HW 1 Solutions

Paul Moon

Fall 2024

## 7.2 Fake-data simulation and regression:

Simulate 100 data points from the linear model, + error, with , , the values of being sampled at random from a uniform distribution on the range , and errors that are normally distributed with mean 0 and standard deviation 3.

### 7.2a

Fit a regression line to these data and display the output.

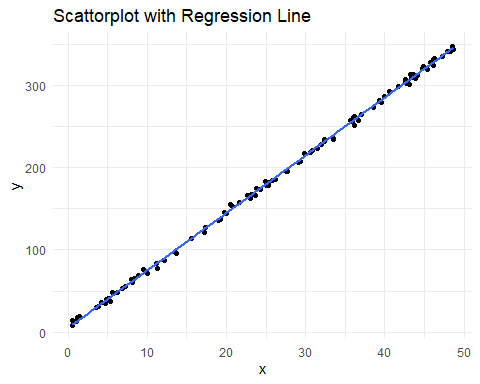
#set.seed(100)  
a <- 5  
b <- 7  
n <- 100  
x <- runif(n, min = 0, max = 50)  
error <- rnorm(n, mean = 0, sd = 3)  
y <- a + b\*x + error  
  
data = data.frame(x, y)  
lm\_model <- stan\_glm(y ~ x, data = data, refresh = 0)  
summary(lm\_model)

##   
## Model Info:  
## function: stan\_glm  
## family: gaussian [identity]  
## formula: y ~ x  
## algorithm: sampling  
## sample: 4000 (posterior sample size)  
## priors: see help('prior\_summary')  
## observations: 100  
## predictors: 2  
##   
## Estimates:  
## mean sd 10% 50% 90%  
## (Intercept) 4.7 0.6 3.9 4.7 5.5   
## x 7.0 0.0 7.0 7.0 7.0   
## sigma 3.0 0.2 2.7 3.0 3.3   
##   
## Fit Diagnostics:  
## mean sd 10% 50% 90%  
## mean\_PPD 188.8 0.4 188.2 188.8 189.3  
##   
## The mean\_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).  
##   
## MCMC diagnostics  
## mcse Rhat n\_eff  
## (Intercept) 0.0 1.0 3434   
## x 0.0 1.0 4373   
## sigma 0.0 1.0 1731   
## mean\_PPD 0.0 1.0 2316   
## log-posterior 0.0 1.0 1391   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

### 7.2b

Graph a scatterplot of the data and the regression line.

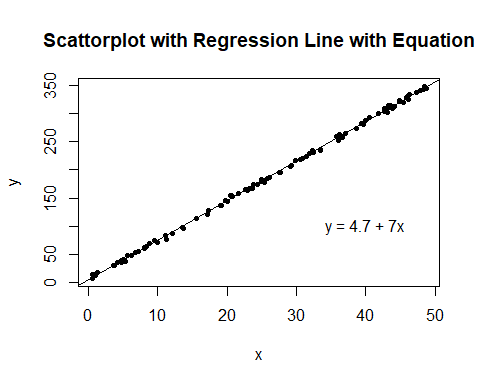
ggplot(data = data.frame(x, y), aes(x, y)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", formula = y ~ x, se = FALSE) +   
 labs(title = "Scattorplot with Regression Line",  
 x = "x",  
 y = "y") +   
 theme\_minimal()



### 7.2c

Use the text function in R to add the formula of the fitted line to the graph.

label = paste("y = ", round(coef(lm\_model)[1], 1), " + ", round(coef(lm\_model)[2], 1), "x", sep = "")  
plot(x, y, pch = 20,main = "Scattorplot with Regression Line with Equation")  
abline(coef(lm\_model))  
text(40,100,label)



## 7.3 Fake-data simulation and fitting the wrong model:

Simulate 100 data points from the model + error, with the values of being sampled at random from a uniform distribution on the range , errors that are normally distributed with mean 0 and standard deviation 3, and , , chosen so that a scatterplot of the data shows a clear nonlinear curve.

### 7.3 a

Fit a regression line stan\_glm(y ~ x) to these data and display the output.

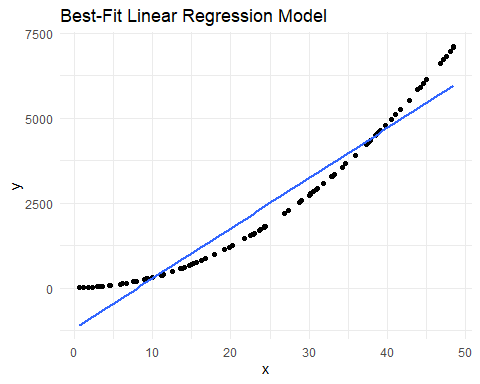
a <- 6  
b <- 1  
c <- 3  
x <- runif(100, min = 0, max = 50)  
y <- rep(0, 100)  
error <- rnorm(100, mean = 0, sd = 3)  
  
for(i in 1:100){  
 y[i] = a + b \* x[i] + c \* x[i]^2 + error[i]  
}  
  
data = data.frame(x = x, y = y)  
stanGlm <- stan\_glm(y ~ x, data = data, refresh = 0)  
summary(stanGlm)

##   
## Model Info:  
## function: stan\_glm  
## family: gaussian [identity]  
## formula: y ~ x  
## algorithm: sampling  
## sample: 4000 (posterior sample size)  
## priors: see help('prior\_summary')  
## observations: 100  
## predictors: 2  
##   
## Estimates:  
## mean sd 10% 50% 90%   
## (Intercept) -1190.4 105.8 -1326.0 -1192.2 -1054.1  
## x 147.6 3.8 142.7 147.7 152.5  
## sigma 534.1 37.8 487.8 531.5 585.2  
##   
## Fit Diagnostics:  
## mean sd 10% 50% 90%   
## mean\_PPD 2346.5 75.7 2251.0 2345.7 2445.0  
##   
## The mean\_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).  
##   
## MCMC diagnostics  
## mcse Rhat n\_eff  
## (Intercept) 1.7 1.0 3857   
## x 0.1 1.0 3975   
## sigma 0.6 1.0 3493   
## mean\_PPD 1.2 1.0 3673   
## log-posterior 0.0 1.0 1756   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

### 7.3b

Graph a scatterplot of the data and the regression line. This is the best-fit linear regression. What does “best-fit” mean in this context?

ggplot(data = data.frame(x, y), aes(x, y)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", formula = y ~ x, se = FALSE) +   
 labs(title = "Best-Fit Linear Regression Model",  
 x = "x",  
 y = "y") +   
 theme\_minimal()



#The best-fit would mean the linear regression with the lowest MSE in the data set.

## 7.6 Formulating comparisons as regression models:

Take the election forecasting model and simplify it by creating a binary predictor defined as if income growth is less than 2% and if income growth is more than 2%.

hibbs <- read.table("https://raw.githubusercontent.com/avehtari/ROS-Examples/master/ElectionsEconomy/data/hibbs.dat", header=TRUE)  
  
hibbs$x <- ifelse(hibbs$growth >= 2,1,0)  
hibbs

## year growth vote inc\_party\_candidate other\_candidate x  
## 1 1952 2.40 44.60 Stevenson Eisenhower 1  
## 2 1956 2.89 57.76 Eisenhower Stevenson 1  
## 3 1960 0.85 49.91 Nixon Kennedy 0  
## 4 1964 4.21 61.34 Johnson Goldwater 1  
## 5 1968 3.02 49.60 Humphrey Nixon 1  
## 6 1972 3.62 61.79 Nixon McGovern 1  
## 7 1976 1.08 48.95 Ford Carter 0  
## 8 1980 -0.39 44.70 Carter Reagan 0  
## 9 1984 3.86 59.17 Reagan Mondale 1  
## 10 1988 2.27 53.94 Bush, Sr. Dukakis 1  
## 11 1992 0.38 46.55 Bush, Sr. Clinton 0  
## 12 1996 1.04 54.74 Clinton Dole 0  
## 13 2000 2.36 50.27 Gore Bush, Jr. 1  
## 14 2004 1.72 51.24 Bush, Jr. Kerry 0  
## 15 2008 0.10 46.32 McCain Obama 0  
## 16 2012 0.95 52.00 Obama Romney 0

### 7.6a

Compute the difference in incumbent party’s vote share on average, comparing those two groups of elections, and determine the standard error for this difference.

x0 <- hibbs[hibbs$x == 1,]$vote  
x1 <- hibbs[hibbs$x == 0,]$vote  
  
avgDiff <- mean(x0) - mean(x1)  
cat("The difference in incumbent party's vote share on average:", avgDiff)

## The difference in incumbent party's vote share on average: 5.5075

c0 <- length(x0)  
c1 <- length(x1)  
StdDev <- (sqrt((var(x0) \* (c0 - 1) + var(x1) \* (c1 - 1)) / (c0 + c1 - 2)))  
StdErr <- StdDev \* sqrt(1 / c0 + 1 / c1)  
cat("\nThe standard error for this difference: ", StdErr)

##   
## The standard error for this difference: 2.502052

### 7.6b

Regress incumbent party’s vote share on the binary predictor of income growth and check that the resulting estimate and standard error are the same as above.

lm\_model <- lm(vote ~ x, data = hibbs)  
summary(lm\_model)

##   
## Call:  
## lm(formula = vote ~ x, data = hibbs)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.2087 -3.3706 0.1287 3.3037 6.9812   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 49.301 1.769 27.866 1.15e-13 \*\*\*  
## x 5.508 2.502 2.201 0.045 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.004 on 14 degrees of freedom  
## Multiple R-squared: 0.2571, Adjusted R-squared: 0.204   
## F-statistic: 4.845 on 1 and 14 DF, p-value: 0.045

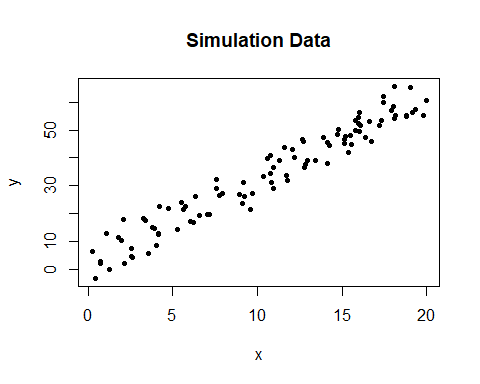
## 8.8 Comparing lm and stan\_glm:

Use simulated data to compare least squares estimation to default Bayesian regression:

### 8.8a

Simulate 100 data points from the model, + error, with predictors drawn from a uniform distribution from 0 to 20 and with independent errors drawn from the normal distribution with mean 0 and standard deviation 5. Fit the regression of on data using lm and stan\_glm (using its default settings) and check that the two programs give nearly identical results.

n <- 100  
x <- runif(n, min = 0, max = 20)  
error <- rnorm(n, mean = 0, sd = 5)  
y <- 2 + 3 \* x + error  
  
data = data.frame(x, y)  
plot(x, y, pch = 20, main = "Simulation Data")



lm\_model <- lm(y ~ x , data = data)  
stanGlm <- stan\_glm(y ~ x, data = data, refresh = 0)  
lm\_model

##   
## Call:  
## lm(formula = y ~ x, data = data)  
##   
## Coefficients:  
## (Intercept) x   
## 2.234 2.995

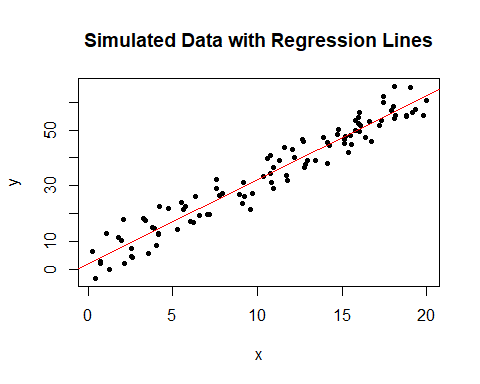
summary(stanGlm)

##   
## Model Info:  
## function: stan\_glm  
## family: gaussian [identity]  
## formula: y ~ x  
## algorithm: sampling  
## sample: 4000 (posterior sample size)  
## priors: see help('prior\_summary')  
## observations: 100  
## predictors: 2  
##   
## Estimates:  
## mean sd 10% 50% 90%  
## (Intercept) 2.2 0.9 1.1 2.3 3.5   
## x 3.0 0.1 2.9 3.0 3.1   
## sigma 4.5 0.3 4.1 4.5 4.9   
##   
## Fit Diagnostics:  
## mean sd 10% 50% 90%  
## mean\_PPD 34.1 0.6 33.3 34.1 34.9   
##   
## The mean\_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).  
##   
## MCMC diagnostics  
## mcse Rhat n\_eff  
## (Intercept) 0.0 1.0 4392   
## x 0.0 1.0 4419   
## sigma 0.0 1.0 3830   
## mean\_PPD 0.0 1.0 3456   
## log-posterior 0.0 1.0 1955   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

### 8.8b

Plot the simulated data and the two fitted regression lines.

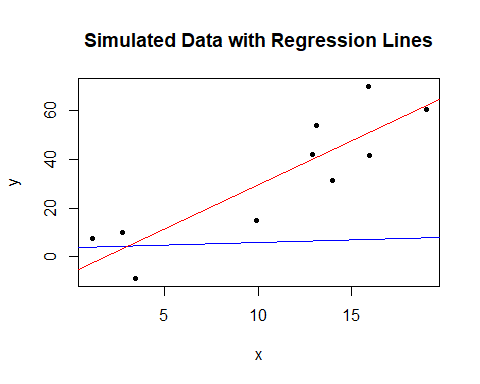
plot(x, y, pch = 20, main = "Simulated Data with Regression Lines")  
abline(coef(lm\_model), col = "blue")  
abline(coef(stanGlm), col = "red")



### 8.8c

Repeat the two steps above, but try to create conditions for your simulation so that lm and stan\_glm give much different results.

n <- 10  
x <- runif(n, min = 0, max = 20)  
error <- rnorm(n, mean = 0, sd = 10)  
y <- 2 + 3 \* x + error  
  
data = data.frame(x, y)  
  
lm\_model <- lm(x ~ y, data = data)  
stanGlm <- stan\_glm(y ~ x, data = data, refresh = 0)  
plot(x, y, pch = 20, main = "Simulated Data with Regression Lines")  
abline(coef(lm\_model), col = "blue")  
abline(coef(stanGlm), col = "red")



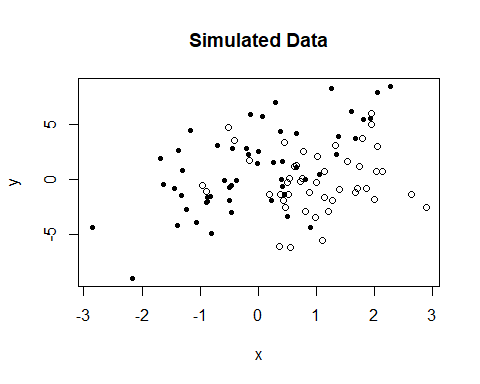
## 10.1 Regression with interactions:

Simulate 100 data points from the model, + error, with a continuous predictor and a binary predictor , coefficients , and errors drawn independently from a normal distribution with mean 0 and standard deviation 3, as follows. For each data point , first draw , equally likely to take on the values 0 and 1. Then draw from a normal distribution with mean and standard deviation 1. Then draw the error from its normal distribution and compute .

### 10.1a

Display your simulated data as a graph of vs , using dots and circles for the points with = 0 and 1, respectively.

b <- c(1, 2, -1, -2)  
error = rnorm(100, mean = 0, sd = 3)  
z <- rbinom(100, size = 1, prob = 0.5)  
x <- rnorm(100, mean = z, sd = 1)  
  
y = b[1] + b[2] \* x + b[3] \* z + b[4] \* x \* z + error  
  
shape = ifelse(z == 1, 1, 20)  
plot(x, y, pch = shape, main = "Simulated Data")



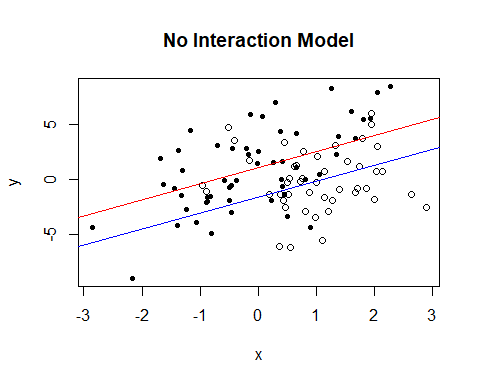
### 10.1b

Fit a regression predicting from and with no interaction. Make a graph with the data and two parallel lines showing the fitted model.

data = data.frame(y = y, x = x, z = z)  
stanGlm <- stan\_glm(y ~ z + x, data = data, refresh = 0)  
stanGlm

## stan\_glm  
## family: gaussian [identity]  
## formula: y ~ z + x  
## observations: 100  
## predictors: 3  
## ------  
## Median MAD\_SD  
## (Intercept) 1.1 0.4   
## z -2.7 0.7   
## x 1.5 0.3   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 3.0 0.2   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

stan <- coef(stanGlm)  
  
plot(x, y, pch = shape, main = "No Interaction Model")  
abline(stan[1] + stan[2], stan[3], col = "blue")  
abline(stan[1], stan[3], col = "red")



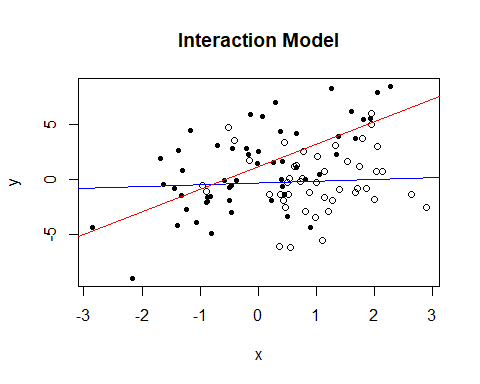
### 10.1c

Fit a regression predicting from , , and their interaction. Make a graph with the data and two lines showing the fitted model.

stanGlm <- stan\_glm(y ~ z + x + z : x,data = data, refresh = 0)  
stanGlm

## stan\_glm  
## family: gaussian [identity]  
## formula: y ~ z + x + z:x  
## observations: 100  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 1.1 0.4   
## z -1.4 0.7   
## x 2.1 0.4   
## z:x -1.9 0.6   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 2.9 0.2   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

stan <- coef(stanGlm)  
  
plot(x, y, pch = shape, main = "Interaction Model")  
abline(stan[1] + stan[2], stan[3] + stan[4], col="blue")  
abline(stan[1], stan[3], col="red")



## 10.2 Regression with interactions:

Here is the output from a fitted linear regression of outcome on pre-treatment predictor , treatment indicator , and their interaction:

Mediam MAD\_SD  
(Intercept) 1.2 0.2  
x 1.6 0.4  
z 2.7 0.3  
x:z 0.7 0.5  
  
Auxiliary parameter(s):  
 Median MAD\_SD  
sigma 0.4 0.0

### 10.2a

Write the equation of the estimated regression line of on for the treatment group and the control group, and the equation of the estimated regression line of on for the control group.

data = data.frame(Median = c(1.2, 1.6, 2.7, 0.7), MAD\_SD = c(0.2, 0.4, 0.3, 0.5))  
row.names(data) = c("(Intercept)","x","z","x:z")  
data

## Median MAD\_SD  
## (Intercept) 1.2 0.2  
## x 1.6 0.4  
## z 2.7 0.3  
## x:z 0.7 0.5

med\_int <- data[c("(Intercept)"), c("Median")]  
med\_x <- data[c("x"), c("Median")]  
med\_z <- data[c("z"), c("Median")]  
med\_xz <- data[c("x:z"), c("Median")]  
cat("The equation for the treatment group is: y(treatment) =", med\_int + med\_z, "+", med\_x + med\_xz, "x\n")

## The equation for the treatment group is: y(treatment) = 3.9 + 2.3 x

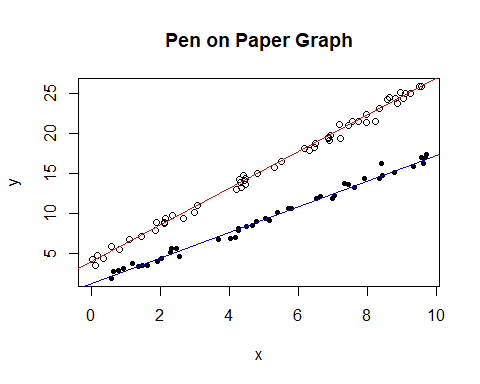
cat("The equation for the treatment group is: y(control) =", med\_int, "+", med\_x, "x")

## The equation for the treatment group is: y(control) = 1.2 + 1.6 x

### 10.2b

Graph with pen on paper the two regression lines, assuming the values of fall in the range . On this graph also include a scatterplot of data (using open circles for treated units and dots for controls) that are consistent with the fitted model.

a <- data$Median  
x = runif(100, min = 0, max = 10)  
z = rbinom(100, size = 1, prob = 0.5)  
error = rnorm(100, mean = 0, sd = 0.5)  
  
y = ifelse(z==0, a[1] + a[2] \* x + error, a[1] + a[3] + (a[2] + a[4]) \* x + error)  
  
shape = ifelse(z==0, 20, 1)  
plot(x, y, pch = shape, main = "Pen on Paper Graph")  
abline(a[1], a[2], col="blue")  
abline(a[1] + a[3], a[2] + a[4], col="red")



## 10.5 Regression modeling and prediction:

The folder KidIQ contains a subset of the children and mother data discussed earlier in the chapter. You have access to children’s test scores at age 3, mother’s education, and the mother’s age at the time she gave birth for a sample of 400 children.

kidiq <- read.csv("kidiq.csv", header = TRUE)

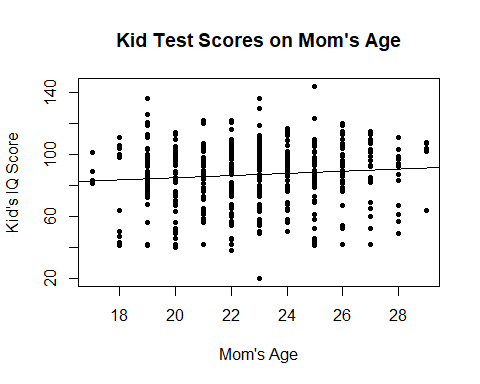
### 10.5a

Fit a regression of child test scores on mother’s age, display the data and fitted model, check assumptions, and interpret the slope coefficient. Based on this analysis, when do you recommend mothers should give birth? What are you assuming in making this recommendation?

stanGlm <- stan\_glm(kid\_score ~ mom\_age, data = kidiq, refresh = 0)  
stanGlm

## stan\_glm  
## family: gaussian [identity]  
## formula: kid\_score ~ mom\_age  
## observations: 434  
## predictors: 2  
## ------  
## Median MAD\_SD  
## (Intercept) 70.7 8.1   
## mom\_age 0.7 0.4   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 20.4 0.7   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

plot(kidiq$mom\_age, kidiq$kid\_score, pch = 20, main = "Kid Test Scores on Mom's Age",  
 xlab = "Mom's Age",  
 ylab = "Kid's IQ Score")  
abline(coef(stanGlm))



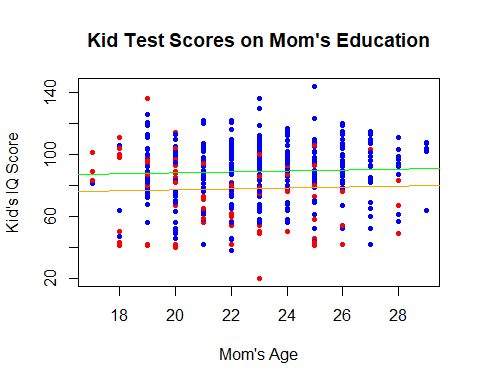
### 10.5b

Repeat this for a regression that further includes mother’s education, interpreting both slope coefficients in this model. Have your conclusions about the timing of birth changed?

stanGlm <- stan\_glm(kid\_score ~ mom\_age + mom\_hs, data = kidiq, refresh = 0)  
stanGlm

## stan\_glm  
## family: gaussian [identity]  
## formula: kid\_score ~ mom\_age + mom\_hs  
## observations: 434  
## predictors: 3  
## ------  
## Median MAD\_SD  
## (Intercept) 70.6 8.2   
## mom\_age 0.3 0.4   
## mom\_hs 11.3 2.3   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 19.9 0.7   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

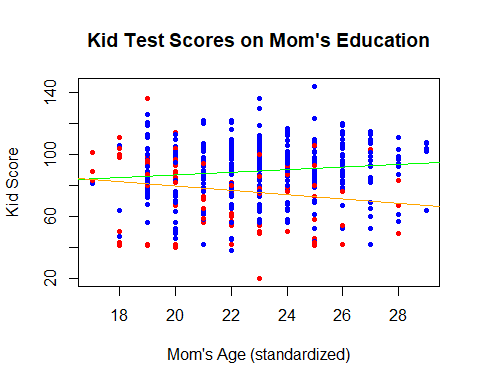
a <- coef(stanGlm)  
  
colors = ifelse(kidiq$mom\_hs == 1, "blue", "red")  
plot(kidiq$mom\_age, kidiq$kid\_score, col = colors, pch = 20, main = "Kid Test Scores on Mom's Education",  
 xlab = "Mom's Age",  
 ylab = "Kid's IQ Score")  
abline(a[1] + a[3], a[2], col = "green")  
abline(a[1], a[2], col = "orange")



### 10.5c

Now create an indicator variable reflecting whether the mother has completed high school or not. Consider interactions between high school completion and mother’s age. Also create a plot that shows the separate regression lines for each high school completion status group.

stanGlm <- stan\_glm(kid\_score ~ mom\_age + mom\_hs + mom\_age:mom\_hs, data = kidiq, refresh = 0)  
  
a <- coef(stanGlm)  
  
plot(kidiq$mom\_age, kidiq$kid\_score, col = colors, pch=20, main = "Kid Test Scores on Mom's Education",  
 xlab="Mom's Age (standardized)",   
 ylab="Kid Score")  
abline(a[1] + a[3], a[2] + a[4],col="green")  
abline(a[1], a[2],col="orange")



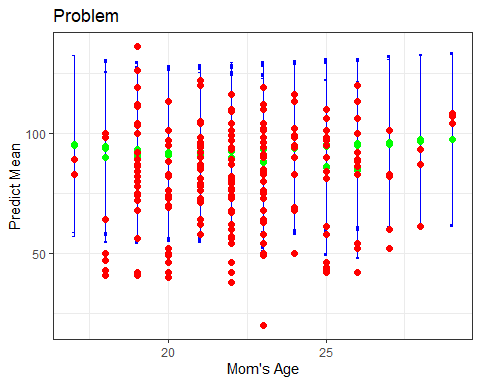
### 10.5d

Finally, fit a regression of child test scores on mother’s age and education level for the first 200 children and use this model to predict test scores for the next 200. Graphically display comparisons of the predicted and actual scores for the final 200 children.

obs = kidiq[1:200, ]  
pre = kidiq[201:400, ]  
stanGlm <- stan\_glm(kid\_score ~ mom\_age + mom\_hs + mom\_age:mom\_hs, data = obs, refresh = 0)  
stanGlm

## stan\_glm  
## family: gaussian [identity]  
## formula: kid\_score ~ mom\_age + mom\_hs + mom\_age:mom\_hs  
## observations: 200  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 115.6 22.5   
## mom\_age -1.2 1.0   
## mom\_hs -36.9 24.8   
## mom\_age:mom\_hs 1.8 1.1   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 17.5 0.9   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

b <- posterior\_predict(stanGlm, pre)  
  
pre$predict\_mean <- apply(b, 2, mean)  
pre$predict\_sd <- apply(b, 2, sd)  
  
ggplot(pre) +   
 aes(x = mom\_age) +   
 geom\_errorbar(aes(ymin = predict\_mean - 2 \* predict\_sd, ymax = predict\_mean + 2 \* predict\_sd), color = "blue", width = .1) +   
 geom\_point(aes(y = predict\_mean), color = "green", size = 2) +   
 geom\_point(aes(y = kid\_score), color = "red", size = 2) +   
 theme\_bw() +   
 labs(title = "Problem",   
 x = "Mom's Age",   
 y = "Predict Mean")



## 10.6 Regression models with interactions:

The folder Beauty contains data (use file beauty.csv) from Hamermesh and Parker (2005) on student evaluations of instructors’ beauty and teaching quality for several courses at the University of Texas. The teaching evaluations were conducted at the end of the semester, and the beauty judgments were made later, by six students who had not attended the classes and were not aware of the course evaluations.

See also Felton, Mitchell, and Stinson (2003) for more on this topic.

beauty <- read.csv("beauty.csv", header = TRUE)

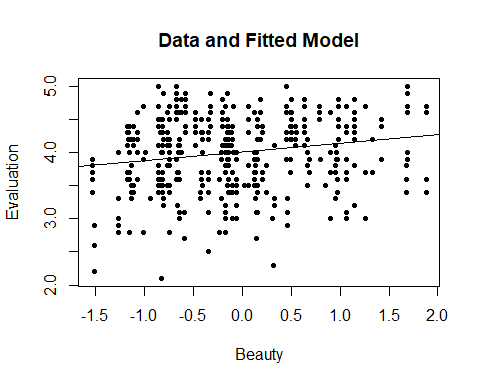
### 10.6a

Run a regression using beauty (the variable beauty) to predict course evaluations (eval), adjusting for various other predictors. Graph the data and fitted model, and explain the meaning of each of the coefficients along with the residual standard deviation. Plot the residuals versus fitted values.

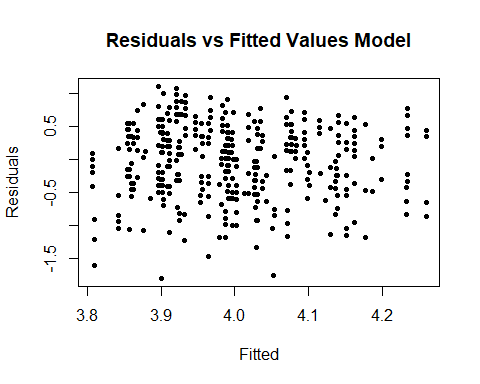
stanGlm <- stan\_glm(eval ~ beauty, data = beauty, refresh = 0)  
stanGlm

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty  
## observations: 463  
## predictors: 2  
## ------  
## Median MAD\_SD  
## (Intercept) 4.0 0.0   
## beauty 0.1 0.0   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.5 0.0   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

plot(beauty$beauty, beauty$eval, pch = 20, main = "Data and Fitted Model",   
 xlab = "Beauty",   
 ylab = "Evaluation")  
abline(coef(stanGlm))



plot(fitted(stanGlm), resid(stanGlm), pch = 20, main = "Residuals vs Fitted Values Model",   
 xlab = "Fitted",   
 ylab = "Residuals")



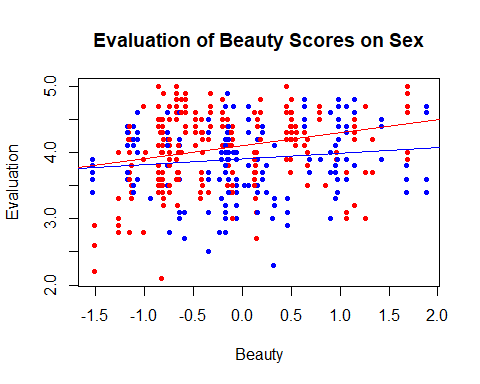
### 10.6b

Fit some other models, including beauty and also other predictors. Consider at least one model with interactions. For each model, explain the meaning of each of its estimated coefficients.

stanGlm1 <- stan\_glm(eval ~ beauty + female + beauty:female, data = beauty, refresh = 0)  
stanGlm1

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty + female + beauty:female  
## observations: 463  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 4.1 0.0   
## beauty 0.2 0.0   
## female -0.2 0.1   
## beauty:female -0.1 0.1   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.5 0.0   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

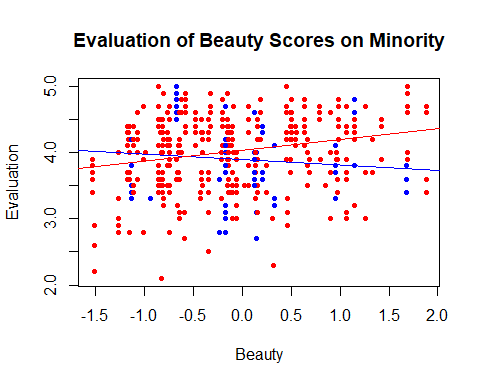
a <- coef(stanGlm1)  
colGraph1 <- ifelse(beauty$female == 1, "blue", "red")  
plot(beauty$beauty, beauty$eval, pch = 20, main = "Evaluation of Beauty Scores on Sex", col = colGraph1,  
 xlab = "Beauty",   
 ylab = "Evaluation")  
abline(a[1] + a[3], a[2] + a[4], col = "blue")  
abline(a[1], a[2], col = "red")



stanGlm2 <- stan\_glm(eval ~ beauty + minority + beauty:minority, data = beauty, refresh = 0)  
stanGlm2

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty + minority + beauty:minority  
## observations: 463  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 4.0 0.0   
## beauty 0.2 0.0   
## minority -0.1 0.1   
## beauty:minority -0.2 0.1   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.5 0.0   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

b <- coef(stanGlm2)  
colGraph2 <- ifelse(beauty$minority == 1, "blue", "red")  
plot(beauty$beauty, beauty$eval, pch = 20, main = "Evaluation of Beauty Scores on Minority", col = colGraph2,   
 xlab = "Beauty",   
 ylab = "Evaluation")  
abline(b[1] + b[3], b[2] + b[4], col = "blue")  
abline(b[1], b[2], col = "red")



cat("\n Males have an average beauty evaluation of 4.1 while females have an average beauty evaluation of 3.9. The slope of 0.2 for males means that for every 1 point of beauty score means a evaluation of 0.2 more. However; for females that slope is brought down to 0.1")

##   
## Males have an average beauty evaluation of 4.1 while females have an average beauty evaluation of 3.9. The slope of 0.2 for males means that for every 1 point of beauty score means a evaluation of 0.2 more. However; for females that slope is brought down to 0.1

cat("\n Non-minorites have an average beauty evaluation of 4.0 while minorities have an average evaluation of 3.9. The slope of 0.1 for non- minorities means that for every 1 point of beauty score means a evaluation of 0.1 more. However; for minorities that slope is brough down to -0.1")

##   
## Non-minorites have an average beauty evaluation of 4.0 while minorities have an average evaluation of 3.9. The slope of 0.1 for non- minorities means that for every 1 point of beauty score means a evaluation of 0.1 more. However; for minorities that slope is brough down to -0.1

## 10.7 Predictive simulation for linear regression:

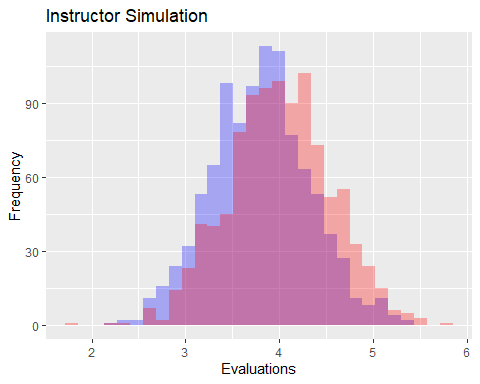
Take one of the models from the previous exercise.

### 10.7a

Instructor A is a 50-year-old woman who is a native English speaker and has a beauty score of -1. Instructor B is a 60-year-old man who is a native English speaker and has a beauty score of -0.5. Simulate 1000 random draws of the course evaluation rating of these two instructors. In your simulation, use posterior\_predict to account for the uncertainty in the regression parameters as well as predictive uncertainty.

#We use stanGlm1 since one instructor is male and another one is female.  
  
IA <- data.frame(beauty = -1, female = 1, age = 50, minority = 0, nonenglish = 0)  
IB <- data.frame(beauty = -0.5, female = 0, age = 60, minority = 0, nonenglish = 0)  
  
SA <- posterior\_predict(stanGlm1, newdata = IA, draws = 1000)  
SB <- posterior\_predict(stanGlm1, newdata = IB, draws = 1000)  
  
ggplot() +   
 geom\_histogram(aes(SA[, 1]), fill = "blue", alpha = 0.3) +   
 geom\_histogram(aes(SB[, 1]), fill = "red", alpha = 0.3) +   
 labs(title = "Instructor Simulation",   
 x = "Evaluations",   
 y = "Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



### 10.7b

Make a histogram of the difference between the course evaluations for A and B. What is the probability that A will have a higher evaluation?

#Histogram shown in Problem 10.7a  
  
pA <- c(apply(SA, 2, mean), apply(SA, 2, sd))  
pB <- c(apply(SB, 2, mean), apply(SB, 2, sd))  
  
pC <- c(pA[1] - pB[1], sqrt(pA[2] ^ 2 + pB[2] ^ 2))  
pD <- 1 - pnorm(0, pC[1], pC[2])  
cat("The probability that A will have a higher evaluation is", pD)

## The probability that A will have a higher evaluation is 0.3937449

## 10.8 How many simulation draws:

Take the model from Exercise 10.6 that predicts course evaluations from beauty and other predictors.

### 10.8a

Display and discuss the fitted model. Focus on the estimate and standard error for the coefficient of beauty.

#This is from Problem 10.6b  
stanGlm1 <- stan\_glm(eval ~ beauty + female + beauty:female, data = beauty, refresh = 0)  
stanGlm1

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty + female + beauty:female  
## observations: 463  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 4.1 0.0   
## beauty 0.2 0.0   
## female -0.2 0.1   
## beauty:female -0.1 0.1   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.5 0.0   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

### 10.8b

Compute the median and mad sd of the posterior simulations of the coefficient of beauty, and check that these are the same as the output from printing the fit.

s <- as.matrix(stanGlm1)  
MEDIAN <- apply(s, 2, median)  
MAD\_SD <- apply(s, 2, mad)  
ss <- cbind(round(MEDIAN, 1), round(MAD\_SD, 1))  
colnames(ss) <- c("MEDIAN", "MAD\_SD")  
ss

## MEDIAN MAD\_SD  
## (Intercept) 4.1 0.0  
## beauty 0.2 0.0  
## female -0.2 0.1  
## beauty:female -0.1 0.1  
## sigma 0.5 0.0

### 10.8c

Fit again, this time setting iter = 1000 in your stan\_glm call. Do this a few times in order to get a sense of the simulation variability.

stanGlm1 <- stan\_glm(eval ~ beauty + female + beauty:female, data = beauty, iter = 1000, refresh = 0)  
stanGlm1

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty + female + beauty:female  
## observations: 463  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 4.1 0.0   
## beauty 0.2 0.0   
## female -0.2 0.1   
## beauty:female -0.1 0.1   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.5 0.0   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

### 10.8d

Repeat the previous step, setting iter = 100 and then iter = 10.

stanGlm2 <- stan\_glm(eval ~ beauty + female + beauty:female, data = beauty, iter = 100, refresh = 0)

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be unreliable.  
## Running the chains for more iterations may help. See  
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quantiles may be unreliable.  
## Running the chains for more iterations may help. See  
## https://mc-stan.org/misc/warnings.html#tail-ess

stanGlm2

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty + female + beauty:female  
## observations: 463  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) 4.1 0.0   
## beauty 0.2 0.0   
## female -0.2 0.1   
## beauty:female -0.1 0.1   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.5 0.0   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

stanGlm3 <- stan\_glm(eval ~ beauty + female + beauty:female, data = beauty, iter = 10, refresh = 0)

## Warning: There were 15 divergent transitions after warmup. See  
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup  
## to find out why this is a problem and how to eliminate them.

## Warning: There were 1 chains where the estimated Bayesian Fraction of Missing Information was low. See  
## https://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 15.28, indicating chains have not mixed.  
## Running the chains for more iterations may help. See  
## https://mc-stan.org/misc/warnings.html#r-hat

## Warning: Markov chains did not converge! Do not analyze results!

stanGlm3

## stan\_glm  
## family: gaussian [identity]  
## formula: eval ~ beauty + female + beauty:female  
## observations: 463  
## predictors: 4  
## ------  
## Median MAD\_SD  
## (Intercept) -1.6 1.8   
## beauty 0.9 0.4   
## female 2.8 2.4   
## beauty:female -0.3 2.6   
##   
## Auxiliary parameter(s):  
## Median MAD\_SD  
## sigma 0.8 0.1   
##   
## ------  
## \* For help interpreting the printed output see ?print.stanreg  
## \* For info on the priors used see ?prior\_summary.stanreg

### 10.8e

How many simulations were needed to give a good approximation to the mean and standard error for the coefficient of beauty?

There needs to be at least 1000 simulations to create a good approximation to the mean and standard error for the coefficient of beauty.