Modeling_Nick

Nick Orangio

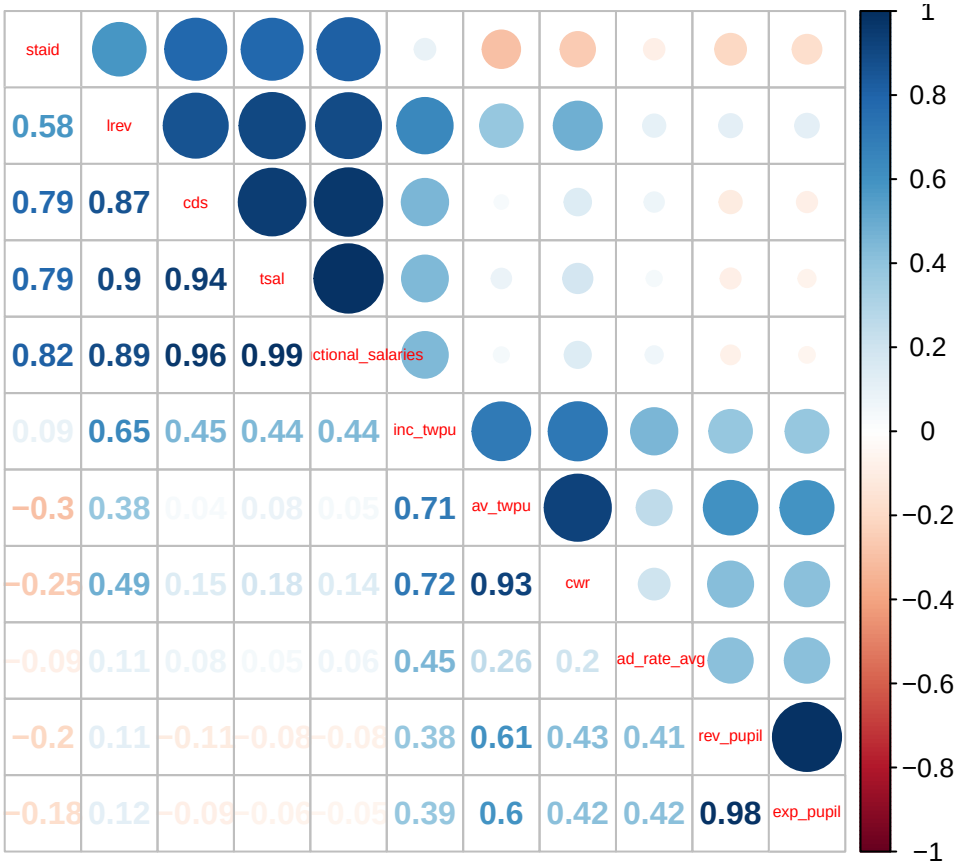
11/9/2020

```
# import county joined and grouped Cornell data for modeling
corn_count_df <- read.csv("/Users/nickorangio/NYC_hs_grad/CSE6242/Data/cornell_modeling_agg.csv")

###
# correlations between factors for use in regression
###
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.2
```

```
## corrplot 0.84 loaded
```

```
# extract columns for correlations
corr_data <- corn_count_df[ , c("lrev", "staid", "tsal", "cds", "rev_pupil", "exp_pupil", "instructional
                                , "av_twpu", "inc_twpu", "cwr", "grad_rate_avg")]

# run correlations
cor_mat <- cor(corr_data, method = "spearman")
corrplot.mixed(cor_mat, tl.cex = 0.5, order = "hclust")
```



Several variables are highly correlated thus should not be included together in a non-regularized multiple linear regression model. As such, the following model includes only a subset of the variables in the correlation matrix above.

```
# regression modeling across all counties
model1 <- lm(grad_rate_avg ~ lrev + staid + rev_pupil + instructional_salaries + cwr, data = corn_count
summary(model1)
```

```
##
## Call:
## lm(formula = grad_rate_avg ~ lrev + staid + rev_pupil + instructional_salaries +
##     cwr, data = corn_count_df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.279768 -0.020666  0.002982  0.025269  0.108745
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)              8.003e-01  8.199e-03  97.602  < 2e-16 ***
## lrev                     1.837e-09  3.176e-10   5.783 1.12e-08 ***
## staid                   -1.074e-09  2.428e-10  -4.425 1.12e-05 ***
## rev_pupil                3.863e-06  4.210e-07   9.176  < 2e-16 ***
## instructional_salaries -1.029e-09  5.107e-10  -2.016   0.0442 *
## cwr                     -4.406e-02  5.862e-03  -7.517 1.78e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.03874 on 678 degrees of freedom
## Multiple R-squared:  0.3198, Adjusted R-squared:  0.3147
## F-statistic: 63.74 on 5 and 678 DF,  p-value: < 2.2e-16
```

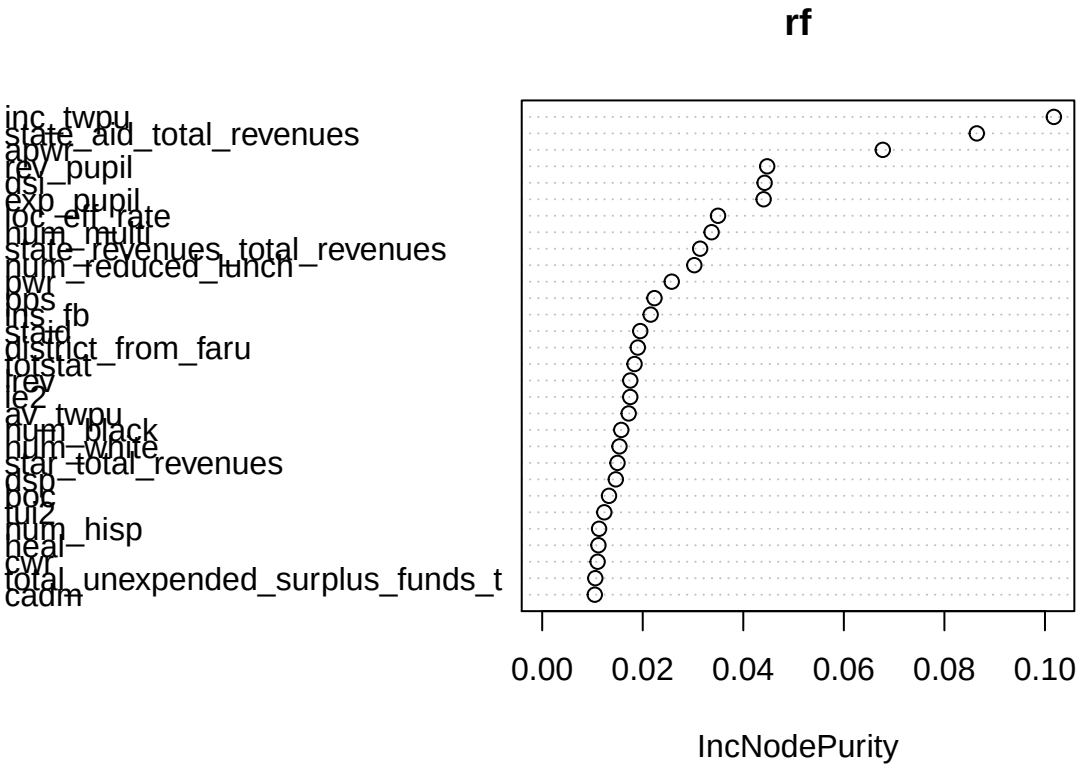
```
###
# random forest model with all variables
###

library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.2
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
# drop columns not used in randomForest modeling -- only select column 20 and onwards
rf_data <- corn_count_df[ , c(20:96) ]
rf_data <- rf_data[, -c(71:73)]
rf_data <- rf_data[, -c(73)]
rf_data <- rf_data[, -c(71:72)]
rf_data <- rf_data[, -c(6:7)]
rf_data <- rf_data[, -c(10:12)]

rf_data <- na.omit(rf_data)

# build train and test dataset
set.seed(1337)
sample_size = floor(0.8 * nrow(rf_data))
train_ind = sample(seq_len(nrow(rf_data)), size = sample_size)

forest_train = rf_data[train_ind, ]
forest_test = rf_data[-train_ind, ]

# build randomForest
rf <- randomForest(grad_rate_avg ~ ., data = forest_train)

varImpPlot(rf, type = 2)
```




```
rf
```

```
##
## Call:
##  randomForest(formula = grad_rate_avg ~ ., data = forest_train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 21
##
##           Mean of squared residuals: 0.0008345512
##                     % Var explained: 60
```

```
# extract data for specific counties
albany_data <- corn_count_df[corn_count_df$county_name == 'Albany', ]
```