MA678 Homework 4

Jing Xu

10/10/2022

Disclaimer (remove after you've read)!

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

13.5 Interpreting logistic regression coefficients

Here is a fitted model from the Bangladesh analysis predicting whether a person with high-arsenic drinking water will switch wells, given the arsenic level in their existing well and the distance to the nearest safe well:

```
stan_glm(formula = switch ~ dist100 + arsenic, family=binomial(link="logit"), data=wells)
             Median MAD_SD
(Intercept)   0.00   0.08
dist100      -0.90   0.10
arsenic       0.46   0.04
```

Compare two people who live the same distance from the nearest well but whose arsenic levels differ, with one person having an arsenic level of 0.5 and the other person having a level of 1.0. You will estimate how much more likely this second person is to switch wells. Give an approximate estimate, standard error, 50% interval, and 95% interval, using two different methods:

(a)

Use the divide-by-4 rule, based on the information from this regression output.

Holding all others constant, an increase of 1 arsenic level is averagely associated with no more than 11.5% increase in the probability of switching wells; an increase of 100 distance is averagely associated with no more than 22.5% decrease in the probability of switching wells.

Therefore, the second person has 5.25% higher probability of switching wells.

(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.

based on the given R model, the first person has -22% probability of switching wells, and the second person has 1% probability of switching wells.

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
```

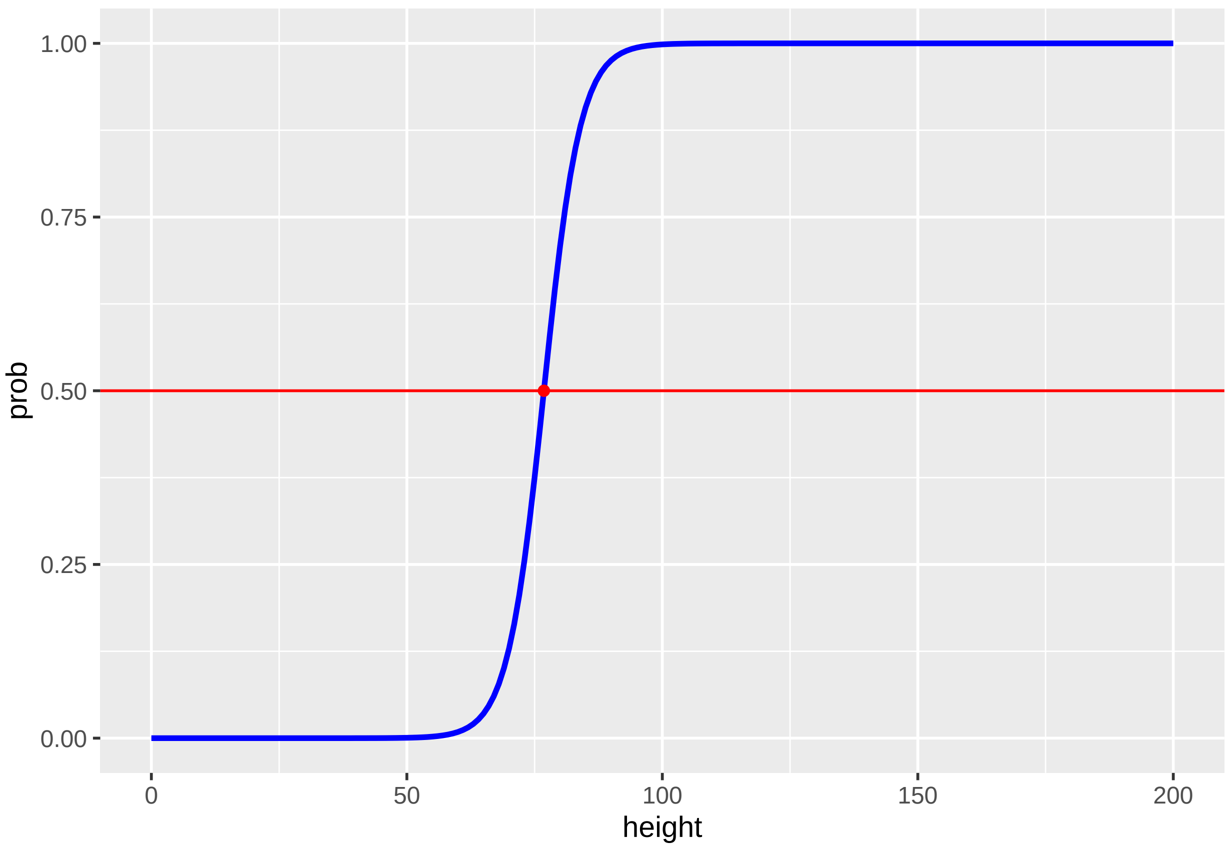
and fit a logistic regression, predicting heavy from `height` (in inches):

```
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.

```
height <- seq(0, 200, 1)
prob <- invlogit(-21.51 + 0.28*height)
df <- data.frame(height, prob)

ggplot(data = df, aes(x = height, y = prob)) +
  geom_line(color = "blue", lwd = 1) +
  geom_hline(yintercept = .50, col = 'red') +
  geom_point(data = data.frame(prob = .50, height = (21.51)/0.28), mapping = aes(x = height, y = prob),
```



(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.

0.28/4

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 inch = 2.56 cm, let height_cm = height*2.56 after transformation we have: height = height_cm/2.56. substitute into the function: Pr(heavy = 1) = logit(-21.51 + 0.28height) = logit(-21.51 + 0.28height_cm/2.56) = logit(-21.51 + 0.11 height_cm).

noticed that intercept = -21.51 and slope = 0.11.

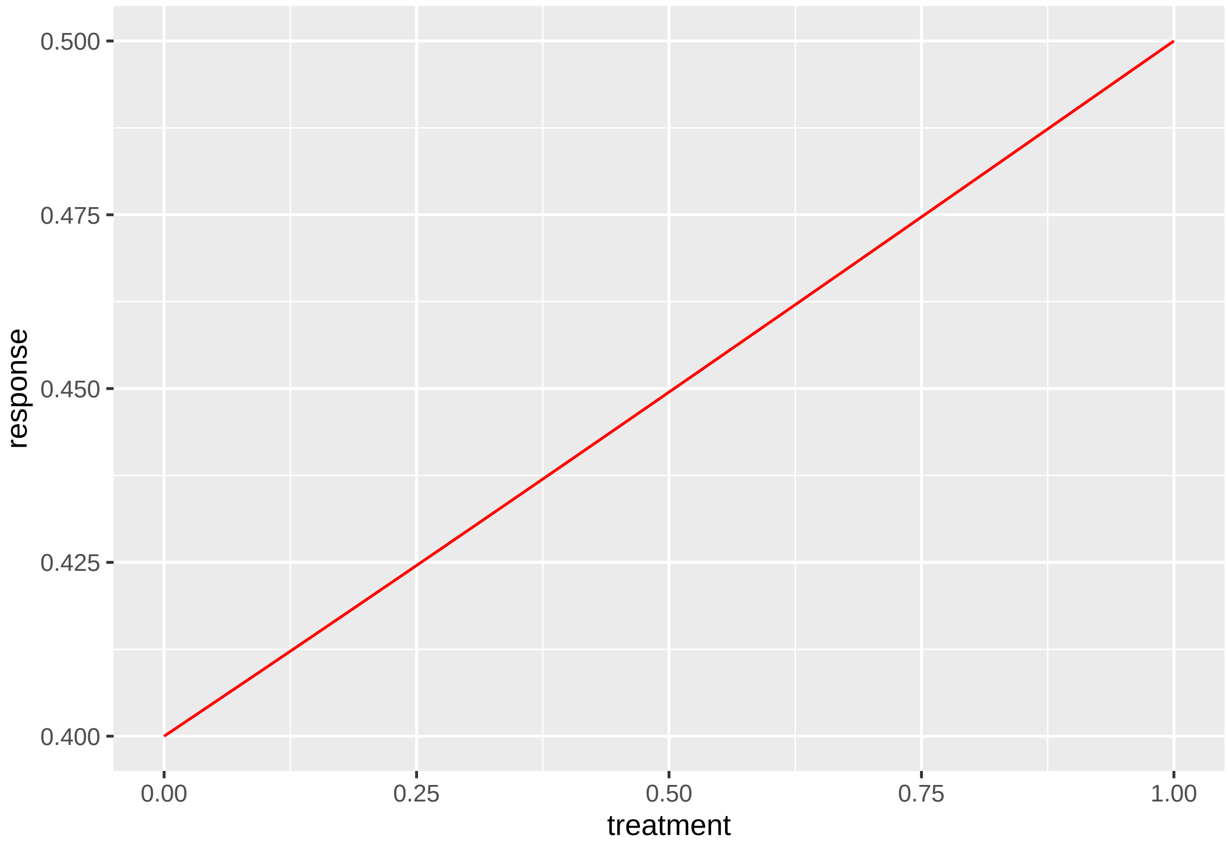
13.10 Expressing a comparison of proportions as a logistic regression

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.

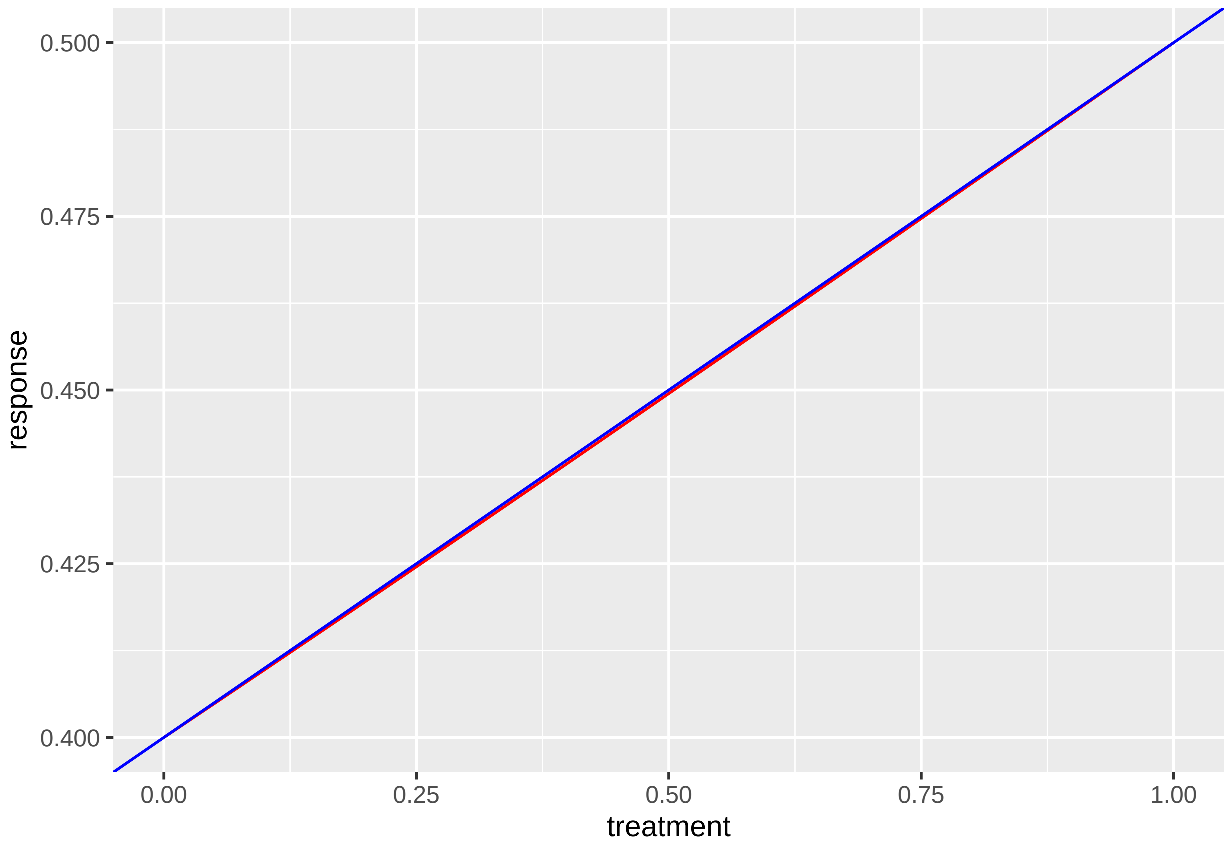
```r
set.seed(100)
x1 <- rep(1, 500)
x2 <- rep(0, 500)
y1 = rep(0, 500)
y2 = rep(0, 500)
y1[sample(500, 250, replace = FALSE)] = 1
y2[sample(500, 200, replace = FALSE)] = 1
df <- data.frame(response = c(y1, y2), treatment = c(x1, x2))
fit_1 <- glm(response ~ treatment, data = df, family = 'binomial')
b <- coef(fit_1)
treatment <- seq(0, 1, 0.01)
response <- invlogit(b[1] + b[2]*treatment)
df_1 <- data.frame(treatment, response)
ggplot() +
  geom_line(data = df_1, mapping = aes(x = treatment, y = response), color = "red")
```



(b)

Compare to the results from Exercise 4.1.

```
ggplot() +
  geom_line(data = df_1, mapping = aes(x = treatment, y = response), color = "red")+
  geom_abline(intercept = 0.4, slope = 0.1, color = "blue")
```



almost the same

13.11 Building a logistic regression model

The folder Rodents contains data on rodents in a sample of New York City apartments.

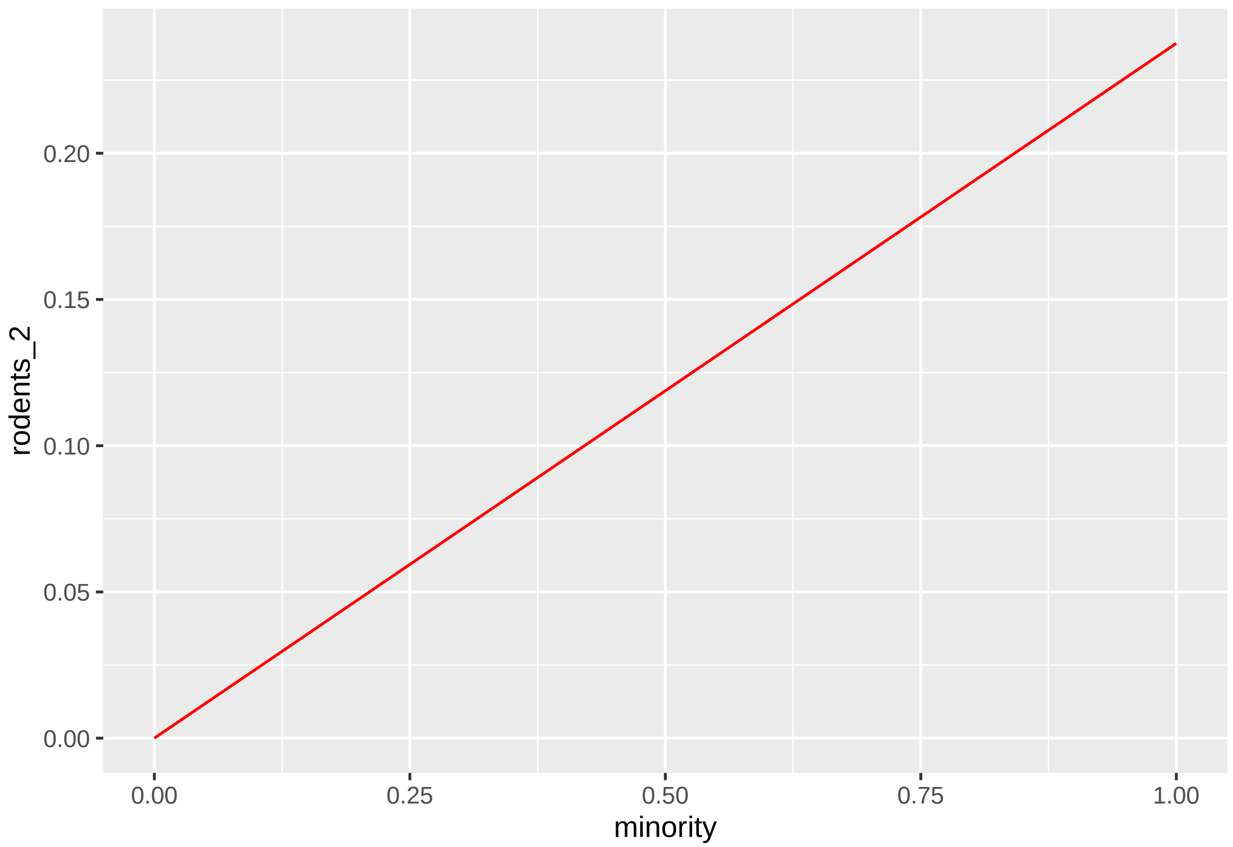
(a)

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("https://github.com/avehtari/ROS-Examples/raw/master/Rodents/rodents.dat", header
# sample_n(Rodents, 100)
# rodents_1 <-Rodents[-which(is.na(Rodents$rodent2)), ]
Rodents$minority <- 1
for (i in 1:length(Rodents)){
  if(Rodents$race[i] ==1){
    Rodents$minority[i] <- 0
  }
}

fit_1 <- glm(rodent2 ~ minority, data = Rodents, family = binomial)
# Householder race/ethnicity (1=White (non-hispanic)
# 2=Black (non-hispanic)
# 3=Puerto Rican
# 4=Other Hispanic
# 5=Asian/Pacific Islander
# 6=Amer-Indian/Native Alaskan
# 7=Two or more races)

print(fit_1)
```

```
##
## Call:  glm(formula = rodent2 ~ minority, family = binomial, data = Rodents)
##
## Coefficients:
## (Intercept)     minority
##      -13.57        12.40
##
## Degrees of Freedom: 1550 Total (i.e. Null);  1549 Residual
##   (197 observations deleted due to missingness)
## Null Deviance:       1700
## Residual Deviance: 1699  AIC: 1703
```

```
b <- coef(fit_1)
ggplot() +
  geom_line(data = data.frame(rodents_2 = invlogit(b[1] + b[2]*seq(0, 1)), minority = seq(0, 1)), mappin
```



(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_2 <- glm(rodent2 ~ housewgt + housing + unitflr2 + minority, data = Rodents, family = binomial)
summary(fit_2)
```

```
##
## Call:
## glm(formula = rodent2 ~ housewgt + housing + unitflr2 + minority,
##     family = binomial, data = Rodents)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1367  -0.7746  -0.6599  -0.4758   2.2726
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.269480 374.933103  -0.030    0.976
## housewgt     -0.006758   0.001407  -4.803 1.56e-06 ***
## housing      -0.358112   0.070660  -5.068 4.02e-07 ***
## unitflr2     -0.012365   0.031238  -0.396    0.692
## minority     12.383962 374.932926   0.033    0.974
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1686.6  on 1539  degrees of freedom
## Residual deviance: 1641.7  on 1535  degrees of freedom
##   (208 observations deleted due to missingness)
## AIC: 1651.7
##
## Number of Fisher Scoring iterations: 12
```

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.

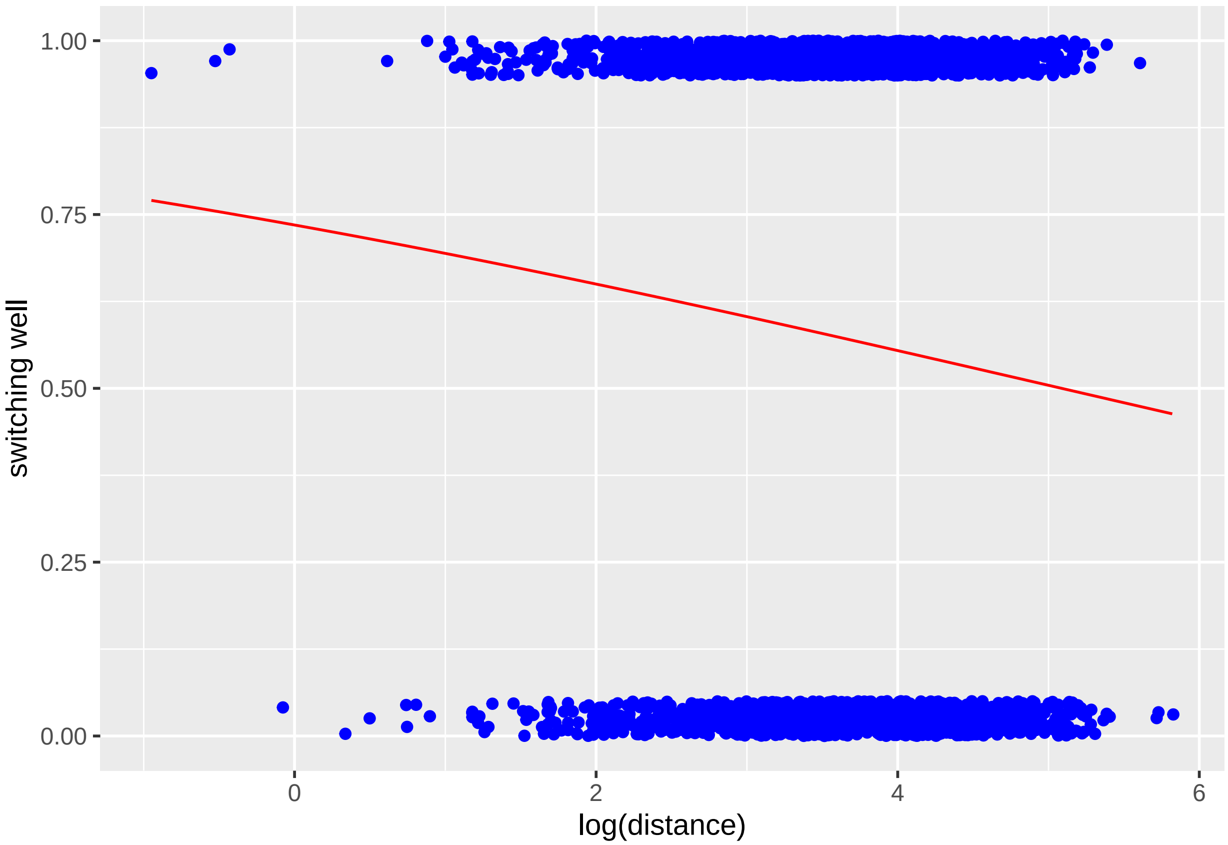
```
wells <- read.csv("https://github.com/avehtari/ROS-Examples/raw/master/Arsenic/data/wells.csv", header
wells_1 <- wells %>%
  data.frame(log_dist = log(wells$dist))
fit_1 <- glm(switch ~ log_dist, data = wells_1, family = binomial)
summary(fit_1)
```

```
##
## Call:
## glm(formula = switch ~ log_dist, family = binomial, data = wells_1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6365  -1.2795   0.9785   1.0616   1.2220
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.01971    0.16314   6.251 4.09e-10 ***
## log_dist    -0.20044    0.04428  -4.526 6.00e-06 ***
## ---
## 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: 4097.3  on 3018  degrees of freedom
## AIC: 4101.3
##
## Number of Fisher Scoring iterations: 4
```

(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.

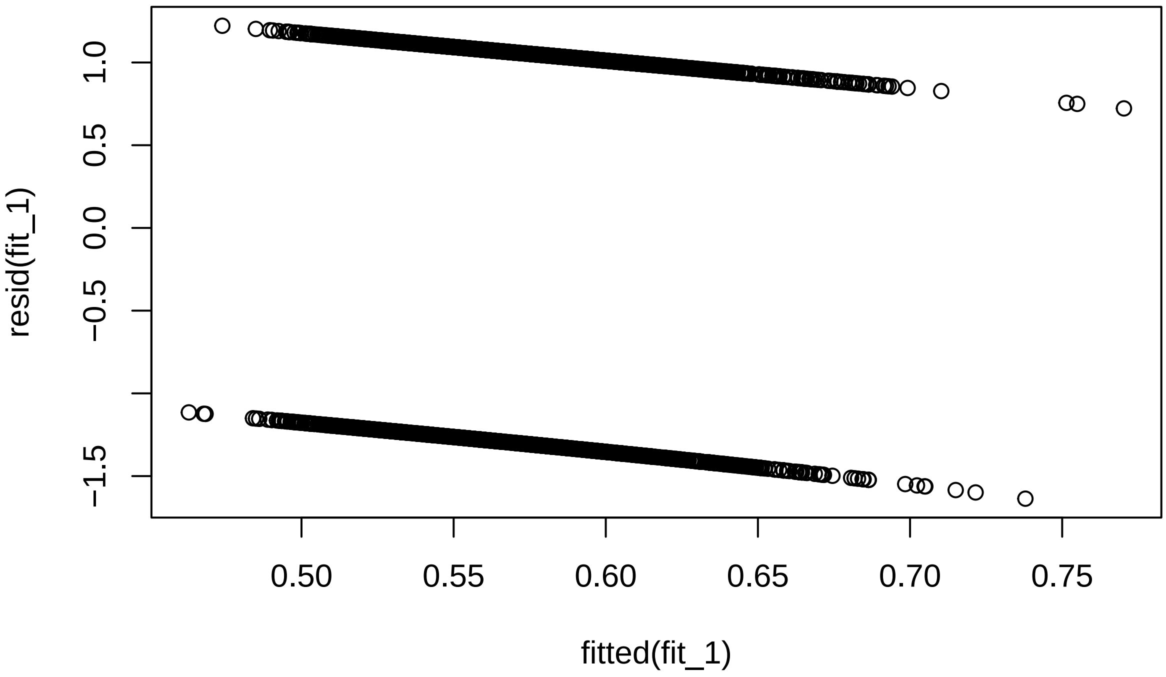
```
jitter_binary <- function(a, jitt=0.05){
  ifelse(a==0, runif(length(a), 0, jitt), runif(length(a), 1 - jitt, 1))
}

wells_1$switch_1 <- jitter_binary(wells_1$switch)
b <- coef(fit_1)

df_1 <-  data.frame(y = invlogit(b[1] + b[2]*seq(min(wells_1$log_dist), max(wells_1$log_dist), 0.01)), 

ggplot() +
  geom_point(data = wells_1, mapping = aes(x = log_dist, y = switch_1), color = "blue") +
  geom_line(data = df_1, mapping = aes(x = x, y = y), color = "red") +
  xlab("log(distance)") + ylab("switching well")
```



(c)

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

```
plot(fitted(fit_1), resid(fit_1))
```



```
library(arm)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:rosdata':
##
##     newcomb

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
## Loading required package: lme4
```

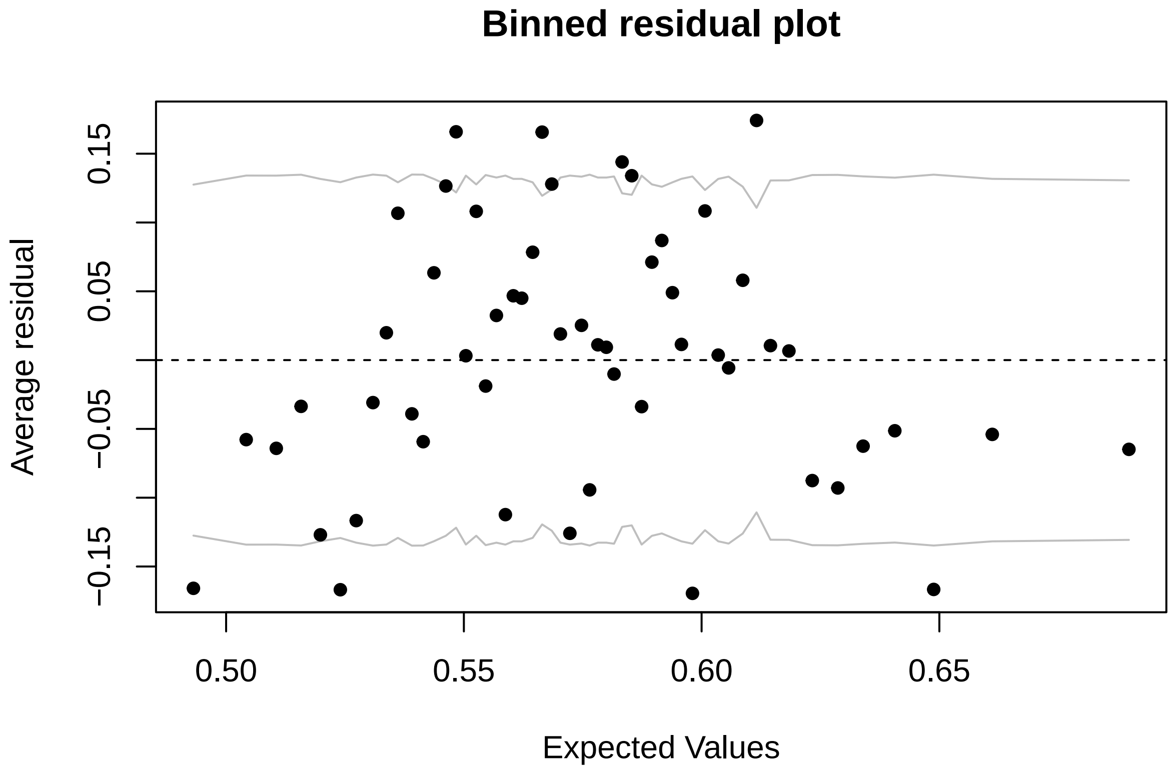
```
##
## arm (Version 1.13-1, built: 2022-8-25)
```

```
## Working directory is C:/Users/x1245/OneDrive/Documents/MA678
```

```
##
## Attaching package: 'arm'
```

```
## The following objects are masked from 'package:rstanarm':
##
##     invlogit, logit
```

```r
# x <- The expected values from the logistic regression.
# y <- The residuals values from logistic regression (observed values minus expected values).
pred <- predict(fit_1, newdata = wells_1, type = 'response')
resid <- wells_1$switch - pred
binnedplot(pred, resid)
```



(d)

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

```
error <- mean((pred>.5 & wells_1$switch == 0) | (pred < .5 & wells_1$switch == 1))
cat("the error rate of the fitted model = ", error, fill = TRUE)
```

```
## the error rate of the fitted model =  0.4192053
```

```
null_p <- mean(wells_1$switch)
error_null <- mean((null_p > .5 & wells_1$switch == 0) | (null_p < .5 & wells_1$switch == 1))
cat("the error rate of the null model = ", error_null, fill = TRUE)
```

```
## the error rate of the null model =  0.4248344
```

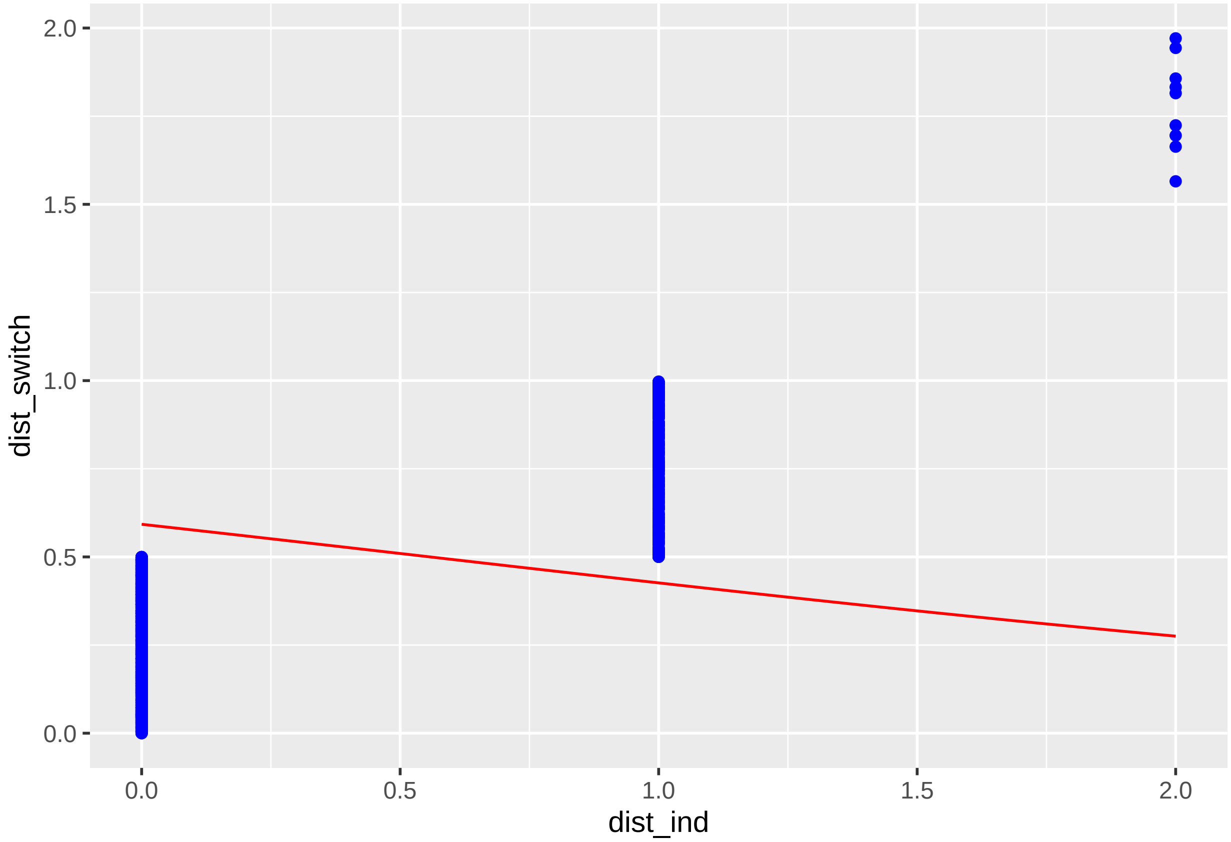
(e)

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_1$dist_ind <- rep(NA, length(wells_1$switch))
for (i in 1:dim(wells_1)[1]) {
  if(wells_1$dist[i]<100){
    wells_1$dist_ind[i] = 0
  }
  if(wells_1$dist[i]>100 & wells_1$dist[i]<200){
    wells_1$dist_ind[i] = 1
  }
    if(wells_1$dist[i]>200){
    wells_1$dist_ind[i] = 2
  }
}

fit_2 <- glm(switch ~ dist_ind, data = wells_1, family = 'binomial')
summary(fit_2)
```

```
##
## Call:
## glm(formula = switch ~ dist_ind, family = "binomial", data = wells_1)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.340  -1.340   1.023   1.023   1.606
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.37435    0.03902   9.593  < 2e-16 ***
## dist_ind    -0.67120    0.11781  -5.697 1.22e-08 ***
## ---
## 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: 4084.8  on 3018  degrees of freedom
## AIC: 4088.8
##
## Number of Fisher Scoring iterations: 4
```

```
b <- coef(fit_2)
jitter_1 <- function(x, a=0.5){
  y = rep(NA, length(x))
  for (i in 1:length(x)) {
    if(x[i]==0){
      y[i] = runif(1, 0, a)
    }
    if(x[i]==1){
      y[i] = runif(1, 1 - a, 1)
    }
    if(x[i]==2){
      y[i] = runif(1, 2 - a, 2)
    }
  }
  return(y)
}

wells_1$dist_switch <- jitter_1(wells_1$dist_ind)


ggplot() +
  geom_point(data = wells_1, mapping = aes(x = dist_ind, y = dist_switch), color = "blue") +
  geom_line(data = data.frame(y = invlogit(b[1] + b[2]*seq(min(wells_1$dist_ind), max(wells_1$dist_ind)
```



## 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_2 <- data.frame(wells, log_arsenic = log(wells$arsenic))
wells_2$switch_jitter = jitter_1(wells_2$switch)
fit_1 <- glm(switch ~ dist + log_arsenic + dist:log_arsenic, data = wells_2, family = binomial)
summary(fit_1)
```

```
##
## Call:
## glm(formula = switch ~ dist + log_arsenic + dist:log_arsenic,
##     family = binomial, data = wells_2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1814  -1.1642   0.7468   1.0470   1.8383
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.491350   0.068119   7.213 5.47e-13 ***
## dist             -0.008735   0.001342  -6.510 7.52e-11 ***
## log_arsenic       0.983414   0.109694   8.965  < 2e-16 ***
## dist:log_arsenic -0.002309   0.001826  -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.

(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_2 <- stan_glm(switch ~ dist + arsenic, data = wells, family = binomial, refresh = 0)
print(fit_2)
```

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

```
b <- coef(fit_2)
dist0 <- 0
dist100 <- 100
dist200 <- 200
ars.5 <- .5
ars1 <- 1
ars2 <- 2
delta_1 <- invlogit(b[1] + b[2]*dist100 + b[3]*wells$arsenic) - invlogit(b[1] + b[2]*dist0 + b[3]*wells$
round(mean(delta_1), 2)
```

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

```
delta_2 <- invlogit(b[1] + b[2]*dist200 + b[3]*wells$arsenic) - invlogit(b[1] + b[2]*dist100 + b[3]*well
round(mean(delta_2), 2)
```

```
## [1] -0.19
```

```
delta_3 <- invlogit(b[1] + b[2]*wells$dist + b[3]*ars1) - invlogit(b[1] + b[2]*wells$dist + b[3]*ars.5)
round(mean(delta_3), 2)
```

```
## [1] 0.06
```

```
delta_4 <- invlogit(b[1] + b[2]*wells$dist + b[3]*ars2) - invlogit(b[1] + b[2]*wells$dist + b[3]*ars1)
round(mean(delta_4), 2)
```

```
## [1] 0.11
```