MA678 Homework 4

Reese Mullen

10/4/2022

Disclaimer (remove after you've read)!

A few things to keep in mind :
1) Use `set.seed()` to make sure that the document produces the same random simulation as when you ran the code.
2) Use `refresh=0` for any `stan_glm()` or stan-based model. `lm()` or non-stan models don't need this!
3) You can type outside of the R chunks and make new R chunks where it's convenient. Make sure it's clear which questions you're answering.
4) Even if you're not too confident, please try giving an answer to the text responses!
5) Please don't print data in the document unless the question asks. It's good for you to do it to look at the data, but not as good for someone trying to read the document later on.
6) Check your document before submitting! Please put your name where "Your Name" is by the author!

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 person with an arsenic level of 1 is 5.75% more likely to switch than a person with a level of 0.5. The standard error of arsenic is 0.01, so the 95% CI is from 0.113 to 0.117 and the 50% CI is from 0.108 to 0.122.

(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.

```r
wells <- read.csv("~/Downloads/wells.csv", header = TRUE)

model13.5b <- stan_glm(switch ~ dist100 + arsenic, binomial(link = "logit"), data = wells, refresh = 0)
summary(model13.5b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist100 + arsenic
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   3
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  0.0    0.1 -0.1   0.0   0.1
## dist100     -0.9    0.1 -1.0  -0.9  -0.8
## arsenic      0.5    0.0  0.4   0.5   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  3736
## dist100       0.0  1.0  3460
## arsenic       0.0  1.0  3281
## mean_PPD      0.0  1.0  3748
## log-posterior 0.0  1.0  1941
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
newdata = data.frame(dist100 = c(0.5, 0.5), arsenic = c(0.5, 1))
pred13.5 <- invlogit(posterior_linpred(model13.5b, newdata = newdata))

estimate <- sprintf("%.4f", mean(pred13.5[, 2] - pred13.5[, 1]))
mse13.5 <- sprintf("%.4f", sd(pred13.5[, 2] - pred13.5[, 1]))
cat("The estimate is:", estimate)
```

```
## The estimate is: 0.0577
```

```r
cat("The standard error is:", mse13.5)
```

```
## The standard error is: 0.0050
```

```
ci5013.5b <- sprintf("%.4f", quantile(pred13.5[, 2] - pred13.5[, 1], c(0.25, 0.75)))
ci9513.5b <- sprintf("%.4f", quantile(pred13.5[, 2] - pred13.5[, 1], c(0.025, 0.975)))
cat("The 50% CI is: (", paste(ci5013.5b, collapse = ", ") , ")")
```

```
## The 50% CI is: ( 0.0543, 0.0611 )
```

```
cat("The 95% CI is: (", paste(ci9513.5b, collapse = ", ") , ")")
```

```
## The 95% CI is: ( 0.0478, 0.0672 )
```

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):

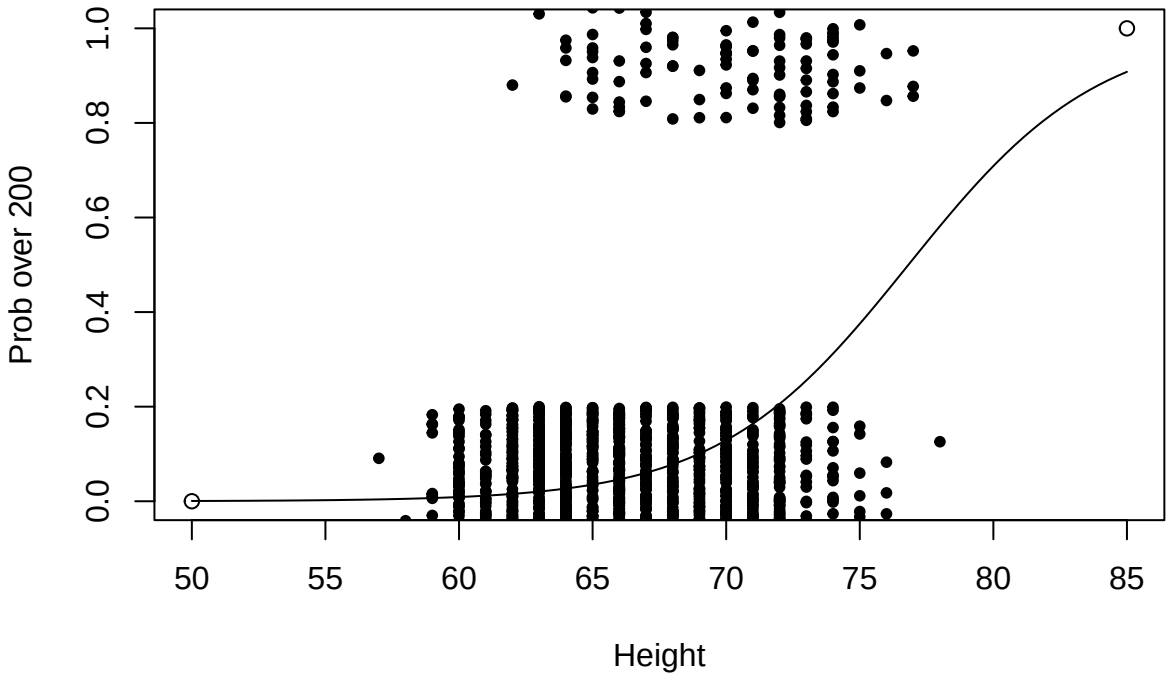
```
stan_glm(formula = heavy ~ height, family=binomial(link="logit"), data=health)
             Median MAD_SD
(Intercept) -21.51   1.60
height        0.28   0.02
```

(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.

```
earnings <- read.csv("~/Downloads/earnings.csv", header = TRUE)
earnings$heavy = ifelse(earnings$weight > 200, 1, 0)

plot(c(50, 85), c(0, 1),
     xlab = "Height",
     ylab = "Prob over 200")
points(earnings$height, jitter(earnings$heavy), pch = 20)
curve(invlogit(-21.51 + 0.28 * x), add = TRUE)
```



(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 0.07 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?

The conversion from height in inches to cm is 2.54 cm per 1 inch so we would multiply the coefficient for height by 2.54.

The new equation would be log(weight over 200)= -21.51 + 0.7112*height.

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.

```r
treat <- 500
control <- 500

response_treated <- rbinom(treat, 1, 0.50)
response_control <- rbinom(control, 1, 0.40)

data13.10a <- data.frame(
  response = c(response_treated, response_control),
  treatment = c(rep(1, treat), rep(0, control))
)

model13.10a <- stan_glm(response ~ treatment, family = binomial(link = "logit"), data = data13.10a, ref

summary(model13.10a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      response ~ treatment
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1000
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -0.5    0.1 -0.7  -0.5  -0.4
## treatment    0.6    0.1  0.4   0.6   0.7
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.4    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  2808
## treatment     0.0  1.0  2827
## mean_PPD      0.0  1.0  3381
## log-posterior 0.0  1.0  1921
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(b)

Compare to the results from Exercise 4.1.

```r
estimate13.10b <- 0.5 - 0.4
mse13.10b <- sprintf("%.4f", sqrt(0.5 ^ 2 / 500 + 0.5 ^ 2 / 500))
cat("The estimate of the average treatment effect is:", estimate13.10b)
```

```
## The estimate of the average treatment effect is: 0.1
```

```
cat("The standard error of the average treatment effect is:", mse13.10b)
```

```
## The standard error of the average treatment effect is: 0.0316
```

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.

THe log odds for the first five races are all significant meaning they are 2 to 7 times more likely to have rodents than race 1. The last two groups are not statistically significant.

```
rodents<-read.table("~/Downloads/rodents.dat", header = TRUE)
model13.11 <- stan_glm(rodent2 ~ as.factor(race), family = binomial(link = "logit"), data = rodents, re

summary(model13.11)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ as.factor(race)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1551
##  predictors:   7
##
## Estimates:
##                    mean   sd   10%   50%   90%
## (Intercept)      -2.2    0.1 -2.4  -2.2  -2.0
## as.factor(race)2  1.4    0.2  1.2   1.4   1.6
## as.factor(race)3  1.7    0.2  1.4   1.7   1.9
## as.factor(race)4  2.0    0.2  1.8   2.0   2.2
## as.factor(race)5  0.8    0.3  0.5   0.8   1.1
## as.factor(race)6  0.1    1.4 -1.7   0.3   1.7
## as.factor(race)7 -0.1    1.3 -1.7   0.1   1.4
##
## 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  1334
## as.factor(race)2  0.0  1.0  1787
## as.factor(race)3  0.0  1.0  2035
## as.factor(race)4  0.0  1.0  1969
## as.factor(race)5  0.0  1.0  2542
## as.factor(race)6  0.0  1.0  2550
## as.factor(race)7  0.0  1.0  2758
## mean_PPD          0.0  1.0  3865
## log-posterior     0.0  1.0  1625
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(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. The same patterns exist for the race variables in this model as the last one.

```
model13.11b <- rstanarm::stan_glm(rodent2 ~ as.factor(race) + as.factor(borough) +
                poverty + extflr5_2 + intcrack2 + inthole2,
              family = "binomial"(link = "logit"), data = rodents, refresh= 0)
summary(model13.11b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ as.factor(race) + as.factor(borough) + poverty + extflr5_2 +
##     intcrack2 + inthole2
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1421
##  predictors:   15
##
## Estimates:
##                        mean   sd    10%   50%   90%
## (Intercept)          -2.0    0.2  -2.3  -2.0  -1.7
## as.factor(race)2      1.2    0.2   0.9   1.2   1.4
## as.factor(race)3      1.3    0.3   1.0   1.3   1.7
## as.factor(race)4      1.9    0.2   1.6   1.9   2.2
## as.factor(race)5      0.9    0.3   0.6   0.9   1.3
## as.factor(race)6      0.2    1.7  -2.0   0.4   2.2
## as.factor(race)7    -30.8   22.5 -61.7 -26.2  -6.1
## as.factor(borough)2  -0.1    0.2  -0.3  -0.1   0.2
## as.factor(borough)3  -0.3    0.2  -0.6  -0.3  -0.1
## as.factor(borough)4  -1.0    0.2  -1.3  -1.0  -0.7
## as.factor(borough)5  -2.3    0.8  -3.4  -2.2  -1.3
## poverty               0.0    0.2  -0.2   0.0   0.2
## extflr5_2             0.8    0.3   0.4   0.8   1.3
## intcrack2             1.1    0.2   0.8   1.1   1.4
```

```
## inthole2              1.3    0.3   0.9   1.3   1.7
##
## 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  1986
## as.factor(race)2     0.0  1.0  2612
## as.factor(race)3     0.0  1.0  2789
## as.factor(race)4     0.0  1.0  2521
## as.factor(race)5     0.0  1.0  3218
## as.factor(race)6     0.0  1.0  3919
## as.factor(race)7     0.6  1.0  1479
## as.factor(borough)2 0.0  1.0  2223
## as.factor(borough)3 0.0  1.0  2452
## as.factor(borough)4 0.0  1.0  2456
## as.factor(borough)5 0.0  1.0  2827
## poverty              0.0  1.0  4125
## extflr5_2            0.0  1.0  4177
## intcrack2            0.0  1.0  3627
## inthole2             0.0  1.0  3581
## mean_PPD             0.0  1.0  4420
## log-posterior        0.1  1.0  1720
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

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.

```
model14.3a <- rstanarm::stan_glm(switch ~ log(dist), binomial(link = "logit"),
                  wells, refresh = 0)
summary(model14.3a)
```

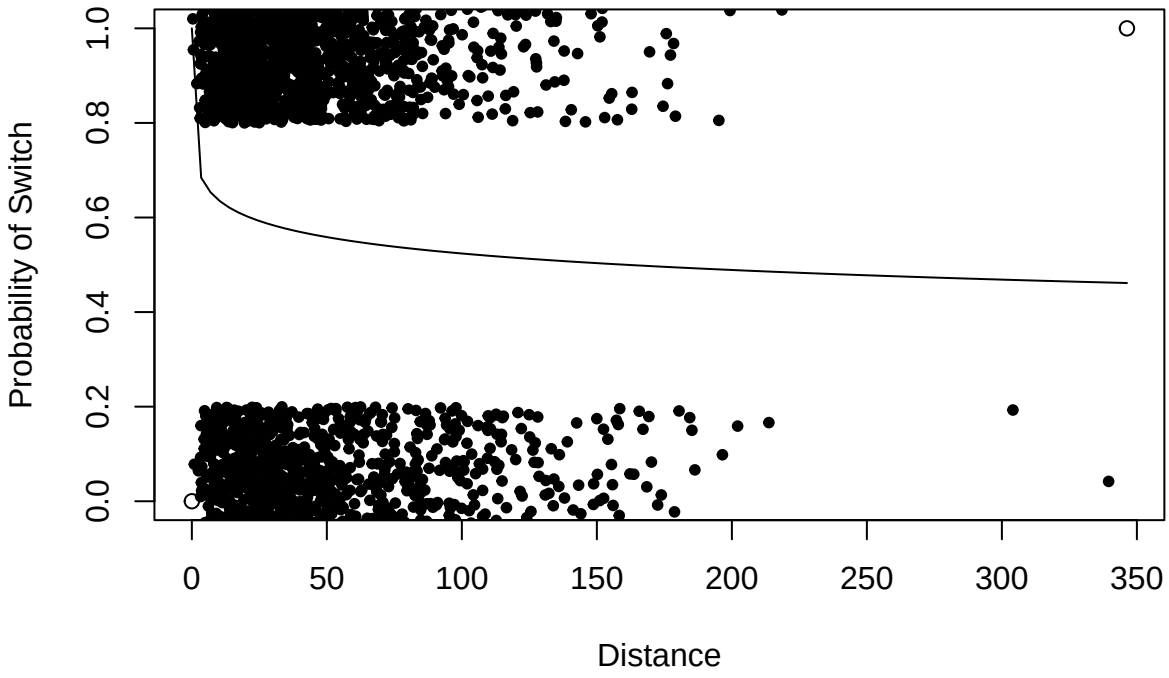
```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ log(dist)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   2
```

```
##
## 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  2872
## log(dist)     0.0  1.0  2816
## mean_PPD      0.0  1.0  3480
## log-posterior 0.0  1.0  1749
##
## 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.

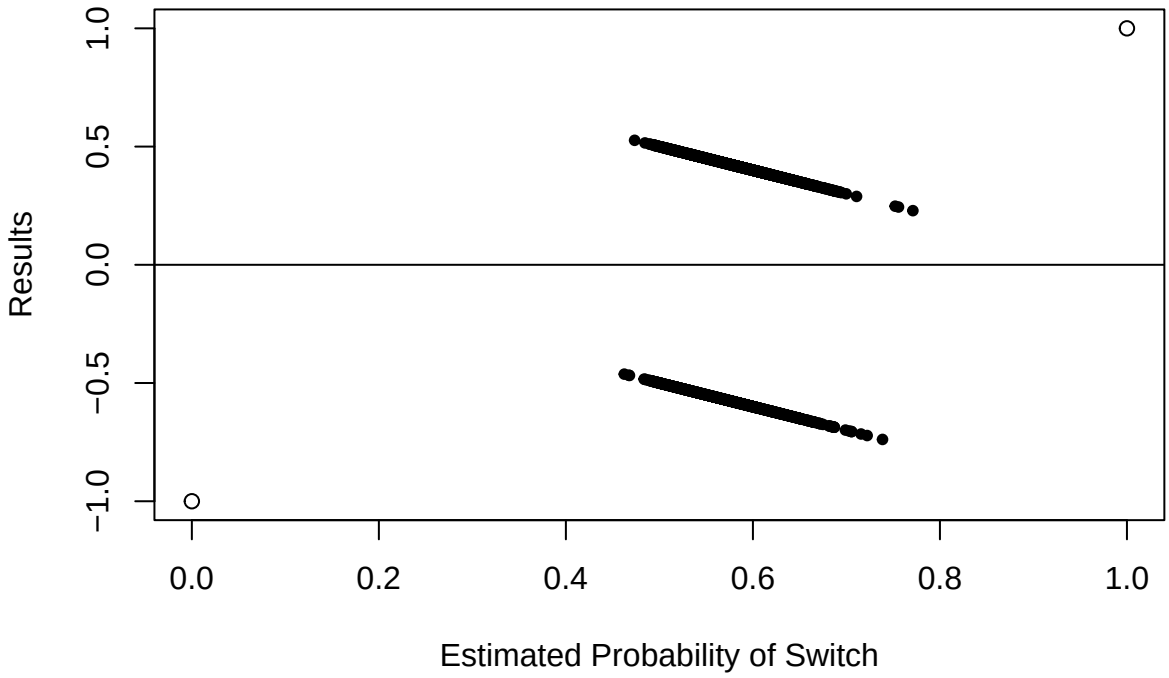
```
plot(c(0, max(wells$dist, na.rm = TRUE) * 1.02), c(0, 1),
     xlab = "Distance",
     ylab = "Probability of Switch")
points(wells$dist, jitter(wells$switch), pch = 20)
curve(invlogit(coef(model14.3a)[1] + coef(model14.3a)[2] * log(x)), add = TRUE)
```



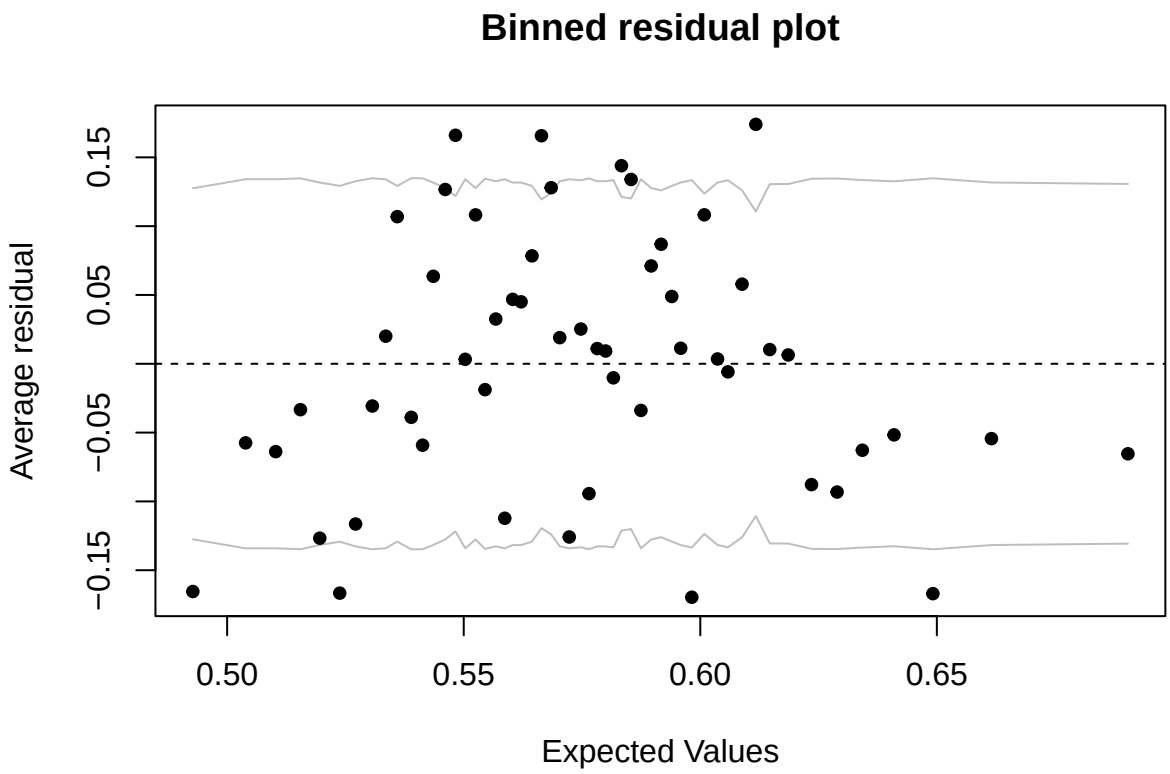
(c)

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

```
plot(c(0, 1), c(-1, 1),
     xlab = "Estimated Probability of Switch",
     ylab = "Results")
abline(0, 0)
points(fitted(model14.3a), wells$switch - fitted(model14.3a), pch = 20)
```




```
binnedplot(fitted(model14.3a), resid(model14.3a))
```



(d)

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

```
error14.3d <- mean((fitted(model14.3a) > 0.5 & wells$switch == 0) |
  (fitted(model14.3a) < 0.5 & wells$switch == 1))
cat("The error rate of the fitted model is:", error14.3d)
```

```
## The error rate of the fitted model is: 0.4188742
```

(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.

```
wells$distL <- ifelse(wells$dist < 100, "1", ifelse(wells$dist < 200, "2", "3"))
model14.3e <- stan_glm(switch ~ distL, binomial(link = "logit"),
                   wells, refresh = 0)

summary(model14.3e)
```

```
##
```

```
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ distL
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   3
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept)  0.4    0.0  0.3   0.4   0.4
## distL2      -0.7    0.1 -0.8  -0.7  -0.5
## distL3      -1.8    0.9 -3.0  -1.7  -0.8
##
## 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  3925
## distL2        0.0  1.0  3764
## distL3        0.0  1.0  3457
## mean_PPD      0.0  1.0  3692
## log-posterior 0.0  1.0  1806
##
## 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.

```
model14.7a <- rstanarm::stan_glm(switch ~ dist100 * log(arsenic), "binomial", data = wells, refresh =0)
summary(model14.7a)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist100 * 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
## dist100              -0.9    0.1 -1.0  -0.9  -0.7
## log(arsenic)          1.0    0.1  0.9   1.0   1.1
## dist100:log(arsenic) -0.2    0.2 -0.5  -0.2   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  2283
## dist100               0.0  1.0  1598
## log(arsenic)          0.0  1.0  1600
## dist100:log(arsenic) 0.0  1.0  1349
## mean_PPD              0.0  1.0  3650
## log-posterior         0.0  1.0  1673
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(b)

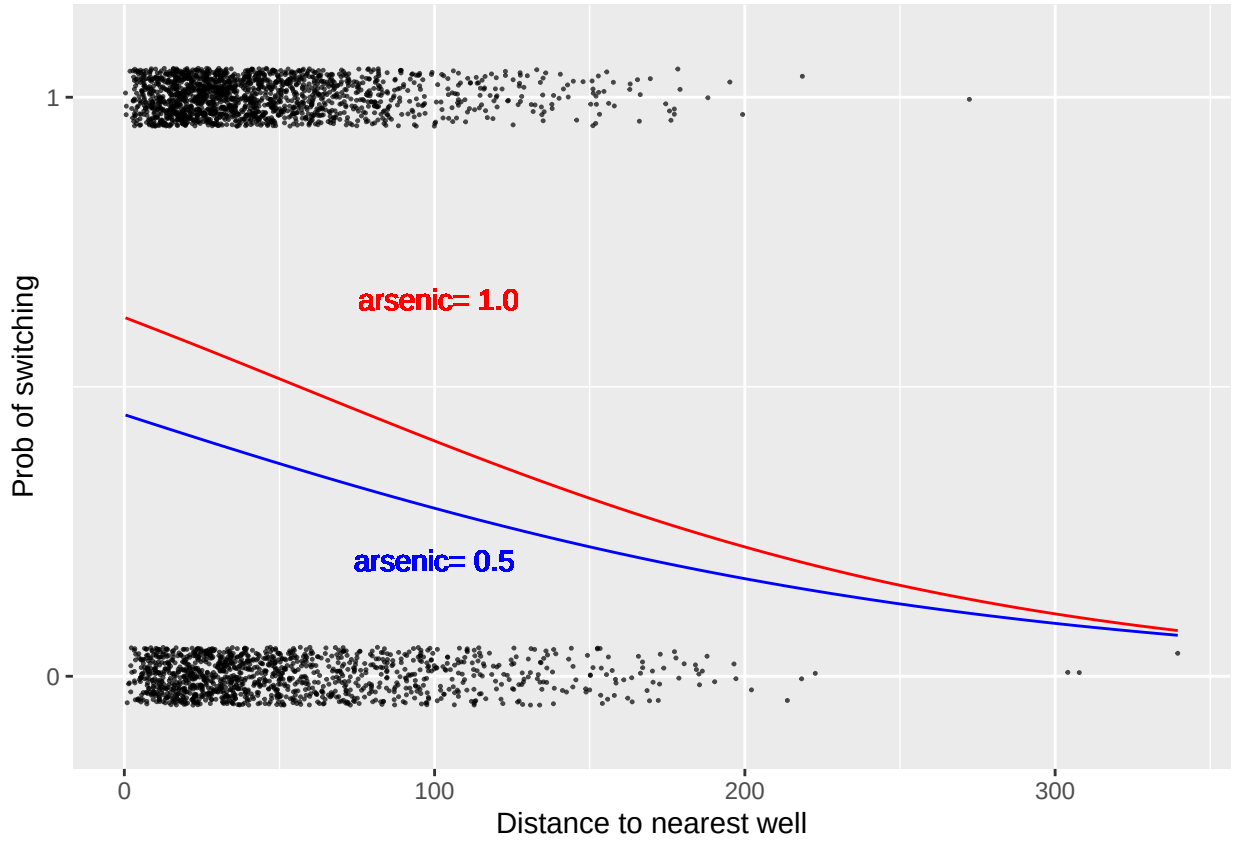
Make graphs as in Figure 14.3 to show the relation between probability of switching, distance, and arsenic level.

```
p_switch=function(dist, arsenic){ y=invlogit(coef(model14.7a)[1]+coef(model14.7a)[2]*dist/100+
coef(model14.7a)[3]*log(arsenic)+ coef(model14.7a)[4]*(dist/100)*log(arsenic))
return(y) }

ggplot(wells)+
geom_jitter(aes(x=dist, y=switch), height = 0.05, alpha=0.7, size=0.2)+ geom_function(aes(x=dist), fun=
  labs(x="Distance to nearest well", y="Prob of switching")
```

```
## Warning in geom_text(aes(x = 100, y = 0.2, label = "arsenic= 0.5"), color = "blue"): All aesthetics l
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```

```
## Warning in geom_text(aes(x = 100, y = 0.65, label = " arsenic= 1.0"), color = "red"): All aesthetics
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```



(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. -.212
ii. A comparison of `dist` = 100 to `dist` = 200, with `arsenic` held constant. -.209
iii. A comparison of `arsenic` = 0.5 to `arsenic` = 1.0, with `dist` held constant. .146
iv. A comparison of `arsenic` = 1.0 to `arsenic` = 2.0, with `dist` held constant. .140

```
difference_i=mean(p_switch(dist = 100, arsenic=wells$arsenic)-p_switch(dist = 0, arsenic=wells$arsenic)

difference_ii=mean(p_switch(dist = 200,arsenic=wells$arsenic)- p_switch(dist = 100, arsenic=wells$arseni

difference_iii=mean(p_switch(dist = wells$dist, arsenic=1)- p_switch(dist = wells$dist, arsenic=0.5))

difference_iv=mean(p_switch(dist = wells$dist, arsenic=2)- p_switch(dist = wells$dist, arsenic=1))
```