HW4: Team [my team #06]

Di Deng, Jae Hyun Lee, Ziang Wang, Weijie Yi

10/14/2019

Preliminaries

Load the college application data from Lab1 and create the variable `Elite` by binning the `Top10perc` variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50 %. We will also save the College names as a new variable and remove `Accept` and `Enroll` as temporally they occur after applying, and do not make sense as predictors in future data.

```
data(College)
College = College %>%
  mutate(college = rownames(College)) %>%
  mutate(Elite = factor(Top10perc > 50)) %>%
  mutate(Elite =
           recode(Elite, 'TRUE' = "Yes", 'FALSE'="No")) %>%
  select(c(-Accept, -Enroll))
```

We are going to create a training and test set by randomly splitting the data. First set a random seed by

```
# do not change this; for a break google `8675309`
set.seed(8675309)
n = nrow(College)
n.train = floor(.75*n)
train = sample(1:n, size=n.train, replace=FALSE)
College.train = College[train,]
College.test = College[-train,]
```

1. Build a model, considering EDA, transformations of the response and predictors, possible interactions, etc with the goal of trying to achieve a model where assumptions for linear regression are satisfied, providing justification for your choices. Comment on how well the assumptions are met and and issues that diagnostic plots may reveal. (complete during lab)

```
College_long <- gather(College.train, key = "variable", value = "value", Apps:Grad.Rate)
```

To explore the data, we investigated histograme of data

```
ggplot(College_long[,!is.factor(College.train)], aes(value)) +
  facet_wrap(~variable, scales = 'free') +
  geom_histogram() +
  geom_freqpoly(color = "red")
```

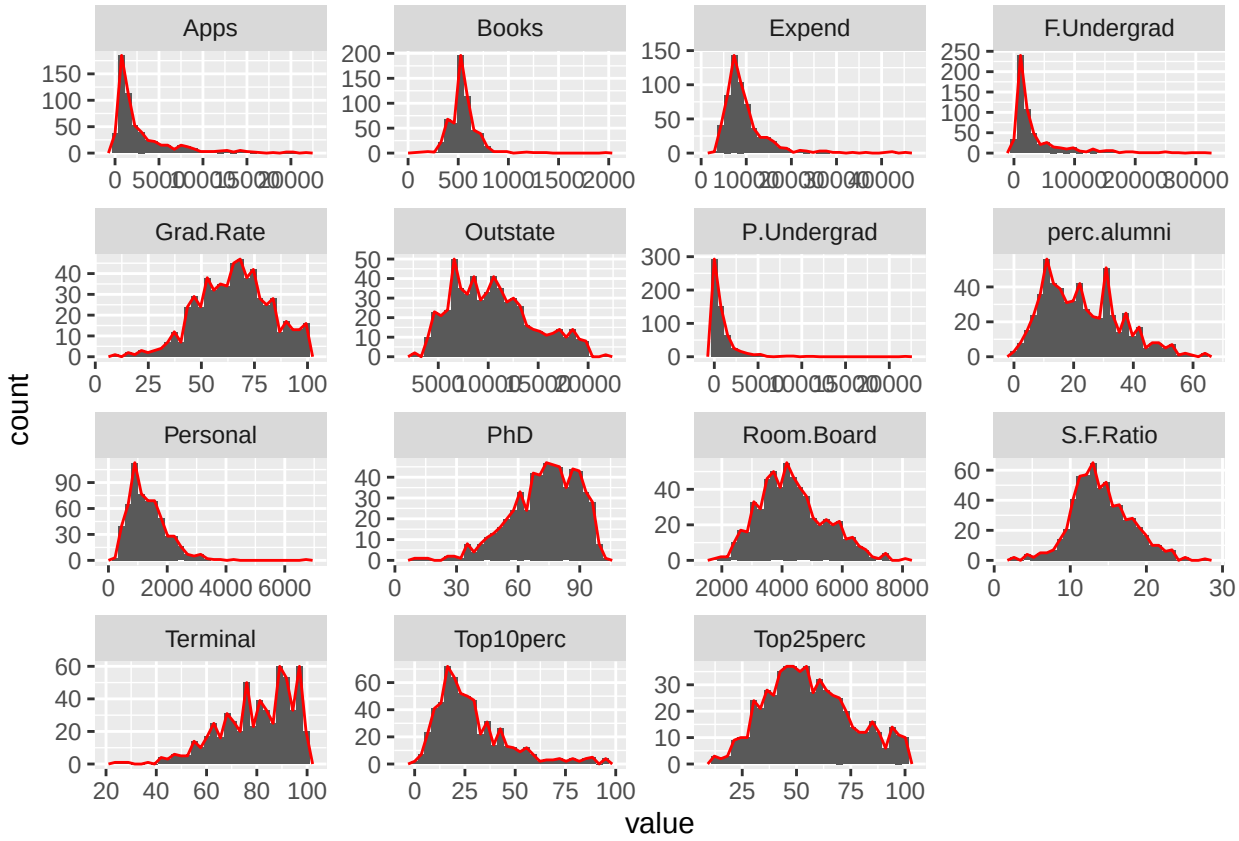


Figure 1: Histogram of variables

We can check that some distributions have skewed shape. Moreover, even plots that show linear relationship between response and predict variables cannot represent relationship because of skewedness. Thus we conclude that we need to transform some of them.

```
College.train1 <- College.train[, c(-1,-17,-18)]
College.train1$perc.alumni <- College.train1$perc.alumni+0.1
kable(summary(car::powerTransform(College.train1))$result, caption = "Summary of Transformations")
```

```
## Registered S3 methods overwritten by 'car':
##   method                           from
##   influence.merMod                 lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod          lme4
##   dfbetas.influence.merMod         lme4
```

Table 1: Summary of Transformations

	Est Power	Rounded Pwr	Wald Lwr Bnd	Wald Upr Bnd
Apps	0.0448420	0.00	-0.0027234	0.0924073
Top10perc	0.4324250	0.43	0.3693058	0.4955441
Top25perc	0.9992387	1.00	0.8454269	1.1530505

	Est Power	Rounded Pwr	Wald Lwr Bnd	Wald Upr Bnd
F.Undergrad	-0.0388875	0.00	-0.0843493	0.0065744
P.Undergrad	0.1275148	0.13	0.0910422	0.1639873
Outstate	0.5843400	0.50	0.4305981	0.7380819
Room.Board	0.2718674	0.50	0.0434929	0.5002419
Books	0.2748179	0.33	0.1443474	0.4052885
Personal	0.1599197	0.16	0.0277140	0.2921253
PhD	2.3660730	2.37	2.1232614	2.6088846
Terminal	2.6578264	2.66	2.3215162	2.9941367
S.F.Ratio	0.7901286	0.79	0.6301489	0.9501082
perc.alumni	0.5863417	0.50	0.4907749	0.6819084
Expend	-0.3200285	-0.33	-0.4189865	-0.2210706
Grad.Rate	1.2801185	1.28	1.0744643	1.4857727

We refer approximated optimal value for lambda based on results of powerTransform.

```
linear.train.norm <- College.train %>%
  mutate(Apps = log(Apps),`F.Undergrad` = log(`F.Undergrad`),
         Expend = log(Expend), `P.Undergrad` = log(`P.Undergrad`),
         Personal = log(Personal)) %>%
  mutate(Outstate = sqrt(Outstate),`Room.Board` = sqrt(`Room.Board`),
         `perc.alumni` = sqrt(perc.alumni)) %>%
  mutate(PhD = PhD^2, Terminal = Terminal^2)
```

```
linear.train.norm %>%
  gather(key = "variable", value = "value", Apps:Grad.Rate) %>%
  ggplot(mapping = aes(value)) +
  facet_wrap(~variable, scales = "free") +
  geom_histogram() +
  geom_freqpoly(color = "red")
```

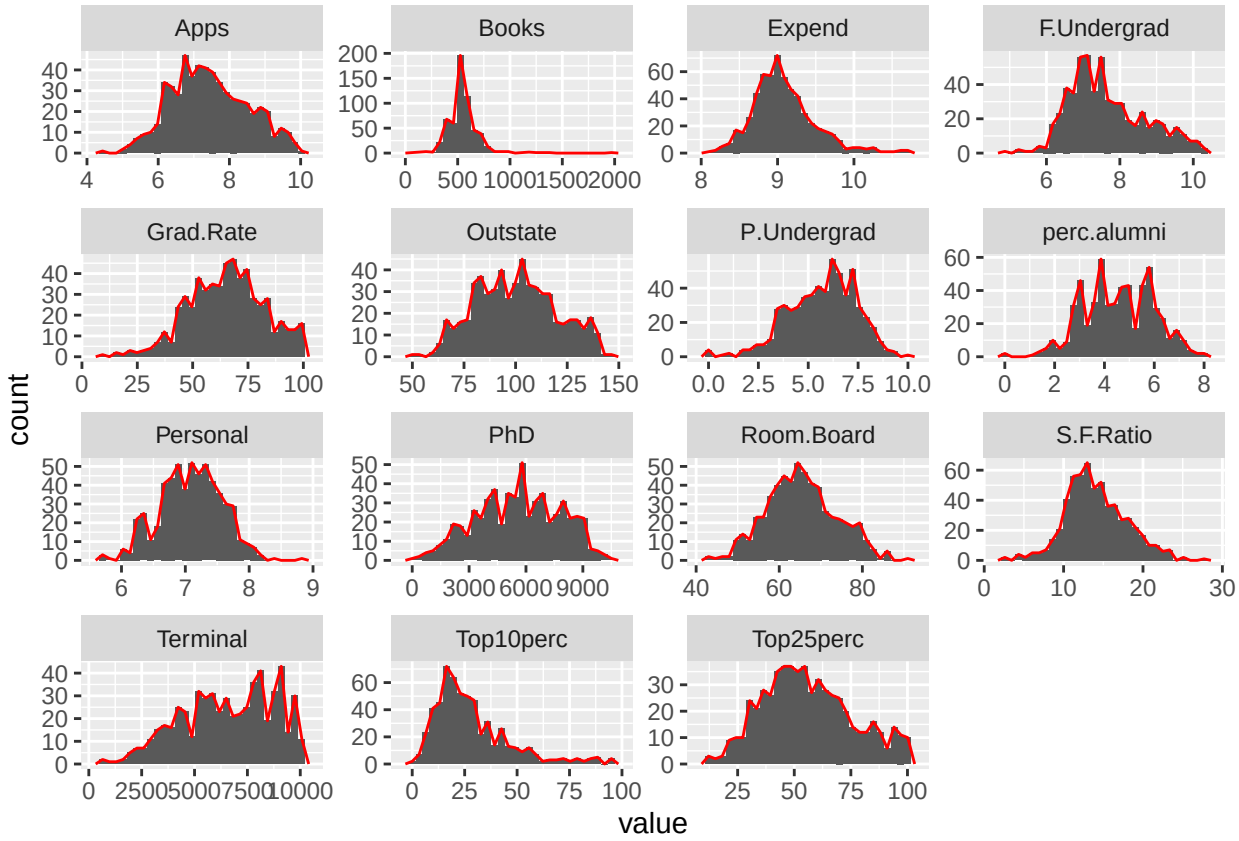


Figure 2: Histogram of transformed data

From this plot, we can check that transformed variables are spread more symmetrically than before

```
linear.train.norm[, -c(1,17,18)] %>%
  gather(-Apps, key = "Variables", value = "Others") %>%
  ggplot(aes(x=Others, y=Apps)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~Variables, scales = "free")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

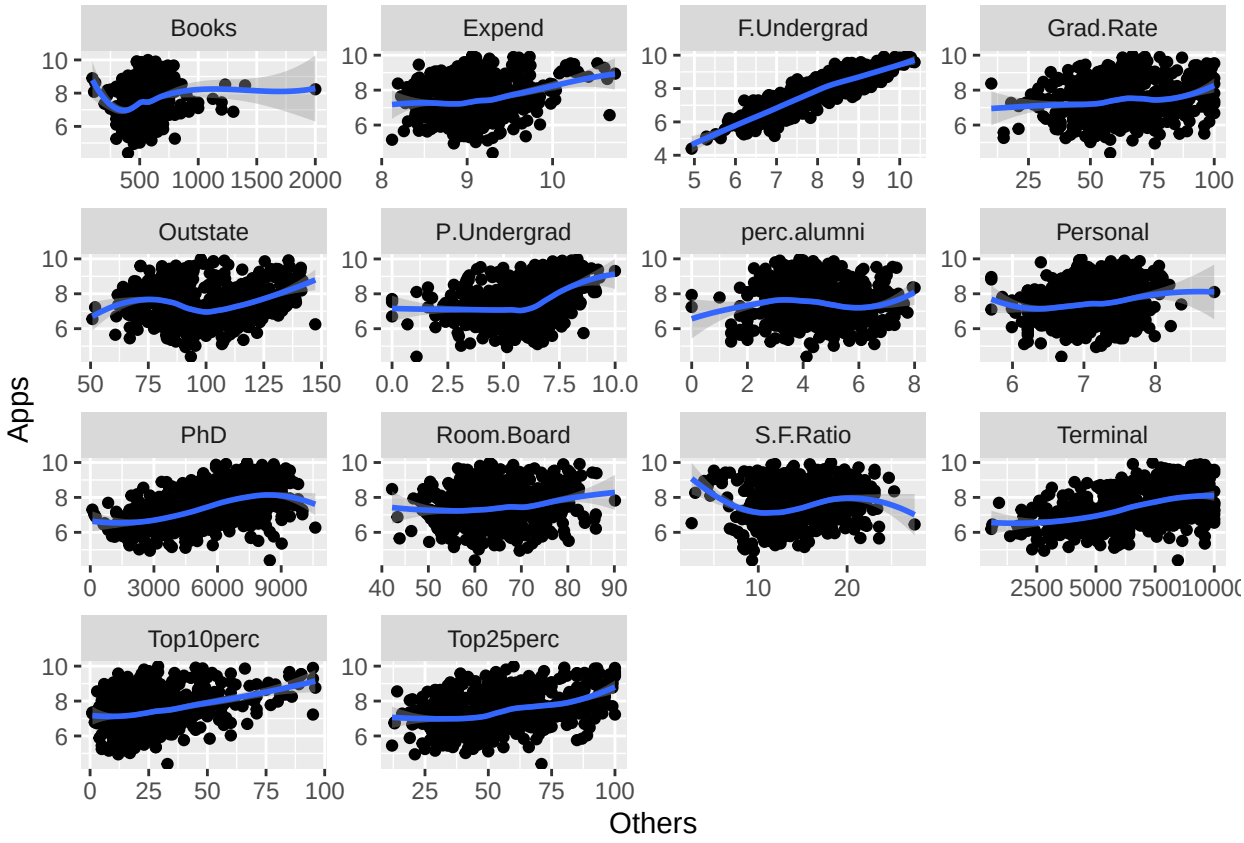


Figure 3: Relationships between Apps and other numeric predictors

```
linear.train.norm[,c(1,2,18)] %>% gather(-Apps, key = "Variables", value = "Others") %>%
  ggplot(aes(x = Others,y = Apps)) +
  geom_point() +
  facet_wrap(~Variables, scales = "free")
```

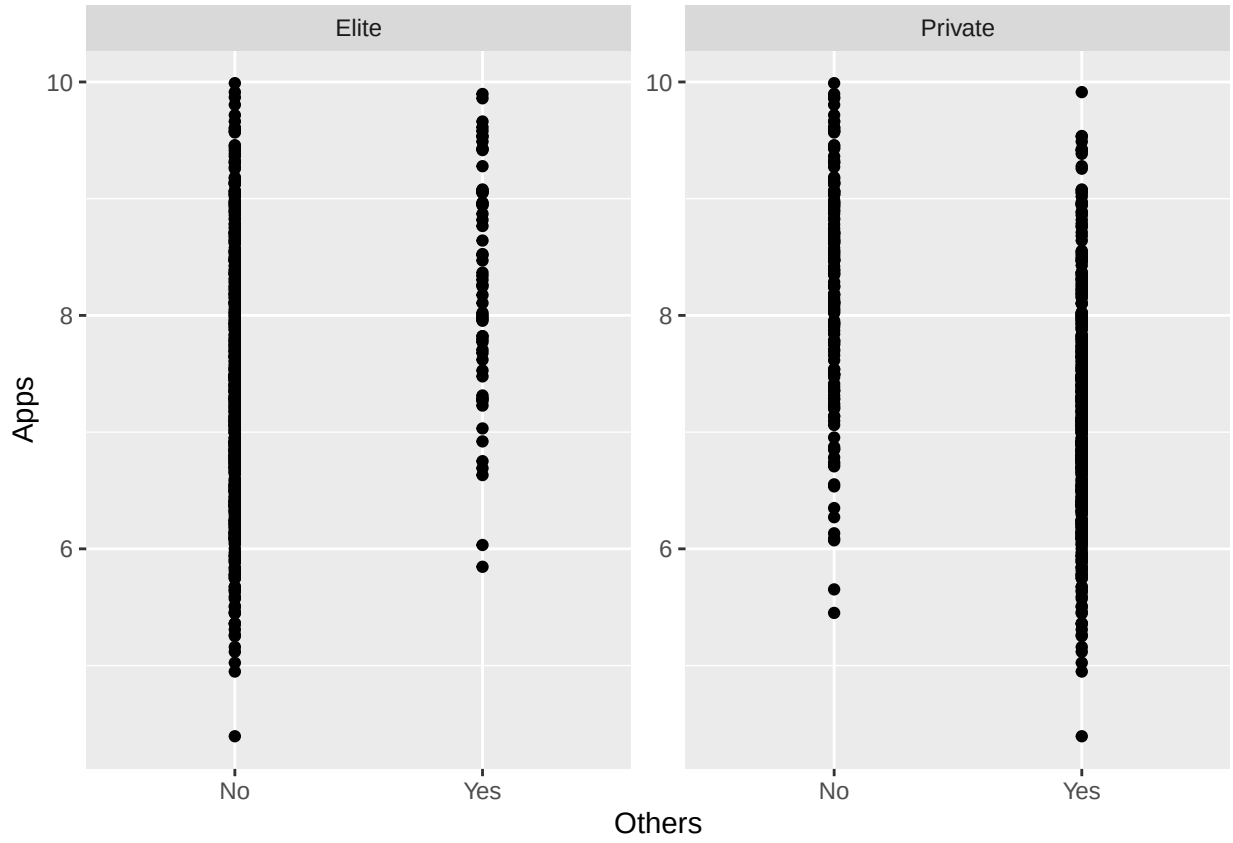


Figure 4: Relationships between Apps and Elite, Private

We draw the scatterplots between Apps and other predictors to find potential relationships. Finally, we decided to use only `Private`, `F.Undergrad`, `Top25perc`, `PhD` so that prevent multilinearity.

```
glm.norm.1 <- glm(data = linear.train.norm,
                  Apps ~ (Private+`F.Undergrad` + Top25perc  + `PhD`)^2,
                  family = gaussian(link = "identity") )

glm.norm.2 <- step(glm.norm.1, k = log(nrow(linear.train.norm)), trace = F)
kable(summary(glm.norm.2)$coefficients,caption = "Summary of the linear model")
```

Table 2: Summary of the linear model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.4367432	0.3430504	1.2731168	0.2034911
PrivateYes	-0.8046522	0.3779185	-2.1291683	0.0336646
F.Undergrad	0.8791448	0.0374829	23.4545763	0.0000000
Top25perc	-0.0044578	0.0025434	-1.7527251	0.0801821
PhD	-0.0000217	0.0000232	-0.9333774	0.3510169
PrivateYes:F.Undergrad	0.1424179	0.0458170	3.1084071	0.0019740
Top25perc:PhD	0.0000015	0.0000004	3.8029606	0.0001583

```
par(mfrow = c(2,2))
plot(glm.norm.2)
```

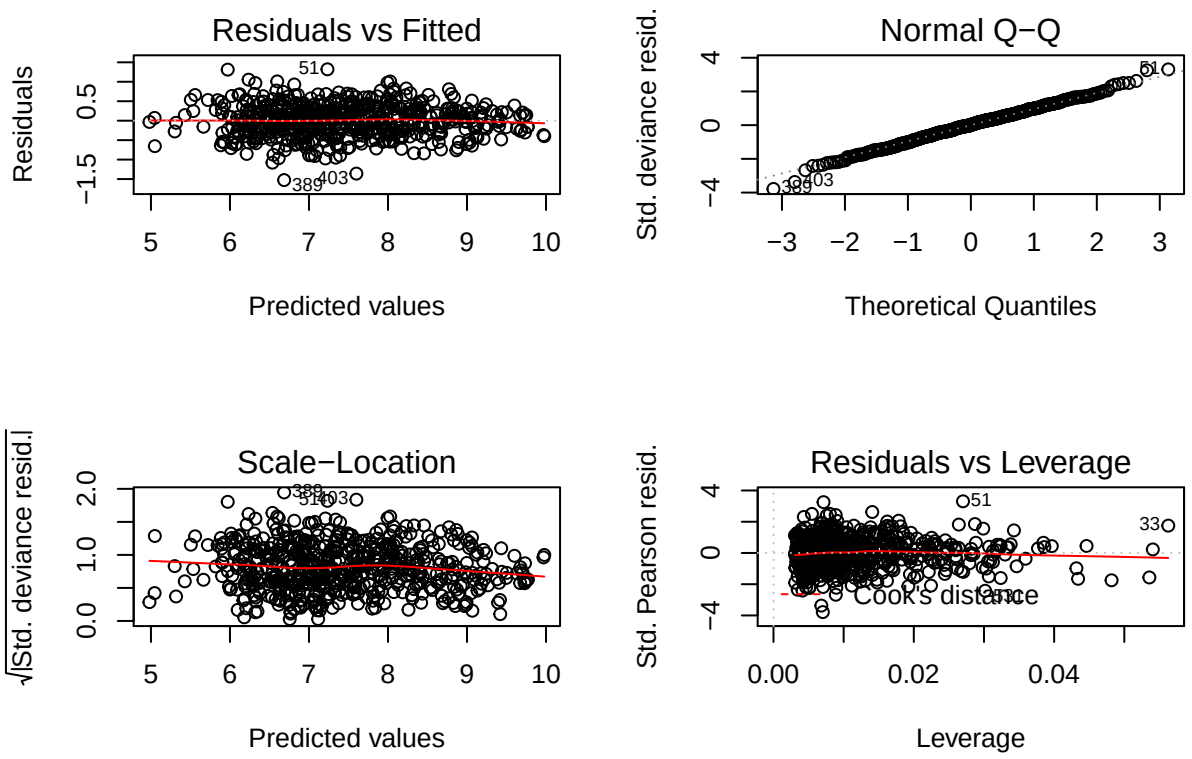


Figure 5: Diagnostic plots for the linear model

First, by EDA of histogram, we could find that some variables have skewed distribution and are not normally distributed. For satisfying assumptions under linear regression, we decided to transform them to make their distribution normal. After transformed, we examined linear relationship between response variable and predictors. We decide to use `Private + F.Understand` + Top25perc + `PhD` which shows strong association with response. For the first model, we included all 2-way interaction term and select significant variables by stepwise selection based on BIC–since the bas.glm() has not implemented Gaussian errors. When we see Diagnostic plots of our model, we could find that most assumptions were satisfied. violation of the constant variance and the unlinear relationship between fitted value versus residual do not exist. Even though violation of normality slightly occured, it is not a severe violation.

2. Generate 1000 replicate data sets using the coefficients from the model you fit above. Using RMSE as a statistic, $\sqrt{\sum_i (y^{\text{rep}} - \hat{y}_i^{\text{rep}})^2/n}$, how does the RMSE from the model based on the training data compare to RMSE's based on the replicated data. What does this suggest about model adequacy? Provide a histogram of the RMSE's with a line showing the location of the observed RMSE and compute a p-value. If you transform the response, you will need to back transform data to the original units in order to compute the RMSE in the original units. Does this suggest that the model is adequate? (complete during lab)

```r
RMSE <- function(err){
  result <- sqrt(mean(err^2))
  return(result)
}
```

```r
nsim <- 1000
n <- nrow(linear.train.norm)
X <- model.matrix(glm.norm.2)
sim.col.lin <- sim(glm.norm.2, nsim)

y.rep.norm <- array(NA,c(nsim,n))
y.hat.norm <- array(NA,c(nsim,n))

for(i in 1:nsim){
  mu <- X %*% sim.col.lin@coef[i,]
  y.rep.norm[i,] <- rnorm(n, mean = mu, sd = sim.col.lin@sigma)
  y.hat.norm[i,] <- mu
}

y.rep.err.norm <- exp(y.rep.norm) - exp(y.hat.norm)
y.err.norm <- exp(linear.train.norm$Apps) - exp(glm.norm.2$fitted.values)

y.rep.RMSE.norm <- apply(y.rep.err.norm, 1, RMSE)
```

```r
ggplot(mapping = aes(y.rep.RMSE.norm)) +
  geom_histogram() +
  geom_vline(xintercept = RMSE(y.err.norm), color = "red")
```

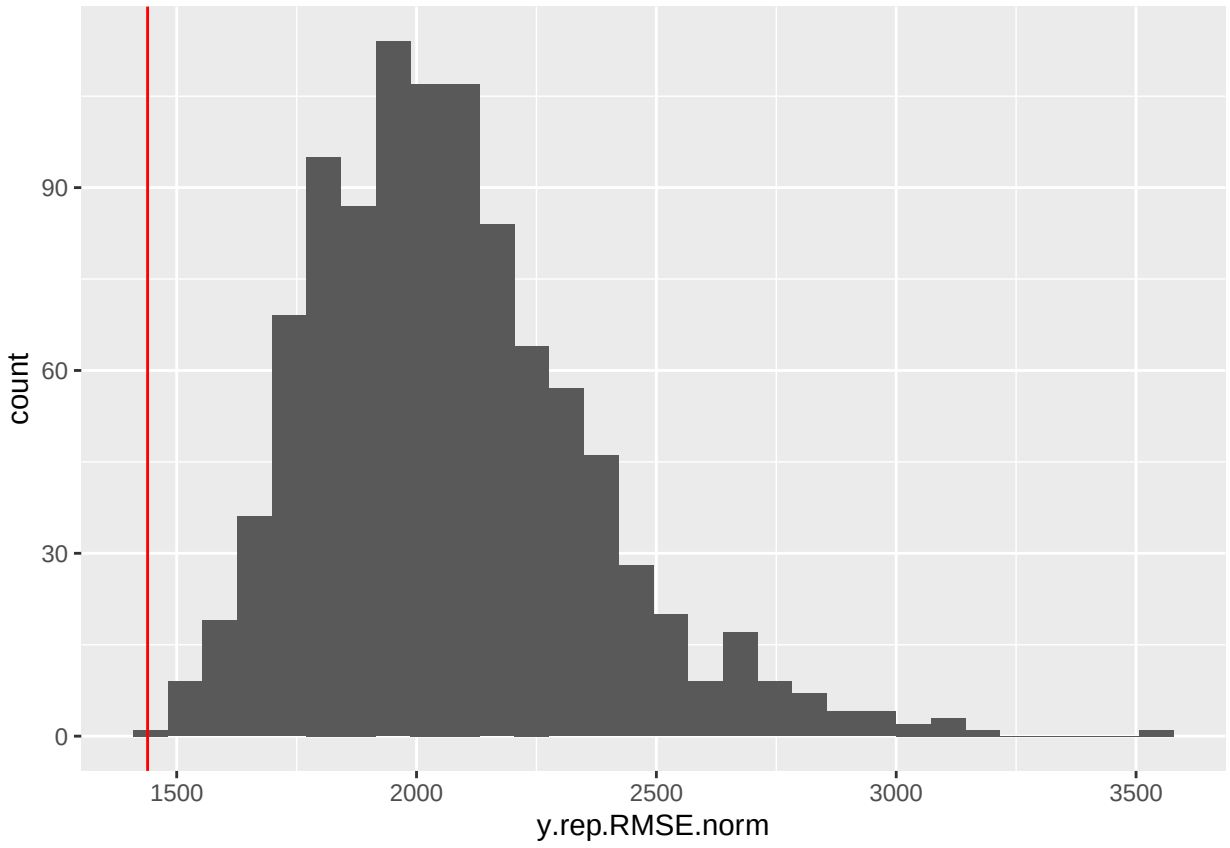


Figure 6: RMSE(Replicated vs Observed) –Linear model

```
mean(y.rep.RMSE.norm > RMSE(y.err.norm))
```

```
## [1] 1
```

RMSE from the model is less than all RMSE from the replicated data, and the p-value is 1. That is RMSE for observed data is much smaller than our model expected. This would suggest that this linear model might overfit, and thus it's not that good.

3. Use your fitted model to predict the number of applications for the testing data, `College.test`. Plot the predicted residuals $y_i - \hat{y}_i$ versus the predictions. Are there any cases where the model does a poor job of predicting? Compute the out-of-sample RMSE using the test data where now RMSE $= \sqrt{\sum_{i=1}^{n.test}(y_i - \hat{y}_i)^2/n.test}$ where the sum is over the test data. (complete during lab)

```
linear.test.norm <- College.test %>%
  mutate(Apps = log(Apps),`F.Undergrad` = log(`F.Undergrad`), Expend = log(Expend),
         `P.Undergrad` = log(`P.Undergrad`), Personal = log(Personal)) %>%
  mutate(Outstate = sqrt(Outstate),`Room.Board` = sqrt(`Room.Board`),
         `perc.alumni` = sqrt(perc.alumni)) %>%
  mutate(PhD = PhD^2, Terminal = Terminal^2)

pred.err.norm <- exp(linear.test.norm$Apps) - exp(predict(glm.norm.2,newdata = linear.test.norm))
```

```
RMSE(pred.err.norm)
```

```
## [1] 3172.233
```

```
p1 <- mean(y.rep.err.norm > RMSE(pred.err.norm))
p1
```

```
## [1] 0.04632818
```

```
ggplot(mapping = aes(x = exp(predict(glm.norm.2, newdata = linear.test.norm)), y = pred.err.norm)) +
  geom_point() +
  geom_smooth() +
  labs(title = "predicted residuals vs predictions", x = "prediction", y = "predicted residual")
```

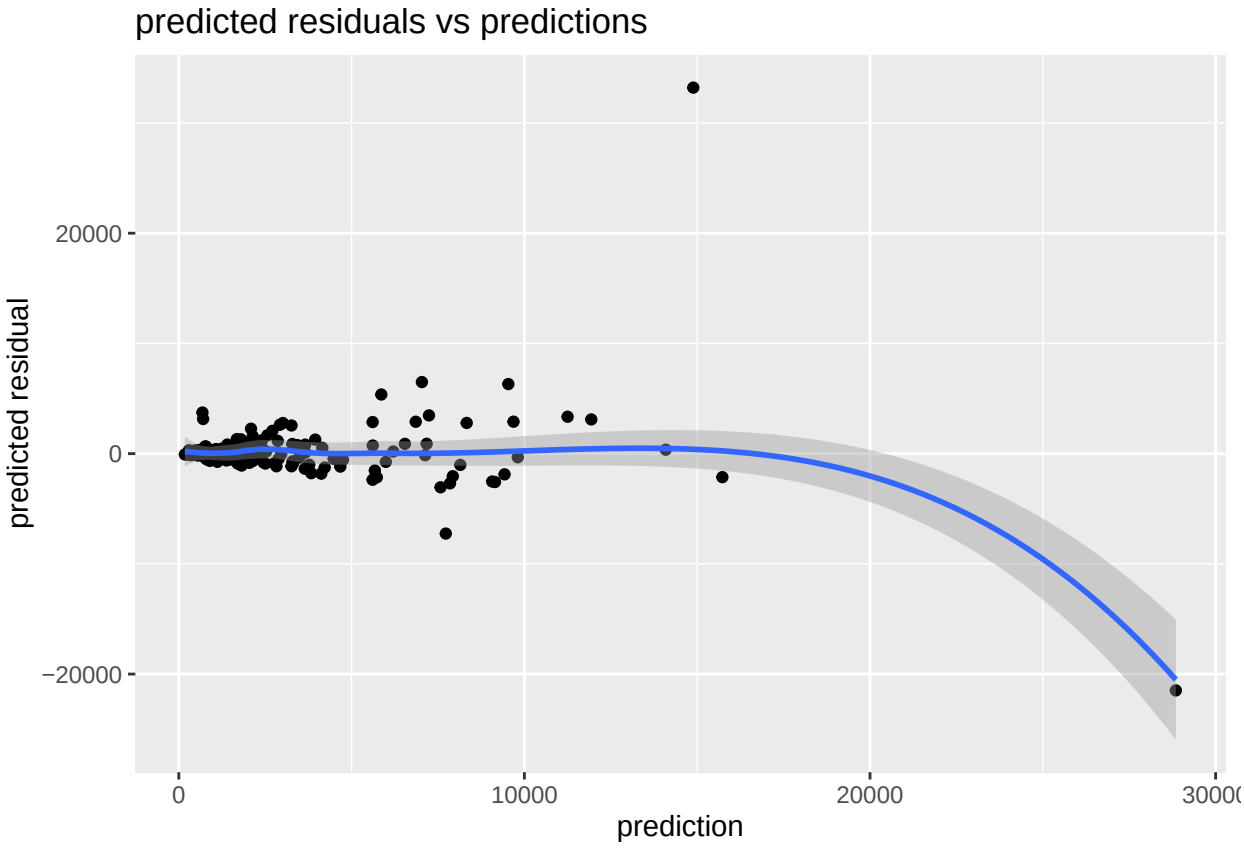


Figure 7: Predicted residuals vs predictions – Linear model

There're a lot of cases where predictions cannot cover. They're not within the predictive interval. This would also suggest that the model is not good enough. The RMSE using the test dataset is 3172.233.

4. As the number of applications is a count variable, a Poisson regression model is a natural alternative for modelling this data. Build a Poisson model using main effects and possible interactions/transformations

(use your model from above as a starting point). Comment on the model adequacy based on diagnostic plots and other summaries. Is there evidence that there is lack of fit?

```
linear.train.pois <- linear.train.norm %>%
  mutate(Apps = exp(Apps))

glm.pois.1 <- glm(Apps ~ (Private+`F.Undergrad` + Top25perc  + `PhD`)^2,
                  data = linear.train.pois, family = poisson(link = "log"))

glm.pois.2 <- step(glm.pois.1, k = log(nrow(linear.train.pois)), trace = F)

kable(summary(glm.pois.2)$coefficients,caption = "Summary of the poisson model")
```

Table 3: Summary of the poisson model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.3605428	0.0387747	-9.298404	0
PrivateYes	-1.1788676	0.0191458	-61.573097	0
F.Undergrad	1.0079263	0.0047011	214.401473	0
Top25perc	0.0031585	0.0005637	5.603136	0
PhD	0.0001157	0.0000062	18.643475	0
PrivateYes:F.Undergrad	0.1949807	0.0023725	82.181953	0
PrivateYes:Top25perc	-0.0008080	0.0001346	-6.002696	0
PrivateYes:PhD	-0.0000127	0.0000015	-8.371304	0
F.Undergrad:Top25perc	-0.0010416	0.0000625	-16.668443	0
F.Undergrad:PhD	-0.0000204	0.0000007	-29.018539	0
Top25perc:PhD	0.0000018	0.0000000	81.641012	0

```
glm.pois.2$deviance/glm.pois.2$df.residual
```

```
## [1] 380.1309
```

```
par(mfrow = c(2,2))
plot(glm.pois.2)
```

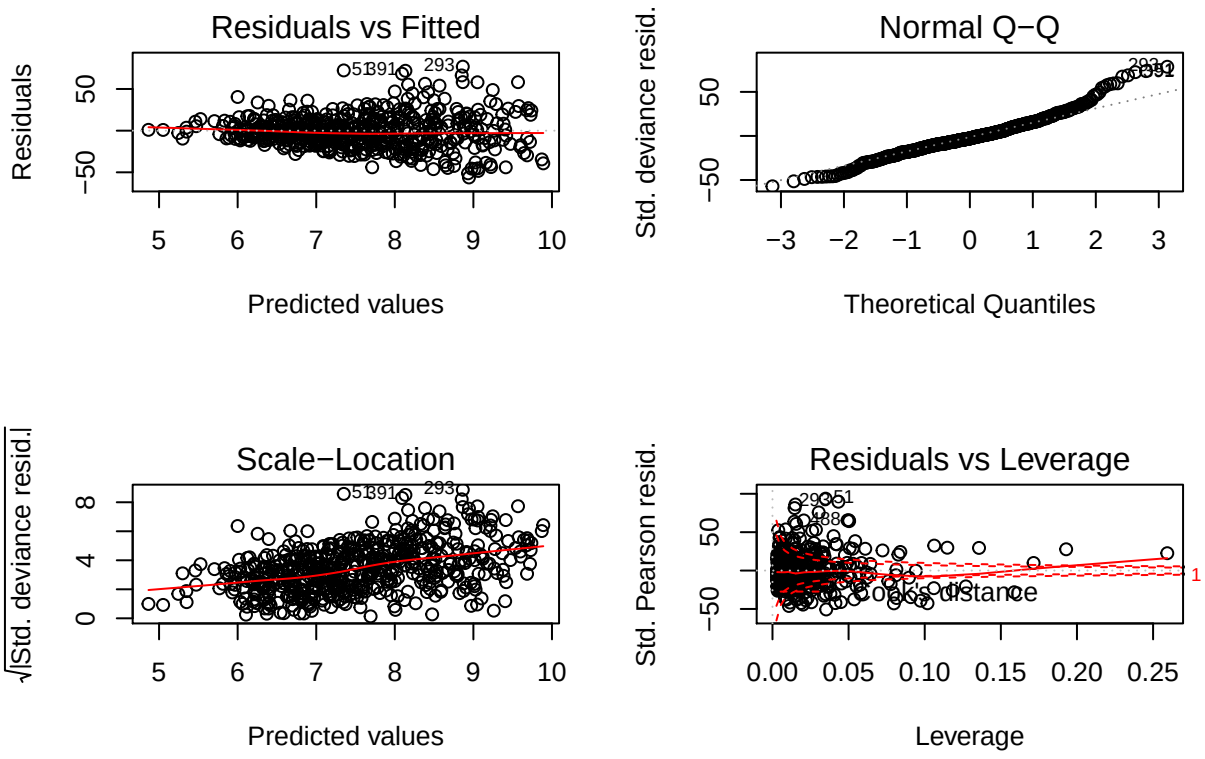


Figure 8: Diagnostic plots for the poisson model

When we see Diagnostic plots of our model, we could find that most assumptions were violated. Severe violation of the constant variance and unlinear relationship between fitted value versus residual does exist. Moreover, violation of normality occured and many observations are considered as outliers. P-values in the summary table would suggest all variables are significant.

Since the Residual Deviance is much greater than the residual df, there's evidence of lack of fit.

```
glm.pois.BIC = bas.glm(Apps ~ (Private+`F.Undergrad` + Top25perc  + `PhD`)^2, data=linear.train.pois,
family=poisson(),
method="MCMC",
n.models=256, MCMC.iterations=10000,
betaprior=bic.prior(n = nrow(linear.train.pois)),
modelprior=uniform())
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
image(glm.pois.BIC, rotate = F)
```

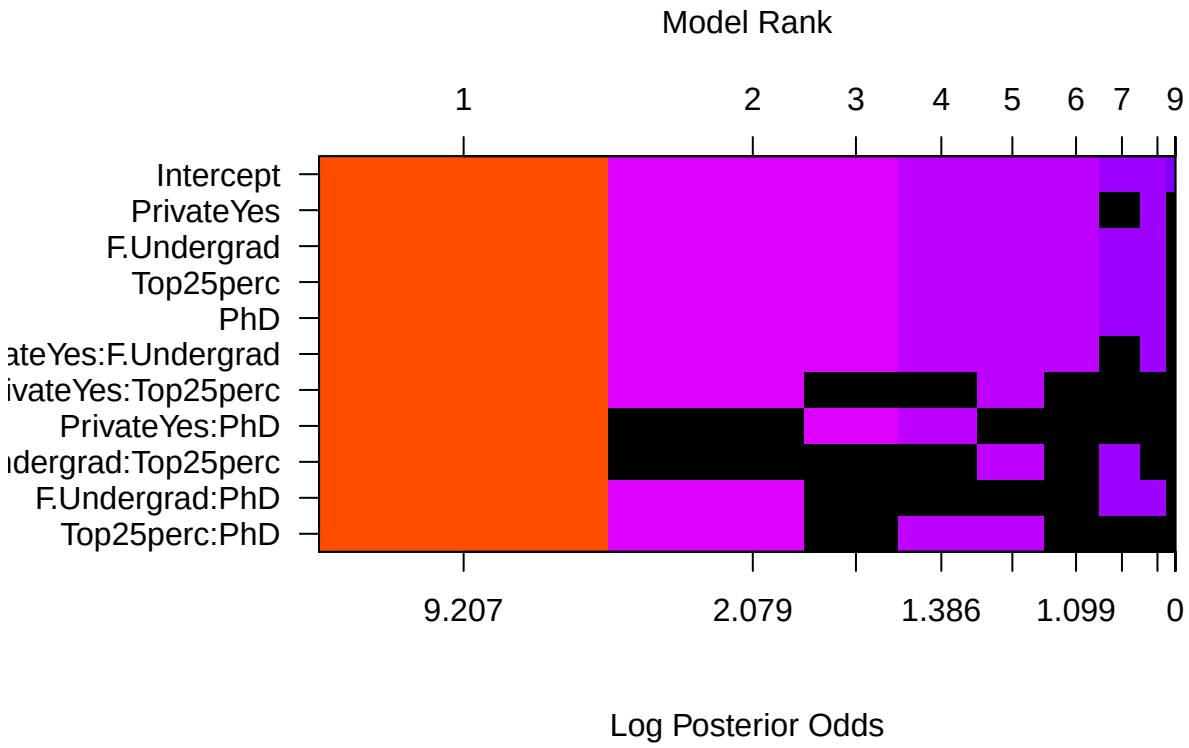


Figure 9: plot for BAS BIC

After using stepwise method, we try another model selection method: Bayesian Adaptive Sampling. From the image above, we can see that this method selects the same model as does the stepwise selection–it has the highest log posterior odds. Therefore, we will stick to this model for our further analysis.

5. Generate 1000 replicate data sets using the coefficients from the Poisson model you fit above. Using RMSE as a statistic, $\sqrt{\sum_i (y^{\text{rep}} - \hat{y}_i^{\text{rep}})^2/n}$, how does the RMSE from the model based on the training data compare to RMSE's based on the replicated data. What does this suggest about model adequacy? Provide a histogram of the RMSE's with a line showing the location of the observed RMSE and compute a p-value.

```
nsim <- 1000
n <- nrow(linear.train.pois)
X <- model.matrix(glm.pois.2)
sim.col.lin <- sim(glm.pois.2, nsim)

y.rep.pois <- array(NA,c(nsim,n))
y.hat.pois <- array(NA,c(nsim,n))

for(i in 1:nsim){
  mu <- exp(X %*% sim.col.lin@coef[i,])
  y.rep.pois[i,] <- rpois(n,lambda = mu)
```

```
  y.hat.pois[i,] <- mu
}

y.rep.err.pois <- y.rep.pois - y.hat.pois
y.err.pois <- linear.train.pois$Apps - glm.pois.2$fitted.values
y.rep.RMSE.pois <- apply(y.rep.err.pois, 1, RMSE)
```

```
ggplot(mapping = aes(y.rep.RMSE.pois)) +
  geom_histogram(bins = 60) +
  geom_vline(xintercept = RMSE(y.err.pois), color = "red")
```

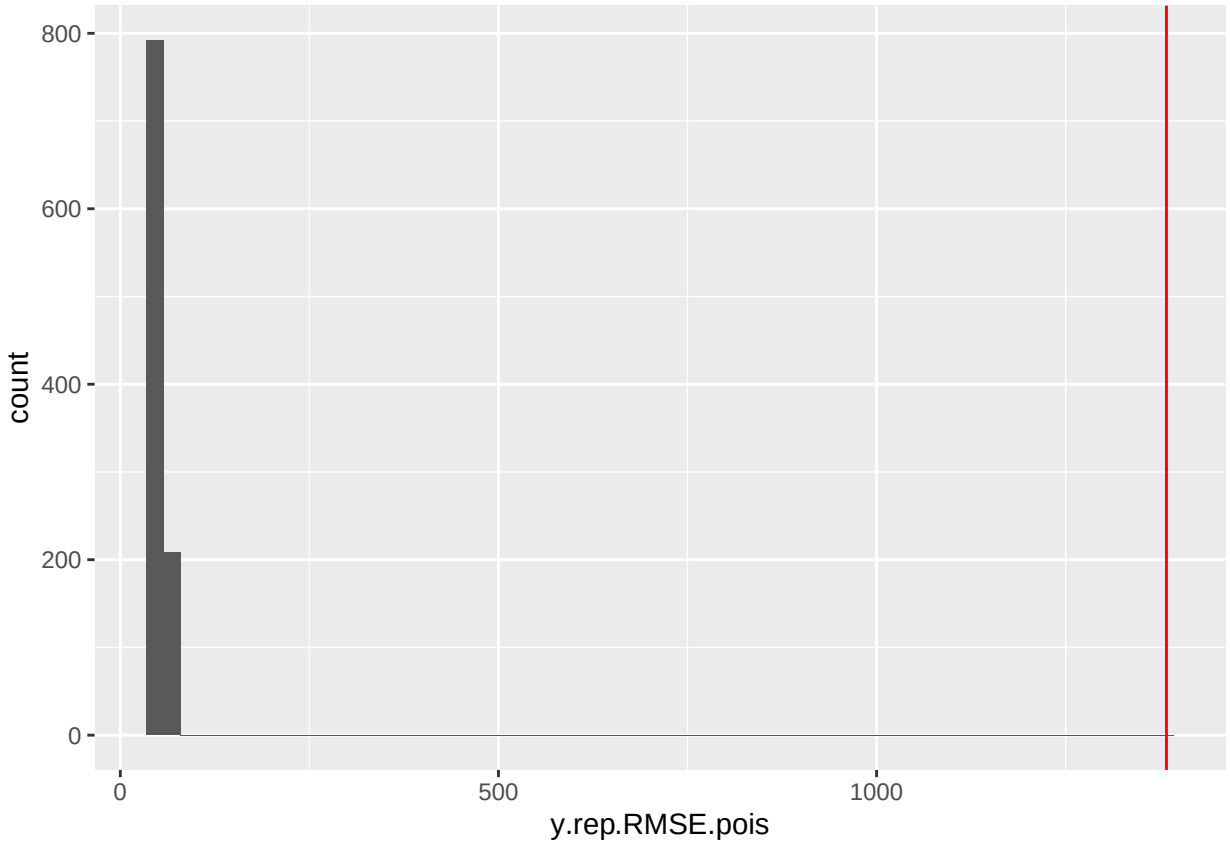


Figure 10: RMSE(Replicated vs observed) –Poisson model

```
mean(y.rep.RMSE.pois > RMSE(y.err.pois))
```

```
## [1] 0
```

RMSE from the observed data is much greater than all RMSE from the replicated data, and the p-value is 0. This would suggest that there's the lack of fit, and thus the model is not good.

6. Using the test data set, calculate the RMSE for the test data (see lab) using the predictions from the Poisson model. How does this compare to the RMSE based on the observed data? Is this model better than the linear regression models in terms of out of sample prediction?

```r
linear.test.pois <- linear.test.norm %>%
  mutate(Apps = exp(Apps))

pred.err.pois <- linear.test.pois$Apps - predict(glm.pois.2,newdata = linear.test.pois)

RMSE(pred.err.pois)
```

```
## [1] 5431.143
```

```r
RMSE(y.err.pois)
```

```
## [1] 1383.284
```

```r
p2 <- mean(y.rep.err.pois > RMSE(pred.err.pois))
p2
```

```
## [1] 0
```

```r
ggplot(mapping = aes(y.rep.RMSE.pois)) +
  geom_histogram() +
  geom_vline(xintercept = RMSE(pred.err.pois), color = "red")
```

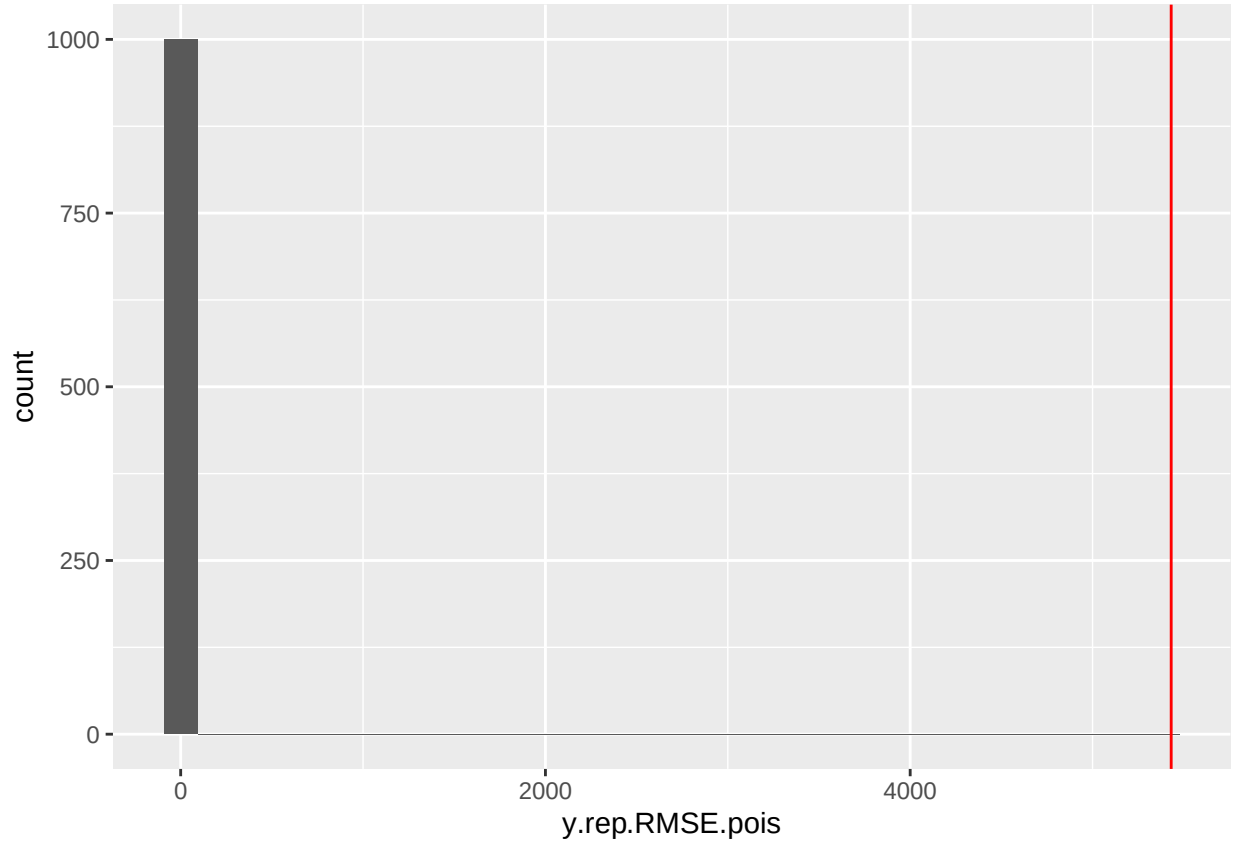


Figure 11: RMSE(test data vs observed)–Poisson Model

RMSE for the test data is calculated to be 5431.136. This RMSE is much greater than the RMSE based on the observed data, which is 1383.284. There's no evidence suggesting that this model is better than the linear model. Because RMSE from poisson model is much larger than RMSE from normal model for out of sample data.

7. Build a model using the negative binomial model (consider transformations and interactions if needed) and examine diagnostic plots. Are there any suggestions of problems with this model?

```
glm.nb.1 <- glm.nb(Apps ~ (Private+`F.Undergrad` + Top25perc  + `PhD`)^2, data = linear.train.pois)
glm.nb.2 <- step(glm.nb.1,k = log(nrow(linear.train.pois)), trace = F)

kable(summary(glm.nb.2)$coefficients,caption = "Summary of the Negative Binomial Model")
```

Table 4: Summary of the Negative Binomial Model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.7059016	0.3346151	2.109593	0.0348935
PrivateYes	-0.9194144	0.3688322	-2.492772	0.0126750
F.Undergrad	0.8634285	0.0365508	23.622706	0.0000000
Top25perc	-0.0048608	0.0024831	-1.957582	0.0502791
PhD	-0.0000260	0.0000227	-1.145898	0.2518375
PrivateYes:F.Undergrad	0.1542148	0.0447130	3.448996	0.0005627
Top25perc:PhD	0.0000015	0.0000004	3.890129	0.0001002

```
par(mfrow = c(2,2))
plot(glm.nb.2)
```

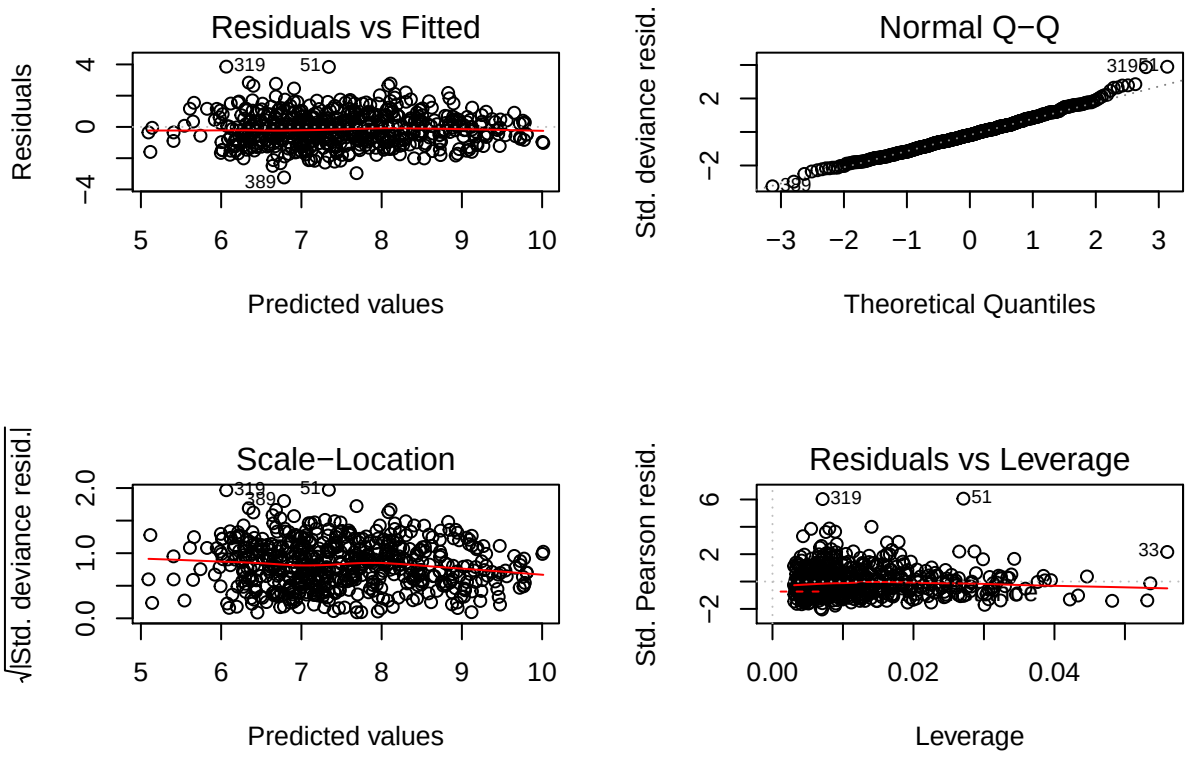


Figure 12: Diagnostic plots for the Negative Binomial Model

When we see Diagnostic plots of our model, we could find that most assumptions were satisfied. Violation of the constant variance and unlinear relationship between fitted value versus residual do not exist. Even though violation of normality slightly occured, it is not a severe violation. P-values in the summary table would suggest all variables are significant. Thus, there're no suggestions of obvious problems

8. Carry out the predictive checks for the negative model model using simulated replicates with RMSE and add RMSE from the test data and observed data to your plot. What do these suggest about 1) model adequacy and 2) model comparison? Which model out of all that you have fit do you recommend?

```
nsim <- 1000
n <- nrow(linear.train.pois)
X <- model.matrix(glm.nb.2)
class(glm.nb.2) <- c("glm", "lm")
sim.col.nb <- sim(glm.nb.2, nsim)
sim.col.nb@sigma = rnorm(nsim,glm.nb.2$theta,glm.nb.2$SE.theta)

y.rep.nb <- array(NA,c(nsim,n))
y.hat.nb <- array(NA,c(nsim,n))

for(i in 1:nsim){
  mu <- exp(X %*% sim.col.nb@coef[i,])
```

```
  y.rep.nb[i,] <- rnegbin(n,mu = mu,theta = sim.col.nb@sigma[i])
  y.hat.nb[i,] <- mu
}

y.rep.err.nb <- y.rep.nb - y.hat.nb
y.err.nb <- linear.train.pois$Apps - glm.nb.2$fitted.values
pred.err.nb <- linear.test.pois$Apps - predict(glm.nb.2,newdata = linear.test.pois)
y.rep.RMSE.nb <- apply(y.rep.err.nb, 1, RMSE)
```

```
RMSE(pred.err.nb)
```

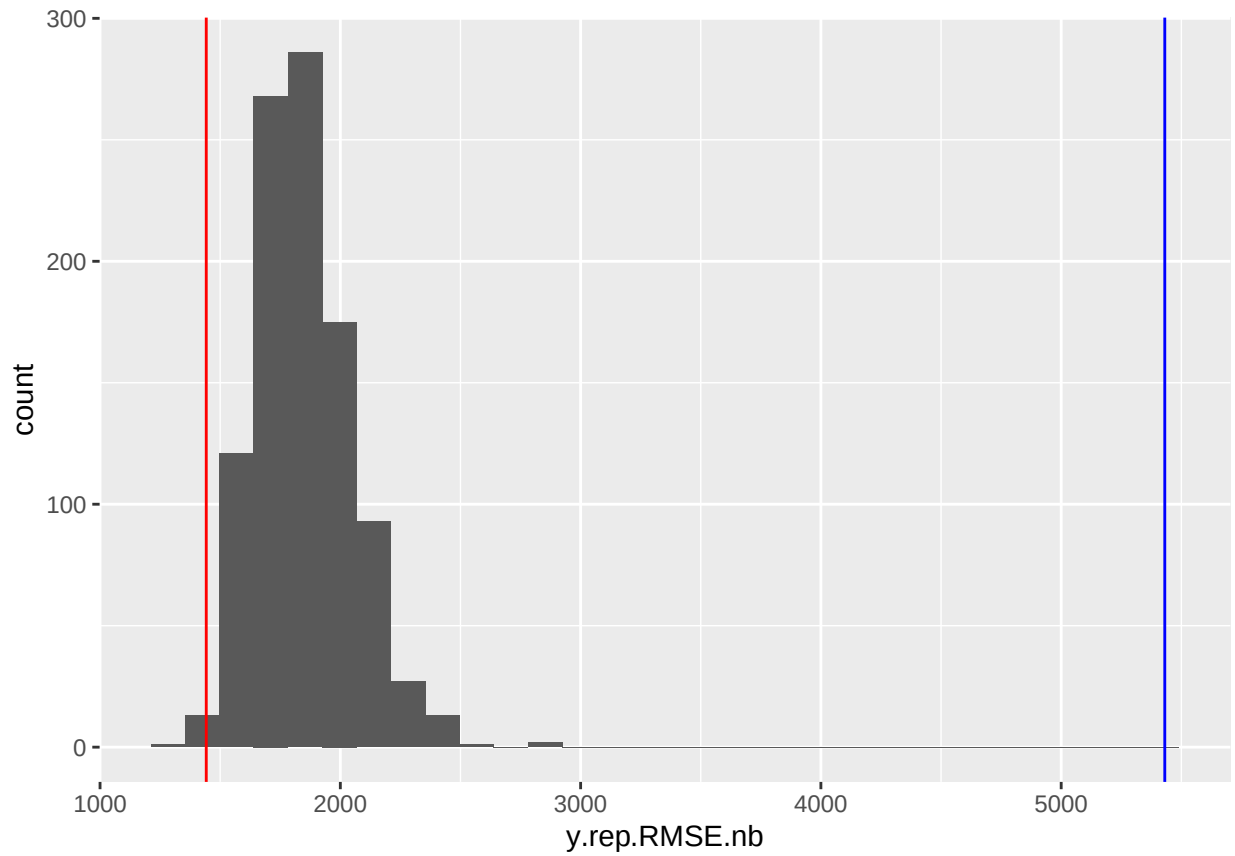
```
## [1] 5431.138
```

```
RMSE(y.err.nb)
```

```
## [1] 1442.009
```

```
ggplot(mapping = aes(y.rep.RMSE.nb)) +
  geom_histogram() +
  geom_vline(xintercept = RMSE(y.err.nb), color = "red") +
  geom_vline(xintercept = RMSE(pred.err.nb), color = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
mean(y.rep.RMSE.nb > RMSE(y.err.nb))
```

```
## [1] 0.993
```

```r
p3 <- mean(y.rep.err.pois > RMSE(pred.err.pois))
p3
```

```
## [1] 0
```

RMSE from the model (observed) is in the region of the RMSE from the replicated data, but the RMSE of the test data is still much bigger. We still don't have strong evidence suggesting that the negative binomial model is adequate enough. However, among all three models, the negative binomial model would be the best in terms of RMSE. Thus, we would recommend the negative binomial model.

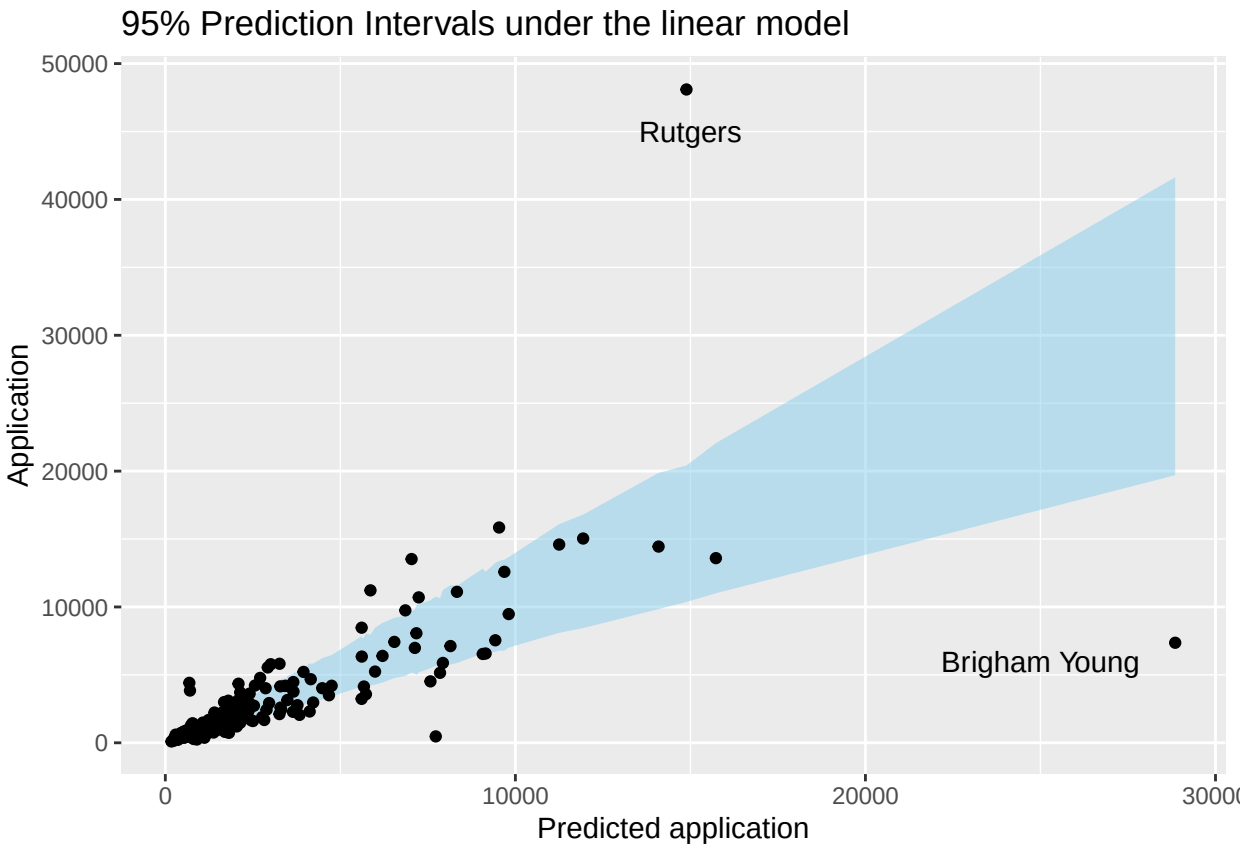
9. While RMSE is a popular summary for model goodness of fit, coverage of confidence intervals is an alternative. For each case in the test set, find a 95% prediction interval based on the observed data. Now evaluate if the response in the test data are inside or outside of the intervals. If we have the correct coverage, we would expect that at least 95% of the intervals would contain the test cases. Write a function to calculate coverage (the input should be the fitted model object and the test data-frame) and then evaluate coverage for each of the models that you fit (normal, Poisson and negative binomial). Include plots of the confidence intervals versus case number ordered by the prediction, with the left out data added as points. Comment on the plots, highlighting any unusual colleges where the model predicts poorly. (See code from lecture)

```r
coverage <- function(y, bound){
  mean(y >= bound[,1] & y <= bound[,2])
}
bound.norm <- function(model, newdata, level = 0.95, nsim = 1000){
  require(mvtnorm)
  n <- nrow(newdata)
  X <- model.matrix(model, data = newdata)
  beta <- rmvnorm(nsim, coef(model), vcov(model))
  y.rep <- matrix(NA, nsim, n)

  for(i in 1:nsim){
    mu = X %*% beta[i,]
    y.rep[i,] <- rnorm(n, mean = mu, sd = summary(model)$dispersion)
  }

  bound <- t(apply(exp(y.rep), 2, function(x) {
    quantile(x, c((1-level)/2, 0.5 + level/2))
  }))
  return(bound)
}
bound.test.norm <- bound.norm(glm.norm.2, linear.test.norm)
coverage.norm <- coverage(exp(linear.test.norm$Apps),bound.test.norm)
#prediction interval plot
df.norm <- data.frame(college = linear.test.norm$college,
                      Apps = exp(linear.test.norm$Apps),
                      pred = exp(predict(glm.norm.2, linear.test.norm, type = "response")),
                      lwr = bound.test.norm[,1], upr = bound.test.norm[,2]) %>%
  arrange(pred)
```

```
ggplot(df.norm, aes(x = pred, y = Apps)) +
  geom_ribbon(aes(ymin = lwr, ymax = upr),
              fill = "skyblue", alpha = 0.5) +
  geom_point() +
  #geom_text() +
  annotate("text", x = c(15000,25000), y = c(45000,6000), label = c("Rutgers", "Brigham Young")) +
  labs(x = "Predicted application", y = "Application",
       title = "95% Prediction Intervals under the linear model")
```



The linear model has a decent coverage rate: 0.5897436. So one can expect that about 60% of the colleges would receive some number of applications between the interval.

Two colleges that the model predicts poorly are Rutgers and Brigham Young University. Gutgers receives a lot more applications than the prediction, while Brigham Young receives fewer than suggested by the model.

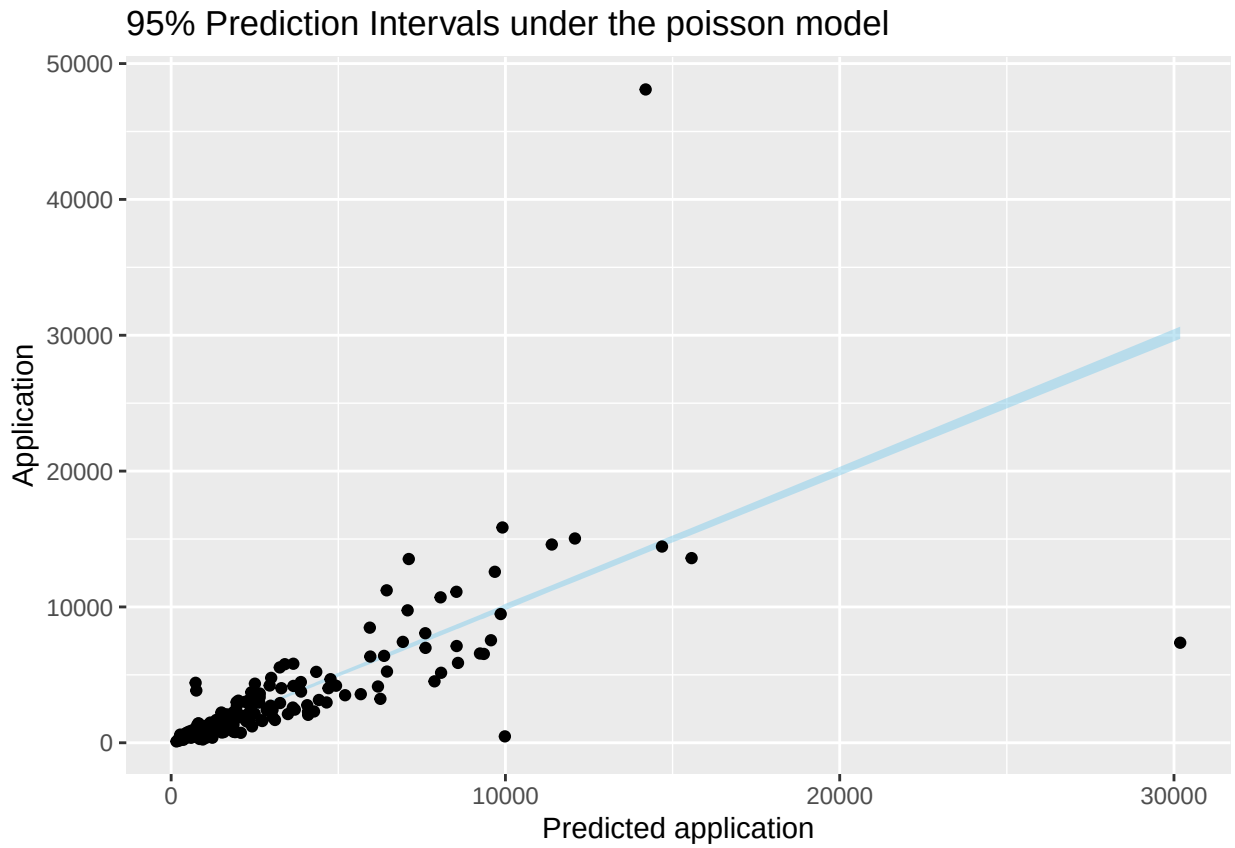
```
bound.pois <- function(model, newdata, level = 0.95, nsim = 1000){
  require(mvtnorm)
  n <- nrow(newdata)
  X <- model.matrix(model, data = newdata)
  beta <- rmvnorm(nsim, coef(model), vcov(model))
  y.rep <- matrix(NA, nsim, n)

  for(i in 1:nsim){
    mu = exp(X %*% beta[i,])
    y.rep[i,] <- rpois(n, lambda = mu)
  }
```

```
  bound <- t(apply(y.rep, 2, function(x) {
    quantile(x, c((1-level)/2, 0.5 + level/2))
  }))
  return(bound)
}
bound.test.pois <- bound.pois(glm.pois.2, linear.test.pois)
coverage.pois <- coverage(linear.test.pois$Apps, bound.test.pois)
```

```
#prediction interval plot
df.pois <- data.frame(Apps = linear.test.pois$Apps,
                      pred = predict(glm.pois.2, linear.test.pois, type = "response"),
                      lwr = bound.test.pois[,1], upr = bound.test.pois[,2]) %>%
  arrange(pred)

ggplot(df.pois, aes(x = pred, y = Apps)) +
  geom_ribbon(aes(ymin = lwr, ymax = upr),
              fill = "skyblue", alpha = 0.5) +
  geom_point() +
  labs(x = "Predicted application", y = "Application",
       title = "95% Prediction Intervals under the poisson model")
```



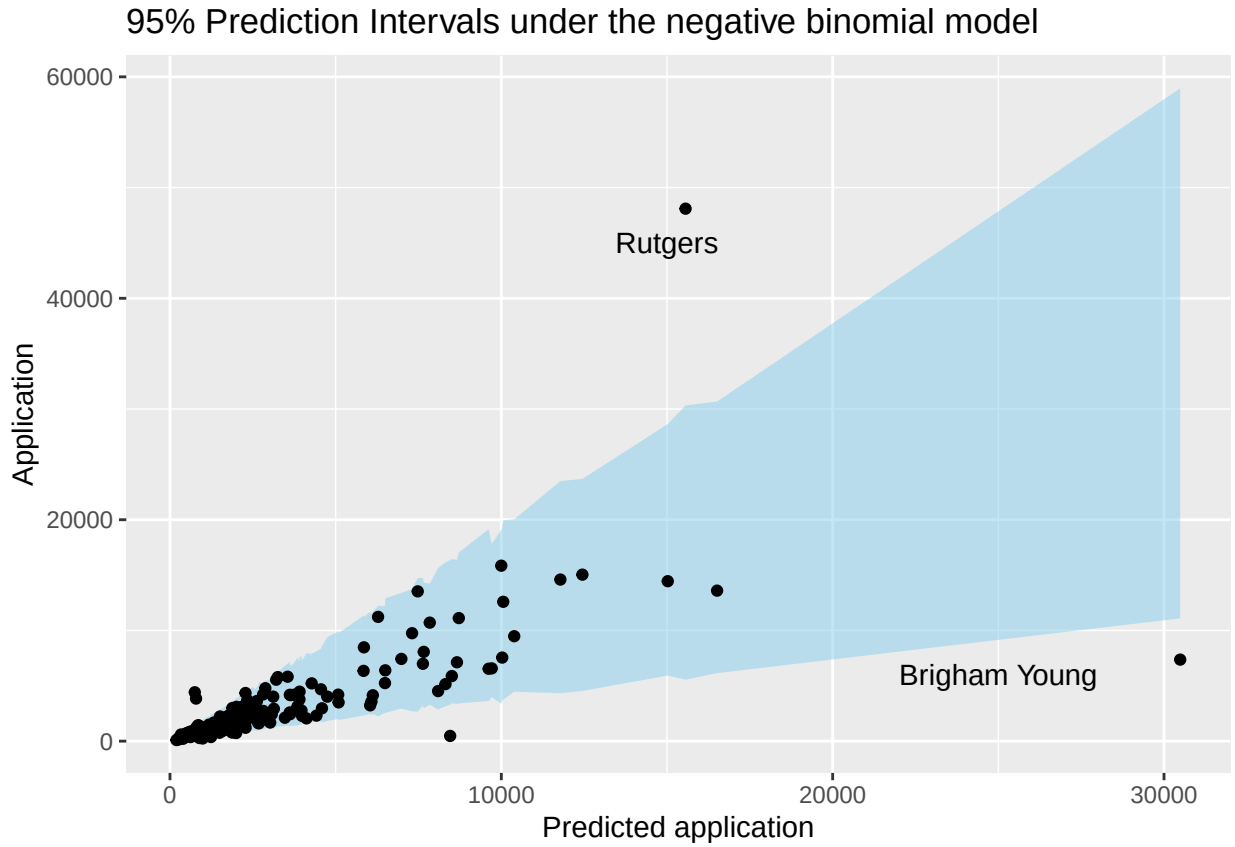
Overall, the Poisson model does poorly on everything–its 95% prediction interval hardly covers any colleges.

```r
bound.nb <- function(model, newdata, level = 0.95, nsim =1000){
  require(mvtnorm)
  n <- nrow(newdata)
  X <- model.matrix(model, data = newdata)
  beta <- rmvnorm(nsim, coef(model), vcov(model))
  theta <- rnorm(nsim, model$theta, model$SE.theta)
  y.rep <- matrix(NA, nsim, n)

  for(i in 1:nsim){
    mu = exp(X %*% beta[i,])
    y.rep[i,] <- rnegbin(n, mu = mu, theta = theta[i])
  }
  bound = t(apply(y.rep, 2, function(x) {
    quantile(x, c((1-level)/2, 0.5 + level/2))
  }))
  return(bound)
}
bound.test.nb <- bound.nb(glm.nb.2, newdata = linear.test.pois)
coverage.nb <- coverage(linear.test.pois$Apps, bound.test.nb)
```

```r
#prediction interval plot
df.nb <- data.frame(Apps = linear.test.pois$Apps,
                      pred = predict(glm.nb.2, linear.test.norm, type = "response"),
                      lwr = bound.test.nb[,1], upr = bound.test.nb[,2]) %>%
  arrange(pred)

ggplot(df.nb, aes(x = pred, y = Apps)) +
  geom_ribbon(aes(ymin = lwr, ymax = upr),
              fill = "skyblue", alpha = 0.5) +
  geom_point() +
  annotate("text", x = c(15000,25000), y = c(45000,6000), label = c("Rutgers", "Brigham Young")) +
  labs(x = "Predicted application", y = "Application",
       title = "95% Prediction Intervals under the negative binomial model")
```



The Negative Binomial model does a great job predicting the number of applications received by colleges, with a coverage rate: 0.9384615.

Even though it covers most of the colleges, Rutgers and Brigham Young are still outside the coverage.

10. Provide a table with the 1) RMSE's on the observed data, 2) RMSE's on the test data, 3) coverage, 4) the predictive check p-value with one row for each of the models and comment the results. Which model do you think is best and why? Consider the job of an administrator who wants to ensure that there are enough staff to handle reviewing applications. Explain why coverage might be useful.

```
summary.data <- data.frame(RMSE.obs = c(RMSE(y.err.norm),RMSE(y.err.pois),RMSE(y.err.nb)),
                           RMSE.test = c(RMSE(pred.err.norm),
                                         RMSE(pred.err.pois),RMSE(pred.err.nb)),
                           Coverage = c(coverage.norm,coverage.pois,coverage.nb),
                           p.value = c(p1,p2,p3))
rownames(summary.data) = c("Linear", "Poisson", "NB")

kable(summary.data, digits = 3, row.names = T,
      col.names = c("Training RMSE", "Testing RMSE", "Prediction Coverage Rate","Predictive check p-valu
      caption = "Summary of RMSE on observed, RMSE on test, coverage and p-value")
```

Table 5: Summary of RMSE on observed, RMSE on test, coverage and p-value

	Training RMSE	Testing RMSE	Prediction Coverage Rate	Predictive check p-value
Linear	1439.292	3172.233	0.590	0.046
Poisson	1383.284	5431.143	0.077	0.000
NB	1442.009	5431.138	0.938	0.000

We would use "coverage" as our main checking criterion, because prediction intervals and the coverage do take into account (over) dispersion, and the coverage could measure how much data the model captures. From the table shown above, one who is interested in predicting the application rate would want to choose the negative binomial model over the other two, since despite of the similar observed RMSEs for all of them, this negative binomial model has an outstanding coverage rate: 0.9384615.

11. For your "best" model provide a nicely formatted table (use `kable()` or `xtable()`) of relative risks and 95% confidence intervals. Pick 5 of the most important variables and provide a paragraph that provides an interpretation of the parameters (and intervals) that can be provided to a university admissions officer about which variables increase admissions.

The final model is written as the following:

$$log(Apps) = 0.7059 - 0.9194 * Private + 0.8634285 * log(F.Undergrad) - 0.0049 * Top25perc - 0.000026 * PhD^2 + 0.154215 * (Private * log(F.Undergrad)) + 0.0000015 * (Top25perc * PhD^2)$$

```
relative.risk = data.frame(cbind(coef(summary(glm.nb.2)), confint(glm.nb.2))) %>%
  select(Estimate, X2.5.., X97.5..) %>%
  mutate_all(exp)
```

```
## Waiting for profiling to be done...
```

```
row.names(relative.risk) = c("(Intercept)", "PrivateYes", "log(F.Undergrad)", "Top25perc",
                             "PhD^2", "PrivateYes:log(F.Undergrad)", "Top25perc:PhD^2" )

kable(relative.risk, col.names = c("Relative Risks", "2.5%", "97.5%"),
      caption = "Relative risks for our best model: Negative Binomial", digits = 3)
```

Table 6: Relative risks for our best model: Negative Binomial

	Relative Risks	2.5%	97.5%
(Intercept)	2.026	0.993	4.173
PrivateYes	0.399	0.181	0.874
log(F.Undergrad)	2.371	2.191	2.565
Top25perc	0.995	0.990	1.000
PhD^2	1.000	1.000	1.000
PrivateYes:log(F.Undergrad)	1.167	1.061	1.284
Top25perc:PhD^2	1.000	1.000	1.000

We choose six important variables Private, log(F.Undergrad), Top25perc, PhD^2, Private*log(F.Undergrad) and Top25perc*PHDˆ2 in our final model.

$$\frac{Y_{private}}{Y_{public}} = e^{\beta_1} e^{\beta_5 log(F.undergrad)}$$

Being a private school, the number of applications is 1.97 times the number of application of public school assuming that the number of full time undergradte student is 31571 with 95% confidence interval of (0.334 ,11.64) times.

Being a private school, the number of applications is 0.85 times the number of application of public school assuming that the number of full time undergradte student is 138 with 95% confidence interval of (0.242 ,2.99) times.

$$Y_{private} = e^{\beta_0} e^{\beta_1} F^{\beta_2+\beta_5} Y_{public} = e^{\beta_0} F^{\beta_2}$$

Being a private school, the number of applications is 1.10 times the original assuming that the number of full time undergraduate increase 10% holding other variables constant with 95% confidence interval of (1.083, 1.120) times.

Being a public school, the number of applications is 1.085 time the original assuming that the number of full time undergraduate increase 10% holding other variables constant with 95% confidence interval of (1.077, 1.093) times.

PhDˆ2 and Top25perc do not have significant effects on the number of applications received– since the confidence intervals contain 1.

Some Theory (work together!)

12. Gamma mixtures of Poissons: From class we said that

$$Y \mid \lambda \sim P(\lambda) \tag{1}$$

$$p(y \mid \lambda) = \frac{\lambda^y e^{-\lambda}}{y!} \tag{2}$$

$$\tag{3}$$

$$\lambda \mid \mu, \theta \sim G(\theta, \theta/\mu) \tag{4}$$

$$p(\lambda \mid \mu, \theta) = \frac{(\theta/\mu)^\theta}{\Gamma(\theta)} \lambda^{\theta-1} e^{-\lambda\theta/\mu} \tag{5}$$

$$\tag{6}$$

$$p(Y \mid \mu, \theta) = \int p(Y \mid \lambda) p(\lambda \mid \theta, \theta/\mu) d\lambda \tag{7}$$

$$= \frac{\Gamma(y+\theta)}{y!\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^\theta \left(\frac{\mu}{\theta+\mu}\right)^y \tag{8}$$

$$Y \mid \mu, \theta \sim \mathsf{NB}(\mu, \theta) \tag{9}$$

Derive the density of $Y \mid \mu, \theta$ in (8) showing your work using LaTeX expressions. (Note this may not display if the output format is html, so please use pdf.) Using iterated expectations with the Gamma-Poisson mixture, find the mean and variance of Y, showing your work.

$$
\begin{aligned}
p(y|\mu,\theta) &= \int_0^{+\infty} p(Y \mid \lambda) p(\lambda \mid \theta, \mu) d\lambda \\
&= \int_0^{+\infty} \frac{\lambda^y e^{-\lambda}}{y!} \frac{(\theta/\mu)^\theta}{\Gamma(\theta)} \lambda^{\theta-1} e^{-\lambda\theta/\mu} d\lambda \\
&= \frac{(\theta/\mu)^\theta}{y!\Gamma(\theta)} \frac{\Gamma(y+\theta)}{(1+\theta/\mu)^{y+\theta}} \int_0^{+\infty} Gamma(y+\theta, 1+\theta/\mu) d\lambda \\
&= \binom{y+\theta-1}{y} \left(\frac{\theta}{\theta+\mu}\right)^\theta \left(\frac{\mu}{\theta+\mu}\right)^y \\
&\sim \text{Negative Binomial } (\theta, \frac{\mu}{\theta+\mu})
\end{aligned}
$$

where
$E[Y|\lambda] = Var(Y|\lambda) = \lambda,$
$E[\lambda|\mu,\theta] = \theta/(\theta/\mu) = \mu,$
$Var(\lambda|\mu,\theta) = \theta/(\theta/\mu)^2 = \mu^2/\theta$
$\Longrightarrow$
$E[Y] = E[E[Y|\lambda]] = E[\lambda] = \mu$
$Var(Y) = E[Var(Y|\lambda)] + Var(E[Y|\lambda])$
$= E[\lambda] + Var(\lambda)$
$= \mu + \mu^2/\theta$