref
predic13.5 <- posterior_epred(Reg13.5,data=wells)
mean(predic13.5)
```

```
## [1] 0.5751456
```

```r
sd(predic13.5)
```

```
## [1] 0.1205389
```

13.7 Graphing a fitted logistic regression

We downloaded data with weight (in pounds) and age (in years) from a random sample of American adults. We then defined a new variable:

```
heavy <- weight > 200
```

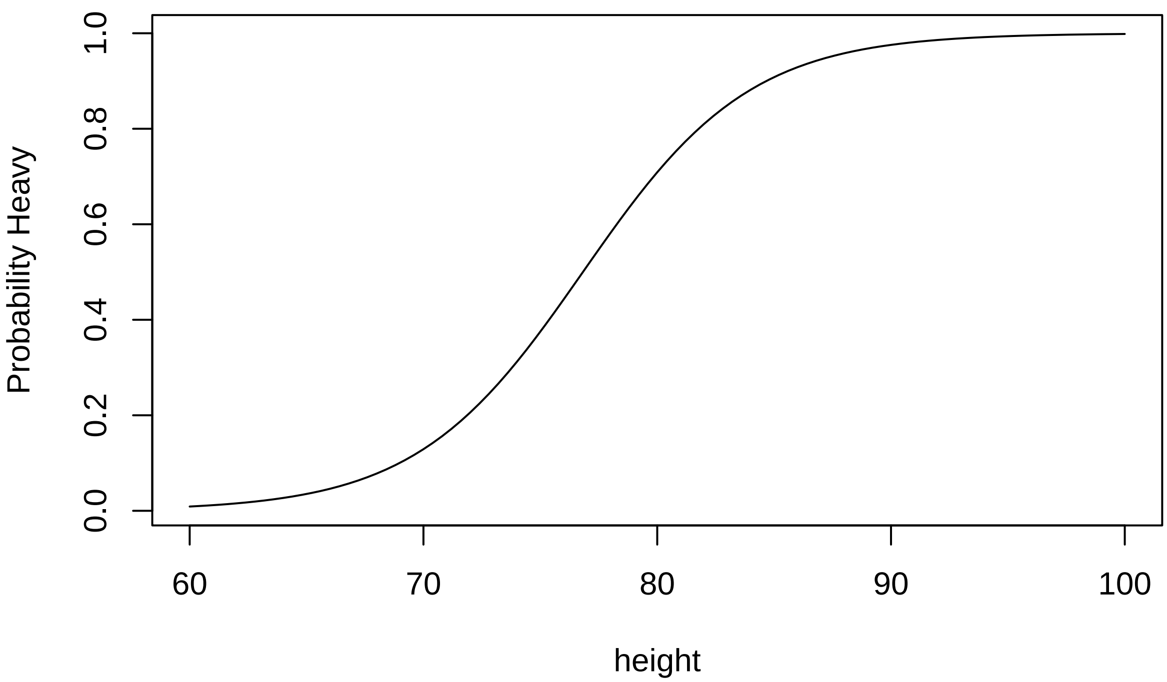
and fit a logistic regression, predicting heavy from `height` (in inches):

```
stan_glm(formula = heavy ~ height, family=binomial(link="logit"), data=health)
              Median MAD_SD
(Intercept)  -21.51   1.60
height         0.28   0.02
```

(a)

Graph the logistic regression curve (the probability that someone is heavy) over the approximate range of the data. Be clear where the line goes through the 50% probability point.

```r
curve(invlogit(-21.51+0.28*x),xlab="height",ylab="Probability Heavy",xlim=c(60,100))
```




```r
# The line goes through the 50% point when -21.51+0.28*x=0. Thus x=76.82
```

(b)

Fill in the blank: near the 50% point, comparing two people who differ by one inch in height, you'll expect a difference of ______ in the probability of being heavy.

```r
# By using the divide by 4 rule, we will expect a difference of 0.28/4= 7%
# in the probability of being heavy.
```

13.8 Linear transformations

In the regression from the previous exercise, suppose you replaced height in inches by height in centimeters. What would then be the intercept and slope?

```
# 1 inch is 2.54 centimeters. Therefore, after we replace it with height in centimeters,
# the intercept is gonna be the same and the slope is gonna be 1/2.54 times the original slope
```

13.10 Expressing a comparison of proportions as a logistic regression

A randomized experiment is performed within a survey, and 1000 people are contacted. Half the people contacted are promised a $5 incentive to participate, and half are not promised an incentive. The result is a 50% response rate among the treated group and 40% response rate among the control group.

(a)

Set up these results as data in R. From these data, fit a logistic regression of response on the treatment indicator.

```
library(arm)
```

```
## Loading required package: MASS

## Loading required package: Matrix

## Loading required package: lme4

##
## arm (Version 1.13-1, built: 2022-8-25)

## Working directory is /Users/billg/Desktop/MA-678-Homework/MA678-HW4

##
## Attaching package: 'arm'

## The following objects are masked from 'package:rstanarm':
##
##     invlogit, logit
```

```
set.seed(114514)
experiment <- c(rep(1,500),rep(0,500))
response_rate <- c(rep(0,250),rep(1,250),rep(0,300),rep(1,200))
Reg13.10 <- stan_glm(response_rate~experiment,family =binomial(link=logit),refresh=0)
```

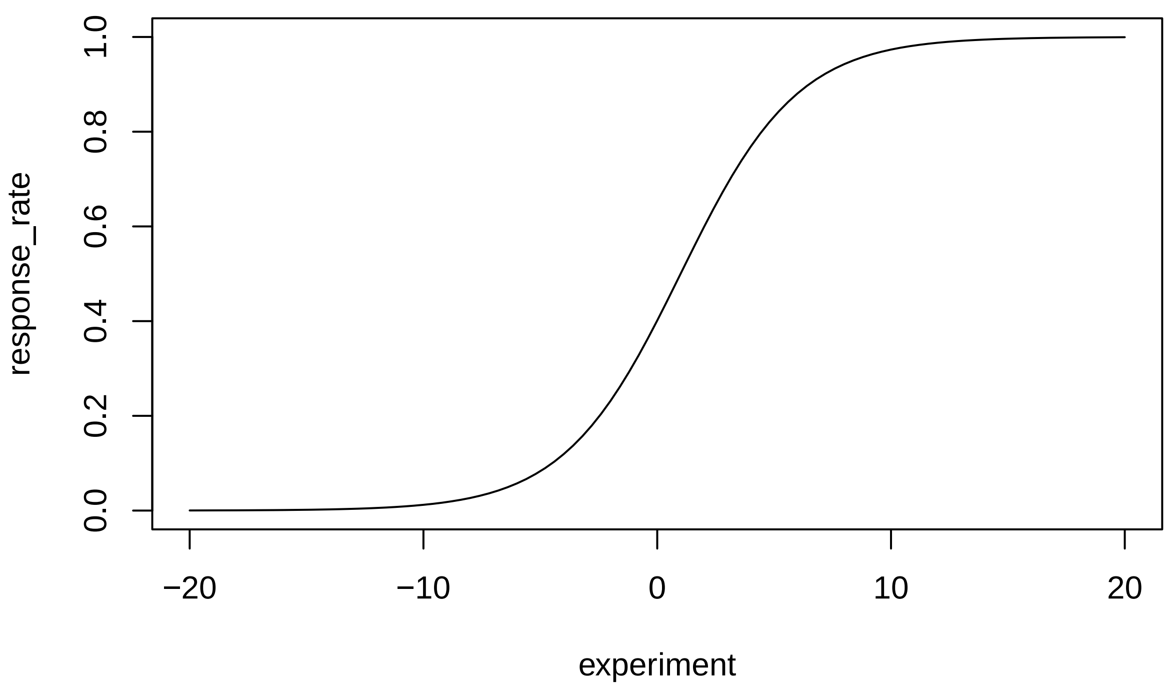
```
## Warning: Omitting the 'data' argument is not recommended and may not be allowed
## in future versions of rstanarm. Some post-estimation functions (in particular
## 'update', 'loo', 'kfold') are not guaranteed to work properly unless 'data' is
## specified as a data frame.
```

```
summary(Reg13.10)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      response_rate ~ experiment
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1000
```

```
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -0.4    0.1 -0.5  -0.4  -0.3
## experiment   0.4    0.1  0.2   0.4   0.6
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.5    0.0  0.4   0.4   0.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2432
## experiment    0.0  1.0  2507
## mean_PPD      0.0  1.0  3194
## log-posterior 0.0  1.0  1674
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
curve(invlogit(-0.4+0.4*x),xlab="experiment",ylab="response_rate",xlim=c(-20,20))
```



(b)

Compare to the results from Exercise 4.1.

```
set.seed(114514)
experiment <- c(rep(1,500),rep(0,500))
response_rate <- c(rep(0,250),rep(1,250),rep(0,300),rep(1,200))
Reg13.10b <- lm(response_rate~experiment,family =binomial(link=logit),refresh=0)
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
```

```
##  extra arguments 'family', 'refresh' will be disregarded
```

```r
summary(Reg13.10b)
```

```
##
## Call:
## lm(formula = response_rate ~ experiment, family = binomial(link = logit),
##     refresh = 0)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.500 -0.425 -0.400  0.500  0.600
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.40000    0.02216  18.052  < 2e-16 ***
## experiment   0.10000    0.03134   3.191  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4955 on 998 degrees of freedom
## Multiple R-squared:  0.0101, Adjusted R-squared:  0.009109
## F-statistic: 10.18 on 1 and 998 DF,  p-value: 0.001461
```

```r
# The results are somewhat consistent
```

13.11 Building a logistic regression model

The folder `Rodents` contains data on rodents in a sample of New York City apartments.

(a)

Build a logistic regression model to predict the presence of rodents (the variable `rodent2` in the dataset) given indicators for the ethnic groups (`race`). Combine categories as appropriate. Discuss the estimated coefficients in the model.

```r
rodents <- read.table("/Users/billg/Desktop/MA-678-Homework/MA678-HW4/rodents.csv")
Reg13.11 <- stan_glm(rodent2~race,data=rodents,family =binomial(link =logit),refresh=0)
summary(Reg13.11)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ race
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1551
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -1.9    0.1 -2.1  -1.9  -1.8
## race         0.3    0.0  0.3   0.3   0.4
##
```

```
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.2    0.0  0.2   0.2   0.3  
## 
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
## 
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2419 
## race          0.0  1.0  2642 
## mean_PPD      0.0  1.0  2949 
## log-posterior 0.0  1.0  1742 
## 
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
# The estimate intercept coefficient is -1.9 meaning the average is -1.9 when race=0
```

(b)

Add to your model some other potentially relevant predictors describing the apartment, building, and community district. Build your model using the general principles explained in Section 12.6. Discuss the coefficients for the ethnicity indicators in your model.

```r
Reg13.11b <- stan_glm(rodent2~race+personrm+housewgt+sequenceno,data=rodents,family =binomial(link =log
summary(Reg13.11b)
```

```
## 
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ race + personrm + housewgt + sequenceno
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1551
##  predictors:   5
## 
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -1.3    0.3 -1.7  -1.3  -0.9 
## race         0.2    0.0  0.2   0.2   0.3 
## personrm     0.8    0.1  0.7   0.8   1.0 
## housewgt     0.0    0.0  0.0   0.0   0.0 
## sequenceno   0.0    0.0  0.0   0.0   0.0 
## 
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.2    0.0  0.2   0.2   0.3  
## 
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
## 
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  4990 
## race          0.0  1.0  4175 
```

```
## personrm      0.0  1.0  4680
## housewgt      0.0  1.0  4902
## sequenceno    0.0  1.0  5528
## mean_PPD      0.0  1.0  4731
## log-posterior 0.0  1.0  1747
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
# The coefficients of the indicators are shown below.
```

14.3 Graphing logistic regressions

The well-switching data described in Section 13.7 are in the folder `Arsenic`.

(a)

Fit a logistic regression for the probability of switching using log (distance to nearest safe well) as a predictor.

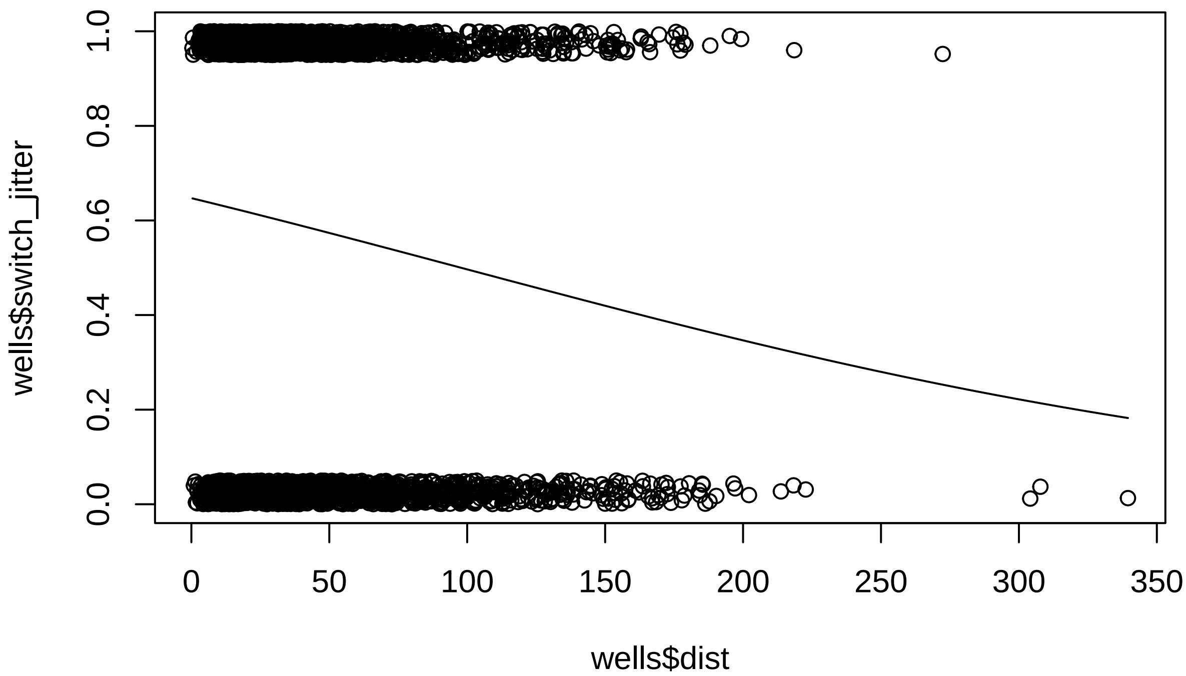
```
wells <- read.csv("/Users/billg/Desktop/MA-678-Homework/MA678-HW4/wells.csv")
Reg14.3 <- stan_glm(switch~log(dist),data=wells,family =binomial(link =logit),refresh=0)
summary(Reg14.3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ log(dist)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  1.0    0.2  0.8   1.0   1.2
## log(dist)   -0.2    0.0 -0.3  -0.2  -0.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.6    0.0  0.6   0.6   0.6
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2974
## log(dist)     0.0  1.0  2959
## mean_PPD      0.0  1.0  3243
## log-posterior 0.0  1.0  1724
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(b)

Make a graph similar to Figure 13.8b displaying Pr(switch) as a function of distance to nearest safe well, along with the data.

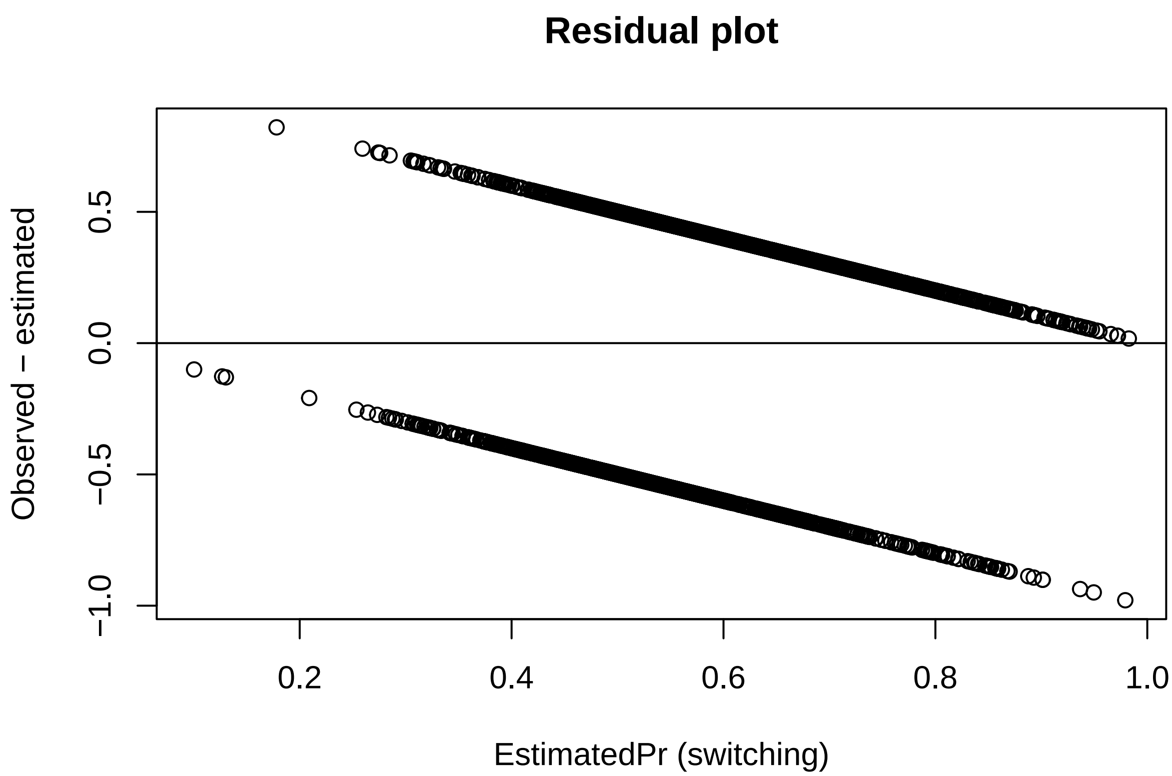
```
fit_1 <- stan_glm(switch ~ dist, family=binomial(link="logit"), data=wells,refresh=0)
jitter_binary <- function(a, jitt=0.05){
     ifelse(a==0, runif(length(a), 0, jitt), runif(length(a), 1 - jitt, 1))
}
wells$switch_jitter <- jitter_binary(wells$switch)
plot(wells$dist, wells$switch_jitter)
curve(invlogit(coef(fit_1)[1] + coef(fit_1)[2]*x), add=TRUE)
```



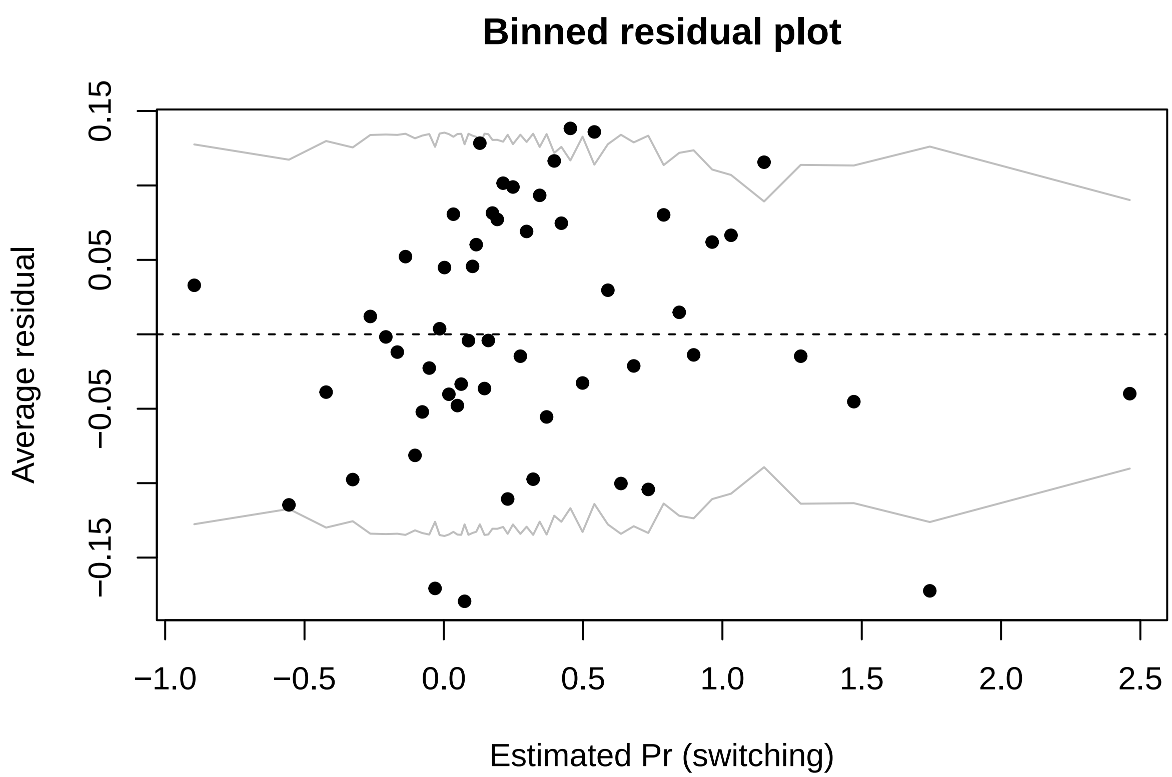
(c)

Make a residual plot and binned residual plot as in Figure 14.8.

```
library(arm)
fit_4 <- stan_glm(switch ~ dist100 + arsenic + dist100:arsenic,
     family=binomial(link="logit"), data=wells,refresh=0)
res <- resid(fit_4)
plot(fitted(fit_4), res,xlab="EstimatedPr (switching)",ylab="Observed - estimated",main="Residual plot"
abline(0,0)
```




```
x <- predict(fit_4)
binnedplot(x,res,xlab="Estimated Pr (switching)")
```



(d)

Compute the error rate of the fitted model and compare to the error rate of the null model.

```
fitted <- fitted(fit_4)
error <- mean(abs(wells$switch-mean(fitted)))
round(error,3)
```

```
## [1] 0.489
```

```
error_null <- mean(abs(wells$switch-fitted))
round(error_null,3)
```

```
## [1] 0.459
```

(e)

Create indicator variables corresponding to `dist < 100`; `dist` between 100 and 200; and `dist > 200`. Fit a logistic regression for Pr(switch) using these indicators. With this new model, repeat the computations and graphs for part (a) of this exercise.

```
variable <- NULL
variable[wells$dist<100] <- 0
variable[wells$dist>100&wells$dist<200] <- 1
variable[wells$dist>200] <- 2
Reg14.3e <-stan_glm(wells$switch~variable,family =binomial(link =logit),refresh=0)
```

```
## Warning: Omitting the 'data' argument is not recommended and may not be allowed
## in future versions of rstanarm. Some post-estimation functions (in particular
## 'update', 'loo', 'kfold') are not guaranteed to work properly unless 'data' is
## specified as a data frame.
```

```
summary(Reg14.3e)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      wells$switch ~ variable
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   2
##
## Estimates:
##                mean   sd   10%   50%   90%
## (Intercept)  0.4    0.0  0.3   0.4   0.4
## variable    -0.7    0.1 -0.8  -0.7  -0.5
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.6    0.0  0.6   0.6   0.6
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                mcse Rhat n_eff
```

```
## (Intercept)   0.0  1.0  3094
## variable      0.0  1.0  2927
## mean_PPD      0.0  1.0  3258
## log-posterior 0.0  1.0  1707
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

14.7 Model building and comparison

Continue with the well-switching data described in the previous exercise.

(a)

Fit a logistic regression for the probability of switching using, as predictors, distance, log(arsenic), and their interaction. Interpret the estimated coefficients and their standard errors.

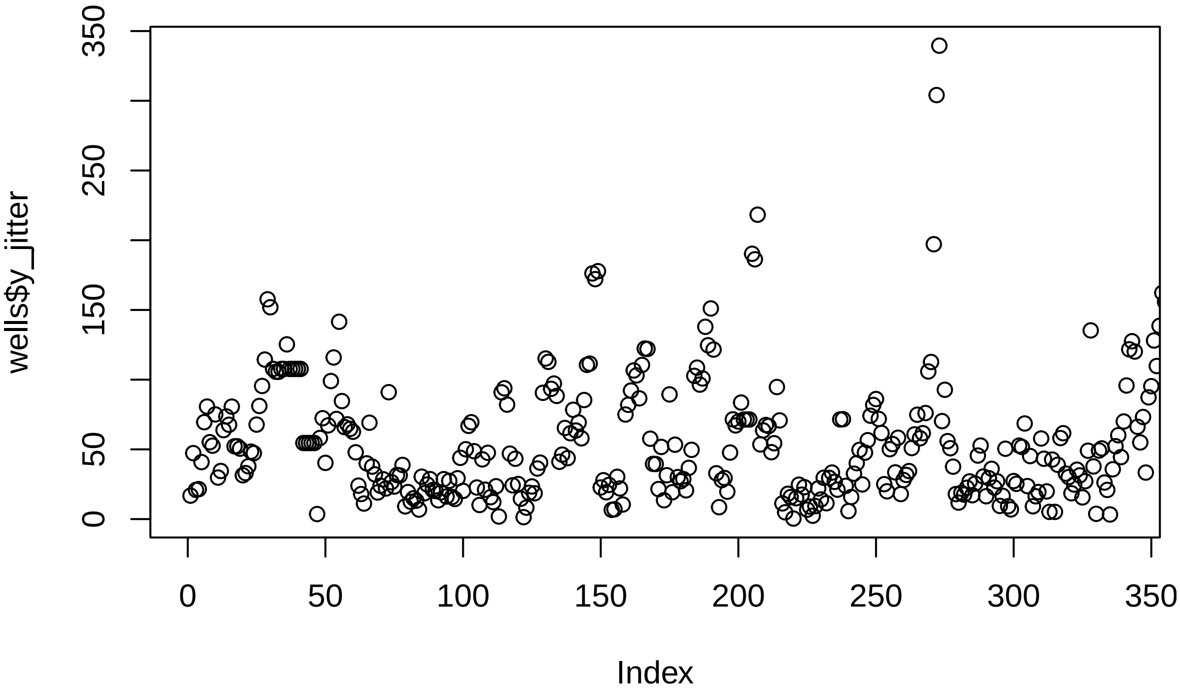
```
wells <- read.csv("wells.csv")
Reg14.7 <- stan_glm(switch~dist+log(arsenic)+dist*log(arsenic),data=wells, family=binomial(link="logit")
summary(Reg14.7)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist + log(arsenic) + dist * log(arsenic)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   4
##
## Estimates:
##                     mean   sd   10%   50%   90%
## (Intercept)       0.5    0.1  0.4   0.5   0.6
## dist              0.0    0.0  0.0   0.0   0.0
## log(arsenic)      1.0    0.1  0.8   1.0   1.1
## dist:log(arsenic) 0.0    0.0  0.0   0.0   0.0
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.6    0.0  0.6   0.6   0.6
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                   mcse Rhat n_eff
## (Intercept)       0.0  1.0  2344
## dist              0.0  1.0  1965
## log(arsenic)      0.0  1.0  1606
## dist:log(arsenic) 0.0  1.0  1396
## mean_PPD          0.0  1.0  3047
## log-posterior     0.0  1.0  1242
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

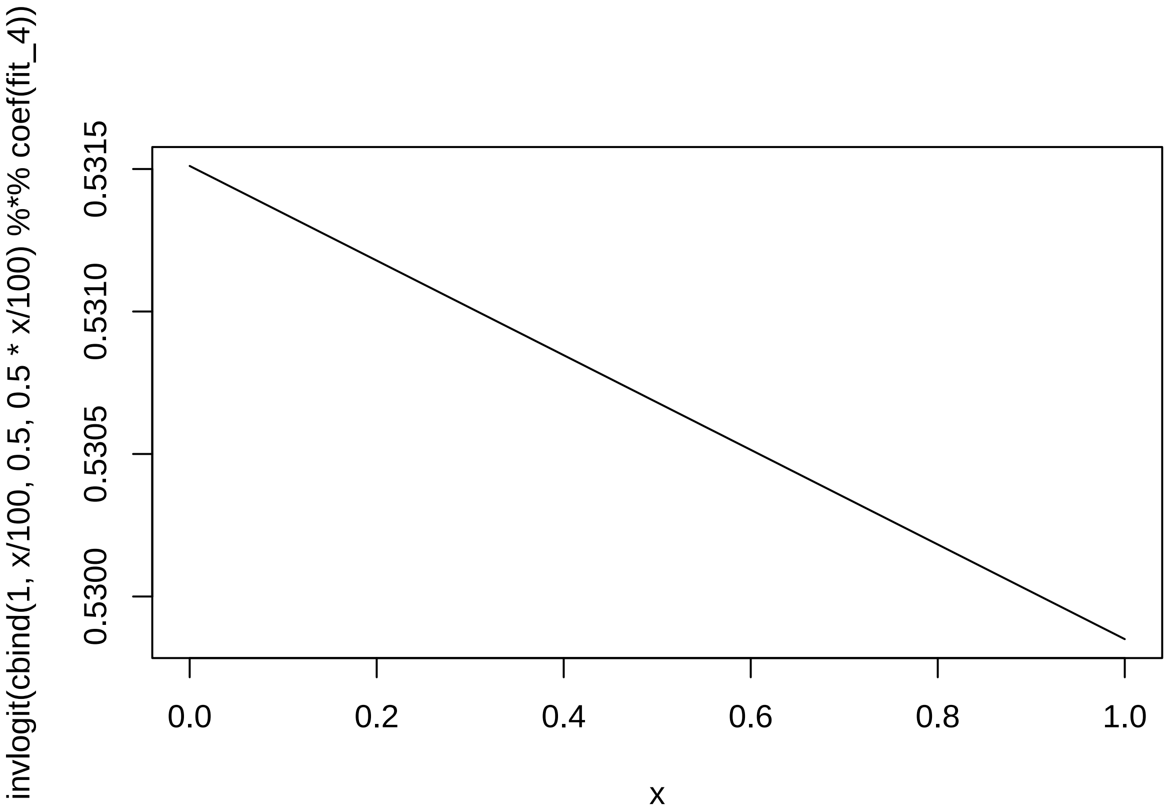
(b)

Make graphs as in Figure 14.3 to show the relation between probability of switching, distance, and arsenic level.

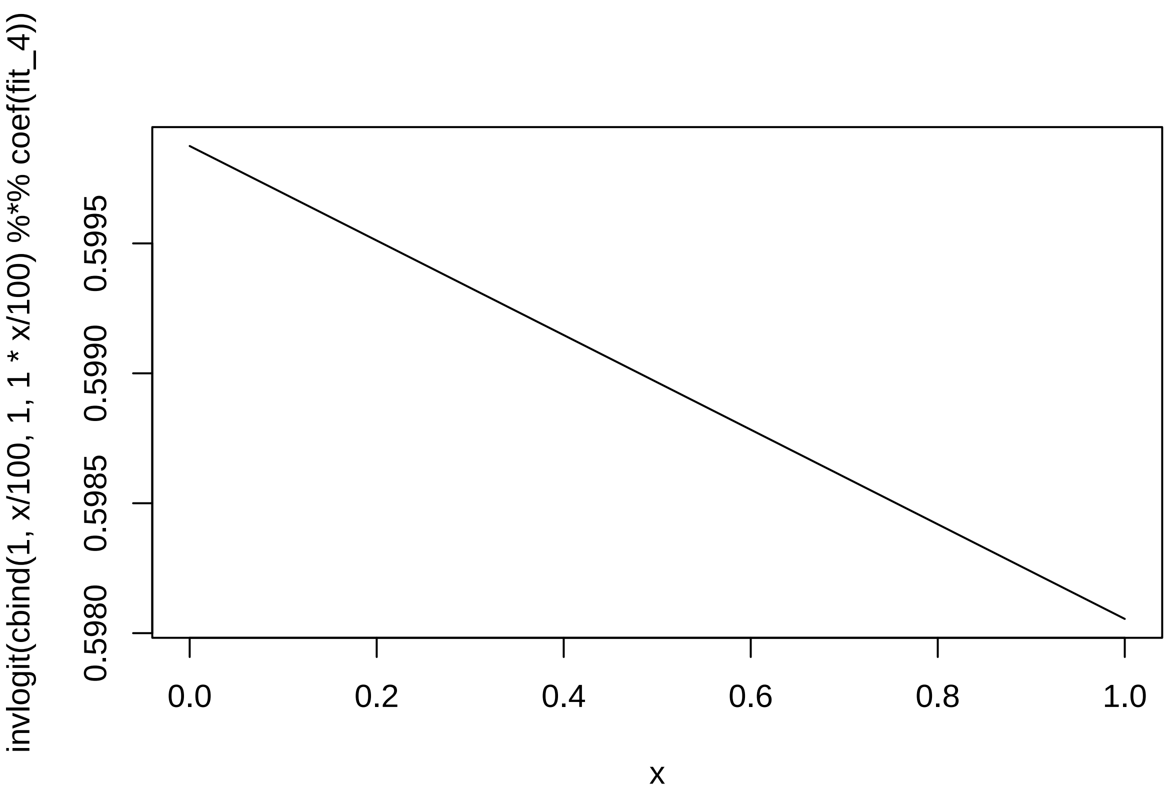
```
plot(wells$dist, wells$y_jitter, xlim=c(0,max(wells$dist)))
```




```
curve(invlogit(cbind(1, x/100, 0.5, 0.5*x/100) %*% coef(fit_4)))
```

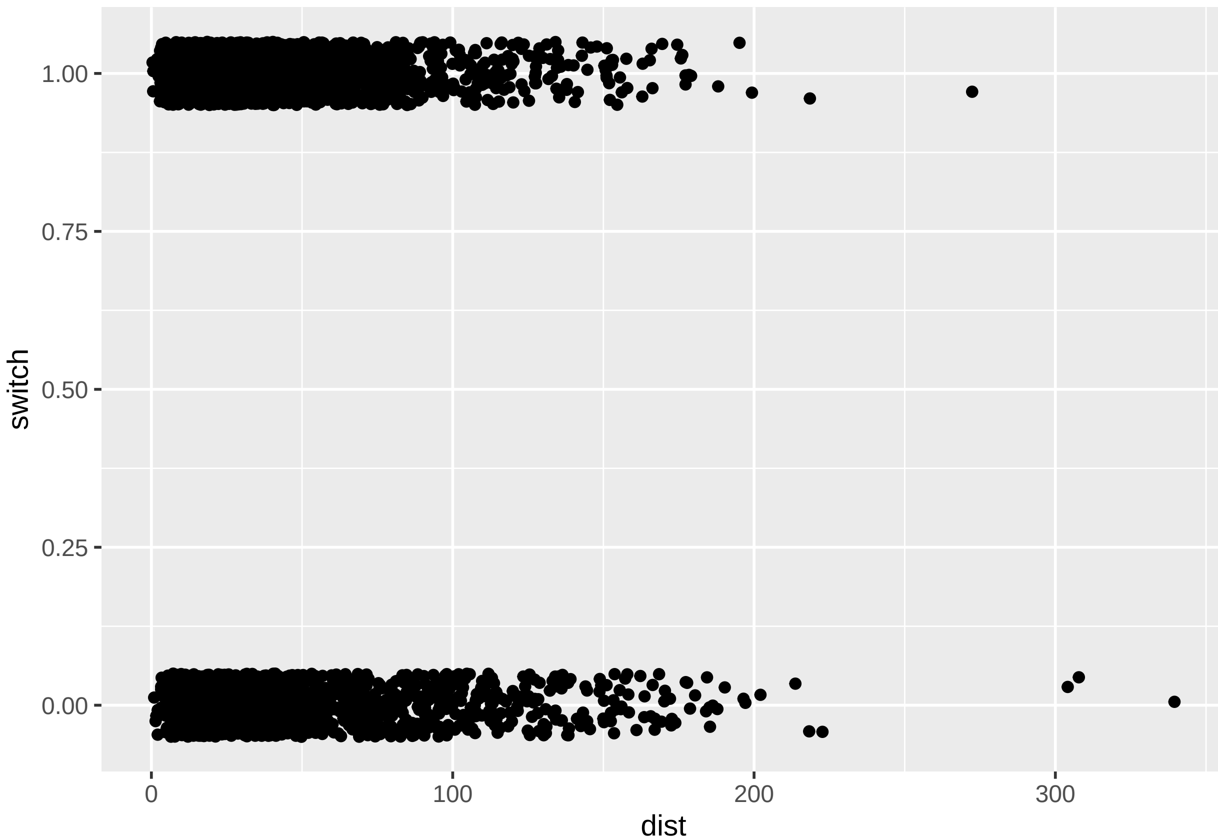



```
curve(invlogit(cbind(1, x/100, 1.0, 1.0*x/100) %*% coef(fit_4)))
```




```
library(ggplot2)
```

```
ggplot(wells,aes(dist,switch))+
  geom_jitter(position=position_jitter(height=0.05))
```



(c)

Following the procedure described in Section 14.4, compute the average predictive differences corresponding to:

i. A comparison of `dist` = 0 to `dist` = 100, with `arsenic` held constant.

ii. A comparison of `dist` = 100 to `dist` = 200, with `arsenic` held constant.

iii. A comparison of `arsenic` = 0.5 to `arsenic` = 1.0, with `dist` held constant.

iv. A comparison of `arsenic` = 1.0 to `arsenic` = 2.0, with `dist` held constant.

Discuss these results.

```
Reg14.7 <- stan_glm(switch~dist+log(arsenic)+dist*log(arsenic),data=wells, family=binomial(link="logit"
b <- coef(Reg14.7)
#i
hi <- 100
lo <- 0
delta <- invlogit(b[1] + b[2]*hi + b[3]*log(wells$arsenic) + b[4]*log(wells$arsenic)*hi) -
          invlogit(b[1] + b[2]*lo + b[3]*log(wells$arsenic) + b[4]*log(wells$arsenic)*lo)
round(mean(delta), 2)
```

```
## [1] -0.21
```

```r
#ii
hi <- 200
lo <- 100
delta2 <- invlogit(b[1] + b[2]*hi + b[3]*log(wells$arsenic) + b[4]*log(wells$arsenic)*hi) -
          invlogit(b[1] + b[2]*lo + b[3]*log(wells$arsenic) + b[4]*log(wells$arsenic)*lo)
round(mean(delta2), 2)
```

```
## [1] -0.21
```

```r
#iii
hi <- 1.0
lo <- 0.5
delta3 <- invlogit(b[1] + b[2]*wells$dist + b[3]*hi + b[4]*log(wells$dist)*hi) -
          invlogit(b[1] + b[2]*wells$dist + b[3]*lo + b[4]*log(wells$dist)*lo)
round(mean(delta3), 2)
```

```
## [1] 0.1
```

```r
#iv
hi <- 2.0
lo <- 1.0
delta4 <- invlogit(b[1] + b[2]*wells$dist + b[3]*hi + b[4]*log(wells$dist)*hi) -
          invlogit(b[1] + b[2]*wells$dist + b[3]*lo + b[4]*log(wells$dist)*lo)
round(mean(delta4), 2)
```

```
## [1] 0.14
```