Stat 139 Final Project: Statistical Sleuthing through Criminal Models

William Chen, Yuan Jiang, Patrick Xu, and Alice Zhao

December 10, 2013

For our project, we each focused on different analyses of the data. Thus, we have split the R code into four portions: multiple linear regression, t-tests, one-way ANOVAS, and visualizations. Since we each conducted our own analyses, variables may be named differently in each section.

R Code: Multiple Linear Regression

```
library(scales)
library(ggplot2)
library(lattice)
library(corrplot)
library(plyr)
library(xtable)

##############################################################################################
# Process the Data
##############################################################################################

# UNNORMALIZED DATASET
cities <- read.csv("~/Dropbox/Stat 139 Final
    Project/data/communities.unnormalized.data.txt", header=F)
names(cities) <- c("county", "state", "community", "communityname", "fold",
    "population", "householdsize", "racepctblack", "racePctWhite", "racePctAsian",
    "racePctHisp", "agePct12t21", "agePct12t29", "agePct16t24", "agePct65up",
    "numbUrban", "pctUrban", "medIncome", "pctWWage", "pctWFarmSelf", "pctWInvInc",
    "pctWSocSec", "pctWPubAsst", "pctWRetire", "medFamInc", "perCapInc", "whitePerCap",
    "blackPerCap", "indianPerCap", "AsianPerCap", "OtherPerCap", "HispPerCap",
    "NumUnderPov", "PctPopUnderPov", "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
    "PctUnemployed", "PctEmploy", "PctEmplManu", "PctEmplProfServ", "PctOccupManu",
    "PctOccupMgmtProf", "MalePctDivorce", "MalePctNevMarr", "FemalePctDiv",
    "TotalPctDiv", "PersPerFam", "PctFam2Par", "PctKids2Par", "PctYoungKids2Par",
    "PctTeen2Par", "PctWorkMomYoungKids", "PctWorkMom", "NumIlleg", "PctIlleg",
    "NumImmig", "PctImmigRecent", "PctImmigRec5", "PctImmigRec8", "PctImmigRec10",
    "PctRecentImmig", "PctRecImmig5", "PctRecImmig8", "PctRecImmig10",
    "PctSpeakEnglOnly", "PctNotSpeakEnglWell", "PctLargHouseFam", "PctLargHouseOccup",
    "PersPerOccupHous", "PersPerOwnOccHous", "PersPerRentOccHous", "PctPersOwnOccup",
    "PctPersDenseHous", "PctHousLess3BR", "MedNumBR", "HousVacant", "PctHousOccup",
    "PctHousOwnOcc", "PctVacantBoarded", "PctVacMore6Mos", "MedYrHousBuilt",
```

```
    "PctHousNoPhone", "PctWOFullPlumb", "OwnOccLowQuart", "OwnOccMedVal",
    "OwnOccHiQuart", "RentLowQ", "RentMedian", "RentHighQ", "MedRent",
    "MedRentPctHousInc", "MedOwnCostPctInc", "MedOwnCostPctIncNoMtg", "NumInShelters",
    "NumStreet", "PctForeignBorn", "PctBornSameState", "PctSameHouse85",
    "PctSameCity85", "PctSameState85", "LemasSwornFT", "LemasSwFTPerPop",
    "LemasSwFTFieldOps", "LemasSwFTFieldPerPop", "LemasTotalReq", "LemasTotReqPerPop",
    "PolicReqPerOffic", "PolicPerPop", "RacialMatchCommPol", "PctPolicWhite",
    "PctPolicBlack", "PctPolicHisp", "PctPolicAsian", "PctPolicMinor",
    "OfficAssgnDrugUnits", "NumKindsDrugsSeiz", "PolicAveOTWorked", "LandArea",
    "PopDens", "PctUsePubTrans", "PolicCars", "PolicOperBudg", "LemasPctPolicOnPatr",
    "LemasGangUnitDeploy", "LemasPctOfficDrugUn", "PolicBudgPerPop",
    "ViolentCrimesPerPop")

# Remove the 0 crime places
cities <- cities[cities$ViolentCrimesPerPop > 0, ]
cities$LogViolentCrimesPerPop <- log(cities$ViolentCrimesPerPop)

# NORMALIZED DATASET
#cities <- read.csv("~/Dropbox/Stat 139 Final Project/data/communities.data.txt",
    header=F)

#state.names <- c("AL", "AK", "T", "AZ", "AR", "CA", "T", "CO", "CT", "DE", "DC", "FL",
    "GA", "T", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD", "MA", "MI",
    "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ", "NM", "NY", "NC", "ND", "OH", "OK",
    "OR", "PA", "T", "RI", "SC", "SD", "TN", "TX", "UT", "VT", "VA", "T", "WA", "WV",
    "WI", "WY")
#cities$state.names <- state.names[cities$state]


##############################################################################################
# Exploratory Visualization
##############################################################################################

# create a function to plot the data
make.a.graph <- function(predictor) {
  df <- cities[, c(predictor, "LogViolentCrimesPerPop")]
  names(df) <- c("x", "y")
  ggplot(df, aes(x = x, y = y)) + geom_point() + geom_smooth(method = "lm") +
    #labs(x = predictor, y = "log violent crimes") +
    labs(x = NULL, y = NULL) +
    ggtitle(paste0(predictor, "\n R-sq:", round(cor(df$x, df$y, use =
        "pairwise.complete.obs")^2, 3))) +
    opts(plot.margin=unit(c(0, 0, 0, 0), "cm"),
    #ggtitle(paste0(predictor, "\n R-squared:", round(cor(df$x, df$y, use =
        "pairwise.complete.obs")^2, 3)))
}

# make.a.graph("state")
# make.a.graph("county")
# make.a.graph("community")
# make.a.graph("communityname")
# make.a.graph("fold")
# cities$logpopulation <- log(cities$population)
# make.a.graph("population")
```

```
# make.a.graph("logpopulation")
# make.a.graph("householdsize")
# make.a.graph("racepctblack")
# make.a.graph("racePctWhite")
# make.a.graph("racePctAsian")
# make.a.graph("racePctHisp")
# make.a.graph("agePct12t21")
# make.a.graph("agePct12t29")
# make.a.graph("agePct16t24")
# make.a.graph("agePct65up")
# make.a.graph("numbUrban")
# make.a.graph("pctUrban")
# make.a.graph("medIncome")
# make.a.graph("pctWWage")
# make.a.graph("pctWFarmSelf")
# make.a.graph("pctWInvInc")
# make.a.graph("pctWSocSec")
# make.a.graph("pctWPubAsst")
# make.a.graph("pctWRetire")
# make.a.graph("medFamInc")
# make.a.graph("perCapInc")
# make.a.graph("whitePerCap")
# make.a.graph("blackPerCap")
# make.a.graph("indianPerCap")
# make.a.graph("AsianPerCap")
# make.a.graph("OtherPerCap")
# make.a.graph("HispPerCap")
# make.a.graph("NumUnderPov")
# make.a.graph("PctPopUnderPov") #
# make.a.graph("PctLess9thGrade")#
# make.a.graph("PctNotHSGrad")#
# make.a.graph("PctBSorMore")
# make.a.graph("PctUnemployed")#
# make.a.graph("PctEmploy")
# make.a.graph("PctEmplManu")
# make.a.graph("PctEmplProfServ")
# make.a.graph("PctOccupManu")
# make.a.graph("PctOccupMgmtProf")
# make.a.graph("MalePctDivorce") #
# make.a.graph("MalePctNevMarr")
# make.a.graph("FemalePctDiv")
# make.a.graph("TotalPctDiv") #
# make.a.graph("PersPerFam")
# make.a.graph("PctFam2Par") #####
# make.a.graph("PctKids2Par") ##
# make.a.graph("PctYoungKids2Par")
# make.a.graph("PctTeen2Par")
# make.a.graph("PctWorkMomYoungKids")
# make.a.graph("PctWorkMom")
# make.a.graph("NumIlleg")
# make.a.graph("PctIlleg") #
# make.a.graph("NumImmig")
# make.a.graph("PctImmigRecent")
# make.a.graph("PctImmigRec5")
```

```
# make.a.graph("PctImmigRec8")
# make.a.graph("PctImmigRec10")
# make.a.graph("PctRecentImmig")
# make.a.graph("PctRecImmig5")
# make.a.graph("PctRecImmig8")
# make.a.graph("PctRecImmig10")
# make.a.graph("PctSpeakEnglOnly") #
# make.a.graph("PctNotSpeakEnglWell") #
# make.a.graph("PctLargHouseFam") #
# make.a.graph("PctLargHouseOccup")
# make.a.graph("PersPerOccupHous")
# make.a.graph("PersPerOwnOccHous")
# make.a.graph("PersPerRentOccHous")
# make.a.graph("PctPersOwnOccup") #
# make.a.graph("PctPersDenseHous")
# make.a.graph("PctHousLess3BR") #
# make.a.graph("MedNumBR")
# make.a.graph("HousVacant")
# make.a.graph("PctHousOccup") #
# make.a.graph("PctHousOwnOcc")#
# make.a.graph("PctVacantBoarded")
# make.a.graph("PctVacMore6Mos")
# make.a.graph("MedYrHousBuilt")
# make.a.graph("PctHousNoPhone") #
# make.a.graph("PctWOFullPlumb") #
# make.a.graph("OwnOccLowQuart")
# make.a.graph("OwnOccMedVal")
# make.a.graph("OwnOccHiQuart")
# make.a.graph("RentLowQ")
# make.a.graph("RentMedian")
# make.a.graph("RentHighQ")
# make.a.graph("MedRent")
# make.a.graph("MedRentPctHousInc")
# make.a.graph("MedOwnCostPctInc")
# make.a.graph("MedOwnCostPctIncNoMtg")
# make.a.graph("NumInShelters")
# make.a.graph("NumStreet")
# make.a.graph("PctForeignBorn")
# make.a.graph("PctBornSameState")
# make.a.graph("PctSameHouse85")
# make.a.graph("PctSameCity85")
# make.a.graph("PctSameState85")
# make.a.graph("LemasSwornFT")
# make.a.graph("LemasSwFTPerPop")
# make.a.graph("LemasSwFTFieldOps")
# make.a.graph("LemasSwFTFieldPerPop")
# make.a.graph("LemasTotalReq")
# make.a.graph("LemasTotReqPerPop")
# make.a.graph("PolicReqPerOffic")
# make.a.graph("PolicPerPop")
# make.a.graph("RacialMatchCommPol")
# make.a.graph("PctPolicWhite")
# make.a.graph("PctPolicBlack")
# make.a.graph("PctPolicHisp")
```

```
# make.a.graph("PctPolicAsian")
# make.a.graph("PctPolicMinor")
# make.a.graph("OfficAssgnDrugUnits")
# make.a.graph("NumKindsDrugsSeiz")
# make.a.graph("PolicAveOTWorked")
# make.a.graph("LandArea")
# make.a.graph("PopDens") #
# make.a.graph("PctUsePubTrans")
# make.a.graph("PolicCars")
# make.a.graph("PolicOperBudg")
# make.a.graph("LemasPctPolicOnPatr")
# make.a.graph("LemasGangUnitDeploy")
# make.a.graph("LemasPctOfficDrugUn")
# make.a.graph("PolicBudgPerPop")
# make.a.graph("ViolentCrimesPerPop")

##############################################################################################################
# Plot for the update
##############################################################################################################

# create scatterplots of transformed explanatory variable and response variable
make.a.graph("logpopulation")
cities$logpopulation <- log(cities$population)

make.a.graph("racepctblack")
cities$logracepctblack <- log(cities$racepctblack)

make.a.graph("racePctWhite")

make.a.graph("agePct16t24")

make.a.graph("pctUrban")

make.a.graph("medIncome")
cities$logmedIncome <- log(cities$medIncome)

make.a.graph("PctPopUnderPov") #

make.a.graph("PctLess9thGrade")#
cities$logPctLess9thGrade <- log(cities$PctLess9thGrade)

make.a.graph("logPctLess9thGrade")#

make.a.graph("PctUnemployed")#
cities$logPctUnemployed <- log(cities$PctUnemployed)

make.a.graph("logPctUnemployed")#

make.a.graph("PctFam2Par") #####

make.a.graph("PctKids2Par") ##

make.a.graph("PctIlleg") #
cities$logPctIlleg <- log(cities$PctIlleg)
```

```
make.a.graph("logPctIlleg") #

make.a.graph("NumImmig")
cities$logNumImmig <- log(cities$NumImmig)

make.a.graph("logNumImmig")

make.a.graph("PctWOFullPlumb") #
cities$logPctWOFullPlumb <- log(cities$PctWOFullPlumb)

make.a.graph("logPctWOFullPlumb") #

# choose 12 predictors and observe their scatterplots
p <- list()
p[[1]] <- make.a.graph("logpopulation")
p[[2]] <- make.a.graph("logracepctblack")
p[[3]] <- make.a.graph("racePctWhite")
p[[4]] <- make.a.graph("logmedIncome")
p[[5]] <- make.a.graph("PctPopUnderPov")
p[[6]] <- make.a.graph("logPctLess9thGrade")
p[[7]] <- make.a.graph("logPctUnemployed")#
p[[8]] <- make.a.graph("PctFam2Par") #####
p[[9]] <- make.a.graph("PctKids2Par") ##
p[[10]] <- make.a.graph("logPctIlleg") #
p[[11]] <- make.a.graph("logNumImmig")
p[[12]] <- make.a.graph("logPctWOFullPlumb") #

# VISUALIZATION 1: scatterplots
multiplot(plotlist = p, cols=4)

covariates.we.like <- cities[,c("LogViolentCrimesPerPop", "logpopulation",
    "logracepctblack", "racePctWhite", "logmedIncome", "PctPopUnderPov",
    "logPctLess9thGrade", "logPctUnemployed", "PctFam2Par", "PctKids2Par",
    "logPctIlleg", "logNumImmig", "logPctWOFullPlumb")]

# VISUALIZATION 2: correlation matrix
corr.matrix <- cor(covariates.we.like)
colnames(corr.matrix) <- 0:12
rownames(corr.matrix) <- 0:12
corrplot(corr.matrix, tl.pos="lt", type="upper",
        tl.col="black", tl.cex=0.6, tl.srt=45,
        addCoef.col="black", addCoefasPercent = TRUE,
        p.mat = 1-abs(corr.matrix), sig.level=0.50, insig = "blank")
# http://stackoverflow.com/questions/15887212/heatmap-or-plot-for-a-correlation-matrix

# create multiple linear models: constant mean model, main effects model, main effecits
    and
# pairwise interactions, backwards stepwise regression model, forwards stepwise
    regression model
simplemodel <- lm(LogViolentCrimesPerPop ~ 1., covariates.we.like)
fullmodel.nointeractions <- lm(LogViolentCrimesPerPop ~ ., covariates.we.like)
fullmodel.interactions <- lm(LogViolentCrimesPerPop ~ .^2, covariates.we.like)
backwards.model <- step(fullmodel.interactions, direction = "backward")
```

```
forwards.model <- step(object = simplemodel, scope = list(upper =
    fullmodel.interactions, lower = simplemodel), direction = "forward")

# will work with backwards stepwise regression model because of highest R^2

# VISUALIZATION 3: Residual plot of backwards stepwise regression model
upper.residual <- 0.29427032 + 1.5 * (0.29427032 + 0.30639917)
lower.residual <- -0.30639917 - 1.5 * (0.29427032 + 0.30639917)

# analyse the residuals for constant variance, independence, and normality
p1 <- qplot(predict(backwards.model), resid(backwards.model)) + geom_hline(yintercept =
    0) + geom_hline(yintercept = c(upper.residual, lower.residual), linetype =
    "longdash") + labs("x" = "Fitted Values", y = "Residuals", title = "Residual Plot")
p2 <- qplot(resid(backwards.model)) + labs(x = "Residual", y = "Count", title =
    "Histogram of Residuals")
p3 <- qplot(sample = resid(backwards.model), stat = "qq") + labs(x = "Theoretical
    Quantiles", y = "Empirical Quantiles", title = "Residual QQ Plot")
multiplot(p1, p2, p3, cols = 3)

# calculate R^2, adjusted R^2, sigma, etc.
modellist <- list(simplemodel, fullmodel.nointeractions, fullmodel.interactions,
    backwards.model, forwards.model)
model.adj.r.squared <- laply(modellist, function(x) summary(x)$adj.r.squared)
model.sigma <- laply(modellist, function(x) summary(x)$sigma)
model.r.squared <- laply(modellist, function(x) summary(x)$r.squared)
model.names <- c("Intercept (0)", "All Pred. (12)", "All Pred. + Inter. (78)",
    "Backwards (35)", "Forwards (5)")

# compare the models based on adjusted R^2
model.comparison <- data.frame(model.names, model.sigma, model.r.squared,
    model.adj.r.squared)
xtable(model.comparison, digits = 4)

# VISUALIZATION 6: histogram of all predictors
list.of.covariates <- c("LogViolentCrimesPerPop", "logpopulation", "logracepctblack",
    "racePctWhite", "logmedIncome", "PctPopUnderPov", "logPctLess9thGrade",
    "logPctUnemployed", "PctFam2Par", "PctKids2Par", "logPctIlleg", "logNumImmig",
    "logPctWOFullPlumb")

# create function for making histrograms of all predictors
make.histogram <- function(i) qplot(covariates.we.like[,i + 1], geom = "blank") +
  geom_histogram(aes(y = ..density..)) +
  geom_density(color = "blue") +
  labs(x = NULL, y = NULL, title = i) +
  opts(axis.text.x = theme_blank(),
       axis.text.y = theme_blank(),
       axis.ticks = theme_blank(),
       plot.margin=unit(c(0, 0, 0, 0), "cm"),
       panel.grid.major = theme_blank(),
       panel.grid.minor = theme_blank(),
       panel.border = theme_blank(),
       panel.background = theme_blank())
```

```
#make.qqplot <- function(i) qplot(sample = covariates.we.like[,i + 1], stat = "qq") +
    labs(x = NULL, y = NULL, title = i) + opts(axis.text.x = theme_blank(), axis.text.y
    = theme_blank(), axis.ticks = theme_blank())
histogram.plot.list <- llply(0:12, make.histogram)

#qq.plot.list <- llply(0:12, make.qqplot)
multiplot(plotlist = histogram.plot.list, cols=13)

#multiplot(plotlist = qq.plot.list, cols=13)

##########################################################################################################
# Helper
##########################################################################################################

#http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:  Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  require(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
```

```
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

R Code: t-tests

```
#load data
cities <- read.csv("~/Dropbox/Stat 139 Final
    Project/data/communities.unnormalized.data.txt", header=F)
names(cities) <- c("county", "state", "community", "communityname", "fold",
    "population", "householdsize", "racepctblack", "racePctWhite", "racePctAsian",
    "racePctHisp", "agePct12t21", "agePct12t29", "agePct16t24", "agePct65up",
    "numbUrban", "pctUrban", "medIncome", "pctWWage", "pctWFarmSelf", "pctWInvInc",
    "pctWSocSec", "pctWPubAsst", "pctWRetire", "medFamInc", "perCapInc", "whitePerCap",
    "blackPerCap", "indianPerCap", "AsianPerCap", "OtherPerCap", "HispPerCap",
    "NumUnderPov", "PctPopUnderPov", "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
    "PctUnemployed", "PctEmploy", "PctEmplManu", "PctEmplProfServ", "PctOccupManu",
    "PctOccupMgmtProf", "MalePctDivorce", "MalePctNevMarr", "FemalePctDiv",
    "TotalPctDiv", "PersPerFam", "PctFam2Par", "PctKids2Par", "PctYoungKids2Par",
    "PctTeen2Par", "PctWorkMomYoungKids", "PctWorkMom", "NumIlleg", "PctIlleg",
    "NumImmig", "PctImmigRecent", "PctImmigRec5", "PctImmigRec8", "PctImmigRec10",
    "PctRecentImmig", "PctRecImmig5", "PctRecImmig8", "PctRecImmig10",
    "PctSpeakEnglOnly", "PctNotSpeakEnglWell", "PctLargHouseFam", "PctLargHouseOccup",
    "PersPerOccupHous", "PersPerOwnOccHous", "PersPerRentOccHous", "PctPersOwnOccup",
    "PctPersDenseHous", "PctHousLess3BR", "MedNumBR", "HousVacant", "PctHousOccup",
    "PctHousOwnOcc", "PctVacantBoarded", "PctVacMore6Mos", "MedYrHousBuilt",
    "PctHousNoPhone", "PctWOFullPlumb", "OwnOccLowQuart", "OwnOccMedVal",
    "OwnOccHiQuart", "RentLowQ", "RentMedian", "RentHighQ", "MedRent",
    "MedRentPctHousInc", "MedOwnCostPctInc", "MedOwnCostPctIncNoMtg", "NumInShelters",
    "NumStreet", "PctForeignBorn", "PctBornSameState", "PctSameHouse85",
    "PctSameCity85", "PctSameState85", "LemasSwornFT", "LemasSwFTPerPop",
    "LemasSwFTFieldOps", "LemasSwFTFieldPerPop", "LemasTotalReq", "LemasTotReqPerPop",
    "PolicReqPerOffic", "PolicPerPop", "RacialMatchCommPol", "PctPolicWhite",
    "PctPolicBlack", "PctPolicHisp", "PctPolicAsian", "PctPolicMinor",
    "OfficAssgnDrugUnits", "NumKindsDrugsSeiz", "PolicAveOTWorked", "LandArea",
    "PopDens", "PctUsePubTrans", "PolicCars", "PolicOperBudg", "LemasPctPolicOnPatr",
    "LemasGangUnitDeploy", "LemasPctOfficDrugUn", "PolicBudgPerPop",
    "ViolentCrimesPerPop")
#Shift by 1 so that we can log data
cities$ViolentCrimesPerPop <- cities$ViolentCrimesPerPop + 1
cities$LogViolentCrimesPerPop <- log(cities$ViolentCrimesPerPop)

#######################################
# California vs. Wisconsin t-test
#######################################
#subsetting data for California/Wisconsin t test
cali <- cities$LogViolentCrimesPerPop[c(cities$state == "CA")]
wisc <- cities$LogViolentCrimesPerPop[c(cities$state == "WI")]
```

```r
#Graphical representation of data
boxplot(cali, wisc, xlab = "States", ylab = "Violent Crimes per Pop", names =
    c("California", "Wisconsin"), main = "Comparing California and Wisconsin")

#Checking for which t-test I should use
mean(cali) #.3322594
mean(wisc) #.1931954
var(cali) #.5626526
var(wisc) #.3557786
length(cali) #279
length(wisc) #60

#decided to go with an pooled t-test
t.test(cali, wisc, var.Equal="TRUE")

#p value of .1218, conclude that the two medians are not statistically different

#######################################
# North vs. South t-test
#######################################

#subset data (North is OR, WA, AK, ID, MT, WY, ND, SD, NE, MN, IA, WI, IL, MI, IN, OH,
    MD, PA, DE, NJ, NY, CT, RI, MA, NH, VT, ME)

northviolent <- cities$LogViolentCrimesPerPop[which(cities$state == "WA" | cities$state
    == "OR" | cities$state == "MT" | cities$state == "ID" | cities$state == "WY" |
    cities$state == "ND" | cities$state == "SD" | cities$state == "NE" | cities$state ==
    "MN" | cities$state == "IA" | cities$state == "WI" | cities$state == "IL" |
    cities$state == "IN" | cities$state == "MI" | cities$state == "OH" | cities$state ==
    "MD" | cities$state == "PA" | cities$state == "NY" | cities$state == "DE" |
    cities$state == "NJ" | cities$state == "CT" | cities$state == "RI" | cities$state ==
    "MA" | cities$state == "NH" | cities$state == "VT" | cities$state == "ME" |
    cities$state == "AK")]

#subset data (South is CA, NV, UT, AZ, CO, NM, KS, OK, TX MO, AR, LA, MS, TN, KY, WV,
    VA, NC, SC, GA, AL, MS, FL)
southviolent <- cities$LogViolentCrimesPerPop[which(cities$state == "CA" | cities$state
    == "NV" | cities$state == "UT" | cities$state == "AZ" | cities$state == "CO" |
    cities$state == "NM" | cities$state == "KS" | cities$state == "OK" | cities$state ==
    "TX" | cities$state == "LA" | cities$state == "AR" | cities$state == "MO" |
    cities$state == "KY" | cities$state == "WV" | cities$state == "VA" | cities$state ==
    "TN" | cities$state == "NC" | cities$state == "SC" | cities$state == "GA" |
    cities$state == "AL" | cities$state == "MS" | cities$state == "FL" | cities$state ==
    "HI")]

#Graphical Representation of Data
boxplot(northviolent, southviolent, xlab = "Regions", ylab = "Violent Crimes per Pop",
    names = c("North", "South"), main = "Comparing the North and South")
#check standard deviations/lengths
mean(northviolent) #.2330686
mean(southviolent) #.3458269
var(northviolent) #.403872
var(southviolent) #.5964718
```

```r
length(northviolent) #1191
length(southviolent) #1023

#Pretty similar, so pooled t-test
t.test(northviolent, southviolent, var.equal = TRUE)

#.04469 - p-value slightly below .05, so difference is statistically significant (South
    > North)

#######################################
# West vs. East t-test
#######################################

#subset data (West is CA, OR, WA, ID, NV, AZ, UT, WY, MT, CO, NM, ND, SD, NE, KS, OK,
    TX, LA, AR, MO, IA, MN, HI, AK)

west <- cities$LogViolentCrimesPerPop[which(cities$state == "WA" | cities$state == "OR"
    | cities$state == "CA" | cities$state == "NV" | cities$state == "ID" | cities$state
    == "MT" | cities$state == "WY" | cities$state == "UT" | cities$state == "CO" |
    cities$state == "AZ" | cities$state == "NM" | cities$state == "ND" | cities$state ==
    "SD" | cities$state == "NE" | cities$state == "KS" | cities$state == "OK" |
    cities$state == "TX" | cities$state == "HI" | cities$state == "AK" | cities$state ==
    "MN" | cities$state == "IA" | cities$state == "MO" | cities$state == "AR" |
    cities$state == "LA" | cities$state == "HI")]

#subset data (East is ME, VT, NH, MA, RI, CT, NY, PA, NH, DE, MD, OH, MI, IN, WI, IL,
    KY, WV, VA, NC, SC, TN, MS, AL, GA, FL)
east <- cities$LogViolentCrimesPerPop[which(cities$state == "WI" | cities$state == "IL"
    | cities$state == "MI" | cities$state == "IN" | cities$state == "OH" | cities$state
    == "KY" | cities$state == "TN" | cities$state == "MS" | cities$state == "AL" |
    cities$state == "GA" | cities$state == "FL" | cities$state == "SC" | cities$state ==
    "NC" | cities$state == "VA" | cities$state == "WV" | cities$state == "MD" |
    cities$state == "PA" | cities$state == "NY" | cities$state == "NJ" | cities$state ==
    "CT" | cities$state == "RI" | cities$state == "MA" | cities$state == "DE" |
    cities$state == "VT" | cities$state == "NH" | cities$state == "ME" )]

#Graphical representation of data
boxplot(west, east, xlab = "Regions", ylab = "Violent Crimes per Pop", names = c("West",
    "East"), main = "Comparing the West and East")

#check standard deviations/lengths
mean(west) #.286957
mean(east) #.2840727
var(west) #.47665
var(east) #.5051065
length(west) #842
length(east) #1372

#Pretty similar variances, so pooled t-test
t.test(west, east, var.equal = TRUE)

#p value of .9253, meaning there is no significant difference
```

```
#####################################
# Blue vs. Red t-test
#####################################

redcities <- cities$LogViolentCrimesPerPop[which(cities$state=='AK' | cities$state=='AZ'
    | cities$state=='UT' |cities$state=='ID' |cities$state=='MT' | cities$state=='WY'
    |cities$state=='ND' |cities$state=='SD' |cities$state=='NE' |cities$state=='KS'
    |cities$state=='OK' |cities$state=='TX' |cities$state=='MO' | cities$state=='AR'
    |cities$state=='LA' |cities$state=='MS' |cities$state=='AL' |cities$state=='GA' |
    cities$state=='FL' |cities$state=='SC' |cities$state=='TN' |cities$state=='NC' |
    cities$state =='VA' |cities$state=='WV' |cities$state=='KY' |cities$state=='IN')]
bluecities <- cities$LogViolentCrimesPerPop[which(cities$state !='T' &
    cities$state!='AK' & cities$state !='AZ' & cities$state !='UT' &cities$state!='ID' &
    cities$state!='MT' & cities$state !='WY' & cities$state !='ND' & cities$state !='SD'
    & cities$state !='NE' & cities$state !='KS' & cities$state !='OK' & cities$state
    !='TX' & cities$state !='MO' & cities$state !='AR' & cities$state !='LA'
    &cities$state !='MS' &cities$state !='AL' & cities$state !='GA' & cities$state
    !='FL' &cities$state !='SC' & cities$state !='TN' &cities$state !='NC' &
    cities$state !='VA' & cities$state !='WV' &cities$state !='KY' &cities$state !='IN')]

#T test comparing red vs blue states. First compare variances.
mean(redcities) #.3363742
mean(bluecities) #.2579846
var(redcities) #.5886017
var(bluecities) #.4412865
length(redcities) #786
length(bluecities) #1429
#Although variances are somewhat similar, population is very different and has inverse
    ratio of ratio of variance
#Thus, we use an unpooled t-test

t.test(redcities, bluecities, var.Equal = "TRUE")
#p-value of .01606, meaning there is a difference between red and blue states (red >
    blue)

t.test(redcities, bluecities, alternative="greater")
```

R Code: ANOVA

```
library(scales)

#############################################################################################
# Process the Data
#############################################################################################

cities <- read.csv("/Users/alicezhao/Dropbox/4-Fall 2013/1-Stat139/Stat 139 Final
    Project/data/communities.unnormalized.data.txt", header=F)

names(cities) <- c("county", "state", "community", "communityname", "fold",
    "population", "householdsize", "racepctblack", "racePctWhite", "racePctAsian",
    "racePctHisp", "agePct12t21", "agePct12t29", "agePct16t24", "agePct65up",
```

```
    "numbUrban", "pctUrban", "medIncome", "pctWWage", "pctWFarmSelf", "pctWInvInc",
    "pctWSocSec", "pctWPubAsst", "pctWRetire", "medFamInc", "perCapInc", "whitePerCap",
    "blackPerCap", "indianPerCap", "AsianPerCap", "OtherPerCap", "HispPerCap",
    "NumUnderPov", "PctPopUnderPov", "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
    "PctUnemployed", "PctEmploy", "PctEmplManu", "PctEmplProfServ", "PctOccupManu",
    "PctOccupMgmtProf", "MalePctDivorce", "MalePctNevMarr", "FemalePctDiv",
    "TotalPctDiv", "PersPerFam", "PctFam2Par", "PctKids2Par", "PctYoungKids2Par",
    "PctTeen2Par", "PctWorkMomYoungKids", "PctWorkMom", "NumIlleg", "PctIlleg",
    "NumImmig", "PctImmigRecent", "PctImmigRec5", "PctImmigRec8", "PctImmigRec10",
    "PctRecentImmig", "PctRecImmig5", "PctRecImmig8", "PctRecImmig10",
    "PctSpeakEnglOnly", "PctNotSpeakEnglWell", "PctLargHouseFam", "PctLargHouseOccup",
    "PersPerOccupHous", "PersPerOwnOccHous", "PersPerRentOccHous", "PctPersOwnOccup",
    "PctPersDenseHous", "PctHousLess3BR", "MedNumBR", "HousVacant", "PctHousOccup",
    "PctHousOwnOcc", "PctVacantBoarded", "PctVacMore6Mos", "MedYrHousBuilt",
    "PctHousNoPhone", "PctWOFullPlumb", "OwnOccLowQuart", "OwnOccMedVal",
    "OwnOccHiQuart", "RentLowQ", "RentMedian", "RentHighQ", "MedRent",
    "MedRentPctHousInc", "MedOwnCostPctInc", "MedOwnCostPctIncNoMtg", "NumInShelters",
    "NumStreet", "PctForeignBorn", "PctBornSameState", "PctSameHouse85",
    "PctSameCity85", "PctSameState85", "LemasSwornFT", "LemasSwFTPerPop",
    "LemasSwFTFieldOps", "LemasSwFTFieldPerPop", "LemasTotalReq", "LemasTotReqPerPop",
    "PolicReqPerOffic", "PolicPerPop", "RacialMatchCommPol", "PctPolicWhite",
    "PctPolicBlack", "PctPolicHisp", "PctPolicAsian", "PctPolicMinor",
    "OfficAssgnDrugUnits", "NumKindsDrugsSeiz", "PolicAveOTWorked", "LandArea",
    "PopDens", "PctUsePubTrans", "PolicCars", "PolicOperBudg", "LemasPctPolicOnPatr",
    "LemasGangUnitDeploy", "LemasPctOfficDrugUn", "PolicBudgPerPop",
    "ViolentCrimesPerPop")

#state.names <- c("AL", "AK", "T", "AZ", "AR", "CA", "T", "CO", "CT", "DE", "DC", "FL",
    "GA", "T", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD", "MA", "MI",
    "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ", "NM", "NY", "NC", "ND", "OH", "OK",
    "OR", "PA", "T", "RI", "SC", "SD", "TN", "TX", "UT", "VT", "VA", "T", "WA", "WV",
    "WI", "WY")
#statename <- c("AL", "AK", "T", "AZ", "AR", "CA", "T", "CO", "CT", "DE", "DC", "FL",
    "GA", "T", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD", "MA", "MI",
    "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ", "NM", "NY", "NC", "ND", "OH", "OK",
    "OR", "PA", "T", "RI", "SC", "SD", "TN", "TX", "UT", "VT", "VA", "T", "WA", "WV",
    "WI", "WY")

#cities$state.names <- state.names[cities$state]
# shift values (since many are0's) and log y value
shiftViolence <- cities$ViolentCrimesPerPop + 1
logshiftViolence <- log(shiftViolence)

#############################################################################################
# ANOVA BY STATE
#############################################################################################

statelogshift <- lm(data = cities, logshiftViolence~ factor(state))
anova(statelogshift)

#ASSUMPTIONS#################################################################################
###look at attached word doc
#Normality###
hist(cities$ViolentCrimesPerPop) #not normal. need to transform
```

```
hist(logshiftViolence)

#Equal Population Variance###
#Create vector, statecrimesvar, of all the variances of Violence Crimes Per Pop for each
    state
#no HI and MT
#statecrimes<-vector()
#for (i in as.character(unique(cities$state))){
#  statecrimes <-logshiftViolence[which(cities$state==i)]
#  statecrimes <-append(statecrimes, var(statecrimes))
#}

#########################
#Anova Test for subetting Top10 and Bottom40 by population vs. Separate Means
#########################

#setting top10, bottom40
cities$top10indic[cities$state != "CA" & cities$state != "TX" & cities$state != "NY" &
    cities$state != "FL" & cities$state != "IL" & cities$state != "PA" & cities$state !=
    "OH" & cities$state != "GA" & cities$state != "MI" & cities$state != "NC"] = 0
cities$top10indic[cities$state == "CA" | cities$state == "TX" | cities$state == "NY" |
    cities$state == "FL" | cities$state == "IL" | cities$state == "PA" | cities$state ==
    "OH" | cities$state == "GA" | cities$state == "MI" | cities$state == "NC"] = 1

#run anova
anova(lm(LogViolentCrimesPerPop ~ as.factor(state), data = cities),
    lm(LogViolentCrimesPerPop ~ as.factor(top10indic), data = cities))

#p value of 1.634*10^-7, thus we can say that separate means is a better model

#########################
#Anova Test for subetting Top10 and Bottom40 by population DENSITY vs. Separate Means
#########################

#Top 10 (New Jersey(NJ), Rhode Island, Massachusetts, Connecticut, Maryland, Delaware,
    New York, Florida, Pennsylvania, Ohio)
cities$top10DensIndic[cities$state == "NJ" | cities$state == "RI" | cities$state == "MA"
    | cities$state == "CT" | cities$state == "MD" | cities$state == "DE" | cities$state
    == "NY" | cities$state == "FL" | cities$state == "PA" | cities$state == "OH"] = 1
cities$top10DensIndic[cities$state != "NJ" & cities$state != "RI" & cities$state != "MA"
    & cities$state != "CT" & cities$state != "MD" & cities$state != "DE" & cities$state
    != "NY" & cities$state != "FL" & cities$state != "PA" & cities$state != "OH"] = 0

#run anova
anova(lm(LogViolentCrimesPerPop ~ as.factor(state), data = cities),
    lm(LogViolentCrimesPerPop ~ as.factor(top10DensIndic), data = cities))

#p value of 1.175*10^-7, thus we can say that separate means is a better model

#########################
#Anova Test for subetting Top10 and Bottom40 by Wealth vs. Separate Means
#########################
```

```
#Top 10 (Delaware, Minnesota, Virginia, New Hampshire, Massachussetts, Hawaii,
    Connecticut, Alaska, New Jersey, Maryland)
cities$top10Wealth[cities$state == "DE" | cities$state == "MN" | cities$state == "VA" |
    cities$state == "NH" | cities$state == "MA" | cities$state == "HI" | cities$state ==
    "CT" | cities$state == "AK" | cities$state == "NJ" | cities$state == "MD"] = 1
cities$top10Wealth[cities$state != "DE" & cities$state != "MN" & cities$state != "VA" &
    cities$state != "NH" & cities$state != "MA" & cities$state != "HI" & cities$state !=
    "CT" & cities$state != "AK" & cities$state != "NJ" & cities$state != "MD"] = 0

#run anova
anova(lm(LogViolentCrimesPerPop ~ as.factor(state), data = cities),
    lm(LogViolentCrimesPerPop ~ as.factor(top10Wealth), data = cities))

#p value of 1.175*10^-7, thus we can say that separate means is a better model

#########################
#Anova Test for subetting Top10 and Bottom40 by Education vs. Separate Means
#########################

#Top 10 Education (Minnesota, New York, New Hampshire, Virginia, New jersey, Vermont,
    Connecticut, Colorado, Maryland, Massachusetts)
cities$top10Educ[cities$state == "MN" | cities$state == "NY" | cities$state == "NH" |
    cities$state == "VA" | cities$state == "NJ" | cities$state == "VT" | cities$state ==
    "CT" | cities$state == "CO" | cities$state == "MD" | cities$state == "MA"] = 1
cities$top10Educ[cities$state != "MN" & cities$state != "NY" & cities$state != "NH" &
    cities$state != "VA" & cities$state != "NJ" & cities$state != "VT" & cities$state !=
    "CT" & cities$state != "CO" & cities$state != "MD" & cities$state != "MA"] = 0

#run anova
anova(lm(LogViolentCrimesPerPop ~ as.factor(state), data = cities),
    lm(LogViolentCrimesPerPop ~ as.factor(top10Educ), data = cities))

#p value of 1.196*10^-7, thus we can say that separate means is a better model

#########################
#Anova Test for subetting Top10 and Bottom40 by Single Parents vs. Separate Means
#########################

#Top 10 Single Parent(New Jersey, Montana, Idaho, Iowa, Minnesota, Nebraska, New
    Hampshire, North Dakota, Utah, Wyoming)
cities$top10Single[cities$state == "NJ" | cities$state == "MT" | cities$state == "ID" |
    cities$state == "IA" | cities$state == "MN" | cities$state == "NE" | cities$state ==
    "NH" | cities$state == "ND" | cities$state == "UT" | cities$state == "WY"] = 1
cities$top10Single[cities$state != "NJ" & cities$state != "MT" & cities$state != "ID" &
    cities$state != "IA" & cities$state != "MN" & cities$state != "NE" & cities$state !=
    "NH" & cities$state != "ND" & cities$state != "UT" & cities$state != "WY"] = 0

#run anova
anova(lm(LogViolentCrimesPerPop ~ as.factor(state), data = cities),
    lm(LogViolentCrimesPerPop ~ as.factor(top10Single), data = cities))

#p value of 1.211*10^-6, thus we can say that separate means is a better model
```

R Code: Visualizations

```
library(rworldmap)
library(RColorBrewer)
library(maptools)

## this example uses downloaded files
## to run it download the files
## and remove the comment symbols   #      from all the lines starting with a single
     #
## US states map downloaded from :
## http://www2.census.gov/cgi-bin/shapefiles2009/national-files

sPDF <- readShapePoly('./tl_2009_us_stateec.shp')
str(sPDF@data)

###################
## use mapPolys to map the sPDF
mapPolys(sPDF,nameColumnToPlot = "ALANDEC")
mapPolys(sPDF,nameColumnToPlot = "AWATEREC",mapRegion=North   America  )
###################
## join some other data to it
## education data downloaded from here as xls then saved as csv
## http://nces.ed.gov/ccd/drpcompstatelvl.asp
dataFile <- './sdr091A.csv'
dF <- read.csv(dataFile,as.is=TRUE)
str(dF)

# read data and define variables
cities <- read.csv("~/Dropbox/Stat 139 Final
    Project/data/communities.unnormalized.data.txt", header=F)
names(cities) <- c("county", "state", "community", "communityname", "fold",
    "population", "householdsize", "racepctblack", "racePctWhite", "racePctAsian",
    "racePctHisp", "agePct12t21", "agePct12t29", "agePct16t24", "agePct65up",
    "numbUrban", "pctUrban", "medIncome", "pctWWage", "pctWFarmSelf", "pctWInvInc",
    "pctWSocSec", "pctWPubAsst", "pctWRetire", "medFamInc", "perCapInc", "whitePerCap",
    "blackPerCap", "indianPerCap", "AsianPerCap", "OtherPerCap", "HispPerCap",
    "NumUnderPov", "PctPopUnderPov", "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
    "PctUnemployed", "PctEmploy", "PctEmplManu", "PctEmplProfServ", "PctOccupManu",
    "PctOccupMgmtProf", "MalePctDivorce", "MalePctNevMarr", "FemalePctDiv",
    "TotalPctDiv", "PersPerFam", "PctFam2Par", "PctKids2Par", "PctYoungKids2Par",
    "PctTeen2Par", "PctWorkMomYoungKids", "PctWorkMom", "NumIlleg", "PctIlleg",
    "NumImmig", "PctImmigRecent", "PctImmigRec5", "PctImmigRec8", "PctImmigRec10",
    "PctRecentImmig", "PctRecImmig5", "PctRecImmig8", "PctRecImmig10",
    "PctSpeakEnglOnly", "PctNotSpeakEnglWell", "PctLargHouseFam", "PctLargHouseOccup",
    "PersPerOccupHous", "PersPerOwnOccHous", "PersPerRentOccHous", "PctPersOwnOccup",
    "PctPersDenseHous", "PctHousLess3BR", "MedNumBR", "HousVacant", "PctHousOccup",
    "PctHousOwnOcc", "PctVacantBoarded", "PctVacMore6Mos", "MedYrHousBuilt",
    "PctHousNoPhone", "PctWOFullPlumb", "OwnOccLowQuart", "OwnOccMedVal",
    "OwnOccHiQuart", "RentLowQ", "RentMedian", "RentHighQ", "MedRent",
    "MedRentPctHousInc", "MedOwnCostPctInc", "MedOwnCostPctIncNoMtg", "NumInShelters",
    "NumStreet", "PctForeignBorn", "PctBornSameState", "PctSameHouse85",
    "PctSameCity85", "PctSameState85", "LemasSwornFT", "LemasSwFTPerPop",
```

```
    "LemasSwFTFieldOps", "LemasSwFTFieldPerPop", "LemasTotalReq", "LemasTotReqPerPop",
    "PolicReqPerOffic", "PolicPerPop", "RacialMatchCommPol", "PctPolicWhite",
    "PctPolicBlack", "PctPolicHisp", "PctPolicAsian", "PctPolicMinor",
    "OfficAssgnDrugUnits", "NumKindsDrugsSeiz", "PolicAveOTWorked", "LandArea",
    "PopDens", "PctUsePubTrans", "PolicCars", "PolicOperBudg", "LemasPctPolicOnPatr",
    "LemasGangUnitDeploy", "LemasPctOfficDrugUn", "PolicBudgPerPop",
    "ViolentCrimesPerPop")
cities$state <- as.character(cities$state)
crime.by.state <- ddply(cities, .(state), summarize, weighted.crime = sum(population *
    ViolentCrimesPerPop) / sum(population))

# make cartogram part 1
sPDF2 <- joinData2Map(crime.by.state
                    , nameMap = sPDF
                    , nameJoinIDMap = "STUSPSEC"
                    , nameJoinColumnData = "state")

#################
## plot one of the attribute variables
mapDevice()# to set nice shape map window

# make cartogram part 2
mapPolys(sPDF2, nameColumnToPlot = "weighted.crime",
        numCats = 50, catMethod = "fixedWidth",
        ylim = c(36, 36), xlim = c(-126, -66),
        colourPalette = brewer.pal(64, "Reds"),
        mapTitle = "\n \n \n \n \n \n (Population Weighted) Violent Crime per 1k
            Population"
        )
```