Exploratory Analysis and Simple Descriptive Statistics

- Download "survey1.csv" and "Metadata_survey1.doc" from the website

- survey1.csv contains data from a large farming survey administered to farmers across various regions of Ethiopia

- The meanings of each of the 65 variables names are listed in Metadata_survey1.doc
    - E.g. the variable *dstmnsrc* means "Distance to the main source of drinking water in km"

The different types of variables

- There are **categorical** variables that describe qualitative data, and **numeric** variables that describe quantitative data.
- Gender is an example of a categorical variable, since it contains two categories with qualitative descriptions: male or female (represented by *h.size* in our data set). Another categorical variable is the name of the participant's home province.
- The categories in the above categorical variables are not ordered in any way. We call these types of categorical variables **nominal** variables. If they can be ordered, the variable is called an **ordinal** variable.
- *fmlfoodc* is a variable that ranks participants according to the security of their previous year's food supplies. There are four categories ranked from least severe to most severe (food shortage throughout the year; occasional food shortage; no food shortage but no surplus; and food surplus). There is an intrinsic ordering to these categories, so we would call *fmlfoodc* an **ordinal** variable.

- A **numeric** variable describes measurable quantities as a number, unlike categorical variables.
- **Continuous** variables are numeric variables that can take any value within a certain set of real numbers. Distance measured by a ruler is an example of a continuous variable, where the values can be as precise as the ruler allows. In our data set, *totcostfood* is an example of a continuous variable; it represents the total dollar amounts spent on food over the last 12 months.
- **Discrete** variables can take a value based on a count of distinct whole values. The number of people living in a participant's household, *hhsize*, is an example of a discrete variable. Likewise is *cattletotno* – the number of cattle owned by a participant.

- Typing in `sdata` into the RStudio console should print out the first 150 or so rows from survey1.csv.

- We can print out a list of the variable names by using the `names` command:

```
names(sdata)
```

```
> [1] "hhldid" "yearinter" "province" "dstvlgmk" [5] "dstmnmkt" "dstseed" "dstferti" "dstherbi" [9] "dstcoop" "dtsfrmgr" "dstagrex" "dtshlthc" [13] "dstmnsrc" "fmlfoodc" "hhsize" "h.educ" [17] "h.sex" "hh.males" "hh.females" "carts" [21] "bicycles" "oxploughs" "maize.pltsize.LR" "maize.pltsize.SR" [25] "Haricotbean.pltsize.LR" "Haricotbean.pltsize.SR" "legume.pltsize.LR" "legume.pltsize.SR" [29] "fert.cst" "oxplwdays.LR" "oxplwdays.SR" "oxplwdays" [33] "maize.prd.SR" "maize.prd.LR" "maize.prd" "Haricotbean.prd.SR" [37] "Haricotbean.prd.LR" "Haricotbean.prd" "legume.prd.SR" "legume.prd.LR" [41] "legume.prd" "maize.qtysld" "Haricotbean.qtysld" "legume.qtysld" [45] "maize.qtycsmd" "Haricotbean.qtycsmd" "legume.qtycsmd" "maize.qtybght" [49] "Haricotbean.qtybght" "legume.qtybght" "cattletotno" "shoatstotno" [53] "chickentotno" "equinetotno" "lvstvalue" "totmlkpdn" [57] "rectrn.cnt" "crprot.trn" "storepest.trn" "famplan.trn" [61] "fieldpest.trn" "cropres.trn" "lvstprd.trn" "maizevar.trn" [65] "legvar.trn" "totcostfood" "Zone"
```

- If you are unsure of how to use a specific function in RStudio, you can access the help file by typing a question mark (?) in the console, followed by the name of the function.

```
?names
>
```

The Names of an Object

Description
Functions to get or set the names of an object.
Usage
names(x) names(x) <- value
Arguments

x	an **R** object.
value	a character vector of up to the same length as x, or NULL.

Details
names is a generic accessor function, and names<- is a generic replacement function. The default methods get and set the "names" attribute of a vector …

… etc.

- To take a quick look of the structure of the data set, use the `head` function to print out the first few rows:

```
head(sdata)
```

- Let's now look at some of the variables within our data and explore their relationships.

- *legume.qtysld* is a quantiative variable representing the amount of legume sold in kg. We can find the mean quantity of legumes sold by using the `mean` function:

```
mean(sdata$legume.qtysold)
```

> NA

- Unfortunately this throws back NA as the mean because there are many NA's (incomplete values) contained within the *legume.qtysold* data. We can ignore the data's NA values and find the mean of the remaining numerical values by affixing `na.rm=TRUE` to the argument:

```
mean(sdata$legume.qtysld, na.rm=TRUE)
```

> 0.8297706

- We can also calculate the standard deviation of *legume.qtysld* using the `sd` function:

```
sd(sdata$legume.qtysld, na.rm=TRUE)
```

> 5.226835

- Suppose we want to investigate how participants' food supply security in the last 12 months (*fmlfoodc*) relates to the gender of the participants' household heads.

- *Fmlfoodc* is a categorical variable containing four categories (food shortage throughout the year; occasional food shortage; no food shortage but no surplus; and food surplus) and *h.sex* is a categorical variable containing two categories (male; female).

- We can print out a quick summary of each variable by using the `summary` command:

```
summary(sdata$h.sex)
>    Female   Male   NA's
     98     794         6
```

- So, across the entire study, 794 heads of households were men while only 98 were women. 6 entries were Not Applicable.

- Let’s look at a summary of the *fmlfoodc* data

```
summary(sdata$fieldpest)
```

```
> Food shortage through the year   Food surplus   No food shortage but no surplus   Occasional food shortage   NA’s
                133                     413                   308                               5              39
```

- Using the `xtabs` function, let’s make a table showing which genders belong to each category of *fmlfoodc*:

```
xtabs(~fmlfoodc + h.sex, data=sdata)
```

```
>                                    h.sex
fmlfoodc                             Female Male
  Food shortage through the year        5     34
  Food surplus                          9    123
  No food shortage but no surplus      42    368
  Occasional food shortage             41    265
```

- The “~” operator is telling RStudio to model the table on the *fmlfoodc* variable.

- Suppose we want to compare participants' food security with the number of people in each household (*hhsize*).
- First, let's use the `summary` function to summarise *hhsize:*

```
summary(sdata$hhsize)
>   Min.   1st Qu.  Median  Mean     3rd Qu.    Max.
   1.000   5.000   6.000    6.619   8.000      22.000
```

- It shows that the **minimum** number of people in a household is 1 and the **maximum** is 22.
- The **median** is 6, which means half the *hhsize* data lies above 6 and half lies below 6.
- The **1st quartile** is 5, which means 25% of the survey's participants live in families containing 1-5 people. Likewise, the **3rd quantile** is 8, so 25% of the participants live in families containing 8-22 people.

- Cross-tabulating with *fmlfoodc* with *hhsize* will produce a large, unwieldy table with many columns.

- For ease of use, we can pool the *hhsize* data into larger categories. Let's make a new variable called *hhsizefactor* that pools *hhsize* data into a group representing between 0 and 6 household members, and a group that represents houses with 6+ people – i.e., *hhsize* was a discrete, quantitative variable, but *hhsizefactor* is a categorical variable.

```
hhsizefactor <- cut(sdata$hhsize,
breaks=c(0,6,Inf),labels=c("0 to 6
members", "6+ members"))
```

- Cross-tabulating *fmlfoodc* with *hhsizefactor* will give us a general idea of how the *fmlfoodc* categories relate to household sizes:

```
xtabs(~fmlfoodc + hhsizefactor,
data=sdata)
```

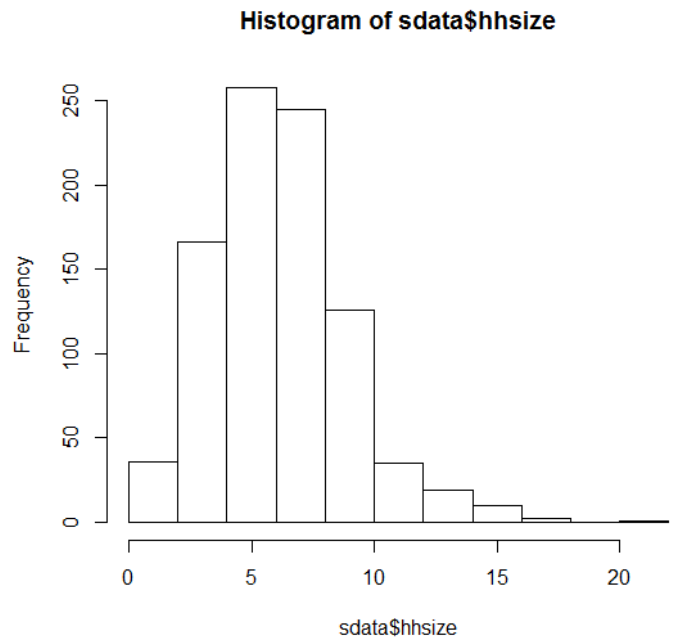
```
>                                   hhsizefactor
fmlfoodc                                 0 to 6 members  6+members
  Food shortage through the year                 20          19
  Food surplus                                   61          72
  No food shortage but no surplus               217         196
  Occasional food shortage                      160         148
```

- Transforming the quantitative variable *hhsize* into the categorical variable *hhsizefactor* can be a good idea if you want a visual representation of your data in the form of a table.

- Histograms are a better graphic to use when you need a visual representation of a quantitative variable. Histograms show how many times particular values of the quantitative variable are recorded at different data points, and so looking at a histogram will give you a good idea of how the data is distributed.

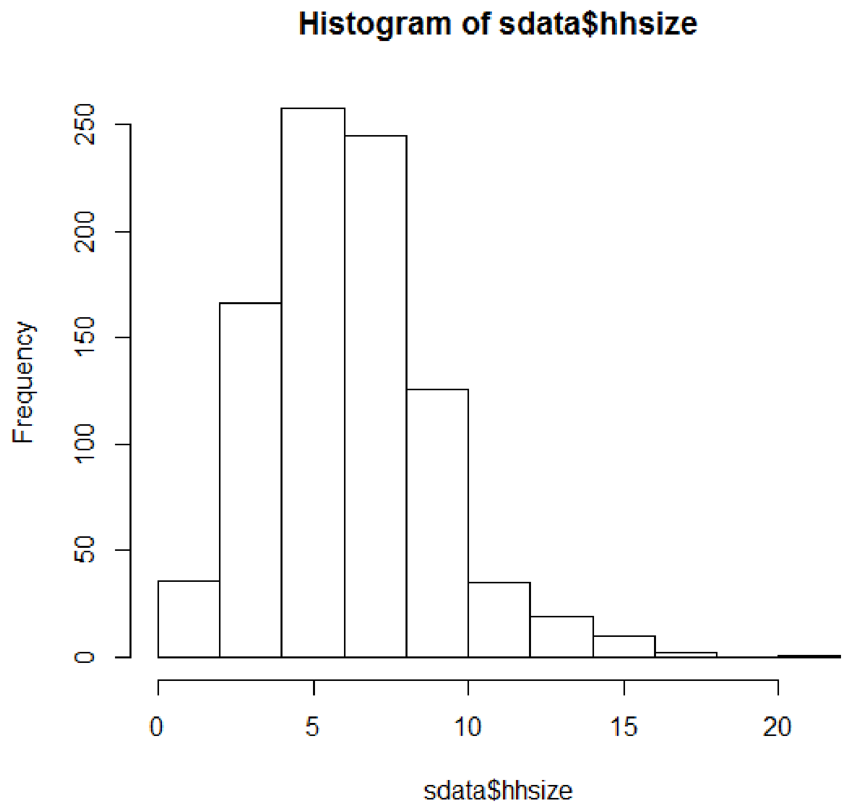
- Using the function `hist`, we can create a a histogram for *hhsize* (number of people per household):

```
hist(sdata$hhsize)
```



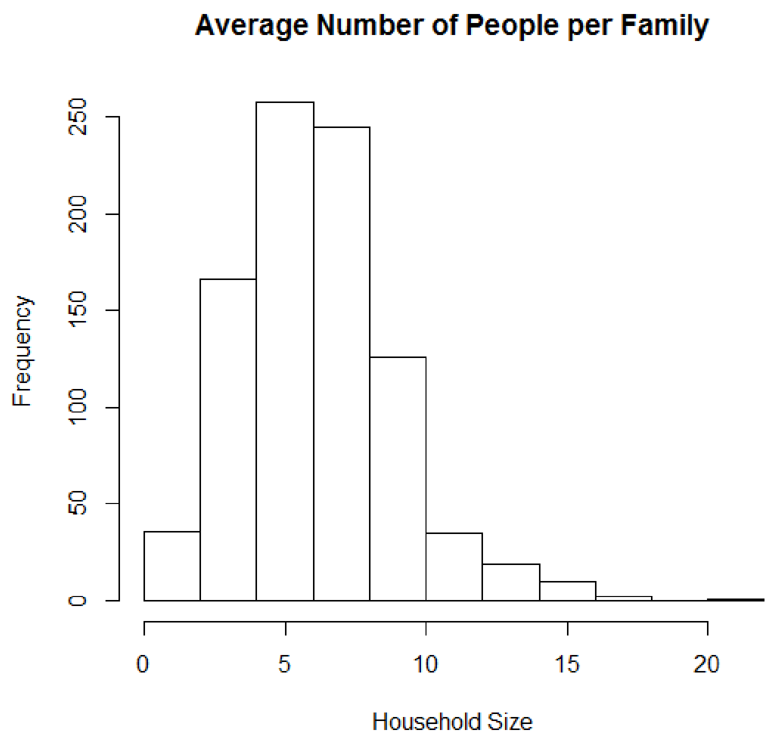
- Using the function `hist`, we can create a a histogram for *hhsize* (number of people per household):

```
hist(sdata$hhsize)
```



- We can change the title and x label of the histogram using the `main` and `x lab` functions:

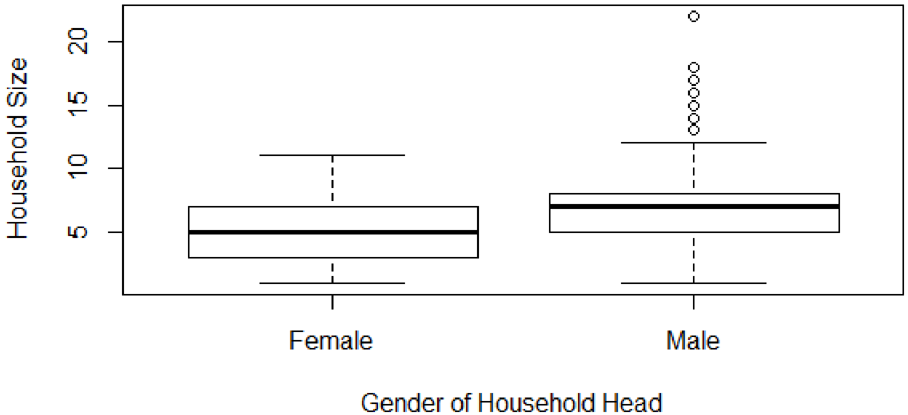
```
hist(sdata$hhsize, main="Average Number of People per Family", xlab="Household Size")
```



- A boxplot is a visual representation of the distribution of a variable using its five number summary (minimum value, 1st quantile (Q1) , median, 3rd quantile (Q3), and maximum value).

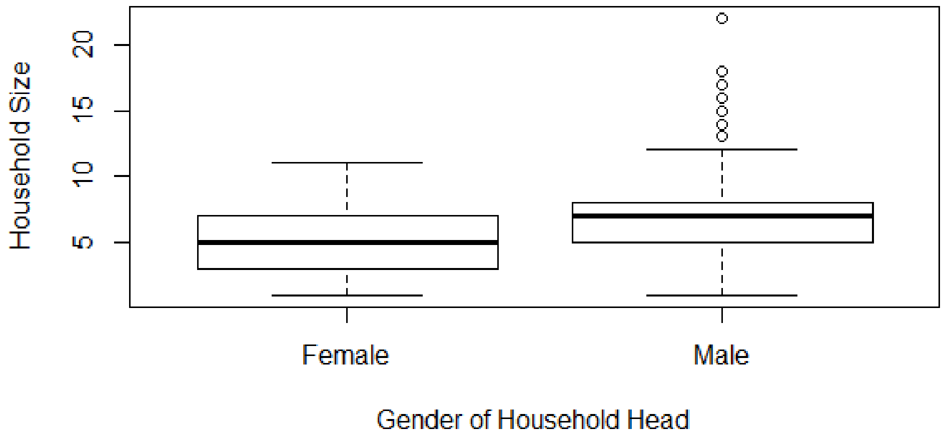
- Using the `boxplot` command, we can create boxplots of *hhsize* for each household gender (*h.sex*):

```
boxplot(hhsize~h.sex, xlab=
“Gender of Household Head”
ylab="Household Size“,
data=sdata)
```



- In a boxplot, the solid black line represents the median, and the top and bottom edges of the box represent Q3 and Q1, respectively.

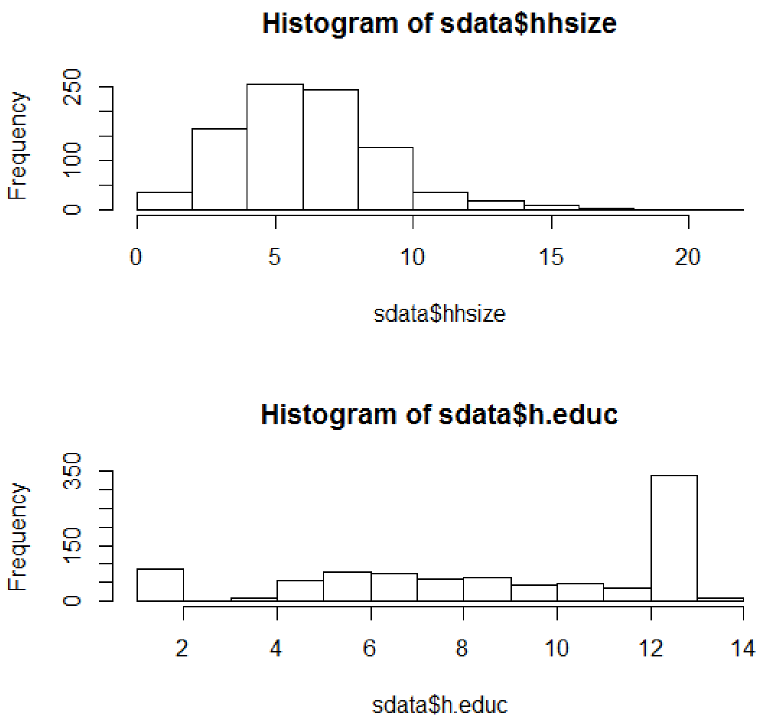
- The interquartile range is defined as IQR = Q3 – Q1. If a value lies in a range outside of 1.5 x IQR, we call that valuean outlier. The arms of a boxplot stretch to the maximum and minimum values, excluding the outliers. You can see that there are several male household heads whose family sizes are shown to be outliers



- Suppose we'd like to visually compare the distribution of household sizes to the distribution of education level (*h.educ*) of the household head, where education level is measured in years.

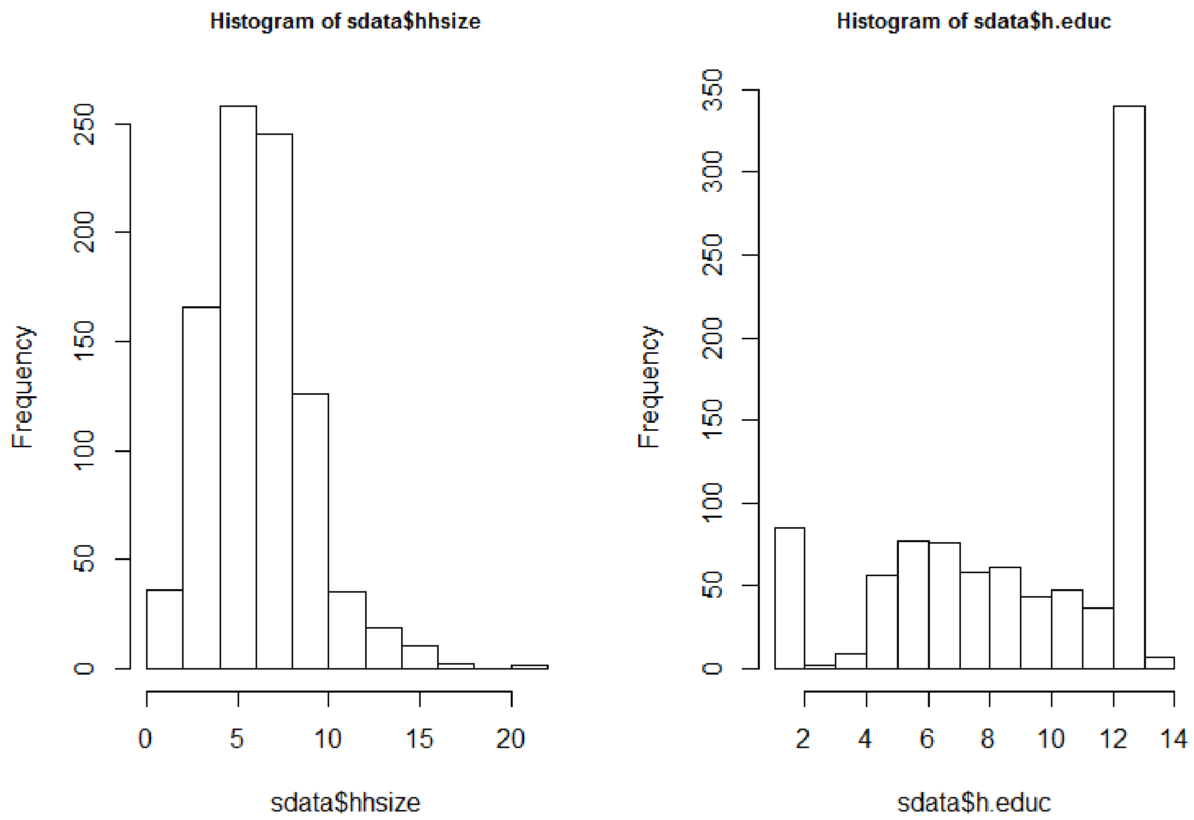
- We can use the `par` and `mfrow` functions to arrange the histograms of *hhsize* and *h.educ* to be neatly arranged on top of each other:

```
par(mfrow=c(2,1))
hist(sdata$hhsize)
hist(sdata$h.educ)
```



- … Or they can be arranged to be beside each other

```
par(mfrow=c(1,2))
hist(sdata$hhsize)
hist(sdata$h.educ)
```



- To make the two histograms more comparable, we can change the size of the bins (`breaks`) as well as the x limits (`x lim`) and y limits (`y lim`):

```
par(mfrow=c(1,2))
hist(sdata$hhsize, breaks=20, xlim=c(0,20), ylim=c(0,350))
hist(sdata$h.educ, xlim=c(0,20), ylim=c(0,350))
```