MA678 Homework 4

Name

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)

```
wellss <- read.csv(file = '/Users/zarawaheed/Documents/BostonUniversity/MA678/Data/ROS-Examples-master/
wells <- data.frame(wellss)
fit_13.5 <- stan_glm(formula = switch ~ dist100 + arsenic, family=binomial(link="logit"),  data=wells, 
fit_13.5
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist100 + arsenic
##  observations: 3020
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept)  0.0    0.1  
## dist100     -0.9    0.1  
## arsenic      0.5    0.0  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

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.

Person with 0.5 arsenic level: Pr(0.5 arsenic) = invlogit(0 + 0.46 * 1) We divide 0.46 by 4 so we get 0.115. A 0.5 unit difference in arsenic level corresponds to a 11.5% increase in probability of switching with standard error. The 95% interval will be 0.115 +- 0.04*2 so [0.035, 0.195]

Person with 1 arsenic level: Pr(0.5 arsenic) = invlogit(0 + 0.46 * 0.5) We divide 0.46 by 4 so we get 0.115. A 0.5 unit difference in arsenic level corresponds to a 11.5/2 = 5.75% increase in probability of switching. The 95% interval will be 0.0575 +- 0.04*2 so [0.018, 0.098]

So the difference between them is approximately 5.75% with some uncertainty.

(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
# Data frame and prediction for arsenic 0.5 given mean dist100
arsenic0.5 <- data.frame(dist100=0.5, arsenic=0.5)
pred_arsen0.5 <- predict(fit_13.5, arsenic0.5)

# Data frame and prediction for arsenic 1 given mean dist100
arsenic1 <- data.frame(dist100=0.5, arsenic=1)
pred_arsen1 <- predict(fit_13.5, arsenic1)

# Compute the difference between the two predictions
diff <- pred_arsen1-pred_arsen0.5
```

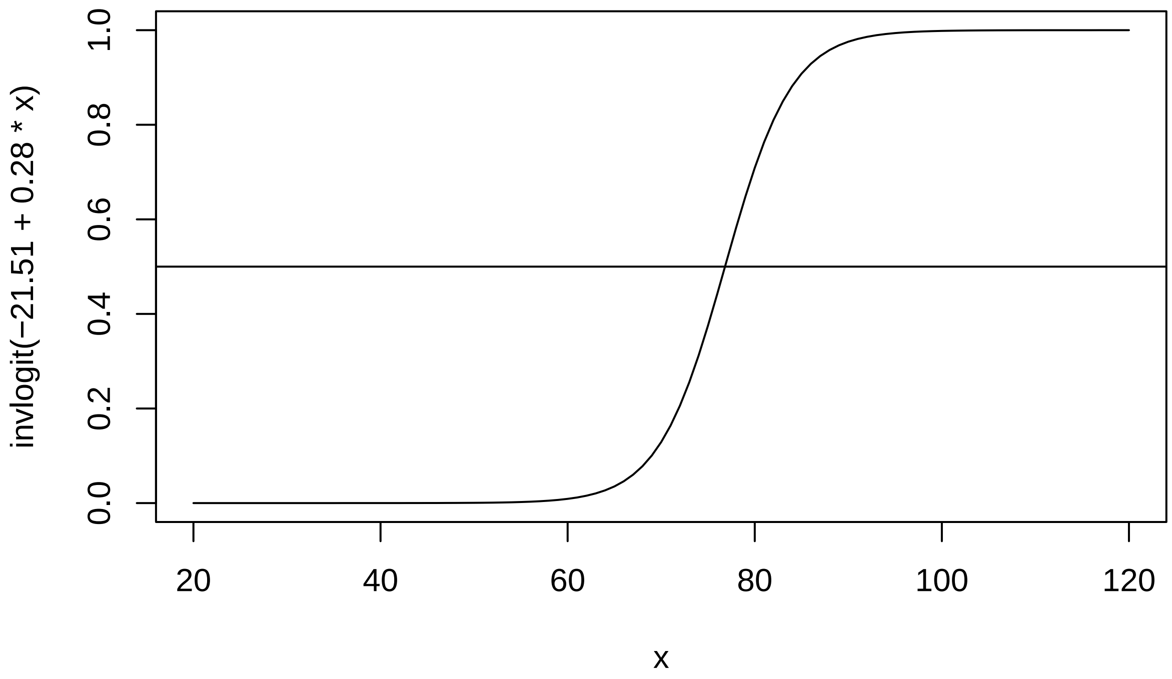
13.7

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(expr = invlogit(-21.51 +0.28*x), from =20, to = 120)
abline(a=0.5, b=0)
```



(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 7 in the probability of being heavy.

Divide by 4 rule: 0.28/4 = 0.07 So 7%

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 inches = 2.54 cm ~ 2.5 cm There will be no change in the intercept but the slope of the graph (coefficient of height) will be 2.5*2.8 = 7

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.

```
set.seed(1)

# Create a data frame with treatment and control group data
treatment <- data.frame(x=1, y=rbinom(500, 1, 0.5))
control <- data.frame(x=0, y=rbinom(500, 1, 0.4))
data <- rbind(treatment, control)
```

```
# Fit the model
fit_13.10 <- stan_glm(formula = y ~ x, family=binomial(link="logit"), data=data, refresh=0)

fit_13.10
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.4    0.1  
## x            0.2    0.1  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

(b)

Compare to the results from Exercise 4.1.

The results come out to be the same at 4.1 when we take build an equation with the results from part a.

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.

```
# Import the data
rodents <- data.frame(read.csv(file= "/Users/zarawaheed/Documents/BostonUniversity/MA678/Homework/HW_4/

# Assign categories
rodents$c.race <- factor(rodents$race, labels = c("White", "Black", "Puerto Rican", "Other Hispanic", "

# Build logistic regression model
rodent_race <- stan_glm(formula = rodent2 ~ c.race, family=binomial(link="logit"), data=rodents, refres

# Display data
rodent_race
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ c.race
##  observations: 13931
##  predictors:   7
## ------
##                        Median MAD_SD
## (Intercept)           -2.2    0.0  
## c.raceBlack            1.5    0.1  
## c.racePuerto Rican     1.6    0.1  
## c.raceOther Hispanic   1.7    0.1  
## c.raceAsian            0.8    0.1  
## c.raceAmer-Indian      1.3    0.4  
## c.raceTwo-or-more      1.1    0.3  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

The coefficients represent the average expected increase on the probability scale of rodents in households of people from different racial groups. The intercept represents the probability of rodents in white households since White was set as the baseline for race. The rest of the coefficients of race are represented as a comparison to white households.

(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.

```
# Build the model
rodent_other <- stan_glm(formula = rodent2 ~ c.race + numunits + poverty, family=binomial(link="logit")

# Display the model
rodent_other
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ c.race + numunits + poverty
##  observations: 13931
##  predictors:   9
## ------
##                        Median MAD_SD
## (Intercept)           -2.8    0.1  
## c.raceBlack            1.4    0.1  
## c.racePuerto Rican     1.4    0.1  
## c.raceOther Hispanic   1.6    0.1  
## c.raceAsian            0.8    0.1  
## c.raceAmer-Indian      1.2    0.4  
## c.raceTwo-or-more      1.0    0.2  
## numunits               0.1    0.0  
## poverty                0.4    0.0  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

I chose number of units and poverty as additional predictors for my model. The coefficients for the ethnicity predictors will still be interpreted in the same way, with white households as a baseline. The coefficient of poverty shows the 6.7% positive difference in probability of rodents between a poor household and a well-off one. The coefficient for numunits shows that for every additional unit, there is a 2.5% increase in probability 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.

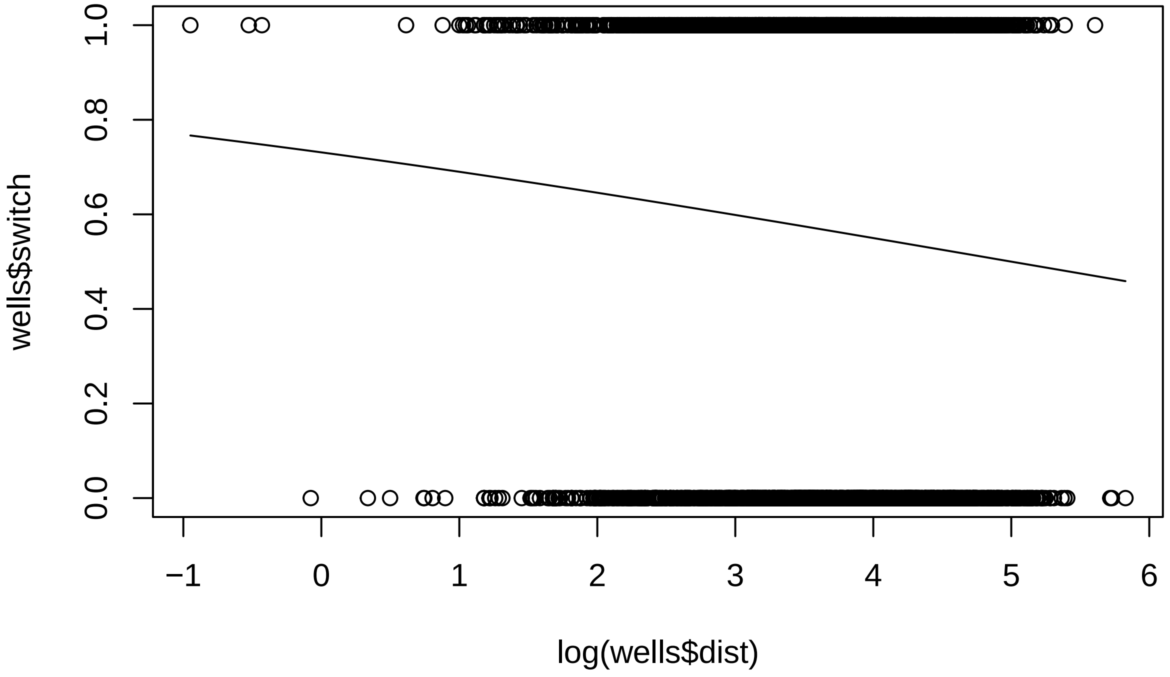
```
# Build the model
fit_14.3a <- stan_glm(formula = switch ~ log(dist), family=binomial(link="logit"),  data=wells, refresh

# Display the model
fit_14.3a
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ log(dist)
##  observations: 3020
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  1.0    0.2  
## log(dist)   -0.2    0.0  
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

(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.

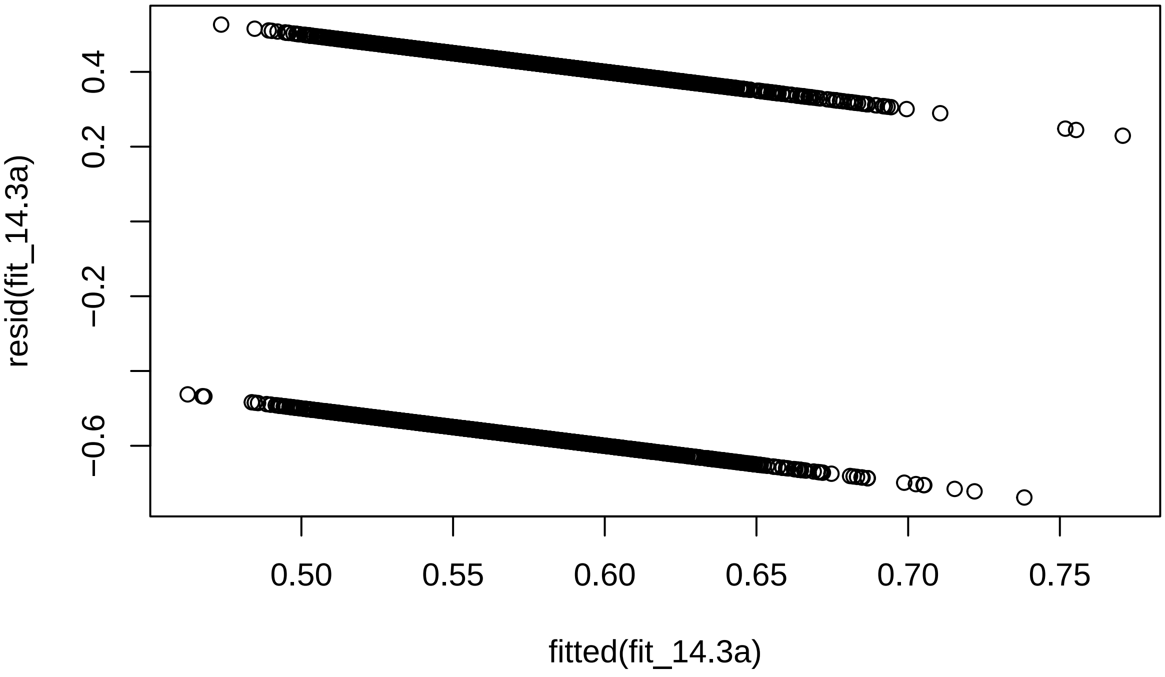
```
plot(log(wells$dist), wells$switch)
curve(expr = invlogit( 1 - 0.2*(x)), add = TRUE)
```



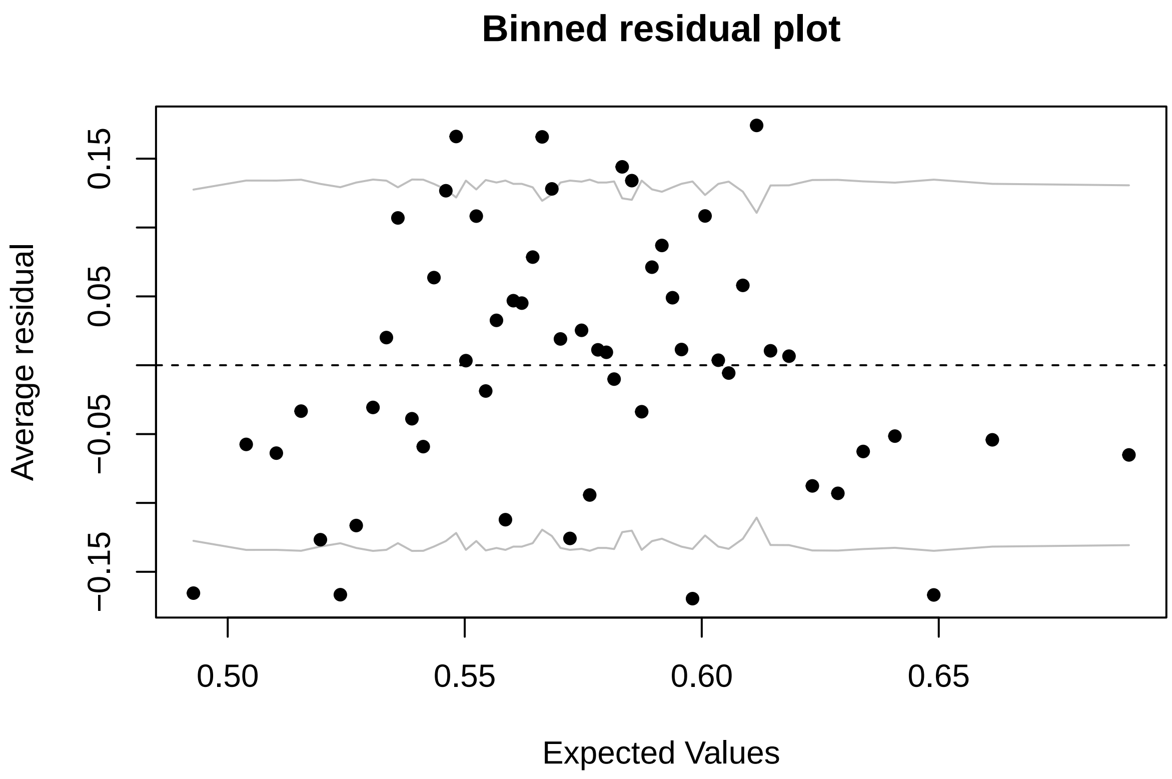
(c)

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

```
plot(fitted(fit_14.3a), resid(fit_14.3a))
```



```
binnedplot(fitted(fit_14.3a),resid(fit_14.3a))
```



(d)

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

```
pred_fit <- predict(fit_14.3a, type="response")
error_rate <- ifelse(pred_fit>0.5,1,0)
mean(error_rate!=wells$switch)
```

```
## [1] 0.4188742
```

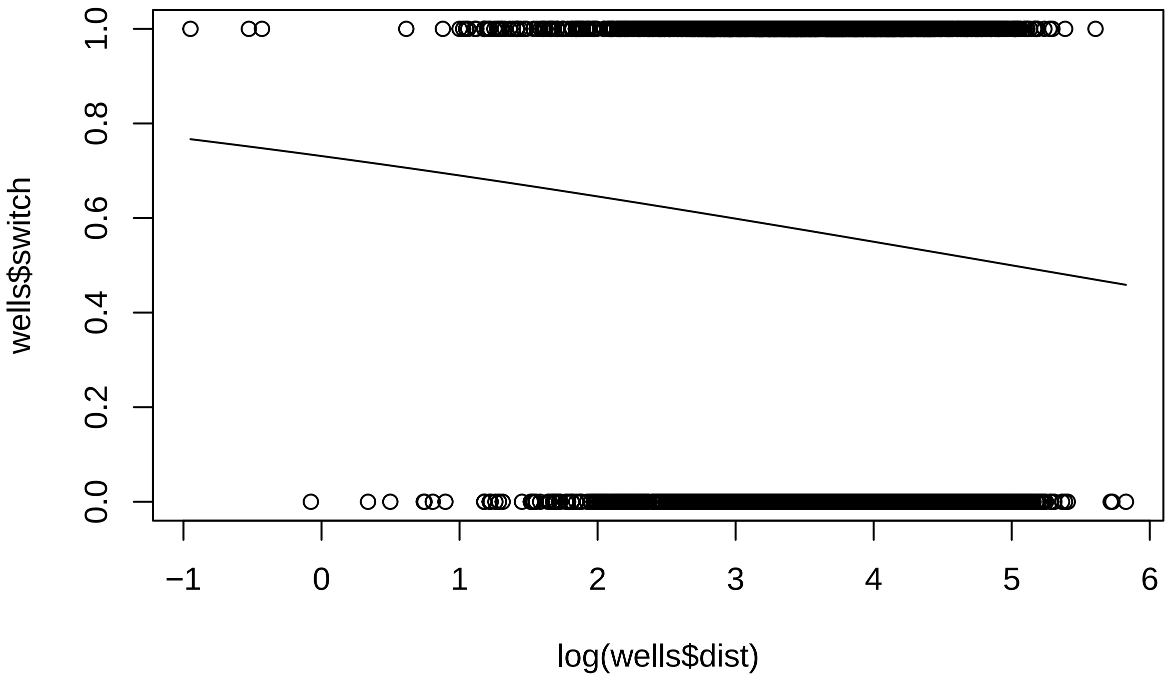
(e)

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.

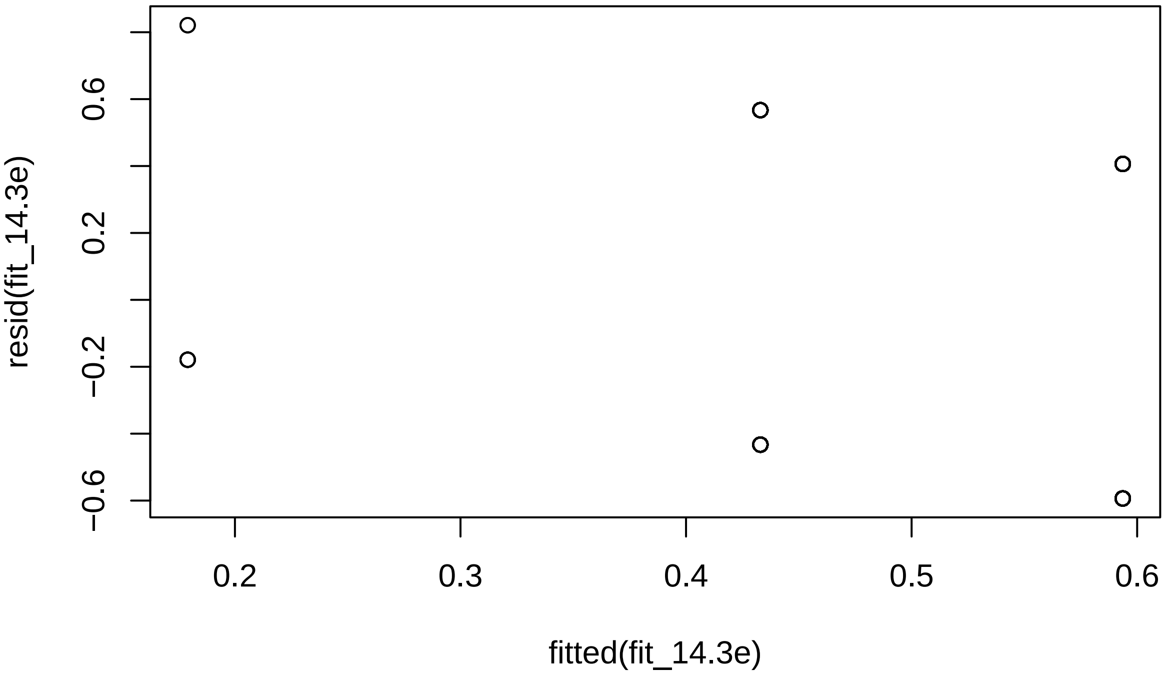
```
set.seed(1)

# Set up the three scenarios as binary predictors
wells$dist_lo <- ifelse(wells$dist<100, 1, 0)
wells$dist_mid <- ifelse(wells$dist>100 & wells$dist<200, 1, 0)
wells$dist_hi <- ifelse(wells$dist>200, 1, 0)

# Fit a model for effect of dist on switching
fit_14.3e <- stan_glm(switch ~ dist_lo + dist_mid + dist_hi, family=binomial(link="logit"), data=wells,

# Display the model
fit_14.3e
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist_lo + dist_mid + dist_hi
##  observations: 3020
##  predictors:   4
## ------
##              Median MAD_SD
## (Intercept)  0.2    6.0
## dist_lo      0.2    6.0
## dist_mid    -0.4    6.0
## dist_hi     -1.7    6.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

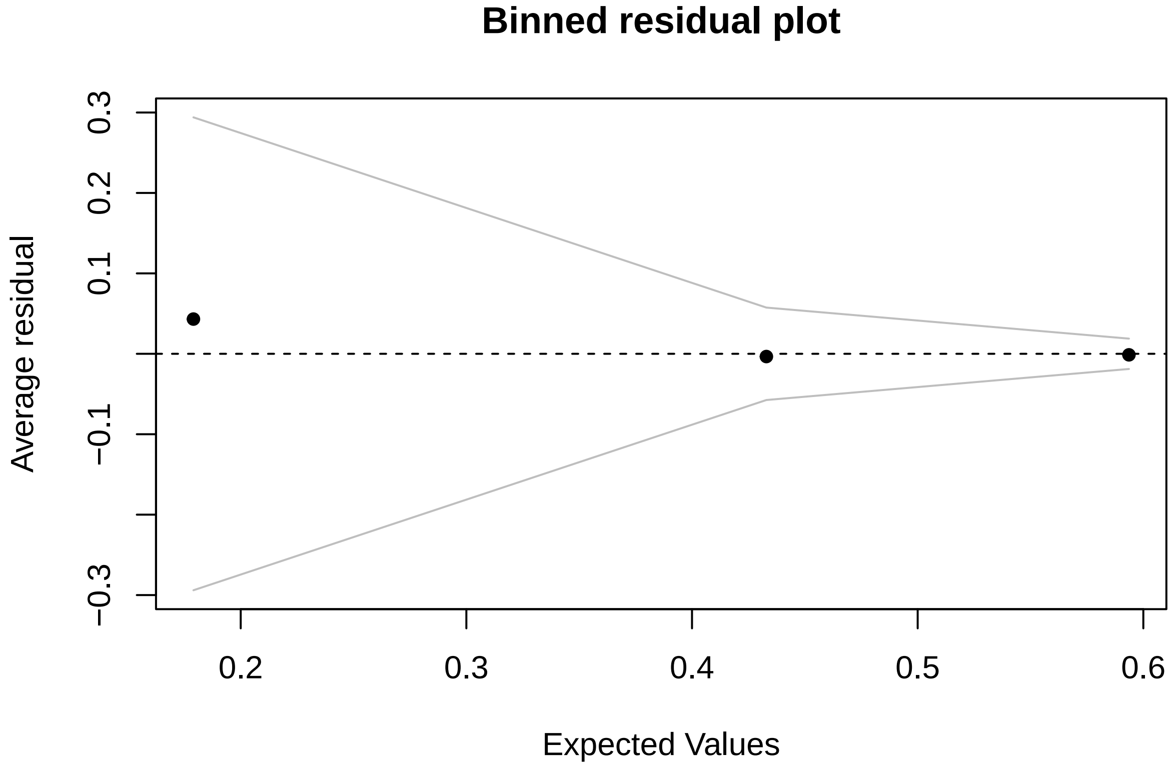
```
# Plot the graph
plot(log(wells$dist), wells$switch)
curve(expr = invlogit( 1 - 0.2*(x)), add = TRUE)
```



```
# Plot the residual and binned residual plots
plot(fitted(fit_14.3e), resid(fit_14.3e))
```



```
binnedplot(fitted(fit_14.3e),resid(fit_14.3e))
```

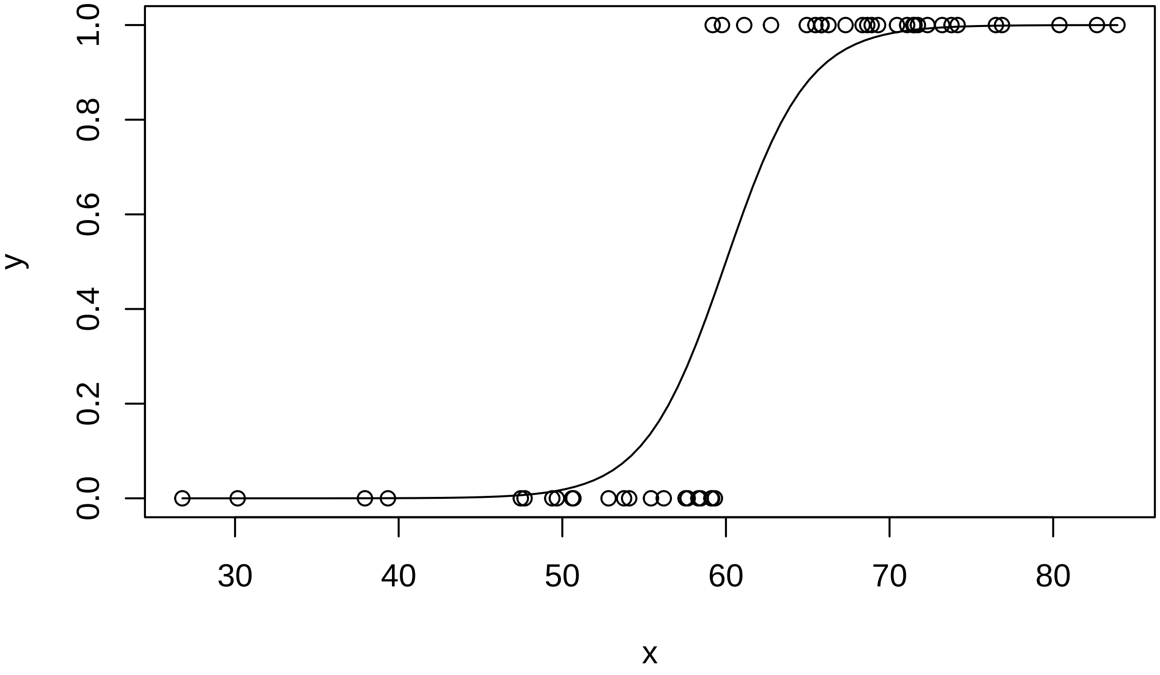


#14.5 Working with logistic regression: In a class of 50 students, a logistic regression is performed of course grade (pass or fail) on midterm exam score (continuous values with mean 60 and standard deviation 15). The fitted model is Pr(pass) = logit-1(-24 + 0.4x).

(a)

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

```
set.seed(1)

x <- rnorm(50,60,15)
prop <- invlogit(-24 + 0.4*x)
y <- rbinom(50,1,prop)

# Plot the graph and add the curve
plot(x,y)
curve(expr = invlogit(-24 + 0.4*x), add = TRUE)
```



(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?

```
data <- data.frame(x, y)
data$z <- (data$x-60)/15
stan_glm(y ~ z, family = binomial(link = "logit"), data = data, refresh = 0)
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ z
##  observations: 50
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.2    0.6  
## z            7.0    1.7  
##
```

```
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

(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?

```
newpred <- rnorm(50,0,1)
stan_glm(y ~ z, family = binomial(link = "logit"), data = data, refresh = 0)
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ z
##  observations: 50
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.2    0.5
## z            6.9    1.7
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

#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.

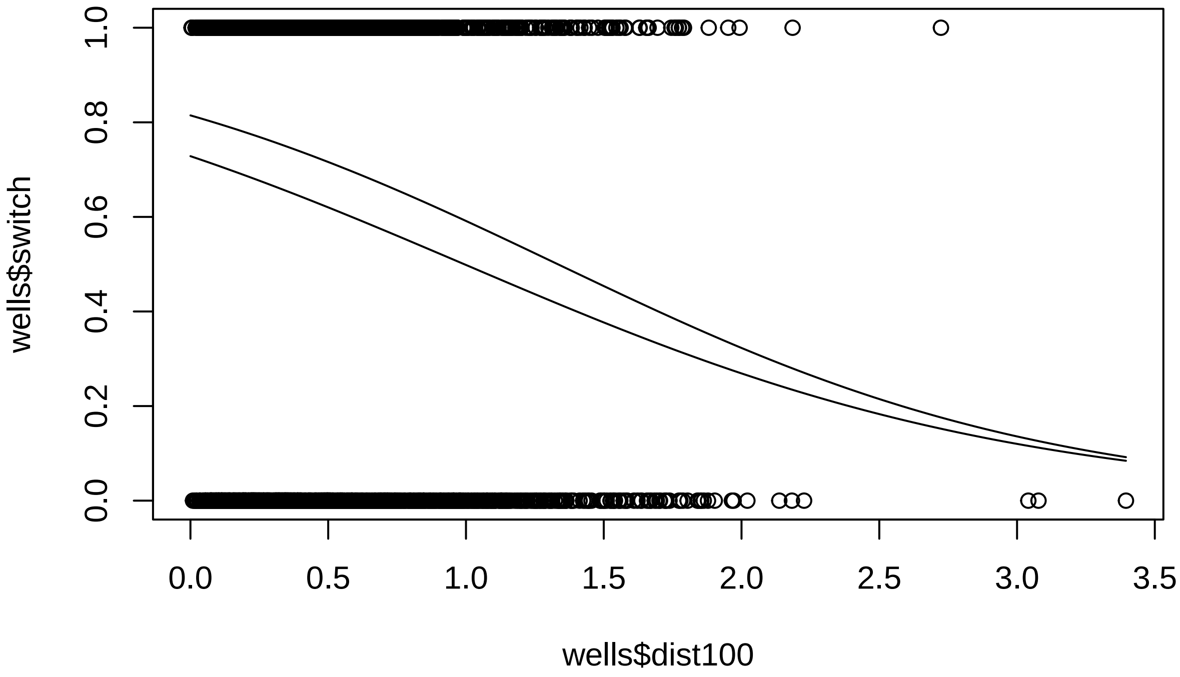
```
set.seed(1)

# Fit the model
fit_14.7a <- stan_glm( switch ~ dist100 + log(arsenic) + dist100:log(arsenic), data=wells, family=binomi

# Display the model
fit_14.7a
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist100 + log(arsenic) + dist100:log(arsenic)
##  observations: 3020
##  predictors:   4
## ------
##                      Median MAD_SD
## (Intercept)           0.5    0.1
## dist100              -0.9    0.1
## log(arsenic)          1.0    0.1
## dist100:log(arsenic) -0.2    0.2
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

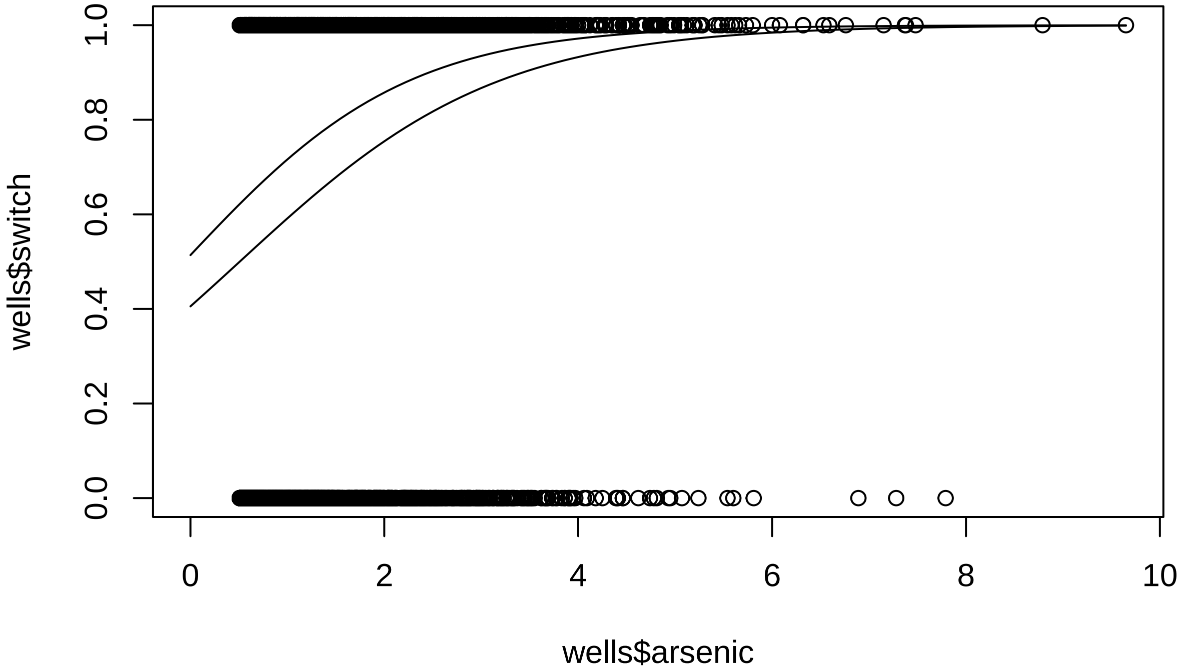
(b)

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

```
# Plot the graph for distance and switch
fig_14.7b <- plot(wells$dist100, wells$switch, xlim=c(0,max(wells$dist100)))

# Fit in the curves using information from GHV page 245
curve(invlogit(cbind(1, x, 0.5, 0.5*x) %*% coef(fit_14.7a)), add=TRUE)
curve(invlogit(cbind(1, x, 1.0, 1.0*x) %*% coef(fit_14.7a)), add=TRUE)
```



```
# Plot the graph for distance and switch
fig_14.7b <- plot(wells$arsenic, wells$switch, xlim=c(0,max(wells$arsenic)))

# Fit in the curves using information from GHV page 245
curve(invlogit(cbind(1, 0.5, x, 0.5*x) %*% coef(fit_14.7a)), add=TRUE)
curve(invlogit(cbind(1, 1, x, 1.0*x) %*% coef(fit_14.7a)), 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.

First we fit the model to be able to compute these results.

```r
set.seed(1)

# Fit the model and define b to read it's coefficients
fit_14.7c <- stan_glm( switch ~ dist100 + log(arsenic) + dist100:log(arsenic), data=wells, family=binomi
b <- coef(fit_14.7c)

# Display the model
fit_14.7c
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist100 + log(arsenic) + dist100:log(arsenic)
##  observations: 3020
##  predictors:   4
## ------
##                      Median MAD_SD
## (Intercept)           0.5    0.1  
## dist100              -0.9    0.1  
## log(arsenic)          1.0    0.1  
## dist100:log(arsenic) -0.2    0.2  
##
```

```
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

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

```
hi <- 1
lo <- 0
delta <- invlogit(b[1] + b[2]*hi + b[3]*log(wells$arsenic) + b[4]*hi*log(wells$arsenic)) -
         invlogit(b[1] + b[2]*lo + b[3]*log(wells$arsenic) + b[4]*lo*log(wells$arsenic))
round(mean(delta), 2)
```

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

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

```
hi <- 2
lo <- 1
delta <- invlogit(b[1] + b[2]*hi + b[3]*log(wells$arsenic) + b[4]*hi*log(wells$arsenic)) -
         invlogit(b[1] + b[2]*lo + b[3]*log(wells$arsenic) + b[4]*lo*log(wells$arsenic))
round(mean(delta), 2)
```

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

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

```
hi <- 1
lo <- 0
delta <- invlogit(b[1] + b[2]*wells$dist100 + b[3]*hi + b[4]*wells$dist100*hi) -
         invlogit(b[1] + b[2]*wells$dist100 + b[3]*lo + b[4]*wells$dist100*lo)
round(mean(delta), 2)
```

```
## [1] 0.2
```

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

```
hi <- 2
lo <- 1
delta <- invlogit(b[1] + b[2]*wells$dist100 + b[3]*hi + b[4]*wells$dist100*hi) -
         invlogit(b[1] + b[2]*wells$dist100 + b[3]*lo + b[4]*wells$dist100*lo)
round(mean(delta), 2)
```

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

The effect on switching wells of distance from a safw well is constant at approximately a 21% decrease in probability of switching with every 100m increase in distance. The probabilistic effect of arsenic levels in a well on switching wells seems to be increasing at an increasing rate as we increase arsenic levels.

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.

```r
set.seed(1)

# Define variables x, z and then y
x <- runif(100, -5, 5)
error <- rnorm(100, 0, 1)
z <- 1 - 1*x + error
y <- ifelse(z>0,1,0)

# Organise the data so it is easier to use to fit a model
data <- data.frame(x,y,z)

# Fit the model
fit_14.9a <- stan_glm(y ~ x, family = binomial(link = "logit"), data=data, refresh = 0)

# Display the model
fit_14.9a
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  2.3    0.6
## x           -2.0    0.4
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

(b)

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

```r
# Fit the model
fit_14.9b <- stan_glm(y ~ x, data=data, refresh = 0)

# Display the data
fit_14.9b
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  0.6    0.0
## x           -0.2    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.3    0.0
##
```

```
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

(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).

We use the formula from GHV 14.4

```
# Define hi and lo to compare 1 unit difference in y
b1 <- coef(fit_14.9a)
b2 <- coef(fit_14.9b)
hi <- 1
lo <- 0

# Compute the difference between hi and lo for both models

# Part a
delta1 <- invlogit(b1[1] + b1[2]*hi) -
         invlogit(b1[1] + b1[2]*lo)
round(mean(delta1), 2)
```

```
## [1] -0.33
```

```
# Part b
delta2 <- invlogit(b2[1] + b2[2]*hi) -
         invlogit(b2[1] + b2[2]*lo)
round(mean(delta2), 2)
```

```
## [1] -0.04
```

The logistic regression model difference is larger than the linear model.

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.

```
set.seed(1)

# Define variables x, z and then y
x <- runif(100, 0, 1)
error <- rnorm(100, 0, 7)
z <- 1 - 1*x + error
y <- ifelse(z>0,1,0)

# Organise the data so it is easier to use to fit a model
data <- data.frame(x,y,z)

# Fit the logistic regression model
```

```
fit_14.10a1 <- stan_glm(y ~ x, family = binomial(link = "logit"), data=data, refresh = 0)

# Display the logistic regression model
fit_14.10a1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.3    0.4  
## x            0.4    0.8  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Define hi and lo to compare 1 unit difference in y
b1 <- coef(fit_14.10a1)
hi <- 1
lo <- 0

# Compute the difference between hi and lo for both models

# Part a
delta1 <- invlogit(b1[1] + b1[2]*hi) -
         invlogit(b1[1] + b1[2]*lo)
round(mean(delta1), 2)
```

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

(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(1)

# Define variables x, z and then y
x <- runif(100, 0, 1)
error <- rnorm(100, 0, 0.1)
z <- 1 - 1*x + error
y <- ifelse(z>0,1,0)

# Organise the data so it is easier to use to fit a model
data <- data.frame(x,y,z)

# Fit the logistic regression model
fit_14.10a1 <- stan_glm(y ~ x, family = binomial(link = "logit"), data=data, refresh = 0)

# Display the logistic regression model
fit_14.10a1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  6.7    2.5  
## x           -3.7    3.4  
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Define hi and lo to compare 1 unit difference in y
b1 <- coef(fit_14.10a1)
hi <- 1
lo <- 0

# Compute the difference between hi and lo for both models

# Part a
delta1 <- invlogit(b1[1] + b1[2]*hi) -
         invlogit(b1[1] + b1[2]*lo)
round(mean(delta1), 2)
```

```
## [1] -0.05
```

(c)

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

If most of our probabilities in fall near the mean, linear models work better than logistic models but if the data has extreme values, all on one end of the data or the other, then logistic models work better than linear models.