HW 2

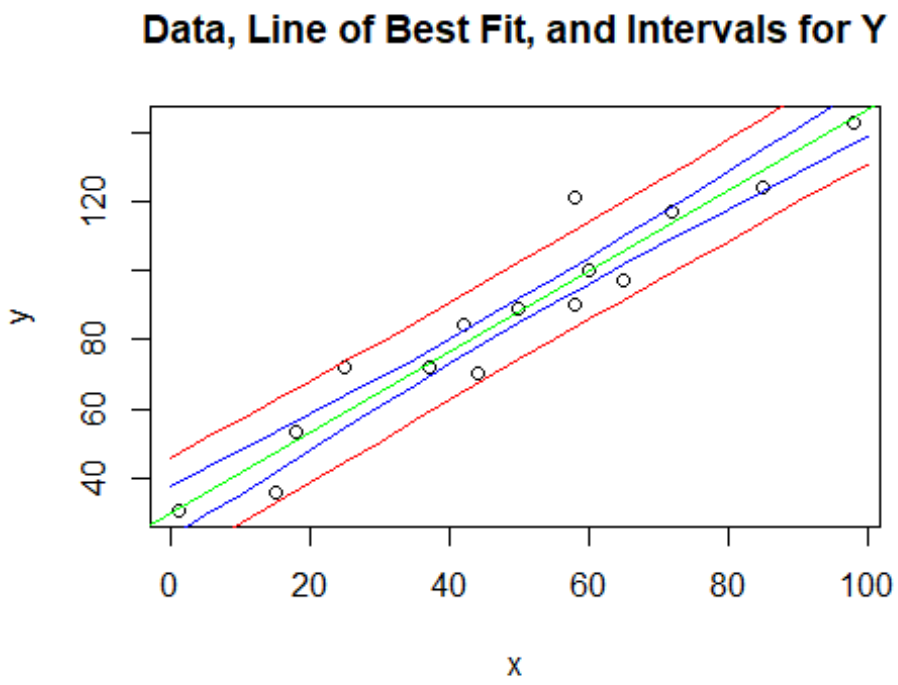
Nathan Hawkins

1/26/2021

1

```
linmdl1 <- "
  model {

    for(i in 1:15){
      y[i] ~ dnorm(mu[i], prec)
      mu[i] <- b0 + b1*x[i]
    }

    #Priors
    b0 ~ dnorm(0, .001)
    b1 ~ dnorm(0, .002)
    prec <- 1/vr

    #Variance
    vr ~ dgamma(1.1, .1667)

  }
"



x <- hw2$x
y <- hw2$y

writeLines(linmdl1, 'linreg.txt')


data.jags <- c('x', 'y')
parms <- c('b0', 'b1', 'vr')

linreg.sim1 <- jags(data.jags,
                    inits = NULL,
                    parameters.to.save = parms,
                    model.file = 'linreg.txt',
                    n.iter = 35000,
                    n.burnin = 10000,
                    n.chains = 4,
                    n.thin = 10)
```

```r
# Intermediary steps

sims <- as.mcmc(linreg.sim1)
chains <- as.matrix(sims)


b0 = chains[,1]
b1 = chains[,2]
vr = chains[,4]


#Make some data
xx <- seq(0,100,by = 5)

#Find Fitted values
#Fit each x value

#Make a matrix
fittedy <- matrix(0, 10000, 21)

#For loop that follows intercept + slope * data
for(i in 1:10000){
  fittedy[i,] <- b0[i] + b1[i] * xx
}


#Find Predicted Values
#Make matrix to hold predicted values
predicted_y <- matrix(0, 10000, 21)

#For loop that follows intercept + slope * data + noise
for(i in 1:10000){
  predicted_y[i,] <- b0[i] + b1[i] * xx + rnorm(1,0,sqrt(vr[i]))
}

#Plot line of best fit
plot(x,y, main = "Data, Line of Best Fit, and Intervals for Y")
abline(mean(b0), mean(b1), col = 'green')

#Plot Posterior Interval
lines(xx,apply(fittedy, 2, quantile, 0.975), col = 'blue')
lines(xx,apply(fittedy, 2, quantile, 0.025), col = 'blue')

#Plot Posterior Predicted Interval
lines(xx,apply(predicted_y, 2, quantile, 0.975), col = 'red')
lines(xx,apply(predicted_y, 2, quantile, 0.025), col = 'red')
```

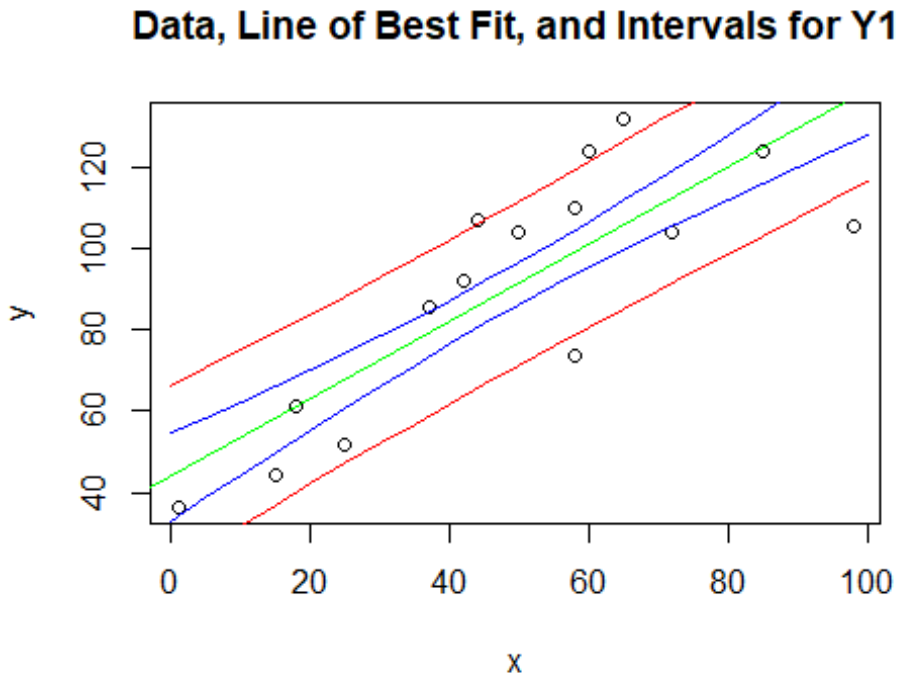


2 Y1

```
linmdl1 <- "
  model {

    for(i in 1:15){
      y[i] ~ dnorm(mu[i], prec)
      mu[i] <- b0 + b1*x[i]
    }

    #Priors
    b0 ~ dnorm(0, .001)
    b1 ~ dnorm(0, .002)
    prec <- 1/vr

    #Variance
    vr ~ dgamma(1.1, .1667)

  }
"

x <- hw2$x
y <- hw2$y1
```

```r
writeLines(linmdl1, 'linreg.txt')


data.jags <- c('x', 'y')
parms <- c('b0', 'b1', 'vr')

linreg.sim1 <- jags(data.jags,
                    inits = NULL,
                    parameters.to.save = parms,
                    model.file = 'linreg.txt',
                    n.iter = 35000,
                    n.burnin = 10000,
                    n.chains = 4,
                    n.thin = 10)

# Intermediary steps

sims <- as.mcmc(linreg.sim1)
chains <- as.matrix(sims)


b0 = chains[,1]
b1 = chains[,2]
vr = chains[,4]


#Make some data
xx <- seq(0,100,by = 5)

#Find Fitted values
#Fit each x value

#Make a matrix
fittedy <- matrix(0, 10000, 21)

#For loop that follows intercept + slope * data
for(i in 1:10000){
  fittedy[i,] <- b0[i] + b1[i] * xx
}


#Find Predicted Values
#Make matrix to hold predicted values
predicted_y <- matrix(0, 10000, 21)

#For loop that follows intercept + slope * data + noise
for(i in 1:10000){
  predicted_y[i,] <- b0[i] + b1[i] * xx + rnorm(1,0,sqrt(vr[i]))
}
```

```r
#Plot line of best fit
plot(x,y, main = "Data, Line of Best Fit, and Intervals for Y1")
abline(mean(b0), mean(b1), col = 'green')

#Plot Posterior Interval
lines(xx,apply(fittedy, 2, quantile, 0.975), col = 'blue')
lines(xx,apply(fittedy, 2, quantile, 0.025), col = 'blue')

#Plot Posterior Predicted Interval
lines(xx,apply(predicted_y, 2, quantile, 0.975), col = 'red')
lines(xx,apply(predicted_y, 2, quantile, 0.025), col = 'red')
```



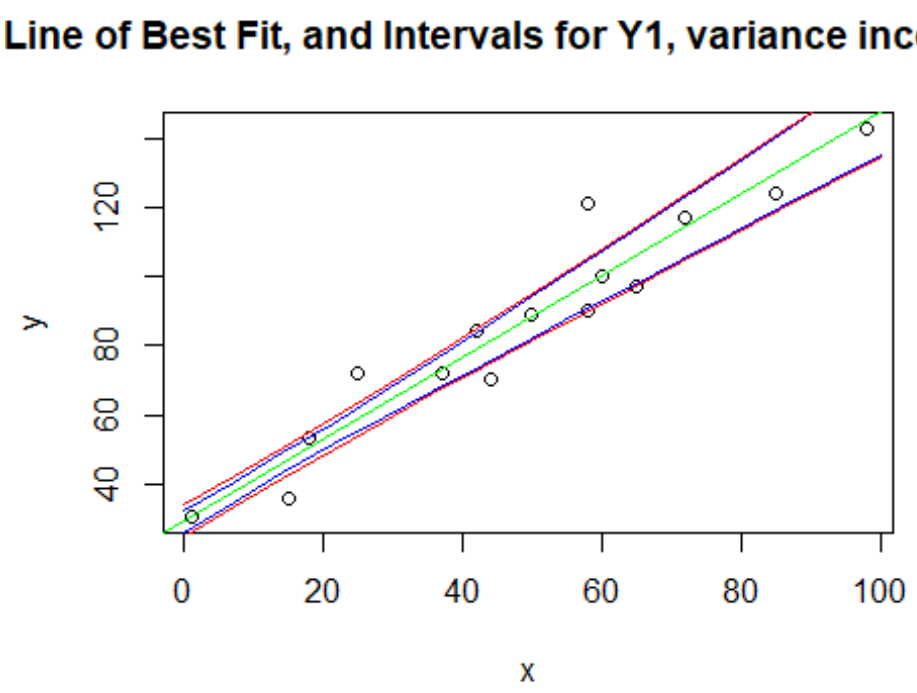
3 Let Variances increase linearly with x

For hw make variance[i] = a0 + a1 * x var[i] <- a0 + a1 * x we need priors for a0 and a1 draw a0 and a1 from a uniform distribution from 0 to 1000 maybe

```r
linmdl3 <- "
  model {

    for(i in 1:15){
      y[i] ~ dnorm(mu[i], 1/s2[i])
      mu[i] <- b0 + b1*x[i]
      s2[i] = a1 * x[i]
    }
```

```
    #Priors
    b0 ~ dnorm(0, .001)
    b1 ~ dnorm(0, .002)
    a1 ~ dgamma(1, .5)


  }
"

x <- hw2$x
y <- hw2$y

writeLines(linmdl3, 'linreg.txt')


data.jags <- c('x', 'y')
parms <- c('b0', 'b1', 'a1', 's2')

linreg.sim3 <- jags(data.jags,
                    inits = NULL,
                    parameters.to.save = parms,
                    model.file = 'linreg.txt',
                    n.iter = 35000,
                    n.burnin = 10000,
                    n.chains = 4,


library(R2jags)
# Intermediary steps

sims <- as.mcmc(linreg.sim3)
chains <- as.matrix(sims)


b0 = chains[,2]
b1 = chains[,3]

s2 = chains[,c(5,12:19,6:11)]


#Make some data
xx <- seq(0,100,length.out = 15)

#Find Fitted values
#Fit each x value

#Make a matrix
fittedy <- matrix(0, 10000, 15)
```

```r
#For loop that follows intercept + slope * data
for(i in 1:10000){
  for(j in 1:15){
      fittedy[i,] <- b0[i] + b1[i] * xx
  }
}

#Find Predicted Values
#Make matrix to hold predicted values
predicted_y <- matrix(0, 10000, 15)

#For loop that follows intercept + slope * data + noise
for(i in 1:10000){
  for(j in 1:15){
     predicted_y[i,] <- b0[i] + b1[i] * xx + rnorm(1,0,sqrt(s2[i,]))
  }
}

#Plot line of best fit
plot(x,y, main = "Data, Line of Best Fit, and Intervals for Y1, variance
inceasing with x")
abline(mean(b0), mean(b1), col = 'green')

#Plot Posterior Interval
lines(xx,apply(fittedy, 2, quantile, 0.975), col = 'blue')
lines(xx,apply(fittedy, 2, quantile, 0.025), col = 'blue')

#Plot Posterior Predicted Interval
lines(xx,apply(predicted_y, 2, quantile, 0.975), col = 'red')
lines(xx,apply(predicted_y, 2, quantile, 0.025), col = 'red')
```



4 SD increase with x

```
linmdl4 <- "
  model {

    for(i in 1:15){
      y[i] ~ dnorm(mu[i], 1/s2[i])
      mu[i] <- b0 + b1*x[i]
      s2[i] = sqrt(a1 * x[i])
    }

    #Priors
    b0 ~ dnorm(0, .001)
    b1 ~ dnorm(0, .002)
    a1 ~ dgamma(1, .5)


  }
"

x <- hw2$x
y <- hw2$y

writeLines(linmdl4, 'linreg.txt')
```

```r
data.jags <- c('x', 'y')
parms <- c('b0', 'b1', 'a1', 's2')

linreg.sim4 <- jags(data.jags,
                    inits = NULL,
                    parameters.to.save = parms,
                    model.file = 'linreg.txt',
                    n.iter = 35000,
                    n.burnin = 10000,
                    n.chains = 4,
                    n.thin = 10)

library(R2jags)
# Intermediary steps

sims <- as.mcmc(linreg.sim3)
chains <- as.matrix(sims)
colnames(chains)
```

```
##  [1] "a1"       "b0"       "b1"       "deviance" "s2[1]"    "s2[10]"
##  [7] "s2[11]"   "s2[12]"   "s2[13]"   "s2[14]"   "s2[15]"   "s2[2]"
## [13] "s2[3]"    "s2[4]"    "s2[5]"    "s2[6]"    "s2[7]"    "s2[8]"
## [19] "s2[9]"
```

```r
b0 = chains[,2]
b1 = chains[,3]

s2 = chains[,c(5,12:19,6:11)]


#Make some data
xx <- seq(0,100,length.out = 15)

#Find Fitted values
#Fit each x value

#Make a matrix
fittedy <- matrix(0, 10000, 15)

#For loop that follows intercept + slope * data
for(i in 1:10000){
  for(j in 1:15){
      fittedy[i,] <- b0[i] + b1[i] * xx
  }
}

#Find Predicted Values
#Make matrix to hold predicted values
predicted_y <- matrix(0, 10000, 15)
```

```r
#For loop that follows intercept + slope * data + noise
for(i in 1:10000){
  for(j in 1:15){
     predicted_y[i,] <- b0[i] + b1[i] * xx + rnorm(1,0,(s2[i,]))
  }
}

#Plot line of best fit
plot(x,y, main = "Data, Line of Best Fit, and Intervals for Y1, Standard
Deviation inceasing with x")
abline(mean(b0), mean(b1), col = 'green')

#Plot Posterior Interval
lines(xx,apply(fittedy, 2, quantile, 0.975), col = 'blue')
lines(xx,apply(fittedy, 2, quantile, 0.025), col = 'blue')

#Plot Posterior Predicted Interval
lines(xx,apply(predicted_y, 2, quantile, 0.975), col = 'red')
lines(xx,apply(predicted_y, 2, quantile, 0.025), col = 'red')
```