MA678 Homework 4

Paul Moon

10/10/2024

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.

```
#Looking at arsenic levels. Thus, using the divide-by-4 rule, we get B/4 = 0.46/4 = 0.115

db4r <- 0.46/4
est1 <- db4r * (1 - 0.5)
mse1 <- sqrt(0.04 ^ 2 / (4 ^ 2))
cat("The approximate estimate is:", est1)
```

```
## The approximate estimate is: 0.0575
```

```
cat("\nThe standard error is:", mse1)
```

```
##
## The standard error is: 0.01
```

```
#Find the confidence interval by getting our estimate ± standard deviation * standard error
cat("\nThe 50% interval range is: (",
    c(est1 - 0.67 * mse1, ",", est1 + 0.67 * mse1),
    ")")
```

```
##
## The 50% interval range is: ( 0.0508 , 0.0642 )
```

```
cat("\nThe 95% interval range is: (",
    c(est1 - 1.96 * mse1, ",", est1 + 1.96 * mse1),
    ")")
```

```
##
## The 95% interval range is: ( 0.0379 , 0.0771 )
```

(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.

```
wells <- read.csv("wells.csv", header = TRUE)

model1 <- stan_glm(switch ~ dist100 + arsenic, binomial(link = "logit"), data = wells, refresh = 0)
#summary(model1)

#Use the model1 from the wells data set to find the inverse logit function
newdata = data.frame(dist100 = c(0.5, 0.5), arsenic = c(0.5, 1))
ep <- invlogit(posterior_linpred(model1, newdata = newdata))

#sprintf is used to make the output more similar to part a so that comparisons are easier to see
est2 <- sprintf("%.4f", mean(ep[, 2] - ep[, 1]))
mse2 <- sprintf("%.4f", sd(ep[, 2] - ep[, 1]))
cat("The approximate estimate is:", est2)
```

```
## The approximate estimate is: 0.0575
```

```
cat("\nThe standard error is:", mse2)
```

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

```r
#collapse is used to make the output more similar to part a so that comparisons are easier to see
ci501 <- sprintf("%.4f", quantile(ep[, 2] - ep[, 1], c(0.25, 0.75)))
ci951 <- sprintf("%.4f", quantile(ep[, 2] - ep[, 1], c(0.025, 0.975)))
cat("\nThe 50% interval range is: (", paste(ci501, collapse = ", ") , ")")
```

```
##
## The 50% interval range is: ( 0.0542, 0.0609 )
```

```r
cat("\nThe 95% interval range is: (", paste(ci951, collapse = ", ") , ")")
```

```
##
## The 95% interval range is: ( 0.0477, 0.0673 )
```

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:

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

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

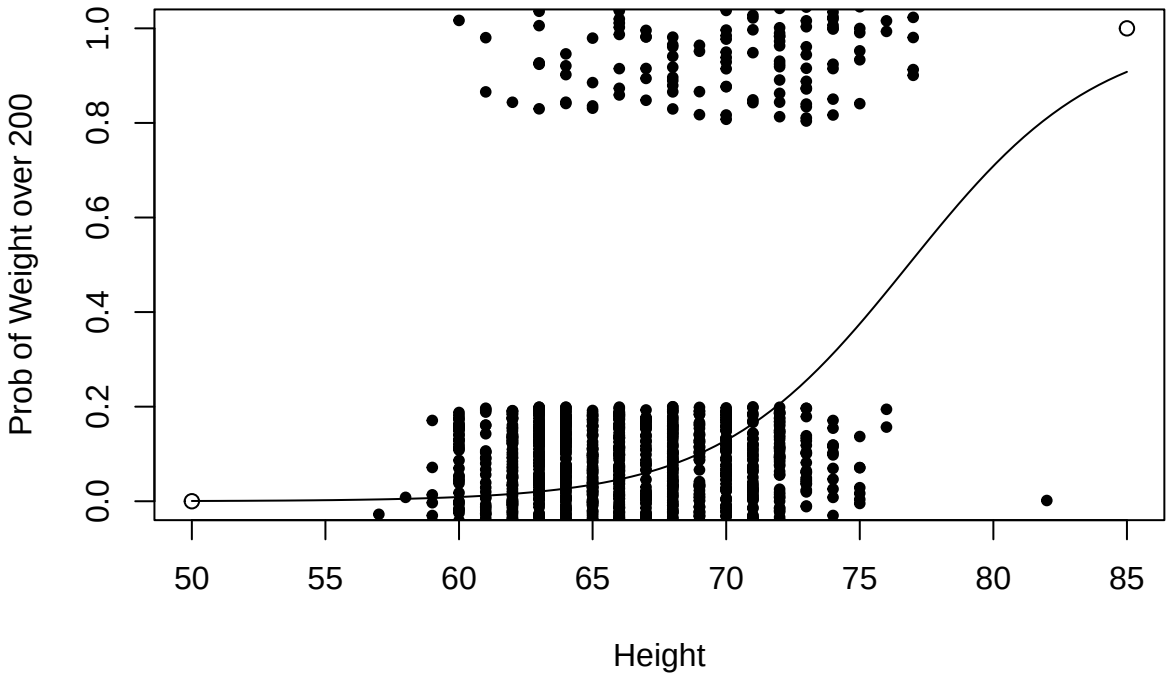
```
stan_glm(formula = heavy ~ height, family=binomial(link="logit"), data=health)
             Median MAD_SD
(Intercept) -21.51   1.60
height        0.28   0.02
```

(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
earnings <- read.csv("earnings.csv", header = TRUE)
#Created a new column with heavy that is true if the weight is over 200
earnings$heavy = ifelse(earnings$weight > 200, 1, 0)

#Plotting the jitter plot
plot(c(50, 85), c(0, 1),
     xlab = "Height",
     ylab = "Prob of Weight 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 0.28 / 4 = 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?

Since 1 inch is equivalent to 2.54 cm, we would have our new equation:

logit(probability of weight over 200) = -21.51 + 0.28 * 2.54 * height logit(probability of weight over 200) = -21.51 + 0.7112 * height

This means that we will still have the intercept of -21.51 since we do not deal with height; however, we will have a different slope of 0.7112 since that is the new slope converted from inches to centimeters.

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.

```
#Setting the response rate to the first and second half of the the experiment
df = data.frame(x = c(rep(1, 500), rep(0, 500)),
                y = c(rbinom(500, 1, 0.5),
                      rbinom(500, 1, 0.4)))

model2 = glm(y ~ x, "binomial", df)
summary(model2)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial", data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.47260    0.09195  -5.140 2.75e-07 ***
## x            0.53663    0.12831   4.182 2.89e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1376.3  on 999  degrees of freedom
## Residual deviance: 1358.6  on 998  degrees of freedom
## AIC: 1362.6
##
## Number of Fisher Scoring iterations: 4
```

(b)

Compare to the results from Exercise 4.1.

```
est3 <- 0.5 - 0.4
mse3 <- sprintf("%.4f", sqrt(0.5 ^ 2 / 500 + 0.5 ^ 2 / 500))
cat("The estimate of the average treatment effect is:", est3)
```

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

```
cat("\nThe standard error of the average treatment effect is:", mse3)
```

```
##
## 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.

```
rodents <- read.table("rodents.dat", header = TRUE)
#Table is unreadable so had to change it to a readable table
rodents <- data.table(rodents)

#Setting the race to it's number as written in the data set
invisible(rodents[, asian := race == 5])
invisible(rodents[, black := race == 2])
invisible(rodents[, hispanic := race == 3 | race == 4])

model3 <- glm(rodent2 ~ asian + black + hispanic, "binomial", rodents)
summary(model3)
```

```
##
## Call:
## glm(formula = rodent2 ~ asian + black + hispanic, family = "binomial",
##     data = rodents)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.1904     0.1308 -16.749  < 2e-16 ***
## asianTRUE      0.8130     0.2486   3.270  0.00108 **
## blackTRUE      1.3759     0.1695   8.116 4.82e-16 ***
## hispanicTRUE   1.8558     0.1687  10.999  < 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: 1699.6  on 1550  degrees of freedom
## Residual deviance: 1551.5  on 1547  degrees of freedom
##   (197 observations deleted due to missingness)
## AIC: 1559.5
##
## Number of Fisher Scoring iterations: 4
```

(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.

```
#Added basically just all of the residuals in order from the data table to see what will happen to the
model4 <- glm(rodent2 ~ asian + black + hispanic + factor(borough) +
                poverty + extwin4_2 + extflr5_2 + intcrack2 + inthole2,
              "binomial", rodents)
summary(model4)
```

```
##
## Call:
## glm(formula = rodent2 ~ asian + black + hispanic + factor(borough) +
##     poverty + extwin4_2 + extflr5_2 + intcrack2 + inthole2, family = "binomial",
##     data = rodents)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.119485   0.221561  -9.566  < 2e-16 ***
## asianTRUE         0.944805   0.279653   3.378 0.000729 ***
## blackTRUE         1.203810   0.194089   6.202 5.56e-10 ***
## hispanicTRUE      1.722880   0.194226   8.870  < 2e-16 ***
## factor(borough)2 -0.002041   0.202321  -0.010 0.991951
## factor(borough)3 -0.231990   0.212611  -1.091 0.275207
## factor(borough)4 -0.863145   0.228258  -3.781 0.000156 ***
## factor(borough)5 -1.932083   0.744782  -2.594 0.009482 **
## poverty          -0.014708   0.166778  -0.088 0.929724
## extwin4_2         0.747339   0.397370   1.881 0.060011 .
## extflr5_2         0.780369   0.331522   2.354 0.018578 *
## intcrack2         0.934231   0.213352   4.379 1.19e-05 ***
## inthole2          1.268860   0.295823   4.289 1.79e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1553.4  on 1418  degrees of freedom
## Residual deviance: 1269.0  on 1406  degrees of freedom
##   (329 observations deleted due to missingness)
## AIC: 1295
##
## Number of Fisher Scoring iterations: 6
```

```
cat("Even with the different residuals added into the regression, we can still see that there is a clea
```

```
## Even with the different residuals added into the regression, we can still see that there is a clear
```

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.

```
model5 <- stan_glm(switch ~ log(dist), binomial(link = "logit"),
                   wells, refresh = 0)
summary(model5)
```

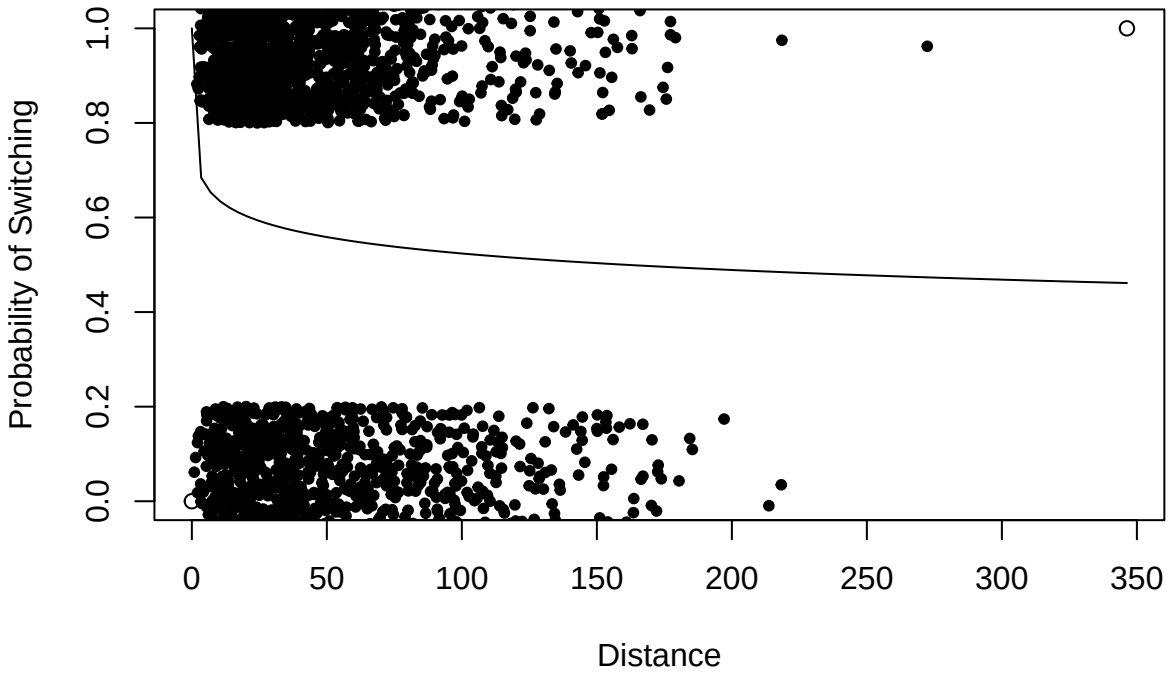
```
##
## Model Info:
```

```
##  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  2503
## log(dist)     0.0  1.0  2545
## mean_PPD      0.0  1.0  2802
## log-posterior 0.0  1.0  1649
##
## 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.

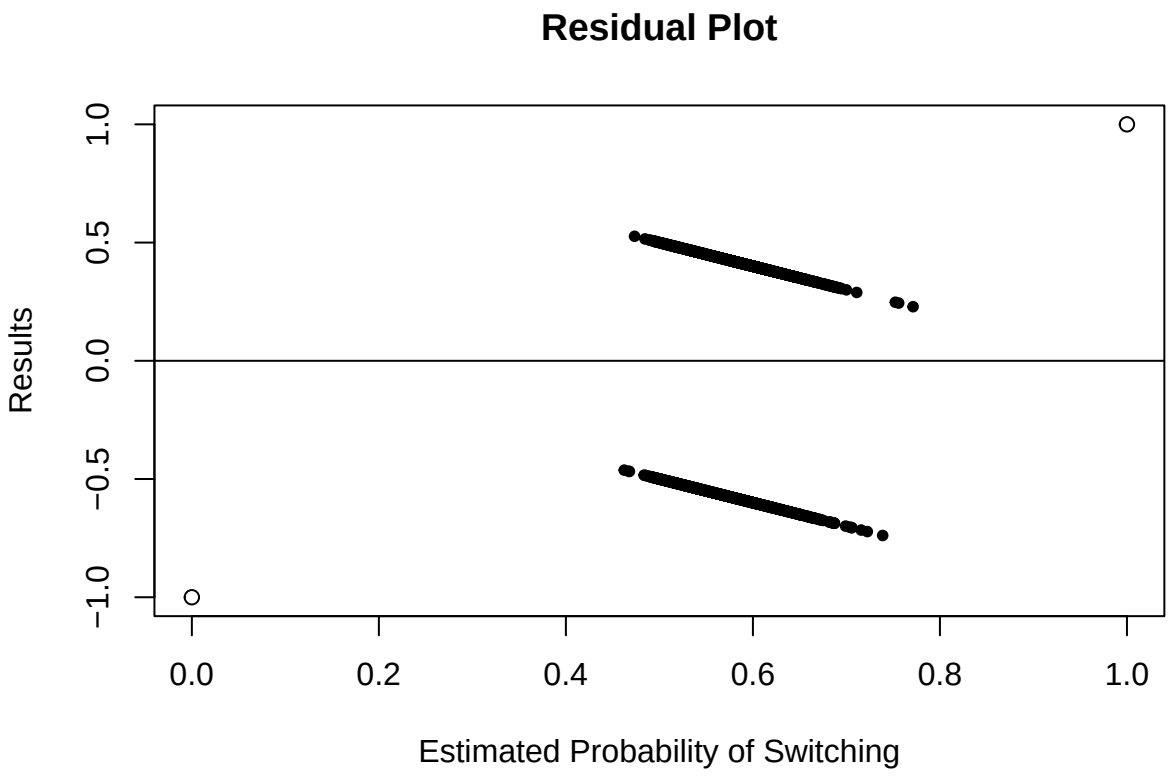
```
plot(c(0, max(wells$dist, na.rm = TRUE) * 1.02), c(0, 1),
     xlab = "Distance",
     ylab = "Probability of Switching")
points(wells$dist, jitter(wells$switch), pch = 20)
curve(invlogit(coef(model5)[1] + coef(model5)[2] * log(x)), add = TRUE)
```



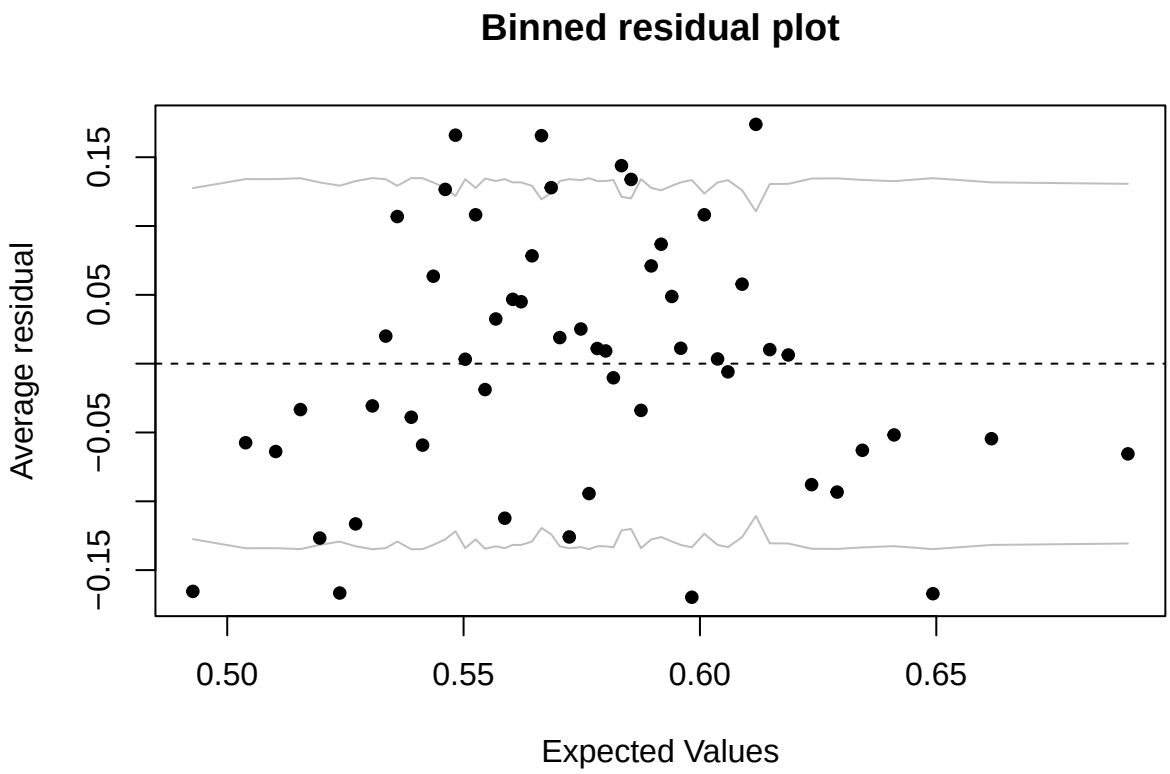
(c)

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

```
#Plotting both the residual plot and the binned plot but with a abline at y = 0
plot(c(0, 1), c(-1, 1),
     xlab = "Estimated Probability of Switching",
     ylab = "Results",
     main = "Residual Plot")
abline(0, 0)
points(fitted(model5), wells$switch - fitted(model5), pch = 20)
```




```
binnedplot(fitted(model5), resid(model5))
```



(d)

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

```
errorR <- mean((fitted(model5) > 0.5 & wells$switch == 0) |
  (fitted(model5) < 0.5 & wells$switch == 1))
cat("The error rate of the fitted model compared to the error rate of the null model is:", errorR)
```

```
## The error rate of the fitted model compared to the error rate of the null 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.

```
#Setting a new distL what takes in distances below 100 and above 200
wells$distL <- ifelse(wells$dist < 100, "1", ifelse(wells$dist < 200, "2", "3"))
model6 <- stan_glm(switch ~ distL, binomial(link = "logit"),
                   wells, refresh = 0)
summary(model6)
```

```
##
```

```
## 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  -1.8  -0.7
##
## 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  3729
## distL2        0.0  1.0  4082
## distL3        0.0  1.0  2825
## mean_PPD      0.0  1.0  3783
## 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
```

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.

```
model7 <- glm(switch ~ dist100 * log(arsenic), "binomial", wells)
summary(model7)
```

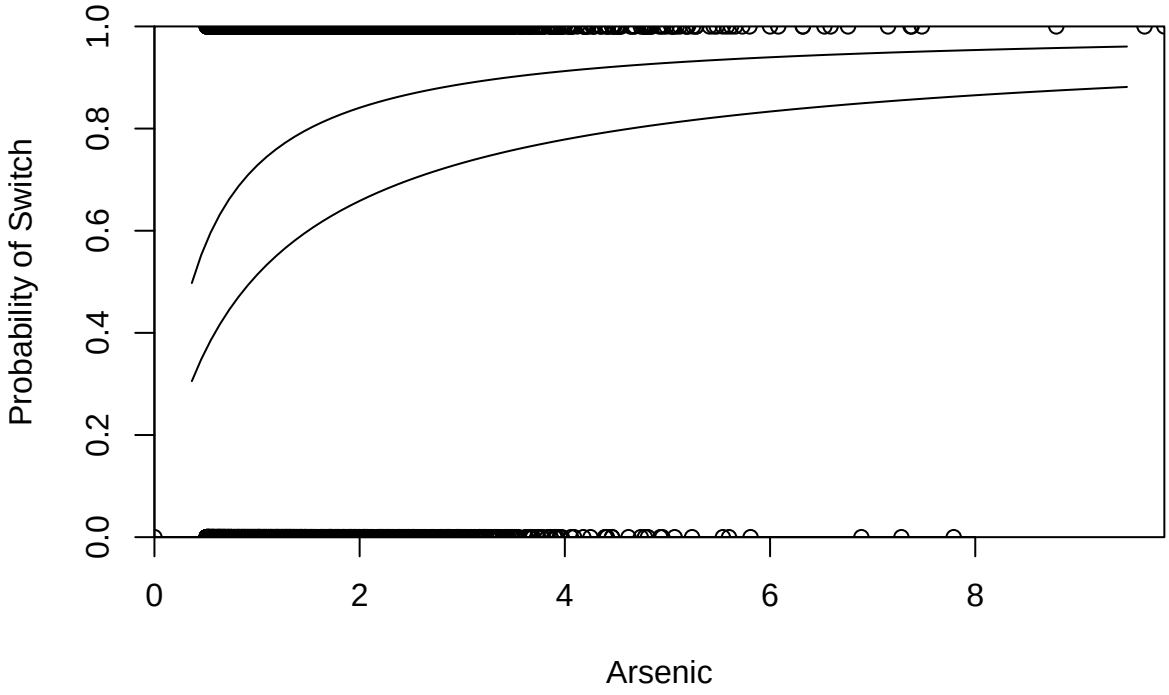
```
##
## Call:
## glm(formula = switch ~ dist100 * log(arsenic), family = "binomial",
##     data = wells)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.49135    0.06812   7.213 5.47e-13 ***
```

```
## dist100              -0.87350    0.13418  -6.510 7.52e-11 ***
## log(arsenic)          0.98341    0.10969   8.965  < 2e-16 ***
## dist100:log(arsenic) -0.23091    0.18261  -1.264    0.206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3896.8  on 3016  degrees of freedom
## AIC: 3904.8
##
## Number of Fisher Scoring iterations: 4
```

(b)

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

```
plot(c(0, max(wells$arsenic, na.rm = TRUE) * 1.02), c(0, 1),
     xlab = "Arsenic",
     ylab = "Probability of Switch",
     xaxs = "i", yaxs = "i")
points(wells$arsenic, wells$switch)
curve(invlogit(coef(model7)[1] + coef(model7)[2] * 0 +
        coef(model7)[1] + coef(model7)[3] * log(x) + coef(model7)[4] * 0), add = TRUE)
curve(invlogit(coef(model7)[1] + coef(model7)[2] * .5 +
coef(model7)[3] * log(x) + coef(model7)[4] * 0.5 * log(x)), add=TRUE)
```



(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.

```
sfn <- function(dist, arsenic){
  y = invlogit(coef(model7)[1] + coef(model7)[2] * dist / 100 +
                 coef(model7)[3] * log(arsenic) + coef(model7)[4] * (dist / 100) *
                 log(arsenic))
  return(y)
}

apd1<-mean(sfn(100,wells$arsenic) - sfn(0, wells$arsenic))
apd2<-mean(sfn(200,wells$arsenic) - sfn(100, wells$arsenic))
```

```r
apd3<-mean(sfn(wells$dist, 1) - sfn(wells$dist, 0.5))
apd4<-mean(sfn(wells$dist, 2) - sfn(wells$dist, 1))

paste(apd1, apd2, apd3, apd4)
```

```
## [1] "-0.211335584202869 -0.209020693854575 0.146017396315659 0.140434359006609"
```