```
##################################
# LCS on EPA data: Analysis R-code...
##################################

# TOPICS:

# 1.     Data Input
# 2.     Quantile Plots using quantreg and splines...
# 3.     Using cor() and packages: RXshrink & LocalControlStrategy
# 4.     LCStrategy "Aggregate" Phase analyses...
# 5.     LCStrategy "Confirm" Phase example...
# 6.     LCStrategy "Explore" Phase termination...
#        LCStrategy "Reveal" Phase analyses...
# 7.        randomForest and PDPlot insights...
# 8.        A single, small Model-Based party Tree...
# 9.        A US Map showing LRCs by County...

# Bob Obenchain, PhD
# Risk - Benefit Statistics, 1006 Pebble Beach Dr.
# Clayton, CA, 94517 - 2211
# wizbob@att.net
# November 2022   --  http://localcontrolstatistics.org/

#############################################################################
#############################################################################

# 1.     Data Input
# -----------------
# Start by using the AnalysisFile.csv file to create an R data.frame named "data".
# It is essential for the "FIPS" variable to be named "fips" (i.e. lower case letters).

data <- read.csv(file="AnalysisFile.csv")
str(data)
# 'data.frame':   2973 obs. of  25 variables:
#  $ fips      : int  1001 1003 1005 1007 1009 1011 1013 1015 ...
#  $ C50       : int  1 2 3 1 2 4 5 1 1 5 ...
#  $ LRC50     : num  0.277 0.393 0.188 0.277 0.393 ...
#  $ County    : chr  "Autauga" "Baldwin" "Barbour" "Bibb" ...
#  $ State     : chr  "AL" "AL" "AL" "AL" ...
#  $ AACRmort  : num  382 294 325 399 427 ...
#  $ pmTOT     : num  8.86 7.56 7.62 9.13 9.19 ...
```

```
#  $ pmOA       : num  4.92 3.98 4.31 5.35 4.56 ...
#  $ pmDUST     : num  1.79 1.5 1.4 1.68 2.21 ...
#  $ pmSO4      : num  1.07 1.027 0.962 1.044 1.089 ...
#  $ pmNH4NO3   : num  0.517 0.389 0.434 0.529 0.783 ...
#  $ pmSOOT     : num  0.32 0.285 0.254 0.303 0.343 ...
#  $ pmSEAspry  : num  0.243 0.372 0.252 0.218 0.202 ...
#  $ pmOC       : num  2.77 2.2 2.42 3.03 2.52 ...
#  $ pmPOA      : num  0.788 0.579 0.785 0.774 0.771 ...
#  $ pmSOA      : num  4.13 3.41 3.53 4.58 3.79 ...
#  $ Avoc       : num  1.41 1.31 1.32 1.4 1.53 ...
#  $ Bvoc       : num  2.72 2.1 2.2 3.18 2.26 ...
#  $ PREMdeath  : num  9409 7468 8929 11742 9359 ...
#  $ ASmok      : num  0.191 0.168 0.215 0.199 0.197 ...
#  $ ChildPOV   : num  0.193 0.176 0.396 0.275 0.194 0.457 ...
#  $ IncomIEQ   : num  4.39 4.6 5.86 4.23 4.07 ...
#  $ Population: int  55416 208563 25965 22643 57704 10362 ...
#  $ NO2        : num  2.29 1.97 1.14 1.94 2.28 ...
#  $ O3         : num  28.6 28.5 28.8 28.9 31.5 ...

################################################################
# y-Outcome Variable is CDC "AACRmort" where AA => Age
#     Adjusted, C => Circulatory, R => Respiratory, and
#     mort => deaths per 100,000 County Residents.
# Primary "Exposure" e-Variable is EPA "Bvoc" => Biogenic
#     volatile organic compounds among Secondary Organic
#     Aerosols within PM2.5 (pmTOT) particulate matter.
################################################################

data9 <- subset(data, select = c(AACRmort, Bvoc, pmSO4, Avoc, PREMdeath,
                      ASmok, ChildPOV, IncomIEQ, pmSOA))
# The "data9" data.frame is our "Essential Subset" of 9 variables.

#################################################################################
#################################################################################

# 2.     Quantile Plots using quantreg and splines...
# ------------------------------------------------
# Create "Figure 1" ...using 2,973 US Counties
library(quantreg)
library(splines)
```

```
DF <- subset(data9, select = c(Bvoc, AACRmort))
o <- order(DF$Bvoc)
Bvoc <- DF$Bvoc[o]
AACRmort <- DF$AACRmort[o]

plot(Bvoc,AACRmort,xlab="Bvoc",ylab="AACRmort",type = "n", cex=.5)
points(Bvoc,AACRmort,cex=.5,col="black")
X <- model.matrix(AACRmort ~ bs(Bvoc, df=15))

fit <- rq(AACRmort ~ bs(Bvoc, df=15), tau=0.5, data=DF)
AACRmort.fit <- X %*% fit$coef
lines(Bvoc, AACRmort.fit, lwd=3, col="red")

fit <- rq(AACRmort ~ bs(Bvoc, df=15), tau=0.75, data=DF)
AACRmort.fit <- X %*% fit$coef
lines(Bvoc, AACRmort.fit, lwd=2, col="blue")

fit <- rq(AACRmort ~ bs(Bvoc, df=15), tau=0.25, data=DF)
AACRmort.fit <- X %*% fit$coef
lines(Bvoc, AACRmort.fit, lwd=2, col="blue")

fit <- rq(AACRmort ~ bs(Bvoc, df=15), tau=0.95, data=DF)
AACRmort.fit <- X %*% fit$coef
lines(Bvoc, AACRmort.fit, col="green3")

fit <- rq(AACRmort ~ bs(Bvoc, df=15), tau=0.05, data=DF)
AACRmort.fit <- X %*% fit$coef
lines(Bvoc, AACRmort.fit, col="green3")

leg <- c("Median (0.5) Fit","Quartiles (0.25/0.75)", "95% Band (0.025/0.975)")
legend("topright", legend=leg, lty=1, col=c("red","blue","green3"))

# Optional memory "clean-up"...
# rm(X, o, DF, leg, fit, Bvoc, AACRmort.fit, AACRmort)

##############################################################################
##############################################################################

# 3.     Using cor() and packages: RXshrink & LocalControlStrategy
# ---------------------------------------------------------------
options(digits=4)
```

```
cor(data9)
#                AACRmort  Bvoc        pmSO4   Avoc      PREMdeath       ASmok ChildPOV IncomIEQ pmSOA
#      AACRmort     1.0000
#      Bvoc         0.4589 1.0000
#      pmSO4        0.3521 0.5023       1.0000
#      Avoc         0.2489 0.5847       0.7228 1.00000
#      PREMdeath    0.6421 0.4217       0.2062 0.08896      1.00000
#      ASmok        0.6047 0.4622       0.3556 0.32707      0.67611       1.0000
#      ChildPOV     0.5524 0.4884       0.2082 0.11336      0.69932       0.6605 1.0000
#      IncomIEQ     0.3040 0.4163       0.1802 0.13933      0.41804       0.3800 0.5708 1.0000
#      pmSOA        0.4253 0.9440       0.6486 0.81956      0.33403       0.4595 0.3911 0.3507  1.0000
#
#      pmSOA is highly redundant with Bvoc and Avoc ...because pmSOA = Bvoc + Avoc
#      Pye et al. (2021) used pmSOA as their primary measure of air pollution.
options(digits=7)

# Next: Create Figure 2...
form7 <- AACRmort ~ Bvoc + pmSO4 + Avoc + PREMdeath + ASmok + ChildPOV + IncomIEQ
library(RXshrink)
# Perform Efficient (7-parameter) Ridge Shrinkage...
effobj <- eff.ridge(form7, data=data9, steps=50)
effobj$dMSE # 0.9996 0.9893 0.9966 0.6926 0.9804 0.9363 0.8902
#                                  ^^^^^^               ^^^^^^

plot(effobj, trace="coef", LP=7) # LP sets "Legend Position"

############################################################
# Apply "LocalControlStrategy" to the data9 "Essential Subset"
############################################################
library(LocalControlStrategy)

xvars <- c("Bvoc","Avoc","pmSO4","PREMdeath","ChildPOV","ASmok")

hclobj <- LCcluster(data9, xvars)
# Missing 3rd argyment ==> use Default clustering method: "ward.D"...

hclobj
# LCcluster object: Hierarchical Clustering for LC
# Data Frame input: data9
# Covariate X variables: Bvoc Avoc pmSO4 PREMdeath ChildPOV ASmok
# Cluster method  : ward.D
# Distance  : euclidean
```

```
# Number of objects: 2973

plot(hclobj) # Save as PDF, then "Crop" it to create: Fig03 Dendrogram...
             # Optional: Edit PDF with PowerPoint to Add a horizontal LINE defining 10 clusters...

# Save LC Strategy parameters to an Environment that can (and will) be Updated...

LCSe <- LCsetup(hclobj, data9, Bvoc, AACRmort)
# The Treatment variable [Bvoc] is an Exposure with 2972 levels.
# Only Local Rank Correlations (LRCs) can be formed Within Clusters.

ls.str(LCSe)
# aggdf : 'data.frame': 1 obs. of 4 variables:
# $ Label : chr "TEMP"
# $ Blocks : num 1
# $ LRCmean: num 0
# $ LRCstde: num 0
# boxdf : 'data.frame': 1 obs. of 2 variables:
# $ LCstat: num 0.474
# $ K : num 1
# Kmax : num 247
# LRCmax : num 0.474
# LRCmin : num 0
# NumLevels : int 2972
# pars : chr [1, 1:4] "hclobj" "data9" "Bvoc" "AACRmort"

##############################################################################
##############################################################################

# 4.     LCStrategy "Aggregate" Phase analyses...
# ----------------------------------------
# Save LRC distributions for 6 Numbers of Clusters...

mort010 <- lrcagg( 10, LCSe)
mort025 <- lrcagg( 25, LCSe)
mort050 <- lrcagg( 50, LCSe)      # Avg. Cluster Size: ~53 US Counties
plot(mort050, show="ecdf", LCSe)    # K50ecdf.pdf

mort075 <- lrcagg( 75, LCSe)      # Avg. Cluster Size: ~35 US Counties
mort100 <- lrcagg(100, LCSe)
mort200 <- lrcagg(200, LCSe)      # Avg. Cluster Size: ~13 US Counties
```

```
LCcompare(LCSe)                       # Fig06: This display favors use of 50 Clusters...

##############################################################################
##############################################################################
# 5.     LCStrategy "Confirm" Phase example...
# ---------------------------------------
system.time( conf050 <- confirm(mort050) )
#     user system elapsed
#     3.61  0.13  3.73  # About 4 seconds...

conf050
# confirm Object: Compare Observed and NULL Distributions of Local Effect-Sizes...
# Simulated NULL Distribution uses Random Clusterings of Experimental Units.
# Data Frame: data9
# Outcome Variable: AACRmort
# Treatment Factor: Bvoc
# Number of Replications: 100
# Number of Clusters per Replication: 50

# Number of Random NULL Local Effect-Sizes: 297300
#     Mean Observed Local Effect-Size = 0.1851901
#     Std. Dev. of Observed Effect-Sizes = 0.1855613
#     Mean Random NULL Effect-Size = 0.4665366
#     Std. Dev. of Random Effect-Sizes = 0.1106045

# Nonstandard Kolmogorov-Smirnov comparison of Discrete Distributions:
# Observed two-sample KS D-statistic = 0.7902893  <<< This is Gigantic! >>>

plot(conf050)                 # Fig04 display for "Confirm" phase...
# opar <- par(no.readonly = TRUE)
abline(v=0.62, lty=2, col=2) # Add Dotted Vertical line @ 62%
# par(opar))                                # i.e. @ ~0.30 on horizontal axis.

system.time( ksd050 <- KSperm(conf050, reps = 1000) )
# Simulation takes ~78 seconds.
# Default number of reps = 1000.
ksd050        # Implicit PRINT
#
#     KSperm: Simulated NULL Distribution of Kolmogorov-Smirnov statistics when given X-covariates
#             are IGNORABLE.
```

```
#     Data Frame: data9
#     Outcome Variable: AACRmort
#     Treatment Variable: Bvoc
#     Effect-Size estimates: Local Rank Correlations (LRCs)
#     Number of Random NULL D-statistics: reps = 1000
#
#     Number of Clusters per replication: 50
#     Observed Kolmogorov-Smirnov D-statistic = 0.7815103
#     Simulated NULL KS-D order statistics =
#
#     0.05048772 0.05057181 0.05213589 0.05423142 0.05498486 0.05521023 0.05606122
#     0.05753448 0.05786411 0.05804575 0.05839220 0.05847292 0.05953919 0.06113354
#     0.06113354 0.06137571 0.06171208 0.06218298 0.06218971 0.06242516 0.06309788
#     0.06356879 0.06382106 0.06462496 [25]     .............................   [984]
#     0.16733939 0.16736630 0.16906828 0.16976791 0.17055163 0.17096199 0.17510595
#     0.17592331 0.18392869 0.18424487 0.18967373 0.19260679 0.19505214 0.20474605
#     0.21069290 0.21642112
#     Simulated adjusted p-value for the Observed D-statistic: 0.001

"plKSp" <-
function(x, ...)  # Create custom "CONFIRM" plots...
{
  opar <- par(no.readonly = TRUE)
  on.exit(par(opar))
  Dvec <- x$Dvec obsD <- x$obsD
  xmax = max(max(Dvec), obsD)
  plot(ecdf(Dvec), verticals=TRUE, do.points=FALSE, ann=FALSE, col="gray70", lwd=2, xlim = c(0, xmax))
  abline(v=0, lty="solid", col="black")
  abline(v=obsD, lty="dashed", lwd=2, col="red")
  title(main = "LC Confirm Inference: Ignorable X-covariates?",
    ylab = "Cumulative Probability",
    sub = paste("Observed D =", round(obsD, 4), ", Simulated p-value =", x$pv.adj))
    if (x$Type == 1) title(xlab = "KS D-statistics for NULL LTDs")
    else title(xlab = "KS D-statistics for NULL LRCs")
}

# Use plKSp() to draw a "Half-Height" plot...
op <- par(mfrow=c(2,1))
plKSp(ksd050)     # Save as PDF & Crop to create Fig05...
par(op)
```

```
##############################################################
# Create data.frame for use in final "REVEAL Phase" analyses...
##############################################################

dataLRC <- reveal.data(mort050, clus.var="C50", effe.var="LRC50")
# C50 is dataLRC[,1] variable: Cluster Number in 1:50
# LRC50 is dataLRC[,2] variable: Local Rank (Spearman) Correlation #    ...constant within each cluster...
# write.csv(dataLRC, file="dataLRC.csv", row.names = FALSE)  # ...Save as a .csv file.

CS <- as.numeric(dataLRC$C50) - 0.5 # i.e. cell "centers" at # 0.5, 1.5, ..., 49.5 [rather than 1,2,...,50]
op <- par(mfrow=c(2,1))
hist(CS, main="Cluster Size Distribution", xlab="Cluster", breaks=51)   # CROP to create Fig07.pdf
par(op)

str(mort050) # List of 12
# $ hiclus : chr "hclobj" # $ dframe      : chr "data9"
# $ trtm    : chr "Bvoc"      # $ yvar    : chr "AACRmort" # $ K   : num 50    # $ actclust : int 50
# $ LRCtabl: 'data.frame':    50 obs. of 5 variables:
#     ..$ c : Factor w/ 50 levels "1","2","3","4",..: 1 2 3 4 ...
#     ..$ LRC: num [1:50] -0.6993 -0.414 -0.4036 -0.1344 ...
#     ..$ w : int [1:50] 12 35 39 36 78 37 112 31 13 28 ...
#     ..$ LAO: num [1:50] 305 454 255 373 340 ...
#     ..$ PS : num [1:50] 1.171 2.075 0.519 0.601 0.747 ...
# $ LRCdist :'data.frame': 2973 obs. of 5 variables:
#     ..$ c.50: Factor w/ 50 levels "1","2","3","4",..: 1 1 1 ...
#     ..$ ID : int [1:2973] 1 2 3 4 5 6 7 8 9 10 ...
#     ..$ y : num [1:2973] 231 246 247 248 278 ...
#     ..$ t : num [1:2973] 1.54 1.488 1.402 1.243 0.999 ...
#     ..$ LRC : num [1:2973] -0.699 -0.699 -0.699 -0.699 ...
# $ infoclus  : int 50 # $ infounits : int 2973
# $ LRCmean : num 0.185 # $ LRCstde : num 0.186 # - attr(*, "class")= chr "lrcagg"

LRC50 <- mort050$LRCdist[,5]
summary(LRC50)
#        Min.       1st Qu.    Median       Mean      3rd Qu.       Max.
#     -0.69930  0.07836 0.20551     0.18519     0.29197     0.60000

op <- par(mfrow=c(2,1))
hist(LRC50, breaks = 60, xlim = c(-0.70,0.60), col = "lightgray",
        main = "LRC Distribution", xlab = "LRCs")     # CROP to create Fig08.pdf...
par(op)
```

```
################################################################################
################################################################################
#        LCStrategy: "Reveal" Phase analyses...
# 7.        randomForest and PDPlot insights...

library(randomForest)             # randomForest 4.7-1
library(randomForestExplainer)        # method +.gg, from ggplot2

# Define a randomForest model formula...
form.rF <- LRC50 ~ ASmok + ChildPOV + PREMdeath + Bvoc + pmSO4 + Avoc + AACRmort + IncomIEQ
set.seed(12345)
forest <- randomForest(form.rF, data = dataLRC, localImp = TRUE)
forest
# Type of random forest: regression
# Number of trees: 500
# No. of variables tried at each split: 2
#     Mean of squared residuals: 0.01288983
#           % Var explained: 62.55

plot(forest, log="y")        # Plot NOT displayed or discussed in our paper...

imp <- importance(forest)
impvar <- rownames(imp)[order(imp[, 1], decreasing=TRUE)]

imp
#            %IncMSE IncNodePurity
# ASmok     67.85752     16.825828
# ChildPOV  60.87109     16.644114
# PREMdeath 55.81210     11.599524
# Bvoc      45.44045     16.404675
# pmSO4     40.58580     11.837301
# Avoc      25.62735     14.601100
# AACRmort  18.19946      5.042794
# IncomIEQ  16.70957      5.224683

impvar
# [1] "ASmok"     "ChildPOV"  "PREMdeath" "Bvoc"
# [5] "pmSO4"     "Avoc"      "AACRmort"  "IncomIEQ"

op <- par(mfrow=c(2,1)) # Half-Height Plots...
```

```
for (i in seq_along(impvar)) {
VXY <- partialPlot(forest, data, impvar[i], lwd=2, xlab=impvar[i],
           main = paste("Partial Dependence on", impvar[i]))

V2 <- data.frame(cbind(VXY$x,VXY$y))
names(V2) <- c("x","y") # ...where V2 is ordered (increasing) on x

LO <- loess(y ~ x, V2, span=0.75)   # Default Wide Span...
lines(y=LO$fitted, x=V2$x, lwd=2, col="red")
scan() # ...Wait for user to press ENTER key after each (Half-Height) Plot (i.e. 8 times, total)...
}

# Enter this final command only after all above plots have been saved...
par(op)

################################################################################
################################################################################
#        LCStrategy: a "Reveal" Phase analysis...
# 8.        A single, small Model-Based "party" Tree...
#                           -----------

library(party)
set.seed(13254)
fmLRC <- mob(LRC50 ~ ASmok | Bvoc, data = dataLRC, model = glinearModel,
                  control = mob_control(minsplit=100))    # i.e., Require at least 100 US Counties per
node...

plot(fmLRC, main="MOB Conditional Tree for LRCs")          # Save as Fig19.pdf...

coef(fmLRC) # Data for first 3 columns of Table 4...       # Node | Intercept | ASmok-coefficient
#     ============================
#     4 -0.004109189 1.0990209
#     5 -0.131834114 1.4210621
#     ============================
#     7     0.336231566 -1.2435042
#     8     0.124474517 0.2902049
#     ============================
#     10 0.459140028 -1.0135984
#     12 0.868234130 -3.2115264
#     13 1.317113175 -5.0983352
```

```
##############################################################################
##############################################################################

# 9.        Create a US Map showing LRCs by County (using Shades of "Blue")
#-----------------------------------------
library(leaflet)
library(tidyverse)
library(ggmap)
library(leaflet.extras)
library(ggplot2)
library(maps)
library(mapproj)
library(mapdata)
library(socviz)
library(dplyr)
library(usmap)

county_full <- left_join(county_map, county_data, by = "id")
data <- read.csv(file="AnalysisFile.csv") # as on page 2...

county_full <- left_join(county_full, data, by = "fips")

summary(county_full$LRC50)
#      Min.  1st Qu. Median    Mean  3rd Qu.   Max.    NA's
# -0.699    0.078 0.206 0.180 0.292 0.600 18285

quantile(county_full$LRC50, probs = seq(0,1,0.05), na.rm=T)
#     0%    5%    10%   15%   20%   25%   30%
#     -0.69930070 -0.06845054 -0.04880924 0.02752081  0.06856302  0.07835954  0.12423898
#     35%   40%   45%   50%   55%   60%   65%
#     0.13609696  0.14799154  0.18823837  0.20551020  0.22264808  0.24595829  0.26742681
#     70%   75%   80%   85%   90%   95%   100%
#     0.27685656  0.29197132  0.29495362  0.30256336  0.39125085  0.51057641  0.60000000

x <- county_full$LRC50
z <- cut(x, breaks = c(-0.6, 0.0, 0.12, 0.2, 0.295, 0.39, 0.6))
summary(z)
# (-0.6,0] (0,0.12] (0.12,0.2] (0.2,0.295] (0.295,0.39] (0.39,0.6]      NA's
#     21314 28836 29258 62144 9366  21446 19018
levels(z) <- c("1","2","3","4","5","6")
```

```
str(z)  # Factor w/ 6 levels "(-0.6,0]","(0,0.12]",..: 4 4 4 4 4 4 4 ...

county_full$pop_dens6 <- z

p <- ggplot(data = county_full, mapping = aes(x = long, y = lat, fill = pop_dens6, group = group))

p1 <- p + geom_polygon(color = "gray90", size = 0.05) + coord_equal()

p2 <- p1 + scale_fill_brewer(palette="Blues",
        labels = c("<.00","0.00-0.12","0.12-0.20","0.20-0.29","0.29-0.40","> 0.40"))

p3 <- p2 + labs(fill ="Local Rank Correlation")

p3    # Display LRC numerical "shades" by County on the US map...
    # Save as PDF and CROP blank space at both top & bottom...

##################################### END ########################################
##################################################################################
```