MA678 Homework 4

Yuxi Wang

Disclaimer

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 "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.

```
wells <- read.csv(file="/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/Arsenic/data/wells.csv")
fit_1 <- stan_glm(formula = switch ~ dist100 + arsenic, family=binomial(link="logit"),  data=wells, ref
summary(fit_1)
```

```
##
## 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  3812
## dist100       0.0  1.0  3174
## arsenic       0.0  1.0  3624
## mean_PPD      0.0  1.0  3856
## log-posterior 0.0  1.0  1855
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

In this situation, we can consider the distance is a constant since these two person live the same distance from the nearest well. So, we can write the fomula like: Pr(switch wells)= invlogit(-0.90+0.46(arsenic)). We can devide 0.46 by 4 to get 0.115. A difference in arsenic corresponds to no more than an 11.5% positive difference in the probablity of switch wells.

In this question we can use the fomula to calculate the result in: invlogit(-0.9+0.46(1))-invlogit(-0.9+0.46(0.5))= 5.75%

At the same time, the standard error of the coefficient is 0.04.

The 50% interval of the coefficient is [0.42,0.50].

The 95% interval of the coefficient is [0.38,0.54].

(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
new1 <- data.frame(dist100=0.5, arsenic=0.5)
new2 <- data.frame(dist100=0.5, arsenic=1)
pred1 <- predict(fit_1, type="response", newdata=new1)
```

```
pred2 <- predict(fit_1, type="response", newdata=new2)
print(paste("The result of the first persion is:", pred1))
```

```
## [1] "The result of the first persion is: 0.446612050634175"
```

```
print(paste("The result of the second persion is:", pred2))
```

```
## [1] "The result of the second persion is: 0.503918598663605"
```

```
diff <- round(pred2-pred1,4)
print(paste("The more probability of second person’s switching than the first person is",diff))
```

```
## [1] "The more probability of second person’s switching than the first person is 0.0573"
```

```
# Compute 50% interval
new <- data.frame(dist100=rep(0.5,2),arsenic=c(0.5,1))
epred <- posterior_epred(fit_1, newdata=new)
head(epred)
```

```
##
## iterations         1         2
##       [1,] 0.4510064 0.5048693
##       [2,] 0.4278522 0.4916778
##       [3,] 0.4508073 0.5069196
##       [4,] 0.4473376 0.5026953
##       [5,] 0.4547908 0.5067758
##       [6,] 0.4441292 0.5013187
```

```
quantile(epred[,2] - epred[,1], c(0.25, 0.75))
```

```
##        25%        75%
## 0.05387123 0.06071001
```

```
# Compute 95% interval
quantile(epred[,2] - epred[,1], c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.04773242 0.06716337
```

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.

```
# Just graph the curve
curve(invlogit(-21.51+ 0.28*x), xlim = c(0,150))
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-3-1.pdf # When we put -21.51+0.28*x=0-> height=76.8, then the line goes through the 50% probability point.

(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? ## Firstly, we know that 1 inch= 2.54 cm. So, the intercept is the same as -21.51, and the slop will be divided by 2.54 that is 0.11.

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.

```
n <- 1000 #1000 people are contacted
treat <- c(rep(1,500),rep(0,500))
response <- c(rep(1,250),rep(0,550),rep(1,200))
dt <- data.frame(treat,response)
fit_10 <- stan_glm(response~treat,family= binomial(link = "logit"), data=dt,refresh=0)
summary(fit_10)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      response ~ treat
##  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
## treat        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  2252
## treat         0.0  1.0  2605
## mean_PPD      0.0  1.0  2751
## log-posterior 0.0  1.0  1778
##
## 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.

In 13.10, a difference of 1 in treatment corresponds to a positive difference of 0.4 in the logit probability of responding to the survey. And standard deviation of treatment in logit is 0.1.

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(file="/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/Rodents/rodents.dat")
# Given indicators for the ethnic groups (race).
rodents$race <- factor(rodents$race, labels=c("White (non-hispanic)",
                                              "Black (non-hispanic)",
                                              "Puerto Rican",
                                              "Other Hispanic",
                                              "Asian/Pacific Islander",
                                              "Amer-Indian/Native Alaskan",
                                              "Two or more races"))
fit_11 <- stan_glm(rodent2~ race+intcrack2+inthole2, family = binomial(link = "logit"),data = rodents, 
summary(fit_11)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ race + intcrack2 + inthole2
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1505
##  predictors:   9
##
## Estimates:
##                                  mean   sd    10%   50%   90%
## (Intercept)                     -2.4    0.1  -2.6  -2.4  -2.2
## raceBlack (non-hispanic)         1.3    0.2   1.0   1.3   1.5
## racePuerto Rican                 1.5    0.2   1.2   1.5   1.8
## raceOther Hispanic               1.9    0.2   1.7   1.9   2.2
## raceAsian/Pacific Islander       0.8    0.3   0.5   0.8   1.1
## raceAmer-Indian/Native Alaskan  -0.4    1.5  -2.4  -0.3   1.4
## raceTwo or more races          -32.4   24.4 -66.9 -27.0  -5.7
## intcrack2                        1.2    0.2   1.0   1.2   1.5
## inthole2                         1.4    0.3   1.0   1.4   1.8
##
## 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  1838
## raceBlack (non-hispanic)       0.0  1.0  2275
## racePuerto Rican               0.0  1.0  2426
## raceOther Hispanic             0.0  1.0  2141
## raceAsian/Pacific Islander     0.0  1.0  2154
## raceAmer-Indian/Native Alaskan 0.0  1.0  2627
## raceTwo or more races          0.7  1.0  1227
## intcrack2                      0.0  1.0  2838
## inthole2                       0.0  1.0  3359
## mean_PPD                       0.0  1.0  4769
## log-posterior                  0.1  1.0  1609
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Discuss:

Intercept: an apartment where white (non-hispanic) live, has probability logitˆ1(2.4)=0.0831. So, has a 8.31 of the presence.

race: In this model, I choose White (non-hispanic) as base level. The coefficient of raceBlack (non-hispanic) is 1.3, so at the same situation (interior cracks and holes in this model), the raceBlack (non-hispanic) will have 1.3/4=0.325=32.5% more probability of presence than white (non-hispanic); the coefficient of Amer-Indian/Native Alaskan is -0.2, so at the same situation (interior cracks and holes in this model), the Amer-Indian/Native Alaskan will have 0.2/4=0.05=5% less probability of presence than white (non-hispanic).

intcrack2: The coefficient of intcrack2 is 1.2, which means that the situation that unit has interior cracks in walls corresponds a positive difference of 1.2 in the logit probability of the presence of rodents.

inthole2: The coefficient of inthole2 is 1.4, which means that the situation that unit has interior holes in floors corresponds a positive difference of 1.4 in the logit probability of the presence of rodents.

(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.

```
fit_11b <- stan_glm(rodent2~race + intleak2 + intcrack2 +inthole2 + old + dilap, family = binomial(link
summary(fit_11b)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ race + intleak2 + intcrack2 + inthole2 + old + dilap
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1483
##  predictors:   12
##
## Estimates:
##                                   mean   sd   10%   50%   90%
## (Intercept)                      -3.0    0.2 -3.2  -3.0  -2.7
## raceBlack (non-hispanic)         1.3    0.2   1.1   1.3   1.5
## racePuerto Rican                 1.5    0.2   1.2   1.5   1.8
## raceOther Hispanic               1.8    0.2   1.6   1.8   2.1
## raceAsian/Pacific Islander       0.9    0.3   0.6   0.9   1.3
## raceAmer-Indian/Native Alaskan  -0.3    1.4  -2.1  -0.2   1.3
## raceTwo or more races          -31.2   23.3 -63.6 -26.1  -5.9
## intleak2                         0.7    0.2   0.5   0.7   0.9
## intcrack2                        1.0    0.2   0.7   1.0   1.3
## inthole2                         1.3    0.3   0.9   1.3   1.7
## old                              0.6    0.2   0.4   0.6   0.8
## dilap                           -0.3    0.4  -0.9  -0.3   0.2
##
## 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  2169
## raceBlack (non-hispanic)       0.0  1.0  2159
## racePuerto Rican               0.0  1.0  2100
## raceOther Hispanic             0.0  1.0  2161
## raceAsian/Pacific Islander     0.0  1.0  3032
## raceAmer-Indian/Native Alaskan 0.0  1.0  2146
## raceTwo or more races          0.6  1.0  1437
## intleak2                       0.0  1.0  2718
## intcrack2                      0.0  1.0  2752
## inthole2                       0.0  1.0  2841
## old                            0.0  1.0  3791
## dilap                          0.0  1.0  3401
## mean_PPD                       0.0  1.0  4723
## log-posterior                  0.1  1.0  1569
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Discuss:

The coefficient of intleak2 is 0.7, which means that the logarithmic probability of the presence of rodents corresponding to the positive difference of the logarithmic probability of the existence of rodents is 0.7.

The coefficient of intcrack2 is 1, which means that the case of internal cracks in the unit corresponds to the positive difference of the log probability of the rodent being 1.

The coefficient of inthole2 is 1.3,which means that the case where the unit has internal holes on the floor corresponds to the positive difference of the log probability of the rodent being 1.3.

The coefficient of old is 0.6, which means that for buildings built before 1947, the positive difference of the rodent log probability of buildings built before 1947 is 0.6.

The coefficient of dilap is -0.3, which means that the situation of water seepage or deterioration in the building corresponds to a negative difference of 0.3 in the log probability of the existence of rodents.

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(file="/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/Arsenic/data/wells.csv")
fit_3 <- stan_glm(formula = switch ~log(dist), family=binomial(link="logit"),  data=wells, refresh=0)
summary(fit_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  2387
## log(dist)      0.0  1.0  2448
## mean_PPD       0.0  1.0  3305
## log-posterior 0.0  1.0  1582
##
## 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.

```
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(log(wells$dist),wells$switch_jitter,xlab = "log(distance to nearest safe well)",ylab = "Pr (switch
curve(invlogit(coef(fit_3)[1]+coef(fit_3)[2]*x),add=T,col="black")
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-8-1.pdf

(c)

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

```
plot(predict(fit_3),residuals(fit_3),main ="Residual plot",
     xlab="Expected Values", ylab="Residuals",pch=20)
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-9-1.pdf

```
binnedplot(predict(fit_3),residuals(fit_3),pch=20)
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-9-2.pdf

(d)

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

```
##error rate
pred <- predict(fit_3)
error_rate_fitted <- mean((pred>0.5&wells$switch==0)|(pred<0.5&wells$switch==1))
error_rate_fitted <- round(error_rate_fitted,3)
print(paste("The error rate of the fitted model is",error_rate_fitted ))
```

```
## [1] "The error rate of the fitted model is 0.559"
```

```
##null model
null <- rep(0,length(wells$switch))
error_rate_null <- round(mean(pred<0.5&wells$switch==1),3)
print(paste("The error rate of the null model is",error_rate_null ))
```

```
## [1] "The error rate of the null model is 0.506"
```

(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$dist_lower_than_100 <- as.numeric(wells$dist<100)
wells$dist_between_100_and_200 <-
  as.numeric(100<=wells$dist&wells$switch<=200)
wells$dist_greater_than_200 <-
  as.numeric(wells$dist>200)
fit_3e <- stan_glm(switch~dist_lower_than_100+dist_between_100_and_200+
                   dist_greater_than_200, data=wells,family = binomial(link = "logit"),
                 refresh=0)
summary(fit_3e)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist_lower_than_100 + dist_between_100_and_200 + dist_greater_than_200
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 3020
##  predictors:   4
##
## Estimates:
##                            mean   sd   10%   50%   90%
## (Intercept)               0.2    5.8 -7.2   0.2   7.7
## dist_lower_than_100       0.2    5.8 -7.3   0.2   7.6
## dist_between_100_and_200 -0.5    5.8 -7.9  -0.5   7.0
## dist_greater_than_200    -1.2    0.9 -2.4  -1.1  -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)              0.2  1.0  1266
## dist_lower_than_100      0.2  1.0  1266
## dist_between_100_and_200 0.2  1.0  1264
## dist_greater_than_200    0.0  1.0  1981
## mean_PPD                 0.0  1.0  3047
## log-posterior            0.0  1.0  1214
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

#14.5 Working with logistic regression: In a class of 50 students, a logistic regression is performed of course grade (pass or fail) on midterm exam score (continuous values with mean 60 and standard deviation 15). The fitted model is Pr(pass) = logit-1(-24 + 0.4x).

(a)

Graph the fitted model. Also on this graph put a scatterplot of hypothetical data consistent with the information given.

```r
# Graph the fitted model, which is is Pr(pass) = logit-1(-24 + 0.4x).
curve(invlogit(-24+0.4*x),xlim=c(0, 120))
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-12-1.pdf

(b)

Suppose the midterm scores were transformed to have a mean of 0 and standard deviation of 1. What would be the equation of the logistic regression using these transformed scores as a redictor?

```r
# After transforming to have a mean of 0 and standard deviation of 1
curve(invlogit(-24*0+0.4*15*x),xlim=c(-2, 2))
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-13-1.pdf

(c)

Create a new predictor that is pure noise; for example, in R you can create newpred <- rnorm(n,0,1). Add it to your model. How much does the leave-one-out cross validation score decrease?

I think pure noise will not affect the bias, so the leave-one-out cross-validation score will not decrease.

#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(file="/Users/mac/Desktop/BU Mssp/MA678/ROS-Examples-master/Arsenic/data/wells.csv")
wells$log_arsenic <- log(wells$arsenic)
fit_7 <- stan_glm(switch~dist+log_arsenic+dist*log_arsenic,
                  family = binomial(link = "logit"), data=wells, refresh=0)
head(fit_7$coefficients)
```

```
##      (Intercept)             dist      log_arsenic dist:log_arsenic
##      0.494227360     -0.008785491      0.981740292     -0.002261611
```

For intercept: One who lives with a clean water in an average distance has invlogit(0.49)=62.1% probability to switch well.

For dist: The other situations are same, if one meter increases in distance from the well with safe water, it will decrease 0.009/4=0.225% of the probability of switching.

For log_arsenic:The other situations are same, if arsenic increase 1%, the difference in the probability of switching well will be 0.987*log(1.01/1)=0.9821% # For dist:log_arsenic: the coefficient of it is close to zero, so it is not vital in this regression model.

(b)

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

```
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) #switch~dist
plot(wells$dist,wells$switch_jitter,
     xlab = "Distance (in meters) to nearest safe well",
     ylab = "Pr (switching)",
     pch=20,
     col="blue")
curve(invlogit(coef(fit_7)[1] +coef(fit_7)[3]*log(1)+ coef(fit_7)[2]*x),xlim=c(0, 350), col="purple",add
curve(invlogit(coef(fit_7)[1] +coef(fit_7)[3]*log(0.5) +coef(fit_7)[2]*x),xlim=c(0, 350),
      col="purple" ,add=TRUE)
text (35, .23, "if Arsenic = 0.5", adj=0, cex=.9)
text (75, .50, "if Arsenic = 1.0", adj=0, cex=.9)
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-15-1.pdf

```
#switch~log_arsenic
plot(wells$log_arsenic,wells$switch_jitter, xlab = "Arsenic in log", ylab = "Pr (switching)",col="orange
curve(invlogit(coef(fit_7)[1]+coef(fit_7)[3]*x+coef(fit_7)[2]*50),col="red",add=T)
curve(invlogit(coef(fit_7)[1]+coef(fit_7)[3]*x+coef(fit_7)[2]*100),col="red",add=T)
text (-0.25, .6, "if distance = 50", adj=0, cex=.9)
text (0, .35, "if distance = 100", adj=0, cex=.9)
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-15-2.pdf

(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.

```
# For i
lower_dist <- 0
upper_dist <- 100
b_hat <- coef(fit_7)
diff <- invlogit(b_hat[1]+b_hat[2]*upper_dist+b_hat[3]*wells$log_arsenic+
                   b_hat[4]*wells$log_arsenic*upper_dist)-
  invlogit(b_hat[1]+b_hat[2]*lower_dist+b_hat[3]*wells$log_arsenic+
             b_hat[4]*wells$log_arsenic*lower_dist)
print(paste("the average predictive differences of 'i' is:",mean(diff)))
```

```
## [1] "the average predictive differences of 'i' is: -0.212136167371315"
```

```
# Interpret: All the situations are same, one who lives 100 meters from the nearest safe well will have

# For ii
lower_dist <- 100
```

```
upper_dist <- 200
diff <- invlogit(b_hat[1]+b_hat[2]*upper_dist+b_hat[3]*wells$log_arsenic+
                   b_hat[4]*wells$log_arsenic*upper_dist)-
  invlogit(b_hat[1]+b_hat[2]*lower_dist+b_hat[3]*wells$log_arsenic+
             b_hat[4]*wells$log_arsenic*lower_dist)
print(paste("the average predictive differences of 'ii' is:",mean(diff)))
```

```
## [1] "the average predictive differences of 'ii' is: -0.209496706687183"
```

```
# Interpret: The result is very close to the first(i)’ result , so it can be infered that every more 10

# For iii
lower_arsenic <- 0.5
upper_arsenic <- 1
diff <- invlogit(b_hat[1]+b_hat[2]*wells$dist+b_hat[3]*lower_arsenic+
                   b_hat[4]*wells$dist*lower_arsenic)-
  invlogit(b_hat[1]+b_hat[2]*wells$dist+b_hat[3]*upper_arsenic+
             b_hat[4]*wells$dist*upper_arsenic)
print(paste("the average predictive differences of 'iii' is:", mean(diff)))
```

```
## [1] "the average predictive differences of 'iii' is: -0.0919957483583698"
```

```
# Interpret: All the situations are same, if one’s arsenic level of well is 1 and the other’s is 0.5, h
# For iv
lower_arsenic <- 1
upper_arsenic <- 2
diff <- invlogit(b_hat[1]+b_hat[2]*wells$dist+b_hat[3]*lower_arsenic+
                   b_hat[4]*wells$dist*lower_arsenic)-
  invlogit(b_hat[1]+b_hat[2]*wells$dist+b_hat[3]*upper_arsenic+
             b_hat[4]*wells$dist*upper_arsenic)
print(paste("the average predictive differences of 'iv' is:", mean(diff)))
```

```
## [1] "the average predictive differences of 'iv' is: -0.135411698152685"
```

```
# Interpret: All the situations are same, if one’s arsenic level of well is 1 and the other’s is 2, he/
```

14.9

Linear or logistic regression for discrete data: Simulate continuous data from the regression model, z = a + bx + error. Set the parameters so that the outcomes z are positive about half the time and negative about half the time.

(a)

Create a binary variable y that equals 1 if z is positive or 0 if z is negative. Fit a logistic regression predicting y from x.

```
set.seed(1213)
x <- c(-50:-1,1:50)
a <- 0
b <- 1
error <- rnorm(100,0,1)
z <- a+b*x+error
y <- rep(NA,100)
for(i in 1:100){
```

```
ifelse(z[i]>0,y[i] <-1,y[i] <- 0 ) }
data <- data.frame(x,y,z)
fit_149 <- stan_glm(y~x, family = binomial(link = "logit"),data=data,refresh=0)
summary(fit_149)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 100
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -0.2    0.5 -0.9  -0.2   0.4
## x            0.2    0.0  0.2   0.2   0.3
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.5    0.0  0.5   0.5   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  2566
## x             0.0  1.0  2284
## mean_PPD      0.0  1.0  3460
## log-posterior 0.0  1.0  1512
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(b)

Fit a linear regression predicting y from x: you can do this, even though the data y are discrete.

```
sims <- as.matrix(fit_149)
n_sims <- nrow(sims)
plot(x,y,pch=20,col="purple")
for(s in 1:20){
curve(invlogit(sims[s,1]+sims[s,2]*x), add = T, col="blue",lwd=0.5)
}
curve(invlogit(mean(sims[,1])+mean(sims[,2])*x),add = T)
```

HW4-Yuxi-Wang_files/figure-latex/unnamed-chunk-18-1.pdf

```r
pred_y <- mean(sims[,1])+mean(sims[,2])*x
dt1 <- data.frame(x,pred_y)
fit1 <- stan_glm(pred_y~x, data = dt1,refresh=0)
```

```
## Warning: There were 429 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 510 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. S
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 2.16, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Markov chains did not converge! Do not analyze results!
```

```r
summary(fit1)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      pred_y ~ x
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 100
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -0.2    0.0 -0.2  -0.2  -0.2
## x            0.2    0.0  0.2   0.2   0.2
## sigma        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  -0.2
##
## 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 1852
## x              0.0  1.0 4105
## sigma          0.0  2.3    3
## mean_PPD       0.0  1.0 2774
## log-posterior 22.1  2.5    3
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

(c)

Estimate the average predictive comparison—the expected difference in y, corresponding to a unit difference in x—based on the fitted logistic regression in (a). Compare this average predictive comparison to the linear regression coefficient in (b).

```
b <- coef(fit_149)
b1 <- coef(fit1)
hi <- 2
lo <- 1
delta <- invlogit(b[1]+b[2]*hi)-invlogit(b[1]+b[2]*lo)
delta1 <- invlogit(b1[1]+b[2]*hi)-invlogit(b1[1]+b1[2]*lo)
data.frame(b[2],b1[2],delta,delta1)
```

```
##        b.2.     b1.2.      delta     delta1
## x 0.2388027 0.2424293 0.05940841 0.05852618
```

14.10

Linear or logistic regression for discrete data: In the setup of the previous exercise:

(a)

Set the parameters of your simulation so that the coefficient estimate in (b) and the average predictive comparison in (c) are close.

After setting the parameters above, which a=0 and b=1, these parameters maintain the requirement that the coefficient estimate in (b) and the average predictive comparison in (c) are close.

(b)

Set the parameters of your simulation so that the coefficient estimate in (b) and the average predictive comparison in (c) are much different.

```
set.seed(1213)
x <- c(-50:-1,1:50)
a <- -1
b <- -8
error <- runif(100,0,100)
z <- a+b*x+error
y <- rep(NA,100)
for(i in 1:100){
  ifelse(z[i]>50,y[i] <-1,y[i] <- 0 ) }
dt <- data.frame(x,y,z)
fit <- stan_glm(y~x, family = binomial(link = "logit"), data=dt,refresh=0)
sims <- as.matrix(fit)
```

```r
n_sims <- nrow(sims)
pred_y <- mean(sims[,1])+mean(sims[,2])*x
dt1 <- data.frame(x,pred_y)
fit1 <- stan_glm(pred_y~x, data = dt1,refresh=0)
```

```
## Warning: There were 315 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 714 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. S
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 1.89, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Markov chains did not converge! Do not analyze results!
```

```r
b <- coef(fit)
b1 <- coef(fit1)
hi <- 2
lo <- 1
delta <- invlogit(b[1]+b[2]*hi)-invlogit(b[1]+b[2]*lo)
delta1 <- invlogit(b1[1]+b[2]*hi)-invlogit(b1[1]+b1[2]*lo)
data.frame(b[2],b1[2],delta,delta1)
```

```
##         b.2.      b1.2.       delta      delta1
## x -0.2180021 -0.2205719 -0.05306205 -0.05230704
```

(c)

In general, when will it work reasonably well to fit a linear model to predict a binary outcome? See also Exercise 13.12.

After doing all these precess, I think the parameters are all binary and the regression is a linear regression, in this situation, the model can predict a binary outcome well.