MA678 Homework 4

Anna Cook

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.

```
# Estimate: 5.75%
## 0.46/4 = 0.115, so there is at most an 11.5% difference in probability that the second person will s

# standard error: 0.04/4 = 0.01

# 50% interval: [0.0507551, 0.0642449]
0.0575 + qnorm(c(0.25, 0.75))*0.01
```

```
## [1] 0.0507551 0.0642449
```

```
# 95% interval: [0.03790036, 0.07709964]
0.0575 + qnorm(c(0.025, 0.975))*0.01
```

```
## [1] 0.03790036 0.07709964
```

(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.

```
set.seed(100)
wells <- read.csv("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Arsenic/data/wells.cs
fit <- stan_glm(formula = switch ~ dist100 + arsenic, family=binomial(link="logit"),  data=wells, refre

#estimate: 5.75%
new1 <- data.frame(dist100=.50, arsenic=0.5)
new2 <- data.frame(dist100=.50, arsenic=1)

epred1 <- posterior_epred(fit, newdata = new1)
epred2 <- posterior_epred(fit, newdata = new2)

estimate <- mean(epred2) - mean(epred1)

#standard error: 0.018
se <- sqrt(sd(epred1)^2 + sd(epred2)^2)

# 50% interval: [0.04525729, 0.06975775]
estimate + qnorm(c(0.25, 0.75))*se
```

```
## [1] 0.04525729 0.06975775
```

```
# 95% inverval: [0.02191022, 0.09310481]
estimate + qnorm(c(0.025, 0.975))*se
```

```
## [1] 0.02191022 0.09310481
```

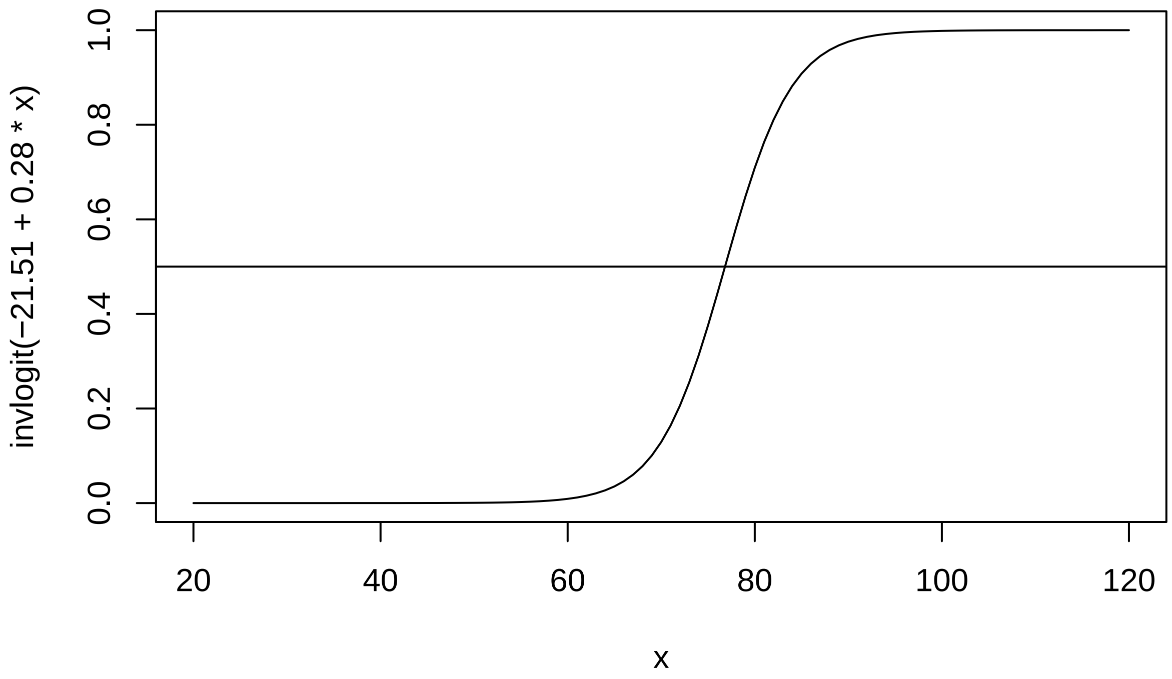
13.7

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.

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



```
(logit(0.5)+21.51)/0.28 #this is the height at which there is a 50% probability of a person being heavy
```

```
## [1] 76.82143
```

(b)

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

0.28/4 = 0.07, so we expect a difference of 7% in the probability of being heavy.

13.8

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

height in cm = height * 2.54

slope: 0.28/2.54 = 0.11

the intercept would stay the same because we are only scaling, not centering.

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
set.seed(108)
x <- rep(c(0,1), c(500, 500))
y <- rep(c(0,1,0,1), c(300, 200, 250, 250))
df <- data.frame(x, y)
fit <- stan_glm(y ~ x, family = binomial(link = "logit"), data = df, refresh = 0)
fit
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.4    0.1  
## x            0.4    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.

```r
#p(y==1) = invlogit(-0.4 + 0.4) = 0.5
#p(y==0) = invlogit(-0.4) = 0.4

# These results are the same as in exercise 4.1
```

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.

```r
rodents <- read.table("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/Rodents/rodents.da
rodents$race <- factor(rodents$race, labels = c("White", "Black", "Puerto Rican", "Other Hispanic", "As
fit <- stan_glm(rodent2 ~ race, family=binomial(link="logit"),  data= rodents, refresh = 0)
fit
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ race
##  observations: 1551
##  predictors:   7
## ------
##                    Median MAD_SD
## (Intercept)        -2.2    0.1  
## raceBlack           1.4    0.2  
## racePuerto Rican    1.6    0.2
## raceOther Hispanic  2.0    0.2
## raceAsian           0.8    0.3
## raceAmer-Indian     0.3    1.2
## raceTwo or more     0.1    1.2
##
## ------
## * 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 in log odds of rodents being present in hous
```

(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.

```
rodents$borough <- factor(rodents$borough, labels = c("Bronx", "Brooklyn", "Manhattan", "Queens", "Stat
fit <- stan_glm(rodent2 ~ race + borough + poverty_Mean, family=binomial(link="logit"),  data= rodents,
fit
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      rodent2 ~ race + borough + poverty_Mean
##  observations: 1551
##  predictors:   12
## ------
##                       Median MAD_SD
## (Intercept)           -2.7    0.3
## raceBlack              1.0    0.2
## racePuerto Rican       1.2    0.2
## raceOther Hispanic     1.7    0.2
## raceAsian              0.9    0.3
## raceAmer-Indian       -0.1    1.2
## raceTwo or more       -0.4    1.2
## boroughBrooklyn        0.1    0.2
## boroughManhattan      -0.1    0.2
## boroughQueens         -0.5    0.2
## boroughStaten Island -1.6    0.7
## poverty_Mean           3.7    0.8
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# The coefficients for the ethnicity indicators still hold the same meaning as above, except in this mo
```

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.

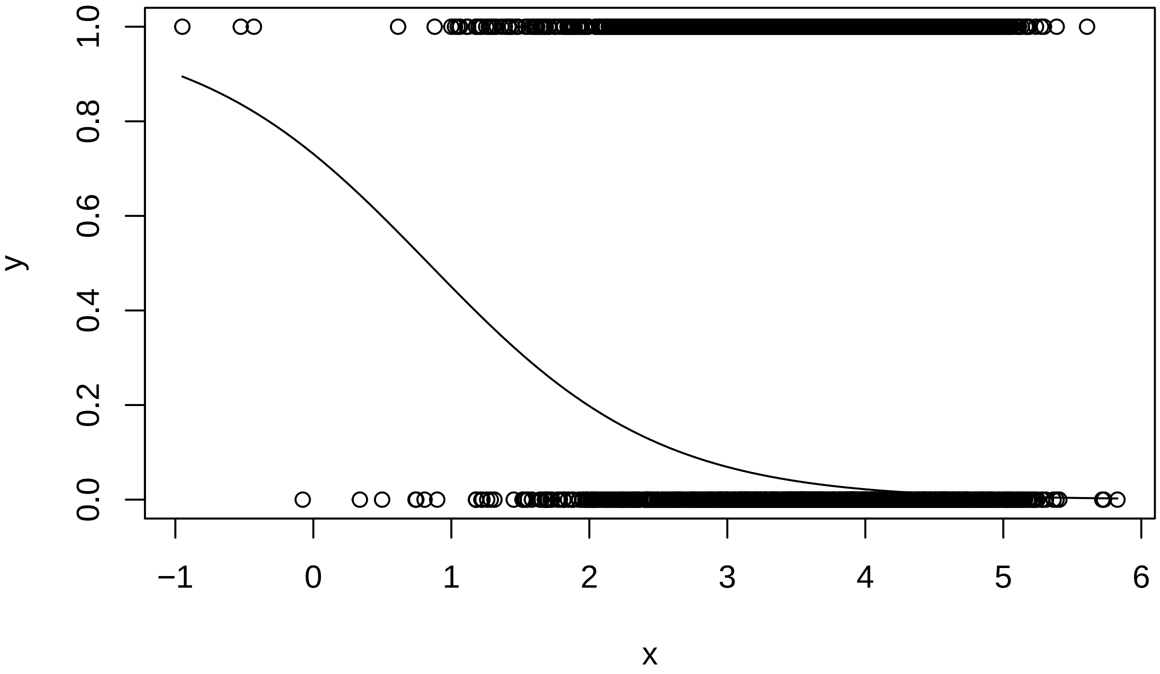
```
fit <- stan_glm(switch ~ log(dist), family = binomial(link = "logit"), data = wells, refresh = 0)
fit
```

```
## 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.

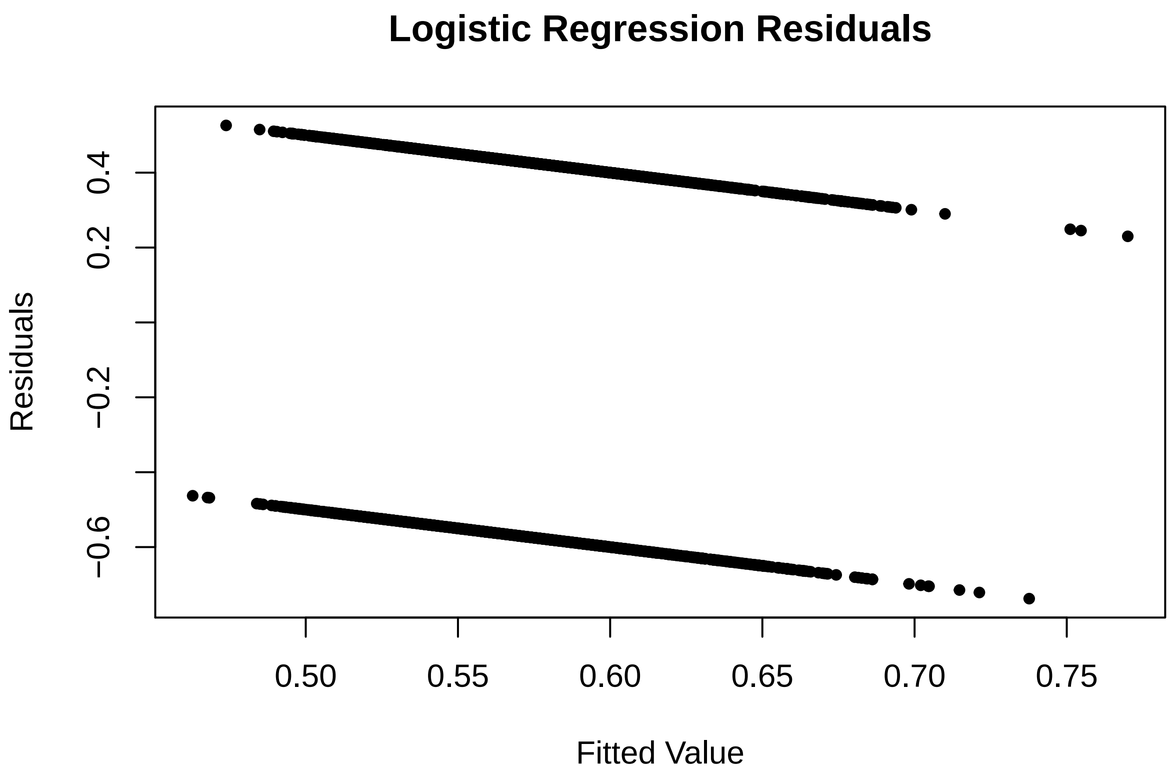
```
x <- log(wells$dist)
y<- wells$switch
plot(x,y)
curve(expr = invlogit(1.0 - 1.2*(x)), add = TRUE)
```



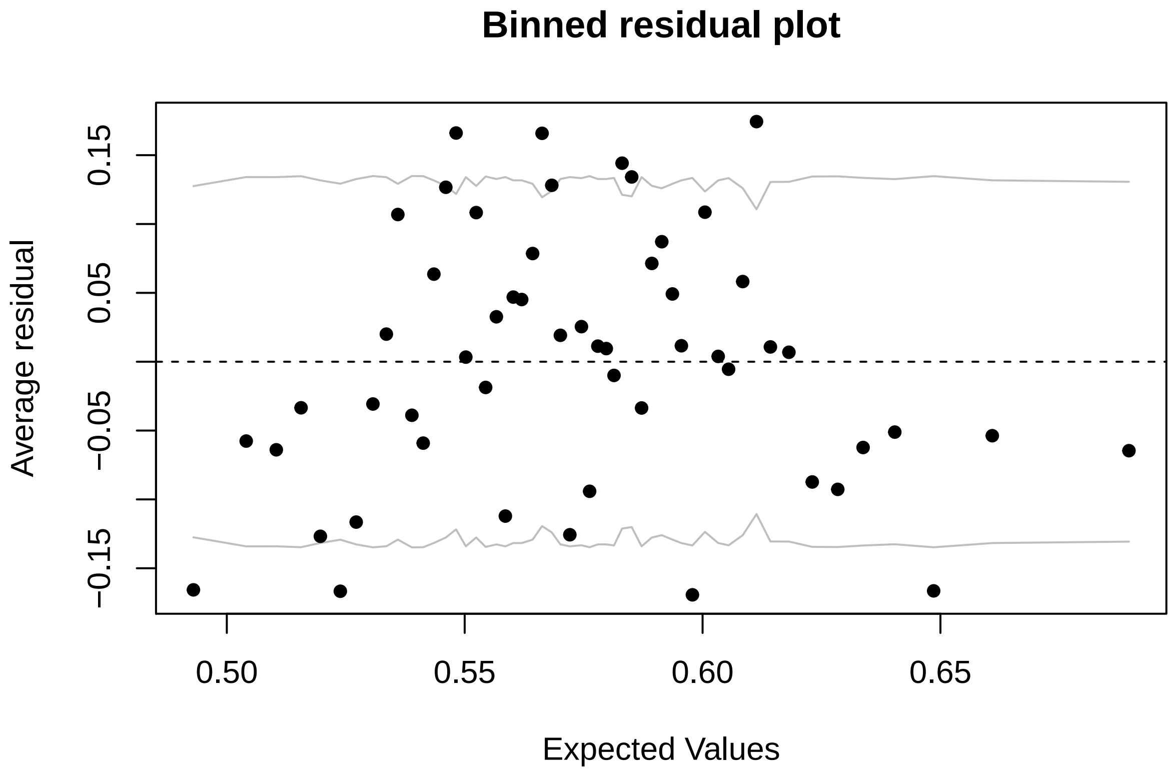
(c)

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

```
plot(fitted(fit),resid(fit),pch=20, main="Logistic Regression Residuals",xlab="Fitted Value",ylab="Resi
```



```
binnedplot(fitted(fit),resid(fit))
```



(d)

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

```
fit.probs <- predict(fit,type="response")
Switch = wells$switch

fit.pred = ifelse(fit.probs>0.5,1,0)
table(fit.pred,Switch)
```

```
##         Switch
## fit.pred    0    1
##        0   35   18
##        1 1248 1719
```

```
mean(fit.pred==Switch) #success rate
```

```
## [1] 0.5807947
```

```
mean(fit.pred!=Switch) #error rate
```

```
## [1] 0.4192053
```

```
sum((wells$switch==0)/(nrow(wells)))
```

```
## [1] 0.4248344
```

```
# the error rate of the fitted model is barely below the error rate of the null model, so this model is
```

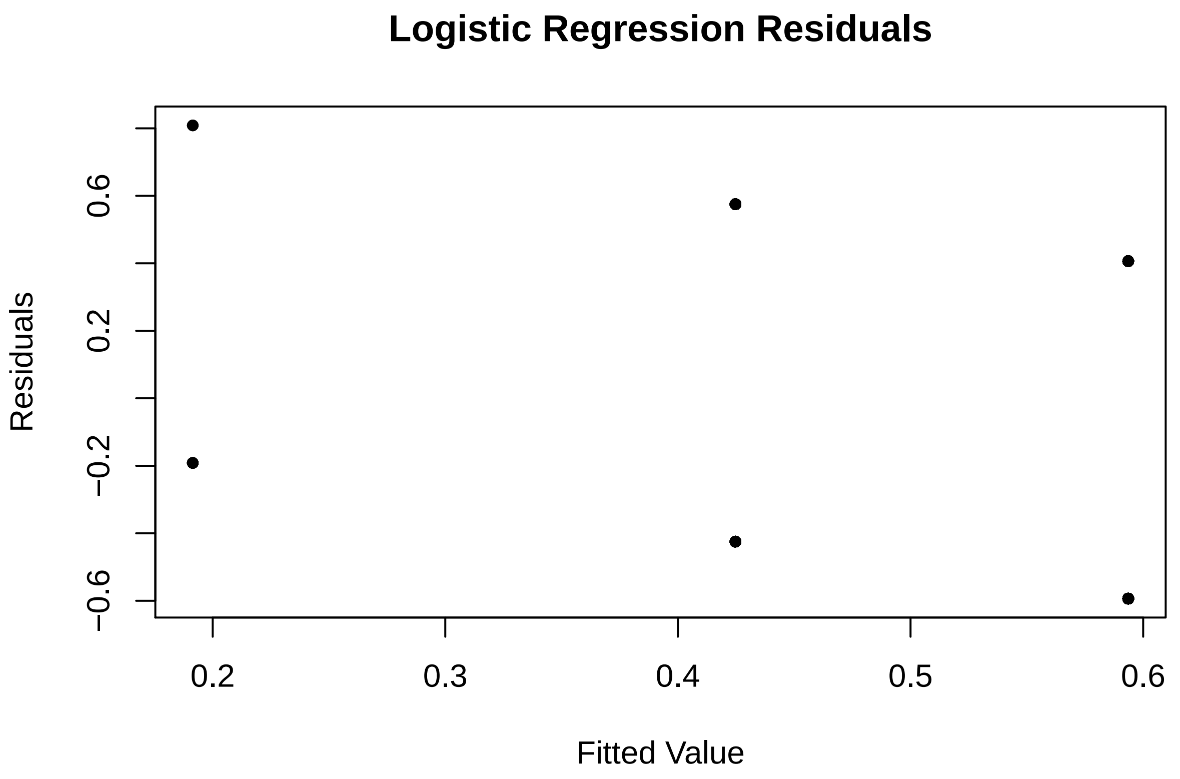
(e)

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.

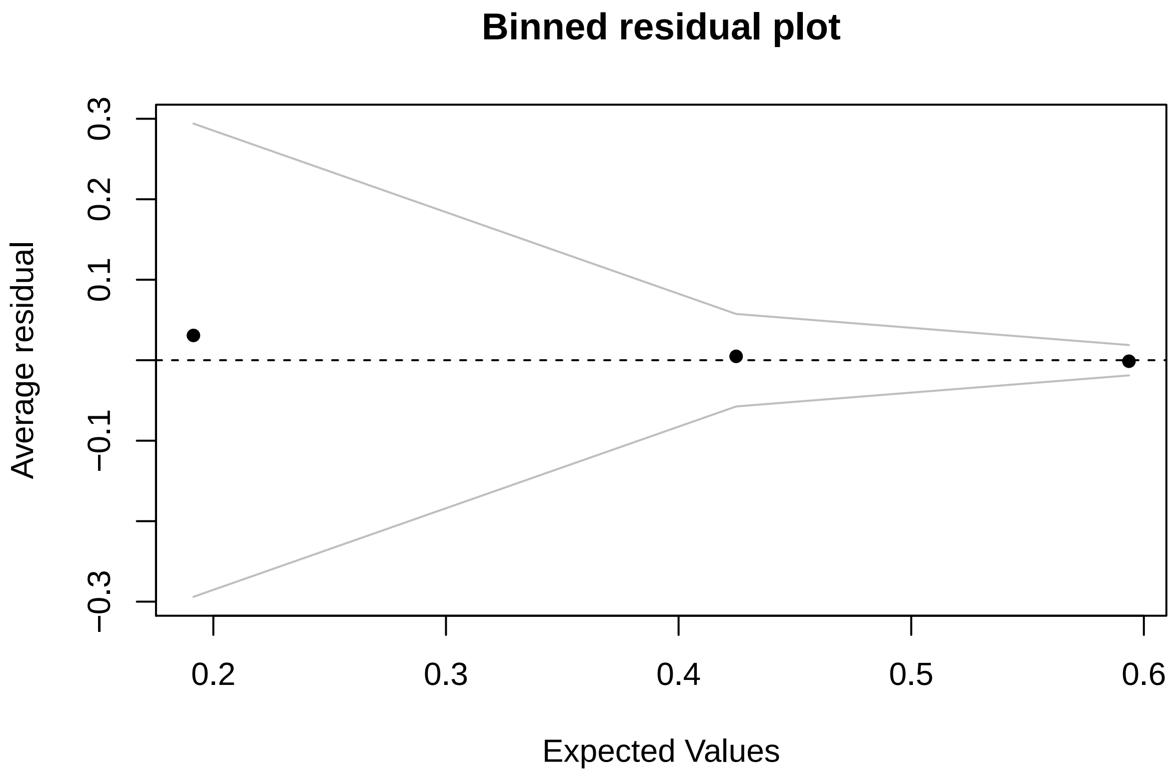
```
wells <- mutate(wells, dist_low = ifelse(wells$dist < 100, 1, 0), dist_mid = ifelse(wells$dist >= 100 &

fit <- stan_glm(switch ~ dist_low + dist_mid + dist_high, family = binomial(link = "logit"), data = wel

plot(fitted(fit),resid(fit),pch=20, main="Logistic Regression Residuals",xlab="Fitted Value",ylab="Resi
```



```
binnedplot(fitted(fit),resid(fit))
```



```
fit.probs <- predict(fit,type="response")
Switch = wells$switch
```

```
fit.pred = ifelse(fit.probs>0.5,1,0)
table(fit.pred,Switch)
```

```
##          Switch
## fit.pred    0    1
##        0  177  130
##        1 1106 1607
```

```
mean(fit.pred==Switch) #success rate
```

```
## [1] 0.5907285
```

```
mean(fit.pred!=Switch) #error rate
```

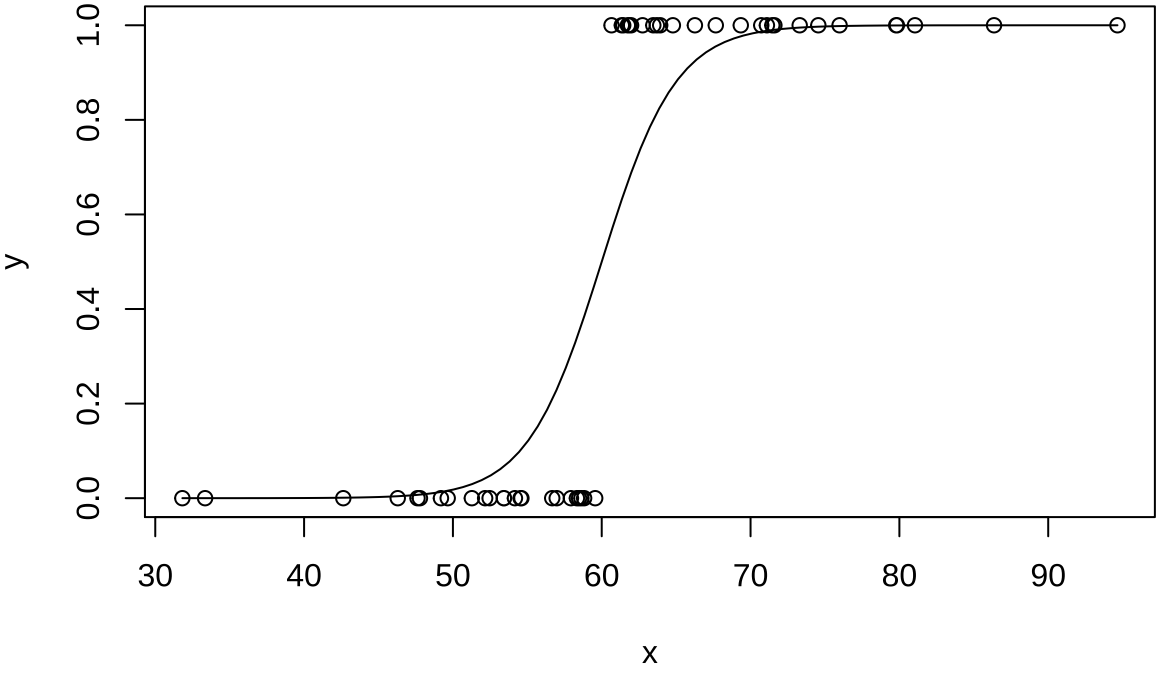
```
## [1] 0.4092715
```

#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(100)
x <- rnorm(50, 60, 15)
y <- ifelse(x >= 60, 1, 0)
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?

```
set.seed(100)
x_z <- (x - mean(x))/sd(x)
df <- data.frame(x_z, y)
stan_glm(y ~ x_z, family = binomial(link = "logit"), data = df, refresh = 0)
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x_z
##  observations: 50
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 0.6    0.5
## x_z         6.3    1.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
# Pr(passing) = invlogit(0.6 + 6.3*x_z)
```

(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)
fit <- stan_glm(y ~ x_z + newpred, family = binomial(link = "logit"), data = df, refresh = 0)
fit
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x_z + newpred
##  observations: 50
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 0.6    0.5
## x_z         6.5    1.6
## newpred     0.1    0.4
##
## ------
## * 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(100)
fit <- stan_glm(switch ~ dist100 + log(arsenic) + dist100:log(arsenic), data = wells, family = binomial
fit
```

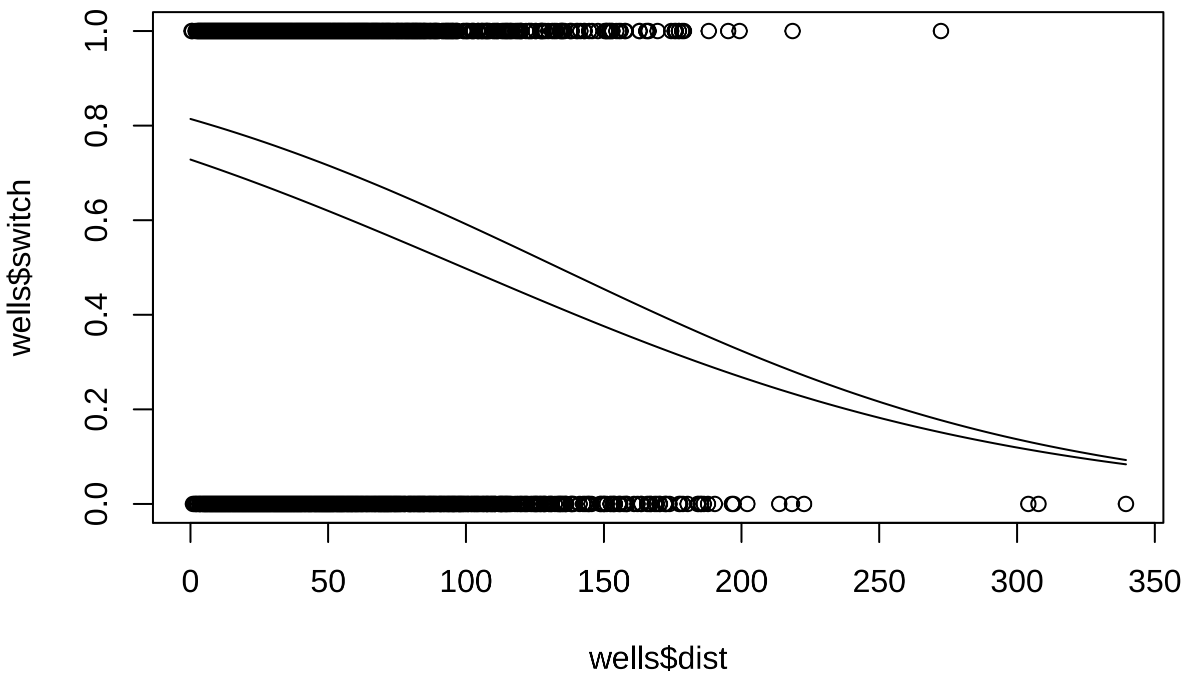
```
## stan_glm
##  family:       binomial [logit]
##  formula:      switch ~ dist100 + log(arsenic) + dist100:log(arsenic)
##  observations: 3020
##  predictors:   4
## ------
##                      Median MAD_SD
## (Intercept)           0.5    0.1  
## dist100              -0.9    0.1  
## log(arsenic)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

```
#dist100 has a negative coefficient, suggesting that the longer the distance to a safe well, the lower
```

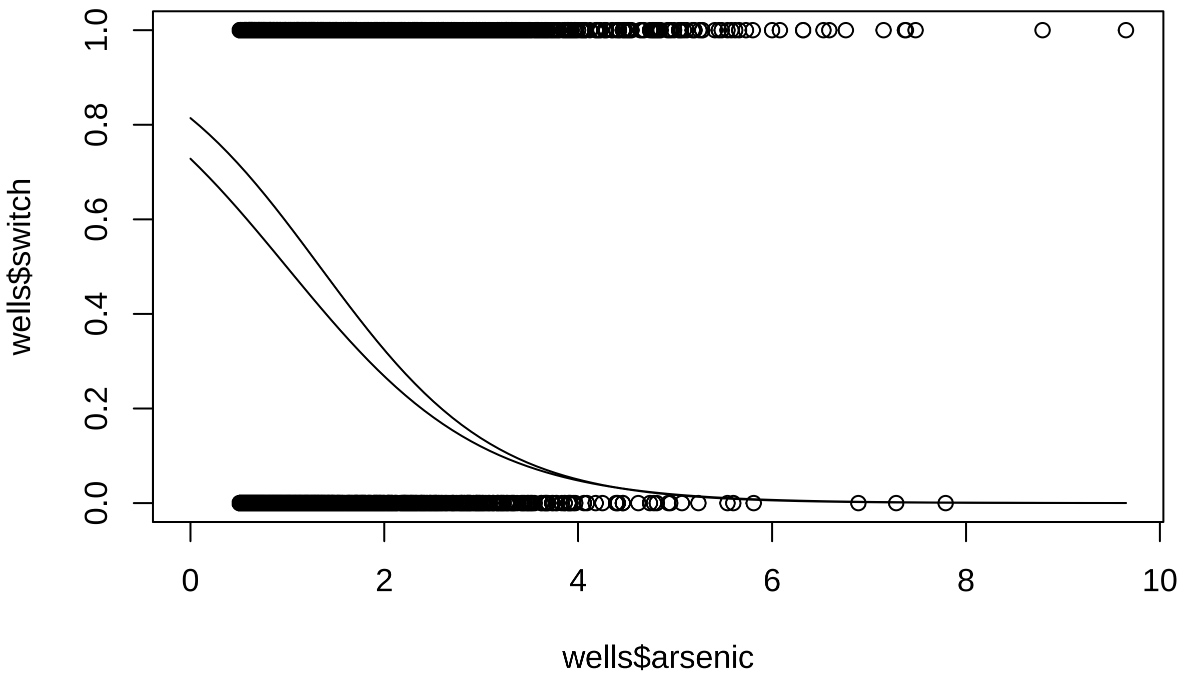
(b)

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

```
plot1 <- plot(wells$dist, wells$switch, xlim=c(0,max(wells$dist)))
curve(invlogit(cbind(1, x/100, 0.5, 0.5*x/100) %*% coef(fit)), add = TRUE)
curve(invlogit(cbind(1, x/100, 1.0, 1.0*x/100) %*% coef(fit)), add = TRUE)
```



```
plot2 <- plot(wells$arsenic, wells$switch, xlim=c(0,max(wells$arsenic)))
curve(invlogit(cbind(1, x, 0.5, 0.5*x) %*% coef(fit)), add = TRUE)
curve(invlogit(cbind(1, x, 1.0, 1.0*x) %*% coef(fit)), 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.

```
fit <- stan_glm(switch ~ dist100 + arsenic, family = binomial(link = "logit"), data = wells, refresh =

b <- coef(fit)

# i
hi <- 1
lo <- 0
delta <- invlogit(b[1] + b[2]*hi + b[3]*wells$arsenic) - invlogit(b[1] + b[2]*lo + b[3]*wells$arsenic)
mean(delta)
```

```
## [1] -0.2065777
```

```
# ii
hi <- 2
lo <- 1
delta <- invlogit(b[1] + b[2]*hi + b[3]*wells$arsenic) - invlogit(b[1] + b[2]*lo + b[3]*wells$arsenic)
mean(delta)
```

```
## [1] -0.1937111
```

```
# iii
hi <- 1
lo <- 0.5
delta <- invlogit(b[1] + b[2]*wells$dist100 + b[3]*hi) - invlogit(b[1] + b[2]*wells$dist100 + b[3]*lo)
```

```
mean(delta)
```

```
## [1] 0.05606727
```

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

```
## [1] 0.1099219
```

```
## These results show that as distance increases, the probability of switching wells decreases (by roug
```

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(100)
x <- runif(100, -10, 10)
a <- 1
b <- -1
z <- a + b*x + rnorm(100,0,1)
y <- ifelse(z>0,1,0)
df <- data.frame(x, z, y)

fit1 <- stan_glm(y ~ x, family = binomial(link = "logit"), data = df, refresh = 0)
fit1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  1.2    0.5  
## x           -1.1    0.2  
## 
## ------
## * 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.

```
fit2 <- stan_glm(y ~ x, data = df, refresh = 0)
fit2
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  0.6    0.0
## x           -0.1    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).

```r
b1 <- coef(fit1)
b2 <- coef(fit2)
hi <- 1
lo <- 0

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

```
## (Intercept)
##  -0.2503152
```

```r
# b
delta2 <- invlogit(b2[1] + b2[2]*hi) - invlogit(b2[1] + b2[2]*lo)
delta2
```

```
## (Intercept)
## -0.01883979
```

```r
## the predicted difference in probability in much larger for the logistic regression model than for th
```

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.

```r
set.seed(100)
x <- runif(100, 0, 1)
```

```
a <- -1
b <- 1
z <- a + b*x + rnorm(100,0,5)
y <- ifelse(z>0,1,0)
df <- data.frame(x, z, y)

fit1 <- stan_glm(y ~ x, family = binomial(link = "logit"), data = df, refresh = 0)
fit1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.4    0.4  
## x            0.0    0.8  
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
fit2 <- stan_glm(y ~ x, data = df, refresh = 0)
fit2
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 0.4    0.1   
## x           0.0    0.2   
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.5    0.0   
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
b1 <- coef(fit1)
b2 <- coef(fit2)
hi <- 1
lo <- 0

delta1 <- invlogit(b1[1] + b1[2]*hi) - invlogit(b1[1] + b1[2]*lo)
delta1
```

```
##  (Intercept)
## -0.002585908
```

```r
delta2 <- invlogit(b2[1] + b2[2]*hi) - invlogit(b2[1] + b2[2]*lo)
delta2
```

```
##  (Intercept)
## -0.001654502
```

(b)

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

```r
set.seed(100)
x <- runif(100, 0, 1)
a <- -1
b <- 1
z <- a + b*x + rnorm(100,0,0.1)
y <- ifelse(z>0,1,0)
df <- data.frame(x, z, y)

fit1 <- stan_glm(y ~ x, family = binomial(link = "logit"), data = df, refresh = 0)
fit1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -8.8    2.8
## x            7.6    3.3
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
fit2 <- stan_glm(y ~ x, data = df, refresh = 0)
fit2
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -0.1    0.0
## x            0.2    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.2    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
b1 <- coef(fit1)
b2 <- coef(fit2)
hi <- 1
lo <- 0

delta1 <- invlogit(b1[1] + b1[2]*hi) - invlogit(b1[1] + b1[2]*lo)
delta1
```

```
## (Intercept)
##   0.2363452
```

```
delta2 <- invlogit(b2[1] + b2[2]*hi) - invlogit(b2[1] + b2[2]*lo)
delta2
```

```
## (Intercept)
##  0.04594281
```

(c)

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

```
# it seems to work reasobably well to use a linear model when the probabilities are moderate. So if the
```