LocalControlStrategy in R

Unsupervised, Nonparametric Methods of Adjustment for Treatment Selection Bias and Confounding in Cross-Sectional Studies.

LCstrategy_in_R.pdf, Version 1.3.2, January 2019

Bob Obenchain
Principal Consultant, Risk Benefit Statistics LLC
5176 Upperwood Court, Indianapolis, IN 46250-1776
wizbob@att.net **http://localcontrolstatistics.org**

This **LC Strategy** user's guide defines syntax and illustrates use of functions that perform calculations and provide visualizations using the **LocalControlStrategy** **R**-package ...an unsupervised, nonparametric approach based upon clustering or "matching" of experimental units within **Cross-Sectional Data.**

Researchers with **Longitudinal** observational data (case-control studies and survival analyses) can use the "NearestNeighbor" and "CompetingRisks" functions from CRAN-package **LocalControl.**

LC Strategy "aggregation" methods create ***x***-Covariate distributions that are more "balanced" by clustering experimental units (patients) in ***x***-Space. The units within each resulting local BLOCK are "relatively well-matched" at baseline. Frankly, data from "randomized" studies tend to be over-valued; the exact distributional balance that is then "expected" in theory may not be even be approximately good in actual practice! Besides, randomized designs are less powerful than blocked designs! LC Strategy can literally "design" better balance into your **Analyses** of unbalanced data!

Note specifically that, while exact "frequency" balance (such as some **fixed ratio** of "new" to "control" treated units within each BLOCK) is rarely achievable, LC Strategy produces unbiased estimates of local effect-sizes by weighting experimental units inversely proportional to their local sample size. Unfortunately, some BLOCKs can become "uninformative" because they contain only "new" units or only "control" units. In any case, each estimate within the full ***distribution*** of "estimable" effect-sizes represents a relatively ***"fair"*** apples-to-apples comparison (Lopiano, Obenchain & Young, 2014).

Table of Contents

Page

1. Introduction 2

2. The Four Phases of Local Control Strategy 5

3. Clustering and the Fundamental (Factoring) Theorem of Propensity Scoring 8

4. LRC Example: Lung Cancer Mortality and Indoor Radon Exposure 11

`LCcluster()`	Compute the Hierarchical Clustering Dendrogram for given X Covariates	13
`LCsetup()`	LC Parameter Settings: Specify the y-Outcome and the t-Treatment or Exposure Variable	14
`lrcagg()`	Compute Local Rank Correlations (LRCs) for a specified value of **K** = Number of Clusters	15
`ivadj()`	Compute IV Local Average Outcomes (LAOs) for a specified value of **K** = Number of Clusters	16
`LCcompare()`	Graphically Summarize Sensitivity of LRC Distributions across several choices for Number of Clusters (starting with **K** = 1)	17
`confirm()`	Compare the observed LRC Distribution for a given **K** with the **NULL** distribution from purely Random Permutations when ***x***-confounders are assumed to be IGNORABLE	18
`KSperm()`	Simulate an adjusted p-value for the Observed Kolmogorov-Smirnov D-statistic when all ***x***-confounders are IGNORABLE	20
`reveal.data()`	Prepare for LC REVEAL Analyses: Append the LRC estimates and corresponding Cluster IDs for a given value of **K** to a copy of the original data.frame specified in **`LCsetup()`.**	22

5. LTD Example: Analysis of 6-Month Survival in Plasmode PCI Data 24

`ltdagg()`	Local Treatment Differences (LTDs) for given **K**	26

6. SUMMARY - and - Choice of Clustering Method 33

7. References 35

8. Syntax for `LocalControlStrategy` R functions 38

*** *** *** *** *** *** *** *** *** *** *** *** ***

1. Introduction.

The ***LocalControlStrategy-package*** provides **R** functions that implement rather new methods for analysis of observational data that are "nonparametric" and use "unsupervised learning." In other words, the methods implemented here are *quite different from* traditional fitting of parametric, regression-like models ...models that tend to make *many* assumptions, rather *strong* assumptions, and even "*wrong*" or "*unrealistic*" assumptions (possibly yielding either over-fitting or over-simplification!)

An inherent feature (and potential problem) with parametric models is that they are traditionally used to simultaneously not only ***estimate*** all sorts of "effects" from observed data but also to ***predict*** unobserved (or unobservable) results with the very same equation(s). Such predictions may be either *interpolations* within regions where experimental units (patients) are sparse or else *extrapolations* to regions where no data are currently available ...and perhaps never will be.

Local Control Strategy focuses almost exclusively upon methods that "***cluster***" or "***match***" experimental units on currently available data (Stuart, 2010). Specifically, patients are objectively formed into "subgroups" or "blocks" within the ***x*-space** defined by their observed baseline ***x***-Covariate characteristics. Then, by focusing on ***local estimation*** (within individual "blocks"), both interpolation and extrapolation can be ***controlled.*** This control can typically be "***strengthened***" simply by making the number, **K**, of "blocks" larger. At least the *average* Size of "blocks" will then be smaller!

Local treatment/exposure ***effect-size estimates*** are defined using the data *within* "blocks" of experimental units and are deliberately chosen to be rather simple *summary statistics*, such as U-statistics (Hoeffding 1948). Besides specifying numeric baseline ***x***-Covariate characteristics, researchers wishing to apply LC strategy must also specify a ***t***-treatment indicator (or a continuous, numeric measure of exposure) as well as a numeric ***y***-Outcome variable that quantifies potential effects (or costs) associated with treatment or exposure. The two types of *effect-size estimates* computed by functions within the **`LocalControlStrategy-package`** are then:

LTDs = **Local Treatment Differences**. These are *within-cluster* measures of Average Treatment Effect (ATE) of the form "new" minus "control" when ***t*** is binary.

or

LRCs = **Local Rank Correlations**. These are also *within-cluster* summary statistics. They are fully standardized measures of the ***slope*** of the fitted line in the "local" regression of ***y***-Outcomes onto continuous ***t***-exposure levels ...where observed numerical values are replaced by their within-cluster ***ranks***.

In spite of their high demand for computing power, "local" approaches to adjustment for treatment selection bias and confounding end up offering two main advantages over traditional ***supervised*** (regression-like) methods…

[1] At least when the number of clusters is relatively large (so that the number of patients within most clusters is relatively small), there is **no need to formally test** whether patients are relatively "well matched" on their baseline ***x***-characteristics within clusters. Clustering of patients on their ***x***-Characteristics has assured that almost any model for prediction that is relatively smooth across ***x***-Space would confirm that any within-cluster comparisons of treatment outcome differences are relatively "fair" comparisons (Lopiano, Obenchain & Young 2014).

[2] Results from clustering lend themselves well to use of ***graphical visualization*** and ***sensitivity analysis*** techniques. This allows researchers to literally "***see***" what they are doing! For example, once a full "hierarchical clustering" Dendrogram (tree) has been

constructed, displays using alternative numbers of clusters can be generated quickly. Thus clustering approaches can provide not only fundamental, robust (non-parametric) insights but also highly relevant information about sensitivity of results to LC "tuning parameter" settings. Furthermore, the resulting graphical displays can dramatically illustrate how traditional parametric modeling approaches (such as simultaneous equations, latent variables and hierarchical models) tend to emphasize some aspects of the data while de-emphasizing other aspects that could be equally important.

Thanks to the considerable computing and graphical display power of modern workstations, clustering / matching methods for analysis of observational data are becoming more and more practically useful. This enables researchers with larger and larger datasets to both **ask & answer** key questions, like: "What is the full range of quantitative treatment effect-size estimates supported by the available data?" Or: "Which patient subsets tend to have extreme outcomes?"

Additionally, LC strategy is fully compatible with both ***"propensity scoring"* (PS) *methods*** [see Section §3 below] as well as with ***cluster-based "instrumental variable" (IV) adjustment methods*** (McClellan, McNeil and Newhouse, 1994). Both of these approaches can adjust for treatment selection bias, characterized by imbalance in patient baseline characteristics between treatment groups (study arms, cohorts) in either nonrandomized or poorly randomized studies.

Thus, in addition to the **LTDs** and **LRCs** introduced above (page 3), the **`LocalControl-Strategy-package`** also provides functions for display of:

Local Average Outcomes (LAOs) when ***x***-confounders are assumed to be ***Instrumental Variables***.

LAO plots, displayed via **`plot(ivadj())`**, show how within-cluster ***y***-Outcome averages vary across clusters. Here, the horizontal coordinate of a cluster is either [i] an ***observed Propensity Score*** (local fraction of units receiving the "new" rather than the "control" treatment), or else [ii] a ***Relative Exposure*** defined on [0, 1]. These analyses are valid only when all ***x***-Covariates used to define clusters are assumed to be ***Instrumental Variables*** which effect y-outcomes at most ***indirectly*** ...i.e. only through treatment choice or exposure level. [Use of ***x***-Covariates that measure disease severity or patient frailty is then questionable; such characteristics typically do have ***direct effects*** on y-Outcomes, thereby invalidating LAO plotting approaches.]

"Unsupervised Propensity Scoring" was my original designation (deprecated **R**-package **`USPS`**) for all facets of ***Local Control Strategy***. Unsupervised methods typically use jargon from literature on artificial intelligence and data mining; see Barlow(1989). Clustering methods proceed without receiving any "hints" from the designated ***y***-Outcome measure or the ***t***-treatment / exposure level to help "guide" formation of well-matched subgroups of experimental units within ***x*-Space**. In addition to clustering, multivariate probability density estimation is another well-known example of an unsupervised statistical method.

The clustering / matching approaches implemented in **`LocalControlStrategy`** are somewhat like the early suggestions of Rubin(1980). Unlike "supervised" (regression-

like) methods that tend to be computationally efficient, genuinely "local" methods are sufficiently computer intensive that they were rather impractical before the advent of modern hardware and open-source statistical software.

Some supervised methods do have to resort to numerical ***search methods*** over a p-dimensional space of parameters (e.g. estimation of a logit or probit regression model) rather than use a closed form expression (such as that for ordinary least squares estimates in a linear regression model.) On the other hand, some unsupervised methods need to compute all $n \times (n-1)/2$ pair-wise comparisons between experimental units to "match" all study subjects. The resulting increases in computing time and memory allocation due to use of unsupervised (Non-Polynomial Hard) methods can be enormous. Besides, clustering results are typically highly sensitive to user choice of similarity metric and/or clustering algorithm.

The computing algorithms discussed here are written in a dialect of Version 3 of the "S" language that is processed by the **R** interpreter, available for download from https://cran.r-project.org/.

2. The Four Phases of Local Control Strategy

All four phases of LC Strategy are introduced and briefly outlined here in Section §2. Together, the initial three phases constitute *Nonparametric Preprocessing* of observational data via *Systematic Sensitivity Analyses*. The decision to either continue accumulating information or else to transition to a final, ultimate **Reveal** phase of predictive analyses resides within the third LC Phase: **Explore**.

Phase One: Aggregate

Objective: Robustly Estimate LTD, LRC or LAO Effect-Sizes within specified subgroups of experimental units and display their *Distribution* across these same subgroups.

Actions: For a specified choice of (a) which ***x***-vector components to use, (b) which dissimilarity ***x***-metric and clustering / matching algorithm to use, and (c) what number, **K**, of clusters (blocks of relatively well-matched units) to form, display the resulting (frequency weighted) across-block distribution of LTD, LRC or LAO effect-size estimates.

Aggregation Phase support from LocalControlStrategy-package functions:

```
hclobj <- LCcluster(dframe, xvars, ...)
      ...designate which xvars are to be used in clustering.
      ...name one of 8 possible clustering algorithms.
      ...save resulting clustering Dendrogram (tree).

LCe <- LCsetup(hclobj, dframe, trex, yvar)
```

```
      ...designate trex variable (binary treatment or exposure).
      ...designate yvar variable (y-outcome effect or cost).
      ...save environmental output object from LCsetup().

ltdobj <- ltdagg(K, LCe, ...)
lrcobj <- lrcagg(K, LCe, ...)
ivobj  <- ivadj(x)
      ...save output object for K = Number of Clusters requested.
```

Phase Two: Confirm

Objective: Eliminate Aggregations that Fail to yield clearly "Visible" *Covariate Adjustment*.

Actions: Pretend that the ***x***-confounders used in clustering are actually *ignorable*. Repeatedly permute the (***y***-outcome, ***t***-exposure) pairs observed for individual experimental units across the same number of clusters (blocks) of the same sizes as those observed. This simulates purely random reassignment of experimental units to blocks, this time ignoring the numerical values of all ***x***-confounder vectors. Next, compare this unadjusted NULL (random permutation) local effect-size distribution with the observed LTD or LRC effect-size distribution from the current ***x***-based aggregation. Eliminate "uninteresting" aggregations where the observed distribution of local effect-size estimates is ***not clearly different*** in location, spread and/or shape from the purely-random permutation NULL distribution.

Confirm Phase support from LocalControlStrategy-package functions:

```
confobj <- confirm(x, ...)
      ...where x is a ltdagg(K,e) or lrcagg(K,e) output object.
KSpobj <- KSperm(x, ...)
      ...where x is a  confirm() output object.
```

Phase Three: Explore

Objective: Decide whether to Continue or to Terminate further attempts to generate and accumulate *interesting* alternative local effect-size distributions.

Actions: Accumulate information on sensitivity of local effect-size distributions to the three primary LC **Aggregation** *parameter settings*: (a), (b) and (c) of Phase One. How stable is the location, spread and shape of LTD or LRC effect-size distributions resulting from changes to the above three settings?

Explore Phase support from LocalControlStrategy-package functions:

```
LCcompare( LCe )  ...where LCe is the name of the environment
                              object output by LCsetup().
```

```
plot(ltdagg(), LCe), plot(lrcagg(), LCe) and/or plot(ivagg()).
```

Phase Four: Reveal

Objective: Determine whether an *interesting distribution* of local effect-size estimates appears to be either ***truly Heterogeneous*** (consists of ***predictable fixed-effects***) or ***mostly Random*** (consists mainly of ***unpredictable deviations*** from any Exposure Main-Effect.)

Actions: Researchers are released from any local / nonparametric restrictions on their attempts at global prediction of Local Effect-Size estimates from any of the "interesting" LTD or LRC distributions that have emerged from the first three *Nonparametric Preprocessing* phases of LC strategy.

Reveal Phase `support from LocalControlStrategy-package functions:`

```
outdf <- reveal.data(x, clus.var="Clus", effe.var="eSiz")
      ...where x is a ltdagg(K,e) or lrcagg(K,e) output object.
      ...outdf also contains the xvars used by LCcluster().

plot(ivadj(x, ...))
      ...where x is a ltdagg(K,e) or lrcagg(K,e) output object.
      ...displays both linear lm() and smooth.spline() fits.
```

Stan Young and I have considerable personal experience applying the above Four-Phase LC Strategy over the last 7+ years. Stan initially proposed performing regressions within clusters when the treatment is an exposure (pm2.5, ozone, radon, etc.) We have found that LC Strategy generally performs remarkably well; it has provided key insights into a wide variety of observational datasets that suffer from treatment selection bias and baseline ***x***-Covariate confounding.

When outcomes researchers with rather diverse perspectives collaborate in hope of ***developing a common (shared)*** "consensus view," team-findings within the LC **Explore Phase** of "sensitivity analyses" tend to be particularly helpful.

Perhaps today's buzz about "data science" has resulted because more-and-more aspiring professionals with strong computing skills and keen practical insights have (somehow) helped data "owners" advance their business goals in new ways. On the other hand, development of ***truly valid*** ("scientific") *principles* and *methods* for analysis of *observational data* would almost certainly be greatly expedited if much more "privately held" data were made ***truly public.*** Having "bigger and better" data for more future researchers to "practice on" could be a distinct plus for advancement of statistical science!

Government officials and regulators traditionally insist upon ***privacy*** of medical records to prevent health insurance discrimination. Yet, when asked, many patients with degenerative diseases (especially those with cancer) are apparently eager to contribute their de-identified health data for

"research use." "Local" methods help preserve patient confidentially; LTDs or LRCs for patients with similar ***x***-characteristics are apparently considered publishable whenever the subgroup size exceeds 10 (Standard Form CMS-R-0235L, 2012.)

3. Clustering and the Fundamental "Factoring" Theorem of Propensity Scoring

Our basic notation for variables will be slightly different from that used by Rosenbaum and Rubin (***Biometrika*** 1983 and ***JASA*** 1984)…

y = observed outcome variable(s)
$\boldsymbol{x}$ = observed baseline patient characteristic(s), covariate(s) and/or instrumental variable(s)
t = observed treatment assignment/choice (either 0 or 1) or exposure level (continuous)
$\boldsymbol{u}$ = unobserved explanatory variable(s) and unmeasured confounders

Note that unobserved ***u*** variables may provide unknown, causal effects on outcomes, y. In statistical/econometric models, it is the existence of ***u*** variables (as well as uncertainty in measuring y and x variables) that necessitates inclusion of "error terms" in parametric models.

NOTE: Patient level genome information is mostly a gigantic ***u***–vector today (2018.) More and more of this sort of information may soon be routinely used as ***x***-variables.

Rosenbaum and Rubin (**R&R**) defined Propensity Scores only for the case where treatment choice is binary: $t_i = 1$ (*new*) or $t_i = 0$ (*control*). The propensity for choice "new" for (or by) the i^{th} patient is formulated as being a function of only his/her given baseline ***x***-characteristics…

$$PS = p(x_i) = Pr(\, t_i = 1 \mid x_i \,) = E(\, t_i \mid x_i \,). \qquad [1]$$

In words, the true propensity score of a patient is the conditional probability that he/she will receive treatment number one given his/her vector of observed, baseline characteristics, x. Thus "true" propensities are (conditional) probabilities, which are numbers between zero and one, inclusive.

> In many practical applications, only the rank orders of (estimated) propensity scores are needed. In this sense, any monotone transformation of a set of propensity scores are another set of propensity scores "equivalent" to the first set. For example, when only one (univariate) x variable is found to be predictive of treatment choice/assignment, that single x variable may (itself) be called a propensity score.
>
> Apparently, one requirement of the above PS formulation is that each patient receives one and only one treatment. In other words, the standard formulation may not apply to treatment of chronic conditions where a cross-over design accesses y-Outcome information on two (or more) treatments applied during sequential periods to the same patient …separated by adequate "wash-out" periods.

Here, we will not attempt any sort of measure-theoretic proofs of propensity concepts. After all, the two original **R&R** propensity publications (***Biometrika*** 1983 and ***JASA*** 1984) may only make complete sense under (Rubin's) "Potential Outcomes" framework, initially outlined in Holland

(1986). Furthermore, the simplified notation used in these two **R&R** publications appears to treat ***x*** as being discrete and clearly assumes that ***t*** has only 2 levels. Following this notational "tradition," we will not worry here about mathematical details for cases where components of *x* have continuous distributions or even when *t* can have more than 2 levels. In fact, we wish to address observational situations here where treatment ***t***-levels are assumed to ***not*** be "strongly ignorable" given ***x*** [***Biometrika*** (1983), Section §1·3].

In their ***JASA*** (1984) discussion of the "fundamental theorem" of propensity scoring, **R&R** outline a simple argument showing that the conditional distribution of baseline patient ***x***-characteristics for given value of propensity {p(***x***) of equation [1]} must be statistically independent of the corresponding conditional treatment choice. Conceptually, this simple argument implies that the joint distribution of *x* and *t* given p must ***factor*** as follows: Pr(***x***, *t* | p) = Pr(***x*** | p) Pr(*t* | p).

A simple proof has four sequential steps, represented below in equations [2] through [5]:

$$\begin{aligned} \Pr(\boldsymbol{x}, t \mid p) &\equiv \Pr(\boldsymbol{x} \mid p)\,\Pr(t \mid \boldsymbol{x}, p) && [2]\\ &= \Pr(\boldsymbol{x} \mid p)\,\Pr(t \mid \boldsymbol{x}) && [3]\\ &= \Pr(\boldsymbol{x} \mid p) \text{ times } p \text{ or } (1-p) && [4]\\ &= \Pr(\boldsymbol{x} \mid p)\,\Pr(t \mid p) && [5] \end{aligned}$$

PROOF: The factoring result given in line [2] follows from the very definition of conditional probability. In line [3], one notes that conditioning upon both p = p(*x*) and *x* cannot contain information not provided by *x* alone. In line [4], one notes simply that Pr(*t* | *x*) must be either p(*x*) or [1 – p(*x*)]. Finally, in line [5], one concludes that, since Pr(*t* | *x*) must be either p or (1–p), then Pr(*t* | *x*) must be the same as Pr(*t* | p).

In other words, ***x*** and *t* are, necessarily, conditionally independent variables given the propensity score, p = Pr(*t* = 1 | ***x***). This is really a very simple theorem in statistics / probability that requires only rather weak assumptions. In fact, the real "problem" in applications is simply that the functional form of the true PS is usually unknown and, thus, needs to be estimated from data!

A "practically important" quibble about basic "propensity" terminology:

> Pr(***x*** | p) could be called the local "Blocking" Factor. All experimental units (receiving either *new* or *control* treatment) thus have the very same *x*-distribution whenever they have the same *scalar value* of p.

> Pr(*t* | p) is the local frequency "Balance" Factor. For all experimental units with the same *scalar value* of p, *t* = 1 (*new* treatment) occurs with probability p, while *t* = 0 (*control* treatment) occurs with probability 1 – p .

Note that true propensity values really do ***BLOCK*** experimental units, but true propensities are rarely known values ...except when analyzing data from truly randomized experiments. In propensity based analysis of observational data, propensity is typically estimated using parametric supervised learning (say, logistic regression). Unfortunately, estimated propensities are not

guaranteed to have the desirable features of true propensities; for example, the assumed PS model can be wrong!

"Blocking" and (frequency) "Balancing" are two distinctly different, fundamental concepts in classical Design of Experiments. Over the last 35 years, the propensity terminology introduced by **R&R** in their 1983 and 1984 papers appears to have "merged" traditional blocking and balancing concepts together. Analyses of observational data are now designed to "balance baseline ***x***-covariate distributions across the *new* and *control* treatment cohorts".

Here, I have simply called Pr(***x***, *t* | p) = Pr(***x*** | p) Pr(*t* | p) a "factoring" theorem. This terminology is unquestionably reasonable; it describes exactly what the equation literally says. However, for general use in practical applications, this equation may potentially be much more meaningfully described as attempting to "block" or "match" subgroups of experimental units in ***x***-space.

Cluster Membership is an asymptotic "Factoring Score."

Now let ***x*** denote a vector of baseline confounder values, let *t* denote ***either*** a binary treatment or continuous exposure level indicator, and let ***C*** denote a cluster (collection) of confounder ***x***-vectors that includes a specific given ***x***-vector and which was formed without reference to any information about the *t* (or *y*) characteristics of any and all experimental units. With Pr(|) again denoting conditional probability, we further write:

$$\Pr(\boldsymbol{x}, t \mid \boldsymbol{C}) \equiv \Pr(\boldsymbol{x} \mid \boldsymbol{C})\,\Pr(t \mid \boldsymbol{x}, \boldsymbol{C}) \qquad [6]$$

$$= \Pr(\boldsymbol{x} \mid \boldsymbol{C})\,\Pr(t \mid \boldsymbol{x}) \quad \text{because } \boldsymbol{x} \text{ is within } \boldsymbol{C}, \text{ and } \boldsymbol{C} \text{ does not depend upon } t \qquad [7]$$

$$\rightarrow \Pr(\boldsymbol{x} \mid \boldsymbol{C})\,\Pr(t \mid \boldsymbol{C}) \quad \text{in the limit as cluster } \boldsymbol{C} \text{ shrinks to contain only } \boldsymbol{x}. \qquad [8]$$

Note that:

- As in equation [2], relationship [6] also follows from the basic definition of conditional probability.
- Whenever cluster formation depends only upon the available ***x***-vectors and *thus does not depend in any way upon treatment, **y**, or exposure*, ***t***, the right-hand side of expression [6] can then be rewritten as [7].
- In the limit as the ***x***-space maximum "diameter" of cluster ***C*** shrinks to zero, the given ***x*** then becomes the only interior point of ***C***, and expression [8] holds asymptotically.

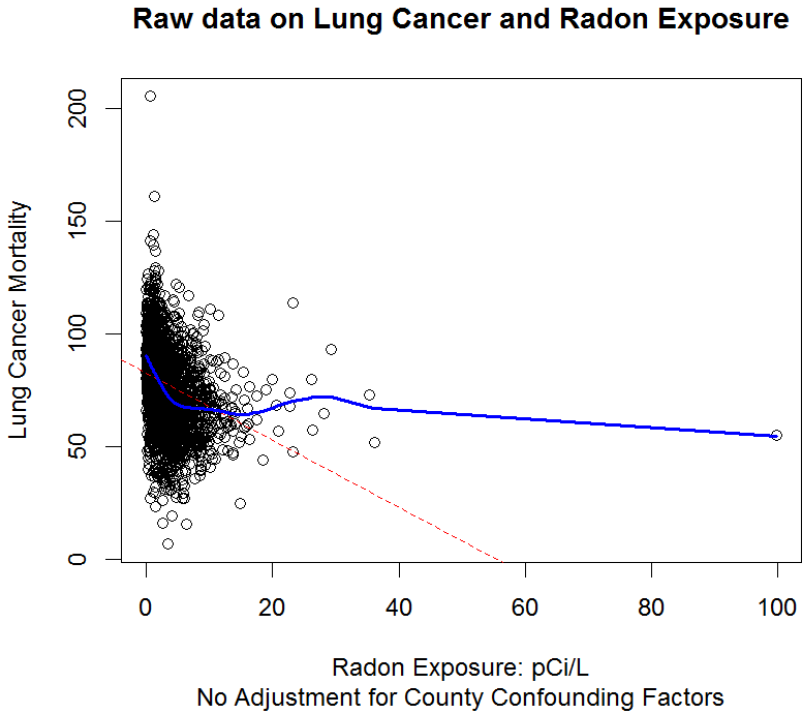
The main practical implication of [8] is that the conditional distributions of ***x***-vectors and ***t***-levels are *asymptotically independent* within given clusters ...making cluster membership an *asymptotic factoring* score.

NOTES: **R&R** (1983) also argued [in their Theorem 2, equation (2·1)] that true propensity scores are the most "coarse" of all possible factoring scores. Since *cluster membership* is an asymptotic factoring score, it follows that *cluster membership* is asymptotically either equivalent to the unknown, true propensity score or else is more "fine" than true propensity.

Indeed, **R&R** (1983, page 43) stated that the given ***x***-confounder vectors of individual experimental units define the "finest" factoring scores. In practical applications, there may be essentially NO exact-matches in the high-dimensional ***x***-space of primary interest. Yet, clusters of ***relatively well-matched*** experimental units could still exit in these situations.

4. LRC Example: Lung Cancer Mortality and Indoor Radon Exposure

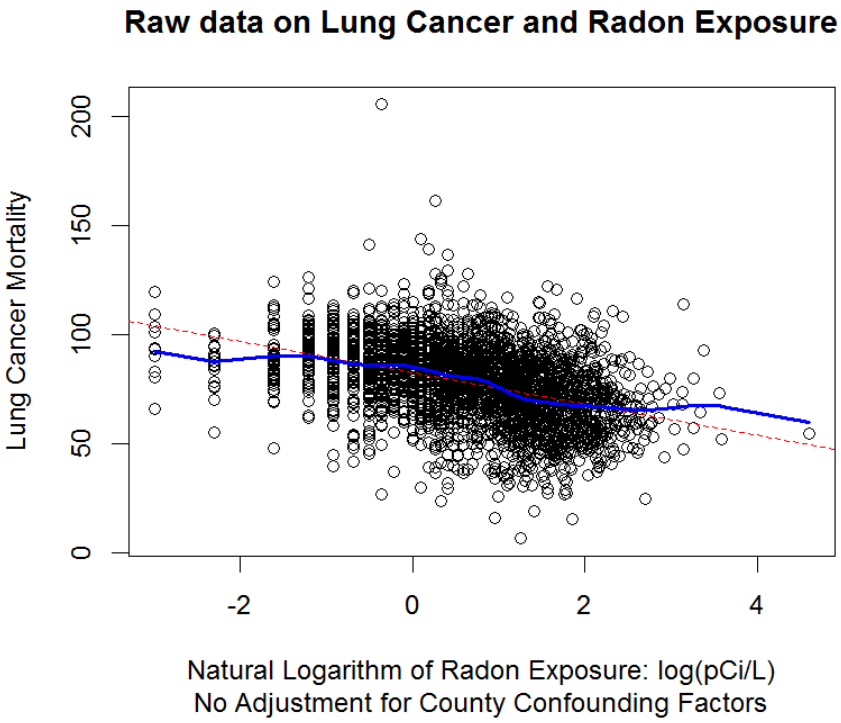
We use the **radon** dataset of Krstic(2016) here in Section §4 to illustrate use of functions from the **`LocalControlStrategy-package`**. We will start by taking a couple of preliminary "looks" at the marginal relationship between Lung Cancer Mortality and Indoor Radon Level for 2,881 US counties. Mortality is reported in deaths per 100,000 person-years, and Indoor Radon level is reported in picocuries per liter (pCi/L), *rounded to a single decimal place*. As a result, only 166 different "rounded" radon exposure levels were observed across the 2,881 US counties.



The **R** code used to generate the above plot is:

```
plot(radon$radon, radon$lcanmort, col = "black", ann = FALSE)
lmfit <- lm(radon$lcanmort ~ radon$radon)
abline(lmfit, lty = 2, col = "red")
lines(smooth.spline(radon$lcanmort ~ radon$radon), col = "blue", lwd = 2)
title(main = "Raw data on Lung Cancer and Radon Exposure",
    xlab = "Radon Exposure: pCi/L",
    ylab = "Lung Cancer Mortality", sub = "No Adjustment for County
            Confounding Factors")
```

Unfortunately, it's really not easy to "see" what is going on at low indoor radon exposure levels in the above plot. Thus we re-plot the data below using the natural logarithm of radon level on the horizontal axis. The indoor radon levels for 10 of the 2,881 counties were reported as 0.0 pCi/L, where **`log(0.0) = -Inf`**. Thus the **`lnradon`** variable simply "Winsorizes" these 10 values to **`log(0.05)`** ≈ **`-3`**.



In this, our second preliminary "look" at the **`radon`** data, we see a general tendency for Lung Cancer Mortality rates to decrease (↓) as Indoor Radon Levels increase (→). Indeed, we have not yet used ***LC Strategy*** to make statistical "adjustments" for potentially important differences in known ***x***-confounding characteristics of county residents across these 2,881 US counties.

Since our ***LC*** analyses of the **`radon`** data will be based upon observed **Rank Correlations** within individual clusters of US counties, let us start by considering all 2,881 counties as constituting a **single cluster**. The observed Pearson and Rank correlations are then:

Correlation	`attach()` and `stats::cor` function calls	Value
	`attach(radon)`	
`Pearson`	`cor(lcanmort, lnradon)`	**`-0.4099`**
`Rank`	`cor(lcanmort, lnradon, method = "spearman")`	**`-0.4483`**

Like our linear **`lm()`** and **`smooth.spline()`** fits displayed on the above plot, our correlation calculations also suggest that lung cancer mortality is ***negatively associated*** with indoor radon levels across the 2,881 US counties contained in the **`radon`** data.frame.

Unless ***LC Strategy*** makes some really big "adjustments" for the ***x***-Confounding characteristics of UC Counties, it's already relatively clear that current federal EPA policy and US state legislation requiring ***Indoor Radon Mitigation*** are NOT supported by the very data that regulators, themselves, have been amassing over the last 20+ years!

Two Preliminary Steps are needed before initiating the very first "Round" of Local Control Strategy on a new Dataset...

 # Load the `LocalControlStrategy-package` Library into the current R session...

```
library(LocalControlStrategy)
```

 # Import (observational) Data ...here the `radon` data.frame of Krstic (2016).

```
data(radon)
```

Additional Preliminary Steps needed when initiating each new "4-Phase Round" of Local Control Strategy...

 # Decide exactly "how many" and "which" baseline *x*-space characteristics of experimental # units will actually be used to form Clusters = BLOCKS of relatively "well-matched" units...

```
xvars  <- c("obesity","over65","cursmoke")
```

 # NOTE: The `radon` data contain information on two more potential *x*-confounder variables, `evrsmoke` and `hhincome`). However, when some variables appear (intuitively) "less important" than others, experimental units may be better "matched" by focusing on only fewer, more apparently relevant confounders.

 # Compute the Dendrogram (Tree) for unsupervised, nonparametric LC analyses:

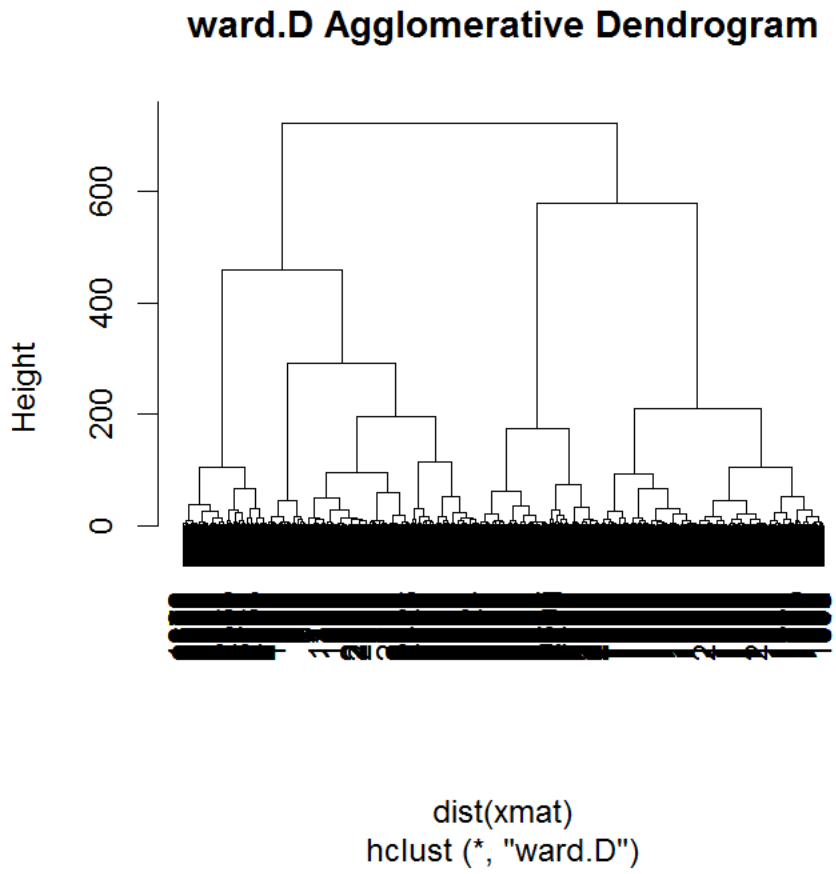
```
system.time( hclobj <- LCcluster(radon, xvars) )   # Clustering takes ~0.8 seconds.
```

```
NOTE: We simply use the default LCcluster( ) algorithm here: method = "ward.D"
Seven other methods are also available: "diana", "ward.D2", "complete",
"average", "mcquitty", "median" or "centroid".
```

```
hclobj  # implicit print()
```

```
LCcluster object: Hierarchical Clustering for LC
Data Frame input: radon
Clustering algorithm used: ward.D
Covariate X variables:[1] obesity  over65   cursmoke
```

```
plot(hclobj)
```




```
e <- LCsetup(hclobj, radon, lnradon, lcanmort)
```

```
The Treatment variable is an Exposure with 166 levels.
Local Treatment Difference (LTD) analyses are not applicable here.
Only Local Rank Correlations (LRCs) can be formed Within Clusters.
```

NOTE: Here, we save the Environment object output by **`LCsetup()`** to a single character
name, **`e`** . Of course, users could use a much longer name ...perhaps, one descriptive of which of
the available ***x***-Covariates were included or excluded from the "xvar" vector used in LCcluster().

outcome: lcanmort ...lung cancer mortality rate (continuous) for the US County
treat: lnradon ...continuous Exposure (natural log, Winsorized to exceed -3.0)

LC Strategy Phase One: Aggregation

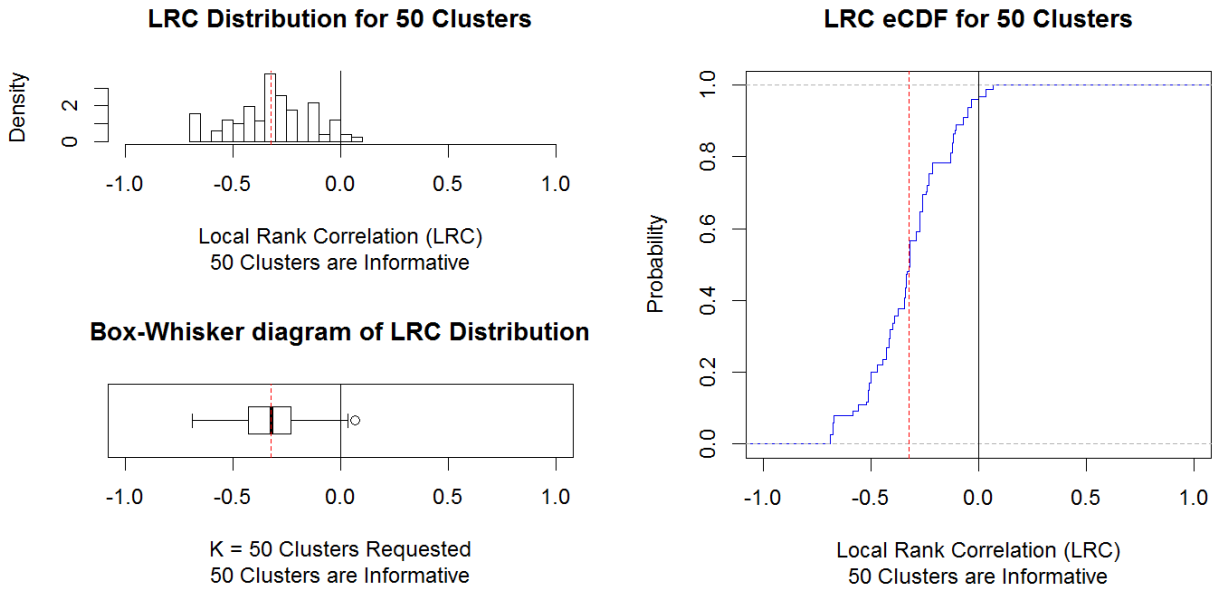
 # Compute and Save LRC distributions for a few values of K = Number of Clusters...

```
mort010 <- lrcagg( 10, e)   # Average Cluster Size:  ~288 US Counties
mort050 <- lrcagg( 50, e)   # Average Cluster Size:   ~58 US Counties
mort100 <- lrcagg(100, e)   # Average Cluster Size:   ~29 US Counties
mort200 <- lrcagg(200, e)   # Average Cluster Size:   ~14 US Counties
```

 # Each of the above 4 `lrcagg()` output objects could now be printed and/or plotted...

 # Below, we focus on visualizing the observed LRC distribution from `mort050` ...

```
plot(mort050, e)    # the default (show="all") displays 3 basic visualizations:
                    # histogram, box-plot and eCDF for the observed LRC
                    # disribution...
```



NOTES:

- Experimental units (US counties) are treated here as being equally important (i.e. are assigned equal weights) in the overall **LRC** distribution within 50 clusters depicted above.
- The importance of the i^{th} estimated **LRC** is then proportional to the size of the i^{th} cluster.
- As **K** increases in consecutive **`lrcagg()`** invocations, the same overall "weight" (**N** = 2,881 Counties) would usually be assigned to each **LRC** distribution. Clusters of 3 or more Counties are unlikely to become "uninformative" here for because **`lnradon`** has many more than 2 levels. The maximum **K** allowed is **`Kmax = floor(N/12)`**, which is 240 for the **`radon`** data.
- The reasonableness of this sort of "weighting" assumption may be somewhat lower here (where experimental units are US counties with varying total populations and geographical sizes) than in our LTD example of Section §5 ...where units are individual PCI pseudo-patients.

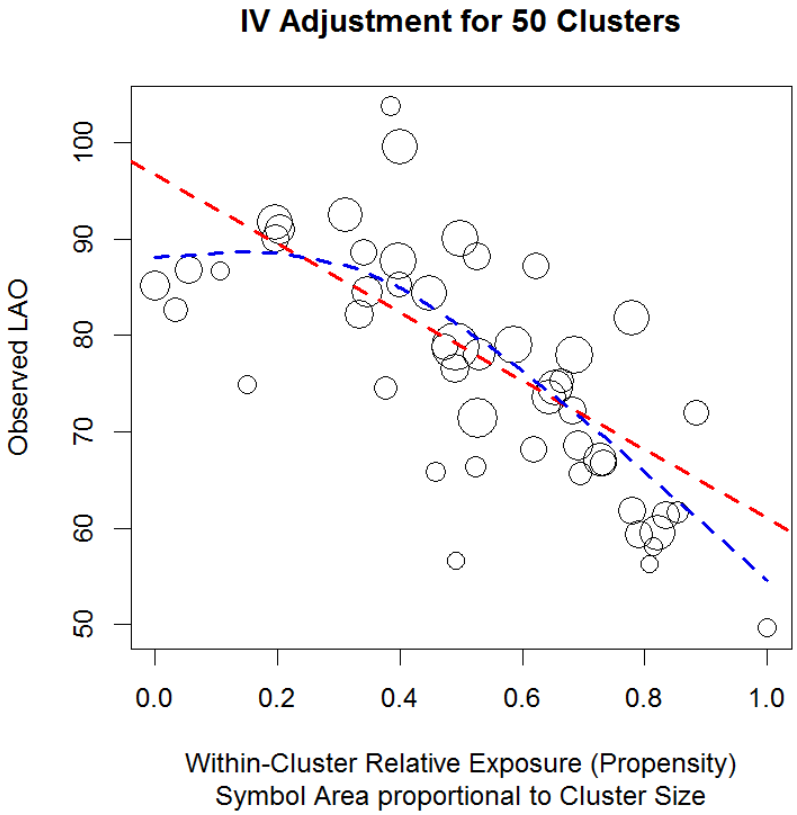
 # For situations where researchers may wish to assume (pretend)
 # that all *x*-confounders specified in `LCcluster( )` are actually
 # Instrumental Variables (IVs), the `LocalControlStrategy-package`
 # provides the `ivadj()` function...

Compute and Save statistics from the "IV distribution" for K = 50 Clusters...

```
iv050  <- ivadj(mort050)
```

NOTE: the input to **`ivadj()`** is an output object from either **`lrcagg()`**
or **`ltdagg()`** that specifies the number of clusters, **`K`** = 50, and the **`envir`**
output by **`LCsetup().`**

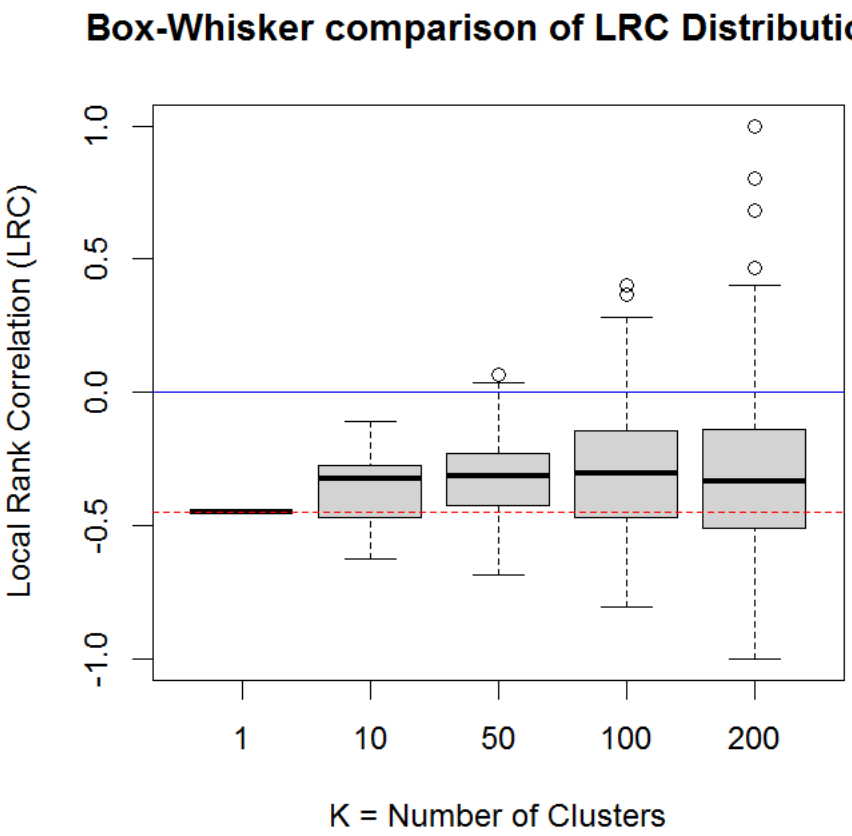
```
plot(iv050)
```
graphical display ...with linear lm() and smoothing.spline() fits.



NOTE: The **Relative Exposure** levels shown above as cluster abcissas are "like" **Propensity Scores** only in the sense that they do fall within the closed interval [0, 1]. An initial "PS-like" variable computed by **`lrcagg()`** consists of ***cluster centroids*** (local means) of Winsorized **`log(radon)`** exposure; so the min(score) starts out being negative while max(score) exceeds +1. Thus **`ivadj()`** first translates abcissas from **`lrcagg()`** by subtracting off the minimum score, then rescales them by the observed range of scores. In particular, NOTE the distinct downward

trend in Observed LAOs as the "propensity" for high radon exposure increases ...which agrees with the above distribution of mostly Negative LRCs.

```
# LC Strategy Phases One: Aggregate
#       and / or Three: Explore
#  View "Sensitivity Analysis" Summary Plot...

LCcompare(e)
```



A wise choice for K takes advantage of Variance-Bias tradeoffs.

- **Large values of K tend to produce smaller clusters with potentially Less Bias due to making "better" within-cluster *x*-Space Matches. Here, LRC medians are essentially the same for K = 100 & 200 as for K = 50, so taking K > 50 does not appear to further reduce Bias.**

- **On the other hand, using "too-many" clusters that then tend to be "too-small" yields LRC estimates that are highly variable and can possibly be misleading.**

- **Above results for K=100 and 200 strike me as mostly Inflating the Variability of the Observed LRC distribution ...relative to K=50 Clusters.**

Thus, K = 50 clusters (of average Size ≈ 58 counties) strike me as delivering an optimal Variance-Bias Trade-Off in the above plot.

Thus, we will use K=50 below to illustrate LC Confirm analyses...

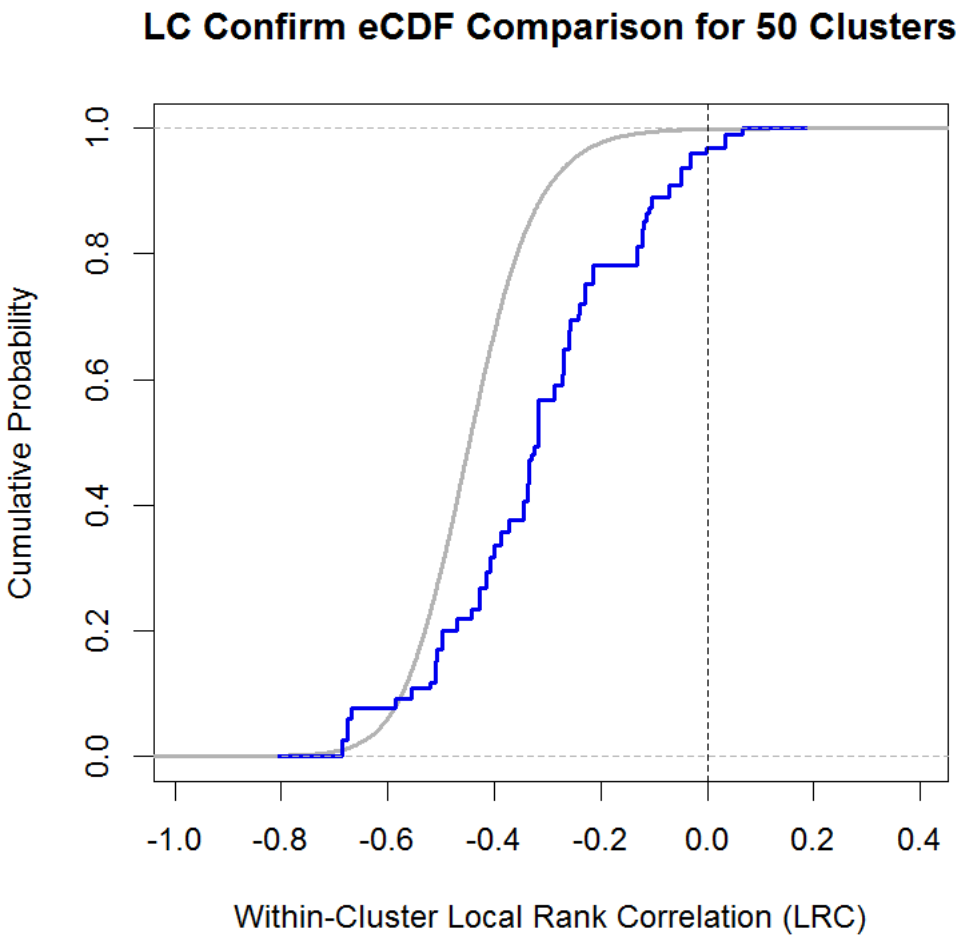
LC Strategy Phase Two: Confirm

```
#    Does the Observed LRC Distribution for 50 clusters differ in
#    clear and important ways from its NULL "Random
#    Permutation" Distribution.  This NULL distribution hypothesizes
#    that the baseline x-confounders used in clustering experimental
#    units (US Counties) are actually IGNORABLE !!!
```

```
system.time( conf050 <- confirm(mort050) )      # Simulation takes ~4.8 seconds.

conf050
```

```
confirm Object: Compare Observed and NULL Distributions of Local Effect-Sizes...
   Simulated NULL Distribution uses Random Clusterings of Experimental Units.

Data Frame: radon
Outcome Variable: lcanmort
Treatment Factor: lnradon
Number of Replications: 100
Number of Clusters per Replication: 50
Number of Random NULL Local Effect-Sizes: 288100

    Mean Observed Local Effect-Size = -0.3216341
    Std. Dev. of Observed Effect-Sizes = 0.1798312
    Mean Random NULL Effect-Size = -0.4412605
    Std. Dev. of Random Effect-Sizes = 0.1090509

Nonstandard Kolmogorov-Smirnov comparison of Discrete Distributions:
Observed two-sample KS D-statistic = 0.4587956
```

```
plot(conf050)
```



NOTES:

- The **NULL random permutation LRC distribution** (assuming ***x***-confounders are ignorable) depicted above looks very smooth and much like a symmetric and continuous (approximately normal) distribution.
- In sharp contrast, the corresponding, overlayed **Observed LRC distribution** (total weight = 2,881 counties) looks quite different from the NULL over the rank-correlation range from about –0.40 to about +0.05.
- **In other words, these two LRC distributions appear to be clearly different!**

The **`confirm()`** and **`KSperm()`** functions within the **`LocalControlStrategy-package`** use the **`stats::ks.test()`** function only to compute the two-sample Kolmogorov-Smirnov "D-statistic" and neither report nor save the **`p-value`** computed by **`ks.test()`**. After all, the "standard" K-S testing situation is where both underlying distributions being compared are continuous. TIED values between eCDF ordinates then occur with probability 0. Sample LTD and LRC distributions consist of estimates that are constant within-clusters, so TIES always occur within every informative cluster, making the **`p-value`** from **`ks.test()`** inappropriate (severely

biased downwards.) The **confirm()** and **KSperm()** functions also call **suppressWarnings(ks.test( ))** because within-cluster TIES are expected to occur in LC analyses!

KSperm() does use the highly intuitive K-S "D-statistic" to compare LTD and LRC distributions with jumps in their CDFs. These jumps tend to occur at "random" numerical values, but jumps at **LTD = 0** are predictable when the y-Outcome variable is binary. For example, see the eCDF plots on pages 27 and 31.

The primary objective of **KSperm(confirm())** is to simulate an appropriate **p-value**, "adjusted" for TIES, when testing the NULL hypothesis that the x-Covariate variables used in clustering are actually IGNORABLE. This testing is based upon random permutation theory (resampling without replacement).

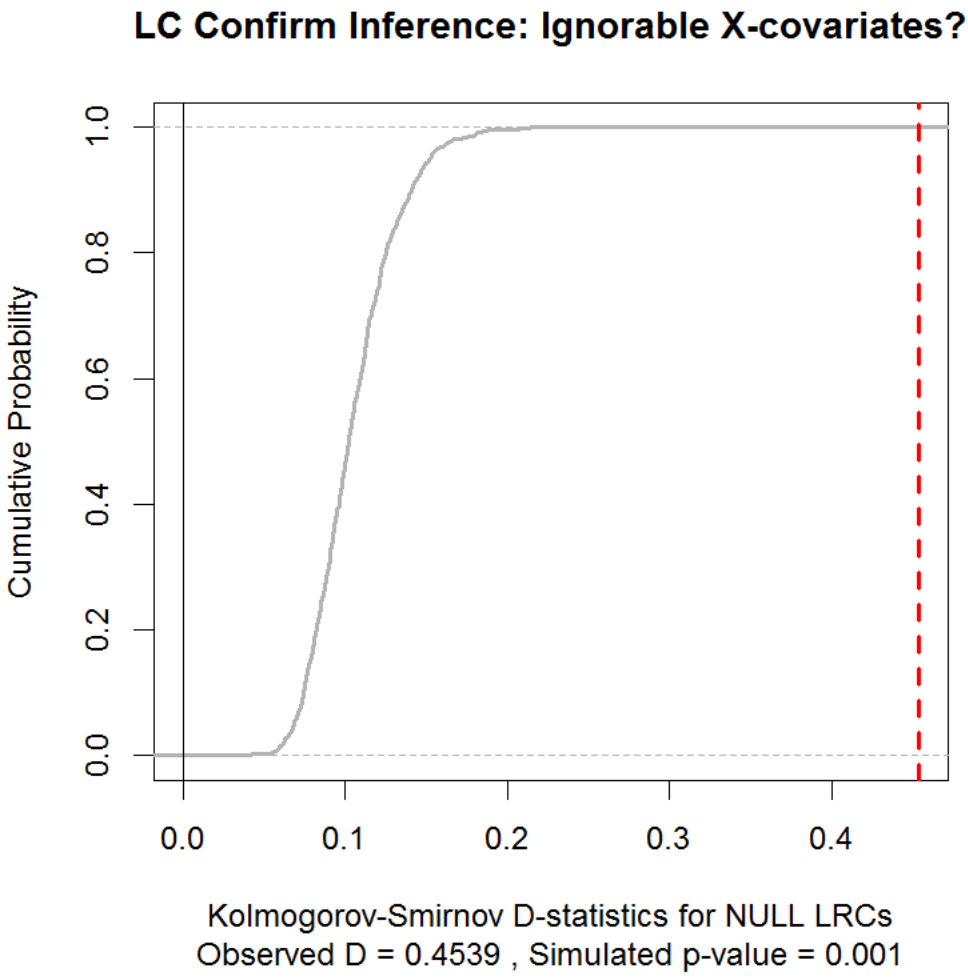
```
# Use KSperm() to simulate a p-value for the
# observed K-S D-statistic from confirm()...

system.time( ksd050 <- KSperm(conf050) )    # Simulation takes ~13.5 seconds.
                                            # Default number of reps = 100.
ksd050      # Implicit PRINT
```

```
KSperm: Simulated NULL Distribution of Kolmogorov-Smirnov D-statistics
    when the given X-covariates are assumed to be IGNORABLE.

Data Frame: radon
Outcome Variable: lcanmort
Treatment Variable: lnradon
Effect-Size estimates: Local Rank Correlations (LRCs)
Number of Random NULL D-statistics: reps = 100
Number of Clusters per replication: 50

    Observed Kolmogorov-Smirnov D-statistic = 0.4587956
    Simulated NULL KS-D order statistics =

  [1] 0.05847969 0.06034710 0.06224575 0.06267615 0.06433183 0.06860465
  [7] 0.06885456 0.06940646 0.07051371 0.07095453 0.07186741 0.07255467
 [13] 0.07382506 0.07493232 0.07531413 0.07570288 0.07639014 0.07664700
 [19] 0.07676501 0.07804235 0.07871572 0.07940646 0.07946199 0.07980562
 [25] 0.08063520 0.08093370 0.08238112 0.08324887 0.08357515 0.08385630
 [31] 0.08399861 0.08405762 0.08409233 0.08532454 0.08558834 0.08812912
 [37] 0.08853870 0.08926414 0.08955224 0.09014231 0.09019438 0.09074280
 [43] 0.09108643 0.09111767 0.09261020 0.09264144 0.09349184 0.09493579
 [49] 0.09653940 0.09687261 0.09690038 0.09915307 0.09927803 0.10013884
 [55] 0.10082263 0.10127039 0.10141617 0.10171815 0.10256508 0.10276987
 [61] 0.10281499 0.10346755 0.10398820 0.10618188 0.10633114 0.10665741
 [67] 0.10677543 0.10725443 0.10754252 0.10766401 0.10791392 0.10939951
 [73] 0.11029504 0.11049288 0.11166609 0.11298160 0.11818466 0.11995488
 [79] 0.12047206 0.12115585 0.12282541 0.12336342 0.12431100 0.12623395
 [85] 0.12652204 0.12677890 0.12761888 0.12802152 0.12872614 0.13212773
 [91] 0.13238459 0.13334606 0.13491496 0.14192294 0.14534537 0.14591114
 [97] 0.14868796 0.15201666 0.16031586 0.17372093

    Simulated adjusted p-value for the Observed D-statistic: 0.01
```

NOTES:

- In each of the (default) 100 replications requested above, 50 new NULL LRC estimates result from 50 new clusters formed by purely random permutations of the 2,881 cluster ID labels on the 50 original, observed clusters of well-matched units.
- Many more full replications (than **`reps = 100`**) could of course be requested. However, the execution time of **`KSperm()`** would then increase linearly with the value of **`reps`**.
- For example, setting **`reps = 500`** would require more than 1 minute of computation. If the maximum simulated NULL LRC order statistic were still less than the observed **D** = **`0.4588`**, then the simulated p-value would reported as **`1/500 = 0.002.`**

```
plot(ksd050)
```



The above eCDF plot for the simulated NULL distribution of "K-S D-statistic" order statistics (listed on page 20) makes it visually clear that the observed D-statistic of 0.4588 is more the than twice as large

as the maximum simulated NULL value of 0.1737. In other words, the true `p-value` (fully "adjusted" for TIES) associated with the observed D-statistic of 0.4588 is clearly MUCH less than the simulated estimate of `0.01` resulting from KSperm() with reps=100.

LC Strategy Phase Four: REVEAL

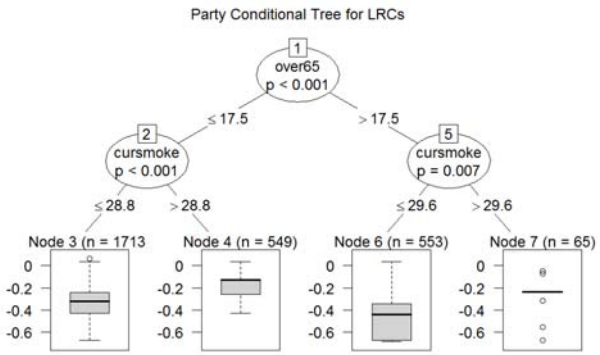
```
# How "predictable" is an observed LRC or LTD effect-
# size Distribution from the baseline x-Characteristics
# of experimental units? Such predictions may use any
# available x-vars. Those specified in LC Phases 1, 2
# and 3 should be considered here, but xvars previously
# excluded may now be reconsidered here in Phase Four.
```

 # The only function provided by the `LocalControlStrategy-package` aimed specifically at helping researchers reach their ultimate "stretch-goal" of predicting local effect-size estimates is `reveal.data()`. This function outputs a `data.frame` resulting from appropriate sorting and appending of LTD or LRC treatment effect-size estimates from `ltdagg()` or `lrcagg()` -- as well as a Cluster membership-number variable -- to a copy of the original `data.frame` input to `LCsetup()`.

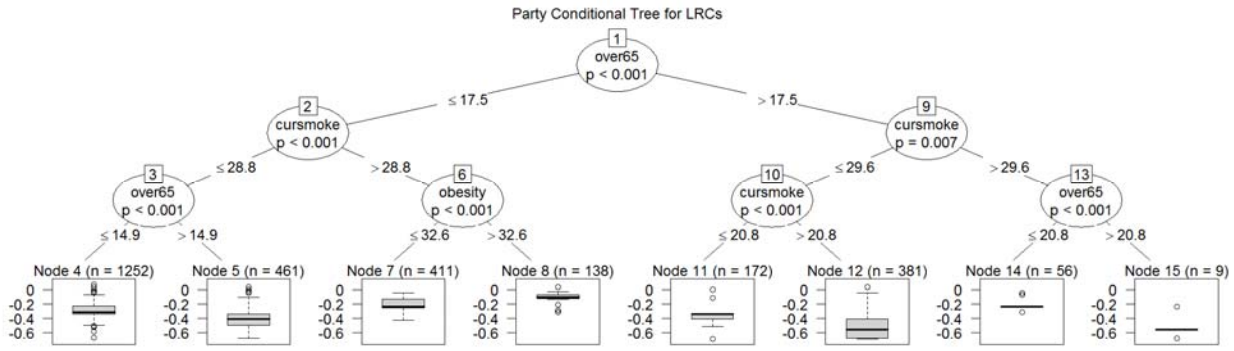
Specifically, the `data.frame` output by `reveal.data( )` is suitable for input to party::ctree() as well as to a number of other "less Visual" prediction methods available in R.

```
radonLRC <- reveal.data(mort050, clus.var="C50", effe.var="LRC50")
```

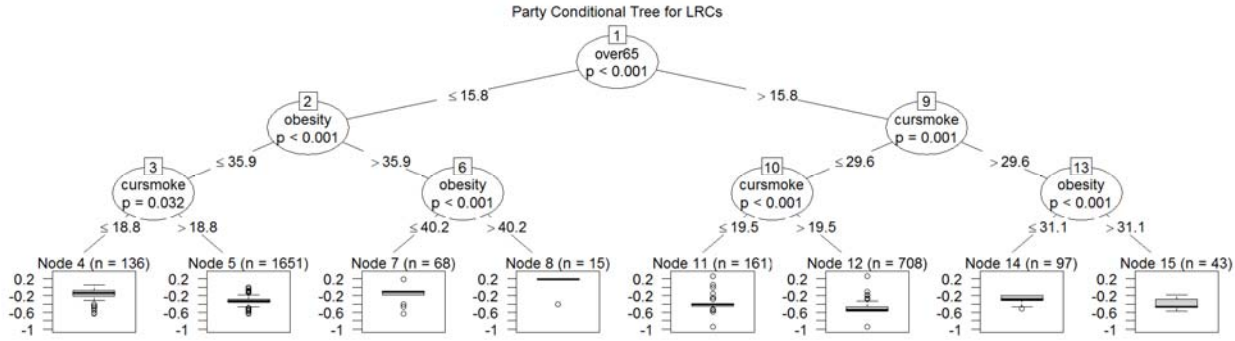
Without giving any details, we simply display below some tree-models for predicting the `LRC50` variable within the `radonLRC` data.frame. Here, we use the R party-package for creating conditional trees based upon permutation theory.



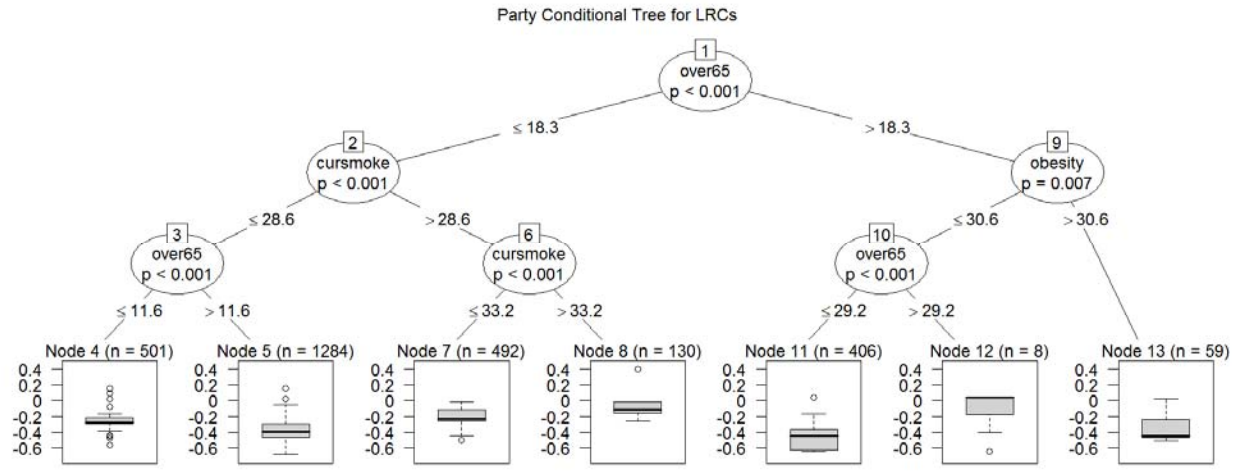
Each `ctree()` below makes binary splits at 3 levels, yielding a total of at most 2^3 = 8 "leaf" nodes at level 4.



The first four-level tree (above) comes from the `"ward.D"` clustering illustrated above, pages 14 - 22.



This second tree from 50 `"diana"` clusters predicts over a somewhat wider rank-correlation range.



This third tree from `"ward.D2"` clustering shows a rank-correlation of +0.4 within Node 8. In fact, Nodes 7 & 8 come from a 3-way split of Node 2 on `cursmoke { <= 28.6, (28.6,33.2] and > 33.2 }`.

5. LTD Example: Simulated Observational data on Augmentation of Percutaneous Coronary Interventions (PCIs) with a new Blood Thinning Agent

Here we use data from a plasmode simulation (Gadbury et al. 2008) based upon the observational study of Kereiakes(2000). Specifically, we illustrate calculation of **Local Treatment Differences** (LTDs) using **R** functions in version 1.3 of the **`LocalControlStrategy-package`**. In the original 1997 study, 996 patients received an initial PCI at Ohio Heart Health, Christ Hospital, Cincinnati and were followed for at least 6 months by the staff of the Lindner Center. The data.frame used here contains baseline characteristics and simulated outcomes for 15,487 pseudo-patients.

The **`pci15k`** data.frame contains 11 numeric variables. There are no NA's.

`patid`	**Patient ID Number.** Integer value of 1 through 15,487.
`surv6mo`	**Survival** for at least 6 months following PCI. **`1 => Yes; 0 => No.`**
`cardcost`	**Cardiac related costs** incurred within 6 months of patient's initial PCI; 1998 dollars. Reported costs were truncated by death for 404 patients with **`surv6mo == 0.`**
`thin`	Binary treatment selection indicator. A value of thin=0 implies that usual PCI care alone was received; thin=1 implies that the usual PCI care received was augmented by either planned or rescue treatment with the new blood thinning agent.
`stent`	Coronary stent deployment: **`1 => Yes ; 0 => No.`**
`height`	Patient height in centimeters: Integer between 108 and 196, inclusive.
`female`	Female gender: **`1 => Yes ; 0 => No.`**
`diabetic`	Diabetes mellitus diagnosis: **`1 => Yes ; 0 => No.`**
`acutemi`	Acute myocardial infarction within the previous 7 days: **`1=>Yes; 0=>No.`**
`ejfract`	Left ejection fraction. Numeric value from 17% to 77%.
`ves1proc`	Number of vessels involved in the patient's initial PCI procedure. Integer value of 0 through 5.

 # Load the `LocalControlStrategy-package` Library into the current R session...

```
library(LocalControlStrategy)
```

 # Input Simulated, Observational Data:

```
data(pci15k)
```

 # Decide "how many" and "which" baseline *x*-space characteristics of patients will actually
 # be used to form Clusters = BLOCKS of relatively "well-matched" patients. Here we use
 # all 7 available *x*-confounders because they could all be roughly equally important.

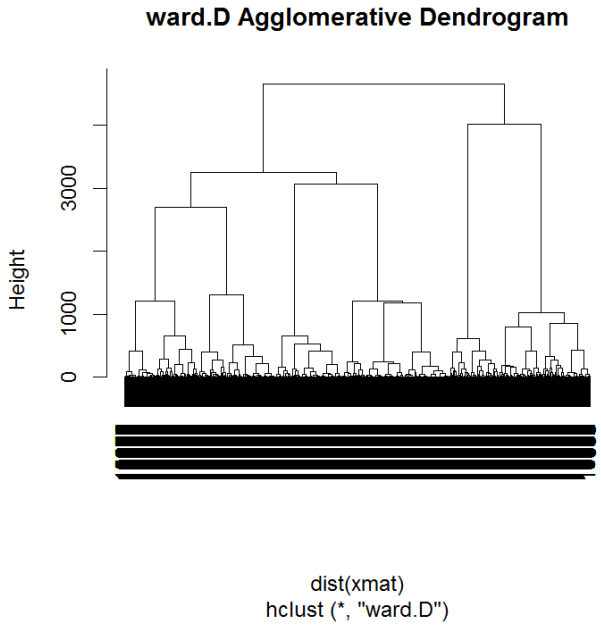
```
xvars <- c("stent", "height", "female", "diabetic", "acutemi",
           "ejfract", "ves1proc")
```

 # Compute the Dendrogram (Tree) for unsupervised, nonparametric LC analyses using one of 8 possible Clustering algorithms...

```
system.time( hclobj <- LCcluster(pci15k, xvars) ) # Takes ~8 seconds.
hclobj
```

```
LCcluster object: Hierarchical Clustering for LC
Data Frame input: pci15k
Clustering algorithm used: ward.D
Covariate X variables:[1] stent   height   female
                diabetic acutemi ejfract ves1proc
```

```
plot(hclobj)
```




```
LCe <- LCsetup(hclobj, pci15k, thin, surv6mo)
```

```
The Treatment variable has 2 levels.
Local Rank Correlation (LRC) analyses are not applicable here.
Only Local Treatment Differences (LTDs) can be formed Within Clusters.
```

 # AGAIN: Saving the Environment object output by `LCsetup()` is ESSENTIAL. Here, we # illustrate using the name `LCe`.

 # Outcome: surv6mo is also binary; 1 => Yes, 0 => No.)
 # Treatment: thin ...0 => usual PCI care alone, 1 => PCI augmented with planned or # rescue use of a new blood thinning agent.

To examine the "structure" of the **`LCsetup()`** output object, use: **`ls.str(LCe)`**

LC Strategy Phase One: Aggregation

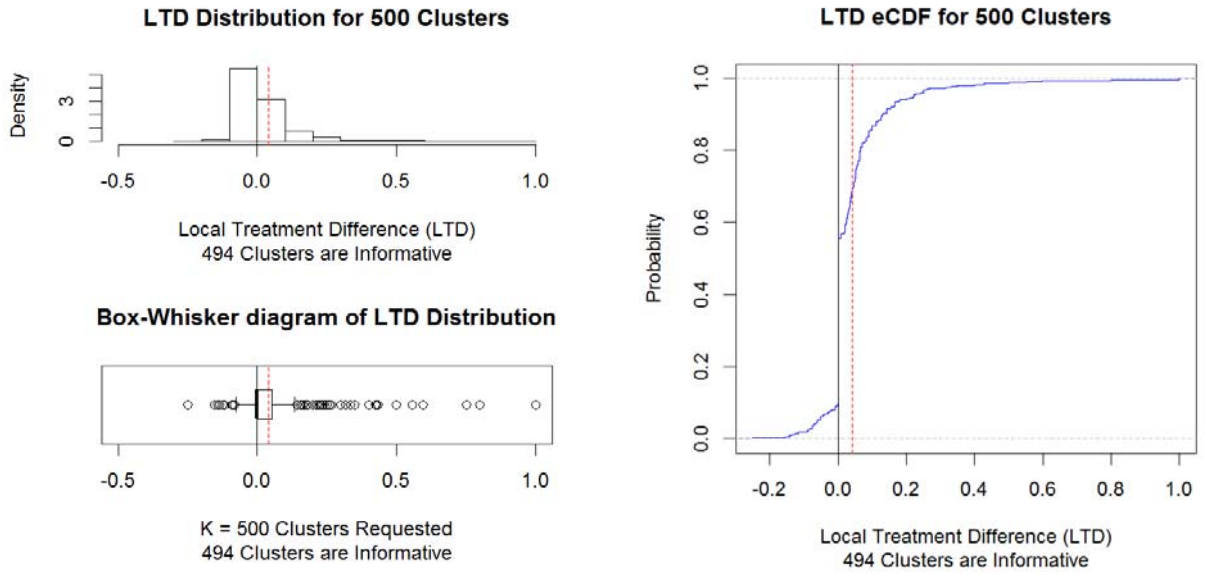
 # Compute and Save LRC distributions for a few values of K = Number of Clusters...

```
surv0050 <- ltdagg(  50, LCe)      # average cluster size ~310 patients
surv0100 <- ltdagg( 100, LCe)      # average cluster size ~155 patients
surv0200 <- ltdagg( 200, LCe)
surv0500 <- ltdagg( 500, LCe)      # average cluster size  ~31 patients
surv0750 <- ltdagg( 750, LCe)
surv1000 <- ltdagg(1000, LCe)      # average cluster size  ~16 patients
```

 # Each of the above 6 `ltdagg()` output objects could now be printed and/or plotted...

 # Below, we focus on visualizing the observed LRC distribution from `surv0500` because this choice appears to optimize variance-bias trade-offs (...see the LCcompare() plot on page 28.)

```
plot(surv0500)   # the default (show="all") displays 3 basic visualizations:
```



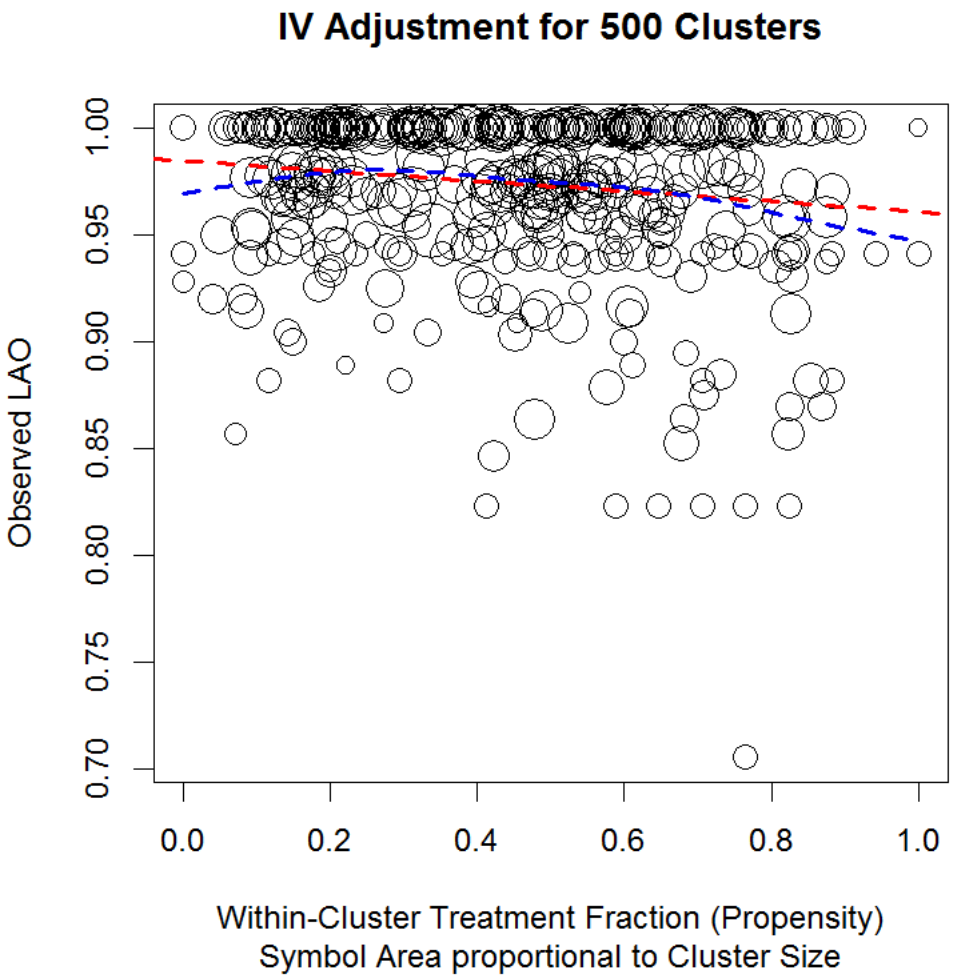

```
# Use of a Binary y-Outcome (surv6mo) creates many LTD = 0 estimates
# from clusters where all patients survived > 6 months!

# Uninformative clusters contain only thin=1 or only thin=0 patients.

# Observed LTD distribution has a more heavy upper tail of positive
# estimates ...i.e. thin=1 patients are more likely to survive!
```

Instrumental Variable (IV), inferences from the **pci15k** for K = 500 Clusters...

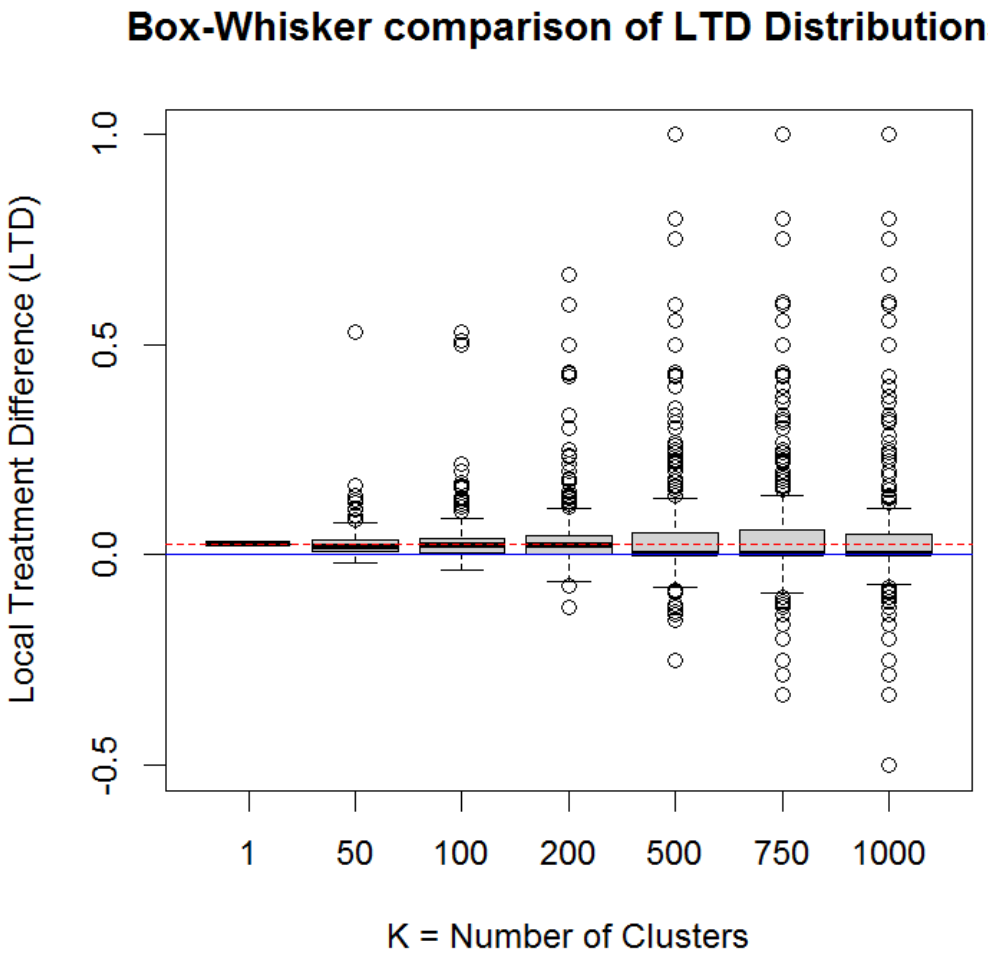
```
iv0500  <- ivadj(surv0500)  # the ivadj() input here is output from ltdagg().
```

...i.e. `surv0500 <- ltdagg( 500, LCe)` as shown near the top of page 26.

```
plot(iv050)
```
IV graphical display ...here, the linear [lm()] and smoothing.spline() fits both slightly favor thin=0 over thin=1.



```
# LC Phases... One: Aggregate & Three: Explore.
# View the "Sensitivity Analysis" Summary to Pick K...

LCcompare(LCe)
```



A wise choice for K again takes advantage of Variance-Bias tradeoffs.

See our discussion of this crucial Trade-Off on page 17.

K = 500 clusters (of average Size = ~31 patients each) is about the upper limit for K in my reading of the above plot. The median LTD drops to essentially Zero for K ≥ 500.

LC Strategy Phase Two: Confirm

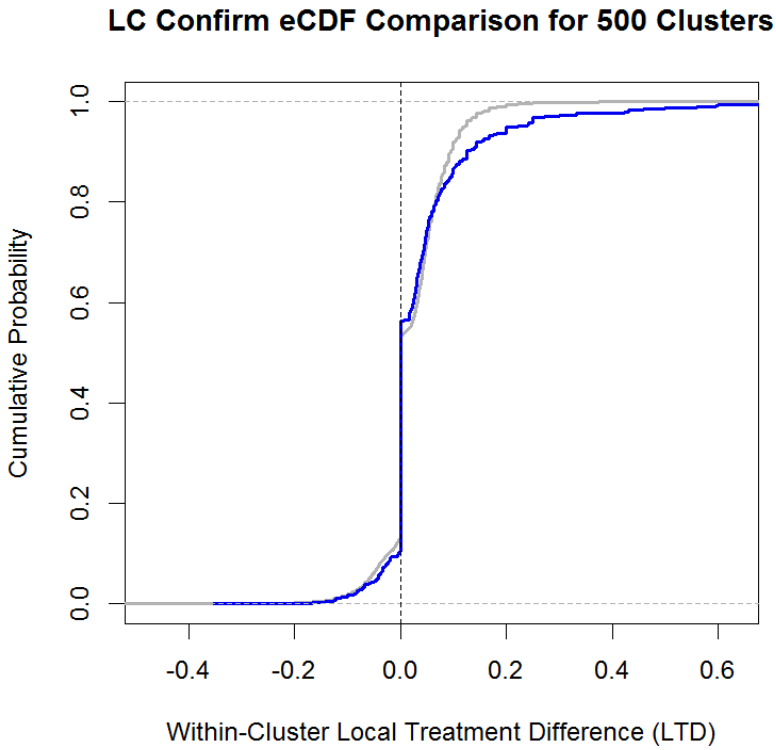
```
#    Does the Observed LRC Distribution for 500 clusters differ in clear and important ways
#    from its NULL Random Permutation Distribution?  When this NULL case holds, the
#    specified baseline x-Confounders are literally be IGNORABLE!
```

```
conf5H <- confirm(surv0500)
```

```
conf5H
```

```
confirm Object: Compare Observed and NULL Distributions of Local Effect-Sizes...
   Simulated NULL Distribution uses Random Clusterings of Experimental Units.

Data Frame: pci15k
Outcome Variable: surv6mo
Treatment Factor: thin
Number of Replications: 100
Number of Clusters per Replication: 500
Number of Random NULL Local Effect-Sizes: 1548700

    Mean Observed Local Effect-Size = 0.0415991
    Std. Dev. of Observed Effect-Sizes = 0.112318
    Mean Random NULL Effect-Size = 0.02525509
    Std. Dev. of Random Effect-Sizes = 0.05595036

Nonstandard Kolmogorov-Smirnov comparison of Discrete Distributions:
Observed two-sample KS D-statistic = 0.06997671
```

```
plot(conf5H)
```



NOTES:

- While the **Observed LTD** and **random NULL LTD** eCDF distributions do look somewhat similar here, closer examination reveals an **important difference**. Specifically, the **Observed LTD distribution** has a **thicker right-hand tail (LTD > 0.1)** than the **random NULL LTD distribution**.

- About 30% of the **random NULL distribution** consists of **LTD = 0 estimates**, while the **Observed LTD distribution** contains even **MORE** Zero LTDs (~ 40%.) This high likelihood of LTD = 0 values is an artifact that occurs simply because the **`surv6mo`** variable is BINARY.

- Again, **`warnings()`** from **`stats::ks.test()`** about within-cluster TIES are suppressed by the **`confirm()`** and **`KSperm()`** functions within the **`LocalControlStrategy-package.`**

- Below, we illustrate use of the **`KSperm()`** function to "simulate" a highly relevant adjusted p-value for the observed KS D-statistic = **`0.06998`** from **`confirm()`** for the **`pci15k`** data.frame ...one that is appropriately "adjusted" for TIES. This **`KSperm()`** simulation will assure us that the **Observed LTD effect-size distribution** contains truly ***Heterogeneous Treatment Effects*** ...predictable from the observed baseline ***x***-Characteristics of 15,487 patients!

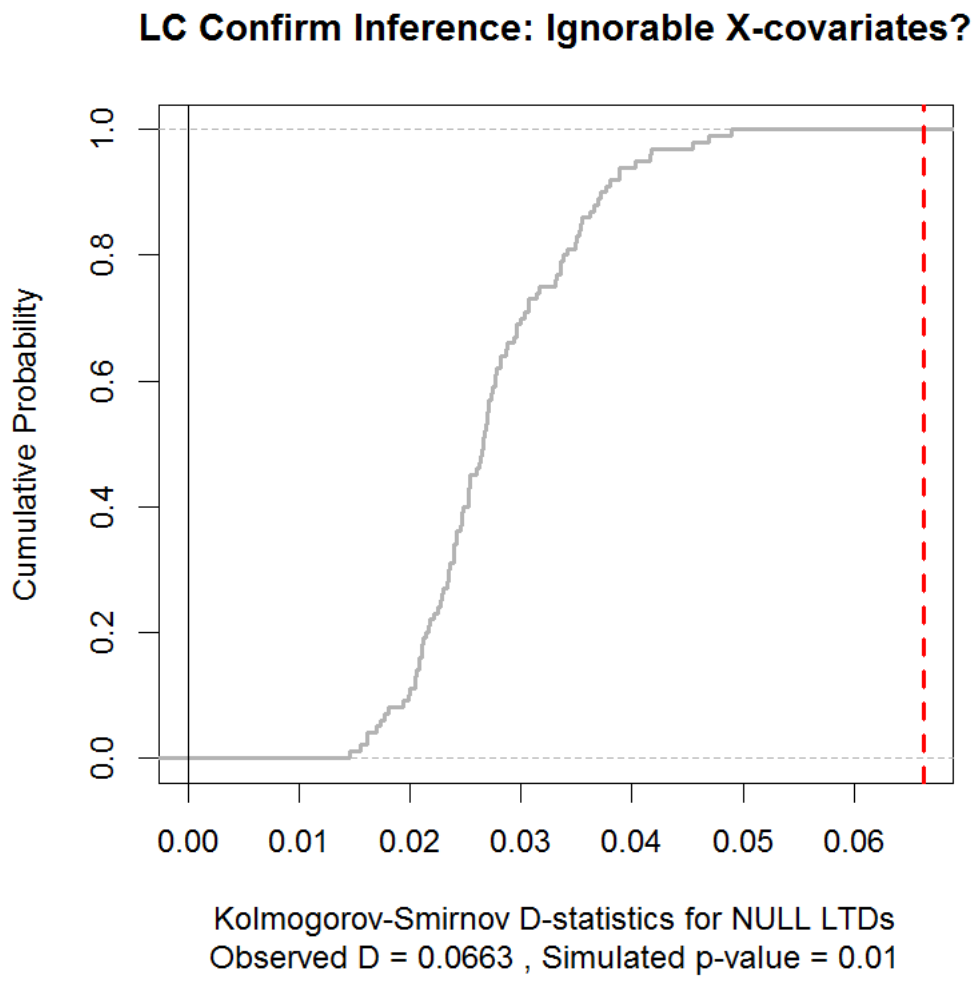
```
# Use KSperm() to simulate a p-value for the
# Kolmogorov-Smirnov D-statistic ...adjusted for TIES.

ksd5H <- KSperm(conf5H)

ksd5H # Implicit PRINT
```

```
  KSperm: Simulated NULL Distribution of Kolmogorov-Smirnov D-statistics
    when the given X-covariates are assumed to be IGNORABLE.

  Data Frame: pci15k
  Outcome Variable: surv6mo
  Treatment Variable: thin
  Effect-Size estimates: Local Treatment Differences (LTDs)
  Number of Clusters per Replication: 500
  Number of Replications: 100

      Observed Kolmogorov-Smirnov D-statistic = 0.06997671
      Sorted NULL D-statistic values =

  [1] 0.01566745 0.01793774 0.01816263 0.01836000 0.01908880 0.01919791
  [7] 0.01976417 0.01996444 0.02064527 0.02109115 0.02118090 0.02129501
 [13] 0.02175825 0.02183345 0.02184453 0.02203069 0.02205039 0.02231997
 [19] 0.02261900 0.02264890 0.02269040 0.02310424 0.02321109 0.02328852
 [25] 0.02352646 0.02371235 0.02411189 0.02422643 0.02425668 0.02429278
 [31] 0.02444308 0.02450218 0.02451578 0.02479690 0.02515066 0.02529539
 [37] 0.02534829 0.02536420 0.02547831 0.02579231 0.02589359 0.02634783
```

```
 [43] 0.02671969 0.02675350 0.02676857 0.02692155 0.02702383 0.02708499
 [49] 0.02729595 0.02738219 0.02749458 0.02756483 0.02770104 0.02770478
 [55] 0.02773865 0.02809596 0.02989248 0.02993065 0.02997812 0.03000664
 [61] 0.03006318 0.03043910 0.03057709 0.03061519 0.03101658 0.03101812
 [67] 0.03134739 0.03141035 0.03161021 0.03173051 0.03186272 0.03201433
 [73] 0.03203064 0.03226561 0.03253896 0.03358066 0.03360755 0.03407578
 [79] 0.03457569 0.03462626 0.03464702 0.03470048 0.03496531 0.03508682
 [85] 0.03533746 0.03593027 0.03657187 0.03714953 0.03774204 0.03778838
 [91] 0.03823750 0.03827738 0.03876721 0.03968821 0.03984248 0.04040203
 [97] 0.04161974 0.04282486 0.04295250 0.04840315 <= Simulated Max of 100
                                                    NULL order statistics

      The simulated p.value = 0.01
```

In other words, the **pci15k** example provides significant evidence **against** the **NULL** hypothesis that all 7 potential ***x***-confounding patient characteristics are IGNORABLE. [See plot below.] Together, the 7 baseline characteristics of 15,487 pseudo-patients in the **pci15k** are likely to be of genuine use in models that use "blocking" and/or "covariate adjustment" to compare **thin=1** patients with **usual-care-alone(thin=0)** patients.

```
plot(ksd5H)
```



LC Strategy Phase Four: REVEAL

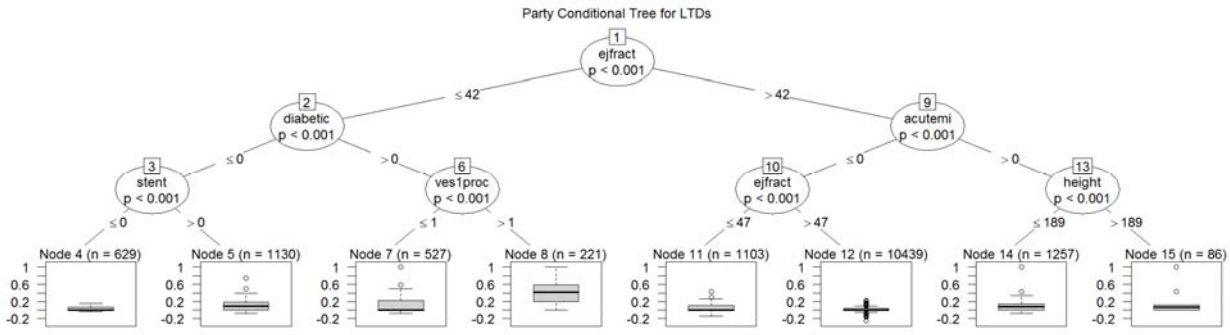
```
# How "predictable" is the LTD effect-size Distribution from
# the baseline x-Characteristics of 15,487 pseudo-patients?
```

We start addressing this question by first observing that the "new" blood thinning agent, **thin=1**, has a traditional **"highly significant" main-effect** within the **pci15k** data.

```
t.test(surv6mo ~ thin, data = pci15k)

        Welch Two Sample t-test

data:  surv6mo by thin
t = -10.318, df = 13967, p-value < 2.2e-16
mean in group 0    mean in group 1 ...Global ATE= +0.0252513.
      0.9624823          0.9877336    [ 2.5% higher surv6mo ]
```

```
pciLTD5H <- reveal.data(surv0500, "C5H", "LTD5H")
```

```
# Display a Tree Model for Predicting the LTDs for 500 "ward.D" clusters...
require(party)
set.seed(13254)

fit3 <- ctree(LTD5H ~ stent+height+female+diabetic+acutemi+ejfract+ves1proc,
      data = na.omit(pciLTD5H), controls = ctree_control(maxdepth = 3))

plot(fit3, main="Party Conditional Tree for LTDs")
```



NOTES:

- **The "statistically significant" overall 6-month survival advantage (2.5%) for PCI patients treated with the new blood thinning agent may not justify a "one-size-fits-all" endorsement. Specifically, tree node #12 is gigantic. This 67% of all PCI patients studied (with no recent `acutemi` and `ejfract` > 47) appear to do roughly equally well with or without the new blood thinner.**

- **Diabetic patients with `ejfract` ≤ 42 do either "better" with the new blood thinner (527 in node #7) or even "quite well" (221 in node #8) when more than 1 vessel was involved in their initial PCI procedure.**

- **The initial and/or long-term monetary costs associated with using the new blood thinning agent may well be an important factor in deciding whether or not to use it on certain patients. Unfortunately, the `cardcost` variable in the `pci15k` data excludes the (potentially steep) acquisition cost of that new agent!**

6. Summary - and - Choice of Clustering Method

This **`LCstrategy_in_R.pdf`** file has introduced and illustrated all four phases of ***Local Control Strategy*** for analysis of observational data. We have reviewed some simple basics of Propensity Scoring theory and shown how they relate to patient Clustering / Blocking methods in Section §3. Additional details on all 9 basic **`LC Strategy`** functions, their arguments and (default) settings, and their **`print()`** and **`plot()`** methods are all included within the official **`LocalControlStrategy-manual.pdf`** file.

An interesting feature of our two ***LC*** case-studies, on use of **LRC**s with the **`radon`** data and use of **LTD**s in the **`pci15k`** example, is that both studies lead to similar conclusions about ***effect-size Heterogeneity*****:** ***local effect-sizes*** are indeed ***predictable*** from the baseline-characteristics of individual experimental units.

> The extra zinger in our **`radon`** example was, of course, that mortality generally decreases as radon (ionizing radiation) exposure increases ...as long as radiation remains at rather low levels; this phenomenon is known as ***radiation hormesis***[1].

Also, both examples of ***LC Strategy*** given here focused on the default method (**`"ward.D"`**) of hierarchical clustering. Other possibilities useful in sensitivity analyses include one divisive method (**`"diana"`**) and six agglomerative methods (**`"ward.D2"`**,**`"complete"`**, **`"average"`**,**`"mcquitty"`**,**`"median"`** or **`"centroid`**".) The single-linkage method available from **`stats::hclust`** (**`"single"`**) is NOT recommended for use in LC analyses.

My experience is that **`"ward.D"`** clustering tends to offer a rather unique advantage: this particular algorithm appears to produce numerous clusters of nearly equal size ...while minimizing creation of clusters that are either really small or unusually large, unlike the **`"ward.D2"`** method.

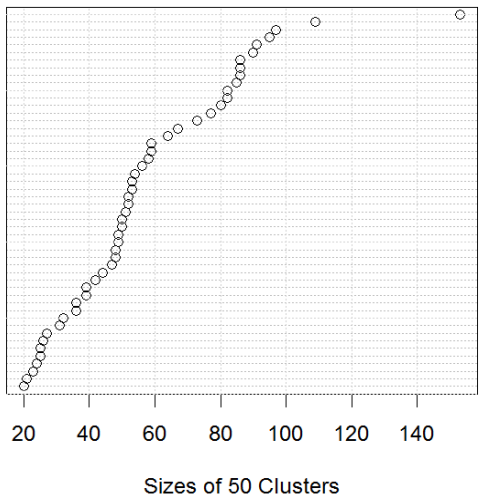
For example, compare the two plots shown near the top of page 34. Since the **R**-code for producing such plots is NOT provided by the **`LocalControlStrategy-package`**, we now discuss how **R**-users can add such functionality to almost any **R**-package.

R users familiar with the **`str()`** command can easily add new functionality not currently implemented within the **`LocalControlStrategy-package`**. For example, by examining the output from **`str(mort050)`** for the **`mort050`** object discussed here on pages 15 through 23, a user wishing to visualize ***variation in cluster sizes*** for the **`radon`** example could write a few lines of code that ends by invoking the **`graphics::dotchart()`** and **`grid()`** functions as follows:

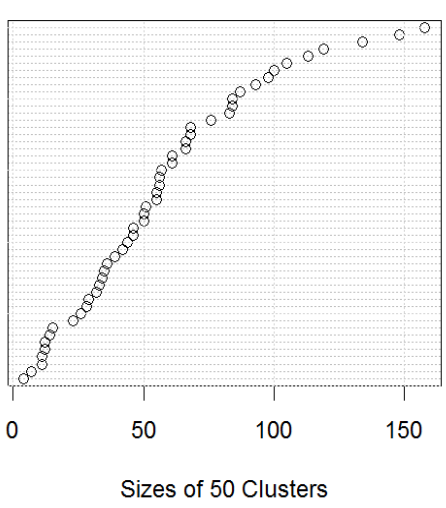
```
co <- order(mort050$LRCtabl$w)  # w = cluster size (frequency weight)
s50 <- mort050$LRCtabl[co,]
```

[1] Radiation hormesis is the hypothesis that low doses of ionizing radiation are beneficial, stimulating the activation of repair mechanisms that protect against disease, that are not activated in absence of ionizing radiation. Wikipedia

```
dotchart(s50$w,
  main = "ward.D Hierarchical Clustering",
  xlab = "Sizes of 50 Clusters")
grid()
```




```
hclobj <- LCcluster(radon, xvars,
  method = "ward.D2")
# etc.
# etc.
```



Note that **`"ward.D"`** clustering produced NO clusters containing fewer than 20 US Counties and only 2 clusters containing more than 100 US Counties. But **`"ward.D2"`** clustering produced 8 clusters containing fewer than 20 US Counties and 6 clusters containing more than 100!

We end this summary with the following **Final Cautionary Note**:

> Instrumental Variable (**IV**) methods make assumptions that, unfortunately, are both STRONG and UNVERIFIABLE. In particular, note the *inferential* "similarity" between a pair of **`radon`** scatter-plots: [1] the "unadjusted" plot on page 12 and [2] the IV plot on page16. Both suggest that lung cancer mortality generally decreases as radon exposure increases! Although a "preliminary" raw-data plot like that of page 12 for 2,881 individual US counties may be well-known to be potentially misleading, the same probably cannot be said about IV plots! In fact, the IV plot of page 16 for 50 clusters has indeed been "adjusted" via traditional BLOCKING ...but has NOT been "adjusted" for ***x***-confounders except via ASSUMPTION! Some researchers apparently think that IV methods are rather "sophisticated" or "high in the pecking-order" of potential statistical inferences. However, because assumptions can turn out to be dead WRONG, I think that both types of plots (and the inferences drawn from them) should generally be considered to provide, at best, only preliminary insights.

ACKNOWLEDGEMENT: While writing this **`LocalControlStrategy-package`** and its documentation, Dr. Obenchain received partial support from grant 1R21-LM012389 to Dr. Christophe Lambert of the University of New Mexico, Health Sciences Center, Albuquerque, NM 87131.

7. References

Alzola C, Harrell FE. ***An Introduction to S-Plus and the Hmisc and Design Libraries.*** School of Medicine, University of Virginia, Charlottesville, VA. 1999. (Plus personal communications.)

Angrist JD, Imbens GW, Rubin DB. "Identification of Causal Effects Using Instrumental Variables." ***J Amer Stat Assoc*** 1996; 91: 444-472.

Barlow HB. "Unsupervised learning." ***Neural Computation*** 1989; 1, 295-311.

Becker RA, Chambers JM, Wilks AR. ***The New S Language.*** Chapman & Hall, New York. 1988.

Chambers JM, Hastie TJ, eds. ***Statistical Models in S.*** Chapman & Hall, New York. 1992.

Cleveland WS, Devlin SJ. "Locally-weighted regression: an approach to regression analysis by local fitting." ***J Amer Stat Assoc*** 1988; 83, 596-610.

Cochran WG. "The effectiveness of adjustment by subclassification in removing bias in observational studies." ***Biometrics*** 1968; 24, 205-213.

D'Agostino RB Jr. "Tutorial in Biostatistics: Propensity score methods for bias reduction in the comparison of a treatment to a non-randomized control group." ***Stat Med*** 1998; 17, 2265-2281.

Efron B. 1979, Computers and the Theory of Statistics: Thinking the Unthinkable, ***SIAM Review*** 1979; 21, 460-480.

Gadbury GL, Xiang Q, Yang L, Barnes S, Page GP, Allison DB. Evaluating Statistical Methods Using Plasmode Data Sets in the Age of Massive Public Databases: An Illustration Using False Discovery Rates. ***PLOS Genetics*** 2008; 4, 1-8, e1000098 (Open Access).

Hastie TJ, Tibshirani RJ. ***Generalized Additive Models.*** Chapman and Hall, London. 1990.

Harrell FE. ***Predicting Outcomes: Applied Survival Analysis and Logistic Regression.*** University of Virginia, Charlottesville, VA. 1997.

Ho DE, Imai K, King G, Stuart EA. Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference, ***Political Analysis*** 2007; 15, 199-236.

Hoeffding, W. A class of statistics with asymptotically normal distribution. ***Annals of Mathematical Statistics***, 1948; 19, 293-325.

Holland PW. Statistics and Causal Inference. ***JASA*** 1986; 81, 945-960.

Hong Q, Obenchain RL, Zagar A, Faries DE. An Archive of R-code and Datasets for Comparison of Observational Data Analysis Methods via calls to SAS/STAT Procedures and Macros. Unpublished Technical Materials. 2011.

Ihaka R, Gentleman R. "**R**: A language for data analysis and graphics." ***J Comp & Graph Stat*** 1996; 5, 299-314.

Kaufman L, Rousseeuw PJ. ***Finding Groups in Data. An Introduction to Cluster Analysis.*** New York: John Wiley and Sons. 1990.

Kereiakes DJ, Obenchain RL, Barber BL, Smith A, McDonald M, Broderick TM, Runyon JP, Shimshak TM, Schneider JF, Hattemer CH, Roth EM, Whang DD, Cocks DL, Abbottsmith CW. Abciximab provides cost effective survival advantage in high volume interventional practice. ***Am Heart J*** 2000; 140, 603-610.

Lopiano KK, Obenchain RL, Young SS. Fair treatment comparisons in observational research. ***Statistical Analysis and Data Mining*** 2014; 7, 376-384, DOI: 10.1002/sam.11235.

McClellan M, McNeil BJ, Newhouse JP. "Does More Intensive Treatment of Myocardial Infarction in the Elderly Reduce Mortality?: Analysis Using Instrumental Variables." ***JAMA*** 1994; 272, 859-866.

Obenchain RL. "Nearest Neighbors Analysis for PRRAP, the Probable Report Rate Analysis Plan." **Bell Laboratories** TM-79-1711-4, Holmdel, NJ. 1979.

Obenchain RL, Melfi CA. "Propensity Score and Heckman Adjustments for Treatment Selection Bias in Database Studies." ***Proceedings of the Biopharmaceutical Section,*** Washington, DC: American Statistical Association. 1997; 297-306.

Obenchain RL. "Unsupervised Propensity Scoring: NN and IV Plots." ***2004 Proceedings of the American Statistical Association*** (on CD.) 8 pages.

Obenchain RL. SAS Macros for Local Control (Phases One and Two), Observational Medical Outcomes Partnership (OMOP), Foundation for the National Institutes of Health (Apache 2.0 License) 2009; http://localcontrolstatistics.org

Obenchain RL. The Local Control Approach using JMP, ***Analysis of Observational Health Care Data Using SAS***, Faries DE, Leon AC, Haro JM, Obenchain RL, eds. Cary, NC: SAS Press 2010; 151–192.

Obenchain RL. Observational Data Analysis Competition: Heterogeneous Response Challenge. ***MBSW 2011;*** http://www.mbswonline.com/presentationyear.php?year=2011

Obenchain RL, Hong Q, Zagar A, Faries DE. Observational Data Simulation Scenarios for Windzorized Yearly Costs of Patients with Major Depressive Disorder, 2011; Unpublished Technical Materials.

Obenchain RL and Young SS. “Advancing Statistical Thinking in Observational Health Care Research.” ***J Stat Theory Practice*** 2013; 7, 456-469, DOI: 10.1080/15598608.2013.772821.

Obenchain RL and Young SS. “Local Control Strategy: Simple Analyses of Air Pollution Data can reveal Heterogeneity in Longevity Outcomes.” ***Risk Analysis*** 2017; 37, 1742-1753.

Rosenbaum PR, Rubin RB. “The Central Role of the Propensity Score in Observational Studies for Causal Effects." ***Biometrika*** 1983; 70, 41-55.

Rosenbaum PR, Rubin DB. "Reducing Bias in Observational Studies Using Subclassification on a Propensity Score." ***J Amer Stat Assoc*** 1984; 79, 516-524.

Rosenbaum PR, Rubin DB. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. ***Amer Statist*** 1985; 39, 33-38.

Rosenbaum PR. “Optimal matching in observational studies.” ***J Amer Stat Assoc*** 1989; 84, 1024-1032.

Rosenbaum PR. “Multivariate matching methods.” In: Kotz S, Read CR, Banks D, eds. ***Encyclopedia of Statistical Sciences***, Update Volume 2. New York: J Wiley 1998; 435-438.

Rosenbaum PR. ***Observational Studies, Second Edition.*** New York: Springer-Verlag 2002.

Rubin DB. “Bias reduction using Mahalanobis metric matching.” ***Biometrics*** 1980; 36, 293-298.

Standard Form CMS-R-0235L. Instructions for Completing the Limited Dataset Data Use Agreement (DUA.) Section 8.a. http://www.cms.gov/Medicare/CMS-Forms/CMS-Forms, 2012.

Stuart EA. Matching Methods for Causal Inference: A Review and a Look Forward, ***Statistical Science*** 2010; 25, 1–21.

van der Laan M, Rose S. Statistics Ready for a Revolution: Next Generation of Statisticians must build Tools for Massive Data Sets, ***AMStat News***, 2010; September: 38-39.

8. Syntax: `LocalControlStrategy-package` R `functions`:

The **`LocalControlStrategy-manual.pdf`** file provides much more complete information about calling sequences and parameter settings than the short summary given below.

```
hclobj <- LCcluster(dframe, xvars, method = "ward.D") ...where
                          xvars <- c("x1", "x2", ..., "xN").
     plot.LCcluster(x, ...)
     print.LCcluster(x, ...)

LCe <- LCsetup(hclobj, dframe, trex, yvar)

ltdobj <- ltdagg(K, LCe)
     plot.ltdagg(x, LCe, show = "all", breaks="Sturges", ...)
     print.ltdagg(x, ...)

lrcobj <- lrcagg(K, LCe)
     plot.lrcagg(x, LCe, show = "all", breaks="Sturges", ...)
     print.lrcagg(x, ...)

ivobj <- ivadj(x) ...for x a ltdagg() or lrcagg() output object.
     plot.ivadj(x, ...)
     print.ivadj(x, ...)

confobj <- confirm(x, reps=100, seed=12345) ...where x is a
                          ltdagg() or lrcagg() output object.
     plot.confirm(x, ...)
     print.confirm(x, ...)

KSpobj <- KSperm(x, reps=100) ...where x is a confirm() output
                                  object.
     plot.KSperm(x, ...)
     print.KSperm(x, ...)

LCcompare( LCe )

outdf <- reveal.data(x, clus.var="Clus", effe.var="eSiz")
            ...where x is a ltdagg() or lrcagg() output object.
```