# Student-Loan-Data

*Camellia Debnath*

*11/28/2018*

## 1. Combining the data for 2008-2013

```
col_09_10 <- read.csv("MERGED2009_10_PP.csv")
college_09_10 <- col_09_10 %>%
  mutate(GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         "Year" = "2009-10") %>%
   mutate(DEBT_TO_EARN = GRAD_DEBT_MDN/MD_EARN_WNE_P8)

col_11_12 <- read.csv("MERGED2011_12_PP.csv")
college_11_12 <- col_11_12 %>%
  mutate(GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         "Year" = "2011-12") %>%
   mutate(DEBT_TO_EARN = GRAD_DEBT_MDN/MD_EARN_WNE_P8)

col_12_13 <- read.csv("MERGED2012_13_PP.csv")
college_12_13 <- col_12_13 %>%
  mutate(GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         "Year" = "2012-13") %>%
   mutate(DEBT_TO_EARN = GRAD_DEBT_MDN/MD_EARN_WNE_P8)

col_13_14 <- read.csv("MERGED2013_14_PP.csv")
college_13_14 <- col_13_14 %>%
  mutate(GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         "Year" = "2013-14") %>%
   mutate(DEBT_TO_EARN = GRAD_DEBT_MDN/MD_EARN_WNE_P8)

col_14_15 <- read.csv("MERGED2014_15_PP.csv")
college_14_15 <- col_14_15 %>%
  mutate(GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         "Year" = "2014-15") %>%
   mutate(DEBT_TO_EARN = GRAD_DEBT_MDN/MD_EARN_WNE_P8)


college_09_15 <- rbind(college_09_10, college_11_12, college_12_13, college_13_14, college_14_15)
```

## 2. Partitioning the data into training and test sets

```
college_09_15_parts <- resample_partition(college_09_15 ,c(train = 0.6, valid = 0.2, test = 0.2))
```

```
college_09_15_train_ <- as_tibble(college_09_15_parts$train)
college_09_15_test_ <- as_tibble(college_09_15_parts$test)
college_09_15_valid_ <- as_tibble(college_09_15_parts$valid)
```

## 3. Subsetting variables to check for potential predictors

Intuitively, we select a subset of variables, and tidy the data for further EDA.

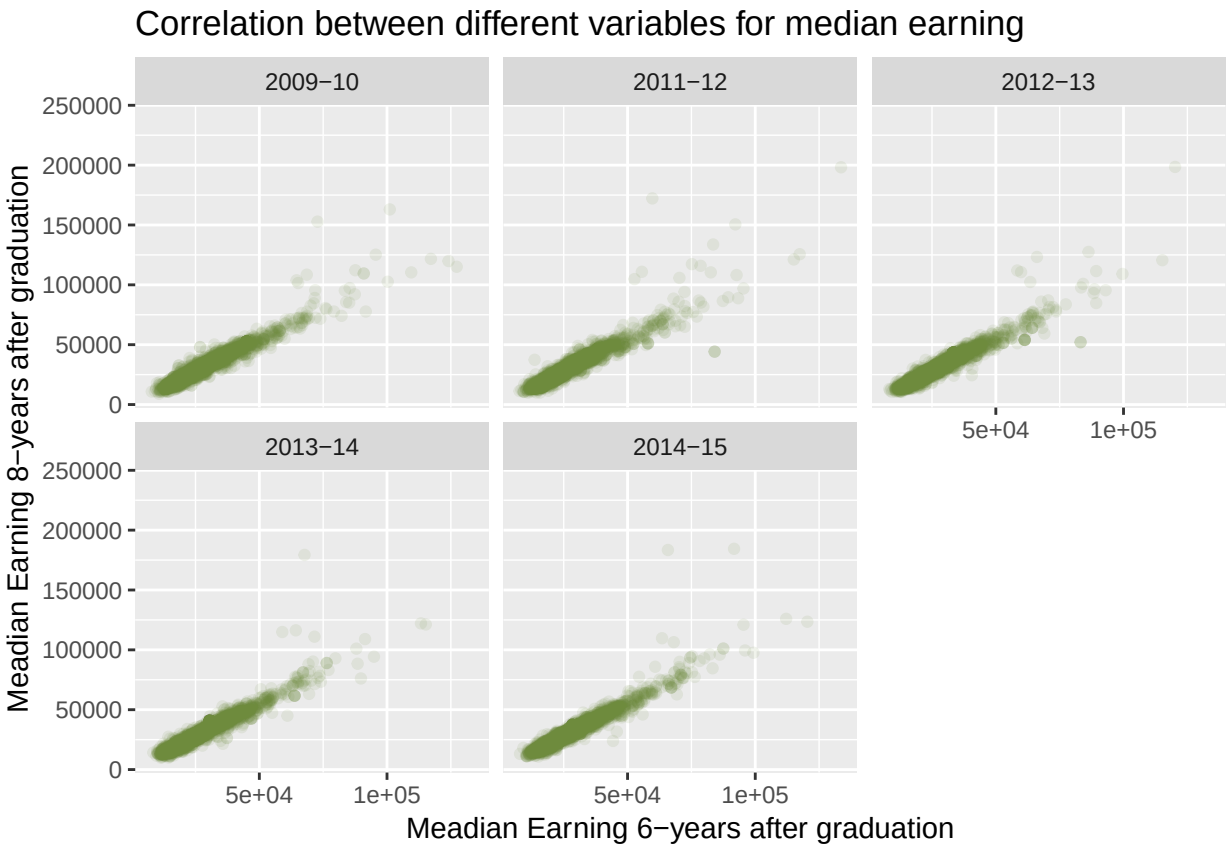
```
college_09_15_train <- college_09_15_train_ %>%
  select(COMPL_RPY_3YR_RT, GRAD_DEBT_MDN,PCTFLOAN, PCTPELL, MD_EARN_WNE_P6, MD_EARN_WNE_P8, MD_EARN_WNE_
         CDR3, MEDIAN_HH_INC, AGE_ENTRY, UGDS, CONTROL, COSTT4_A, COSTT4_P, Year, DEBT_TO_EARN, MD_FAMI
  mutate(COMPL_RPY_3YR_RT = as.numeric(as.character(COMPL_RPY_3YR_RT)),
         GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P6 = as.numeric(as.character(MD_EARN_WNE_P6)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         MD_EARN_WNE_P10 = as.numeric(as.character(MD_EARN_WNE_P10)),
         CDR3 = as.numeric(as.character(CDR3)), # three-year cohort default rate
         MEDIAN_HH_INC = as.numeric(as.character(MEDIAN_HH_INC)), #Median household income
         AGE_ENTRY = as.numeric(as.character(AGE_ENTRY)),
         UGDS = as.numeric(as.character(UGDS)),
         CONTROL =as.character(CONTROL),
         COSTT4_A = as.numeric(as.character(COSTT4_A)), #avg cost of attendance for academic year insti
         COSTT4_P = as.numeric(as.character(COSTT4_P)), #avg cost of attendance for program year instit
         PCTFLOAN = as.numeric(as.character(PCTFLOAN)),
         PCTPELL = as.numeric(as.character(PCTPELL)),
         MD_FAMINC = as.numeric(as.character(MD_FAMINC)))

college_09_15_test <- college_09_15_test_ %>%
  select(COMPL_RPY_3YR_RT, GRAD_DEBT_MDN,PCTFLOAN, PCTPELL, MD_EARN_WNE_P6, MD_EARN_WNE_P8, MD_EARN_WNE_
         CDR3, MEDIAN_HH_INC, AGE_ENTRY, UGDS, CONTROL, COSTT4_A, COSTT4_P, Year, DEBT_TO_EARN, MD_FAMI
  mutate(COMPL_RPY_3YR_RT = as.numeric(as.character(COMPL_RPY_3YR_RT)),
         GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P6 = as.numeric(as.character(MD_EARN_WNE_P6)),
         MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
         MD_EARN_WNE_P10 = as.numeric(as.character(MD_EARN_WNE_P10)),
         CDR3 = as.numeric(as.character(CDR3)), # three-year cohort default rate
         MEDIAN_HH_INC = as.numeric(as.character(MEDIAN_HH_INC)), #Median household income
         AGE_ENTRY = as.numeric(as.character(AGE_ENTRY)),
         UGDS = as.numeric(as.character(UGDS)),
         CONTROL =as.character(CONTROL),
         COSTT4_A = as.numeric(as.character(COSTT4_A)), #avg cost of attendance for academic year insti
         COSTT4_P = as.numeric(as.character(COSTT4_P)), #avg cost of attendance for program year instit
         PCTFLOAN = as.numeric(as.character(PCTFLOAN)),
         PCTPELL = as.numeric(as.character(PCTPELL)),
         MD_FAMINC = as.numeric(as.character(MD_FAMINC)))

college_09_15_valid <- college_09_15_valid_ %>%
  select(COMPL_RPY_3YR_RT, GRAD_DEBT_MDN,PCTFLOAN, PCTPELL, MD_EARN_WNE_P6, MD_EARN_WNE_P8, MD_EARN_WNE_
         CDR3, MEDIAN_HH_INC, AGE_ENTRY, UGDS, CONTROL, COSTT4_A, COSTT4_P, Year, DEBT_TO_EARN, MD_FAMI
  mutate(COMPL_RPY_3YR_RT = as.numeric(as.character(COMPL_RPY_3YR_RT)),
         GRAD_DEBT_MDN = as.numeric(as.character(GRAD_DEBT_MDN)),
         MD_EARN_WNE_P6 = as.numeric(as.character(MD_EARN_WNE_P6)),
```

```
        MD_EARN_WNE_P8 = as.numeric(as.character(MD_EARN_WNE_P8)),
        MD_EARN_WNE_P10 = as.numeric(as.character(MD_EARN_WNE_P10)),
        CDR3 = as.numeric(as.character(CDR3)), # three-year cohort default rate
        MEDIAN_HH_INC = as.numeric(as.character(MEDIAN_HH_INC)), #Median household income
        AGE_ENTRY = as.numeric(as.character(AGE_ENTRY)),
        UGDS = as.numeric(as.character(UGDS)),
        CONTROL =as.character(CONTROL),
        COSTT4_A = as.numeric(as.character(COSTT4_A)), #avg cost of attendance for academic year insti
        COSTT4_P = as.numeric(as.character(COSTT4_P)), #avg cost of attendance for program year instit
        PCTFLOAN = as.numeric(as.character(PCTFLOAN)),
        PCTPELL = as.numeric(as.character(PCTPELL)),
        MD_FAMINC = as.numeric(as.character(MD_FAMINC)))
```

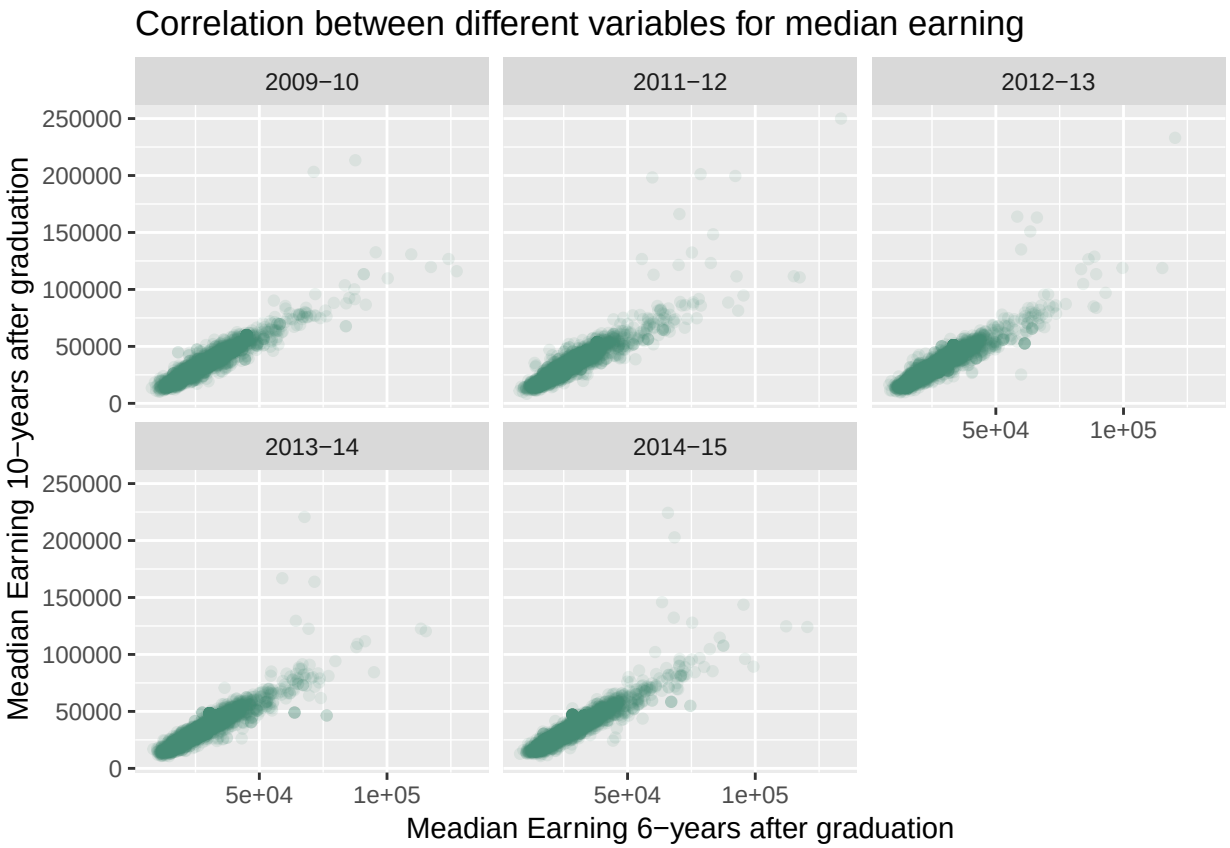
## 4. EDA for deciding predictors for prediction of response variable: COMPL_RPY_3YR_RT

We have multiple variables related to earnings, such as MD_EARN_WNE_P6, MD_EARN_WNE_P8 and MD_EARN_WNE_P10. We can see if there's a strong correlation between these three, if there is, then we can use only one of them in our modelling.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = MD_EARN_WNE_P6, y = MD_EARN_WNE_P8), color = "darkolivegreen4", alpha = 0.1) +
  facet_wrap(~Year)+
  labs(title = "Correlation between different variables for median earning",
       x = "Meadian Earning 6-years after graduation",
       y = "Meadian Earning 8-years after graduation")
```

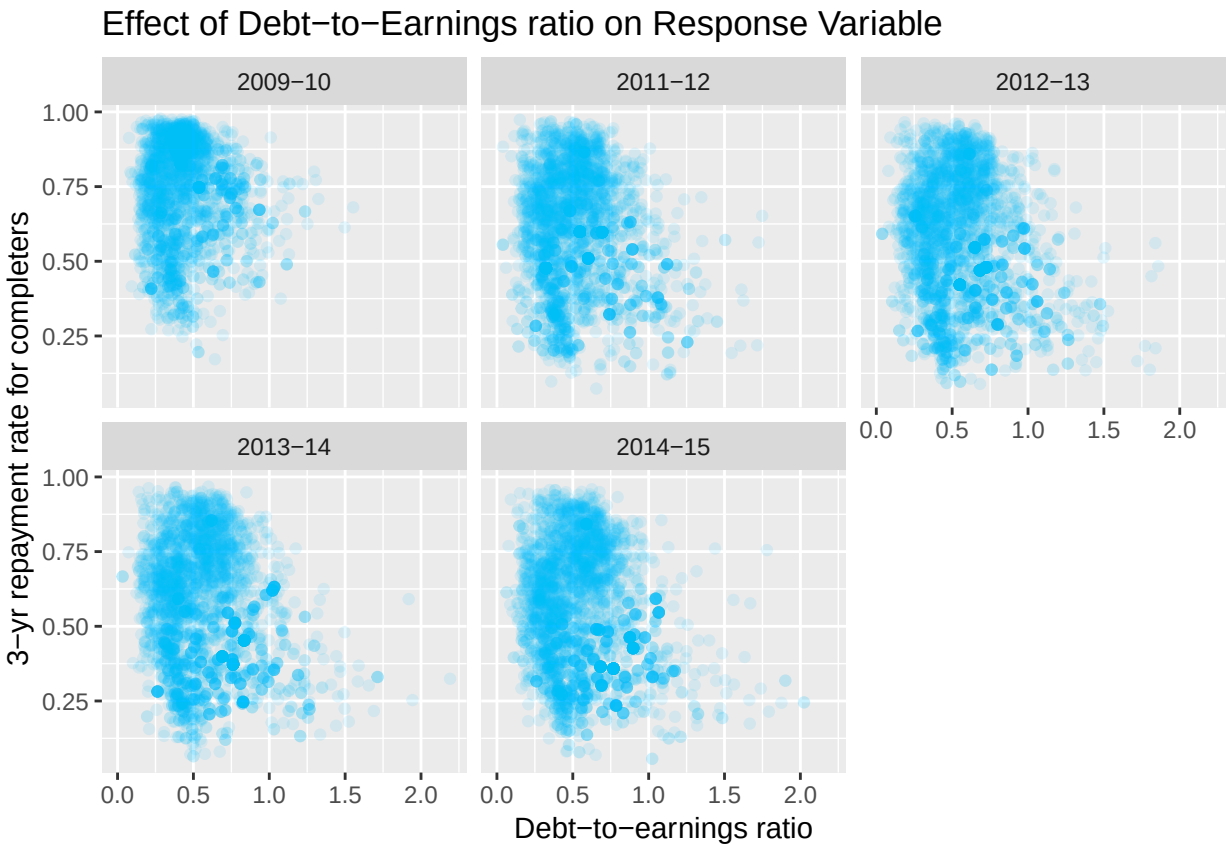



```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = MD_EARN_WNE_P6, y = MD_EARN_WNE_P10), color = "aquamarine4", alpha = 0.1) +
  facet_wrap(~Year) +
  labs(title = "Correlation between different variables for median earning",
       x = "Meadian Earning 6-years after graduation",
       y = "Meadian Earning 10-years after graduation")
```



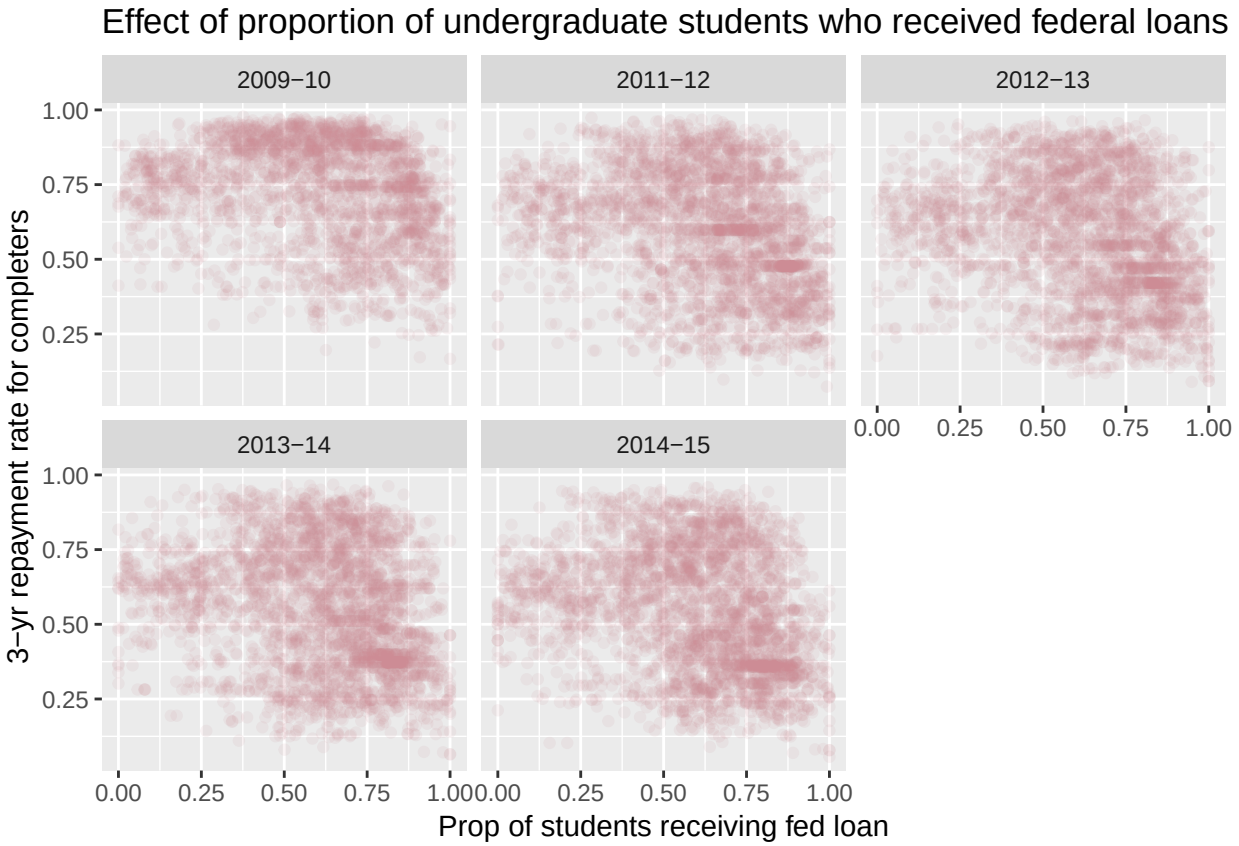

Since we see a high linear correlation between these three, we can arbitrarily decide to keep MD_EARN_WNE_P8 for our predictor model.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = DEBT_TO_EARN, y = COMPL_RPY_3YR_RT), color = "deepskyblue", alpha = 0.1) +
  facet_wrap(~Year)+
  labs(title = "Effect of Debt-to-Earnings ratio on Response Variable",
       x = "Debt-to-earnings ratio",
       y = "3-yr repayment rate for completers")
```



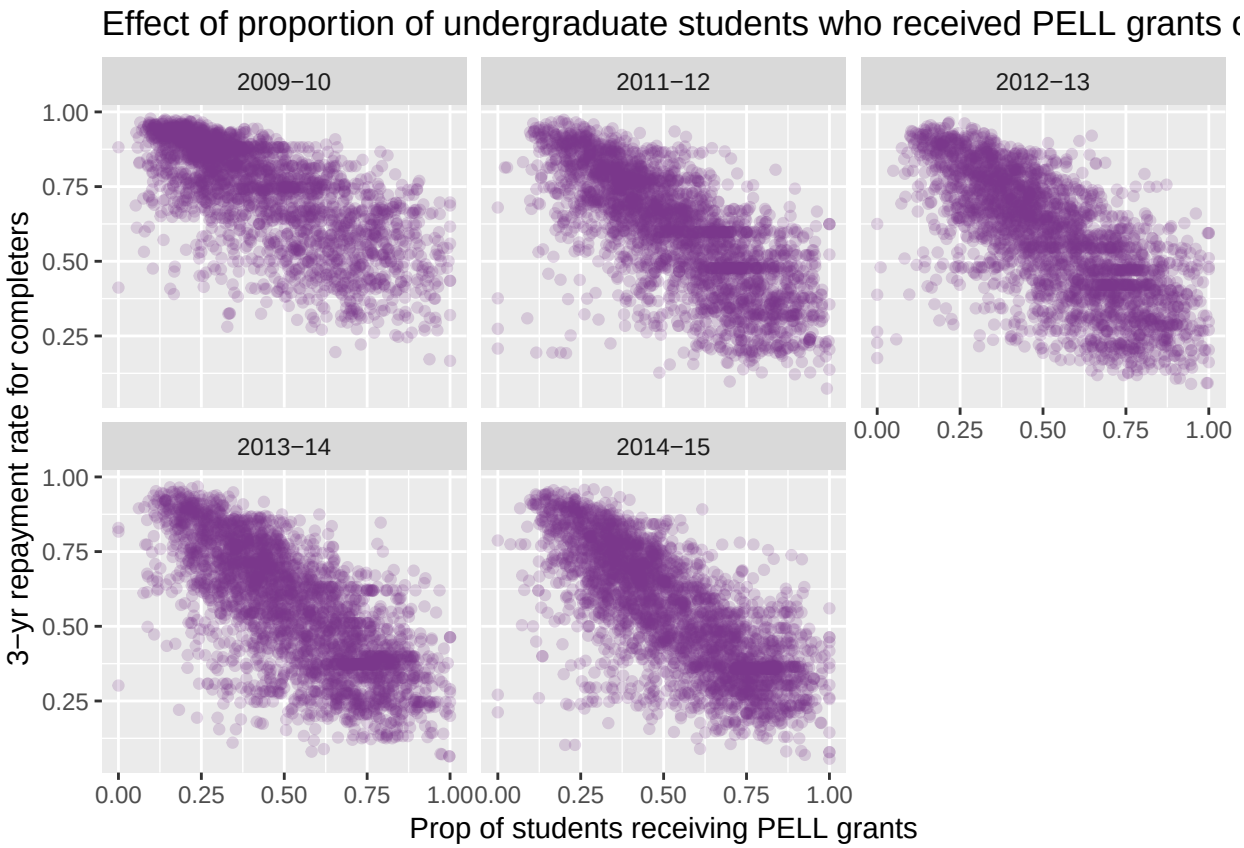

We observe some sort of negative correlation, we can keep DEBT_TO_EARN ration for our prediction.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = PCTFLOAN, y = COMPL_RPY_3YR_RT), color = "lightpink3", alpha = 0.1) +
  facet_wrap(~Year)+
  labs(title = "Effect of proportion of undergraduate students who received federal loans on Response Va
       x = "Prop of students receiving fed loan",
       y = "3-yr repayment rate for completers")
```



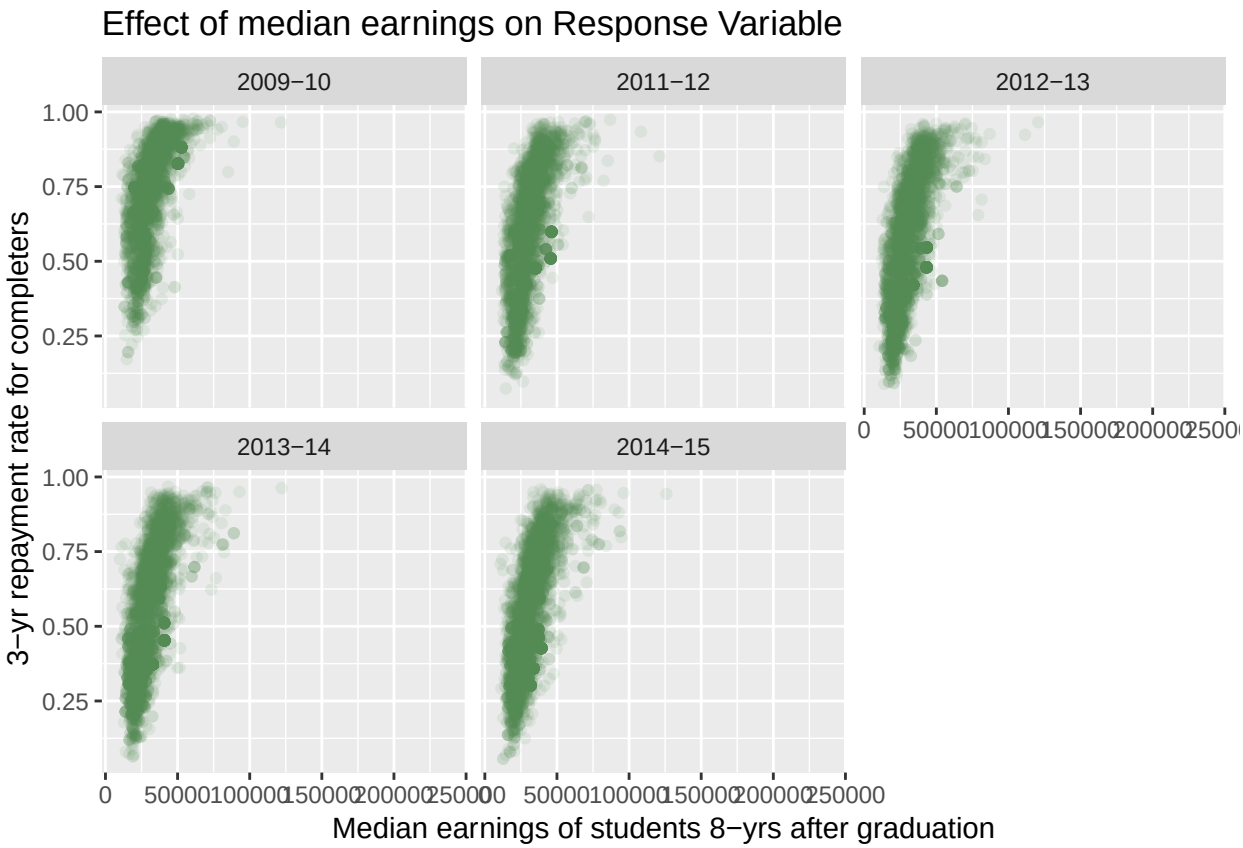

Mostly random, disregard.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = PCTPELL, y = COMPL_RPY_3YR_RT), color = "mediumorchid4", alpha = 0.2) +
  facet_wrap(~Year)+
  labs(title = "Effect of proportion of undergraduate students who received PELL grants on Response Vari
       x = "Prop of students receiving PELL grants",
       y = "3-yr repayment rate for completers")
```



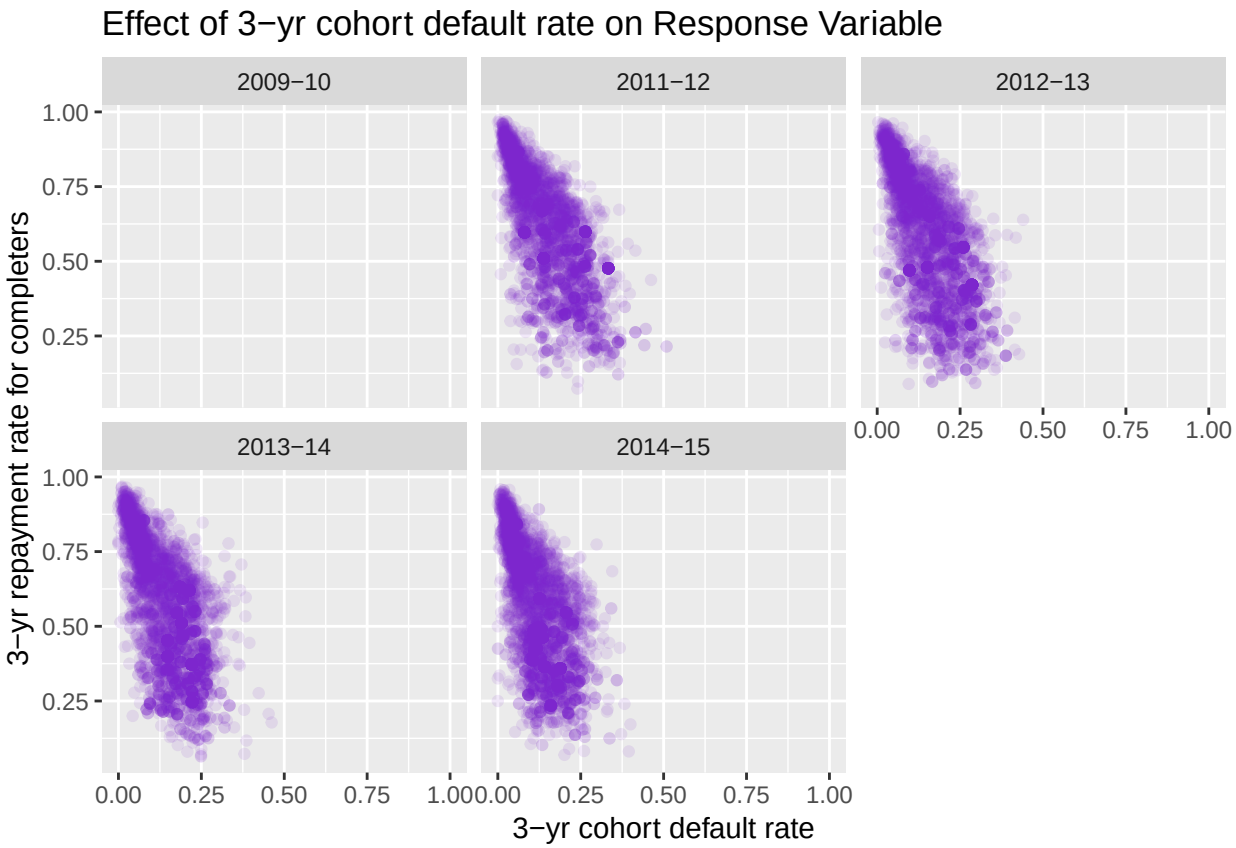

Strong negative correlation, keep.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = MD_EARN_WNE_P8, y = COMPL_RPY_3YR_RT), color = "palegreen4", alpha = 0.1) +
  facet_wrap(~Year)+
  labs(title = "Effect of median earnings on Response Variable",
       x = "Median earnings of students 8-yrs after graduation",
       y = "3-yr repayment rate for completers")
```



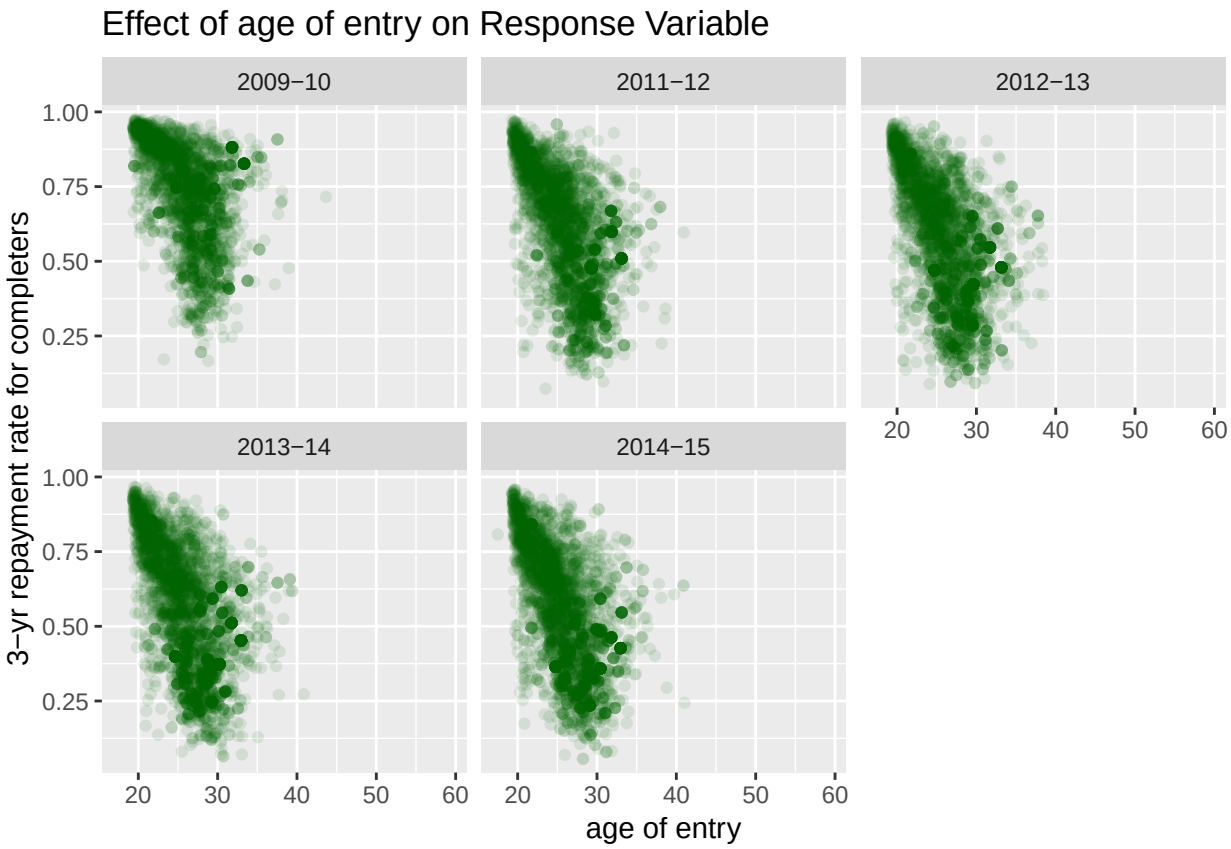

Keep MD_EARN_WNE_P8.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = CDR3, y = COMPL_RPY_3YR_RT), color = "purple3", alpha = 0.1) +
  facet_wrap(~Year)+
  labs(title = "Effect of 3-yr cohort default rate on Response Variable",
       x = "3-yr cohort default rate",
       y = "3-yr repayment rate for completers")
```



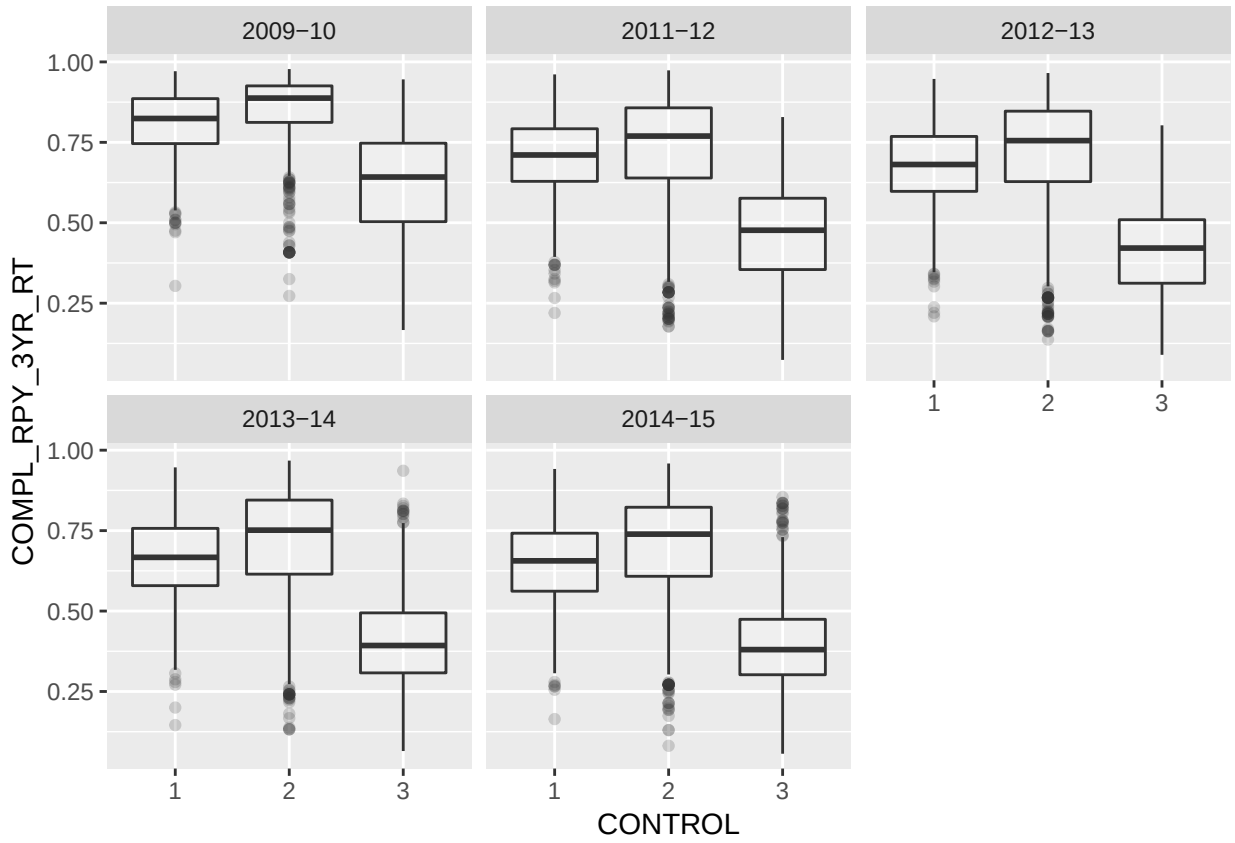

Keep CDR3.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = AGE_ENTRY, y = COMPL_RPY_3YR_RT), color = "darkgreen", alpha = 0.1) +
  facet_wrap(~Year)+
  labs(title = "Effect of age of entry on Response Variable",
       x = "age of entry",
       y = "3-yr repayment rate for completers")
```



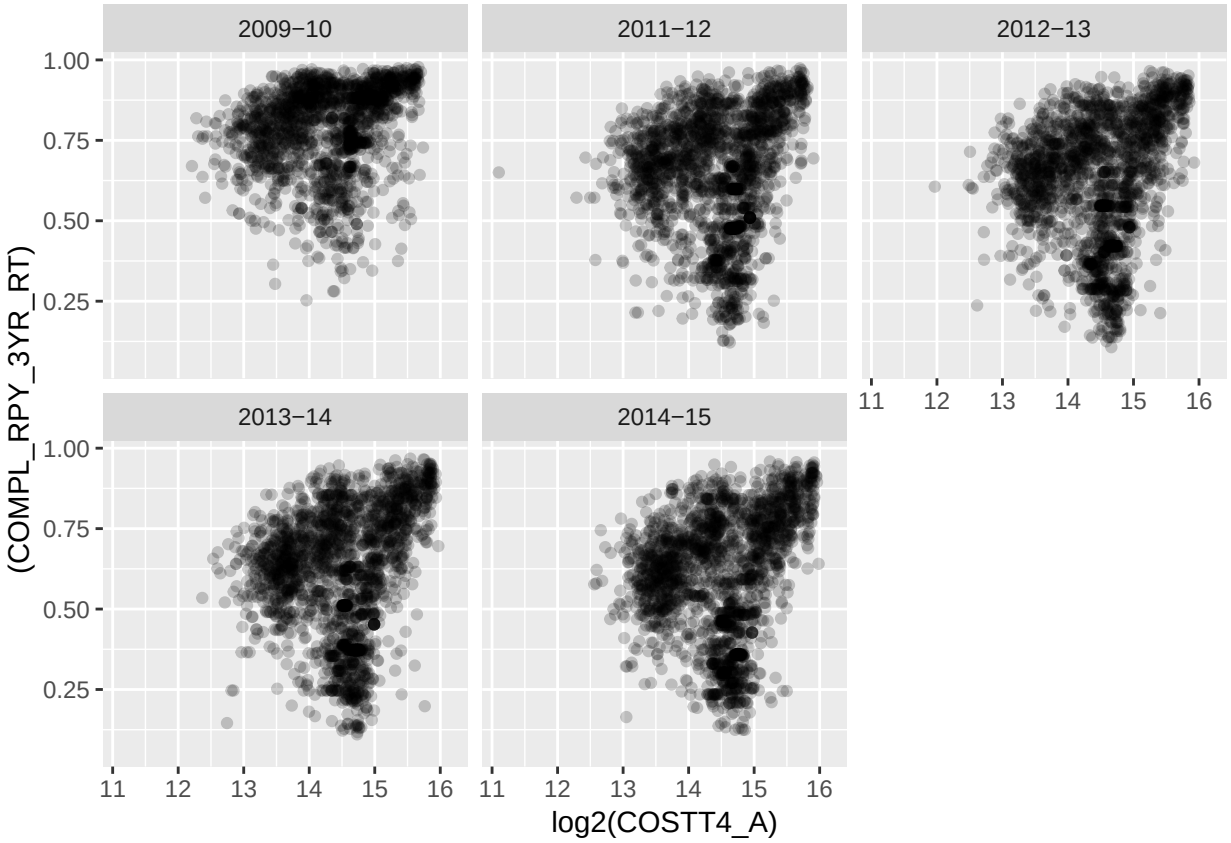

Keep AGE_ENTRY.

```
college_09_15_train %>%
  ggplot() +
  geom_boxplot(aes(x = CONTROL, y = COMPL_RPY_3YR_RT), alpha = 0.2) +
  facet_wrap(~Year)
```

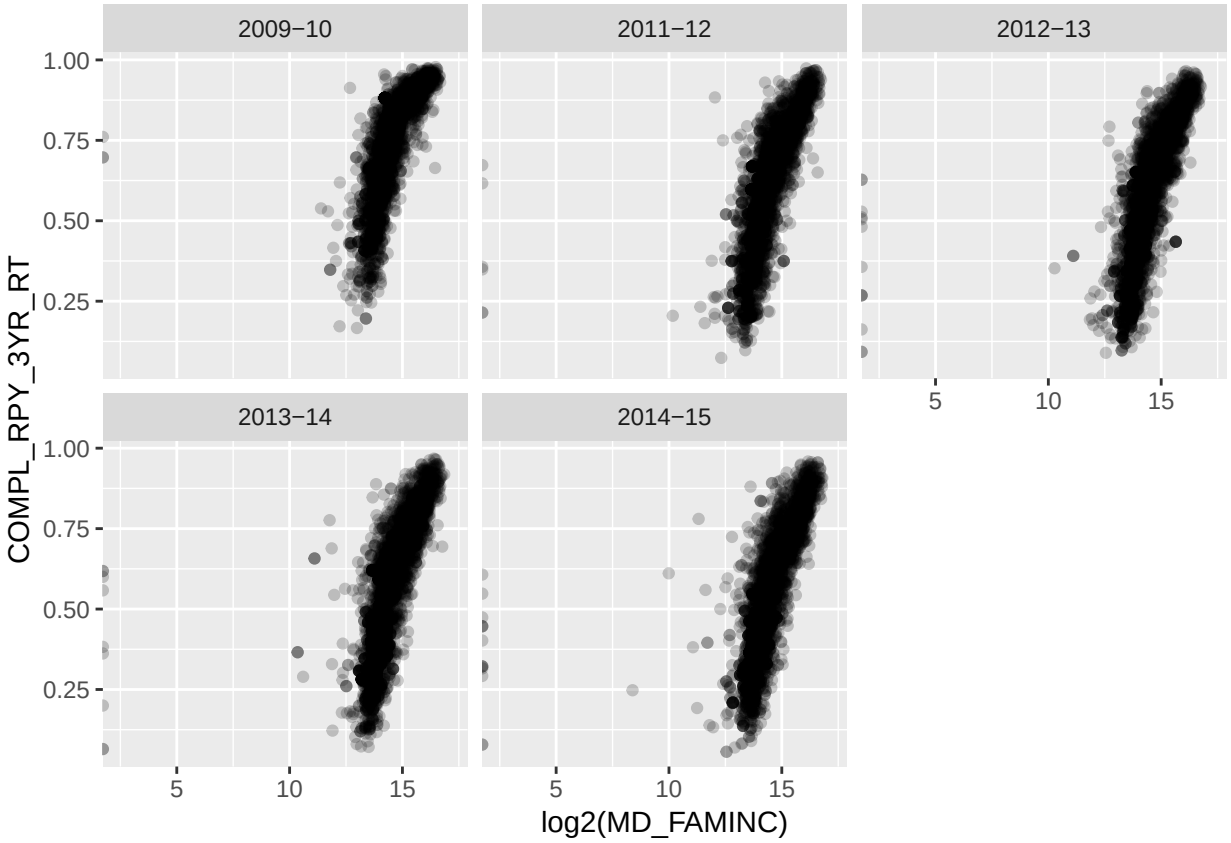



wide fluctuation among control groups, so keeping CONTROL as one of the predictors.

```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = log2(COSTT4_A), y = (COMPL_RPY_3YR_RT)), alpha = 0.2) +
  facet_wrap(~Year)
```




```
college_09_15_train %>%
  ggplot() +
  geom_point(aes(x = log2(MD_FAMINC), y = COMPL_RPY_3YR_RT), alpha = 0.2) +
  facet_wrap(~Year)
```




Keeping MD_FAMINC.

**5. Now we can try fitting a linear model with the above predictor variables.**

```
set.seed(1)

college_09_15_train <- college_09_15_train %>%
  filter(Year != "2009-10") %>%
  mutate(log_MD_FAMINC = log2(MD_FAMINC)) %>%
  filter(log_MD_FAMINC >= 0)

college_09_15_test <- college_09_15_test %>%
  filter(Year != "2009-10") %>%
  mutate(log_MD_FAMINC = log2(MD_FAMINC)) %>%
  filter(log_MD_FAMINC >= 0)

college_09_15_valid <- college_09_15_valid %>%
  filter(Year != "2009-10") %>%
  mutate(log_MD_FAMINC = log2(MD_FAMINC)) %>%
  filter(log_MD_FAMINC >= 0)


# this model was decided after trying various combinations of predictors, checking their R-squared valu
# the analysis of this can be found later in this document in the 6th section titled: "R-Square analysi
model <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN + PCTPELL  + MD_EARN_WNE_P8 + log2(COSTT4_A) + log_MD_FAMINC
```

```
              data = college_09_15_train)

summary(model)
```

```
##
## Call:
## lm(formula = COMPL_RPY_3YR_RT ~ DEBT_TO_EARN + PCTPELL + MD_EARN_WNE_P8 +
##     log2(COSTT4_A) + log_MD_FAMINC, data = college_09_15_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41054 -0.05527  0.00068  0.05651  0.69726
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -8.098e-01  3.379e-02  -23.96   <2e-16 ***
## DEBT_TO_EARN   -7.650e-02  5.084e-03  -15.05   <2e-16 ***
## PCTPELL        -2.027e-01  8.166e-03  -24.82   <2e-16 ***
## MD_EARN_WNE_P8  2.531e-06  1.582e-07   16.00   <2e-16 ***
## log2(COSTT4_A) -3.373e-02  2.067e-03  -16.32   <2e-16 ***
## log_MD_FAMINC   1.346e-01  2.084e-03   64.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09218 on 7921 degrees of freedom
##   (9631 observations deleted due to missingness)
## Multiple R-squared:  0.7682, Adjusted R-squared:  0.7681
## F-statistic:  5251 on 5 and 7921 DF,  p-value: < 2.2e-16
```

```
print("RMSE for train data")
```

```
## [1] "RMSE for train data"
```

```
rmse(model, college_09_15_train) #0.092267
```

```
## [1] 0.09214048
```

```
print("RMSE for test data")
```

```
## [1] "RMSE for test data"
```

```
rmse(model, college_09_15_test)  #0.09390385
```

```
## [1] 0.0932137
```

```
print("Mean Absolute Error for Test Data" )
```

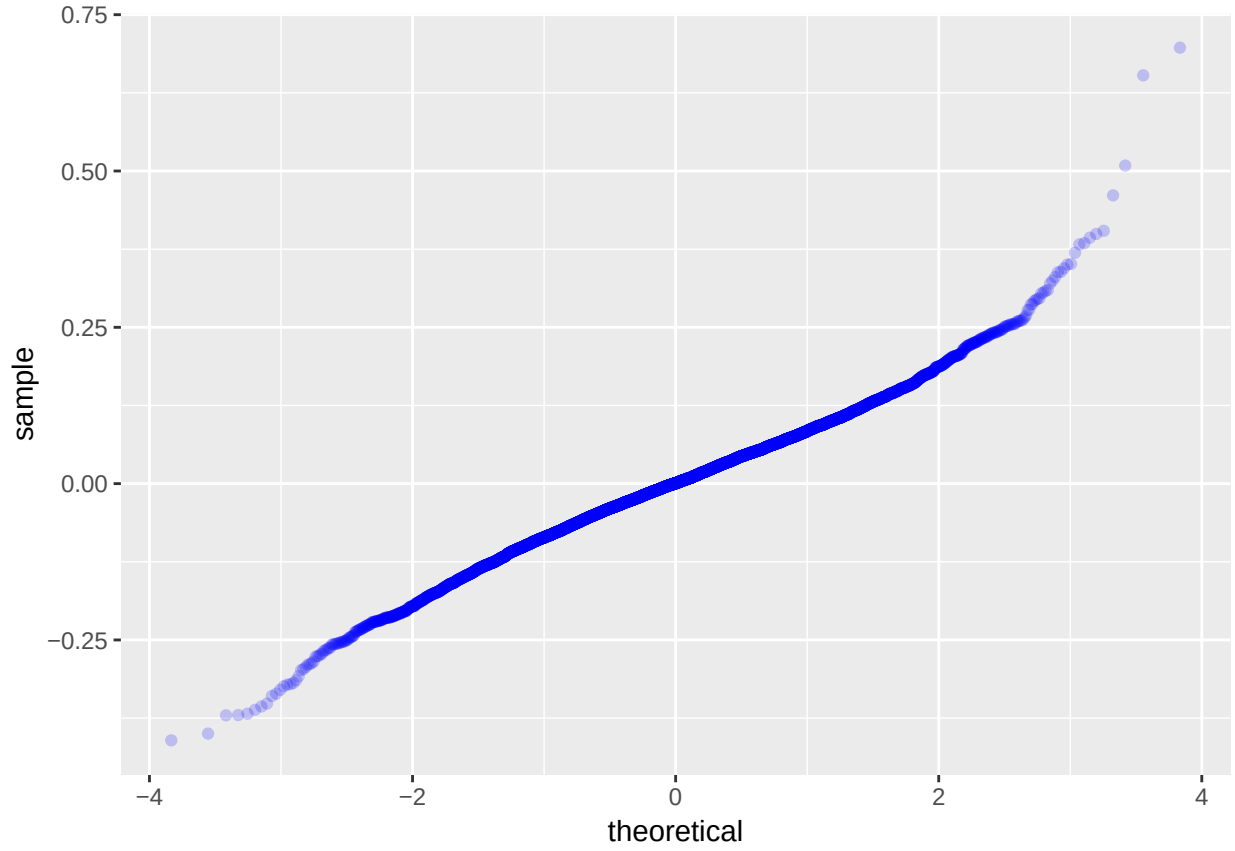
```
## [1] "Mean Absolute Error for Test Data"
```

```
mae(model, college_09_15_test)
```

```
## [1] 0.07194262
```

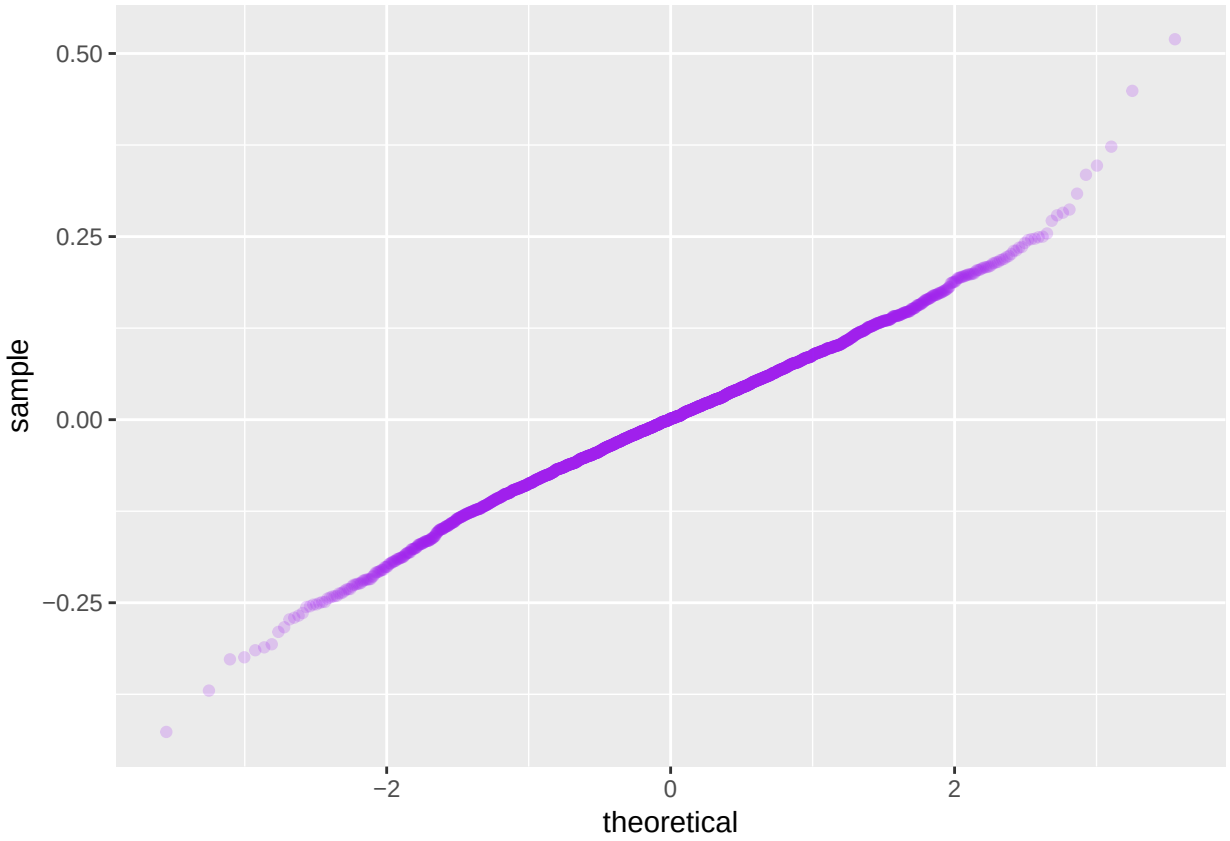
Plotting the residuals for the training data:

```
college_09_15_train %>%
  add_residuals(model) %>%
  ggplot(aes(sample=resid)) +
  geom_qq(color = "blue", alpha = 0.2)
```




Plotting the residuals for the test data:

```
college_09_15_test %>%
  add_residuals(model) %>%
  ggplot(aes(sample=resid)) +
  geom_qq(color = "purple", alpha = 0.2)
```




### 6.R-Square analysis for overfitting tests.

```
#' @description Returns lm model fit statistics R-squared, adjucted R-squared,
#'      predicted R-squared and PRESS.
#'      Thanks to John Mount for his 6-June-2014 blog post, R style tip: prefer functions that return d
#'      the idea \link{http://www.win-vector.com/blog/2014/06/r-style-tip-prefer-functions-that-return-
#' @return Returns a data frame with one row and a column for each statistic
#' @param linear.model A \code{lm()} model.
model_fit_stats <- function(linear.model) {
  r.sqr <- summary(linear.model)$r.squared
  adj.r.sqr <- summary(linear.model)$adj.r.squared
  pre.r.sqr <- pred_r_squared(linear.model)
  PRESS <- PRESS(linear.model)
  return.df <- data.frame(r.squared = r.sqr, adj.r.squared = adj.r.sqr, pred.r.squared = pre.r.sqr, pres
  return(return.df)
}
```

```
#' @description returns the predictive r-squared. Requires the function PRESS(), which returns
#'              the PRESS statistic.
#' @param linear.model A linear regression model (class 'lm'). Required.
#'
pred_r_squared <- function(linear.model) {
  #' Use anova() to get the sum of squares for the linear model
  lm.anova <- anova(linear.model)
```

```
  #' Calculate the total sum of squares
  tss <- sum(lm.anova$'Sum Sq')
  # Calculate the predictive R^2
  pred.r.squared <- 1-PRESS(linear.model)/(tss)

  return(pred.r.squared)
}
```

```
#' @description Returns the PRESS statistic (predictive residual sum of squares).
#'              Useful for evaluating predictive power of regression models.
#' @param linear.model A linear regression model (class 'lm'). Required.
#'
PRESS <- function(linear.model) {
  #' calculate the predictive residuals
  pr <- residuals(linear.model)/(1-lm.influence(linear.model)$hat)
  #' calculate the PRESS
  PRESS <- sum(pr^2)

  return(PRESS)
}
```

```
model_1 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN,
              data = college_09_15_train)
rmse(model_1, college_09_15_valid) #0.1986109
```

```
## [1] 0.2007086
```

```
model_2 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN, + MD_EARN_WNE_P8,
              data = college_09_15_train)
rmse(model_2, college_09_15_valid) #0.199981
```

```
## [1] 0.2010461
```

```
model_3 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN, + MD_EARN_WNE_P8 + PCTPELL,
              data = college_09_15_train)
rmse(model_3, college_09_15_valid) #0.1997196
```

```
## [1] 0.2014494
```

```
model_4 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN + PCTPELL  + MD_EARN_WNE_P8 + log2(COSTT4_A) + log_MD_FAM
              data = college_09_15_train)
rmse(model_4, college_09_15_valid) #0.0921737
```

```
## [1] 0.09328924
```

```
model_5 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN, + MD_EARN_WNE_P8 +PCTPELL + CDR3 + log2(COSTT4_A),
              data = college_09_15_train) #0.2250866
rmse(model_5, college_09_15_valid)
```

```
## [1] 0.2154019
```

```
model_6 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN, + MD_EARN_WNE_P8 +PCTPELL + CDR3 + log2(COSTT4_A) + log_
              data = college_09_15_train)
rmse(model_6, college_09_15_valid) #0.2343732
```

```
## [1] 0.2078171
```

```
model_7 <- lm(COMPL_RPY_3YR_RT ~ DEBT_TO_EARN, + MD_EARN_WNE_P8 +PCTPELL + CDR3 + log2(COSTT4_A) + log_
              data = college_09_15_train)
rmse(model_7, college_09_15_valid) #0.2104708
```

```
## [1] 0.2258148
```

```
ldply(list(model_1, model_2, model_3, model_4, model_5, model_6, model_7), model_fit_stats)
```

```
##     r.squared adj.r.squared pred.r.squared      press
## 1 0.033949306   0.033867830     0.03366424 469.305009
## 2 0.047882901   0.046874302     0.04460748  29.910666
## 3 0.052887370   0.051771807     0.04929148  27.069816
## 4 0.768238881   0.768092586     0.76774659  67.442134
## 5 0.004304082  -0.005362869    -0.02819778   3.925688
## 6 0.137466911   0.128091552     0.10426754   2.941548
## 7 0.023328357   0.013753145    -0.01265230   3.108386
```

```
rmse(model_4, college_09_15_test)
```

```
## [1] 0.0932137
```