Valid Inference from Early Stage Surveys

Richard Garfield
Columbia University School of Nursing

Jared P. Lander
JP Lander Consulting

May 31, 2012

Contents

1 Distribution Functions

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

```
# 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)
}
```

2 Initial Stuff

The data is as described in Section 3.

We examined the answer to the question"What percentage of rice crops were lost due to the flood?" We then randomly chose five Tehsils from each province, then 10, then 15 and performed the same analysis on the reduced data.

In situations where a province has fewer than five, 10 or 15 Tehsils sampled, all were used.

3 The Data

The data was collected following the floods in Pakistan in 2010. Small Changes.

It surveyed affected villages in GB, KPK, Punjab and Sindh.

The distribution of villages within Tehsils within Provinces is seen in Figure 1.

Here is the code to run it.

```
ggplot(vills, aes(x = Tehsil)) + geom_bar(aes(y = Village), stat
= "identity") +
    opts(axis.text.x = theme_text(angle = 270, hjust = 0)) +
facet_wrap(~Province,
    scales = "free_x")
```

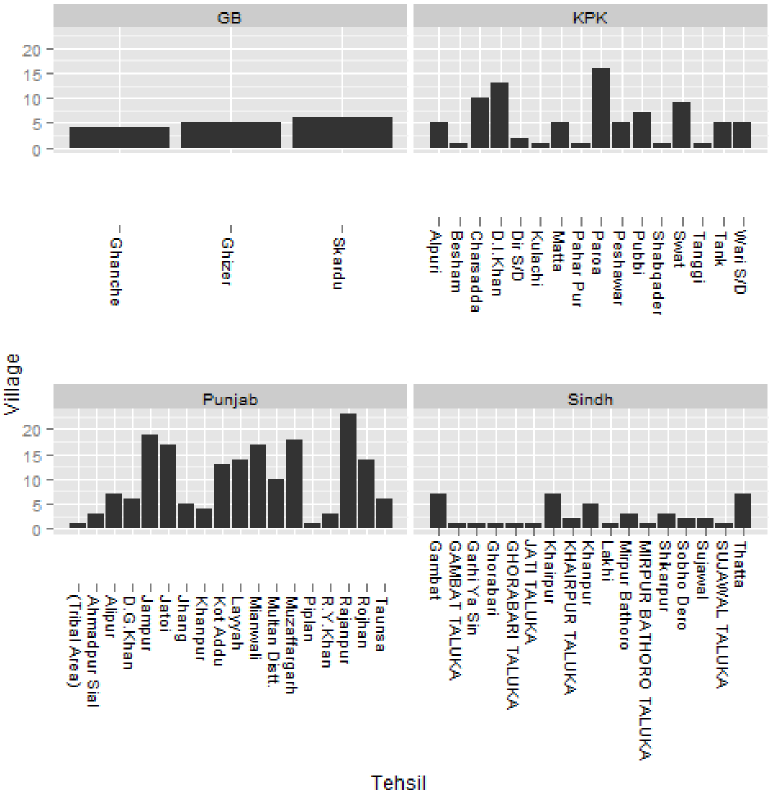

Figure 1: Distribution of villages within Tehsils within the four Provinces.

The analysis begins in Section 4.

4 Analyzing All Data

Here we analyze all of the data.

First we load the data and view a portion of it. Some more details.

```
require(useful)
load("C:/Users/Jared/week2/data/pakistan/pak.rdata")
source("C:/Users/Jared/week2/R/distFuncs.r")
corner(pak, c = 15)
```

```
##   New_ID Age  Sex     Date Province District Tehsil        Village
## 1   1288  26 Male 29082010      KPK  Shangla Besham abaseen colony
## 2   1290  30 Male 29082010      KPK  Shangla Besham abaseen colony
## 3   1370  54 Male 28082010      KPK  Shangla Besham abaseen colony
## 4   1372  53 Male 28082010      KPK  Shangla Besham abaseen colony
## 5   1371  64 Male 28082010      KPK  Shangla Besham abaseen colony
##   Latitude Longitude Total Urban Rural
## 1    34.94     72.88  90.6     -  90.6
## 2    34.94     72.88  90.6     -  90.6
## 3    34.94     72.88  90.6     -  90.6
## 4    34.94     72.88  90.6     -  90.6
## 5    34.94     72.88  90.6     -  90.6
##                                  Accommodation StagnantWater
## 1 Collective centers (school/Public building)           Few
## 2                                  Host family           Few
## 3          On the site of the house (Damaged)           Few
## 4          On the site of the house (Damaged)          None
## 5          On the site of the house (Damaged)          None
```

Now we build a distribution for all the data and visualize it in Figure 2 with the code here:.

```
ricePerc <- build.dist(data = pak, lhs = "New_ID", group =
"Province",
    question = "RiceLost")
ricePerc$Size <- "All"
ggplot(ricePerc, aes(x = RiceLost, y = Percent)) + geom_bar(stat
= "identity") +
    facet_wrap(~Province) + opts(axis.text.x = theme_text(angle =
90))
```

In Section 5 we analyze the distribution of responses for samples of fewer Tehsils.

```
ricePerc <- build.dist(data = pak, lhs = "New_ID", group =
"Province",
    question = "RiceLost")
ricePerc$Size <- "All"
ggplot(ricePerc, aes(x = RiceLost, y = Percent)) + geom_bar(stat
= "identity") +
    facet_wrap(~Province) + opts(axis.text.x = theme_text(angle =
90))
```

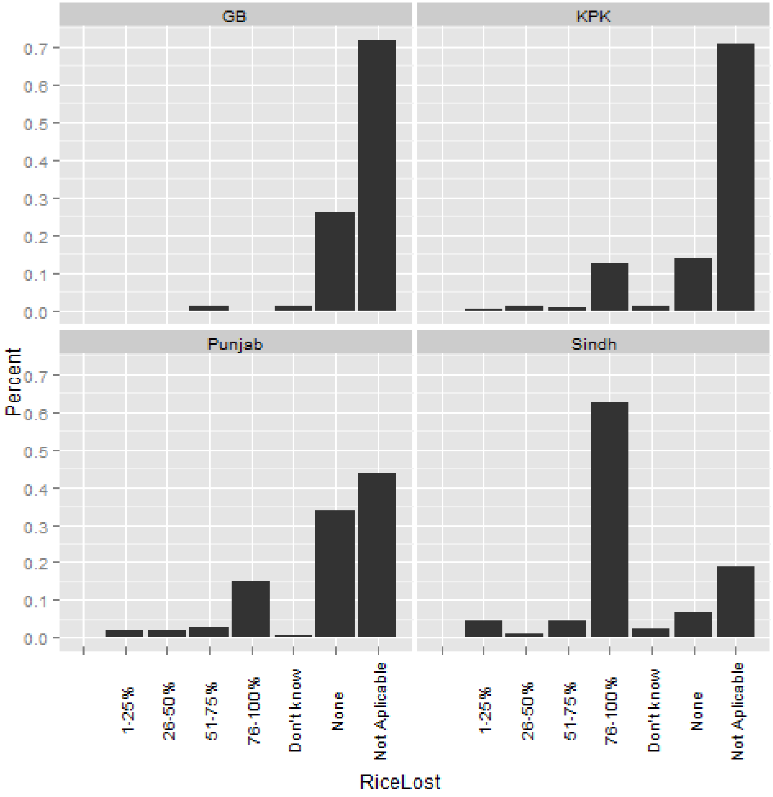

Figure 2: Graphical view of the distribution of responses for all the data.

5 Analyzing Smaller Samples

We build a similar distribution using just 5 Tehsils (max) per province as seen in Figure 3 with the code here:.

```
pak5 <- pak[pak$Tehsil %in% unlist(dlply(pak, .variables =
"Province",
    .fun = village.list, num = 5, unit = "Tehsil")), ]
pak5$Tehsil <- factor(pak5$Tehsil)
rice5Perc <- build.dist(data = pak5, lhs = "New_ID", group =
"Province",
    question = "RiceLost")
rice5Perc$Size <- "5"
compare5 <- compare.dist(ricePerc, rice5Perc, by = c("Province",
    "RiceLost"))
compare5$Partial.Size <- impute.col(col = compare5$Partial.Size,
    5)
ggplot(rice5Perc, aes(x = RiceLost, y = Percent)) + geom_bar(stat
= "identity") +
    facet_wrap(~Province) + opts(axis.text.x = theme_text(angle =
90))
```

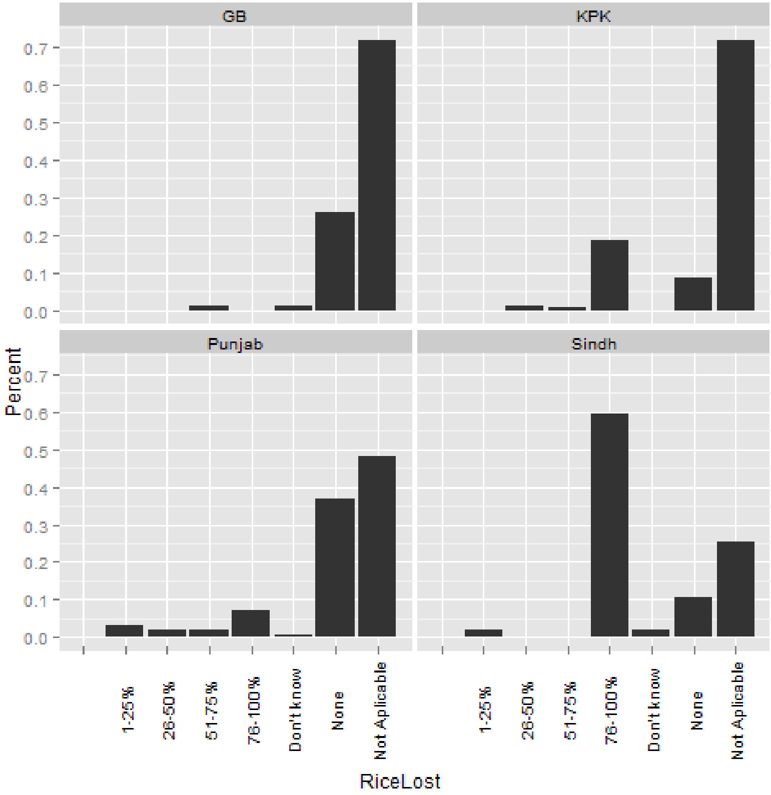

Figure 3: Distribution for five Tehsils per Province.

The same distributions for 10 and 15 samples are seen in Figures 4 and 5 respectively. This time, for brevity, the code will not be displayed.

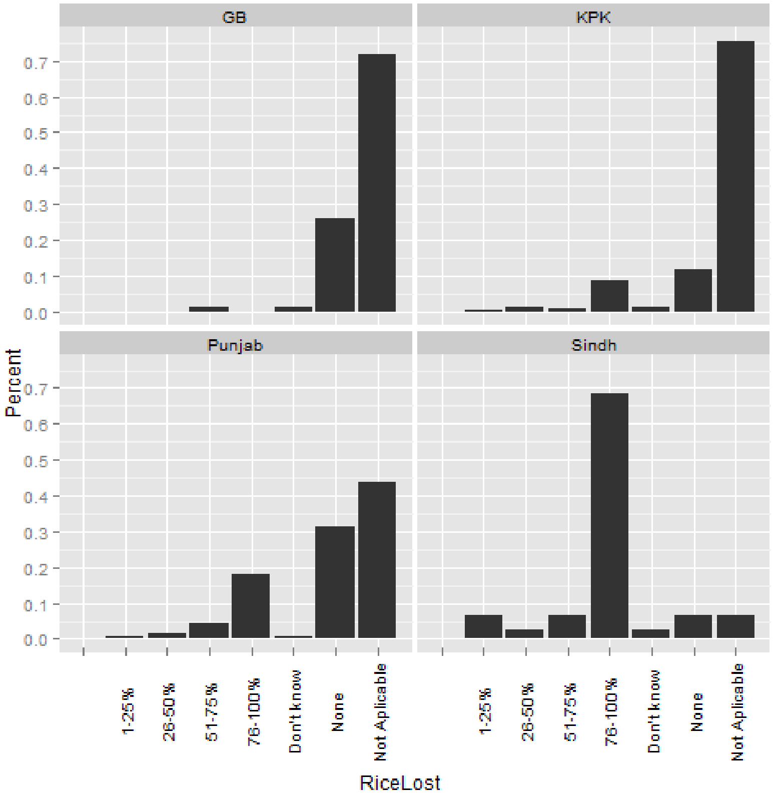

Figure 4: Distribution for ten Tehsil per Province.

Now we wish to to look at the various distributions in a single plot. Here is the code:

```
# plot distributions for all measurement sizes on same graph
allT <- rbind(rice5Perc, rice10Perc, rice15Perc, ricePerc)
allT$Size <- ordered(allT$Size, levels = c(5, 10, 15, "All"))
ggplot(allT, aes(x = RiceLost, y = Percent)) + geom_bar(aes(group
= Size,
    fill = Size), stat = "identity", position = "dodge") +
opts(axis.text.x = theme_text(angle = 90)) +
    facet_wrap(~Province)
```

The plot is in Figure 6.

Figure 7 displays the error between the smaller samples and the full distribution. The code to do so is here:

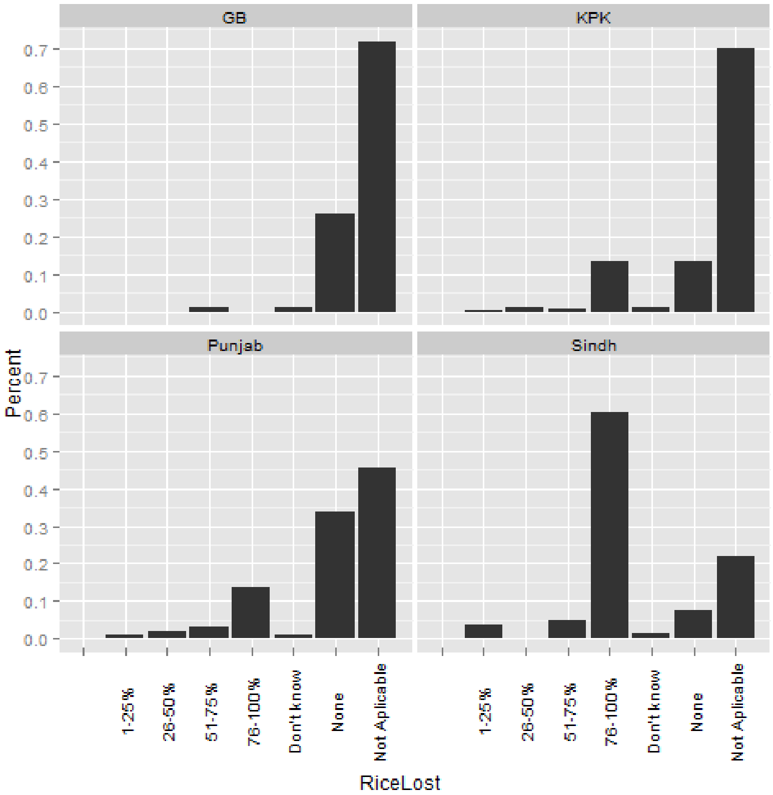

Figure 5: Distribution for 15 Tehsils per Province.

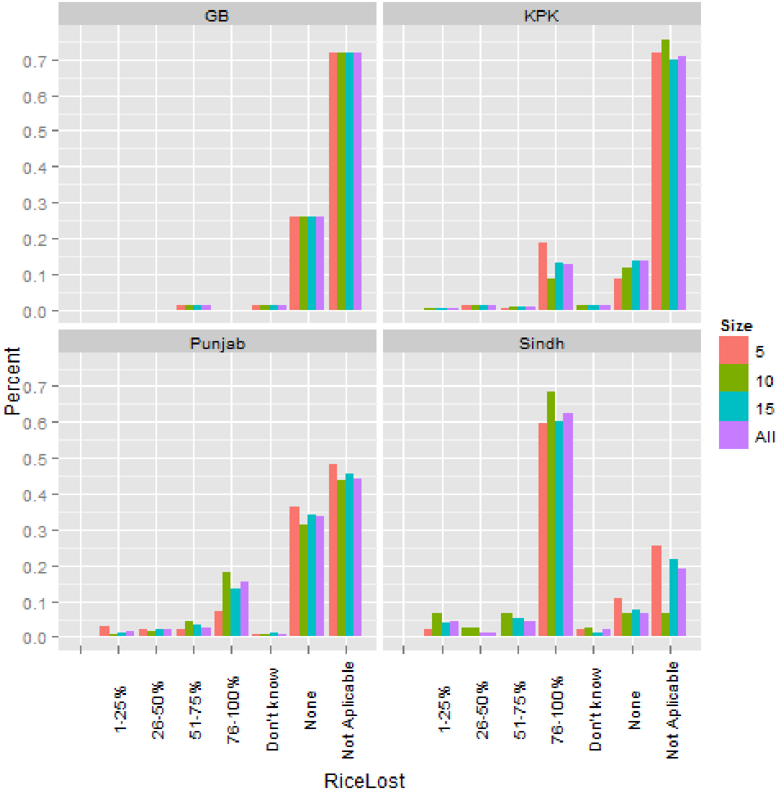

Figure 6: The distribution for all sampling types in one plot.

```
allC <- rbind(compare5, compare10, compare15)
allC$Partial.Size <- ordered(allC$Partial.Size, levels = c(5, 10,
    15))
allC$Province <- factor(allC$Province)
ggplot(allC, aes(x = RiceLost, y = .Diff)) + geom_line(aes(fill =
Province,
    colour = Province, group = Province)) + opts(axis.text.x =
theme_text(angle = 90)) +
    facet_wrap(~Partial.Size) + geom_hline(yintercept = 0, colour
= "grey",
    linetype = 2)
```

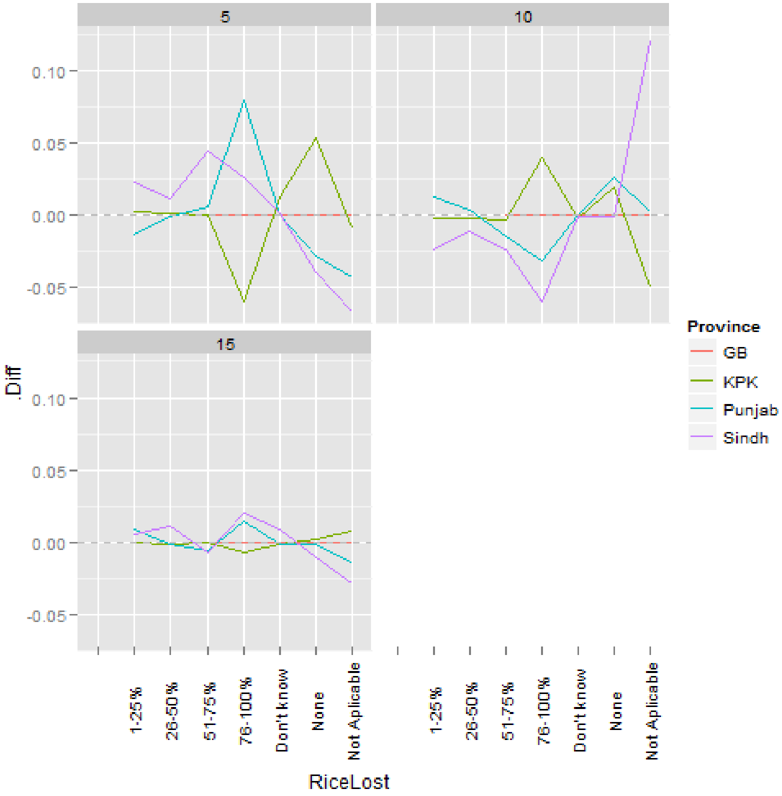

Figure 7: The difference between the true distribution and the smaller smaples.

Closing text.

List of Figures