Exploratory Data Analysis of Survey Data of SKEP Phase I

Sith J

February 19, 2558 BE

Contents

Load library 1
Load data 2
Start Explore the data 6

Load library

```
#### Load Library ####
library(gdata) # load xls file
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
##
## The following object is masked from 'package:stats':
##
##     nobs
##
## The following object is masked from 'package:utils':
##
##     object.size
```

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:gdata':
##
##     combine
##
```

```
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following object is masked from 'package:plyr':
##
##     here
```

```
library(ggplot2)
#### end load libraries ####
```

Load data

```
#### Set working directory and filepath ####
wd <- "~/Documents/R.github/network.analysis.skep1"
setwd(wd)
#### End directory and filepath ####
```

```
##### Load raw data (Survey data in SKEP 1) ####
Filepath <- "~/Google Drive/1.SKEP1/SKEP1survey.xls"
# please check your file in shared google drive
data <- read.xls(Filepath,
                 sheet = 1,
                 header = TRUE,
                 stringsAsFactor = FALSE)
#### End load raw data ####
```

```
#### Examine the raw data ####
#head(data)
#str(data) # check the class of each variable
#summary(data)
#### end raw data ####
```

```
#### clean define the missing value ####
data[data == "-"] <- NA # replace '-' with NA
data[data == ""] <- NA # replace 'missing data' with NA
#### end cleaning of data ####
```

```
#### to lower variable names ####
names(data) <- tolower(names(data))
#### end setting the varibales ####
```

```r
data <- transform(data,
                  phase = as.factor(phase),
                  fno = as.character(fno),
                  identifier = as.character(identifier),
                  country = as.factor(country),
                  year = as.factor(year),
                  season  = as.character(season),
                  lat = as.numeric(lat),
                  long = as.numeric(long),
                  village = as.character(village),
                  fa = as.numeric(fa),
                  fn = as.character(fn),
                  lfm = as.character(lfm),
                  pc = as.factor(pc),
                  fp = as.character(fp),
                  cem = as.factor(cem),
                  ast = as.factor(ast),
                  nplsqm = as.numeric(nplsqm),
                  ced = dmy(ced),# Date data try to use as.Data(., format = '%d-%b-%y') it is not worki
                  cedjul = as.numeric(cedjul),
                  hd = dmy(hd),
                  hdjul = as.numeric(hdjul),
                  ccd = as.numeric(ccd),
                  cvr = as.character(cvr),
                  vartype = as.factor(vartype),
                  varcoded = as.factor(varcoded),
                  fym = as.character(fym),
                  fym.coded = as.factor(fym.coded),
                  n = as.numeric(n),
                  p = as.numeric(p) ,
                  k = as.numeric(k),
                  mf = as.numeric(mf),
                  wcp = as.factor(wcp),
                  mu = as.character(mu) ,
                  iu = as.numeric(iu),
                  hu = as.numeric(hu),
                  fu = as.numeric(fu),
                  cs  = as.factor(cs),
                  ldg  =  as.numeric(ldg),
                  yield = as.numeric(yield) ,
                  dscum = as.factor(dscum),
                  wecum = as.factor(wecum),
                  ntmax = as.numeric(ntmax),
                  npmax = as.numeric(npmax),
                  nltmax = as.numeric(nltmax),
                  nlhmax = as.numeric(nltmax),
                  waa = as.numeric(waa),
                  wba = as.numeric(wba) ,
                  dhx =  as.numeric(dhx),
                  whx =  as.numeric(whx),
                  ssx  = as.numeric(ssx),
                  wma = as.numeric(wma),
                  lfa = as.numeric(lfa),
```

```
                  lma = as.numeric(lma),
                  rha  = as.numeric(rha) ,
                  thrx = as.numeric(thrx),
                  pmx = as.numeric(pmx),
                  defa  = as.numeric(defa) ,
                  bphx = as.numeric(bphx),
                  wbpx = as.numeric(wbpx),
                  awx  = as.numeric(awx),
                  rbx =as.numeric(rbx),
                  rbbx = as.numeric(rbbx),
                  glhx  = as.numeric(glhx),
                  stbx=as.numeric(stbx),
                  rbpx = as.numeric(rbpx),
                  hbx= as.numeric(hbx),
                  bbx = as.numeric(bbx),
                  blba = as.numeric(blba),
                  lba = as.numeric(lba),
                  bsa = as.numeric(bsa),
                  blsa = as.numeric(blsa),
                  nbsa = as.numeric(nbsa),
                  rsa  = as.numeric(rsa),
                  lsa = as.numeric(lsa),
                  shbx = as.numeric(shbx) ,
                  shrx = as.numeric(shrx),
                  srx= as.numeric(srx),
                  fsmx = as.numeric(fsmx),
                  nbx =  as.numeric(nbx),
                  dpx = as.numeric(dpx),
                  rtdx  = as.numeric(rtdx),
                  rsdx  = as.numeric(rsdx),
                  gsdx  =as.numeric(gsdx),
                  rtx = as.numeric(rtx)
)
```

```
## Warning: 1 failed to parse.
```

```
## Warning: 3 failed to parse.
```

```
## Warning: NAs introduced by coercion
```

```
##### End of type convertion ####
```

```
#### Delete the unnessary variables variables without data (NA) ####
data$phase <- NULL # there is only one type yype of phase in the survey
data$identifier <- NULL # this variable is not included in the analysis
data$village <- NULL
data$fa <- NULL # field area is not include in the analysis
data$fn <- NULL # farmer name can not be included in this survey analysis
data$fp <- NULL # I do not know what is fp
data$lfm <- NULL # there is only one type of land form in this survey
data$ced <- NULL # Date data can not be included in the network analysis
data$cedjul <- NULL
```

```r
data$hd <- NULL # Date data can not be included in the network analysis
data$hdjul <- NULL
data$cvr <- NULL
data$varcoded <- NULL # I will recode them
data$fym.coded <- NULL
data$mu <- NULL # no record
data$nplsqm <- NULL
#### Delete the unnessary variables variables without data (NA) ####
```

```r
#### Recoding the factor ####
# Previous crop
data$pc <- ifelse(data$pc == "rice", 1, 0)


#### end of recoding the factor ####
```

```r
# fym there are two type 0 and 1, raw data are recorded as no, yes, and value, if the value is 0 which r

data$fym <- ifelse(data$fym == "no", 0,
                   ifelse(data$fym == "0", 0, 1
                          )
                   )
```

```r
# vartype there are three type treditional varieties, modern varities and hybrid
data$vartype <- ifelse(data$vartype == "tv", 1,
                       ifelse(data$vartype == "mv", 2,
                              ifelse(data$vartype == "hyb", 3, NA
                                     )
                              )
                       )
```

```r
#Crop establisment method
levels(data$cem)[levels(data$cem) == "trp"] <- 1
levels(data$cem)[levels(data$cem) == "TPR"] <- 1
levels(data$cem)[levels(data$cem) == "DSR"] <- 2
levels(data$cem)[levels(data$cem) == "dsr"] <- 2
```

```r
# wcp weed control management
levels(data$wcp)[levels(data$wcp) == "hand"] <- 1
levels(data$wcp)[levels(data$wcp) == "herb"] <- 2
levels(data$wcp)[levels(data$wcp) == "herb-hand"] <- 3
```

```r
# Crop Status
levels(data$cs)[levels(data$cs) == "very poor"] <- 1
levels(data$cs)[levels(data$cs) == "poor"] <- 2
levels(data$cs)[levels(data$cs) == "average"] <- 3
levels(data$cs)[levels(data$cs) == "good"] <- 4
levels(data$cs)[levels(data$cs) == "very good"] <- 5
```

Start Explore the data

```
library(Amelia)
```

```
## Loading required package: Rcpp
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.3, built: 2014-11-14)
## ## Copyright (C) 2005-2015 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(data, main = 'Survey Data PhaseI - Missing data',
        col = c("red", "black"), legend = FALSE)
```

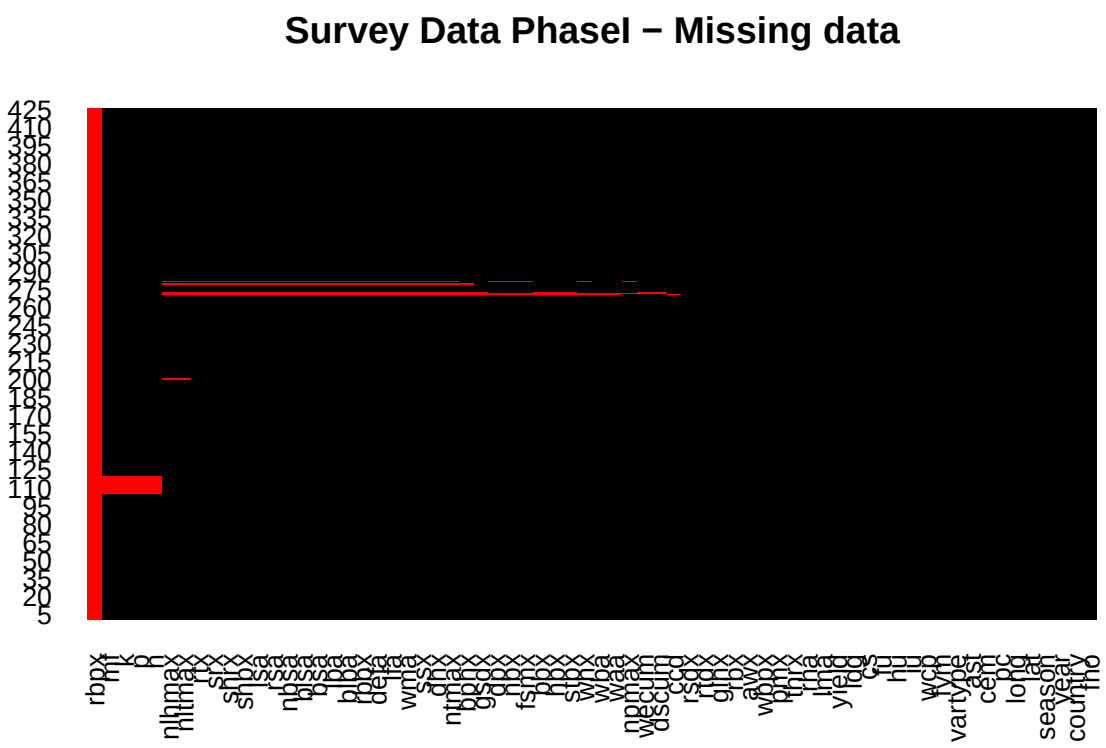
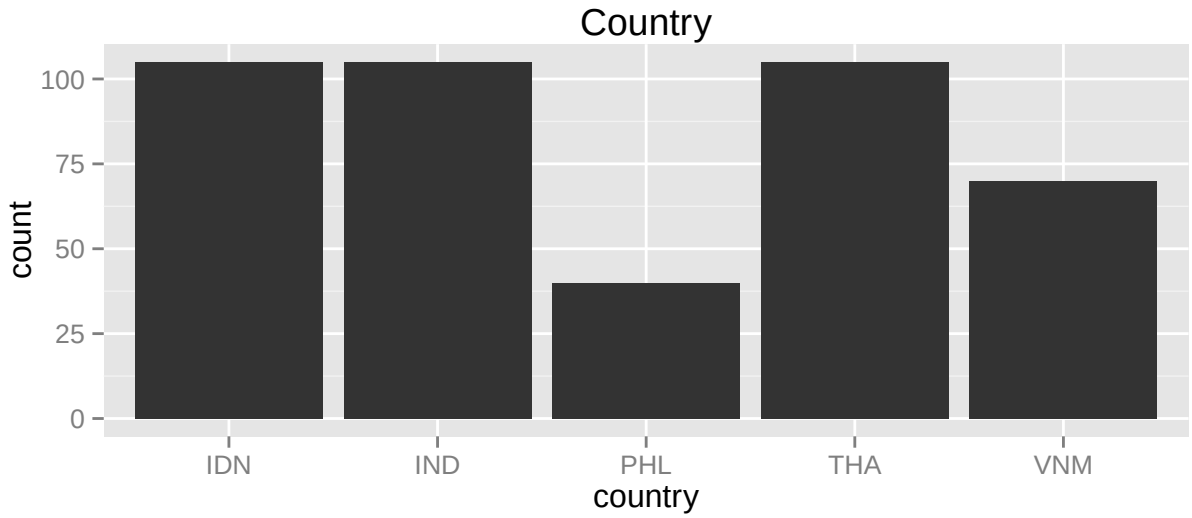


Fig 1 shows that there are a few missing data in columns,

```
# delete column
data$rbpx <- NULL
```

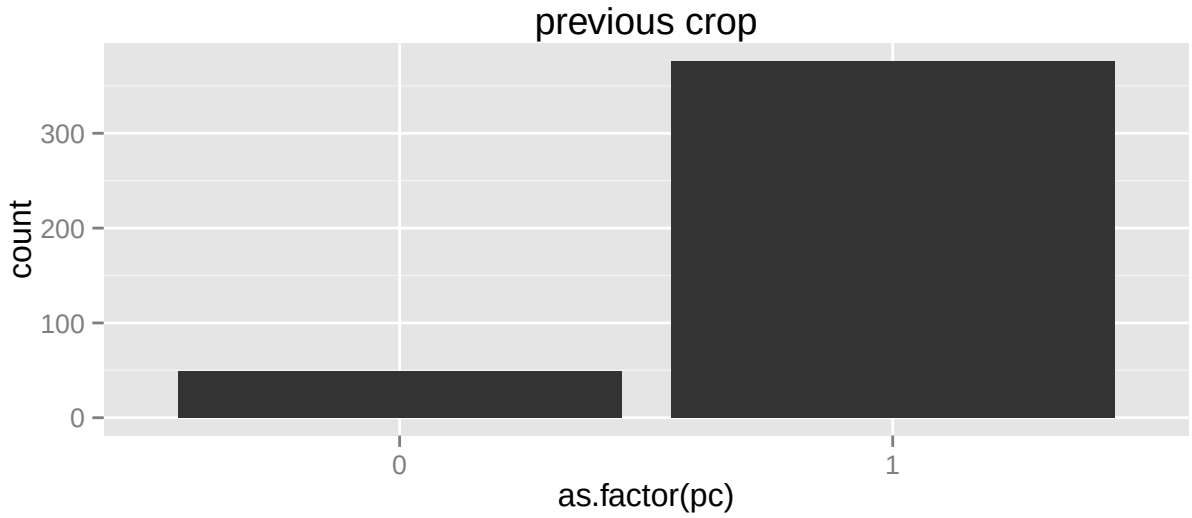
Data from Country

```
ggplot(data=data, aes(x=country)) + geom_bar() + ggtitle("Country")
```



The previous crop

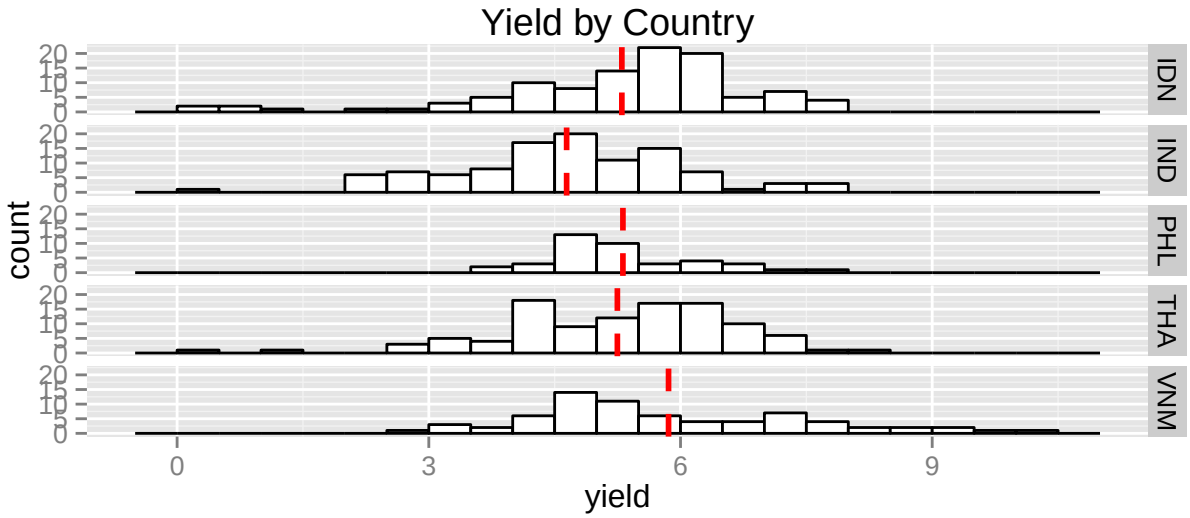
```
ggplot(data=data, aes(x = as.factor(pc))) + geom_bar(stat ="bin") + ggtitle("previous crop")
```



The data of yield

```
datYmean <- ddply(data, "country", summarise, yield.mean=mean(yield))

ggplot(data, aes(x=yield)) + geom_histogram(binwidth=.5, colour="black", fill="white") +
    facet_grid(country ~ .) +
    geom_vline(data=datYmean, aes(xintercept=yield.mean),
               linetype="dashed", size=1, colour="red") +
        ggtitle("Yield by Country")
```



Crop Establishment

```
ggplot(data=data, aes(x = as.factor(cem))) +
        geom_bar(stat ="bin") +
        ggtitle("Crop establisment")
```