Contents

Chapter 1

Introduction

Chapter 2

Basic Probability and Statistics

Problem 1

Think about what it means to scale this data. We have a set of data X and this data has a mean μ_x and standard deviation σ. We want to linearly transform the data which means:

$$\begin{aligned} Y &= a + bX \\ \mu_y &= a + b\mu_x \\ \sigma_y &= |b|\sigma_x \end{aligned}$$

Now to answer (a) and (b) we can just solve these equations. Let's solve for standard deviation.

$$\begin{aligned} 15 &= |b|10 \\ 1.5 &= |b| \end{aligned}$$

This means that $b = \pm 1.5$ and we can now solve for the mean to get that $a = 47.5$ or $a = 152.5$. We can transform it with either the first set of values or the second. You don't want to use the second because that will reverse the ordering in the data.

Problem 2

(a)

Doing this in R is easy:

```
girls <- c(.4777, .4875, .4874, .4859, .4754, .4864,.4813,.4787,.4895,
          .4797,.4876,.4859,.4857,.4907, .5010,.4903,.4860,.4911,.4871,
```

```
            .4725,.4822,.4870,.4823,.4973)

num_births <- 3903
std <- sd(girls)
avg <- mean(girls)
expected_std <- sqrt(avg * (1 - avg) / num_births)
```

We get that std= .0064 and expected_std= .008

Let's quickly prove this formula for the expected standard error of a proportion.

Proof. We are trying to prove that $SE(X) = \sqrt{\frac{p(1-p)}{n}}$ for binary variables x_i which take on only values 0 and 1.

Begin by assuming that we have a population with m instances where $x = 1$ and $n - m$ instances where $x = 0$ We know that the standard deviation of a population is:

$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

Let's just concern ourselves with the summation portion of this equation and see if we can massage it into the desirable form.

$$\begin{aligned}
\sum (x_i - \bar{x})^2 &= \sum (x_i - \bar{x})(x_i - \bar{x}) \\
&= \sum x_i^2 - 2x_i\bar{x} + \bar{x}^2 \\
&= \sum_{x_i=0} \bar{x}^2 + \sum_{x_i=1} 1 - 2\bar{x} + \bar{x}^2 \\
&= (n-m)\bar{x}^2 + m - 2m\bar{x} + m\bar{x}^2 \\
&= (n-m)\frac{m^2}{n^2} + m - 2m\frac{m}{n} + m\frac{m^2}{n^2} && \bar{x} = \frac{m}{n} \\
&= m + \frac{m^2}{n} - \frac{m^3}{n^2} - \frac{2m^2}{n} + \frac{m^3}{n^2} \\
&= m - \frac{m^2}{n} \\
&= m\left(1 - \frac{m}{n}\right) \\
&= np(1-p) && p = m/n
\end{aligned}$$

So we can finish by substituting this back into our original equation!

$$\begin{aligned}
\sigma &= \sqrt{\frac{np(1-p)}{n}} \\
&= \sqrt{p(1-p)}
\end{aligned}$$

Since the sample proportion is a mean the standard error is calculated like normal yielding

$$\begin{aligned} SE(X) &= \frac{\sigma_x}{\sqrt{n}} \\ &= \sqrt{\frac{p(1-p)}{n}} \end{aligned}$$

□

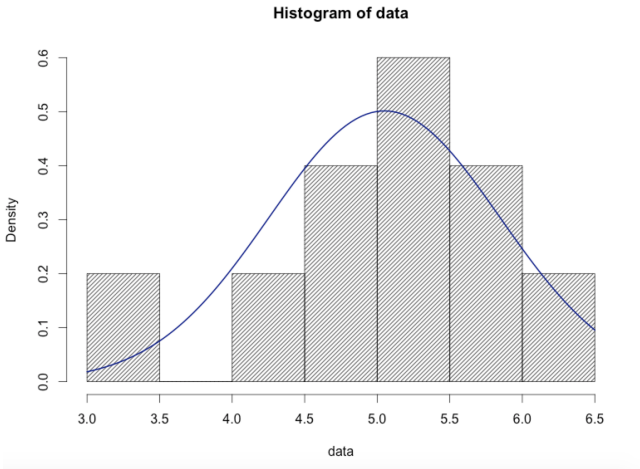
(b)

No they are not significant. If we run a χ^2 test we get $\chi^2 = .0019$ which is no where near what we need for a significant result.

Problem 3

```
cols <- 20
rows <- 1000
x <- replicate(cols, runif(rows))
data <- apply(a, 1, sum)
avg <- mean(data)
std <- sd(data)
hist(data, density=20)
curve(dnorm(x, mean=avg, sd=std),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

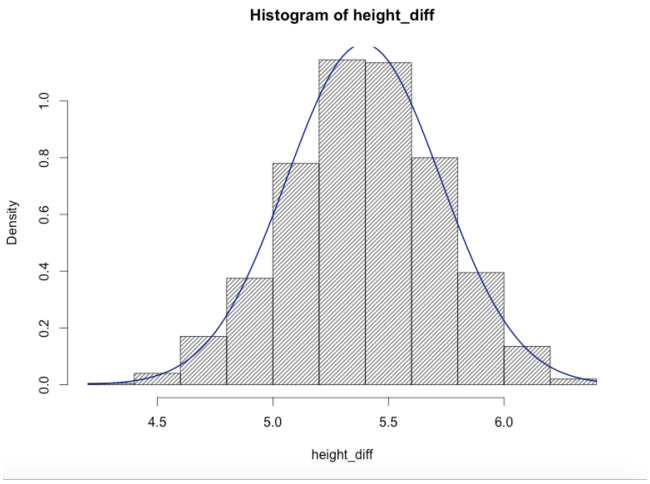
The plot is:



Problem 4

```
i = 1
height_diff = seq(1000) * 0
while (i <= 1000) {
  men_height <- rnorm(100, 69.1, 2)
  women_height <- rnorm(100, 63.7, 2.7)
  height_diff[i] = mean(men_height) - mean(women_height)
  i = i + 1
}

avg <- mean(height_diff)
std <- sd(height_diff)
hist(height_diff, density=20, freq=FALSE)
curve(dnorm(x, mean=avg, sd=std),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

The mean of the difference is 5.05 and the standard deviation is .79 The plot is:



Problem 5

Expectation is linear so:

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

As to standard deviation:

$$\begin{aligned} \mathrm{Var}[X+Y] &= \mathrm{Var}[X] + \mathrm{Var}[Y] + \mathrm{Cov}[X,Y] \\ &= \mathrm{Var}[X] + \mathrm{Var}[Y] + \mathrm{Corr}[X,Y]\sigma_x\sigma_y \end{aligned}$$

Chapter 3

Linear Regression: The Basics

Problem 1

(a)

```
library(arm)
setwd('~/Code/workspace/Gelman/')
data <- read.table('exercise2.1.txt', header=TRUE)
fit.1 <- lm(y ~ x1 + x2, data=data)
summary(fit.1)
```

(b)

```
beta.hat <- coef(fit.1)
beta.sim <- sim(fit.1)

par(mfrow=c(1, 2))
plot(data$x1, data$y)
apply(coef(sim(fit.1)), 1, function(beta) {curve(cbind(1, x, mean(data$x2)) %*% beta, add
curve(cbind(1, x, mean(data$x2)) %*% coef(fit.1), add=TRUE)

plot(data$x2, data$y)
apply(coef(sim(fit.1)), 1, function(beta) {curve(cbind(1, mean(data$x1), x) %*% beta, add
curve(cbind(1, mean(data$x1), x) %*% coef(fit.1), add=TRUE)
```

(c)

```
par(mfrow=c(1,2))
plot(data$x1[1:40], fit.1$residuals, xlab='X1', ylab='Residuals')
```

```
plot(data$x2[1:40], fit.1$residuals, xlab='X2', ylab='Residuals')

plot(predict(fit.1, data[1:40, ]), fit.1$residuals)
```

For X1 it appears that all of the assumptions are met. However, for X2 we can see some heteroscedasticity so we could potentially improve our model with some additional feature or a variable transformation.

d)

```
predictions <- data.frame(predict(fit.1, data[41:dim(data)[1], ], level=.95, interval="pr
predictions
```

I feel pretty good about these predictions. the standard error seems to be about 1, so the 95% confidence interval contains ± 2 of our estimate.

Problem 2

(a)

We want to find the coefficients of the following equation:

$$\log(y) = \alpha + \beta \log(h)$$

We are given $\beta = .8$ now we just need to solve for alpha and get $\alpha = 6.957$

To calculate the standard deviation of the residuals. Notice that our 95% confidence interval is within a factor of 1.1 of our prediction. in other words it is $[x/1.1, 1.1x]$. Examining this on the log scale:

$$\begin{aligned} 2\hat{\sigma} &= \log(1.1x) \\ &= \log(x) + \log(1.1) \end{aligned}$$

Which means that $\hat{\sigma} = .047$

(b)

We can just use the equation for R^2

$$\begin{aligned} R^2 &= 1 - \frac{\hat{\sigma}^2}{\sigma^2} \\ &= 1 - \frac{.047}{.05} \\ &= .94 \end{aligned}$$

Problem 3

(a)

```
var1 <- rnorm(1000, 0 , 1)
var2 <- rnorm(1000, 0 , 1)
fit.3 <- lm(var1 ~ var2)
summary(fit.3)
```

No the p-value of the intercept is .226, indicating it does not meet the widely accepted .05 significance level.

(b)

```
z.scores.all <- rep(NA, 1000)
for (j in 1:1000) {
  z.scores <- rep(NA, 100)
  for (k in 1:100) {
    var1 <- rnorm(1000, 0, 1)
    var2 <- rnorm(1000, 0, 1)
    fit <- lm(var1 ~ var2)
    z.scores[k] <- coef(fit)[2] / se.coef(fit)[2]
  }
  z.scores.all[j] <- sum(z.scores >= 2)
}

z.scores.sorted <- sort(z.scores.all)
median(z.scores.sorted)
z.scores.sorted[50]
z.scores.sorted[950]
```

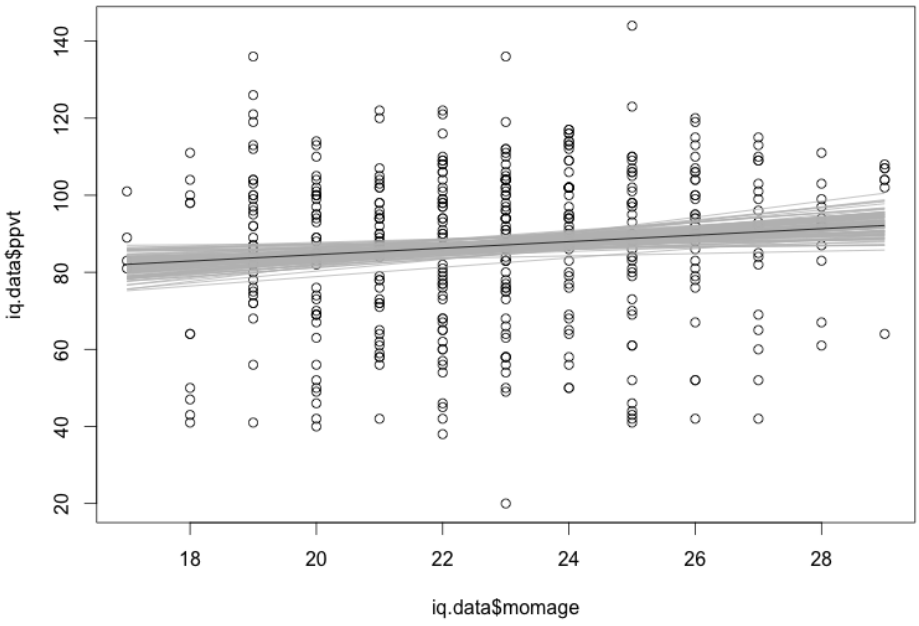
I ran the analysis for 1000 trials. I find that the median value is 2. I find that the 95% confidence interval is [0, 5]

Problem 4

(a)

```
Residuals:
    Min      1Q  Median      3Q     Max
-67.109 -11.798   2.971  14.860  55.210

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  67.7827     8.6880   7.802 5.42e-14 ***
momage        0.8403     0.3786   2.219    0.027 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 20.34 on 398 degrees of freedom
Multiple R-squared:  0.01223,Adjusted R-squared:  0.009743
F-statistic: 4.926 on 1 and 398 DF,  p-value: 0.02702
```

The slope coefficient implies that for each unit increase in momage the child's test score will increase by .8403 units. It looks like mothers should give birth later in life. In making these recommendations I'm assuming that momage is the only feature relevant to the test performance of the children.



(b)

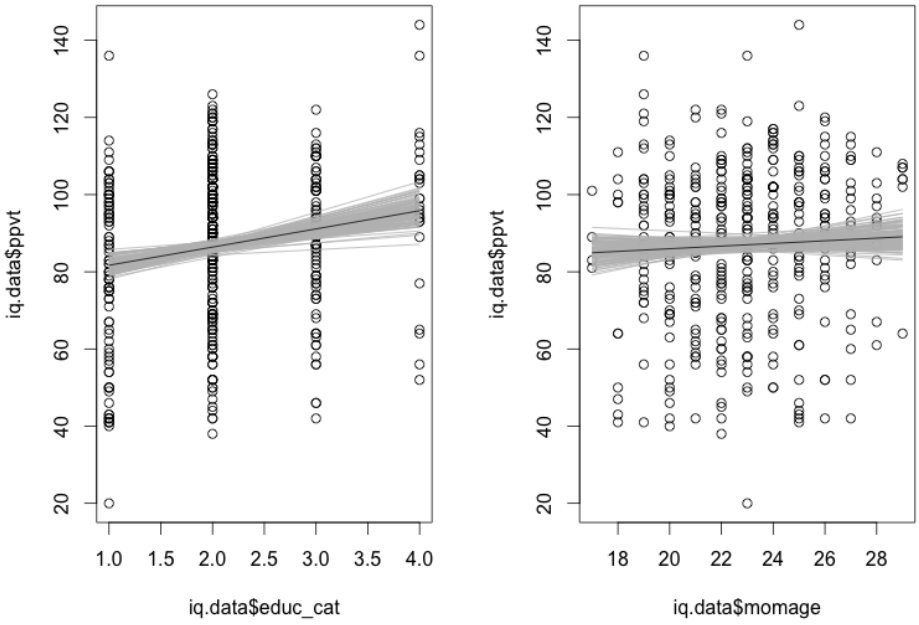
```
library("foreign")
iq.data <- read.dta("child.iq.dta")
fit.4 <- lm(ppvt ~ ., data=iq.data)
summary(fit.4)

Residuals:
    Min      1Q  Median      3Q     Max
-61.763 -13.130   2.495  14.620  55.610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  69.1554     8.5706   8.069 8.51e-15 ***
educ_cat      4.7114     1.3165   3.579 0.000388 ***
momage        0.3433     0.3981   0.862 0.389003
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 20.05 on 397 degrees of freedom
Multiple R-squared:  0.04309,Adjusted R-squared:  0.03827
F-statistic: 8.939 on 2 and 397 DF,  p-value: 0.0001594
```

From the Summary statistics we can see that momage does not appear to contribute significantly to the model. To test this let's run ANOVA.

```
fit.4.noage <- lm(ppvt ~ educ_cat, data=iq.data)
anova(fit.4.noage, fit.4)

Analysis of Variance Table

Model 1: ppvt ~ educ_cat
Model 2: ppvt ~ educ_cat + momage
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    398 159816
2    397 159517  1    298.82 0.7437  0.389
```

These ANOVA results indicate that the addition of the momage variable does not add any predictive power. Thus we should change our conclusions in (a). This information suggests that it does not matter when a mother gives birth.

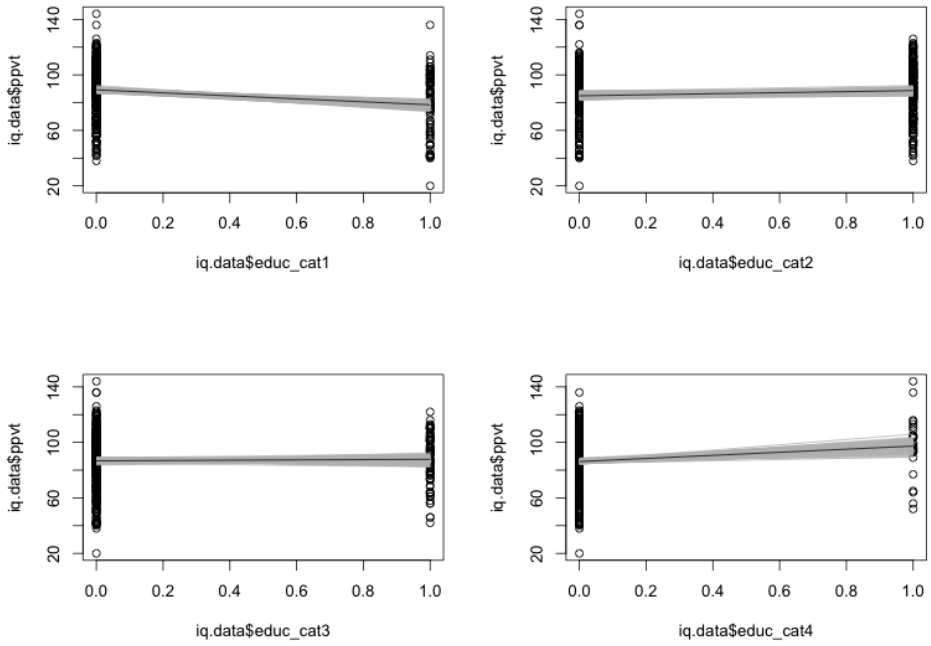
Next I plot the regression line for both variables.



(c)

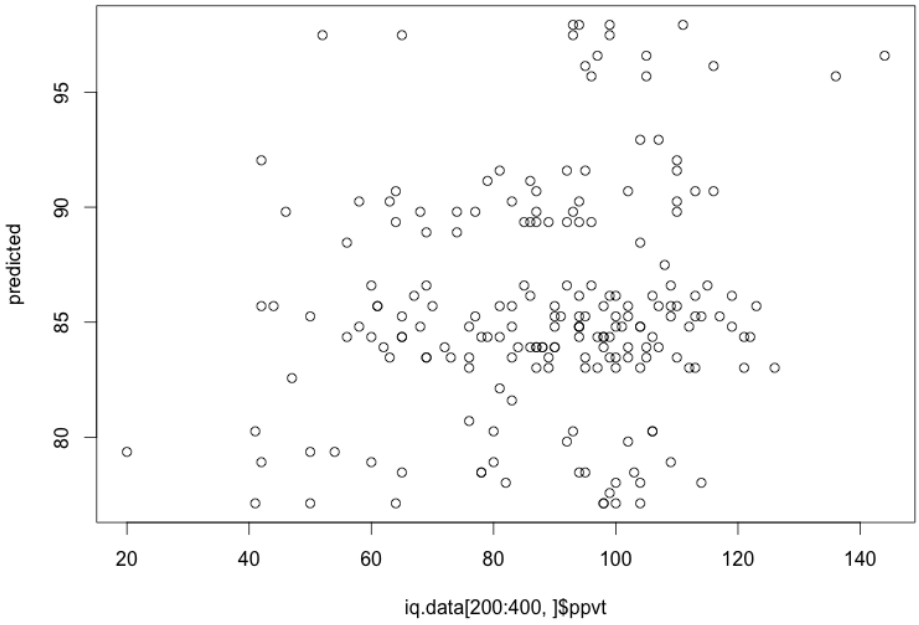
```
# Create Highschool completion factor
iq.data$educ_cat <- as.factor(iq.data$educ_cat)
temp <- model.matrix( ~ educ_cat - 1, data=iq.data )
iq.data <- cbind(iq.data, temp)
head(iq.data)

fit.4.c <- lm(ppvt ~ momage + momage*educ_cat2, data=iq.data)
summary(fit.4.c)

lm(formula = ppvt ~ momage + momage * educ_cat2, data = iq.data)

Residuals:
    Min      1Q  Median      3Q     Max
-65.041 -11.594   2.896  14.886  56.995

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       62.4546    11.8408   5.275  2.2e-07 ***
momage             0.9820     0.5132   1.914   0.0564 .
educ_cat2          9.6726    17.4080   0.556   0.5788
momage:educ_cat2  -0.2517     0.7587  -0.332   0.7402
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Residual standard error: 20.29 on 396 degrees of freedom
Multiple R-squared:  0.02175,Adjusted R-squared:  0.01433
F-statistic: 2.934 on 3 and 396 DF,  p-value: 0.0333
```



(d)

```
par(mfrow=c(1, 1))
iq.data <- read.dta("child.iq.dta")
fit.4.d <- lm(ppvt ~ ., data=iq.data[1:200, ])
predicted = predict(fit.4.d, iq.data[200:400, ])
plot(iq.data[200:400, ]$ppvt, predicted)
```



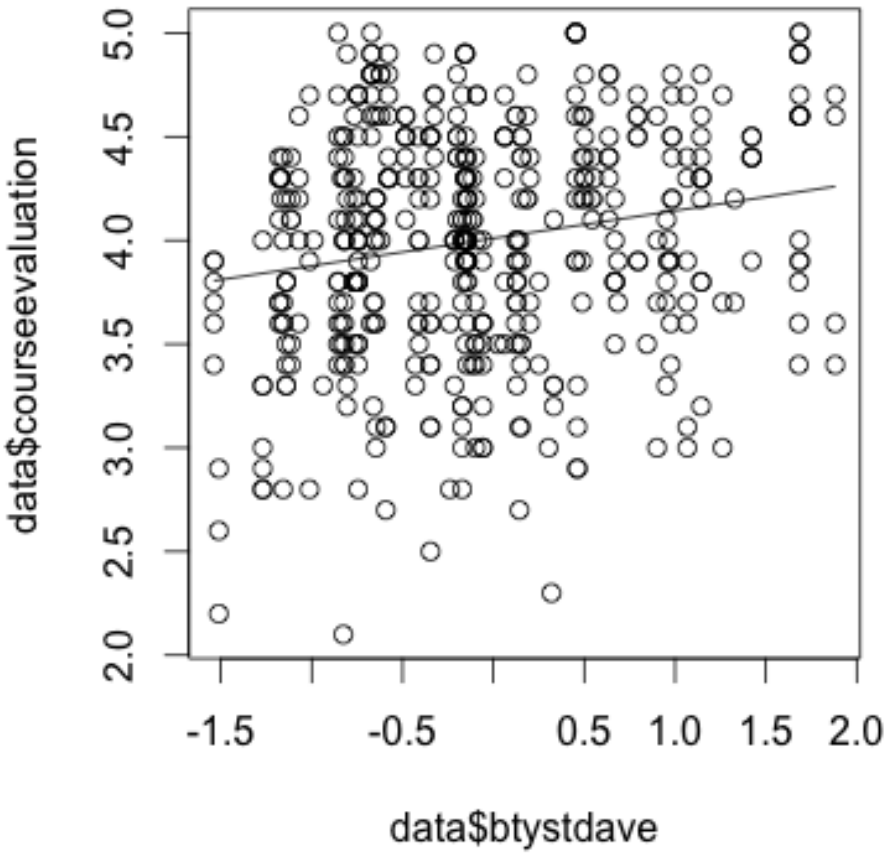
Problem 5

(a)

```
# Do more beautiful professors get better marks
par(mfrow=c(1, 1))
plot(data$btystdave, data$courseevaluation)

fit.5.beauty <- lm(courseevaluation ~ btystdave, data=data)
curve(fit.5.beauty$coef[1] + fit.5.beauty$coef[2] * x, add=TRUE)

# Add dotted lines to show +/- 1 standard deviation
curve (fit.5.beauty$coef[1] + fit.5.beauty$coef[2]*x + summary(fit.5.beauty)$sigma, lty=2
curve (fit.5.beauty$coef[1] + fit.5.beauty$coef[2]*x - summary(fit.5.beauty)$sigma, lty=2

# Look at male and female professors
fit.5.gender <- lm(courseevaluation ~ btystdave + female, data=data)

par(mfrow=c(1,2))
plot(data$btystdave[data$female == 0], data$courseevaluation[data$female == 0],
     xlab='beauty', ylab='average teaching evaluation', main='Men')
curve(fit.5.gender$coef[1] + fit.5.gender$coef[2] * x, add=TRUE)

plot(data$btystdave[data$female == 1], data$courseevaluation[data$female == 1],
     xlab='beauty', ylab='average teaching evaluation', main='Females')
```

```
curve(fit.5.gender$coef[1] + fit.5.gender$coef[2] * x + fit.5.gender$coef[3] * 1, add=TRU
```

Regressing on just beauty gives us an $R^2 = .035$ so the model fit isn't great. Below I plot the fit for beauty vs course evaluation.

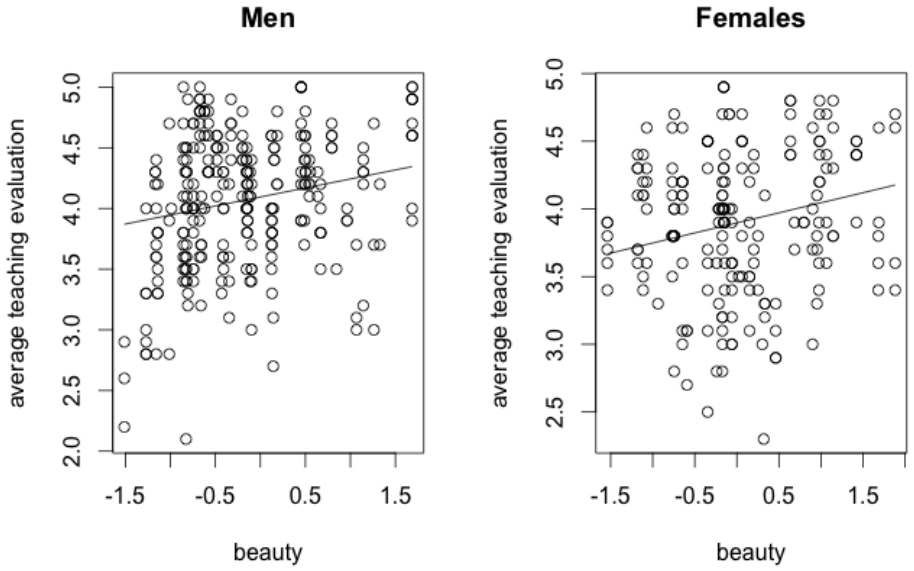


Now let's see if gender plays a role in the courseevaluations. We find that our R^2 has doubled and both beauty and gender seem to be statistically significant. We can interpret the coefficients as follows:

I- (Intercept) For a female teacher with 0 beauty we can expect a course rating between [4.02, 4.16] with 95% confidence

- (beauty) We can expect that for a female instructor an increase in beauty of 1 will result in an increase in the course evaluation by .148

- (female) A male teacher with 0 beauty will have coursevaluation reduced by .198 on average.


Men
Females
average teaching evaluation
beauty
2.0
2.5
3.0
3.5
4.0
4.5
5.0
-1.5
-0.5
0.5
1.5

Chapter 4

Linear Regression: Before and After Fitting the Model

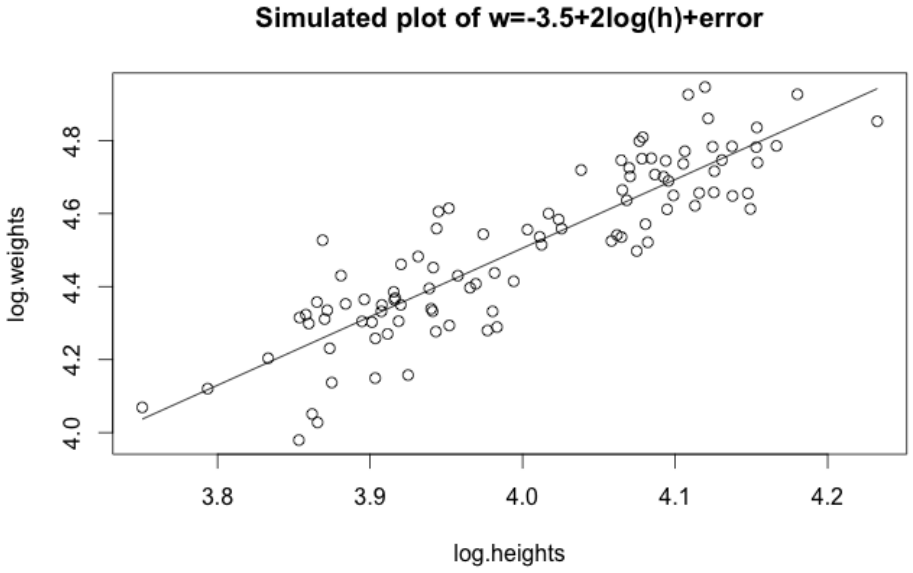
Problem 1

(a)

The bounds are easy to scale, just apply the exponential. So the weights will be within a factor of $[e^{-.25}, e^{.25}]$.

(b)

```
# Create a fictional population composed of two distinct groups with different means
group_1 <- rnorm(n=50, 50, sd = 2.5)
group_2 <- rnorm(n=50, 60, sd = 3)
heights <- c(group_1, group_2)
log.heights <- log(heights)

# Use the model to generate weights
log.weights <- -3.5 + 2 * log.heights + rnorm(n=100, mean=0, sd=.1)

# Fit the model
fit.1 <- lm(log.weights ~ log.heights)


# plot the model and data points
plot(log.heights, log.weights,main='Simulated plot of w=-3.5+2log(h)+error')
curve(cbind(1, x) %*% coef(fit.1), add=TRUE)
```



Problem 2

(a)

I looked at the data and found that there were about 670 ish rows that had missing values so I removed them.

```
heights <- read.dta('heights.dta')

# How many missing values
sum(is.na(heights))

# Remove all rows with missing values
heights <- heights[complete.cases(heights), ]
```

(b)

I trained a plain model but the intercept represents the earnings of someone who is 0 ft tall. This doesn't make sense because it is impossible. In order to interpret the intercept as the average earnings of someone of average height we need to subtract off the mean height and divide by the standard deviation. In other words standardize using a z-score.

```
# Fit model with height as predictor
fit.2 <- lm(earn ~ height, data=heights)
summary(fit.2)
```

```
# Transform the data so that the intercept is average earnings for people with average he
heights.scaled <- heights
heights.scaled$height <- (heights$height - mean(heights$height)) / sd(heights$height)
fit.2 <- lm(earn  ~ height, data=heights.scaled)
summary(fit.2)
```

(c)

```
# Transform variables in various ways
sex <- heights$sex - 1
height <- heights$height
log.height <- log(heights$height)
z.height <- (heights$height - mean(heights$height)) / sd(heights$height)
earn <- heights$earn
log.earn <- heights$earn

# Fit model using sex, height
fit.2.c <- lm(earn ~ sex + height)
summary(fit.2.c)

# Fit model using log height and log earn
fit.2.c <- lm(log.earn ~ sex + log.height)
summary(fit.2.c)

# Fit model with interaction between sex and height
fit.2.c <- lm(earn ~ sex + z.height + sex*z.height)
summary(fit.2.c)
```

(d)

The model that I chose to stick with is the last one earn = sex + z.height + sex * z.height. I felt like it was the easiest to interpret and also had the highest R^2 of the models that I examined, mainly due to the interaction term. Below is the summary:

```
Call:
lm(formula = earn ~ sex + z.height + sex * z.height)

Residuals:
   Min     1Q Median     3Q    Max
-31209 -12591  -3172   7223 171109

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     26259       1248  21.041  < 2e-16 ***
```

```
sex            -10868       1492  -7.286 5.36e-13 ***
z.height         2940       1047   2.809  0.00505 **
sex:z.height    -1536       1412  -1.088  0.27670
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 18460 on 1375 degrees of freedom
Multiple R-squared:  0.1296,Adjusted R-squared:  0.1277
F-statistic: 68.22 on 3 and 1375 DF,  p-value: < 2.2e-16
```

The interpretation of the coefficients is as follows:

- (Intercept) The intercept is the average earnings for an average height male.
- (sex) The sex represents the difference in salary between a male of average height and a female of average height. With the average height female being 10868 fewer dollars than the male.
- (sex*height) This term helps control for the difference in the average heights between males and females.

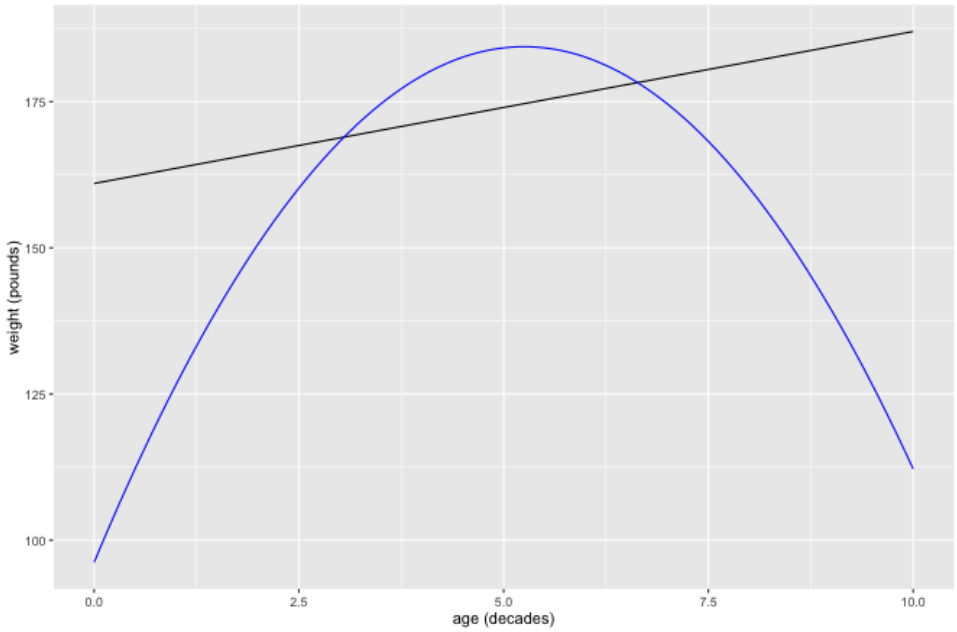
Problem 3

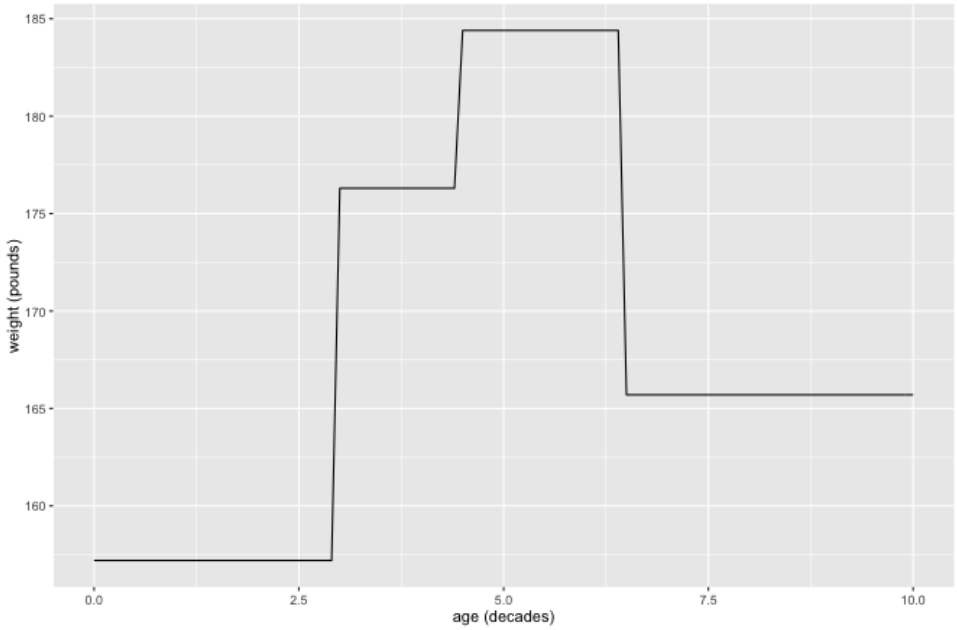
Here we just need to plot a few things so below you can find the code for the plotting, and the plots :)

```
ggplot(data.frame(x=c(0, 10)), aes(x)) +
    stat_function(fun=function(x) 161 + 2.6 * x) +
    stat_function(fun=function(x) 96.2 + 33.6 * x - 3.2 * (x)^2, col="blue") +
    labs(x="age (decades)", y="weight (pounds)")

ggplot(data.frame(x=c(0, 10)), aes(x)) +
    stat_function(fun=function(x) 157.2 +
                    19.1 * ifelse(3 <= x & x < 4.5, 1, 0) +
                    27.2 * ifelse(4.5 <= x & x < 6.5, 1, 0) +
                    8.50 * ifelse(6.5 <= x, 1, 0)) +
    labs(x="age (decades)", y="weight (pounds)")
```

Problem 4

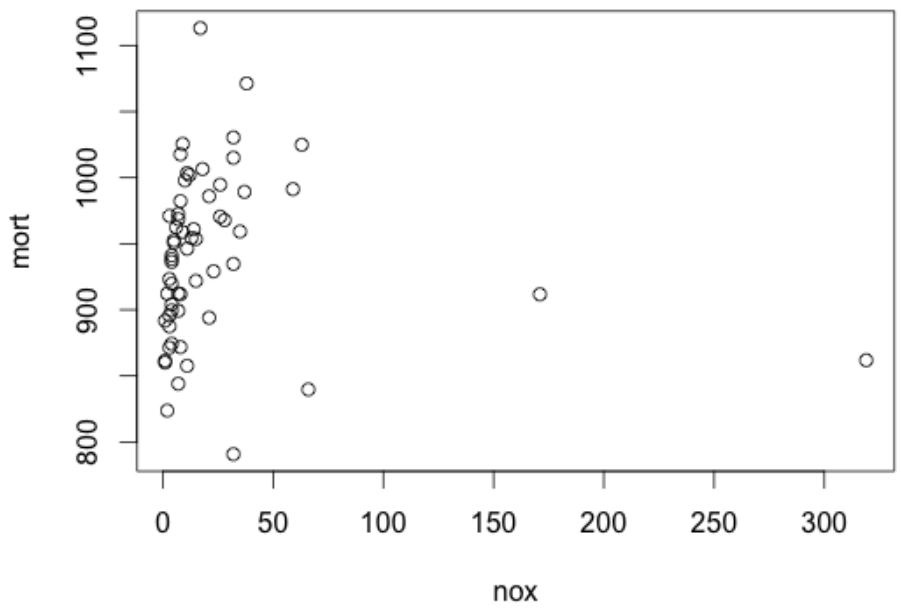
It looks like a linear regression model would fit this data well. There are a few outliers, however just between these two it looks like it follows a pretty linear trend.





(a)

```
# Load in Data
pollution <- read.dta('pollution.dta')
attach(pollution)

# Plot nitric oxide vs mortality
plot(nox, mort)

# Fit the model
```

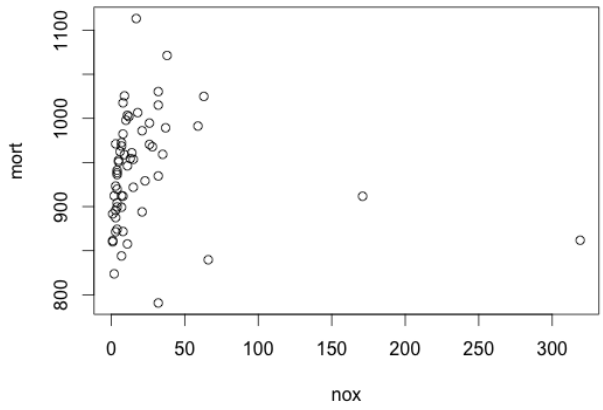


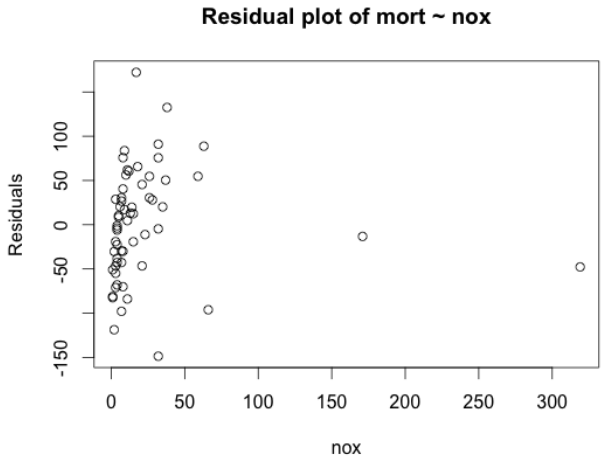

```
fit.4.a <- lm(mort ~ nox)
summary(fit.4.a)

# Examine Residuals
plot(nox, resid(fit.4.a), ylab='Residuals', main='Residual plot of mort ~ nox')
par(mfrow=c(2,2))
plot(fit.4.a)

# Examine fit on plot
par(mfrow=c(1,1))
plot(nox, mort)
abline(fit.4.a)
```

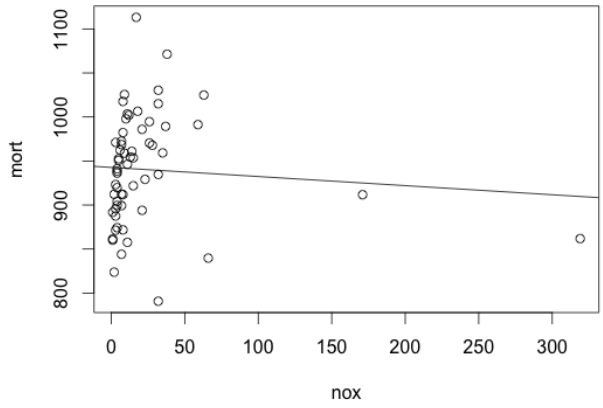
It looks like a linear regression model would fit this data well. There are a few outliers, however just between these two it looks like it follows a pretty linear trend.

The residual plot is peculiar because it looks exactly like the original. This indicates to me that the fitted line is almost flat through the data. This most likely happened due to the outliers.

We can see that this hypothesis is reasonable in the above plot.



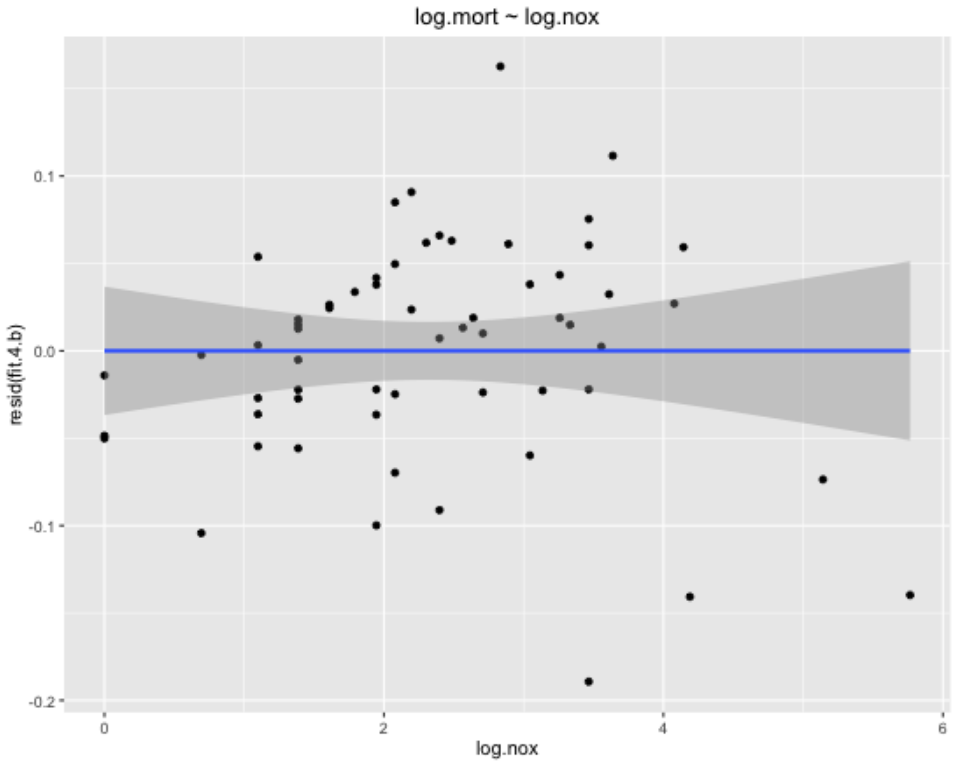




(b)

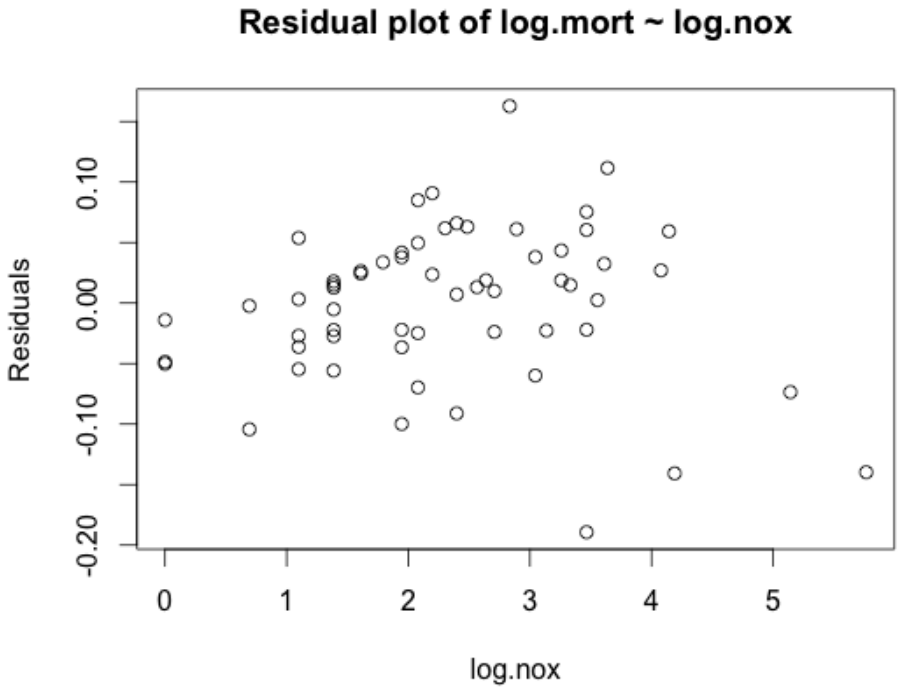
Log transform the predictor and the response. This will fix our problem, by drastically reducing the perceived size of the outliers.

```
# Log transform the data
log.nox <- log(nox)
log.mort <- log(mort)
fit.4.b <- lm(log.mort ~ log.nox)
plot(log.nox, log.mort)
abline(fit.4.b)

ggplot(data=pollution, aes(x=log.nox, y=log.mort)) + geom_point() +
  stat_smooth(method="lm", formula=y ~ x, se=TRUE) +
  labs(title='log.mort ~ log.nox')

ggplot(data=pollution, aes(x=log.nox, y=resid(fit.4.b))) + geom_point() +
  labs(title='Residuals for log.mort ~ log.nox')
```



(c)

```
Call:
lm(formula = log.mort ~ log.nox)

Residuals:
     Min       1Q   Median       3Q      Max
-0.18930 -0.02957  0.01132  0.03897  0.16275
```




```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.807175   0.018349 370.975   <2e-16 ***
log.nox     0.015893   0.007048   2.255   0.0279 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.06412 on 58 degrees of freedom
Multiple R-squared:  0.08061,Adjusted R-squared:  0.06476
F-statistic: 5.085 on 1 and 58 DF,  p-value: 0.02792
```

- (Intercept) The intercept represents the log(mortality) when nox is 0. We can transform back to the original scale with $e^{6.8}$ to get the mortality at nox=0.
- (log.nox) For each 1% increase in nitrous oxide we can expect a 1.5% increase in mortality rate

(d)

```
z.nox <- (nox - mean(nox)) / sd(nox)
z.hc <- (hc - mean(hc)) / sd(hc)
z.so2 <- (so2 - mean(so2)) / sd(so2)

fit.4.d <- lm(log(mort) ~ z.nox + z.hc + z.so2)
```

```
summary(fit.4.d)
```

I fitted the log mortality rate vs the z-scored predictors. This allowed for the easiest interpretation in my opinion.

- (Intercept) The average mortality rate for the average levels of nox, hc, and so2 is $e^{6.84}$
- (z.nox) one standard deviation difference in nox results in $e^{.148} = 1.15$ times, or a 15%, increase in mortality rate
- (z.hc) One standard deviation difference in hydrocarbons results in $e^{-.16=.84}$ times, or -16% decrease in mortality rate
- (z.so2) One standard deviation difference in SO_2 results in a 1.3% increase in mortality rate.

(e)

```
# Create the dataframe
df <- data.frame(mort, z.nox, z.hc, z.so2)

# Split into training and testing sets
train <- df[1:round(nrow(df) / 2), ]
test <- df[(round(nrow(df) / 2) + 1):nrow(df), ]

# Fit the model and make predictions
fit.4.e <- lm(log(mort) ~ z.nox + z.so2 + z.hc, data=train)
predictions <- predict(fit.4.e, test)
cbind(truth=test$mort, prediction=exp(predictions), diff=test$mort-exp(predictions))
```

Problem 5

(a)

- (Difference) It is a simple statistic and is easy to interpret. It can lead to difficulties in interpreting proportions. You can have the same difference in fundraising and have widely different levels of closeness.
- (Ratio) The ratio addresses the proportional problem with difference. However, it is not symmetric. It is centered at 1 when both earn the same amount of money.
- (Log Difference) This is very similar to the first transformation, however it is less sensitive to outliers.
- (Relative Proportion) Centered at 0 and is symmetric. It will do well at capturing relative differences

(b)

You could transform them into an indicator feature 0 if republicans raised more and 1 if democrats did. This is advantageous because we are trying to predict voter shares in a larger model. This feature captures a large part of the relative information, did democrats do better or not? A major drawback is that we are discarding a lot of information. Namely how much better the democrats are doing.

Problem 6

β tells us that for a 1% change in the average price of cigarettes we can expect to see a .3% change in their sales.

Problem 7

Problem 8

I begin by training a model on the whole data set and looking to see what elements are statistically significant. This gives me an idea of what to make my base model out of.

```
Call:
lm(formula = courseevaluation ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82802 -0.07752  0.01949  0.10759  0.53467

Coefficients: (1 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.590e+05  4.706e+05  -0.550  0.58237
tenured           -2.345e-02  3.251e-02  -0.721  0.47126
profnumber         4.557e-04  5.212e-04   0.874  0.38250
minority          -5.702e-02  3.564e-02  -1.600  0.11045
age                3.651e-03  1.543e-03   2.366  0.01848 *
beautyf2upper      3.661e+04  2.895e+04   1.265  0.20669
beautyflowerdiv   -2.461e+04  4.209e+04  -0.585  0.55910
beautyfupperdiv    2.103e-01  1.511e-01   1.392  0.16472
beautym2upper     -4.851e+04  6.593e+04  -0.736  0.46226
beautymlowerdiv    1.848e+04  4.864e+04   0.380  0.70424
beautymupperdiv    7.557e+04  3.150e+04   2.399  0.01690 *
btystdave         -3.632e+04  9.304e+04  -0.390  0.69645
btystdf2u         -6.877e+04  6.360e+04  -1.081  0.28026
btystdfl           5.426e+04  8.438e+04   0.643  0.52060
```

```
btystdfu           7.962e+03  1.632e+04   0.488  0.62593
btystdm2u          8.821e+04  1.122e+05   0.786  0.43235
btystdml          -2.348e+04  7.988e+04  -0.294  0.76892
btystdmu          -1.566e+05  6.830e+04  -2.293  0.02237 *
class1            -2.145e-02  9.286e-02  -0.231  0.81740
class2             1.014e-01  1.346e-01   0.753  0.45172
class3            -1.213e-01  7.363e-02  -1.648  0.10022
class4            -6.729e-02  5.071e-02  -1.327  0.18530
class5             1.087e-01  9.545e-02   1.139  0.25531
class6            -4.252e-02  9.175e-02  -0.463  0.64328
class7            -1.310e-01  1.058e-01  -1.238  0.21638
class8             2.711e-01  1.365e-01   1.987  0.04762 *
class9            -3.772e-02  7.273e-02  -0.519  0.60426
class10            8.624e-02  9.420e-02   0.915  0.36049
class11           -1.358e-01  1.463e-01  -0.928  0.35394
class12           -2.774e-01  1.110e-01  -2.499  0.01287 *
class13           -7.218e-02  1.179e-01  -0.612  0.54090
class14            2.763e-01  1.156e-01   2.391  0.01729 *
class15           -1.641e-01  1.376e-01  -1.193  0.23369
class16            3.620e-02  1.032e-01   0.351  0.72582
class17            1.364e-01  7.756e-02   1.759  0.07941 .
class18            2.822e-01  1.008e-01   2.800  0.00536 **
class19           -1.861e-01  8.865e-02  -2.099  0.03644 *
class20           -2.665e-03  9.905e-02  -0.027  0.97855
class21           -2.323e-02  6.807e-02  -0.341  0.73308
class22           -5.503e-02  7.575e-02  -0.726  0.46798
class23            8.993e-02  8.697e-02   1.034  0.30176
class24           -8.296e-02  1.122e-01  -0.739  0.46024
class25           -1.251e-01  1.111e-01  -1.127  0.26048
class26            1.922e-01  1.116e-01   1.722  0.08590 .
class27            2.960e-01  1.357e-01   2.182  0.02971 *
class28           -6.401e-02  1.111e-01  -0.576  0.56469
class29           -2.297e-01  1.380e-01  -1.664  0.09688 .
class30           -6.287e-02  8.616e-02  -0.730  0.46601
didevaluation     -1.271e-03  1.367e-03  -0.930  0.35283
female            -3.238e-02  2.848e-02  -1.137  0.25632
formal             2.614e-02  3.680e-02   0.710  0.47796
fulldept          -4.235e-03  4.083e-02  -0.104  0.91745
lower              6.739e-03  3.031e-02   0.222  0.82417
multipleclass             NA         NA      NA       NA
nonenglish        -1.550e-01  5.011e-02  -3.093  0.00212 **
onecredit          7.193e-02  5.747e-02   1.251  0.21148
percentevaluating  1.687e-03  8.997e-04   1.875  0.06147 .
profevaluation     9.242e-01  2.001e-02  46.186  < 2e-16 ***
students           7.974e-04  8.792e-04   0.907  0.36496
tenuretrack       -3.992e-03  3.580e-02  -0.111  0.91129
```

```
blkandwhite         1.106e-03  3.667e-02   0.030  0.97594
btystdvariance     -1.373e-03  9.674e-03  -0.142  0.88724
btystdavepos       -1.145e+04  8.874e+03  -1.291  0.19761
btystdaveneg       -1.145e+04  8.874e+03  -1.291  0.19761
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1839 on 400 degrees of freedom
Multiple R-squared:  0.9049,Adjusted R-squared:  0.8901
F-statistic: 61.38 on 62 and 400 DF,  p-value: < 2.2e-16
```

We see right away that the following factors at least appear meaningful at the 5% level. [profevaluation, nonenglish, btystdmu, beautymupperdiv, age]. We can now train a model on this reduced set of features.

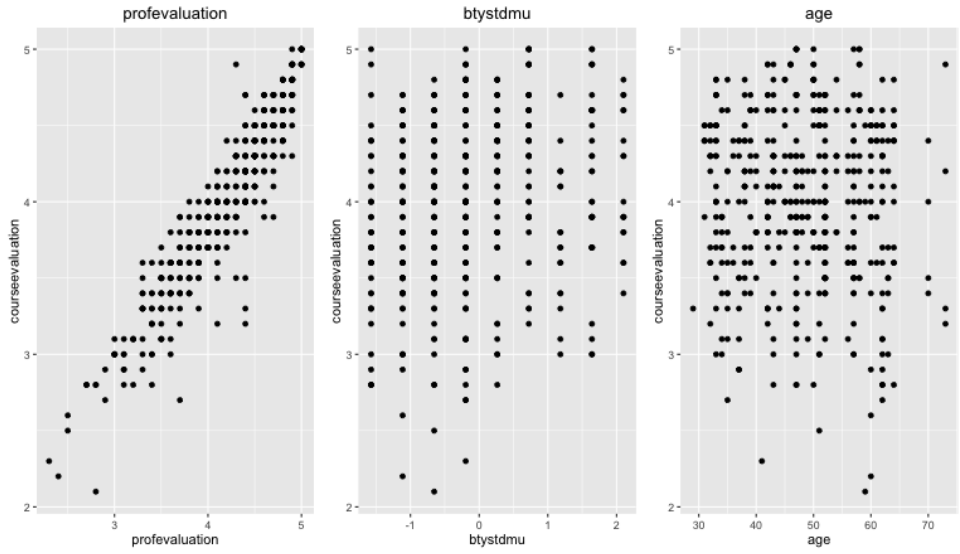
```
Call:
lm(formula = courseevaluation ~ profevaluation + nonenglish +
    btystdmu + beautymupperdiv + age, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.92102 -0.10370  0.01617  0.12830  0.72921

Coefficients: (1 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.1210840  0.0872760  -1.387 0.166004
profevaluation   0.9469193  0.0168967  56.042  < 2e-16 ***
nonenglish      -0.0978276  0.0378164  -2.587 0.009991 **
btystdmu         0.0287699  0.0099357   2.896 0.003965 **
beautymupperdiv         NA         NA      NA       NA
age              0.0036353  0.0009702   3.747 0.000202 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1926 on 458 degrees of freedom
Multiple R-squared:  0.8805,Adjusted R-squared:  0.8795
F-statistic:   844 on 4 and 458 DF,  p-value: < 2.2e-16
```

We see immediately that beautymupperdiv is colinear with another feature so we drop it from the next model that we train.

```
Call:
lm(formula = courseevaluation ~ profevaluation + nonenglish +
    btystdmu + age, data = data)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.92102 -0.10370  0.01617  0.12830  0.72921

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.1210840  0.0872760  -1.387 0.166004
profevaluation  0.9469193  0.0168967  56.042  < 2e-16 ***
nonenglish     -0.0978276  0.0378164  -2.587 0.009991 **
btystdmu        0.0287699  0.0099357   2.896 0.003965 **
age             0.0036353  0.0009702   3.747 0.000202 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1926 on 458 degrees of freedom
Multiple R-squared:  0.8805,Adjusted R-squared:  0.8795
F-statistic:   844 on 4 and 458 DF,  p-value: < 2.2e-16
```

Now we have a model with only 4 features instead of 64, and only a 2% drop in R^2. Looking at the magnitude of the coefficients it is clear that the most valuable feature is profevaluation. It seems like this is highly correlated with the courseevaluation. Now let's look and see if log scaling might help in any cases.



It appears that none of our variables contain any significant outliers which might be dealt with on the log axis. I'm just going to z-scale the features to help with interpretability.

```
Call:
lm(formula = courseeval ~ z.profeval + z.age + z.btystdmu + nonenglish)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.92102 -0.10370  0.01617  0.12830  0.72921

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.004188   0.009239 433.401  < 2e-16 ***
z.profeval   0.514996   0.009190  56.042  < 2e-16 ***
z.age        0.035636   0.009510   3.747 0.000202 ***
z.btystdmu   0.027885   0.009630   2.896 0.003965 **
nonenglish  -0.097828   0.037816  -2.587 0.009991 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1926 on 458 degrees of freedom
Multiple R-squared:  0.8805,Adjusted R-squared:  0.8795
F-statistic:   844 on 4 and 458 DF,  p-value: < 2.2e-16
```

We can interpret the coefficients in the following way:

- (Intercept) The average courseevaluation for an english speaking professor of average age, beauty, and professor evaluation is 4.
- (z.profeval) For someone of average age, and beauty, who is a native english speaker we can expect that one standard deviation increase in the profeval score will results in .515 increase in the courseevaluation
- (age) For a professor with an average score and beauty who speaks english we can expect that one standard deviation change in age will result in .035 change in courseevaluation
- (btystdmu) similar interpretation
- (nonenglish) All else kept at average the non English speaking professors receive on average a score .097 points lower than their English speaking colleagues.

Chapter 5

Logistic Regression

Problem 1

(a-c)

I fit a number of models with different predictors. I began with just

```
fit.1 <- glm(vote ~ race, data=nes, family=binomial(link='logit'))
display(fit.1)
```

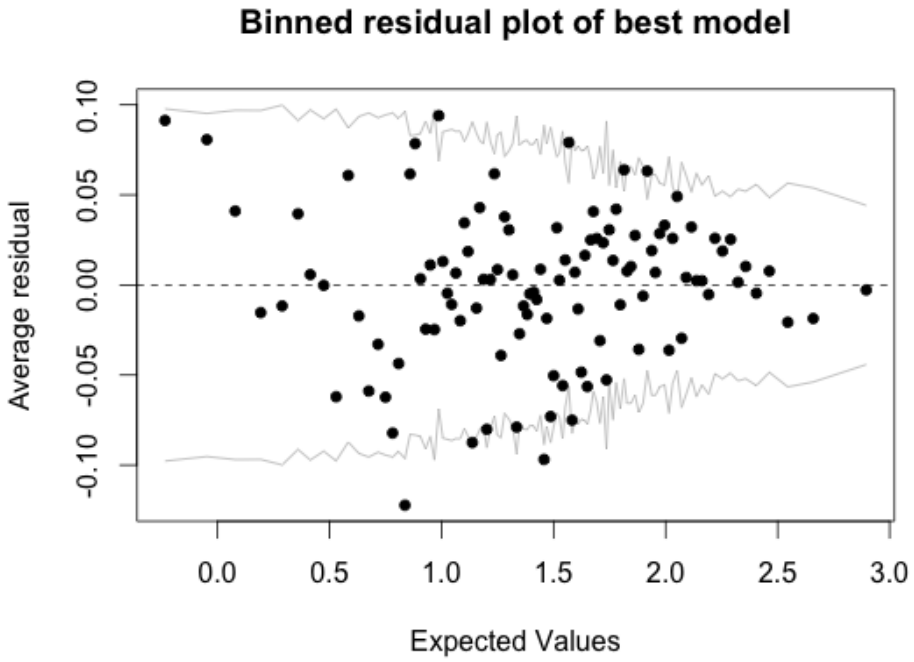
And found the deviance to be 44 659. I then looked at adding gender which reduced the deviance to 44619. To check that deviance is behaving as I would expect I added black, and saw no change, because this feature is accounted for in race. I then added income which saw a huge drop to 39107. Then age which droped it down to 37518. Lastly I added parent_party and saw the deviance fall to 11604. I took this as my best model and examined the residuals

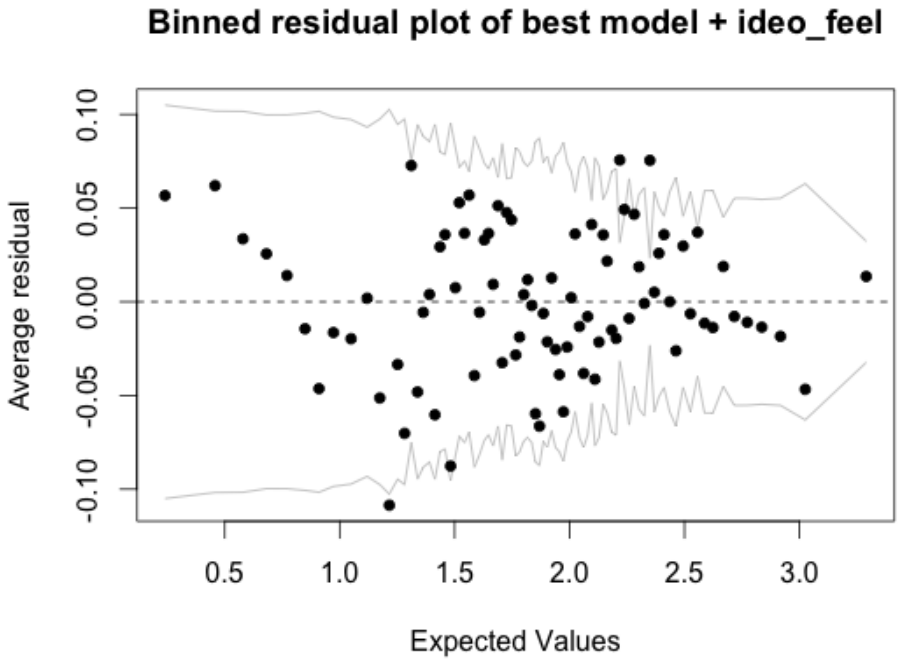
```
fit.1 <- glm(vote~race + age + income + gender + parent_party, data=nes, family=binomial(
binnedplot(predict(fit.1, resid(fit.1, type='response'), main="Binned residual plot of be
```

As you can see the residuals differences appear to fall more or less within the confidence bounds. This leads me to say that the model assumptions are at least in the ball park. However, there is a little bit of heteroskedasticity to this plot. This probably means that there is some factor that I'm missing which is affecting the voting habits. After poking around a bit I found the feature ideo_feel which flattens out the residual plot significantly.

```
fit.1 <- glm(vote~race + age + income + gender + parent_party + idea_feel, data=nes, fami
binnedplot(predict(fit.1, resid(fit.1, type='response'), main="Binned residual plot of be
```

Lastly we need to interpret the coefficients of the model.

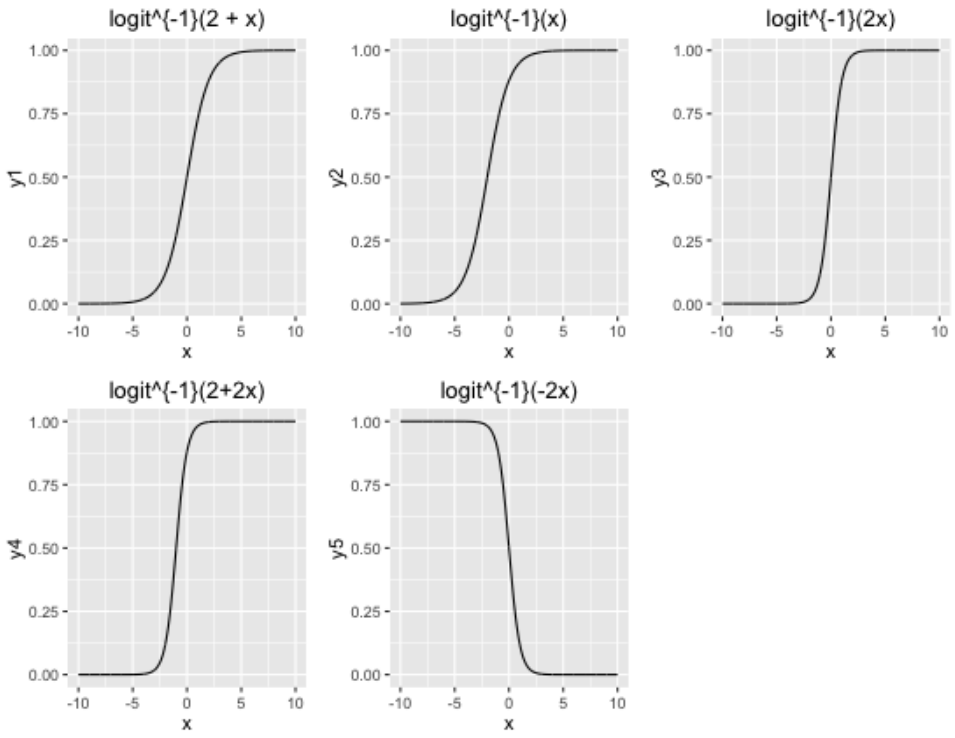




Problem 2

```
library('ggplot2')
library('gridExtra')
x <- seq(-1000,1000) / 100
```

```
y1 <- 1 / (1 + exp(-x))
y2 <- 1 / (1 + exp(-(2 + x)))
y3 <- 1 / (1 + exp(-2*x))
y4 <- 1 / (1 + exp(-(2 + 2*x)))
y5 <- 1 / (1 + exp(2*x))
df <- data.frame(x, y1, y2, y3, y4, y5)
p1 <- ggplot(df, aes(x=x, y=y1), group=rating) +
      geom_line() +
      ggtitle("logit^{-1}(2 + x)")
p2 <- ggplot(df, aes(x=x, y=y2), group=rating) +
      geom_line() +
      ggtitle("logit^{-1}(x)")
p3 <- ggplot(df, aes(x=x, y=y3), group=rating) +
      geom_line() +
      ggtitle("logit^{-1}(2x)")
p4 <- ggplot(df, aes(x=x, y=y4), group=rating) +
      geom_line() +
      ggtitle("logit^{-1}(2+2x)")
p5 <- ggplot(df, aes(x=x, y=y5), group=rating) +
      geom_line() +
      ggtitle("logit^{-1}(-2x)")

grid.arrange(p1, p2, p3, p4, p5, ncol=3)
```



Problem 3

This is just some simple algebra:

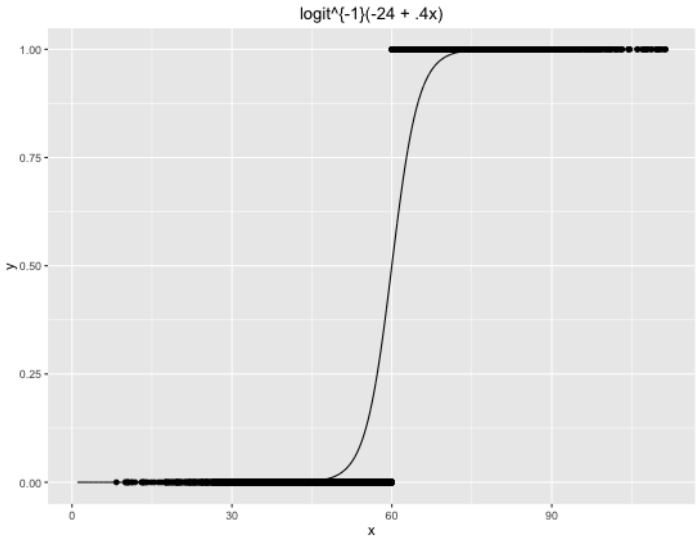
$$
\begin{aligned}
y &= \frac{1}{1+e^{-x}} \\
y(1+e^{-x}) &= 1 \\
ye^{-x} &= 1-y \\
e^{-x} &= \frac{1-y}{y} \\
\frac{1}{e^{x}} &= \frac{1-y}{y} \\
e^{x} &= \frac{y}{1-y} \\
x &= \log\left(\frac{y}{1-y}\right) \\
\alpha+\beta x &= \log\left(\frac{y}{1-y}\right) \\
\alpha &= \log\left(\frac{.27}{1-.27}\right) && x=0 \\
\alpha &= -.995
\end{aligned}
$$

And solving for when $x = 6$ and $p = .88$

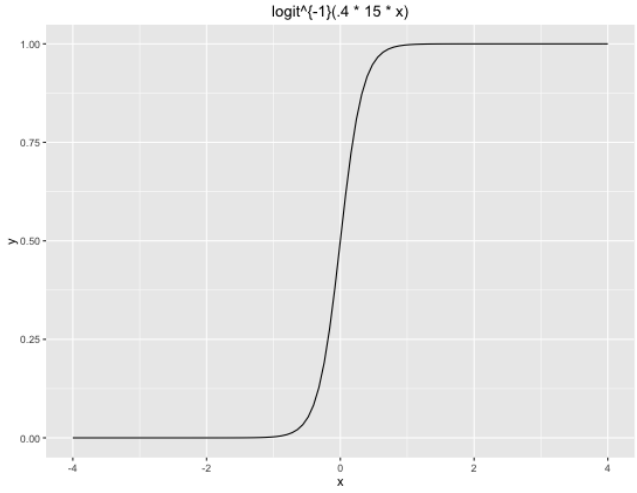
$$
\begin{aligned}
6\beta - .995 &= \log\left(\frac{.88}{1-.88}\right) \\
\beta &= 2.987
\end{aligned}
$$

Problem 5

(a)



(b)



(c)

```
x <- seq(100, 10000) / 100
samples <- rnorm(9901, 60, 15)
fake <- rnorm(9901, 0, 1)
y <- 1 / (1 + exp(24 - .4*x))
df <- data.frame(x, y, samples, fake)
df['prediction'] <- ((1 / (1 + exp(24 - .4 * df['samples']))) > .5) + 0
```

```
fit.5 <- glm(prediction ~ x, data=df, family=binomial(link="logit"))
display(fit.5)

fit.5 <- glm(prediction ~ x + fake, data=df, family=binomial(link="logit"))
display(fit.5)
```

The deviance decreased by .1 between the two settings. Adding noise can improve fit, but in this case it is super minor.

Problem 6

Problem 7

In order to create this poorly fitting example. I sampled my predictions randomly from a uniform distribution. This way the 1's and 0's would be randomly dispersed amongst the x values. I find the the fit is terrible, though I don't think it's fair to make the statement that these data point are inconsistent with ANY logistic regression fit because it could be possible for a kernel to make these points linearly separable.

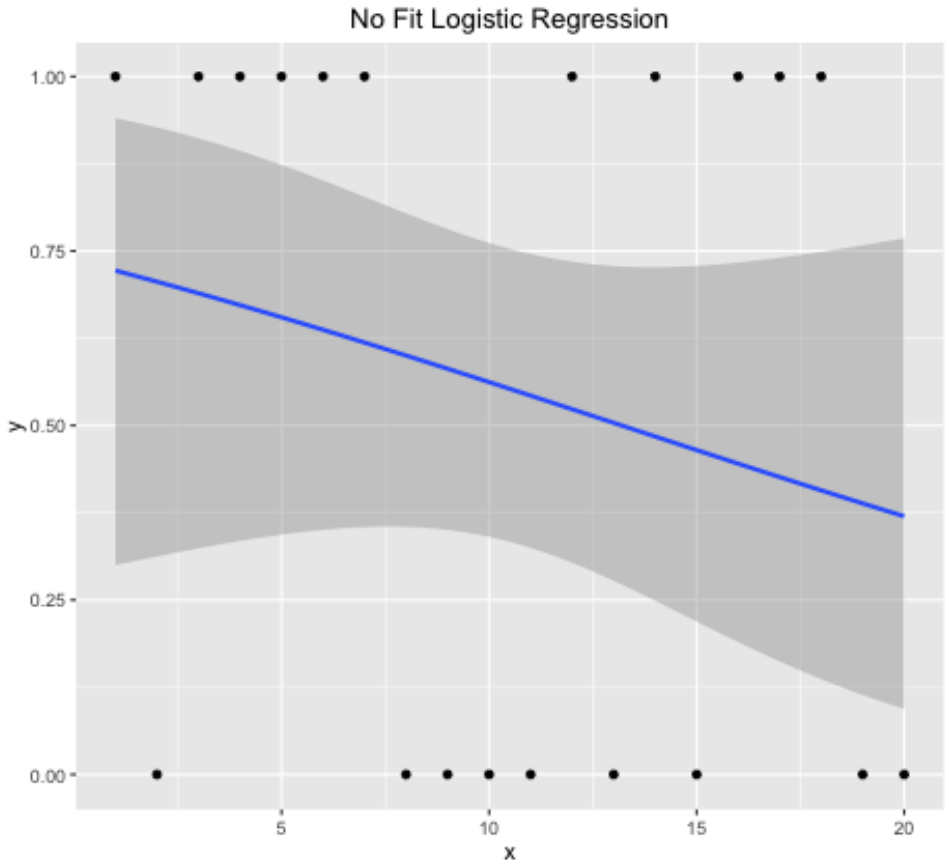
```
x <- seq(1, 20)
y <- (runif(20) > .5) + 0
fit.7 <- glm(y ~ x, family=binomial(link="logit"))
display(fit.7)

glm(formula = y ~ x, family = binomial(link = "logit"))
            coef.est coef.se
(Intercept)  1.03     0.99
x           -0.08     0.08
---
  n = 20, k = 2
  residual deviance = 26.6, null deviance = 27.5 (difference = 1.0)
```

It is clear from the standard error of the coefficient for x that it is not statistically significant indicating that x is a poor predictor of y. You can also see that the confidence intervals are large indicating a large degree of uncertainty. You can also notice how the slope of the line is very shallow. This leads to a large degree of uncertainty over the whole range of known values.

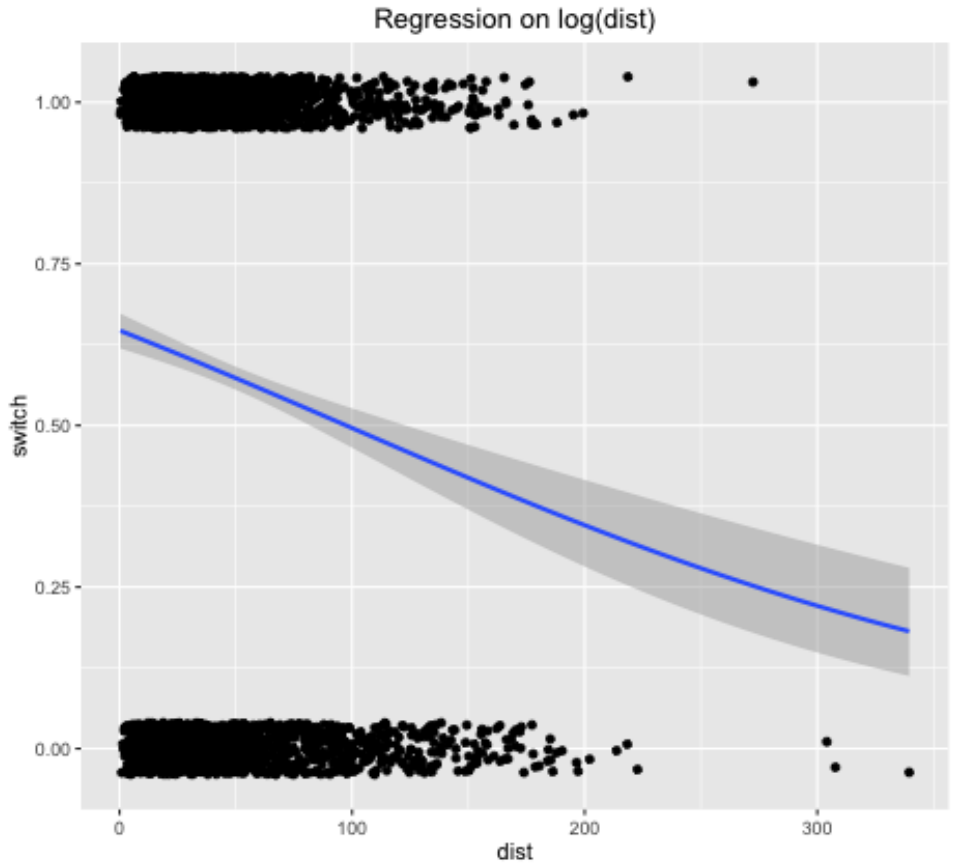
Problem 8

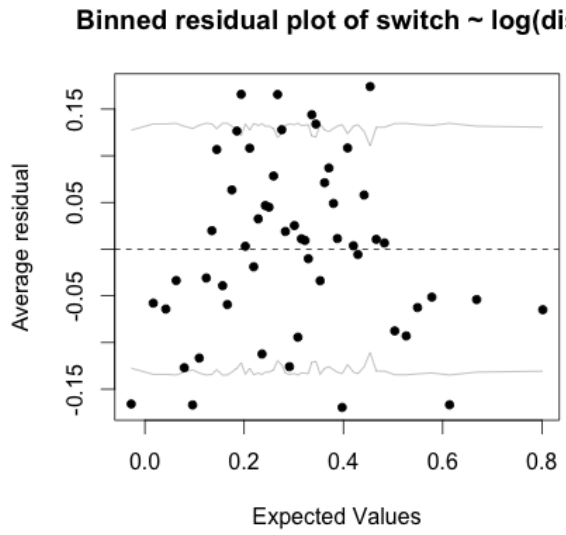
Problem 9

```
fit.9 <- glm(switch ~ log(dist), data=df,family=binomial(link="logit"))
```




```
display(fit.9)

ggplot(data=df, aes(x=dist, y=switch)) +
  stat_smooth(method="glm", method.args = list(family = "binomial")) +
  geom_point(data=df, aes(x=jitter(dist, .2), y=jitter(switch, .2))) +
  ggtitle("Regression on log(dist)")


# (c)
par(mfrow=c(1,2))
binnedplot(predict(fit.9), resid(fit.9, type='response'), main="Binned residual plot of s
plot(log(df$dist), resid(fit.9), ylab='Residuals', main='Residual plot of switch ~ log(di


# (d)
fit.9 <- glm(switch ~ dist, data=df,family=binomial(link="logit"))
new_df <- data.frame(dist = df$dist)
df$predict <- predict(fit.9, newdata=new_df)
error.rate <- mean((df$predict > .5 & df$switch == 0) | (df$predict <=.5 & df$switch ==1)

.542053
```





Problem 10

(a)

```
df$log_ars <- log(df$arsenic)
```

```
fit.10 <- glm(switch ~ dist + log_ars + log_ars:dist, data=df,family=binomial(link="logit
display(fit.10)

             coef.est coef.se
(Intercept)   0.49     0.07
dist         -0.01     0.00
log_ars       0.98     0.11
dist:log_ars  0.00     0.00
---
  n = 3020, k = 4
  residual deviance = 3896.8, null deviance = 4118.1 (difference = 221.3)
```

1. (Intercept) indicates that the probability of switching wells for someone with average arsenic levels living an average distance from a clean well would be $\frac{1}{1+e^{-.49+.01\overline{dist}+.98\overline{arsenic}}} = .577$.

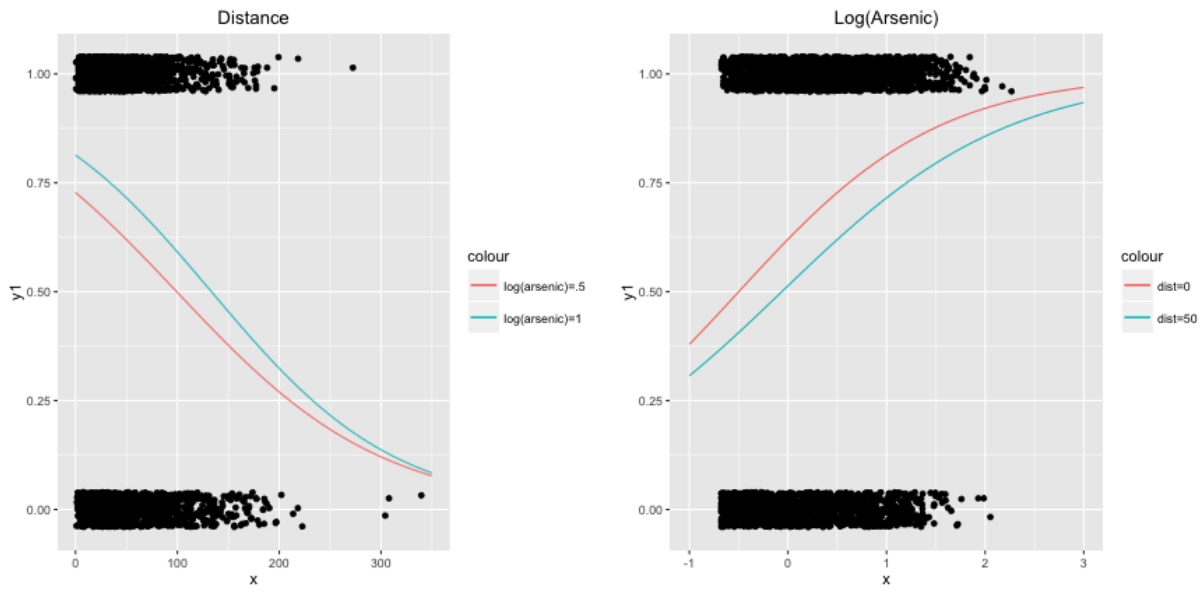
2. (dist) We can evaluate the effect of distance by evaluating the difference in probabilities of two different predictions holding all else constant. I set the value of log_arsenic to its mean and look at the difference in probability for someone who is 48 meters from the nearest clean well and someone who is 49 meters to the nearest clean well. I find that increasing the distance decreases the likelihood of switching by .2%.

3. (log_ars) Lastly we repeat the above analysis but holding distance at it's mean instead and look at the difference between between a log_arsenic of .3 and .4. I find that a log increase of .1 results in a 2% increase in the probability of switching wells.

(b)

```
# Plot Distance with fixed arsenic
x <- seq(0, 35000) / 100
new_df <- data.frame(dist=x, log_ars=1)
y1 <- predict(fit.10, newdata=new_df, type='response')
new_df <- data.frame(dist=x, log_ars=.5)
y2 <- predict(fit.10, newdata=new_df, type='response')
graph_df <- data.frame(x, y1, y2)
p1 <- ggplot(graph_df, aes(x=x), group=rating) +
      geom_line(aes(y = y1, colour = "log(arsenic)=1 ")) +
      geom_line(aes(y = y2, colour = "log(arsenic)=.5")) +
      ggtitle("Distance") +
      geom_point(data=df, aes(x=jitter(dist, .2), y=jitter(switch, .2)))
p1

# Plot Arsenic with fixed distance
x <- seq(-100, 300) / 100
```

```
new_df <- data.frame(dist=0, log_ars=x)
y1 <- predict(fit.10, newdata=new_df, type='response')
new_df <- data.frame(dist=50, log_ars=x)
y2 <- predict(fit.10, newdata=new_df, type='response')
graph_df <- data.frame(x, y1, y2)
p2 <- ggplot(graph_df, aes(x=x), group=rating) +
      geom_line(aes(y = y1, colour = "dist=0")) +
      geom_line(aes(y = y2, colour = "dist=50")) +
      ggtitle("Log(Arsenic)") +
      geom_point(data=df, aes(x=jitter(log_ars, .2), y=jitter(switch, .2)))
p2


grid.arrange(p1, p2, ncol=2)
```



(c)

(i)

```
b <- coef(fit.10)
hi <- 100
low <- 0
delta <- invlogit(b[1] +
                  b[2] * hi +
                  b[3] * df$log_ars +
                  b[4] * df$log_ars * hi) -
         invlogit(b[1] +
```

```
                  b[2] * low +
                  b[3] * df$log_ars +
                  b[4] * df$log_ars * low)
mean(delta)
```

This yields -.211. This means that individuals who are 100 meters from the nearest safe well holding arsenic levels constant are 21% less likely to switch.

(ii)

```
hi <- 200
low <- 100
delta <- invlogit(b[1] +
                  b[2] * hi +
                  b[3] * df$log_ars +
                  b[4] * df$log_ars * hi) -
         invlogit(b[1] +
                  b[2] * low +
                  b[3] * df$log_ars +
                  b[4] * df$log_ars * low)
mean(delta)
```

This yields -.209. which is roughly the same as before. This means that increasing the distance maintains a fairly steady effect on switching.

All remaining analysis is similar and is omitted.

Problem 11

Some of the predictors in 1968 are colinear.