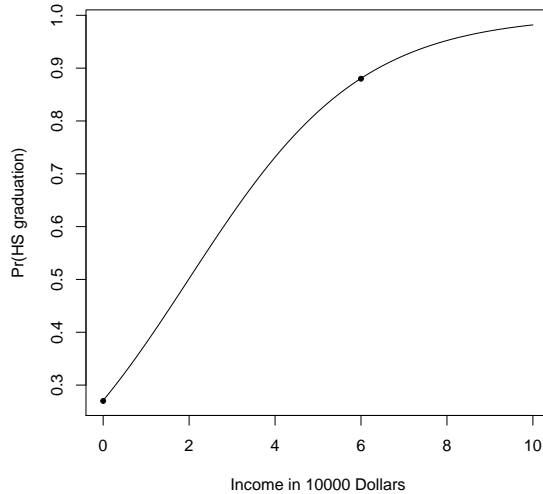


Stat 505 Assignment 6 Solutions

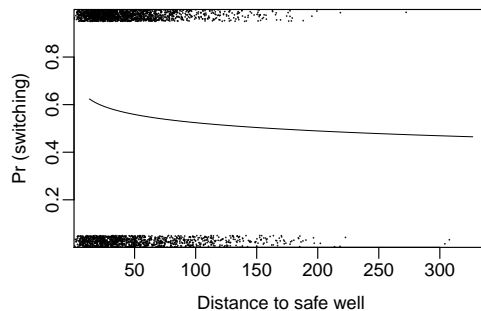
1. Exercise 5.3 p 105

In logit scale we are connecting the two points: $(0, -0.99)$ and $(6, 2.00)$ so the line has equation: $\hat{p} = -0.99 + 0.498x$ where x is income in 10000\$. Inverse logit of that line is plotted below with the two points.

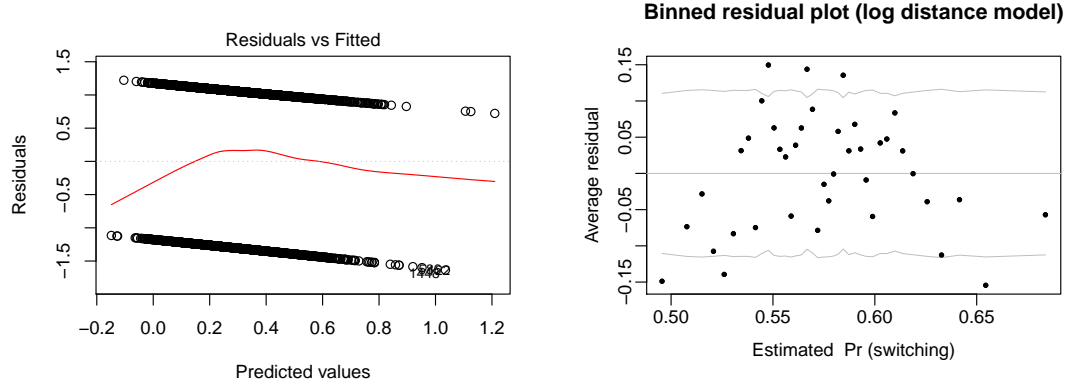


2. Exercise 5.9 p 106-7

- (a) Using $\log(\text{distance})$ to predict switching. The fitted model is: $\text{logit}(\mu_i) = 1.02 - 0.20 \log(\text{distance})$ with SE of the slope 0.04. When $\log(\text{dist}) = 0$ ($\text{dist} = 1\text{m}$), the predicted probability of switching is 0.73.
- (b) Plotting the above fit:



- (c) Residual plots

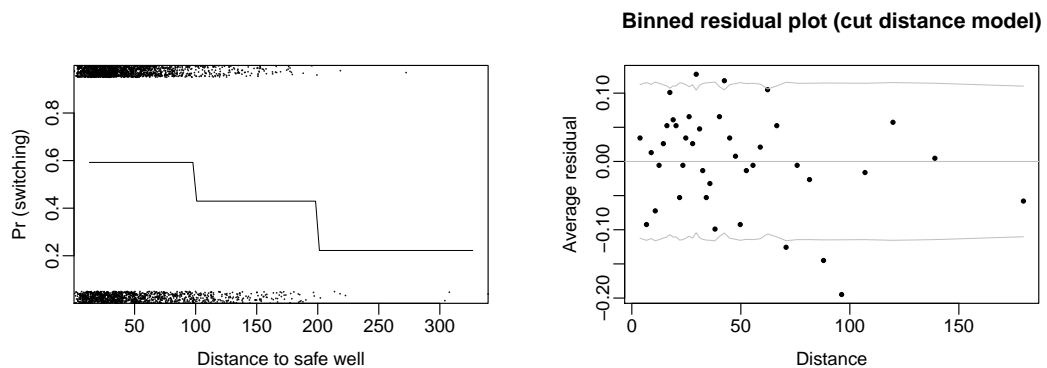


Based on the smoother in the residuals vs fits plot, I think the log distance model is overfitting the probability of switching when log distance is large (predicted response is small). The binned residuals plot agrees in that the first six bins have negative means, then we get a batch of positives.

- (d) Error rates of $\log(\text{dist})$ and null models

The error rate for the log distance model is 41.8%, which barely improves on the null model's 42.5%. Not impressive!

- (e) Build a model with distance as a class variable cut into intervals (0,100], (100,200], and (200,500).



This model provides three estimated probabilities: .59 for distances < 100m, .43 for distances in (100, 200)m, and .22 for distances > 200m. It does well except near 100m and has error rate of 40.9%, again barely better than the null model.

3. Exercise 5.10

- (a) Fit switching to distance and log arsenic and interaction.

```
glm(formula = switch ~ I(dist/100) * log(arsenic), family = binomial,
    data = wells)
```

	coef.est	coef.se
(Intercept)	0.49	0.07
I(dist/100)	-0.87	0.13

log(arsenic)	0.98	0.11
I(dist/100):log(arsenic)	-0.23	0.18

n = 3020, k = 4

residual deviance = 3896.8, null deviance = 4118.1 (difference = 221.3)

Interpretation:

Interpretations are complicated by the fact that we have an interaction in the model. We can't look at main effects by themselves.

The intercept estimate of 0.49 (SE=0.07) implies that when arsenic is 1 and distance is zero, there is a 0.62 probability of switching. (Not very meaningful)

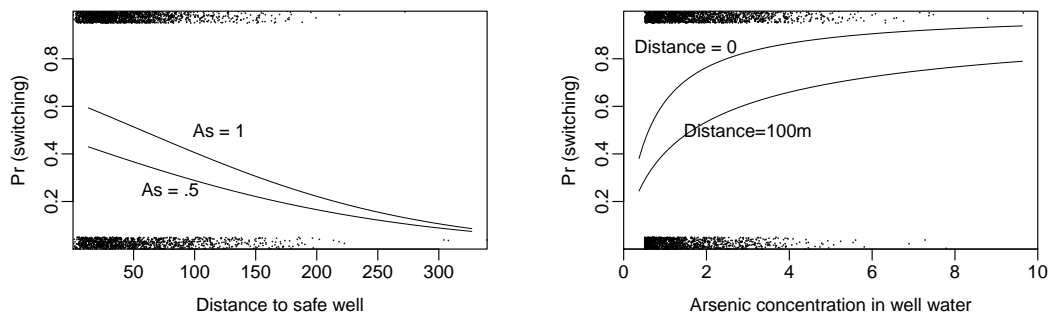
With an increase of 100m in distance and $\log(As = 0)$ (implies $As = 1$), the odds of switching decrease by a factor of $e^{-0.87} = .42$. (SE = 0.13). The further people are from a safe well, the less likely they are to switch.

When arsenic increases by a factor of e and distance is 0, odds of switching increase by a factor of $e^{0.98} = 2.67$ (SE = 0.10). That makes sense, as people are more likely to switch from a high arsenic well than from a moderate arsenic well.

The interaction coefficient of -0.231 (SE = 0.18) per 100m and in $\log(As)$ scale, is the only term which could be dropped from the model because it is less than 2 SE's from 0. It implies that the effect for $\log(As)$ decreases with distance (distance trumps safety?) or that the slope on distance is steeper (more negative) when arsenic is higher.

At the mean of the predictors ($\log(As) = 0.31$ and $dist = 48.3$) the coefficient with respect to distance is -0.94 with slope in probability scale of roughly -23.5%. In the other dimension, the log arsenic coefficient is 0.909 with slope in probability scale of roughly 22.7%.

(b) Graph as in Fig 5.12



(c) Average predicted differences.

- i. Going from distance= 0 to distance = 100m the average predicted probability decreases by -0.21 (SD = 0.018).
- ii. Going from distance = 100 to distance = 200m the average predicted probability decreases by -0.21 (SD = 0.050)

- iii. Going from arsenic = .5 to arsenic = 1 the average predicted probability increases by 0.15 (SD = 0.021)
- iv. Going from arsenic = 1 to arsenic = 2 the average predicted probability increases by 0.14 (SD = 0.013)

4. 5.11 What's wrong with the nes fit in 1964?

First show I get results similar to those in the book: Though our variables and output

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.11	0.22	0.48	0.63
female	0.22	0.14	1.61	0.11
black	-1.07	0.36	-2.93	0.00
income2. 17 to 33 percentile	-0.09	0.27	-0.32	0.75
income3. 34 to 67 percentile	-0.36	0.24	-1.48	0.14
income4. 68 to 95 percentile	-0.23	0.23	-1.00	0.32
income5. 96 to 100 percentile	0.93	0.39	2.39	0.02

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.80	0.20	-3.92	0.00
female	-0.07	0.14	-0.48	0.63
black	-16.85	419.79	-0.04	0.97
income2. 17 to 33 percentile	0.10	0.25	0.42	0.68
income3. 34 to 67 percentile	-0.13	0.23	-0.55	0.58
income4. 68 to 95 percentile	0.49	0.22	2.24	0.03
income5. 96 to 100 percentile	0.73	0.29	2.53	0.01

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.67	0.25	2.70	0.01
female	-0.03	0.15	-0.17	0.86
black	-3.68	0.60	-6.17	0.00
income2. 17 to 33 percentile	-0.34	0.29	-1.19	0.23
income3. 34 to 67 percentile	-0.30	0.26	-1.17	0.24
income4. 68 to 95 percentile	-0.38	0.26	-1.44	0.15
income5. 96 to 100 percentile	0.12	0.38	0.31	0.76

differ from the book's, our 1964 fit is odd in that the coefficient for black is far from 0 (-16.85) with huge standard error (420). This happens because no blacks in the sample voted for the Republican (Barry Goldwater), so the logistic regression is trying to fit two points, one of which is on the asymptote at one. It can't do that, and reports very large standard error.

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.91	0.17	5.34	0.00
female	-0.25	0.12	-2.20	0.03
black	-2.59	0.26	-9.82	0.00
income2. 17 to 33 percentile	-0.02	0.23	-0.07	0.94
income3. 34 to 67 percentile	-0.03	0.18	-0.16	0.87
income4. 68 to 95 percentile	0.05	0.18	0.27	0.79
income5. 96 to 100 percentile	0.78	0.30	2.57	0.01

	0. dk/na if voted/didn't vote for pres/i	1. democrat	2. republican
0	0	651	357
1	0	92	0

A fix is to use the bayesglm function provided by Gelman & Hill which does give reasonable estimates.

```
> display(bayesglm(presvote_2party ~ female + black + income, family = binomial,
+ data = nes, subset = year == 1964))
```

```
bayesglm(formula = presvote_2party ~ female + black + income,
family = binomial, data = nes, subset = year == 1964)
```

```
coef.est coef.se
(Intercept) -0.79 0.20
female -0.07 0.14
black -4.90 1.62
income2. 17 to 33 percentile 0.09 0.25
income3. 34 to 67 percentile -0.14 0.23
income4. 68 to 95 percentile 0.47 0.21
income5. 96 to 100 percentile 0.71 0.28
```

n = 1062, k = 7

residual deviance = 1246.4, null deviance = 1337.7 (difference = 91.3)

R Code

```
### R code from vignette source '/home/jimrc/classes/stat505/homework/assn6/assn6_sltnsF11.Rnw'
```

```
#####
```

```
### code chunk number 1: 5dot3
```

```
#####
```

```
require(arm)
```

```
curve(invlogit(-.99 + 0.498* x), from = 0, to = 10, xlab = "Income in 10000 Dollars", ylab = "Pr(HS grad",
points(c(0,6),c(.27,.88),pch=20)
```

```
#####
```

```

### code chunk number 2: readin
#####
require(arm)
wells <- read.table(".././data/ARM_Data/arsenic/wells.dat",head=T)

#####
### code chunk number 3: fitLogD
#####
wells.fitlogD <- glm( switch ~ log(dist), wells, family=binomial)
display(wells.fitlogD)

#####
### code chunk number 4: plotfitLogD
#####
binary.jitter <- function(a, jitt=.05){
  jitter <- runif (length(a), 0, jitt)
  a + (a==0)*jitter - (a==1)* jitter
}
par(mfrow=c(1,1))
plot(wells$dist,binary.jitter(wells$switch),
      xlab="Distance to safe well", ylab="Pr (switching)",
      xaxs="i", yaxs="i", mgp=c(2,.5,0), pch=20, cex=.1)
curve (invlogit(cbind(1, log(x)) %*% coef(wells.fitlogD)), lwd=.5, add=TRUE)

#####
### code chunk number 5: plotresidLogD
#####
par(mfrow=c(1,2))
plot(wells.fitlogD,which=1)
binned.resids <- function (x, y, nclass=sqrt(length(x))){
  shinglex <- co.intervals(x, number=nclass, overlap=0)
  break.x <- cut(x, c(shinglex[,1], shinglex[nrow(shinglex),2]))
  n <- tapply(x, break.x, length)
  twoSE <- 2*tapply(y, break.x, sd)/sqrt(n)
  output <- cbind( tapply(x, break.x, mean),
                  tapply(y, break.x, mean),
                  n,
                  shinglex[,1],
                  shinglex[,2],
                  twoSE )
  colnames(output) <- c ("xbar", "ybar", "n", "x.lo", "x.hi", "twoSE")
  output
}
binLogD <- binned.resids(fitted(wells.fitlogD),wells$switch-fitted(wells.fitlogD),40)
plot(binLogD[,1],binLogD[,2], pch=19, cex=.5,
      xlab="Estimated Pr (switching)", ylab="Average residual",
      type="p", main="Binned residual plot (log distance model)", mgp=c(2,.5,0))
abline (0,0, col="gray", lwd=.5)
lines (binLogD[,1], binLogD[,6], col="gray", lwd=.5)
lines (binLogD[,1], -binLogD[,6], col="gray", lwd=.5)
points (binLogD[,1], binLogD[,2], pch=19, cex=.5)

```

```
#####
### code chunk number 6: errorRate
#####
errorRate <- function(y,pred,c){
  sum( y != (pred >= c))/length(y) }
c( errorRate(wells$switch, fitted(wells.fitlogD) ,.5), errorRate(wells$switch, fitted(update(wells.fi

#####
### code chunk number 7: cutDist
#####
wells.fitcutD <- glm( switch ~ cut(wells$dist, c(0,100,200,500)), wells, family=binomial)
par(mfrow=c(1,2))
plot(wells$dist,binary.jitter(wells$switch),
  xlab="Distance to safe well", ylab="Pr (switching)",
  xaxs="i", yaxs="i", mgp=c(2,.5,0), pch=20, cex=.1)
curve (invlogit(cbind(1, (x>100 & x <=200), (x>200)) %*% coef(wells.fitcutD)), lwd=.5, add=TRUE)
bincutD <- binned.resids(wells$dist,wells$switch-fitted(wells.fitcutD),40)
plot(bincutD[,1],bincutD[,2], pch=19, cex=.5,
  xlab="Distance", ylab="Average residual",
  type="p", main="Binned residual plot (cut distance model)", mgp=c(2,.5,0))
abline (0,0, col="gray", lwd=.5)
lines (bincutD[,1], bincutD[,6], col="gray", lwd=.5)
lines (bincutD[,1], -bincutD[,6], col="gray", lwd=.5)

#####
### code chunk number 8: interactionmodel
#####
wells.intfit <- glm( switch ~ I(dist/100)*log(arsenic), wells, family=binomial)
display(wells.intfit)

#####
### code chunk number 9: interactionPlot
#####
par(mfrow=c(1,2))
plot(wells$dist,binary.jitter(wells$switch),
  xlab="Distance to safe well", ylab="Pr (switching)",
  xaxs="i", yaxs="i", mgp=c(2,.5,0), pch=20, cex=.1)
curve (invlogit(cbind(1,x/100,0,0 ) %*% coef(wells.intfit)), lwd=.5, add=TRUE)
curve (invlogit(cbind(1,x/100,log(.5),x*log(.5)/100 ) %*% coef(wells.intfit)), lwd=.5, add=TRUE)
text(c(80,120),c(.25,.5), c("As = .5","As = 1"))

plot(wells$arsenic,binary.jitter(wells$switch), xlim=c(0,10),
  xlab="Arsenic concentration in well water", ylab="Pr (switching)",
  xaxs="i", yaxs="i", mgp=c(2,.5,0), pch=20, cex=.1)
curve (invlogit(cbind(1,0,log(x),0 ) %*% coef(wells.intfit)), lwd=.5, add=TRUE)
curve (invlogit(cbind(1,1,log(x),1*log(x) ) %*% coef(wells.intfit)), lwd=.5, add=TRUE)
text(c(1.5,3),c(.85,.5), c("Distance = 0","Distance=100m"))
```

```

#####
### code chunk number 10: delta1
#####
delta <- with(wells, invlogit( cbind(1, 0, log(arsenic),0) %*% coef(wells.intfit)) - invlogit( cbind(1,
c(mean(delta), sd(delta))

#####
### code chunk number 11: delta2
#####
delta <- with(wells, invlogit( cbind(1, 100, log(arsenic),100*log(arsenic)) %*% coef(wells.intfit)) - i
c(mean(delta), sd(delta))

#####
### code chunk number 12: delta3
#####
delta <- with(wells, invlogit( cbind(1, dist, log(.5),log(.5)*dist) %*% coef(wells.intfit)) - invlogit(
c(mean(delta), sd(delta))

#####
### code chunk number 13: delta4
#####
delta <- with(wells, invlogit( cbind(1, dist, log(1),log(1)*dist) %*% coef(wells.intfit)) - invlogit(
c(mean(delta), sd(delta))

#####
### code chunk number 14: nes1
#####
#require(foreign);require(arm)
#nes<-read.dta("http://www.math.montana.edu/~jimrc/classes/stat505/data/nes5200_processed_voters_realid
for(yr in seq(1960,1972,4)){
print(xtable::xtable(summary( glm(presvote_2party ~ female + black + income, family = binomial, data =
  }

#####
### code chunk number 15: fix
#####
xtable::xtable( with(subset(nes, year==1964), table(black, presvote_2party)))

#####
### code chunk number 16: fix2
#####
display( bayesglm(presvote_2party ~ female + black + income, family = binomial, data = nes, subset = ye

```