Generalizing from Purposive Surveys
How large a Sample is Needed

Richard Garfield
Columbia University School of Nursing

Jared P. Lander
JP Lander Consulting

December 20, 2012

Contents

1 Distribution Functions

These are the functions used to calculate the distribution of each answer. They are general and should work with any question.

```
getwd()
```

```
[1] "C:/Users/Jared/week2/writeup/distFuncs"
```

```
# Distribution functions
require(useful)
## builds the distribution for a given question
build.dist <- function(data, lhs, group, question)
{
    theFormula <- build.formula(lhs = lhs, rhs = c(group,
        question))
    agg <- aggregate(theFormula, data, length)
    agg <- ddply(agg, .variables = group, .fun = function(x)
    {
        x$Percent <- x[[lhs]]/sum(x[[lhs]])
        return(x)
```

```
    })
    agg
}
## get random tehsils from a province
village.list <- function(x, num = 5, unit = "Tehsil")
{
    # get list of units
    units <- unique(x[, unit])

    # sample num of those without replacement
    keepers <- sample(x = units, size = min(num, length(units)),
        replace = FALSE)

    return(as.character(keepers))
}
# function to make names of dist's better
change.names <- function(names, include = names, prefix = "")
{
    theOnes <- which(!names %in% include)
    names[theOnes] <- sprintf("%s.%s", prefix, names[theOnes])
    return(names)
}
## function to impute missing
impute.col <- function(col, value = 0)
{
    col[is.na(col)] <- value
    return(col)
}
## this compares two distributions and computes an MSE
compare.dist <- function(full, partial, compare = "Percent",
    by = intersect(names(full), names(partial)))
{
    # prepend Pull onto certain names in full
    names(full) <- change.names(names = names(full), include = by,
        prefix = "Full")

    # prepend Partial onto certain names in full
    names(partial) <- change.names(names = names(partial),
        include = by, prefix = "Partial")

    full.compare <- sprintf("Full.%s", compare)
    partial.compare <- sprintf("Partial.%s", compare)

    # join the two together
    both <- join(x = full, y = partial, by = by, type = "left")
```

```
    rm(full, partial)

    ## fill in any NA's with zero
    both[[full.compare]] <- impute.col(col = both[[full.compare]],
        value = 0)
    both[[partial.compare]] <- impute.col(col = both[[partial.compare]],
        value = 0)

    both$.Diff <- both[[full.compare]] - both[[partial.compare]]

    both$.MSE <- mean(both$.Diff^2)

    # attr(x=both, which='MSE') <- mean(both$.Diff^2)

    # aggregate(build.formula(lhs='.Diff', rhs=

    return(both)
}
```

List of Figures