Assignment 10

Based on your previous assignments concerning predictors and classifiers, do the following:

1. Create *a 10-fold cross* validation of your predictor (linear regression) OR classifier (logistic regression, decision tree, or other). Provide a summary table or graphic of the RMSEs OR Accuracies from the cross validation.

Hint:

The `crossv_kfold` command creates a list of datasets from our original dataset, each of which contains a testing and training dataset. The proportion of cases held out for testing is determined by the number of folds: 10 folds would indicate 1/10 of the data to be held out.

```{r}
pd_cf<-pd%>%
  crossv_kfold(10)
pd_cf
```

The `pd_cf` dataset is now a nested dataset, as described in (Chapter 25)[http://r4ds.had.co.nz/many-models.html] of the Wickham r4ds book.

The next bit of code is key. It starts by converting all of the individual training datasets to tibbles. Then the model is run on each training dataset. Then apply the predictions from the model to each testing dataset, and finally pull the rmse from each of the testing datasets.

```{r}
tic()
rmse_mod1<-pd_cf %>%
  mutate(train = map(train, as_tibble)) %>% ## Convert to tibbles
  mutate(model = map(train, ~ lm(mod1_formula,
                                 data = .))) %>%
  mutate(rmse = map2_dbl(model, test, rmse)) %>% ## apply model, get rmse
  select(.id, rmse) ## pull just id and rmse
toc()
```

The resulting dataset includes the id for the cross validation and the rmse. We can summarize and plot this new data frame to see what our likely range of rmse happens to be.

```{r}
summary(rmse_mod1$rmse)

gg<-ggplot(rmse_mod1,aes(rmse))
gg<-gg+geom_density()
gg
```

2. Using a *random partition, create 100 separate cross validations* of your linear model predicting reading scores as a function of at least two covari- ates. Provide a summary table or graphic of the RMSEs from this cross validation.

Hint:
The `crossv_mc` command provides for a generalization of the crossfold command. For this command, we can specify the proportion to be randomly held out in each iteration, via `test=p` where `p` is the proportion to be held out.

```{r}
pd_cv<-pd%>%
  crossv_mc(n=1000,test=.2)
pd_cv
```

The `pd_cv` dataset is a dataset of 1000x2 datasets, with each row containing a training and testing dataset. The testing dataset is .2 of the sample, but it's different each time.

Now we use the same approach, but with the MUCH larger qf_cv dataset. This will take a bit of time.

```{r}
tic()
mod1_rmse_cv<-pd_cv %>%
  mutate(train = map(train, as_tibble)) %>% ## Convert to tibbles
  mutate(model = map(train, ~ lm(mod1_formula, data = .)))%>%
  mutate(rmse = map2_dbl(model, test, rmse))%>%
  select(.id, rmse) ## pull just id and rmse

mod1_rmse_cv
toc()
```

```{r}
summary(mod1_rmse_cv$rmse)

gg<-ggplot(mod1_rmse_cv,aes(rmse))
gg<-gg+geom_density(bins=50,fill="blue",alpha=.2)
gg

```