MA678 Homework 4

Wendy Liang

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!

```
library(arm)
library(rstanarm)
library(foreign)
library(ggplot2)
```

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.

1 unit more in arsenic concentration corresponds to an approximately 11% positive difference in Pr(switching).

the logistic regression coefficients corresponding to 1-standard-deviation differences are d 0.46 $*$ 1.10 = 0.51 for arsenic level

a difference of 1 standard deviation in arsenic level corresponds to an expected 0.51/4 or approximately 13% positive difference in Pr(switch).

```
estimate=0.5*(0.46/4)
se=0.04/4
lower95=estimate-se*2
upper95=estimate+se*2
ci95=data.frame(lower95,upper95)

lower50=estimate-se
upper50=estimate+se
ci50=data.frame(lower50,upper50)

print(c(estimate=estimate,se=se))
```

```
## estimate       se
##   0.0575   0.0100
```

```
print(cbind(ci95,ci50))
```

```
##   lower95 upper95 lower50 upper50
## 1  0.0375  0.0775  0.0475  0.0675
```

(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.

```
Pr1=plogis(-0.9*50+0.46*0.5)
Pr2=plogis(-0.9*50+0.46*1)

estimate=Pr1-Pr2
se=0.04

lower95=estimate-se*2
upper95=estimate+se*2
ci95=data.frame(lower95,upper95)

lower50=estimate-se
upper50=estimate+se
ci50=data.frame(lower50,upper50)

print(c(estimate=estimate,se=se))
```

```
##      estimate            se
## -9.316753e-21  4.000000e-02
```

```
print(cbind(ci95,ci50))
```

```
##   lower95 upper95 lower50 upper50
## 1   -0.08    0.08   -0.04    0.04
```

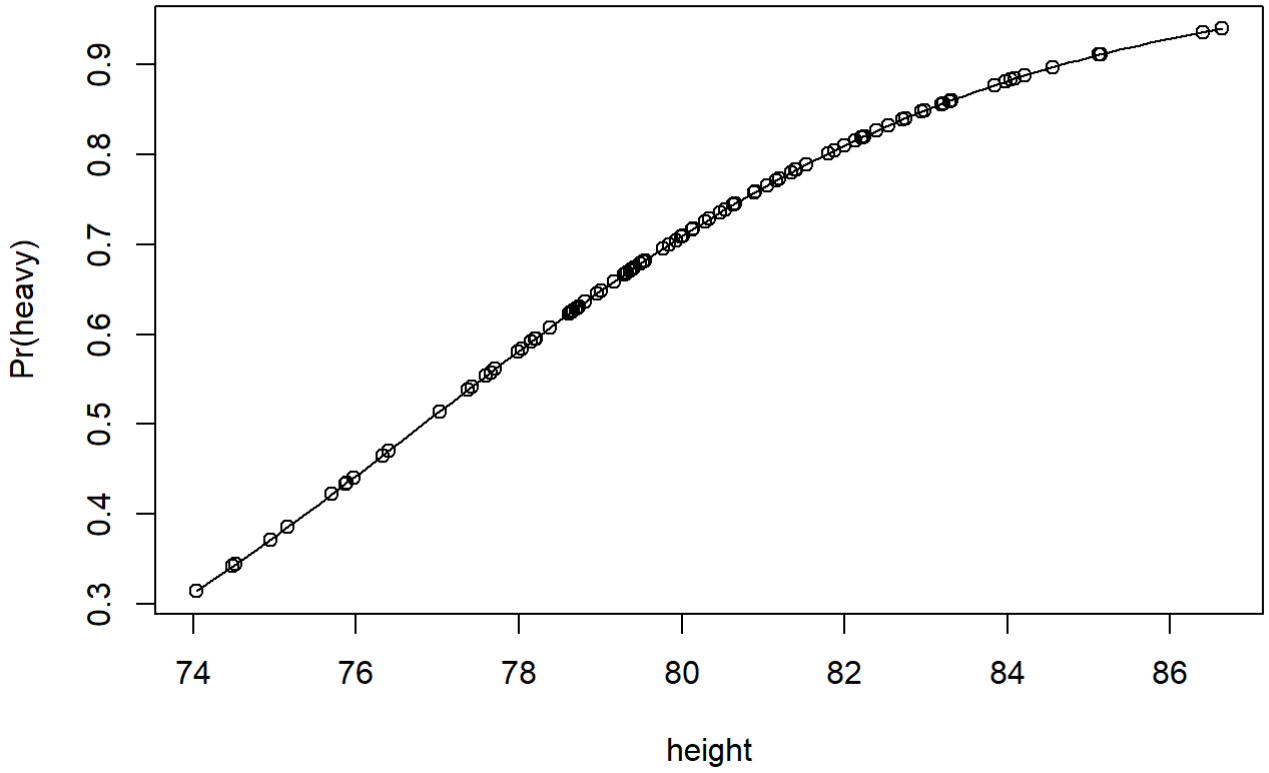
13.7

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.

```
invlogit=plogis
height=rnorm(100,80,2.5)
Pr_heavy=invlogit(-21.51+0.28*height)
plot(x=height,y=Pr_heavy,ylab="Pr(heavy)")
curve(invlogit(-21.51+0.28*x),add=T)
```



(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.

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 inche = 2.54 centimeter

height1*30.84=height2

so Pr_heavy=invlogit(-21.51+0.28hight2/2.54)=invlogit(-21.51+0.11height2)

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.

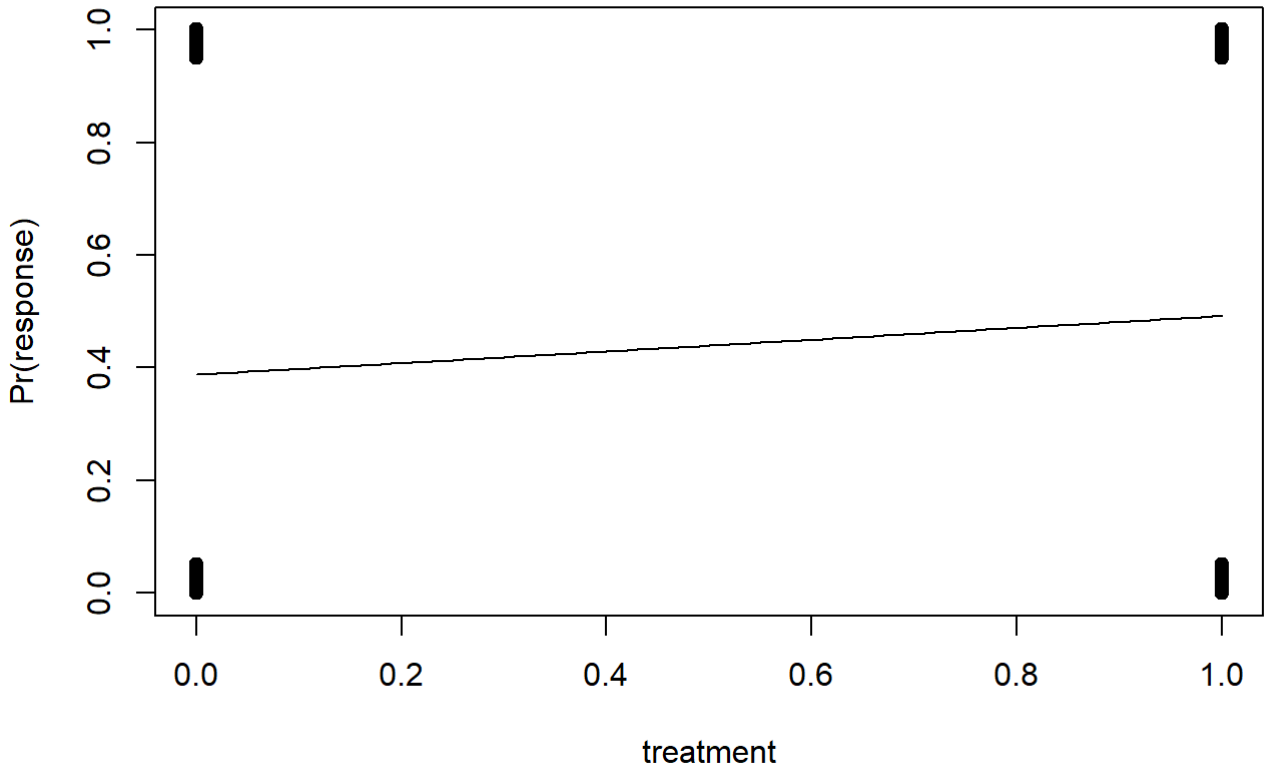
(a)

Set up these results as data in R. From these data, fit a logistic regression of response on the treatment indicator.

```
library(rstanarm)
set.seed(12)
y_treat=rbinom(500,1,0.5)
x_treat=rep(1,500)
treat=data.frame(x=x_treat,y=y_treat)
y_control=rbinom(500,1,0.4)
x_control=rep(0,500)
control=data.frame(x=x_control,y=y_control)
mydata=rbind(treat,control)
fit=glm(formula = y ~ x, family=binomial(link="logit"), data=mydata)
summary(fit)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial(link = "logit"), data = mydata)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.164  -1.164  -0.991   1.191   1.376
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.45573    0.09177  -4.966 6.85e-07 ***
## x            0.42372    0.12816   3.306 0.000946 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1371.9  on 999  degrees of freedom
## Residual deviance: 1360.9  on 998  degrees of freedom
## AIC: 1364.9
##
## Number of Fisher Scoring iterations: 4
```

```
jitter_binary <- function(a, jitt=0.05){
ifelse(a==0, runif(length(a), 0, jitt), runif(length(a), 1 - jitt, 1))
}
mydata$y_jitter <- jitter_binary(mydata$y)
plot(mydata$x, mydata$y_jitter,xlab="treatment",ylab = "Pr(response)")
curve(invlogit(coef(fit)[1] + coef(fit)[2]*x), add=TRUE)
```



(b)

Compare to the results from Exercise 4.1.

Pr(response)=invlogit(-0.46+0.42treatment), it means the average treatment effect is 0.42, and standard error of the average treatment effect is 0.13.

In the 4.1 the estimate of the average treatment effect is 0.1, and standard error of the average treatment effect is 0.031.

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 dataset
rod=read.table("rodents.dat")
rod=data.frame(rod)

#process race
rod$race = factor(rod$race)
fit1=glm(rodent2~race,data=rod,family=binomial(link="logit"))
summary(fit1)
```

```
##
## Call:
## glm(formula = rodent2 ~ race, family = binomial(link = "logit"),
##     data = rod)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0909  -0.8563  -0.4579  -0.4579   2.1482
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.2025     0.1328 -16.591  < 2e-16 ***
## race2         1.3880     0.1711   8.114 4.89e-16 ***
## race3         1.6570     0.2204   7.517 5.62e-14 ***
## race4         1.9955     0.1891  10.553  < 2e-16 ***
## race5         0.8252     0.2497   3.305 0.000951 ***
## race6         0.5931     1.1035   0.537 0.590951
## race7         0.4107     1.0883   0.377 0.705852
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.6  on 1550  degrees of freedom
## Residual deviance: 1548.8  on 1544  degrees of freedom
##   (197 observations deleted due to missingness)
## AIC: 1562.8
##
## Number of Fisher Scoring iterations: 4
```

- Intercept: an apartment where race1 people live, situated in an area with average other race population, has probability \(logit^{-1}(-2.62) = 0.0679 = 6.79%\) of having rodent infestation in the building
- race coefficients: this is the coefficient for race (on the logit scale) when any other predictor is at its average value. The base level for this factor is race1.In particular, if anything else is at the average point, apartments where race3 (\(\frac{1.6570}{4} = 0.4143 = 41.43%\) more likely) and race4 (\(\frac{1.9955}{4} = 0.4989 = 49.89%\) more likely) live have a higher chance to be in building infestated by roditors

(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.

```
rod$borough = factor(rod$borough)
rod$old = as.factor(rod$old)
rod$housing=factor(rod$housing)
rod$hispanic_Mean10 =rod$hispanic_Mean*10
rod$black_Mean10=rod$black_Mean*10

fit2 = glm(rodent2 ~ race + hispanic_Mean10 + black_Mean10 + borough + old + housing + personrm
+ struct + foreign, data=rod, family=binomial(link="logit"))
summary(fit2)
```

```
##
## Call:
## glm(formula = rodent2 ~ race + hispanic_Mean10 + black_Mean10 +
##     borough + old + housing + personrm + struct + foreign, family = binomial(link = "logi
t"),
##     data = rod)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9177  -0.6863  -0.4399  -0.1507   2.7787
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.34218    0.39717  -5.897 3.70e-09 ***
## race2            0.87633    0.23855   3.674 0.000239 ***
## race3            1.03046    0.27040   3.811 0.000138 ***
## race4            1.13803    0.25115   4.531 5.86e-06 ***
## race5            0.55575    0.29590   1.878 0.060364 .
## race6            0.09678    1.17070   0.083 0.934114
## race7           -0.65166    1.11763  -0.583 0.559848
## hispanic_Mean10  0.12661    0.04721   2.682 0.007326 **
## black_Mean10     0.01167    0.03442   0.339 0.734470
## borough2         0.39215    0.23375   1.678 0.093416 .
## borough3         0.13813    0.23300   0.593 0.553296
## borough4        -0.39510    0.25782  -1.532 0.125405
## borough5        -1.15237    0.64769  -1.779 0.075208 .
## old1             0.55873    0.16629   3.360 0.000779 ***
## housing2         0.26082    0.28483   0.916 0.359828
## housing3        -0.31002    0.31390  -0.988 0.323338
## housing4        -0.02701    0.28720  -0.094 0.925067
## personrm         0.51678    0.15675   3.297 0.000978 ***
## struct          -0.96078    0.14734  -6.521 7.00e-11 ***
## foreign          0.05550    0.16063   0.345 0.729729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1685.4  on 1524  degrees of freedom
## Residual deviance: 1366.4  on 1505  degrees of freedom
##   (223 observations deleted due to missingness)
## AIC: 1406.4
##
## Number of Fisher Scoring iterations: 6
```

- race: at the mean level of all other predictors, any non race1 has a higher probability to be associated with a building infestated by rodents. As on the previous model, race3 and race4 are more likely than other races to live in such conditions

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.

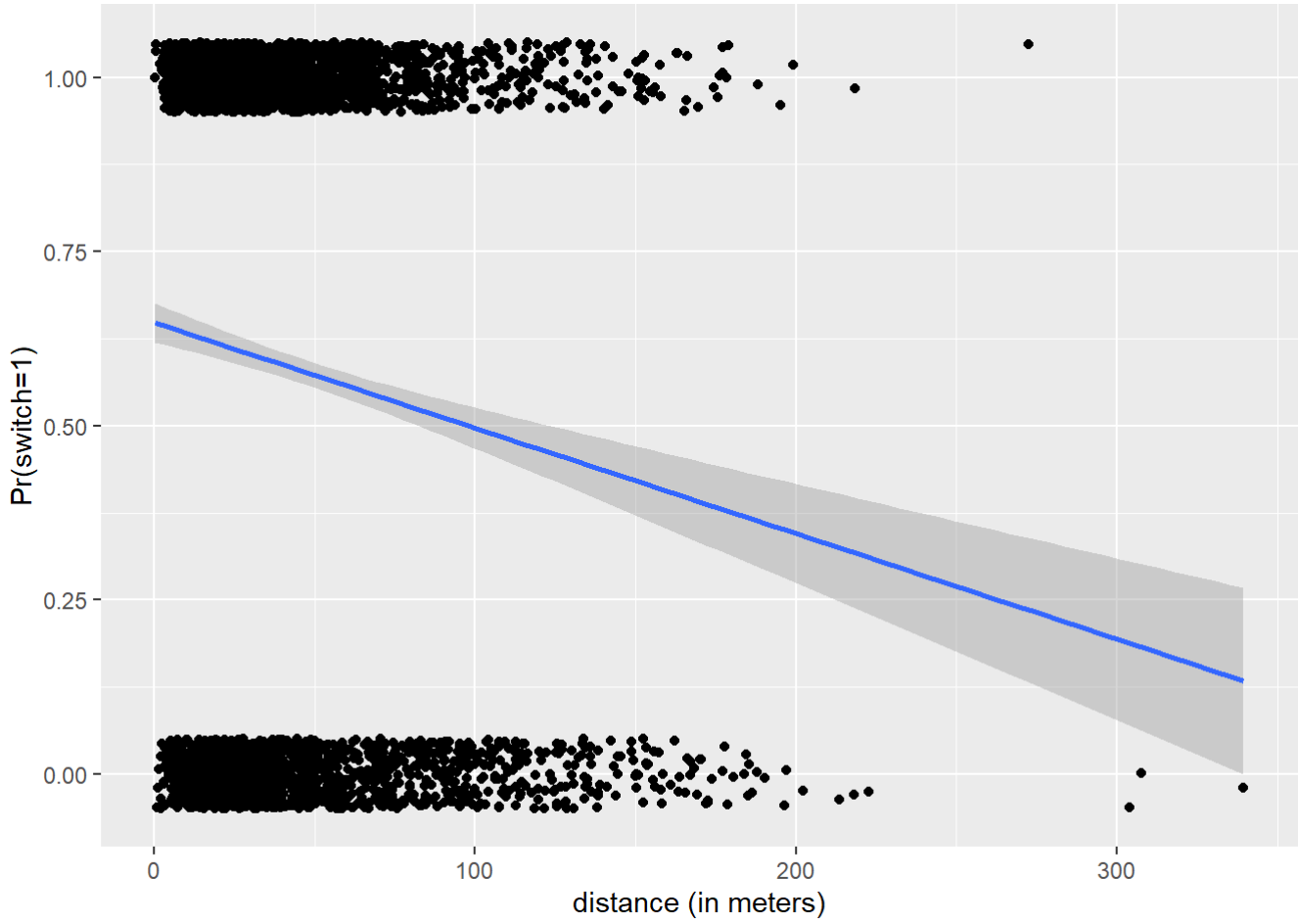
```
ars = read.table("http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat")

fit3 = glm(switch ~ dist, data=ars, family=binomial(link="logit"))
summary(fit3)
```

```
##
## Call:
## glm(formula = switch ~ dist, family = binomial(link = "logit"),
##     data = ars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4406  -1.3058   0.9669   1.0308   1.6603
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.6059594  0.0603102  10.047  < 2e-16 ***
## dist        -0.0062188  0.0009743  -6.383 1.74e-10 ***
## ---
## 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: 4076.2  on 3018  degrees of freedom
## AIC: 4080.2
##
## 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.

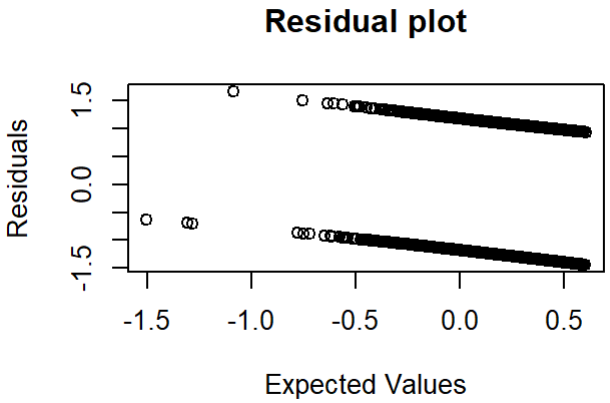
```
library(ggplot2)
ggplot(data=ars, aes(x=dist, y=switch)) + geom_jitter(position = position_jitter(height=.05))
 + geom_smooth(method="glm", family="binomial") + labs(x="distance (in meters)", y="Pr(switch=
1)")
```

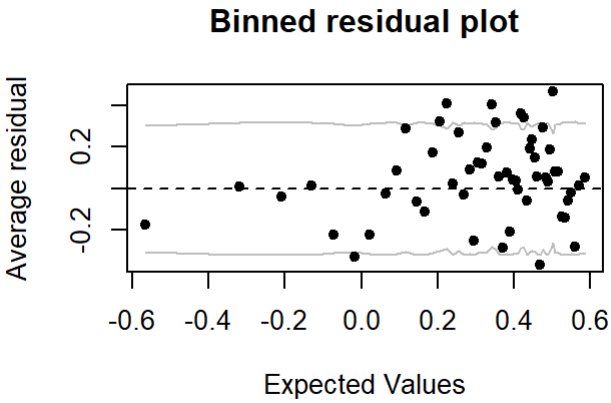


(c)

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

```
par(mfrow=c(2,2))
plot(predict(fit3),residuals(fit3), main="Residual plot", xlab="Expected Values", ylab="Residua
ls")
binnedplot(predict(fit3),residuals(fit3))
```





(d)

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

```
predicted = predict(fit3)

y = ars$switch

predicted.null = seq(0, 0, length.out=length(y))

print(c(fitted=mean((predicted>0.5 & y==0) | (predicted<0.5 & y==1)),null=mean((predicted.null>
0.5 & y==0) | (predicted.null<0.5 & y==1))))
```

```
##    fitted      null
## 0.5420530 0.5751656
```

(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.

```
ars$dist1 = as.numeric(ars$dist < 100)
ars$dist2 = as.numeric(100 <= ars$dist & ars$dist < 200)
ars$dist3 = as.numeric(ars$dist <= 200)

fit4 = glm(switch ~ dist1 + dist2 + dist3, data=ars, family=binomial(link="logit"))
summary(fit4)
```

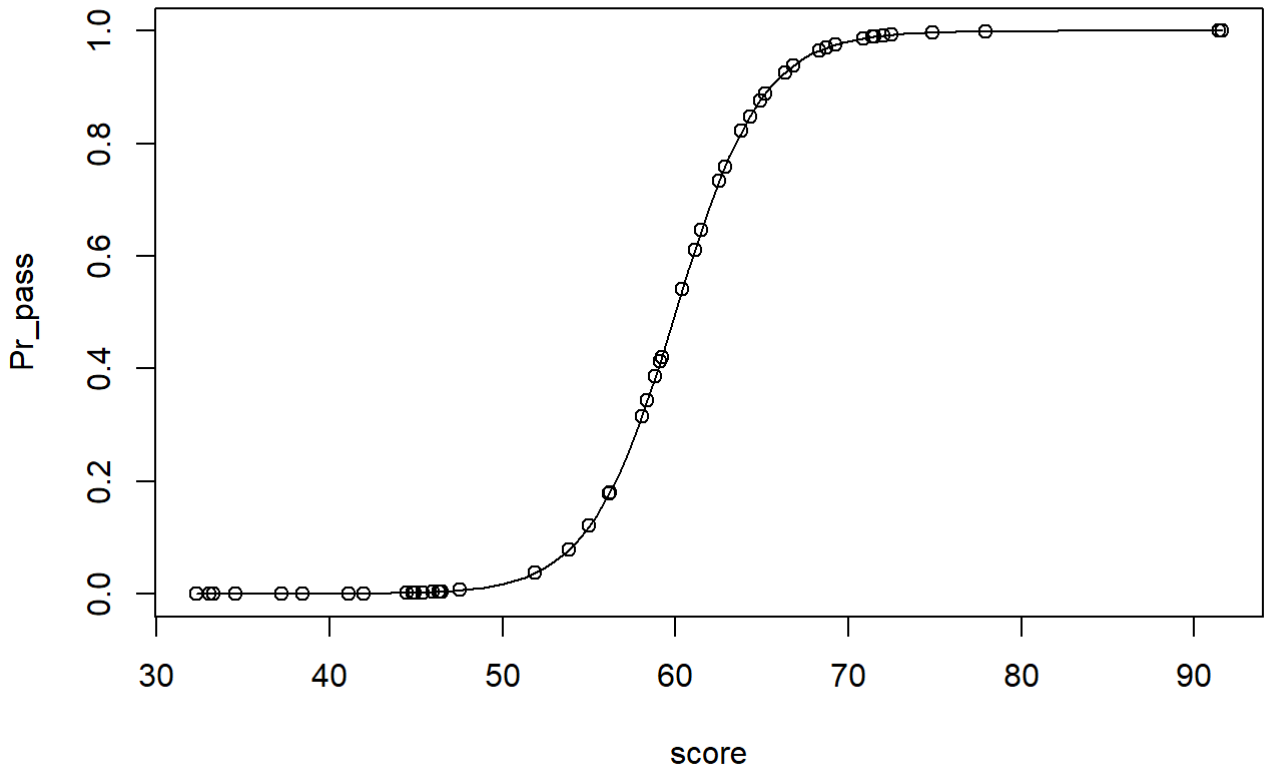
```
##
## Call:
## glm(formula = switch ~ dist1 + dist2 + dist3, family = binomial(link = "logit"),
##     data = ars)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.340  -1.340   1.023   1.023   1.734
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2528     0.8018  -1.562   0.1182
## dist1         1.6264     0.8027   2.026   0.0428 *
## dist2         0.9690     0.8103   1.196   0.2317
## dist3             NA         NA      NA       NA
## ---
## 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: 4084.7  on 3017  degrees of freedom
## AIC: 4090.7
##
## Number of Fisher Scoring iterations: 4
```

#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.

```
score=rnorm(50,60,15)
Pr_pass=invlogit(-24+0.4*score)
plot(x=score,y=Pr_pass)
curve(invlogit(-24 + 0.4*x),add=T)
```

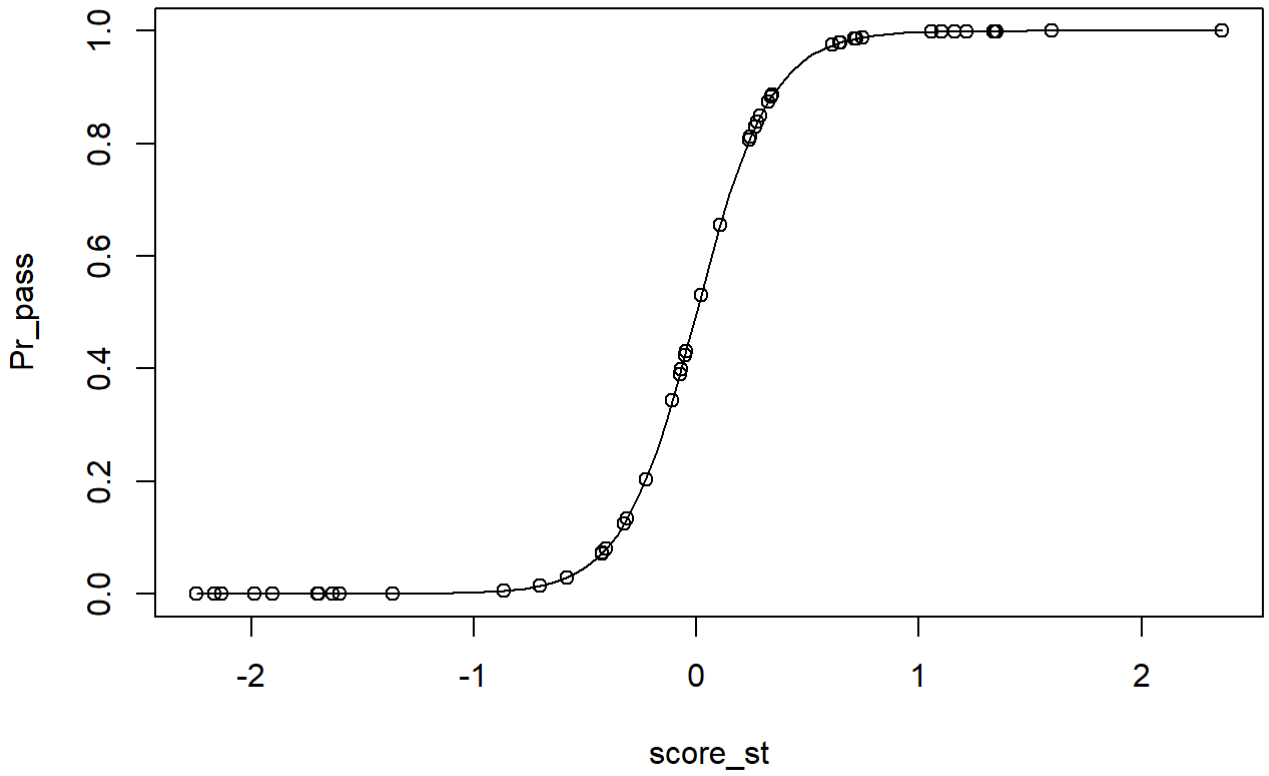


(b)

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?

the equation is Pr(pass) = logit-1(-24 + 0.4(15*x+60)) = logit-1(6x)

```
score=rnorm(50,60,15)
score_st=(score-60)/15
Pr_pass=invlogit(6*score_st)
plot(x=score_st,y=Pr_pass)
curve(invlogit(6*x),add=T)
```



(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?

```
score=rnorm(50,60,15)
Pr_pass=invlogit(-24+0.4*score)
newpred=rnorm(50,0,1)
Pr_pass=ifelse(Pr_pass>0.5,1,0)
fit5=stan_glm(Pr_pass~score+newpred,family=binomial(link="logit"),refresh=0)
loo(fit5)
```

```
##
## Computed from 4000 by 50 log-likelihood matrix
##
##          Estimate  SE
## elpd_loo     -5.9 1.3
## p_loo         1.0 0.5
## looic        11.7 2.7
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     38    76.0%   2360
##  (0.5, 0.7]   (ok)       11    22.0%   630
##    (0.7, 1]   (bad)       1     2.0%   3164
##    (1, Inf)   (very bad)  0     0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

#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.

```
ars$log.arsenic = log(ars$arsenic)
fit6 = glm(switch ~ dist * log.arsenic, family=binomial(link="logit"), data=ars)
summary(fit6)
```

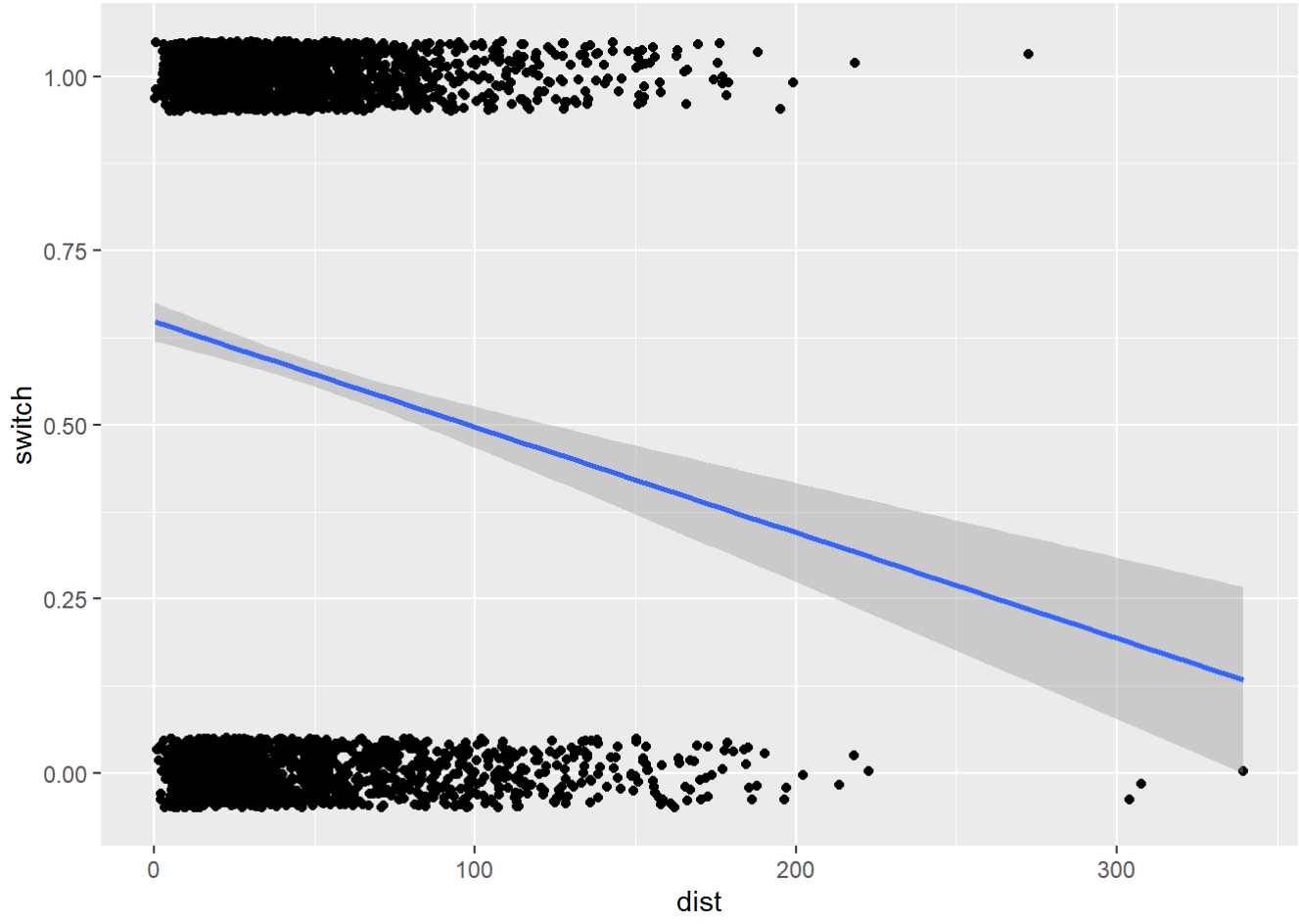
```
##
## Call:
## glm(formula = switch ~ dist * log.arsenic, family = binomial(link = "logit"),
##     data = ars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1814  -1.1642   0.7468   1.0470   1.8383
##
## Coefficients:
##                   Estimate Std. Error z value 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
```

- Intercept: a person with an average distance from a well with clean water and average log.arsenic has a \(logit^{-1}(0.63) = 65.25%\) probability to switch well
- dist: all other predictors hold at their mean, one meter increase in distance has the effect of decreasing the probability of switch well by \(\frac{-0.01}{4} = -0.25%\).
- log.arsenic: all other predictors hold at their mean, 1 unit increase in log.arsenic corresponds in a difference in the expected probability of switching well of \(logit^{-1}(0.91) = 71.30%\)
- dist:log.arsenic: it's not significant because p-value is 0.627

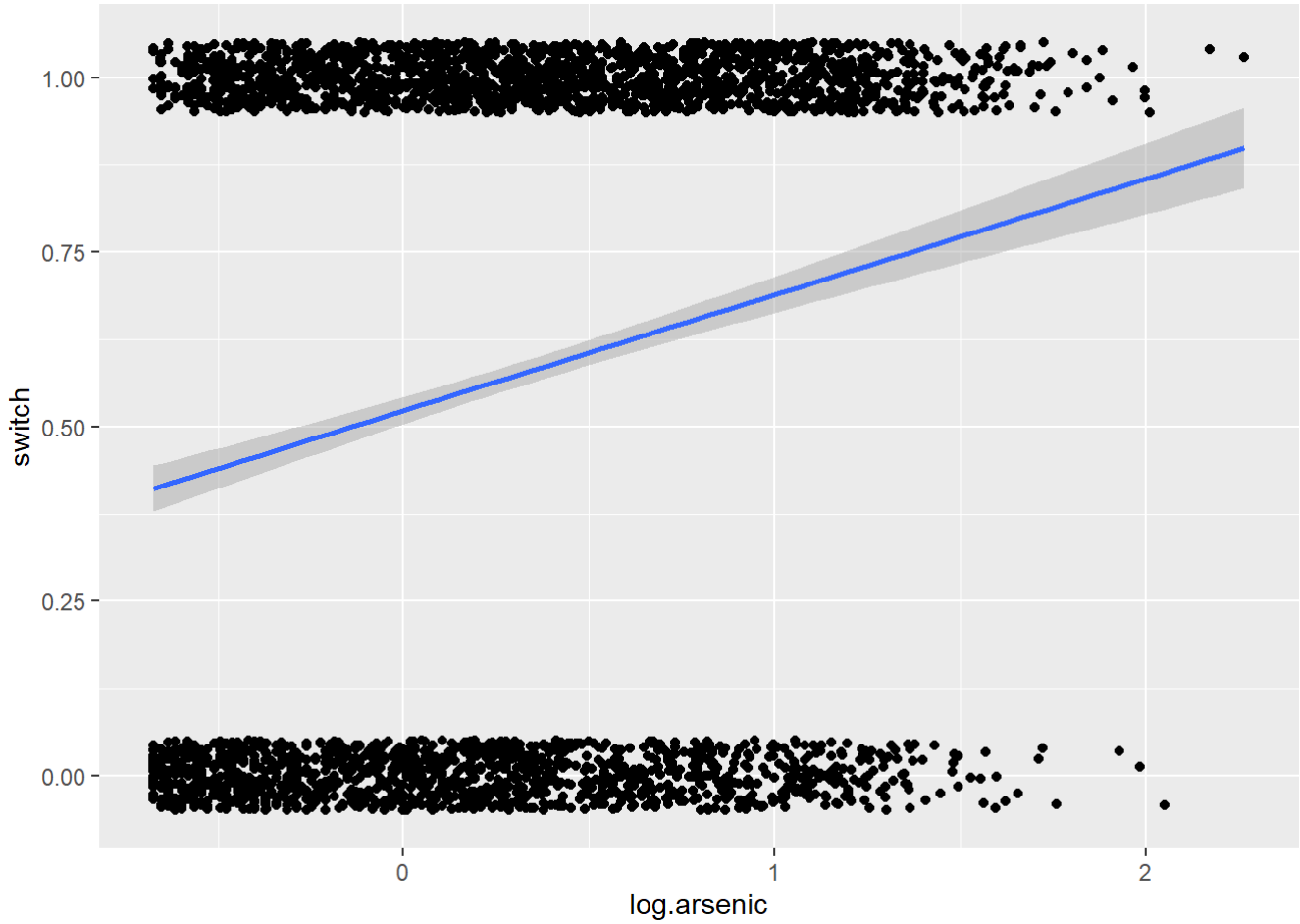
(b)

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

```
ggplot(data=ars, aes(x=dist, y=switch)) +
  geom_jitter(position=position_jitter(height=.05)) +
  geom_smooth(method="glm", family="binomial")
```




```
ggplot(data=ars, aes(x=log.arsenic, y=switch)) +
  geom_jitter(position=position_jitter(height=.05)) +
  geom_smooth(method="glm", family="binomial")
```



(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.

```
b = coef(fit6)
up = 100
low = 0
delta = invlogit(b[1] + b[2]*up + b[3]*ars$log.arsenic +
                   b[4]*ars$log.arsenic*up) -
  invlogit(b[1] + b[2]*low + b[3]*ars$log.arsenic + b[4]*ars$log.arsenic*low)
print(mean(delta))
```

```
## [1] -0.2113356
```

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

```
b = coef(fit6)
up = 200
low = 100
delta = invlogit(b[1] + b[2]*up + b[3]*ars$log.arsenic +
                   b[4]*ars$log.arsenic*up) -
  invlogit(b[1] + b[2]*low + b[3]*ars$log.arsenic + b[4]*ars$log.arsenic*low)
print(mean(delta))
```

```
## [1] -0.2090207
```

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

```
b = coef(fit6)
up = 1.0
low = 0.5
delta = invlogit(b[1] + b[2]*ars$dist + b[3]*up +
                   b[4]*ars$dist*up) -
  invlogit(b[1] + b[2]*ars$dist + b[3]*low + b[4]*ars$dist*low)
print(mean(delta))
```

```
## [1] 0.09195206
```

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

```
b = coef(fit6)
up = 2.0
low = 1.0
delta = invlogit(b[1] + b[2]*ars$dist + b[3]*up +
                   b[4]*ars$dist*up) -
  invlogit(b[1] + b[2]*ars$dist + b[3]*low + b[4]*ars$dist*low)
print(mean(delta))
```

```
## [1] 0.1353431
```

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(12)
x=runif(100,-4,0)
error=rnorm(100,0,1)
a=2
b=1
z=a+b*x+error
y=ifelse(z>0,1,0)

fit7=glm(y ~ x, family=binomial(link="logit"))
summary(fit7)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2070  -0.6396  -0.2114   0.5197   2.3028
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.2379     0.8507   4.982 6.31e-07 ***
## x             2.0390     0.3755   5.429 5.66e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.589  on 99  degrees of freedom
## Residual deviance:  78.553  on 98  degrees of freedom
## AIC: 82.553
##
## Number of Fisher Scoring iterations: 5
```

(b)

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

```
fit8=lm(y ~ x)
summary(fit8)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85592 -0.25975 -0.00738  0.20792  0.91074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.15097    0.07723  14.903  < 2e-16 ***
## x            0.31760    0.03281   9.681 5.95e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.361 on 98 degrees of freedom
## Multiple R-squared:  0.4889, Adjusted R-squared:  0.4836
## F-statistic: 93.73 on 1 and 98 DF,  p-value: 5.95e-16
```

(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).

```
#logistic regression at mean(x)=-2.08
diff1=abs(invlogit(4.24+2.04*(-3))-invlogit(4.24+2.04*(-2)))
diff1
```

```
## [1] 0.407526
```

```
#linear regression
diff2=0.32
```

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.

```
x=runif(100,0,100)
z=rbinom(100,1,0.5)
a=0.7
b=0
c=0
y=invlogit(a+b*x+c*z)
summary(lm(y~x+z))
```

```
##
## Call:
## lm(formula = y ~ x + z)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.895e-16 -1.806e-16 -1.000e-16 -1.440e-17  9.771e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  6.682e-01  2.105e-16  3.174e+15   <2e-16 ***
## x            1.742e-18  3.460e-18  5.030e-01    0.616
## z           -2.071e-16  2.019e-16 -1.026e+00    0.308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.003e-15 on 97 degrees of freedom
## Multiple R-squared:  0.4999, Adjusted R-squared:  0.4895
## F-statistic: 48.47 on 2 and 97 DF,  p-value: 2.547e-15
```

(b)

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

```
x=runif(100,0,100)
z=rbinom(100,1,0.5)
a=100
b=1
c=1
y=invlogit(a+b*x+c*z)
summary(lm(y~x+z))
```

```
##
## Call:
## lm(formula = y ~ x + z)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.844e-14  1.460e-17  1.695e-16  3.681e-16  5.551e-16
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 1.000e+00  4.232e-16 2.363e+15   <2e-16 ***
## x           3.645e-18  6.954e-18 5.240e-01    0.601
## z           3.730e-16  3.796e-16 9.830e-01    0.328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.894e-15 on 97 degrees of freedom
## Multiple R-squared:  0.4998, Adjusted R-squared:  0.4895
## F-statistic: 48.47 on 2 and 97 DF,  p-value: 2.55e-15
```

(c)

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

I don't know how to figure it out.