Analysis of Factors Associated with Pell Grant Recipient Graduation Rates

Martha Curi, Daniel Parel, Peter Rosenquist

May 2020

1 Introduction

Access to higher education has been the primary driving force for social mobility in modern society. Over the past decades, drastic increases in college tuition, decreased federal and state funding, and enlarged income gaps all have limited access to college for students from lower socioeconomic backgrounds.

According to the latest research by the National Center for Education Statistics, students who were economically and socially disadvantaged were less likely to complete their college education. The research states that "after graduating high school, only 14 percent of low-SES students received a bachelor's or higher degree within eight years compared to 29 percent of middle-income students" [1]. Understanding the factors that influence graduation rates among low income students is the motivation behind this paper.

The study focuses on low-income students enrolled in top US colleges and universities. In particular, we examine the 6-year graduation rates of Pell award recipients in the post-secondary institutions with the highest Carnegie Classification in research activities (high and very high)[2]. Our objective is to explore various institutional characteristics that are significantly associated with 6-year graduation rates among Pell recipients, and to explore which type of model is the most accurate in its predictions of this graduation rate. The data for the study comes from the Integrated Post-secondary Education Data System (IPEDS)[3]. This is a public database owned by the National Center for Education Statistics. We will use the most recent IPEDS data available for degree granting institutions with high or very high research activities.

The remainder of this paper is organized as follows: Section 2 will discuss the Bayesian multi-variate regression models that were built to meet the objective of the study, Section 3 describes the results of our models, and Section 4 interprets our results, provides our conclusions, and suggests further related avenues of study.

2 Methods

2.1 Bayesian linear regression

An initial analysis assume our outcome variable for 6-year graduation rates of Pell recipients is approximately normally distributed. Therefore, a natural choice for modeling the outcome variable was a Bayesian linear regression using a Gibbs sampler which was derived using a multivariate normal likelihood and a non-informative prior.

Our posterior will then be:

$$p(\beta, \sigma^2 | y, X) \propto (\sigma^2)^{-\frac{n}{2}-1} exp[-\frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)]$$

[1]CFA Staff — Posted In: Academics, and CFA Staff —"Addressing the College Completion Gap Among Low-Income Students."Addressing the College Completion Gap Among Low-Income Students, 3 Oct. 2018, collegeforamerica.org/collegecompletion-low-income-students/.

[2]Pell Award status is a widely-used measure to identify low-income students in colleges and universities.

[3]https://nces.ed.gov/ipeds/use-the-data

Taking the conditional distributions[4] of the above posterior yields the following Gibbs Sampler:

1. Set the starting values for σ^2 and β using MSE and coefficient results from a frequentist linear regression respectively.
2. Sample $(\sigma^2)^b$ from a $InverseGamma[\frac{n}{2}, \frac{1}{2}(y - X\beta^{b-1})'(y - X\beta^{b-1})]$ distribution.
3. Sample β^b from a $MVN[(X'X)^{-1}X'y, (\sigma^2)^{b-1}(X'X)^{-1}]$ distribution.
4. Repeat steps 2 and 3 until the desired number of samples are generated.

We implemented three Gibbs samplers using starting values from the results (mean squared error (MSE) and coefficients) of three different linear regressions which used slightly different explanatory variables across each model. Additionally, for later use in cross-validation, each model was trained on a random split of the data the data was split into a test and a training set by removing a random 25% of the original data set and using the remaining 75% as the test dataset.

We fit the first regression with parameters that we expected to be associated with our outcome variable. These included average Pell grant amount awarded to recipients, percent of students that received the Pell award, admission rate to the university, out of state price of living, instruction expenses as a percent of budget, and ACT composite scores. We expect these parameters to affect graduation rate because they are proxies for the financial burden that a student may face, the aid that they may or may not receive, and the rigor, prestige, and selectivity of the institution that they are attending.

The second regression only included the parameters that we found to be significant in the first regression, which included percent of students that received the Pell award, admission rate to the university, out of state price of living, and ACT composite scores.

Finally, the third regression included all of the original parameters we selected in the first regression, with the addition of 2 quadratic terms: one on the percent of students that received the Pell grant and the other on the admissions rate to the university.

We assessed model convergence on all three linear models by producing running mean plots and calculating Geweke diagnostics after burn-in. Diagnostics for the variance were also run across all three models.

After running the regressions on the test data set and calculating their respective MSEs, we determined that the polynomial model was the best model and its parameterization was chosen for comparison to the Laplace model that was built next.

2.2 Normal versus Laplace distributed errors

The Laplace model was derived using a mixture representation of normals and a inverse gamma priors as is described in Ding and Blitzstein (2016)[5]. A Gibbs-MH Sampler was implemented to sample from σ^2,β, and each α_i. The full posterior distribution is as follows:

The full posterior is then of the form:

$$p(\beta, \sigma^2, A|y, X) \propto (\sigma^2)^a exp[-\frac{b}{\sigma^2}] \times \prod_{i=1}^{n} (\frac{\alpha_i}{\sigma^2})^{1/2} exp[-\frac{\alpha_i}{2 \times 4\sigma^2}(y_i - X\beta)^2] \times \alpha_1^{-2} exp[-\frac{1}{2\alpha}]$$

The full conditionals[6] for σ^2 and (individually) $\alpha_1, ..., \alpha_n$ are recognizable as an inverse Gamma with shape $n/2 + a$ and rate $b + \frac{1}{8}\sum_{i=1}^{n} \alpha_i (y_i - X\beta)^2$ and an inverse Gaussian with mean $\frac{2\sigma}{|y_1 - X_1\beta|}$ and shape 1. The a and b terms in the conditional distribution of σ^2 are set to 0.0001 so as to attempt to be uninformative. Sampling from β will be performed in a Metropolis-Hastings step with proposal density MVN($\hat{\beta}, \tau\Sigma$) where $\hat{\beta}$ is the vector of estimated coefficients from ordinary least squares linear regression, Σ is the variance-covariance matrix derived from that same model, and τ is a tuning parameter used to achieve optimal acceptance rates for the M-H step. All together the Gibbs-MH sampler will proceed as follows:

[4] See appendix B for derivations of full conditionals

[5] Ding P, Blitzstein JK. On the Gaussian Mixture Representation of the Laplace Distribution. The American Statistician. 2018;72(2):172-174. doi:10.1080/00031305.2017.1291448

[6] see Appendix B for derivations of full conditionals

1. Set the starting values for σ^2 and β using MSE and coefficient results from a frequentist linear regression respectively. Set the starting values of each α_i as the mean of its inverse Gaussian distribution.

2. Sample β^b using a MH step with proposal MVN($\hat{\beta}, \tau\Sigma$) with $(\sigma^2)^{(b-1)}$ and each $\alpha_i^{(b-1)}$ applied in the conditional distribution of $/beta$.

3. Sample $(\sigma^2)^b$ from a $InverseGamma[\frac{n}{2} + a, b + \frac{1}{8}\sum_{i=1}^{n} \alpha_i(y_i - X\beta)^2]$ distribution.

4. Sample each α_i^b from a $InverseGaussian(\frac{2\sigma^b}{|y_1 - X_1\beta^b|}, 1)$ distribution.[7]

5. Repeat steps 2-4 until the desired number of samples are generated.

We chose a Laplace likelihood as an alternative because it has fatter tails and should be a better fit for less concentrated outcome observations. Furthermore, it serves as a check on the normality assumption for the residuals.

In order to test model convergence, we ran running mean plots on β and σ^2 as well as running Geweke diagnostics after burn-in as well as calculating the acceptance rate.

Finally, we looked at the credible intervals between the Laplace and the polynomial model to select our final model.

3 Results

Between the three normal models that were implemented, the model containing polynomial terms was determined to be the best due to the fact that it had the lowest MSE as shown in Figure 1. Low MSE is a valid measure to compare models because, intuitively, we would expect the best parameterized model to perform the best on out-of-sample data.

	Full Normal	**Reduced Variable Normal**	**Polynomial Normal**
MSE	72.88	73.31	64.38

Figure 1: Comparison of the MSE values of the Normal regression models

Figure 2 shows a comparison of the 95 percent credible intervals between the Normal Polynomial Model and the Laplace model that we selected. Of note, the M-H step employed to sample the conditional distribution of beta for the Laplace likelihood model achieved a satisfactory acceptance rate of approximately 18%. Based on the "width" column, it is clear that the Laplace model was consistently more precise at estimating the coefficients of almost all of our selected parameters.

	Normal Polynomial Model				**Laplace Model**			
	50%	**2.5%**	**97.5%**	**width**	**50%**	**2.5%**	**97.5%**	**width**
(Intercept)	33.16723	9.46436	56.37757	46.91321	32.98797	26.25897	39.88706	13.62809
avg_pell_amnt	-0.00241	-0.0059	0.00111	0.00701	-0.00224	-0.00325	-0.0013	0.00195
percent_given_pell	-0.9188	-1.2274	-0.60829	0.61911	-0.9172	-1.00316	-0.82527	0.17789
I(percent_given_pell^2)	0.00806	0.00451	0.01157	0.00706	0.00805	0.007	0.00903	0.00203
admit_rate	-0.32072	-0.48171	-0.1565	0.32521	-0.32281	-0.37	-0.27652	0.09348
I(admit_rate^2)	0.00243	0.00099	0.00386	0.00287	0.00242	0.00201	0.00282	0.00081
OOS_live_on_campus_price	0.00034	0.00023	0.00045	0.00022	0.00035	0.00031	0.00038	7e-05
instruction_expenses	-0.00507	-0.10646	0.09721	0.20367	-0.00954	-0.03849	0.01805	0.05654
ACT_composite	1.72944	1.15249	2.31783	1.16534	1.71934	1.54469	1.88775	0.34306

Figure 2: Comparison of model coefficients between the Normal and Laplace models

For both the Normal Polynomial model and the Laplace model Geweke Diagnostics were calculated for all generated variables. All Geweke values were between -2 and 2. Running mean and ACF plots for

[7]Alphas will be sampled with the current step's values of the other parameters due to overparameterization.

each variable in each model showed the desired stabilization of the mean and low autocorrelation after few generated samples, with the exception of the MH step for β in the Laplace likelihood model which required thinning (kept every 10th sample) to reach an minimal level of autocorrelation.

4 Discussion

Overall, we found that the Gibbs-MH sampler which modeled our initial assumptions of a Laplace likelihood and non-informative prior was the best candidate to model our outcome variable. The Laplace model was able to pick up an additional significant variable which was the average Pell amount awarded to students on top of its superior precision in general. Pell amount awarded has a negative impact on 6 year Pell graduations rates which suggests that the greater the need of an incoming student, the less likely they are to graduate.

In both models, the instructional expenses were found to be not significant. At first this was surprising, but the pool of post-secondary institutions that we selected happened to be some of the most well-funded in the United States. We deduced that even if a school had a low percentage of their budget dedicated to instructional expenses, the quality of education and available resources would not be affected.

Another surprising result is that we found that the out of state on campus living expenses had a positive correlation with the graduation rates of Pell recipients. More research is needed in order to determine the cause of this correlation. A potential factor could be that universities with higher cost of living expenses are located in safer or more enriching locations thus providing a more stable environment for students to focus on academics.

The results of this paper lend themselves to multiple potential expansions of our research. One such extension would be to create separate models for the graduate rate of Pell grant recipients and non-Pell Grant recipients. A dataset with separate parameters for Pell and non-Pell students could be used to compare the important factors in an education for lower income students versus their peers. This information would be particularly useful for universities looking to target their policies towards the retention of Pell Grant recipients who may have differing needs from the general student body.

Additionally, expanding the scope of the data in this research to include universities with lower Carnegie classifications are another potential extension of this research. Lower income students at the most prestigious universities may have differing needs than lower income students at other schools and this distinction is not one that can be caught using the methods detailed in this paper.

Nevertheless, the analysis performed on what factors are important in determining graduation rates among Pell recipients provided insight into what we can do at an institutional level to understand the needs of incoming students.

5 Appendix A

```
library(ggplot2)
library(car)
library(mvtnorm)
library(MCMCpack)
library(mcmcplots)

##################################################################
### Set up, split for train, test set ###
##################################################################
# read in data
ipeds = read.csv('IPEDS_cleaned.csv', header = T)

# select variables for model
cols   = c("Pell.Grant.recipients...Overall.graduation.rate.within.150.percent.of.normal.tim
          "Average.amount.of.Pell.grant.aid.awarded.to.full.time.first.time.undergraduates'
          "Percent.of.full.time.first.time.undergraduates.awarded.Pell.grants",
          "Percent.admitted...total",
          "Total.price.for.out.of.state.students.living.on.campus",
          "Instruction.expenses.as.a.percent.of.total.core.expenses..FASB.",
          "ACT.Composite.75th.percentile.score" )
ipeds = ipeds[ipeds$Sector.of.institution == 2, cols]

# rename columns
colnames(ipeds) = c('six_yr_grad',
                    'avg_pell_amnt',
                    'percent_given_pell',
                    'admit_rate',
                    'OOS_live_on_campus_price',
                    'instruction_expenses',
                    'ACT_composite')
ipeds           = ipeds[complete.cases(ipeds),] # select complete cases

# split for test and train
set.seed(999)
train_index = sample.int(dim(ipeds)[1], size = 240, replace = FALSE)
ipeds_test  = ipeds[-train_index,]
ipeds       = ipeds[train_index,]

# visualize relationships
ggpairs(ipeds)

##################################################################
### Model 1, assumed relevant terms ###
##################################################################
# OLS model with assumed best predictors
lm = lm(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
        admit_rate +
        OOS_live_on_campus_price +
        instruction_expenses  + ACT_composite, data = ipeds)

summary(lm)
```

```
# OLS results for use in Gibbs Sampler
#sigma = vcov(lm)
bhat  = lm$coefficients

# set up
set.seed(42)
B        <- 20000
k        <- dim(ipeds)[2]
n        <- dim(ipeds)[1]
y        <- ipeds[,1]
X <- model.matrix(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
                        admit_rate +
                        OOS_live_on_campus_price +
                        instruction_expenses  + ACT_composite, data = ipeds)

sig              <- rep(0,B)
betamat <- matrix(0, nrow = B, ncol = k)
v                          <- solve(t(X)%*%X)

sig[1]                     <- (1/(n-k))*sum((y-X%*%bhat)^2) # MSE
betamat[1,]      <- bhat # OLS

for (b in 2:B) {
  # Sample Beta #
  betamat[b,] <- c(rmvnorm(1, bhat, sig[b-1]*v))

  # Sample Sigma^2 #
  sig[b] <- rinvgamma(1, n/2, sum((y-X%*%betamat[b-1,])^2)/2)
}
# assess convergence
rmeanplot(betamat)

# subset accordingly, here I've chosen last 10000 samples for inclusion
betai      <- betamat[(B/2+1):B,]
sigi             <- sig[(B/2+1):B]

# looking for values below 2
geweke.diag(betai, frac2 = .5)
geweke.diag(sigi, frac2 = .5)

# reset graphing par
par(mfrow = c(1,1))
#caterplot(betai, denstrip = TRUE)

# single value for sigma-squared...inspect it with mcmcplot1
sigMat             <- matrix(sigi, ncol = 1)
colnames(sigMat)         <- 'sigma^2'
mcmcplot1(sigMat, greek = TRUE)

# print out regression results
coefs             = apply(betai, 2, quantile, probs = c(0.5, 0.025, 0.975))
colnames(coefs) = colnames(X)
results_normal_errors = round(t(coefs), 5)
```

```
# View final results of normal model, flat prior
results_normal_errors

####################################################################
### Model 2, only significant terms ###
####################################################################
# OLS model including only statistically significant terms from previous model
lm = lm(six_yr_grad ~ percent_given_pell +
          admit_rate +
          OOS_live_on_campus_price +
          ACT_composite, data = ipeds)

summary(lm)

# OLS results for use in Gibbs Sampler
#sigma = vcov(lm)
bhat  = lm$coefficients

# set up
set.seed(90)
B       <- 20000
k       <- dim(ipeds)[2]-2
n       <- dim(ipeds)[1]
y       <- ipeds[,1]
X <- model.matrix(six_yr_grad ~ percent_given_pell +
                    admit_rate +
                    OOS_live_on_campus_price +
                    ACT_composite, data = ipeds)

sig              <- rep(0,B)
betamat <- matrix(0, nrow = B, ncol = k)
v                         <- solve(t(X)%*%X)

sig[1]                    <- (1/(n-k))*sum((y-X%*%bhat)^2) # MSE
betamat[1,]        <- bhat # OLS

for (b in 2:B) {
  # Sample Beta #
  betamat[b,] <- c(rmvnorm(1, bhat, sig[b-1]*v))

  # Sample Sigma^2 #
  sig[b] <- rinvgamma(1, n/2, sum((y-X%*%betamat[b-1,])^2)/2)
}
# assess convergence
rmeanplot(betamat)

# subset accordingly, here I've chosen last 10000 samples for inclusion
betai      <- betamat[(B/2+1):B,]
sigi            <- sig[(B/2+1):B]

# looking for values below 2
geweke.diag(betai, frac2 = .5)
geweke.diag(sigi, frac2 = .5)
```

```r
par(mfrow = c(1,1))
#caterplot(betai, denstrip = TRUE)

# single value for sigma-squared...inspect it with mcmcplot1
sigMat              <- matrix(sigi, ncol = 1)
colnames(sigMat)        <- 'sigma^2'
mcmcplot1(sigMat, greek = TRUE)

# print out regression results
coefs              = apply(betai, 2, quantile, probs = c(0.5, 0.025, 0.975))
colnames(coefs) = colnames(X)
results_normal_errors_small_model = round(t(coefs), 5)

# View final results, normal model flat prior, only significant predictors
results_normal_errors_small_model

###################################################################
### Model 3, polynomial terms ###
###################################################################
# OLS model including quadratic terms for percent_given_pell and admit_rate
lm = lm(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
          I(percent_given_pell^2) + admit_rate +
          I(admit_rate^2) + OOS_live_on_campus_price +
          instruction_expenses  + ACT_composite, data = ipeds)

summary(lm)

# OLS results for use in Gibbs Sampler
#sigma = vcov(lm)
bhat  = lm$coefficients

# set up
set.seed(88)
B        <- 20000
#k       <- dim(ipeds)[2]-1
k        <- dim(ipeds)[2] +2
n        <- dim(ipeds)[1]
y        <- ipeds[,1]
X <- model.matrix(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
                    I(percent_given_pell^2) + admit_rate +
                    I(admit_rate^2) + OOS_live_on_campus_price +
                    instruction_expenses  + ACT_composite, data = ipeds)

sig              <- rep(0,B)
betamat <- matrix(0, nrow = B, ncol = k)
v                          <- solve(t(X)%*%X)

sig[1]                     <- (1/(n-k))*sum((y-X%*%bhat)^2) # MSE
betamat[1,]        <- bhat # OLS

for (b in 2:B) {
  # Sample Beta #
  betamat[b,] <- c(rmvnorm(1, bhat, sig[b-1]*v))
```

```r
  # Sample Sigma^2 #
  sig[b] <- rinvgamma(1, n/2, sum((y-X%*%betamat[b-1,])^2)/2)
}
# assess convergence
rmeanplot(betamat)

# subset accordingly, here I've chosen last 10000 samples for inclusion
betai     <- betamat[(B/2+1):B,]
sigi            <- sig[(B/2+1):B]

# looking for values below 2
geweke.diag(betai, frac2 = .5)
geweke.diag(sigi, frac2 = .5)

par(mfrow = c(1,1))
#caterplot(betai, denstrip = TRUE)

# single value for sigma-squared... inspect it with mcmcplot1
sigMat              <- matrix(sigi, ncol = 1)
colnames(sigMat)         <- 'sigma^2'
mcmcplot1(sigMat, greek = TRUE)

# print out regression results
coefs              = apply(betai, 2, quantile, probs = c(0.5, 0.025, 0.975))
colnames(coefs) = colnames(X)
results_normal_errors_poly_model = round(t(coefs), 5)

# View final results of normal model, flat prior, quadratic terms
results_normal_errors_poly_model

#####################################################################
# Compare models via cross validation (MSE)
#####################################################################

# function for MSE calculation
calc_mse = function(betas, testX, testY){
  # takes vector of coefficients and calculates y_hat from textX
  # calculates MSE from 1/n*(testY - y_hat)^2

  y_hat = testX %*% betas
  MSE   = 1 / nrow(testX) * sum((testY-y_hat)^2)
  return(MSE)
}

# calculate MSE for results_normal_errors
X <- model.matrix(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
                     admit_rate +
                     OOS_live_on_campus_price +
                     instruction_expenses  + ACT_composite, data = ipeds_test)

MSE_normal_errors = calc_mse(results_normal_errors[,1],
                             testX = X,
                             testY = ipeds_test[,'six_yr_grad'])
```

```
# calculate MSE for results_normal_errors_small_model
X <- model.matrix(six_yr_grad ~ percent_given_pell +
                    admit_rate +
                    OOS_live_on_campus_price +
                    ACT_composite, data = ipeds_test)

MSE_normal_errors_small_model = calc_mse(results_normal_errors_small_model[,1],
                                          testX = X,
                                          testY = ipeds_test[,'six_yr_grad'])

# calculate MSE for results_normal_errors_poly_model
X <- model.matrix(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
                    I(percent_given_pell^2) + admit_rate +
                    I(admit_rate^2) + OOS_live_on_campus_price +
                    instruction_expenses  + ACT_composite, data = ipeds_test)

MSE_normal_errors_poly_model = calc_mse(results_normal_errors_poly_model[,1],
                                         testX = X,
                                         testY = ipeds_test[,'six_yr_grad'])

MSE = data.frame(MSE_normal_errors, MSE_normal_errors_small_model, MSE_normal_errors_poly_
MSE
# based on these results, select poly_model as best model, use its parameterization for co

#####################################################################
# Rerun normal_errors_poly_model on full dataset
# Run Laplace_errors_model on full dataset
# Compare credible intervals
#####################################################################

# read in data
ipeds = read.csv('IPEDS_cleaned.csv', header = T)

# select variables for model
cols  = c("Pell.Grant.recipients...Overall.graduation.rate.within.150.percent.of.normal.tim
          "Average.amount.of.Pell.grant.aid.awarded.to.full.time.first.time.undergraduates"
          "Percent.of.full.time.first.time.undergraduates.awarded.Pell.grants",
          "Percent.admitted...total",
          "Total.price.for.out.of.state.students.living.on.campus",
          "Instruction.expenses.as.a.percent.of.total.core.expenses..FASB.",
          "ACT.Composite.75th.percentile.score" )
ipeds = ipeds[ipeds$Sector.of.institution == 2, cols]

# rename columns
colnames(ipeds) = c('six_yr_grad',
                    'avg_pell_amnt',
                    'percent_given_pell',
                    'admit_rate',
                    'OOS_live_on_campus_price',
                    'instruction_expenses',
                    'ACT_composite')
ipeds           = ipeds[complete.cases(ipeds),] # select complete cases
```

```
#########################################################################
# Model 4 Normal Errors, polynomial terms, full data
#########################################################################

# OLS model including quadratic terms for percent_given_pell and admit_rate
lm = lm(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
          I(percent_given_pell^2) + admit_rate +
          I(admit_rate^2) + OOS_live_on_campus_price +
          instruction_expenses  + ACT_composite, data = ipeds)

summary(lm)

# OLS results for use in Gibbs Sampler
#sigma = vcov(lm)
bhat  = lm$coefficients

# set up
set.seed(88)
B        <- 20000
#k       <- dim(ipeds)[2]-1
k        <- dim(ipeds)[2] +2
n        <- dim(ipeds)[1]
y        <- ipeds[,1]
X <- model.matrix(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
                    I(percent_given_pell^2) + admit_rate +
                    I(admit_rate^2) + OOS_live_on_campus_price +
                    instruction_expenses  + ACT_composite, data = ipeds)

sig              <- rep(0,B)
betamat <- matrix(0, nrow = B, ncol = k)
v                        <- solve(t(X)%*%X)

sig[1]                   <- (1/(n-k))*sum((y-X%*%bhat)^2) # MSE
betamat[1,]      <- bhat # OLS

for (b in 2:B) {
  # Sample Beta #
  betamat[b,] <- c(rmvnorm(1, bhat, sig[b-1]*v))

  # Sample Sigma^2 #
  sig[b] <- rinvgamma(1, n/2, sum((y-X%*%betamat[b-1,])^2)/2)
}
# assess convergence
rmeanplot(betamat)

# subset accordingly, here I've chosen last 10000 samples for inclusion
betai      <- betamat[(B/2+1):B,]
sigi             <- sig[(B/2+1):B]

# looking for values below 2
geweke.diag(betai, frac2 = .5)
geweke.diag(sigi, frac2 = .5)

par(mfrow = c(1,1))
```

```
#caterplot(betai, denstrip = TRUE)

# single value for sigma-squared...inspect it with mcmcplot1
sigMat              <- matrix(sigi, ncol = 1)
colnames(sigMat)         <- 'sigma^2'
mcmcplot1(sigMat, greek = TRUE)

# print out regression results
coefs              = apply(betai, 2, quantile, probs = c(0.5, 0.025, 0.975))
colnames(coefs) = colnames(X)
results_normal_errors_poly_model_full_data = round(t(coefs), 5)
results_normal_errors_poly_model_full_data = cbind(results_normal_errors_poly_model_full_da
                                                   width = results_normal_errors_poly_mode
                                                    results_normal_errors_poly_model_full_

# View final results of normal model, flat prior, quadratic terms
results_normal_errors_poly_model_full_data

####################################################################
# Model 5 Laplace, full data
####################################################################

# OLS model including quadratic terms for percent_given_pell and admit_rate
lm = lm(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
           I(percent_given_pell^2) + admit_rate +
           I(admit_rate^2) + OOS_live_on_campus_price +
           instruction_expenses  + ACT_composite, data = ipeds)
#lm = lm(six_yr_grad ~ ACT_composite, data = ipeds)

summary(lm)

# OLS results for use in MH algorithm
bhat  = lm$coefficients
vbeta = vcov(lm)

# set up
set.seed(42)
B          <- 80000
k          <- dim(ipeds)[2] + 3 -1
n          <- dim(ipeds)[1]
y0       <- ipeds[,1]
X0  <- model.matrix(six_yr_grad ~ avg_pell_amnt + percent_given_pell +
                       I(percent_given_pell^2) + admit_rate +
                       I(admit_rate^2) + OOS_live_on_campus_price +
                       instruction_expenses  + ACT_composite, data = ipeds)

# initialize empty parameter objects
sig              <- rep(0,B)
betamat <- matrix(0, nrow = B, ncol = k)
alpha   <- matrix(0, nrow = B, ncol = n)

# set first values
sig[1]                  <- (1/(n-k)) * sum((y0-X0%*%bhat)^2) # MSE
betamat[1,]       <- bhat # OLS
```

```r
for(i in 1:n){
  alpha[1,i] = 2*sig[1]^.5 / abs(y0[i] - X0[i, ] %*% betamat[1,]) # mean of respective in
}

# initialize acceptance vec
ar = vector('numeric', length = B)

tau     <- .10 # need to tune this

log_tdens = function(Beta, alph, sigsq, y = y0, X = X0){
  return(-1/(2*4*sigsq) * sum(alph*(y - X%*%Beta)^2))
}

set.seed(90211)
for(t in 2:B){
  # draw candidate vector of betas
  bstar <- rmvnorm(1, betamat[1,], sigma = tau*vbeta)

  # evaluate candidate compared to previous value
  # print('candidate ')
  # print(exp(log_tdens(Beta = t(bstar),
  #                     sigsq = sig[t-1],
  #                     alph  = alpha[t-1,])))
  # print('previous')
  # print(exp(log_tdens(Beta = betamat[t-1,],
  #                     sigsq = sig[t-1],
  #                     alph  = alpha[t-1,])))

  r              <- exp(log_tdens(Beta = t(bstar),
                     sigsq = sig[t-1],
                     alph  = alpha[t-1,])) / exp(log_tdens(Beta = betamat[t-1,],
                                                          sigsq = sig[t-1],
                                                          alph  = alpha[t-1,]))
  U              <- runif(1)

  # maybe select, maybe exclude based on density ratios
  if(U < min(1,r)){
    betamat[t,] <- bstar
    ar[t]            <- 1
  } else{
    betamat[t,] <- betamat[t-1,]
    ar[t]            <- 0
  }

  # Sample Sigma^2
  sig[t] <- rinvgamma(1,
                      n/2+.0001,
                      .0001 + 1/8*sum(alpha[t-1,] * (y0-X0%*%betamat[t-1,])^2))

  # Sample alphas
  for(i in 1:n){
    alpha[t,i] = rinvgauss(1,
                           mean = 2*sig[t]^.5/abs(y0[i] - X0[i, ] %*% betamat[t,]),
                           shape = 1)
```

```
  }
  #pause(.1)
}

# assess convergence, looking for values below 2
rmeanplot(betamat)
rmeanplot(sig)

# subset accordingly, here I've chosen last 40000 samples for inclusion for betas and alpha
betai     <- betamat[(1*B/2+1):B,]
sigi           <- sig[(1*B/2+1):B]
alphai  <- alpha[(1*B/2 +1):B,]

# looking for values below 2
geweke.diag(betai, frac2 = .5)
geweke.diag(sigi, frac2 = .5)
which(geweke.diag(alphai, frac2 = .5)$z > 2)
# alpha has relatively few (out of 319) stats >2
#--could rerun with larger B to attempt full convergence

# acceptance rate
mean(ar[-c(1:(1*B/2))]) # approx 19 percent

par(mfrow = c(1,1))
#caterplot(betai, denstrip = TRUE)

# single value for sigma-squared...inspect it with mcmcplot1
sigMat              <- matrix(sigi, ncol = 1)
colnames(sigMat)          <- 'sigma^2'
mcmcplot1(sigMat, greek = TRUE)

# thin by selecting every 20th sample check autocorrelation, 2000 final samples
betai = betai[seq(1,40000, 10),]
acf(betai)

# print out regression results
coefs = apply(betai, 2, quantile, probs = c(0.5, 0.025, 0.975))
colnames(coefs) = colnames(X0)
results_laplace_errors_full_data = round(t(coefs), 5)
results_laplace_errors_full_data = cbind(results_laplace_errors_full_data,
                                         width = results_laplace_errors_full_data[,3] -
                                          results_laplace_errors_full_data[,2])

# final results
results_laplace_errors_full_data

# compare
results_normal_errors_poly_model_full_data
```

6 Appendix B

6.1 Normal likelihood model derivations

Full derivation of conditionals for Gibbs sampler assuming a normal likelihood and a non-informative prior

$$Y \sim MVN(X\beta, \sigma^2 I_{nxn}), p(\beta, \sigma^2) \propto (\sigma^2)^{-1}$$

$$L(y|X, \beta, \sigma^2) \propto \prod_{i=1}^{n} (\sigma^2)^{-\frac{1}{2}} exp[-\frac{1}{2\sigma^2}(y_i - X\beta)^2] = (\sigma^2)^{-\frac{n}{2}} exp[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - X_i\beta)^2]$$

Our posterior will then be:

$$p(\beta, \sigma^2|y, X) \propto (\sigma^2)^{-\frac{n}{2}-1} exp[-\frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)]$$

Our first full conditional for σ^2 (immediately evident):

$$\sigma^2|y, X, \beta \sim IG(\frac{n}{2}, \frac{1}{2}(y - X\beta)'(y - X\beta))$$

Next we derive the full conditional of β:

$$p(\beta|\sigma^2, y, X) \propto exp[-\frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)]$$

$$\propto exp[-\frac{1}{2\sigma^2}(y'y - y'(X\beta) - (X\beta)'y + (X\beta)'(X\beta))]$$

$$= exp[-\frac{1}{2\sigma^2}(y'y - 2\beta'X'y + \beta'X'X\beta)]$$

$$\propto exp[-\frac{1}{2\sigma^2}(\beta'X'X\beta - 2\beta'X'y)]$$

$$= exp[-\frac{1}{2\sigma^2}(\beta'X'X\beta - 2\beta'(X'X)(X'X)^{-1}X'y)]$$

$$\propto exp[-\frac{1}{2\sigma^2}((\beta - (X'X)^{-1}X'y)'(X'X)(\beta - (X'X)^{-1})X'y))]$$

$$\beta|\sigma^2, y, X \sim MVN[(X'X)^{-1}X'y, \sigma^2(X'X)^{-1}]$$

Final Gibbs Sampler:

$$1.(\sigma^2)^b \sim IG[\frac{n}{2}, \frac{1}{2}(y - X\beta^{b-1})'(y - X\beta^{b-1})]$$

$$2.\beta^b \sim MVN[(X'X)^{-1}X'y, (\sigma^2)^{b-1}(X'X)^{-1}]$$

6.2 Laplace likelihood model, full derivations

Derivation of full posterior using mixture representation:

$$L(y_1,...,y_n|A,\sigma^2,\beta,y,X) \propto \prod_{i=1}^{n}(\frac{\alpha_i}{\sigma^2})^{1/2}exp[-\frac{\alpha_i}{2\times 4\sigma^2}(y_i - X\beta)^2]$$

$$p(\sigma^2) \propto (\sigma^2)^a exp[-\frac{b}{\sigma^2}]$$

$$p(\alpha_1) \propto \alpha_1^{-2} exp[-\frac{1}{2\alpha}]$$

$$p(\beta,\sigma^2,A|y,X) \propto (\sigma^2)^a exp[-\frac{b}{\sigma^2}] \times \prod_{i=1}^{n}(\frac{\alpha_i}{\sigma^2})^{1/2}exp[-\frac{\alpha_i}{2\times 4\sigma^2}(y_i - X\beta)^2] \times \alpha_1^{-2}exp[-\frac{1}{2\alpha}]$$

Full conditionals can be shown to be the following:

$$p(\beta|\sigma^2,A,y,X) \propto exp[-\frac{1}{2\times 4\sigma^2}\sum_{i=1}^{n}\alpha_i(y_i - X\beta)^2]$$

$$p(\sigma^2|\beta,A,y,X) \propto (\sigma^2)^{-n/2+a+1}exp[-\frac{1}{\sigma^2}(b+\frac{1}{8}\sum_{i=1}^{n}\alpha_i(y_i - X\beta)^2)]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto a_1^{-3/2}exp[-\frac{1}{2}\frac{(\alpha_1 - \frac{2\sigma}{|y_1 - X_1\beta|})^2}{(\frac{2\sigma}{|y_1 - X_1\beta|})^2\alpha_1}]$$

Full derivation of the full conditional for σ^2 follows:

$$p(\sigma^2|\beta,A,y,X) \propto\propto (\sigma^2)^a exp[-\frac{b}{\sigma^2}] \times \prod_{i=1}^{n}(\frac{\alpha_i}{\sigma^2})^{1/2}exp[-\frac{\alpha_i}{2\times 4\sigma^2}(y_i - X\beta)^2]$$

$$p(\sigma^2|\beta,A,y,X) \propto (\sigma^2)^{-n/2+a+1}exp[-\frac{1}{\sigma^2}(b+\frac{1}{8}\sum_{i=1}^{n}\alpha_i(y_i - X\beta)^2)]$$

Full derivation of the full conditional for α_1 follows:

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \prod_{i=1}^{n}(\frac{\alpha_i}{\sigma^2})^{1/2}exp[-\frac{\alpha_i}{2\times 4\sigma^2}(y_i - X\beta)^2] \times \alpha_1^{-2}exp[-\frac{1}{2\alpha}]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{\alpha_1}{4\sigma^2}(y_1 - X_1\beta)^2 + \frac{1}{\alpha_1})]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{\alpha_1^2(y_1 - X_1\beta)^2 + 4\sigma^2}{4\sigma^2\alpha_1})]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{\alpha_1^2 + \frac{4\sigma^2}{(y_1 - X_1\beta)^2}}{\frac{\alpha_1 4\sigma^2}{(y_1 - X_1\beta)^2}})]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{\alpha_1^2 - 2\alpha_1\frac{2\sigma}{|y_1 - X_1\beta|} + \frac{\alpha_1 4\sigma^2}{(y_1 - X_1\beta)^2} + 2\alpha_1\frac{2\sigma}{|y_1 - X_1\beta|}}{\frac{\alpha_1 4\sigma^2}{(y_1 - X_1\beta)^2}})]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{(\alpha_1 - \frac{2\sigma}{|y_1 - X_1\beta|})^2 + 2\alpha_1\frac{2\sigma}{|y_1 - X_1\beta|}}{\alpha_1(\frac{2\sigma}{|y_1 - X_1\beta|})^2})]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{(\alpha_1 - \frac{2\sigma}{|y_1 - X_1\beta|})^2}{\alpha_1(\frac{2\sigma}{|y_1 - X_1\beta|})^2} + \frac{2\alpha_1\frac{2\sigma}{|y_1 - X_1\beta|}}{\alpha_1(\frac{2\sigma}{|y_1 - X_1\beta|})^2})]$$

$$p(\alpha_1|\beta,\sigma^2,y,X) \propto \alpha^{-3/2}exp[-\frac{1}{2}(\frac{(\alpha_1 - \frac{2\sigma}{|y_1 - X_1\beta|})^2}{\alpha_1(\frac{2\sigma}{|y_1 - X_1\beta|})^2})]$$