DaigleEconRProject.R

2011home

Mon Apr 23 11:56:41 2018

# Chris Daigle  
# Econ5495 - R Programming  
# Class Project  
#  
rm(list = ls())  
  
# Employing packages ####  
#  
library('dplyr')

library('tidyr')

library('car')

library('data.table')

library('plm')

# Importing ####  
#  
setwd(  
 '/Users/2011home/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/R/Project'  
)  
  
IDSAT <-  
 fread('ISAT District Master 23Mar2016.csv', stringsAsFactors = FALSE  
 )  
  
# Renaming ####  
#  
names(IDSAT)[names(IDSAT) == 'Dist #'] <- 'Dist'  
names(IDSAT)[names(IDSAT) == 'Admin ID'] <- 'AdID'  
names(IDSAT)[names(IDSAT) == 'Year'] <- 'Yr'  
names(IDSAT)[names(IDSAT) == 'grade'] <- 'Grade'  
names(IDSAT)[names(IDSAT) == 'Annual Expenditure'] <- 'AnnExp'  
names(IDSAT)[names(IDSAT) == 'Taxes'] <- 'Tax'  
names(IDSAT)[names(IDSAT) == 'Other Local Sources'] <- 'OthLcl'  
names(IDSAT)[names(IDSAT) == 'State Sources'] <- 'State'  
names(IDSAT)[names(IDSAT) == 'Federal Sources'] <- 'Fed'  
names(IDSAT)[names(IDSAT) == 'Other Sources'] <- 'OthLcl'  
names(IDSAT)[names(IDSAT) == 'Membership'] <- 'Mem'  
names(IDSAT)[names(IDSAT) == 'TotalPerPupilExpenditure'] <- 'TotPPE'  
names(IDSAT)[names(IDSAT) == '4DAY WEEKS'] <- 'Day'  
names(IDSAT)[names(IDSAT) == 'FRLNumber'] <- 'FRL'  
names(IDSAT)[names(IDSAT) == 'EstimatedTotalPopulation(OfTheSchoolDistrict)'] <-  
 'PopEstDist'  
names(IDSAT)[names(IDSAT) == 'EstimatedPopulation517(ChildrenInSchool)'] <-  
 'Pop517EstDist'  
names(IDSAT)[names(IDSAT) == 'PovEstimatedNumberOfRelevantChildren5To17YearsOldInPovertyWhoAreRelatedToTheHouseholder(NumberOfChildrenInPoverty)'] <-  
 'Pov'  
names(IDSAT)[names(IDSAT) == 'FY Inflation (Index)'] <- 'InfInd13'  
names(IDSAT)[names(IDSAT) == 'Annual Expenditure Adjusted for Inflation (2013)'] <-  
 'AnnInfExp'  
names(IDSAT)[names(IDSAT) == 'Total PPE Adjusted for Inflation'] <-  
 'InfPPE'  
IDSAT <- as.data.frame(IDSAT)

#   
# Simple function to replace the spaces data.table introduced to the variable names to keep in line with the data.frame/base-R syntax  
  
spaceless <- function(x) {  
 colnames(x) <- gsub(" ", ".", colnames(x))  
 x  
}  
IDSAT <- spaceless(IDSAT)  
  
# Assign proper classes to variables ####  
#  
for (i in c(1:2)) {  
 IDSAT[, i] <- as.character(IDSAT[, i])  
}  
  
IDSAT$Dist <- factor(IDSAT$Dist, ordered = FALSE)  
IDSAT$Yr <- as.Date(paste0(IDSAT$Yr, '-01-01'))  
IDSAT$Grade <- as.numeric(as.character(IDSAT$Grade))  
# IDSAT <- IDSAT[order(IDSAT$Grade), ]  
# IDSAT$Grade <- factor(IDSAT$Grade)  
  
for (i in c(97:112)) {  
 IDSAT[, i] <- as.numeric(as.character(IDSAT[, i]))  
}

# Make dummies ####  
#  
IDSAT$int0 <- 0  
for (i in c(3:9)) {  
 for (j in unique(IDSAT$Yr)){  
 IDSAT$int0[IDSAT$Grade == i & IDSAT$Yr == j] <- 1  
 }  
}  
#  
IDSAT$int1 <- 0  
for (i in c(4:10)) {  
 for (j in unique(IDSAT$Yr)){  
 IDSAT$int1[IDSAT$Grade == i & IDSAT$Yr == j] <- 1  
 }  
}  
IDSAT$int2 <- 0  
for (i in c(5:11)) {  
 for (j in unique(IDSAT$Yr)){  
 IDSAT$int2[IDSAT$Grade == i & IDSAT$Yr == j] <- 1  
 }  
}  
#  
IDSAT$int3 <- 0  
for (i in c(6:12)) {  
 for (j in unique(IDSAT$Yr)){  
 IDSAT$int3[IDSAT$Grade == i & IDSAT$Yr == j] <- 1  
 }  
}  
  
# Subsetting for Years and Subjects####  
  
# Create a vector of list of names to use for assignment  
IDSATYears <- 0  
for (i in c(7:13)) {  
 IDSATYears[i - 6] <- paste('IDSAT', i, sep = '')  
}  
# Create a vector of list of names to use for assignment  
YearNames <- 0  
for (i in c(2007:2013)) {  
 YearNames[i - 2006] <- paste0(i, '-01-01')  
}  
# Subset by years  
for (i in c(1:length(unique(IDSAT$Yr)))) {  
 assign(IDSATYears[i], IDSAT[IDSAT$Yr == unique(IDSAT$Yr)[i], ])  
}  
# Subset by subject  
for (i in c(1:length(unique(IDSAT$Subject)))) {  
 nam <- paste('IDSAT', unique(IDSAT$Subject)[i], sep = ".")  
 assign(nam, IDSAT[IDSAT$Subject == unique(IDSAT$Subject)[i], ])  
}  
# Subset each subject by year  
for (j in c(1:length(IDSATYears))) {  
 for (i in c(1:4)) {  
 nam <- paste(IDSATYears[j], unique(IDSAT$Subject)[i], sep = ".")  
 assign(nam, IDSAT[IDSAT$Subject == unique(IDSAT$Subject)[i] &  
 IDSAT$Yr == YearNames[j], ])  
 }  
}  
# Remove all NA values on the endogenous variable measuring pass rate  
balanced13 <- subset(IDSAT13, (!is.na(IDSAT13$All.P)))  
  
# Plots ####  
#  
# Histograms of percent of students passing for all years and then by subjects  
# for all years  
#  
colorsV <-  
 c(  
 rgb(1, 0, 0, 0.2),  
 rgb(0, 1, 0, 0.2),  
 rgb(0, 0, 1, 0.2),  
 rgb(0, 0, 0.5, 0.2),  
 rgb(0.5, 0, 0.5, 0.2),  
 rgb(0.5, 0.5,0, 0.2)  
 )  
require(ggplot2)

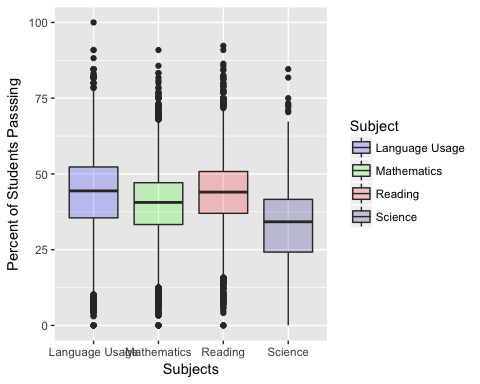
## Loading required package: ggplot2

IDSATm <- melt(IDSAT, id.var = 'Subject')

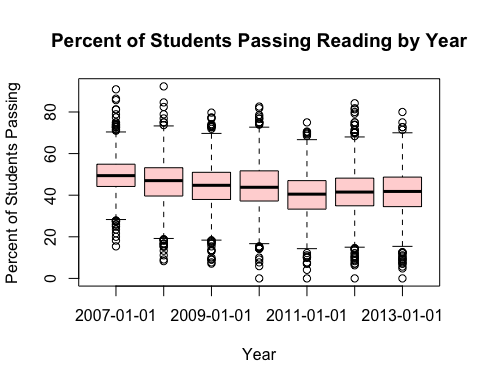
## Warning: attributes are not identical across measure variables; they will  
## be dropped

#  
ggplot(data = IDSAT, aes(x = IDSAT$Subject, y = IDSAT$All.P)) + geom\_boxplot(aes(fill = Subject)) + labs(x = 'Subjects', y = 'Percent of Students Passsing') + scale\_fill\_manual(values = c(colorsV[3], colorsV[2], colorsV[1], colorsV[4]))

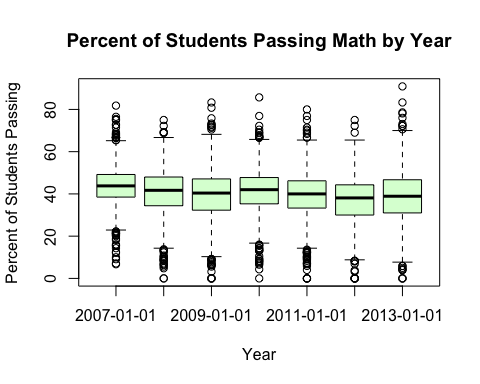
## Warning: Removed 5097 rows containing non-finite values (stat\_boxplot).



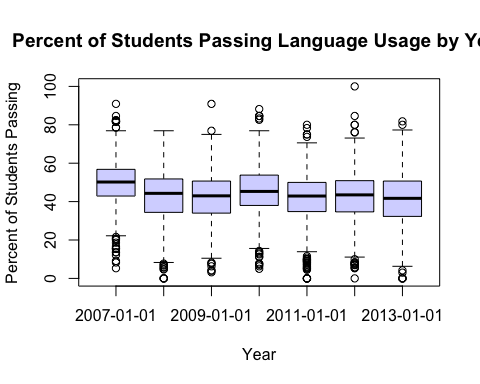
#  
par(mfrow = c(1, 1))  
boxplot(IDSAT.Reading$All.P ~ IDSAT.Reading$Yr, ylab = 'Percent of Students Passing', xlab = 'Year', main = 'Percent of Students Passing Reading by Year', col = colorsV[1])



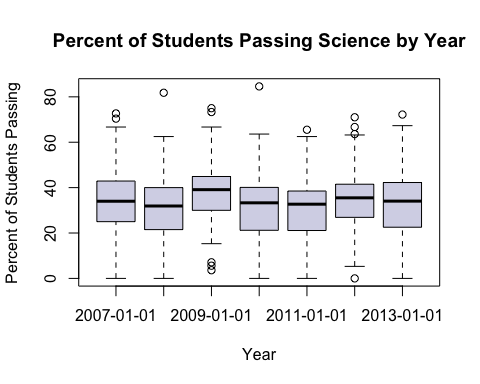
#  
boxplot(IDSAT.Mathematics$All.P ~ IDSAT.Mathematics$Yr, ylab = 'Percent of Students Passing', xlab = 'Year', main = 'Percent of Students Passing Math by Year', col = colorsV[2])



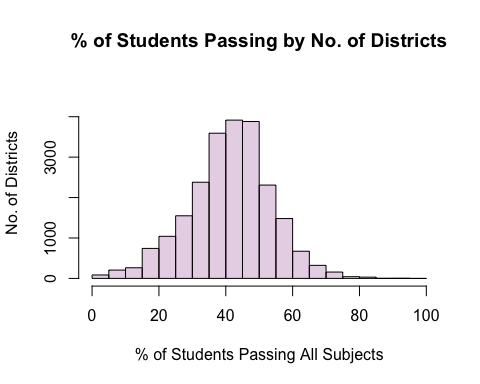
#  
boxplot(`IDSAT.Language Usage`$All.P ~ `IDSAT.Language Usage`$Yr, ylab = 'Percent of Students Passing', xlab = 'Year', main = 'Percent of Students Passing Language Usage by Year', col = colorsV[3])



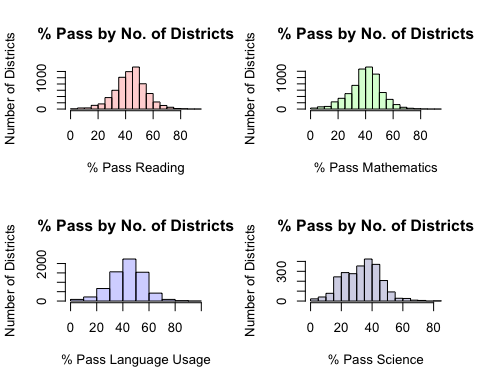
#  
boxplot(IDSAT.Science$All.P ~ IDSAT.Science$Yr, ylab = 'Percent of Students Passing', xlab = 'Year', main = 'Percent of Students Passing Science by Year', col = colorsV[4])



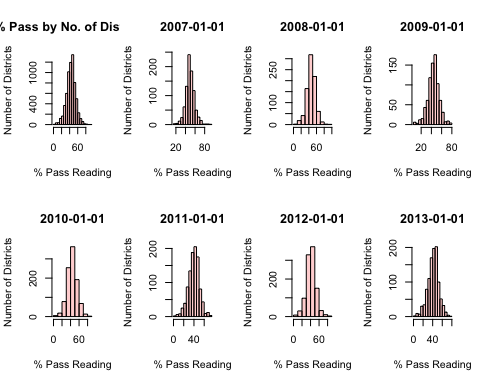
#  
# All Passing  
hist(  
 IDSAT$All.P,  
 xlab = '% of Students Passing All Subjects',  
 ylab = 'No. of Districts',  
 ylim = c(0, 4750) ,  
 main = '% of Students Passing by No. of Districts',  
 col = colorsV[5]  
)



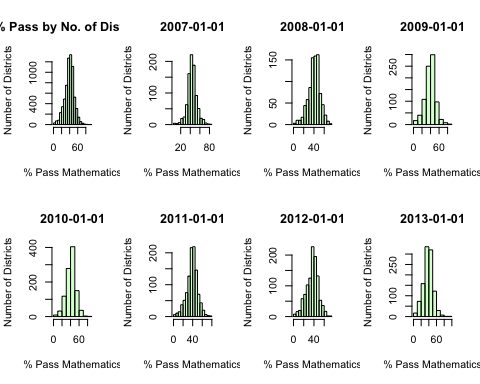
#  
par(mfrow = c(2, 2))  
for (j in c(1:4)) {  
 hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[j]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[j]),  
 ylab = 'Number of Districts',  
 main = '% Pass by No. of Districts',  
 col = colorsV[j]  
 )  
}



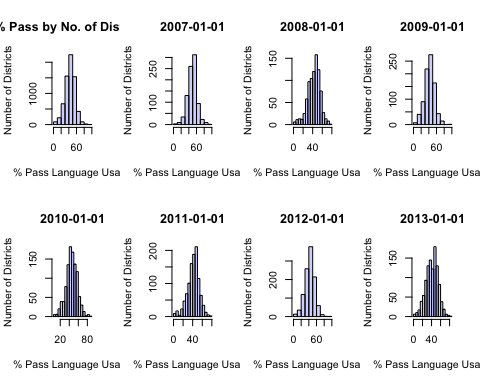
# Reading  
par(mfrow = c(2, 4))  
hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[1]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[1]),  
 ylab = 'Number of Districts' ,  
 main = '% Pass by No. of Districts',  
 col = colorsV[1]  
)  
for (i in c(1:length(IDSATYears))) {  
 hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[1] &  
 IDSAT$Yr == unique(IDSAT$Yr)[i]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[1]),  
 ylab = 'Number of Districts' ,  
 main = unique(IDSAT$Yr)[i],  
 col = colorsV[1]  
 )  
}



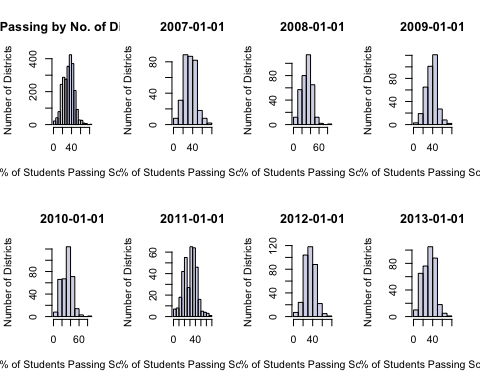
# Math  
hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[2]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[2]),  
 ylab = 'Number of Districts' ,  
 main = '% Pass by No. of Districts',  
 col = colorsV[2]  
)  
for (i in c(1:length(IDSATYears))) {  
 hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[2] &  
 IDSAT$Yr == unique(IDSAT$Yr)[i]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[2]),  
 ylab = 'Number of Districts' ,  
 main = unique(IDSAT$Yr)[i],  
 col = colorsV[2]  
 )  
}



# Language Usage  
hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[3]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[3]),  
 ylab = 'Number of Districts' ,  
 main = '% Pass by No. of Districts',  
 col = colorsV[3]  
)  
for (i in c(1:length(IDSATYears))) {  
 hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[3] &  
 IDSAT$Yr == unique(IDSAT$Yr)[i]],  
 xlab = paste('% Pass', unique(IDSAT13$Subject)[3]),  
 ylab = 'Number of Districts' ,  
 main = unique(IDSAT$Yr)[i],  
 col = colorsV[3]  
 )  
}



#  
hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[4]],  
 xlab = paste('% of Students Passing', unique(IDSAT13$Subject)[4]),  
 ylab = 'Number of Districts' ,  
 main = '% Passing by No. of Districts',  
 col = colorsV[4]  
)  
for (i in c(1:length(IDSATYears))) {  
 hist(  
 IDSAT$All.P[IDSAT$Subject == unique(IDSAT$Subject)[4] &  
 IDSAT$Yr == unique(IDSAT$Yr)[i]],  
 xlab = paste('% of Students Passing', unique(IDSAT13$Subject)[4]),  
 ylab = 'Number of Districts' ,  
 main = unique(IDSAT$Yr)[i],  
 col = colorsV[4]  
 )  
}



#  
summary(balanced13$TotPPE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 4336 7368 8787 9740 11229 39939 3

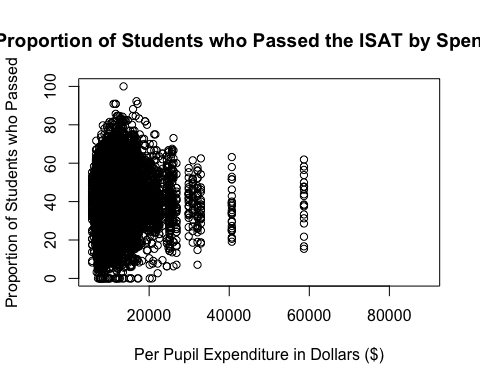
quantile(IDSAT$TotPPE, na.rm = TRUE)

## 0% 25% 50% 75% 100%   
## 3795.79 7763.63 9341.45 12112.36 83955.56

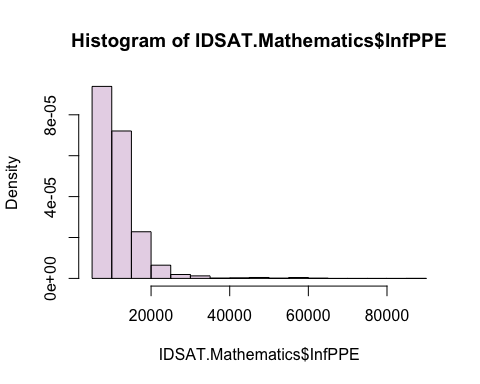
anova(lm(IDSAT$All.P ~ IDSAT$Dist))

## Analysis of Variance Table  
##   
## Response: IDSAT$All.P  
## Df Sum Sq Mean Sq F value Pr(>F)   
## IDSAT$Dist 151 367257 2432.16 17.327 < 2.2e-16 \*\*\*  
## Residuals 22528 3162294 140.37   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

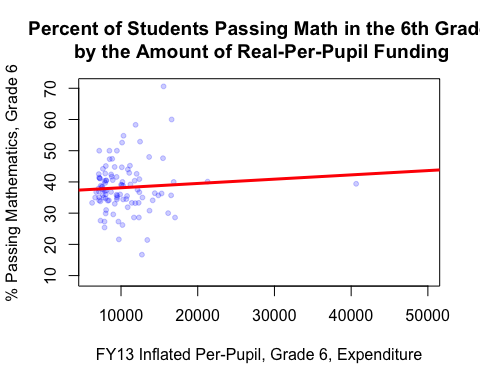
reg1 <- lm(All.P ~ Mem + FRL + Day, data = IDSAT)  
  
# Graphics ####  
par(mfrow = c(1,1))  
plot(  
 All.P ~ InfPPE,  
 data = IDSAT,  
 xlab = 'Per Pupil Expenditure in Dollars ($)',  
 ylab = 'Proportion of Students who Passed',  
 main = 'Proportion of Students who Passed the ISAT by Spending'  
)



#  
hist(IDSAT.Mathematics$InfPPE, col = colorsV[5], freq = FALSE)



#  
attach(IDSAT13.Mathematics)  
plot(  
 All.P[Grade == 6] ~ InfPPE[Grade == 6],  
 pch = 20,  
 col = colorsV[3],  
 xlab = 'FY13 Inflated Per-Pupil, Grade 6, Expenditure',  
 ylab = '% Passing Mathematics, Grade 6',  
 main = 'Percent of Students Passing Math in the 6th Grade\n by the Amount of Real-Per-Pupil Funding'  
)  
reg1 <-  
 lm(All.P[Grade == 6] ~ InfPPE[Grade == 6])  
abline(reg1, col = 'red', lwd = 3)



reg1

##   
## Call:  
## lm(formula = All.P[Grade == 6] ~ InfPPE[Grade == 6])  
##   
## Coefficients:  
## (Intercept) InfPPE[Grade == 6]   
## 3.676e+01 1.377e-04

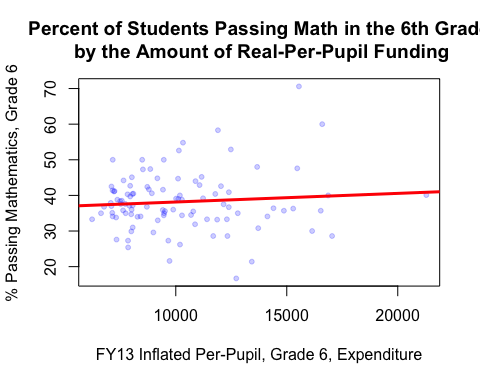
# Balance and identify the outlier  
balanced13M <- IDSAT13.Mathematics[!(is.na(IDSAT13.Mathematics$All.P) | is.na(IDSAT13.Mathematics$InfPPE)), ]  
spendOutlier <- max(balanced13M$InfPPE)  
spendOutlier

## [1] 40649.87

detach()  
b13MNoOut <- balanced13M[!(balanced13M$InfPPE == spendOutlier),]  
attach(b13MNoOut)  
spendOutlier <- max(InfPPE)  
#  
plot(  
 All.P[Grade == 6] ~ InfPPE[Grade == 6],  
 pch = 20,  
 col = colorsV[3],  
 xlab = 'FY13 Inflated Per-Pupil, Grade 6, Expenditure',  
 ylab = '% Passing Mathematics, Grade 6',  
 main = 'Percent of Students Passing Math in the 6th Grade\n by the Amount of Real-Per-Pupil Funding'  
)  
reg2 <-  
 lm(All.P[Grade == 6] ~ InfPPE[Grade == 6])  
reg2

##   
## Call:  
## lm(formula = All.P[Grade == 6] ~ InfPPE[Grade == 6])  
##   
## Coefficients:  
## (Intercept) InfPPE[Grade == 6]   
## 3.572e+01 2.429e-04

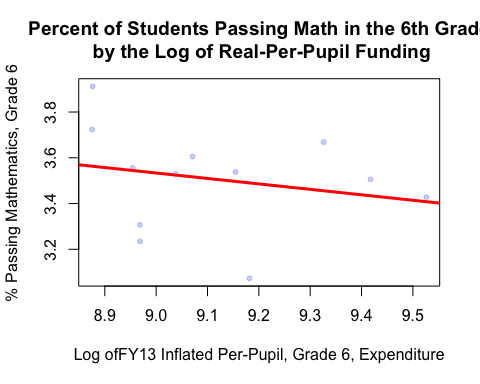
abline(reg2, col = 'red', lwd = 3)



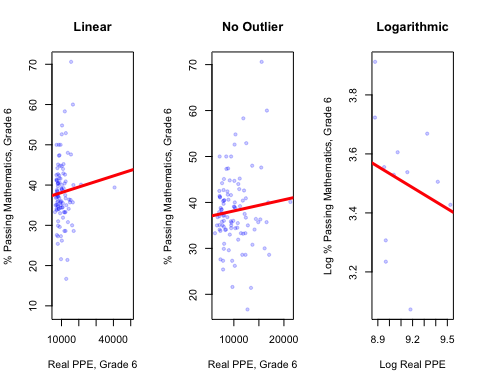
#  
lInfPPE <- log(InfPPE[Grade == 6])  
lAll.P <- log(All.P[Grade == 6])  
plot(  
 lAll.P[Grade == 6] ~ lInfPPE[Grade == 6],  
 pch = 20,  
 col = colorsV[3],  
 xlab = 'Log ofFY13 Inflated Per-Pupil, Grade 6, Expenditure',  
 ylab = '% Passing Mathematics, Grade 6',  
 main = 'Percent of Students Passing Math in the 6th Grade\n by the Log of Real-Per-Pupil Funding'  
)  
reg3 <-  
 lm(lAll.P[Grade == 6] ~ lInfPPE[Grade == 6])  
reg3

##   
## Call:  
## lm(formula = lAll.P[Grade == 6] ~ lInfPPE[Grade == 6])  
##   
## Coefficients:  
## (Intercept) lInfPPE[Grade == 6]   
## 5.6818 -0.2387

abline(reg3, col = 'red', lwd = 3)



#  
#  
detach()  
attach(IDSAT13.Mathematics)  
par(mfrow = c(1,3))  
plot(  
 All.P[Grade == 6] ~ InfPPE[Grade == 6],  
 pch = 20,  
 col = colorsV[3],  
 xlab = 'Real PPE, Grade 6',  
 ylab = '% Passing Mathematics, Grade 6',  
 main = 'Linear'  
)  
abline(reg1, col = 'red', lwd = 3)  
  
detach()  
attach(b13MNoOut)  
plot(  
 All.P[Grade == 6] ~ InfPPE[Grade == 6],  
 pch = 20,  
 col = colorsV[3],  
 xlab = 'Real PPE, Grade 6',  
 ylab = '% Passing Mathematics, Grade 6',  
 main = 'No Outlier'  
)  
abline(reg2, col = 'red', lwd = 3)  
  
lInfPPE <- log(InfPPE[Grade == 6])  
lAll.P <- log(All.P[Grade == 6])  
plot(  
 lAll.P[Grade == 6] ~ lInfPPE[Grade == 6],  
 pch = 20,  
 col = colorsV[3],  
 xlab = 'Log Real PPE',  
 ylab = 'Log % Passing Mathematics, Grade 6',  
 main = 'Logarithmic'  
)  
abline(reg3, col = 'red', lwd = 3)



#  
# Time Series ####  
detach()  
#  
# Subset to watch a cohort travel through time:  
intZero <- IDSAT.Mathematics[IDSAT.Mathematics$int0 == 1,]  
intZero <- ts(intZero$Yr, start = c(2007, 1), end = c(2013,1), frequency = 1)  
# frequency(intZero)  
# str(intZero)  
# plot(intZero$All.P, col = 'blue', lwd = 3, ylab = 'Exchange Rate')  
# abline(reg = lm(intZero$All.P~time(intZero)), lwd = 3)  
#  
intOne <- IDSAT.Mathematics[IDSAT.Mathematics$int1 == 1,]  
intOne <- ts(intOne, start = c(2007, 1), end = c(2013,1), frequency = 1)

## Warning in data.matrix(data): NAs introduced by coercion

## Warning in data.matrix(data): NAs introduced by coercion  
  
## Warning in data.matrix(data): NAs introduced by coercion

frequency(intOne)

## [1] 1

str(intOne)

## Time-Series [1:7, 1:116] from 2007 to 2013: NA NA NA NA NA NA NA NA NA NA ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:116] "uniqueid" "AdID" "Dist" "Yr" ...

#  
intTwo <- IDSAT.Mathematics[IDSAT.Mathematics$int2 == 1,]  
intTwo <- ts(intTwo, start = c(2007, 1), end = c(2013,1), frequency = 1)

## Warning in data.matrix(data): NAs introduced by coercion  
  
## Warning in data.matrix(data): NAs introduced by coercion  
  
## Warning in data.matrix(data): NAs introduced by coercion

frequency(intTwo)

## [1] 1

str(intTwo)

## Time-Series [1:7, 1:116] from 2007 to 2013: NA NA NA NA NA NA NA NA NA NA ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:116] "uniqueid" "AdID" "Dist" "Yr" ...

#  
intThree <- IDSAT.Mathematics[IDSAT.Mathematics$int3 == 1,]  
intThree <- ts(intThree, start = c(2007, 1), end = c(2013,1), frequency = 1)

## Warning in data.matrix(data): NAs introduced by coercion  
  
## Warning in data.matrix(data): NAs introduced by coercion  
  
## Warning in data.matrix(data): NAs introduced by coercion

frequency(intThree)

## [1] 1

str(intThree)

## Time-Series [1:7, 1:116] from 2007 to 2013: NA NA NA NA NA NA NA NA NA NA ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:116] "uniqueid" "AdID" "Dist" "Yr" ...

#  
# Variable Meanings ####  
#  
# Definition: Dist "District number"  
#  
# Definition: Yr "Year of observation (2007-2013)"  
#  
# Definition: Grade "Grade of tested students in a district (3-12)"  
#  
# Definition: AnnExp "Total annual expenditure in a district for a particular  
# year"  
#  
# Definition: Tax "Total amount of tax revenue in a district for a particular  
# year"  
#  
# Definition: OthLcl "Total amount of revenue from local sources not contained  
# in other revenue streams"  
#  
# Definition: State "Total amount of revenue from state sources"  
#  
# Definition: Fed "Total amount of revenue from federal sources"  
#  
# Definition: Oth "Total amount of revenue from sources not contained in any  
# other revenue stream"  
#  
# Definition: Mem "Total number of students in a district on a particular day"  
#  
# Definition: TotPPE "Total per-pupil-expenditure by district"  
#  
# Definition: Day "Binary variable indicating if a school district has 4-day  
# school weeks"  
#  
# Definition: FRL "Total number of students in a district receiving free or  
# reduced lunch"  
#  
# Definition: PopEstDist "Estimate from US Census of the population in a  
# district"  
#  
# Definition: Pop517EstDist "Estimate from US Census of the population aged 5 to  
# 17 years in a district"  
#  
# Definition: Pov "Estimate from US Census of the population aged 5 to 17 years  
# living in poverty in a district"  
#  
# Definition: InfInd13 "Index for Inflation in a fiscal year with fiscal year  
# 2013 as the base year (Jul-Jun)"  
#  
# Definition: AnnInfExp "Annual expenditure Inflated to 2013 dollars"  
#  
# Definition: InfPPE "Per-pupil-expenditure Inflated to 2013 dollars"  
#  
# Definition: allss "Average scaled score for all tested"  
#  
# Definition: allbb "Percent of below basic for all tested"  
#  
# Definition: allb "Percent of basic for all tested"  
#  
# Definition: allp "Percent of proficient for all tested"  
#  
# Definition: alla "Percent of advanced for all tested"  
#  
# Definition: maless "Average scaled score for all Males tested"  
#  
# Definition: maletested "Number of Males tested"  
#  
# Definition: malebb "Percent of below basic for Males tested"  
#  
# Definition: maleb "Percent of basic for Males tested"  
#  
# Definition: malep "Percent of proficient for Males tested"  
#  
# Definition: malea "Percent of advanced for Males tested"  
#  
# Definition: femaless "Average scaled score for all Females tested"  
#  
# Definition: femaletested "Number of Females tested"  
#  
# Definition: femalebb "Percent of below basic for Females tested"  
#  
# Definition: femaleb "Percent of basic for Females tested"  
#  
# Definition: femalep "Percent of proficient for Females tested"  
#  
# Definition: femalea "Percent of advanced for Females tested"  
#  
# Definition: aianss "Average scaled score for all American Indian or Alaskan  
# Native tested"  
#  
# Definition: aiantested "Number of American Indian or Alaskan Native tested"  
#  
# Definition: aianbb "Percent of below basic for American Indian or Alaskan  
# Native tested"  
#  
# Definition: aianb "Percent of basic for American Indian or Alaskan Native  
# tested"  
#  
# Definition: aianp "Percent of proficient for American Indian or Alaskan Native  
# tested"  
#  
# Definition: aiana "Percent of advanced for American Indian or Alaskan Native  
# tested"  
#  
# Definition: asianss "Average scaled score for all Asian or Pacific Islander  
# tested"  
#  
# Definition: asiantested "Number of Asian or Pacific Islander tested"  
#  
# Definition: asianbb "Percent of below basic for Asian or Pacific Islander  
# tested"  
#  
# Definition: asianb "Percent of basic for Asian or Pacific Islander tested"  
#  
# Definition: asianp "Percent of proficient for Asian or Pacific Islander  
# tested"  
#  
# Definition: asiana "Percent of advanced for Asian or Pacific Islander tested"  
#  
# Definition: bafamss "Average scaled score for all Black / African American  
# tested"  
#  
# Definition: bafamtested "Number of Black / African American tested"  
#  
# Definition: bafambb "Percent of below basic for Black / African American  
# tested"  
#  
# Definition: bafamb "Percent of basic for Black / African American tested"  
#  
# Definition: bafamp "Percent of proficient for Black / African American tested"  
#  
# Definition: bafama "Percent of advanced for Black / African American tested"  
#  
# Definition: nhopiss "Average scaled score for all Native Hawaiian / Other  
# Pacific Islander tested"  
#  
# Definition: nhopitested "Number of Native Hawaiian / Other Pacific Islander  
# tested"  
#  
# Definition: nhopibb "Percent of below basic for Native Hawaiian / Other  
# Pacific Islander tested"  
#  
# Definition: nhopib "Percent of basic for Native Hawaiian / Other Pacific  
# Islander tested"  
#  
# Definition: nhopip "Percent of proficient for Native Hawaiian / Other Pacific  
# Islander tested"  
#  
# Definition: nhopia "Percent of advanced for Native Hawaiian / Other Pacific  
# Islander tested"  
#  
# Definition: whitess "Average scaled score for all White tested"  
#  
# Definition: whitetested "Number of White tested"  
#  
# Definition: whitebb "Percent of below basic for White tested"  
#  
# Definition: whiteb "Percent of basic for White tested"  
#  
# Definition: whitep "Percent of proficient for White tested"  
#  
# Definition: whitea "Percent of advanced for White tested"  
#  
# Definition: hisplatss "Average scaled score for all Hispanic or Latino tested"  
#  
# Definition: hisplattested "Number of Hispanic or Latino tested"  
#  
# Definition: hisplatbb "Percent of below basic for Hispanic or Latino tested"  
#  
# Definition: hisplatb "Percent of basic for Hispanic or Latino tested"  
#  
# Definition: hisplatp "Percent of proficient for Hispanic or Latino tested"  
#  
# Definition: hisplata "Percent of advanced for Hispanic or Latino tested"  
#  
# Definition: tworacesss "Average scaled score for all Other/Unknown tested"  
#  
# Definition: tworacestested "Number of Other/Unknown tested"  
#  
# Definition: tworacesbb "Percent of below basic for Other/Unknown tested"  
#  
# Definition: tworacesb "Percent of basic for Other/Unknown tested"  
#  
# Definition: tworacesp "Percent of proficient for Other/Unknown tested"  
#  
# Definition: tworacesa "Percent of advanced for Other/Unknown tested"  
#  
# Definition: frlss "Average scaled score for all Free or Reduced Lunch tested"  
#  
# Definition: frltested "Number of Free or Reduced Lunch tested"  
#  
# Definition: frlbb "Percent of below basic for Free or Reduced Lunch tested"  
#  
# Definition: frlb "Percent of basic for Free or Reduced Lunch tested"  
#  
# Definition: frlp "Percent of proficient for Free or Reduced Lunch tested"  
#  
# Definition: frla "Percent of advanced for Free or Reduced Lunch tested"  
#  
# Definition: lepss "Average scaled score for all Limited English Proficient  
# tested"  
#  
# Definition: leptested "Number of Limited English Proficient tested"  
#  
# Definition: lepbb "Percent of below basic for Limited English Proficient  
# tested"  
#  
# Definition: lepb "Percent of basic for Limited English Proficient tested"  
#  
# Definition: lepp "Percent of proficient for Limited English Proficient tested"  
#  
# Definition: lepa "Percent of advanced for Limited English Proficient tested"  
#  
# Definition: migss "Average scaled score for all Migrant tested"  
#  
# Definition: migtested "Number of Migrant tested"  
#  
# Definition: migbb "Percent of below basic for Migrant tested"  
#  
# Definition: migb "Percent of basic for Migrant tested"  
#  
# Definition: migp "Percent of proficient for Migrant tested"  
#  
# Definition: miga "Percent of advanced for Migrant tested"#  
#  
# Definition: spess "Average scaled score for all Special Education tested"  
#  
# Definition: spetested "Number of Special Education tested"  
#  
# Definition: spebb "Percent of below basic for Special Education tested"  
#  
# Definition: speb "Percent of basic for Special Education tested"  
#  
# Definition: spep "Percent of proficient for Special Education tested"  
#  
# Definition: spea "Percent of advanced for Special Education tested"  
#  
# Definition: tiass "Average scaled score for all Title 1 A tested"  
#  
# Definition: tiatested "Number of Title 1 A tested"  
#  
# Definition: tiabb "Percent of below basic for Title 1 A tested"  
#  
# Definition: tiab "Percent of basic for Title 1 A tested"  
#  
# Definition: tiap "Percent of proficient for Title 1 A tested"  
#  
# Definition: tiaa "Percent of advanced for Title 1 A tested"