# 2.2 Explicitly Defined Population Attributes

## Contents


**2.2.1 Population Attributes** **2**
Location Attributes . . . 3
Spread Attributes . . . 3
Order Statistics . . . 4
Location Attributes based on Order Statistics . . . 5
Variability Attributes based on Order Statistics . . . 5
Skewness Attributes . . . 6
Agriculture Data Example . . . 6
NAs in R . . . 9

**2.2.2 Attribute Properties** **9**
Location Invariance and Equivariance . . . 10
Example . . . 10
Scale Invariance and Equivariance . . . 10
Example . . . 10
Replication . . . 11
Example . . . 11

**2.2.3 Influence, Sensitivity Curves and Breakdown Points** **11**
Influence . . . 11
Example . . . 12
Sensitivity Curve . . . 13
Example: Arithmetic Mean . . . 14
Example: Maximum . . . 16
Example: $2^{nd}$ Order Statistic . . . 17
Example: $3^{rd}$ Quantile . . . 18
Example: Median . . . 19
Example: Median & Mean SC Plot . . . 19
Breakdown Points . . . 20

**2.2.4 Graphical Attributes** **21**
Histograms . . . 21
Rules for the Number of Bins . . . 23
Scatter-plots . . . 24
Fire Emblem Heroes . . . 26

**2.2.5 Power Transformations** **31**
How to pick $\alpha$? . . . 34
Bump Rule 1: Making histograms more symmetric . . . 35
Bump Rule 2: Straightening Scatter-plots . . . 35

**2.2.6 Order and Rank Statistics** **41**
Example . . . 42
Scatter-plot for $y_u$ vs. rank . . . 42

**2.2.7 Quantiles** **43**
Quantiles that measure center . . . 44
Quantiles that measure Spread . . . 45
Concentration in Quantile Plots . . . 46

"Statistics is: the fun of finding patterns in data; the pleasure of making discoveries; the import of deep philosophical questions; the power to shed light on important decisions, and the ability to guide decisions... in business, science, government, medicine, and industry..."

– Professor David Hand

## 2.2.1 Population Attributes

- The population is typically a set or collection of units, each with one or more variates that we can measure.
- **Variates** are characteristics of each unit in the population, and they can take on numerical or categorical values.
  - The values of variates typically differ from unit to unit.
  - If we are only interested in the variate $y$s we might write:

$$\mathcal{P} = \{y_1, y_2, \ldots, y_N\}$$

- **Population attributes** are summaries describing characteristics of the population.
  - Formally, an attribute is a function applied to the entire population and determined by the variate values observed for each of the population's units.

$$a(\mathcal{P}) = f(y_1, y_2, \ldots, y_N)$$

- Some examples of attributes are
  - the population total:

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$$

  - or various counts over the population

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} I_A(y_u)$$

Ex: $y$ = height (inches)
$A$ = heights ($y$-values) larger than 72"

where $I_A(y)$ is the indicator function

$$I_A(y) = \begin{cases} 1 & y \in A \\ 0 & y \notin A \end{cases}$$

$I_A(y) = \begin{cases} 1 & \text{if } y > 72 \\ 0 & \text{if } y \leq 72 \end{cases}$

  - Example: Let $\mathcal{P} = \{1, 2, ..., 100\}$ and calculate $a(\mathcal{P})$ : the total number of units which are multiple of 10.

$I_A(y) = \begin{cases} 1 & \text{if } y \bmod 10 = 0 \\ 0 & \text{if } y \bmod 10 \neq 0 \end{cases}$

$a(P) = \sum_{u \in P} I_A(y) = 10$

- In general, attributes can be numerical or graphical – as long as they summarize the whole population.
  - A histogram of $y_u$ values
  - A scatter-plot of the $(x_u, y_u)$ pairs
  - The least squares estimate of the line-of-best-fit
  - The residual variation around the line-of-best-fit

## Location Attributes

These attributes measure or describe the centre of the distribution of variate values in a dataset.

- the population average:
$$a(\mathcal{P}) = \overline{y} = \frac{1}{N}\sum_{u\in\mathcal{P}} y_u$$
- the population proportion:
$$a(\mathcal{P}) = \frac{1}{N}\sum_{u\in\mathcal{P}} I_A\left(y_u\right)$$
- Other examples include the mode, the median, etc.

## Spread Attributes

These attributes measure variability or spread of the variate values in a data set. Some are

- the population variance:
$$a(\mathcal{P}) = \frac{1}{N}\sum_{u\in\mathcal{P}} \left(y_u - \overline{y}\right)^2$$
- the population standard deviation:
$$a(\mathcal{P}) = SD_{\mathcal{P}}(y) = \sqrt{\frac{\sum_{u\in\mathcal{P}} \left(y_u - \overline{y}\right)^2}{N}}$$
- coefficient of variation:
$$a(\mathcal{P}) = \frac{SD_{\mathcal{P}}(y)}{\overline{y}}$$
- **Note**: the population variance or standard deviation could also be defined using $N-1$ in the denominator.
- Other examples are the range, the inter-quartile range, etc.

## Order Statistics

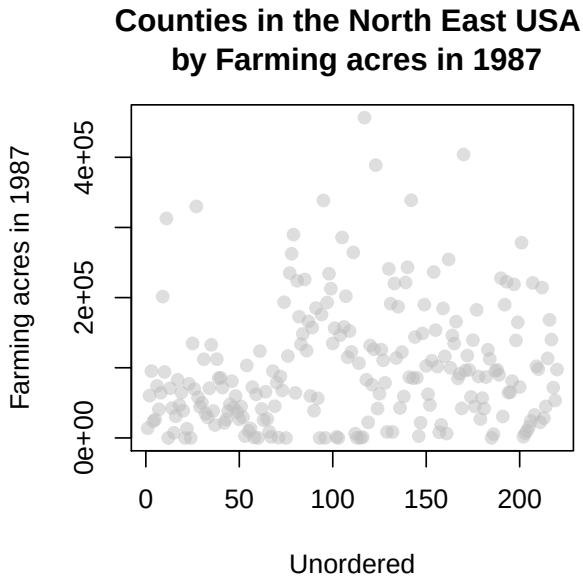
- Population attributes can also be based on an indexed collection of values,

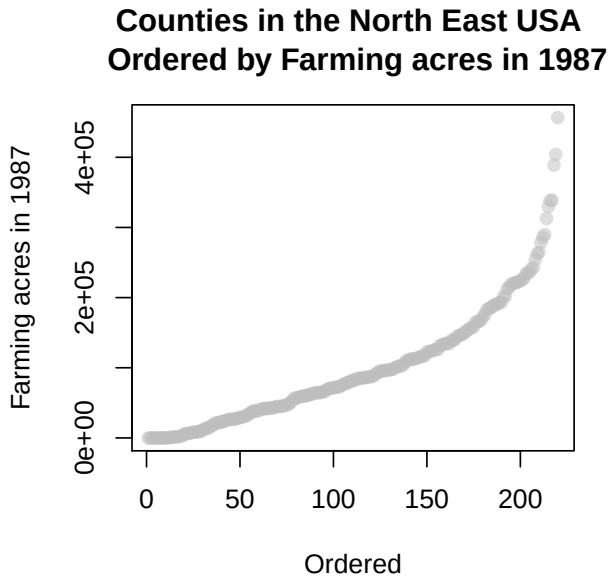
$$y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(N)}$$

  which are the the variate values $y_u \in \mathcal{P}$ ordered from smallest to largest (including ties).

- This ordering can be informative, for example:

```
### read the data from wherever it is stored, e.g. in some directory named Data
directory <- "../Data"
dirsep <-"/"
filename <- paste(directory, "agpop_data.csv", sep=dirsep)
agpop <- read.csv(filename, header=TRUE)

par(mfrow=c(1,2))
y <- agpop$acres87[agpop$region == "NE"]
y = na.omit(y)


plot(y, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlab = "Unordered",
     ylab = "Farming acres in 1987",
     main = "Counties in the North East USA \n by Farming acres in 1987")

yordered <- sort(y)
yrank <- rank(y, ties.method = "first")  # Now ensure ties appear in data set order

plot(yrank, y, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlab = "Ordered",
     ylab = "Farming acres in 1987",
     main = "Counties in the North East USA \n Ordered by Farming acres in 1987")
```



**Location Attributes based on Order Statistics**

- the population minimum:
$$a(\mathcal{P}) = \min_{u \in \mathcal{P}} y_u = y_{(1)}$$
- the population maximum:
$$a(\mathcal{P}) = \max_{u \in \mathcal{P}} y_u = y_{(N)}$$
- the population mid-range:
$$a(\mathcal{P}) = \frac{1}{2}\left[\min_{u \in \mathcal{P}} y_u + \max_{u \in \mathcal{P}} y_u\right] = \frac{y_{(1)} + y_{(N)}}{2}$$
- the population median:
$$a(\mathcal{P}) = \operatorname*{median}_{u \in \mathcal{P}} y_u = \begin{cases} y_{([N+1]/2)} & \text{if } N \text{ is odd} \\ \frac{y_{(N/2)} + y_{(N/2+1)}}{2} & \text{if } N \text{ is even} \end{cases}$$
- the population quartiles:
  - $Q_1$ is $25^{th}$ percentile, or the first quartile,
  - $Q_2$ is $50^{th}$ percentile, or the median, and
  - $Q_3$ is $75^{th}$ percentile, or the third quartile.

**Variability Attributes based on Order Statistics**

- the population range:
$$a(\mathcal{P}) = \max_{u \in \mathcal{P}} y_u - \min_{u \in \mathcal{P}} y_u = y_{(N)} - y_{(1)}$$
- the population inter-quantile range IQR:
$$a(\mathcal{P}) = Q_3 - Q_1$$
where $Q_1$ and $Q_3$ are $25^{th}$ and $75^{th}$ percentiles or the first and third quartiles, as above.
- the Median Absolute Deviation (MAD) is the median of the absolute differences between each $y_u$ and the median:
$$a(\mathcal{P}) = \operatorname*{median}_{u \in \mathcal{P}} \left| y_u - \operatorname*{median}_{u \in \mathcal{P}} y_u \right|$$

## Skewness Attributes

These are measures of asymmetry in a population. A symmetric distribution of population values should result in a skewness attribute of zero.

- Pearson's moment coefficient of Skewness:

$$a(\mathcal{P}) = \frac{\frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \overline{y})^3}{[SD_{\mathcal{P}}(y)]^3}$$

- Pearson's second skewness coefficient (median skewness) given by

$$a(\mathcal{P}) = 3 \times \frac{(\overline{y} - \text{median}_{u\in\mathcal{P}}\, y_u)}{SD_{\mathcal{P}}(y)}$$

- Bowley's measure of skewness based on the quartiles:

$$a(\mathcal{P}) = \frac{(Q_3 + Q_1)/2 - Q_2}{(Q_3 - Q_1)/2}$$

Quartile analog of

## Agriculture Data Example

Read the data from wherever it is stored, e.g. in some directory named Data

```
directory <- "../Data"
dirsep <-"/"
filename <- paste(directory, "agpop_data.csv", sep=dirsep)
agpop <- read.csv(filename, header=TRUE)
```

A number of population attributes can be calculated via summary(agpop)

```
summary(agpop)
```

```
##                county         state         acres92
##  WASHINGTON COUNTY:  30   TX     : 254   Min.   :    -99
##  JEFFERSON COUNTY :  25   GA     : 159   1st Qu.:  80903
##  FRANKLIN COUNTY  :  24   KY     : 120   Median : 191648
##  JACKSON COUNTY   :  23   MO     : 114   Mean   : 306677
##  LINCOLN COUNTY   :  23   KS     : 105   3rd Qu.: 366886
##  MADISON COUNTY   :  19   IL     : 102   Max.   :7229585
##  (Other)          :2934   (Other):2224
##     acres87           acres82           farms92          farms87
##  Min.   :    -99   Min.   :    -99   Min.   :   0.0   Min.   :   0.0
##  1st Qu.:  86236   1st Qu.:  96397   1st Qu.: 295.0   1st Qu.: 318.5
##  Median : 199864   Median : 207292   Median : 521.0   Median : 572.0
##  Mean   : 313016   Mean   : 320194   Mean   : 625.5   Mean   : 678.3
##  3rd Qu.: 372224   3rd Qu.: 377065   3rd Qu.: 838.0   3rd Qu.: 921.0
##  Max.   :7687460   Max.   :7313958   Max.   :7021.0   Max.   :7590.0
##
##     farms82          largef92         largef87         largef82
```

```
##  Min.   :   0.0   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 345.0   1st Qu.:  8.00   1st Qu.:  8.00   1st Qu.:  8.00
##  Median : 616.0   Median : 30.00   Median : 27.00   Median : 25.00
##  Mean   : 728.1   Mean   : 56.18   Mean   : 54.86   Mean   : 52.62
##  3rd Qu.: 991.0   3rd Qu.: 75.00   3rd Qu.: 70.00   3rd Qu.: 65.00
##  Max.   :7394.0   Max.   :579.00   Max.   :596.00   Max.   :546.00
##
##     smallf92          smallf87          smallf82       region
##  Min.   :   0.00   Min.   :   0.00   Min.   :   0.00   NC:1054
##  1st Qu.:  13.00   1st Qu.:  17.00   1st Qu.:  16.00   NE: 220
##  Median :  29.00   Median :  35.00   Median :  34.00   S :1382
##  Mean   :  54.09   Mean   :  59.54   Mean   :  60.97   W : 422
##  3rd Qu.:  59.00   3rd Qu.:  67.00   3rd Qu.:  67.00
##  Max.   :4298.00   Max.   :3654.00   Max.   :3522.00
##
```

- The first two variates (`county` and `state`) are categorical
  - because there are more than six values, only the six most frequent values are shown
- The last variate `region`, only takes on four different values (`NC`, `NE`, `S`, `W`) so each count appears.

```
summary(agpop[,c(1,2,15)])
```

```
##                 county          state      region
##  WASHINGTON COUNTY:  30   TX     : 254   NC:1054
##  JEFFERSON COUNTY :  25   GA     : 159   NE: 220
##  FRANKLIN COUNTY  :  24   KY     : 120   S :1382
##  JACKSON COUNTY   :  23   MO     : 114   W : 422
##  LINCOLN COUNTY   :  23   KS     : 105
##  MADISON COUNTY   :  19   IL     : 102
##  (Other)          :2934   (Other):2224
```

- The remaining variates are numeric, so the summary provides
  - the average (`mean`)
  - the minimum (`min`) and maximum (`max`)
  - the first and third quartiles
  - the median (`median`)

```
summary(agpop[,-c(1,2,15)])
```

```
##     acres92           acres87           acres82          farms92
##  Min.   :    -99   Min.   :    -99   Min.   :    -99   Min.   :   0.0
##  1st Qu.:  80903   1st Qu.:  86236   1st Qu.:  96397   1st Qu.: 295.0
##  Median : 191648   Median : 199864   Median : 207292   Median : 521.0
##  Mean   : 306677   Mean   : 313016   Mean   : 320194   Mean   : 625.5
##  3rd Qu.: 366886   3rd Qu.: 372224   3rd Qu.: 377065   3rd Qu.: 838.0
##  Max.   :7229585   Max.   :7687460   Max.   :7313958   Max.   :7021.0
##     farms87          farms82          largef92         largef87
##  Min.   :   0.0   Min.   :   0.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 318.5   1st Qu.: 345.0   1st Qu.:  8.00   1st Qu.:  8.00
##  Median : 572.0   Median : 616.0   Median : 30.00   Median : 27.00
##  Mean   : 678.3   Mean   : 728.1   Mean   : 56.18   Mean   : 54.86
##  3rd Qu.: 921.0   3rd Qu.: 991.0   3rd Qu.: 75.00   3rd Qu.: 70.00
##  Max.   :7590.0   Max.   :7394.0   Max.   :579.00   Max.   :596.00
##     largef82         smallf92          smallf87          smallf82
```

```
##  Min.   :  0.00   Min.   :   0.00   Min.   :   0.00   Min.   :   0.00
##  1st Qu.:  8.00   1st Qu.:  13.00   1st Qu.:  17.00   1st Qu.:  16.00
##  Median : 25.00   Median :  29.00   Median :  35.00   Median :  34.00
##  Mean   : 52.62   Mean   :  54.09   Mean   :  59.54   Mean   :  60.97
##  3rd Qu.: 65.00   3rd Qu.:  59.00   3rd Qu.:  67.00   3rd Qu.:  67.00
##  Max.   :546.00   Max.   :4298.00   Max.   :3654.00   Max.   :3522.00
```

- Looking at the number of acres devoted to farms (i.e. `acres92`, `acres87`, `acres82`) reveals something curious
  - the minimum of each is `-99` which is a strange value for the number of acres!
  - No acreage should be less than zero.
  - Missing data are encoded as `-99` in this data set.
  - These should be replaced by `NA` which is the standard representation for missing data in `R`.

```
##     acres92           acres87           acres82
##  Min.   :    -99   Min.   :    -99   Min.   :    -99
##  1st Qu.:  80903   1st Qu.:  86236   1st Qu.:  96397
##  Median : 191648   Median : 199864   Median : 207292
##  Mean   : 306677   Mean   : 313016   Mean   : 320194
##  3rd Qu.: 366886   3rd Qu.: 372224   3rd Qu.: 377065
##  Max.   :7229585   Max.   :7687460   Max.   :7313958
```

Having encoded the missing values as `NA`, the summary of these variates will now reflect the changes.

```
### which values are missing can be determined with a logical query
missing92 <- agpop[,"acres92"] == -99
### missing92 is a logical vector of the same length
### as agpop[,"acres92"] containing a TRUE in every
### position where a -99 appeared and FALSE everywhere else.
### The total number of missing values can be had by
### summing (because logical TRUE is treated as 1, and FALSE as 0)
#sum(missing92)

### Alternatively, the `which` function could be used
### to identify the row numbers
rowNumsMissing <- which(agpop[,"acres92"] == -99)

### The values can be changed to NA by using these locations
### (either rowNumsMissing or missing92) to identify the rows
### and replace the values
agpop[missing92, "acres92"] <- NA

### The same can be done for the other two acreages
agpop[agpop[,"acres87"] == -99, "acres87"] <- NA
agpop[agpop[,"acres82"] == -99, "acres82"] <- NA
```

```
summary(agpop[,c("acres92", "acres87", "acres82")])
```

```
##     acres92           acres87           acres82
##  Min.   :      0   Min.   :      0   Min.   :      0
##  1st Qu.:  82446   1st Qu.:  87530   1st Qu.:  97835
##  Median : 193688   Median : 201728   Median : 209222
##  Mean   : 308582   Mean   : 315374   Mean   : 321973
##  3rd Qu.: 368482   3rd Qu.: 374576   3rd Qu.: 379172
##  Max.   :7229585   Max.   :7687460   Max.   :7313958
```

```
##  NA's   :19        NA's   :23        NA's   :17
```

### NAs in R

- **Note** that many programs in R accommodate missing data (represenated as NAs) and do something appropriate (typically they omit them).
  - For your own code and analyses, you either need to decide what to do with NAs or ensure that the data do not have any NAs.
  - If you choose to simply omit NAs, for example, the function `na.omit(...)` may be helpful (it will remove rows which contain an NA from a data set). For other possibilities see `help("na.omit")` in R.

or ? na.omit

## 2.2.2 Attribute Properties

- A population attribute is a function of measured variates $y_u$:

$$a(\mathcal{P}) = f(y_1, y_2, \ldots, y_N)$$

  and the variates $y_u$ are typically associated with some measurement units. In general we are interested in understanding how an attribute changes
  - when we change the units of measurement, and
  - when we change the population.
- Sometimes only the **scale** of measurement is changed:
  - 1 yard = 3 feet; 1 mile = 5280 feet; 1 metre = 1000 mm;
  - 1 imperial gallon = 4.54609 litres; 1 US gallon = 0.832674 imperial gallon;
  - 1 kilogram = 1000 grams = 2.20462 pounds.
- Sometimes only the **location** of the zero for that measurement is changed:
  - absolute zero is 0° Kelvin or −273° Celsius
  - 1 Celsius degree = 1 Kelvin degree (no change in scale of measurement)
- Sometimes both the **location** and the **scale** change:
  - water freezes at 0° Celsius = 32° Fahrenheit (location change)
  - 1 Celsius degree = 1.8 Fahrenheit degrees (scale change)
- Sometimes the change involves more than just a change in location and/or scale of measurement:
  - fuel economy might be reported in miles per gallon (US or Imperial) or litres per hundred kilometres.
  - the Richter scale for earthquakes is a logarithmic measure of the amplitude of seismic waves.

- Running speed: miles per hour vs. minutes per km

## Location Invariance and Equivariance

For an attribute $a(\mathcal{P}) = a(y_1, \ldots, y_N)$ we say that for any $m > 0$ and $b \in \mathbb{R}$, that the attribute is

- **location invariant** if
$$a(y_1 + b, \ldots, y_N + b) = a(y_1, \ldots, y_N)$$
- **location equivariant** if
$$a(y_1 + b, \ldots, y_N + b) = a(y_1, \ldots, y_N) + b$$

**Example**

The population average is location equivariant.

$$a(\mathcal{P}) = a(y_1, \ldots, y_N) = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$a(y_1+b, \ldots, y_N+b) = \frac{1}{N}\sum_{i=1}^{N}(y_i+b) = \frac{1}{N}\sum_{i=1}^{N} y_i + \frac{Nb}{N} = a(\mathcal{P}) + b$$

## Scale Invariance and Equivariance

For an attribute $a(\mathcal{P}) = a(y_1, \ldots, y_N)$ we say that for any $m > 0$ and $b \in \mathbb{R}$, that the attribute is

- **scale invariant** if
$$a(m \times y_1, \ldots, m \times y_N) = a(y_1, \ldots, y_N)$$
- **scale equivariant** if
$$a(m \times y_1, \ldots, m \times y_N) = m \times a(y_1, \ldots, y_N)$$
- **location-scale invariant** if it is both location invariant and scale invariant, i.e.
$$a(m \times y_1 + b, \ldots, m \times y_N + b) = a(y_1, \ldots, y_N)$$
- **location-scale equivariant** if it is both location equivariant and scale equivariant, i.e.
$$a(m \times y_1 + b, \ldots, m \times y_N + b) = m \times a(y_1, \ldots, y_N) + b$$

**Example**

The population average is location-scale equivariant

$$a(my_1+b, \ldots, my_N+b) = \frac{1}{N}\sum_{i=1}^{N}(my_i+b)$$

$$= \frac{m}{N}\sum_{i=1}^{N} y_i + \frac{Nb}{N} = ma(\mathcal{P}) + b$$

### Replication

- Another invariance/equivariance property of interest for population attributes is **replication invariance** and **replication equivariance**.
  - If a population $\mathcal{P}$ is duplicated $k-1$ times (so that there are $k$ copies of it), how does the attribute change on this new population denoted by $\mathcal{P}^k$?

$$\mathcal{P}^k = \{\underbrace{y_1, \ldots, y_N}_{k \text{ copies}}, \underbrace{y_1, \ldots, y_N}, \ldots, \underbrace{y_1, \ldots, y_N}\} \equiv \{x_1, x_2, \ldots, x_{kN}\}$$

- The attribute $a(\mathcal{P})$ is
  - **replication invariant** whenever $a(\mathcal{P}^k) = a(\mathcal{P})$ and
  - **replication equivariant** whenever $a(\mathcal{P}^k) = k \times a(\mathcal{P})$.

#### Example

The population average is replication invariant.

$$a(\mathcal{P}^k) = \frac{1}{Nk} \sum_{j=1}^{Nk} x_j = \frac{1}{Nk} \sum_{i=1}^{N} k y_i = \frac{1}{N} \sum_{i=1}^{N} y_i = a(\mathcal{P})$$

## 2.2.3 Influence, Sensitivity Curves and Breakdown Points

- If we wish to evaluate a population attribute

$$a(\mathcal{P}) = a(y_1, \ldots, y_N)$$

  we could consider the effect of adding or removing a single variate $y_u$ to examine its impact on $\mathcal{P}$.
- To quantify this effect, we could look at the difference in the attribute when the variate value is added or removed.

### Influence

- If we remove variate $y_u$, (i.e. remove unit $u$) then the **influence** of that variate on the population attribute is quantified by

$$\Delta(a, u) = a(y_1, \ldots, y_{u-1}, y_u, y_{u+1}, \ldots, y_N) - a(y_1, \ldots, y_{u-1}, y_{u+1}, \ldots, y_N)$$

  with unit $u$ — without unit $u$
- Ideally, no single unit's value should have greater influence than any other.
- If a unit had larger influence than the rest;
  - (1) – it would require further investigation as it might be in error, or
  - (2) – it might be the most interesting unit in the population.

**Example**

From the agricultural census, we can examine the number of farms per county in 1987.

The population average, $a(y_1, \ldots, y_N) = \overline{y}$, is

```
y    = agpop$farms87
ybar = mean(y)
ybar
```

```
## [1] 678.2843
```

and the average without unit $u$ can be written as

$$a(y_1, \ldots, y_{u-1}, y_{u+1}, \ldots, y_N) = \frac{1}{N-1} \sum_{k \in \mathcal{P}, k \neq u} y_k = \frac{\sum_{k \in \mathcal{P}} y_k - y_u}{N-1} = \frac{N\overline{y} - y_u}{N-1}$$

and $\Delta(a, u)$, the influence for a given $u$, is:

$$\Delta(a, u) = \overline{y} - \frac{N\overline{y} - y_u}{N-1} = \frac{(N-1)\overline{y} - (N\overline{y} - y_u)}{N-1} = \frac{y_u - \overline{y}}{N-1}$$

We can use R to calculate this quantity in many ways.

- Using a loop and calculating the $\Delta$:

```
delta = rep(0, length(y))
for (i in 1:length(y)) {
  ## y[-i] removes the ith element from a vector
  delta[i] = ybar - mean(y[-i])
}
```

- Creating matrix of populations of size $N-1$ and using the `apply` function:

```
N = length(y)
popN_1 = matrix(rev(y), nrow=N-1, ncol=N)
attN_1 = apply(popN_1, 2, mean)
delta = ybar - attN_1
```

- Using a loop and the simplified expression:

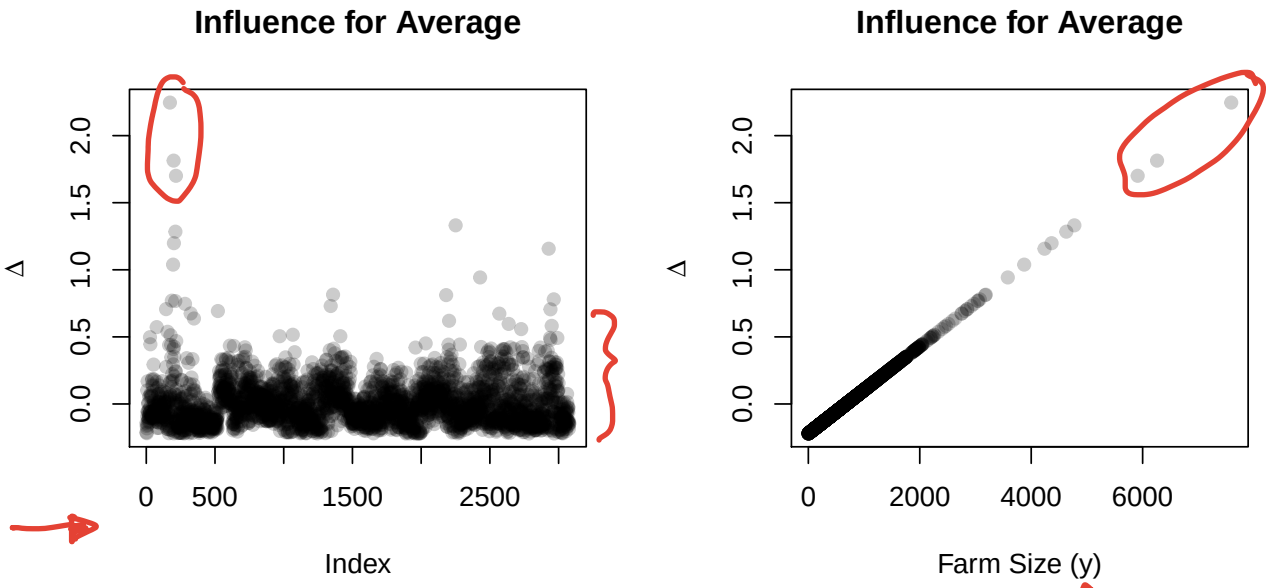
```
delta = rep(0, length(y))
for (i in 1:length(y)) {
  delta[i] = (y[i]-ybar)/(length(y)-1)
}
```

- Summing two numeric vectors:
  - Note: when a vector $\mathbf{v}$ of dimension $N$ is added to a scalar $c$, R converts the scalar into $c\mathbf{1}_N$

```
delta = (y-ybar)/(length(y)-1)
```

To summarize the influece for every unit $u$ we might plot the influence ($\Delta$ values) by observation number or by $y$.

```r
y = agpop$farms87
N = length(y)
delta = sum(y)/N - (sum(y)-y)/(N-1)

par(mfrow=c(1,2))
plot(delta, main="Influence for Average", pch=19,
     col=adjustcolor("black", alpha = 0.2),
     xlab  = "Index",
     ylab = bquote(Delta))
plot(y, delta, main="Influence for Average", pch=19,
     col=adjustcolor("black", alpha = 0.2),
     xlab='Farm Size (y)',
     ylab = bquote(Delta))
```




- There is, at least, one (if not a few) counties whose farmsizes seem to be more influential on the population average compared to other counties.
- These counties are, as expected (why?), those with larger farm sizes.
  - it turns out that the three highest influence points are observation numbers 172 (Fresno), 199 (San Diego), and 216 (Tulare) with farm sizes 7590, 6259, and 5911 acres, respectively. All these counties are in California.
  - These are the three most extreme values in the right panel.

which(delta > 1.5) → this would give the indices of the three units with the largest delta values.

## Sensitivity Curve

- We can also examine the effect on attribute when we add a variate. To examine this effect,
  - suppose we have a population of size $N-1$ and
  - add a variate with the value $y$.
  - Then our new population with $N$ elements is $\{y_1, \ldots, y_{N-1}, y\}$.

$P = \{y_1, y_2, \ldots, y_{N-1}\} \rightarrow P^* = \{y_1, y_2, \ldots, y_{N-1}, y\}$

* Influence evaluates the population

* Sensitivity evaluates the attribute

- We define the **sensitivity curve** of an attribute as

$$SC(y\ ;\ a(\mathcal{P})) = \frac{a(y_1,\ldots,y_{N-1},y)-a(y_1,\ldots,y_{N-1})}{\frac{1}{N}} = N\left[a(P^*)-a(P)\right]$$

$$= N\left[\ a(y_1,\ldots,y_{N-1},y) - a(y_1,\ldots,y_{N-1})\ \right]$$

- We can then plot the **sensitivity curve** as a function of the new variate value $y$.
  - the sensitivity curve gives a scaled measure of the effect that a single variate value $y$ has on the value of a population attribute $a(\mathcal{P})$.
- We can explore the sensitivity curve for any attribute. These can be determined **mathematically** in general, but can also be determined **computationally** for any particular population and any particular attribute.
- The following is a general-purpose sensity curve function in R which accommodates any population and any attribute:

inputs

```
sc = function(y.pop, y, attr, ...) {
  N <- length(y.pop) +1
  sapply( y, function(y.new) {  N*(attr(c(y.new, y.pop),...) - attr(y.pop,...))  } )
}
```

this is the code that executes when sc( ) is called.

**Example: Arithmetic Mean**

$$a(y_1,\ldots,y_N) = \frac{1}{N}\sum_{i=1}^{N} y_i = \overline{y}$$

- Derive the sensitivity curve.

$$P=\{y_1,\ldots,y_{N-1}\} \rightarrow a(P) = \frac{1}{N-1}\sum_{i=1}^{N-1} y_i = \overline{y}_{N-1}$$

$$P^*=\{y_1,\ldots,y_{N-1},y\} \rightarrow a(P^*) = \frac{1}{N}\left(\sum_{i=1}^{N-1} y_i + y\right)$$

$$= \frac{1}{N}\left((N-1)\overline{y}_{N-1} + y\right)$$

$$SC(y) = N\left[a(P^*) - a(P)\right]$$

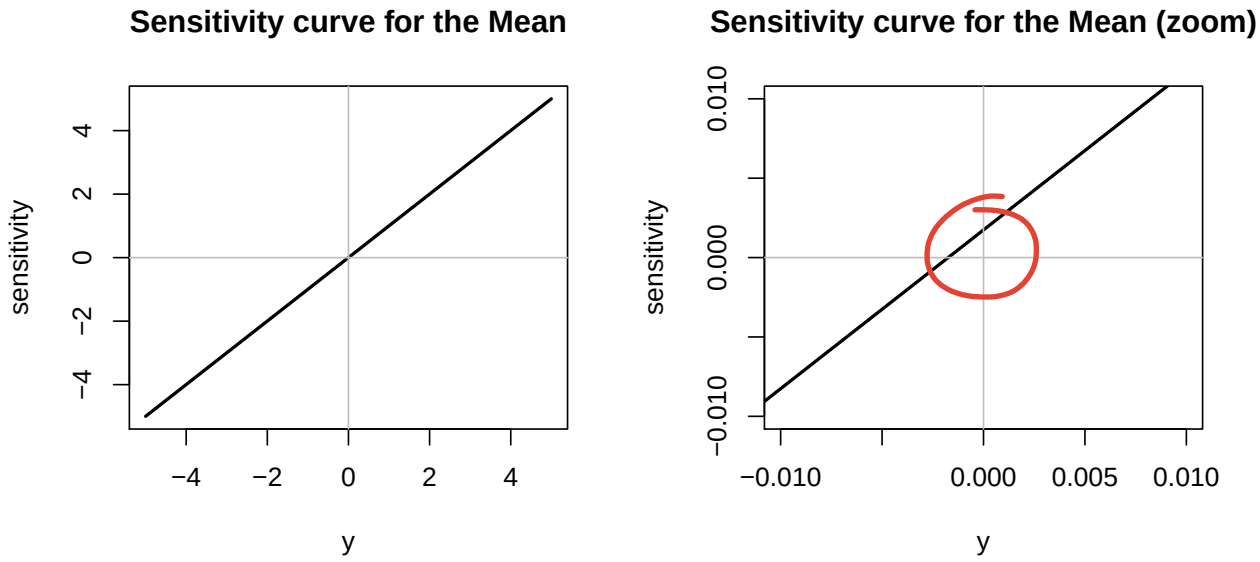
$$= \left[\frac{N-1}{N}\overline{y}_{N-1} + \frac{y}{N} - \overline{y}_{N-1}\right] \times N$$

$$= (N-1)\overline{y}_{N-1} + y - N\overline{y}_{N-1}$$

$$= y - \overline{y}_{N-1}$$

- We found the sensitivity curve to be $SC(y) = y - \overline{y}_{N-1}$
- The sensitivity curve for the arithmetic average is

```r
set.seed(341)
ys <- rnorm(1000)
y <- seq(-5,5, length.out=1000)
par(mfrow=c(1,2))
plot(y, sc(ys, y, mean), type="l", lwd = 2,
     main="Sensitivity curve for the Mean",
     ylab="sensitivity")
abline(h=0, v=0, col="grey")

plot(y, sc(ys, y, mean), type="l", lwd = 2,
     main="Sensitivity curve for the Mean (zoom)",
     ylab="sensitivity",
     xlim=c(-.01,.01),ylim=c(-.01,.01))
abline(h=0, v=0, col="grey")
```




**Note**

- The sensitivity curve here gets higher (or lower) without bound as $y \to \infty$ (or as $y \to -\infty$).
- A single observation can change the average by a huge (even infinite) amount.
- Averages may not be the best choice for a population attribute representing the location of a population – particularly if extreme values exist in the population.

**Example: Maximum**

$$a(y_1, \ldots, y_N) = \max\{y_1, \ldots, y_N\} = y_{(N)}$$

- Derive the sensitivity curve.

$P = \{y_1, \ldots, y_{N-1}\} \longrightarrow a(P) = \max\{y_1, \ldots, y_{N-1}\} = y_{(N-1)}$

$P^* = \{y_1, \ldots, y_{N-1}, y\} \longrightarrow a(P^*) = \max\{y_1, \ldots, y_{N-1}, y\}$

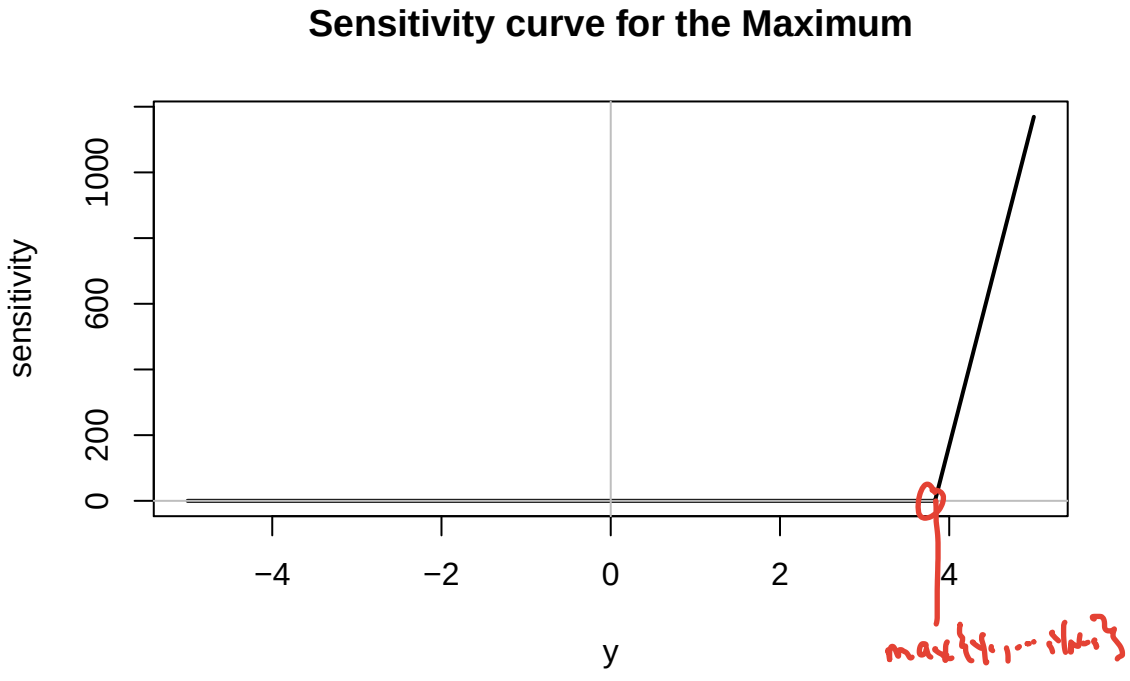
$$= \begin{cases} y_{(N-1)} & \text{if } y < \max\{y_1, \ldots, y_{N-1}\} \\ y & \text{if } y \geq \max\{y_1, \ldots, y_{N-1}\} \end{cases}$$

$$SC(y) = N[a(P^*) - a(P)]$$

$$= \begin{cases} 0 & \text{if } y < \max\{y_1, \ldots, y_{N-1}\} \\ N(y - y_{(N-1)}) & \text{if } y \geq \max\{y_1, \ldots, y_{N-1}\} \end{cases}$$

- The sensitivity curve for the maximum is

```
plot(y, sc(ys, y, max), type="l", lwd = 2,
     main="Sensitivity curve for the Maximum",
     ylab="sensitivity")
abline(h=0, v=0, col="grey")
```



How do you interpret this curve?

The fact that the curve is unbounded for large y means that the maximum is very sensitive to large values.

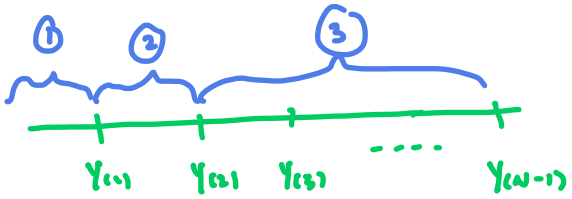
**Example:** $2^{nd}$ **Order Statistic**

$$a(y_1, \ldots, y_N) = y_{(2)}$$

- Derive the sensitivity curve.

$P = \{y_1, \ldots, y_{N-1}\} \longrightarrow a(P) = y_{(2)}$

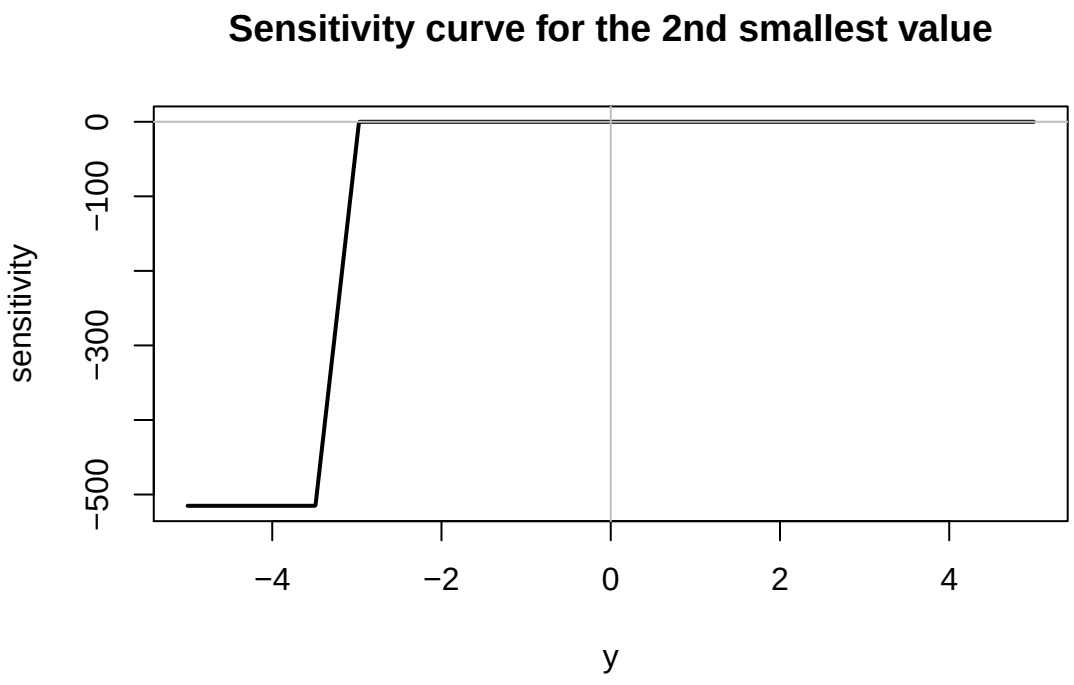
$P^* = \{y_1, \ldots, y_{N-1}, y\} \longrightarrow a(P^*) = \begin{cases} y_{(1)} & \text{if } y < y_{(1)} \quad (1) \\ y & y_{(1)} \le y < y_{(2)} \quad (2) \\ y_{(2)} & y \ge y_{(2)} \quad (3) \end{cases}$




$$SC(y) = \begin{cases} N \times (y_{(1)} - y_{(2)}) & \text{if } y < y_{(1)} \\ N \times (y - y_{(2)}) & \text{if } y_{(1)} \le y < y_{(2)} \\ 0 & \text{if } y \ge y_{(2)} \end{cases}$$

- $2^{nd}$ Order Statistic - Sensitivity Curve

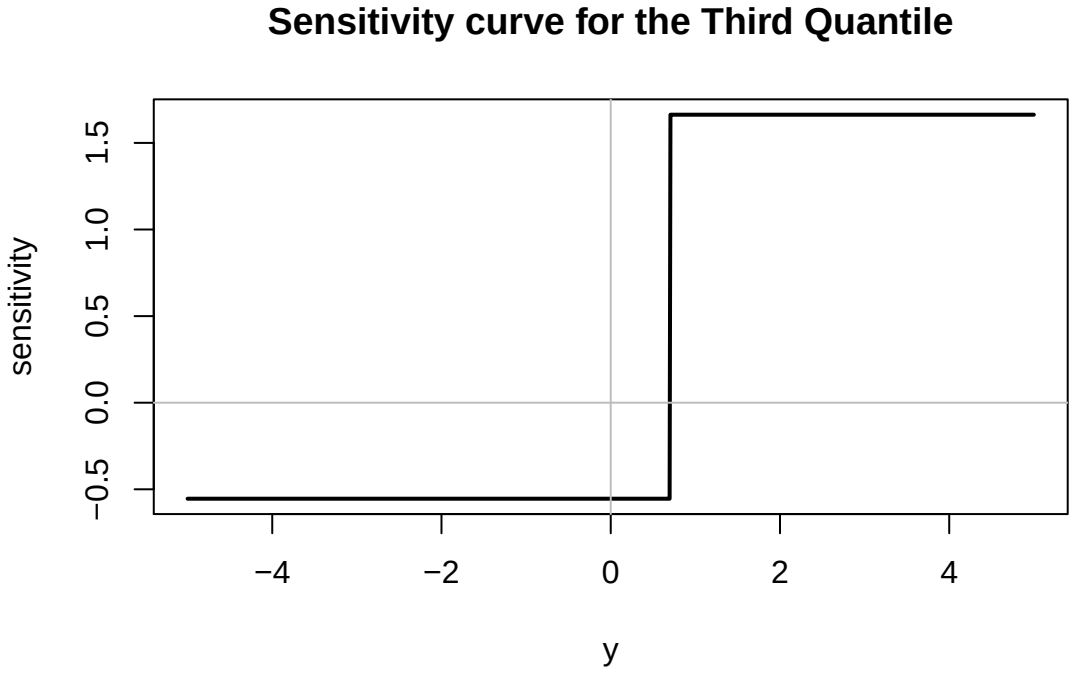
```
order.stat <- function(pop, k=1) {  sort(pop)[k]  }

plot(y, sc(ys, y, order.stat, k=2), type="l", lwd = 2,
     main="Sensitivity curve for the 2nd smallest value",
     ylab="sensitivity")
abline(h=0, v=0, col="grey")
```




**Example:** $3^{rd}$ **Quantile**

- Third Quantile - Sensitivity Curve

```
plot(y, sc(ys, y, quantile, p=0.75), type="l", lwd = 2,
     main="Sensitivity curve for the Third Quantile",
     ylab="sensitivity")
abline(h=0, v=0, col="grey")
```



**Example: Median**

Suppose the size of the original population (excluding the $y$ value) is an even number $N-1=2m$. Then the median is

$$a_{N-1}(y_1,\ldots,y_{N-1}) = \frac{1}{2}\left(y_{(m)} + y_{(m+1)}\right).$$

Now, adding the new value $y$, the median $a_N(y_1,\ldots,y_N)$ is calculated based on $N=2m+1$ values. The ordered values are

$$y_{(1)} \le y_{(2)} \le \cdots \le y_{(N)}$$

and the sensitivity curve for the median is (show this)

$$SC(y) = \begin{cases} -\frac{N}{2}\left(y_{(m+1)} - y_{(m)}\right) & \text{if} \quad y < y_{(m)} \\ \frac{N}{2}\left(2y - y_{(m+1)} - y_{(m)}\right) & \text{if} \quad y_{(m)} \le y \le y_{(m+1)} \\ \frac{N}{2}\left(y_{(m+1)} - y_{(m)}\right) & \text{if} \quad y > y_{(m+1)} \end{cases}$$

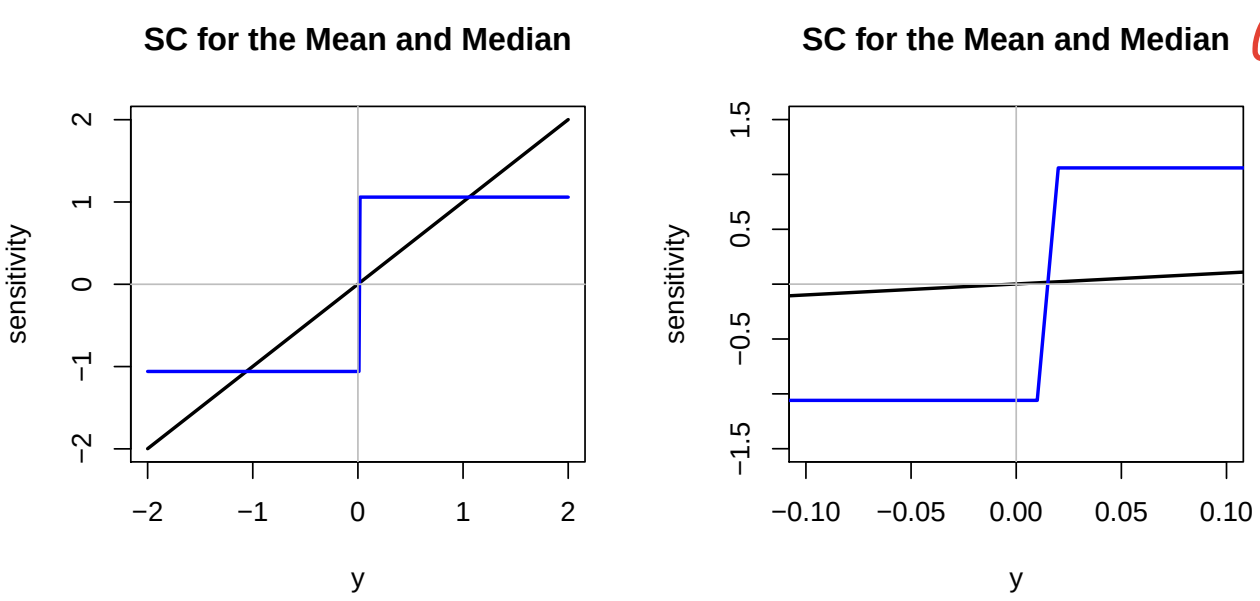
which looks like a

- negative constant when $y < y_{(m)}$,
- a positive constant at $y_{(m+1)} < y$,
- and a simple straight line with positive slope when $y$ is between $y_{(m)}$ and $y_{(m+1)}$.

* Exercise: Consider $N-1$ is odd

As with the arithmetic average, we can draw the sensitivity curve now for the median for any particular sample.

**Example: Median & Mean SC Plot**




- Unlike the arithmetic mean, the sensitivity curve for the median is at least bounded.

- A single observation cannot change the median by very much ("robust" to outliers).
- This makes the median an interesting population location attribute.

the point at which we "break" the attribute (like sensitivity curves when $y = \pm\infty$

## Breakdown Points

- Another measure of robustness that exists is called the **breakdown point**.
  - It gives an assessment of just how large a proportion of the data must be contaminated before the statistic breaks down.

Suppose we have the following population

```
x = c(22,5,3,2,21)
x
```

```
## [1] 22  5  3  2 21
```

We can calculate the mean and median:

```
c(mean(x), median(x))
```

```
## [1] 10.6  5.0
```

Then if we change the first value in the population to infinity the mean and median become:

```
y = x
y[1] = Inf
c(mean(y), median(y))
```

```
## [1] Inf   5
```

The difference between the two mean and median calculations (called **error**) is:

```
c(mean(y), median(y)) - c(mean(x), median(x))
```

```
## [1] Inf   0
```

- The breakdown point of a statistic is the *smallest possible fraction* of the observations that can be changed to something very extreme (plus or minus infinity) to make the error large (infinite)

- e.g. the break-point for
  - the average is $1/N$ (or asymptotically zero), and
  - the median is $1/2$ (i.e., that is half of the data has to go to infinity before the median breaks down).
- Attributes with high breakdown points are called **resistant** or **robust**.

## 2.2.4 Graphical Attributes

Population attributes can also be entirely graphical as in

- histograms of $y_u$ values
- bar plots of $y_u$ values
- box plots of $y_u$ values
- scatter-plots of pairs $(x_u, y_u)$
- scatter-plots of quantiles and ranks of $y_u$.

univariate – summarize the shape of a distribution

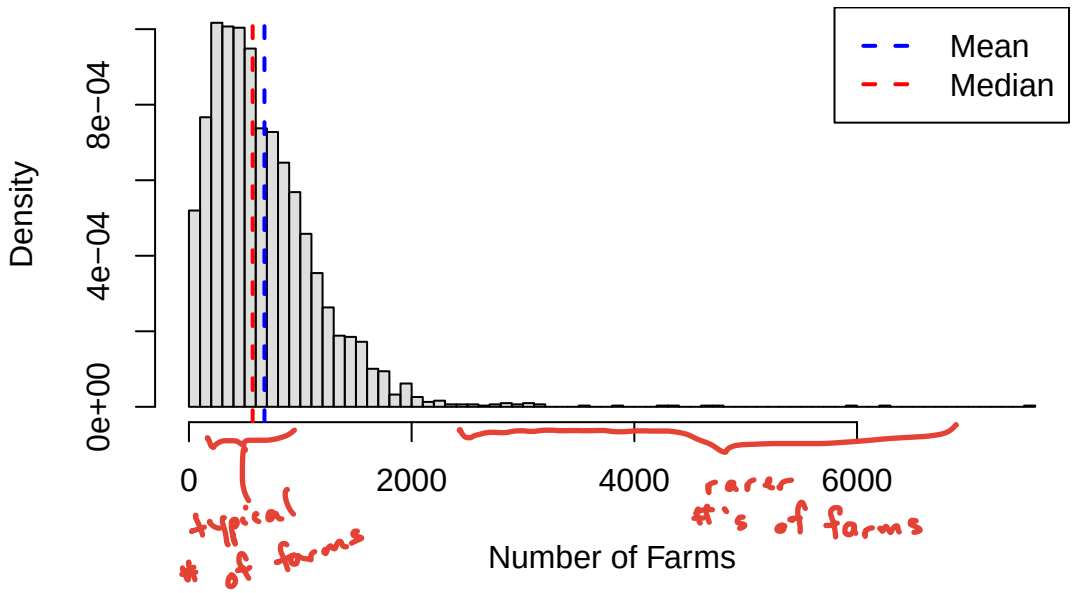
bivariate – summarize relationships

Each of these plots summarizes the entire population, and so it is an attribute.

### Histograms

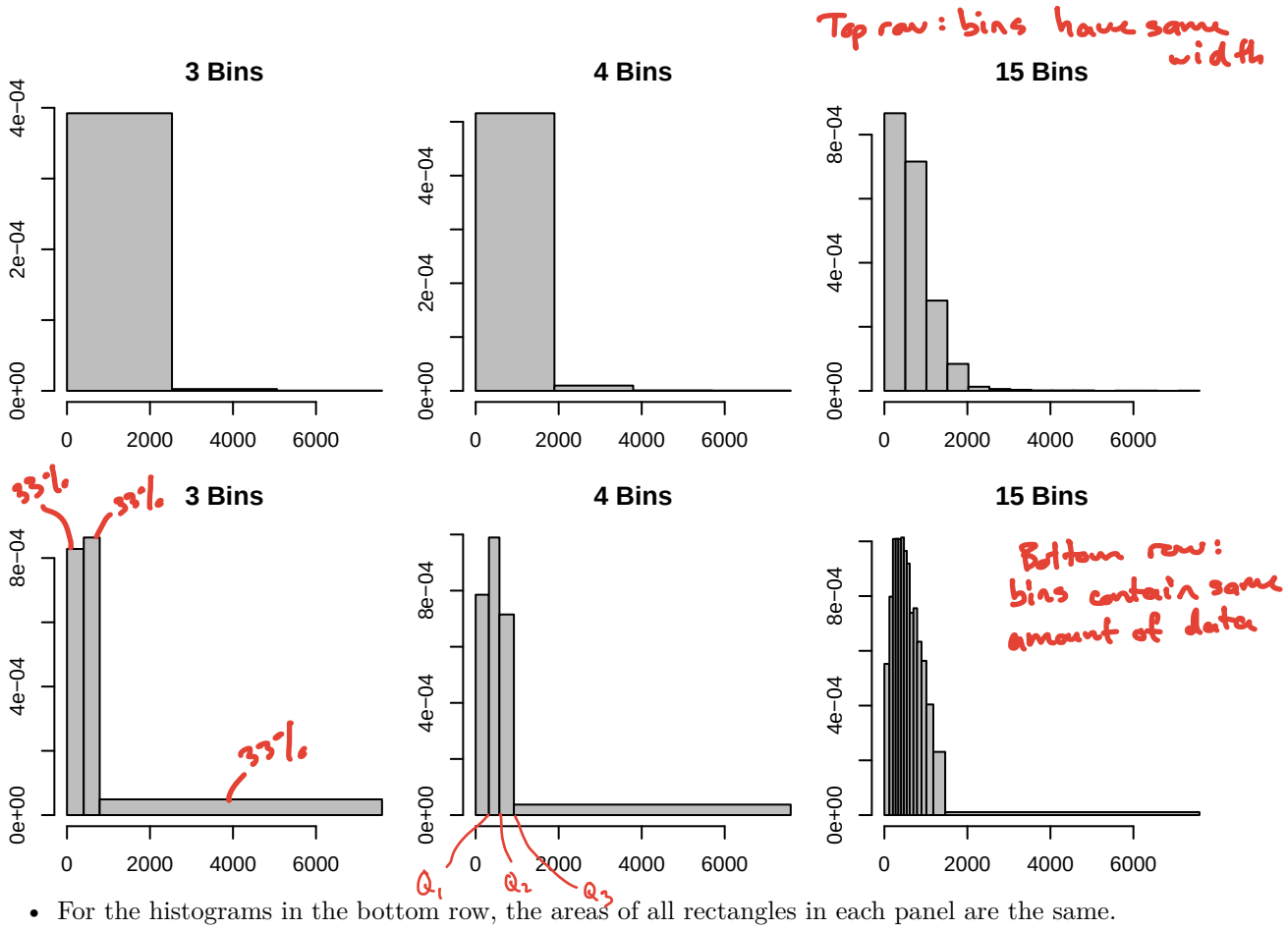
- Consider the population $\mathcal{P} = \{y_1, y_2, \ldots, y_N\}$.
  - Partition the range of the population into $k$ non-overlapping intervals, called **bins**, $I_j = [a_{j-1}, a_j)$, $j = 1, 2, \ldots, k$ and then calculate the number or proportion of observations in the $j^{\text{th}}$ bin for $j = 1, \ldots, k$.
  - Histograms help determine how the values are concentrated.

```
hist(agpop$farms87, col=adjustcolor("grey", alpha = 0.5),
     main="Number of Farms per County in 1987",
     xlab="Number of Farms",
     breaks=100, prob=TRUE )

abline(v=c(mean(agpop$farms87), median(agpop$farms87) ),
       col=c("blue","red"),lwd=2,lty=2)

legend("topright", c("Mean", "Median"), lwd=2, lty=2, col=c("blue","red"))
```




- We can define bins two ways:
  1. bins of equal size, or ← this is common
  2. bins with equal number of elements but varying size. ← less common, but can be informative
- Below are some examples of histograms with equal-sized bins (top row) and bins of varying sizes (bottom row)

```
x = agpop$farms87
par(mfrow=c(2,3), mar=2.5*c(1,1,1,0.1))
rx = range(x)
hist(x, breaks=seq(rx[1], rx[2], length.out=4), prob=TRUE, main="3 Bins", col = "grey")
hist(x, breaks=seq(rx[1], rx[2], length.out=5), prob=TRUE, main="4 Bins", col = "grey")
hist(x, breaks=seq(rx[1], rx[2], length.out=16), prob=TRUE, main="15 Bins", col = "grey")

hist(x, breaks=quantile(x, p=seq(0, 1, length.out=4)), prob=TRUE, main="3 Bins", col = "grey")
hist(x, breaks=quantile(x, p=seq(0, 1, length.out=5)), prob=TRUE, main="4 Bins" , col = "grey")
hist(x, breaks=quantile(x, p=seq(0, 1, length.out=16)), prob=TRUE, main="15 Bins", col = "grey")
```




- For the histograms in the bottom row, the areas of all rectangles in each panel are the same.
- The bins with equal numbers of elements but varying size can help identify asymmetry in the population.
- Notice also how varying the bin size changes the coarseness of the histogram.

## Rules for the Number of Bins (when bins have equal size)

- Sturges rule:

$$\text{the number of bins should be } = \lceil \log_2(N) + 1 \rceil$$

- Freedman–Diaconis rule:

$$\text{Bin size} = 2\frac{\text{IQR}(x)}{N^{1/3}}$$

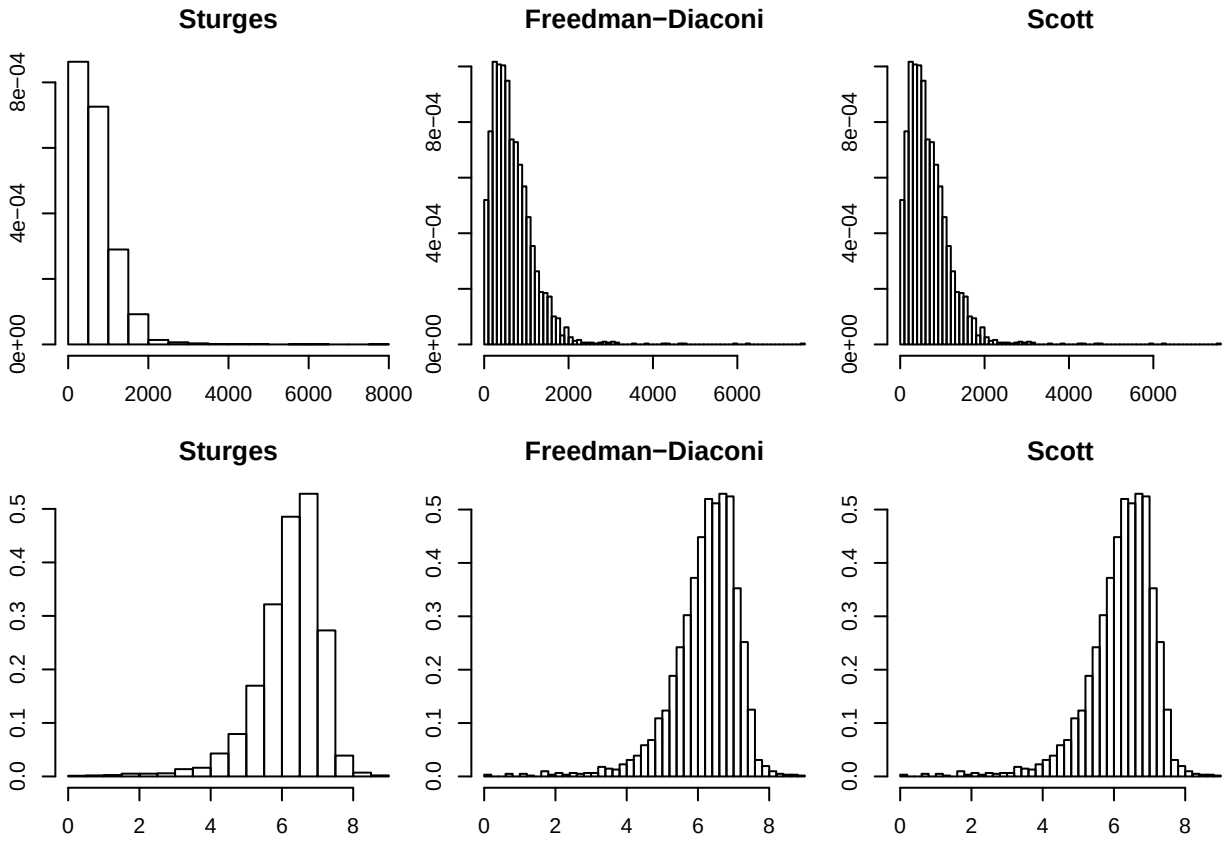
- Scott's rule:

$$\text{Bin size} = 3.5\frac{\sigma}{N^{1/3}}$$

default in R

- Histograms using different rules for bin size selection:
  - the first row is `Number of farms` and
  - the second row is `log(Number of farms+1)`. ← transformed / re-expressed version of the data

```r
par(mfrow=c(2,3), mar=2.5*c(1,1,1,0.1))
x = agpop$farms87
hist(x, prob=TRUE, xlab="", main="Sturges")
hist(x, breaks="FD", prob=TRUE, xlab="", main="Freedman-Diaconi")
hist(x, breaks="scott", prob=TRUE, xlab="", main="Scott")

x = log(agpop$farms87 +1)
hist(x, prob=TRUE, xlab="", main="Sturges")
hist(x, breaks="FD", prob=TRUE, xlab="", main="Freedman-Diaconi")
hist(x, breaks="scott", prob=TRUE, xlab="", main="Scott")
```




- **Aside:** Which scale would you prefer to work with? The original scale or the transformed scale?

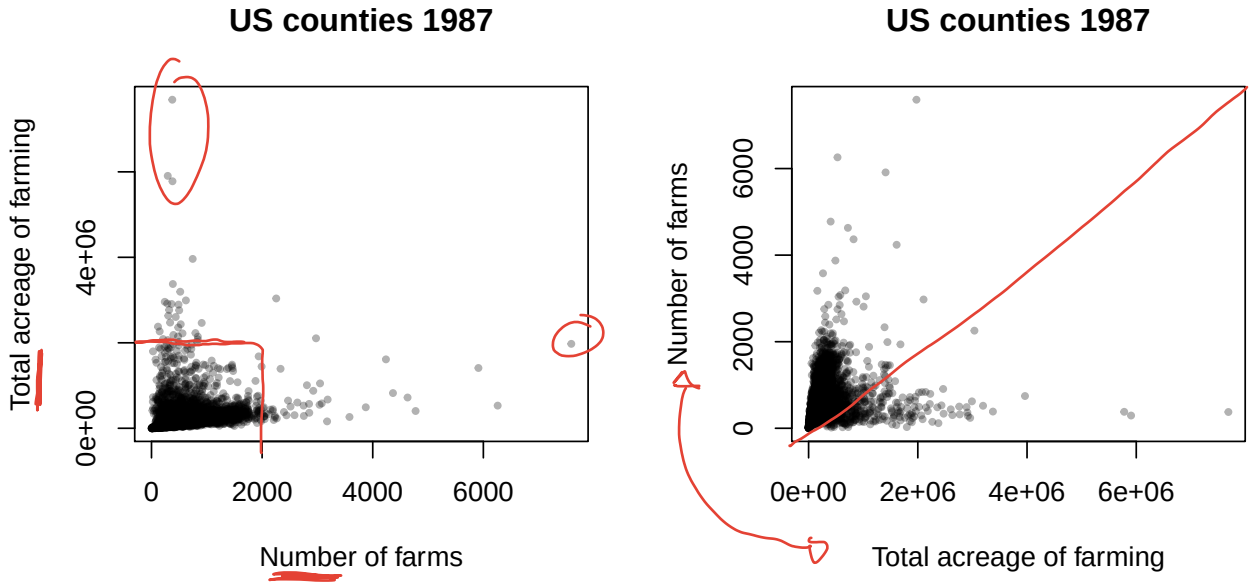
Raw Data Advantage: histogram is more interpretable

Transformed Data Advantage: histograms are more symmetric

## Scatter-plots

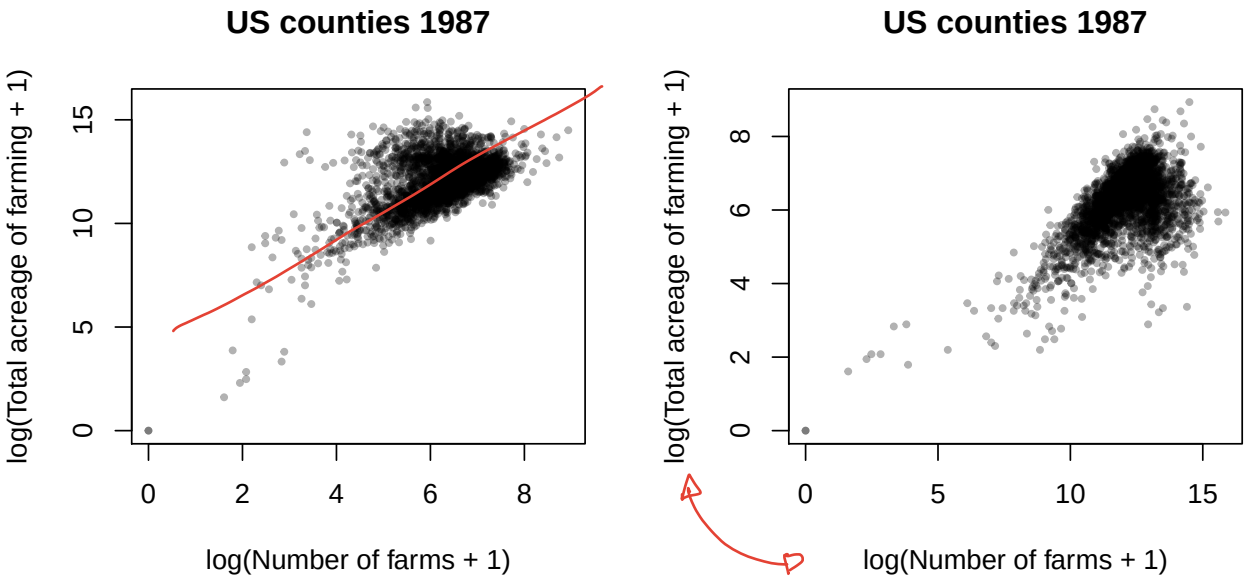
- A scatter-plot is a plot of the points $(x_u, y_u)$ for all units in the population.
  - It is used to see whether two variates $x$ and $y$ are related in some way
- A scatter-plot of the number of farms and total acreage of farming in 1987 by US county is below.

```
par(mfrow=c(1,2))
plot(agpop$farms87, agpop$acres87, pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3),
     xlab = "Number of farms", ylab = "Total acreage of farming",
     main = "US counties 1987")

plot(agpop$acres87, agpop$farms87, pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3),
     ylab = "Number of farms", xlab = "Total acreage of farming",
     main = "US counties 1987")
```




- Sometimes, the scatter-plot of a transformed version of the data provides more insight. We will discuss this later.

```
par(mfrow=c(1,2))
plot(log(agpop$farms87+1), log(agpop$acres87+1), pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3),
     xlab = "log(Number of farms + 1)", ylab = "log(Total acreage of farming + 1)",
     main = "US counties 1987")

plot(log(agpop$acres87+1),log(agpop$farms87+1), pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3),
     xlab = "log(Number of farms + 1)", ylab = "log(Total acreage of farming + 1)",
     main = "US counties 1987")
```




**Fire Emblem Heroes**


Fire Emblem Heroes is a free-to-play tactical role-playing game developed by Intelligent Systems and Nintendo for iOS and Android devices. The Population is 168 characters with varying type and movement.

- Some variates are below but the game mechanics allow for the development of new variates.

```
feh <- read.csv("../Data/feh.csv", header = TRUE)
head(feh)
```

```
##      Name       Type     Move HP ATK SPD DEF RES Total
## 1    Abel Blue Lance  Cavalry 39  33  32  25  25   154
## 2 Alfonse  Red Sword Infantry 43  35  25  32  22   157
## 3     Alm  Red Sword Infantry 45  33  30  28  22   158
## 4  Amelia  Green Axe  Armored 47  34  34  35  23   173
## 5    Anna  Green Axe Infantry 41  29  38  22  28   158
## 6  Arthur  Green Axe Infantry 43  32  29  30  24   158
```

- Here we will consider the following two variates in particular:
  - `RES`: Resistance, the ability to absorb magical attacks
  - `DEF`: Defense, the ability to absorb physical attacks

```
plot(feh$RES, feh$DEF, main="Fire Emblem Heroes", pch = 19, cex=1,
     col=adjustcolor("black", alpha = 1), xlab="Resistance", ylab="Defense" )
```

changes aspects of the plotting region

this is better when data points are exactly duplicated

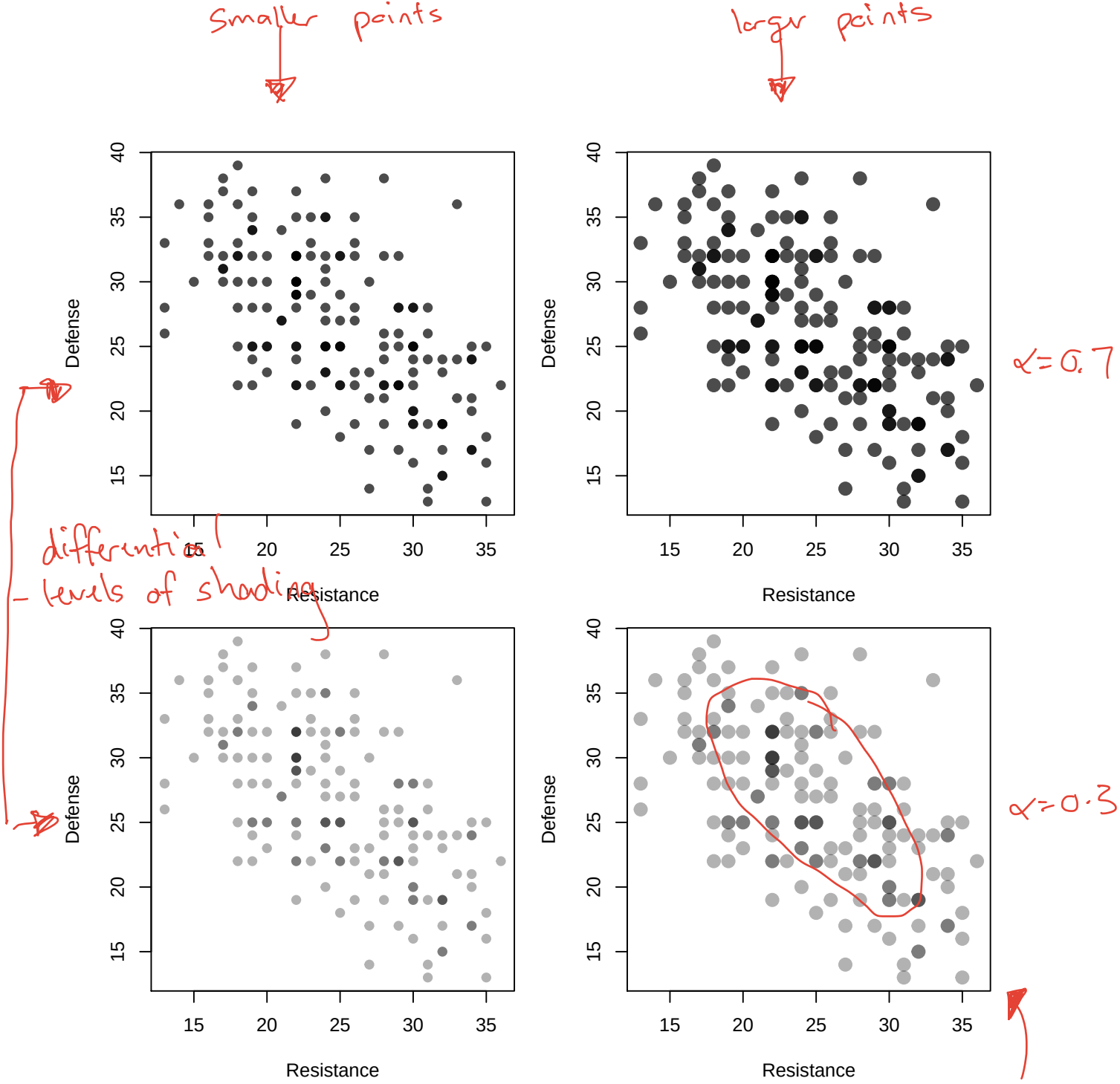
- Problem: The population values are integer-valued making duplicate values difficult to identify.
- Solution: Try changing the shading and varying the size of bullets.

```
par(mfrow=c(2,2), mar=c(4,4,1,1))

cex.seq   = rep( c(1,1.5), times=2)
shade.seq = rep( c(0.7,0.3), each=2)

for (i in 1:4) {
 plot(feh$RES, feh$DEF, main="", pch = 19,
   cex=cex.seq[i],
      col=adjustcolor("black", alpha = shade.seq[i]),
      xlab="Resistance", ylab="Defense" )
}
```

this changes size

cex = 1 is default

this is better when data is tightly clustered

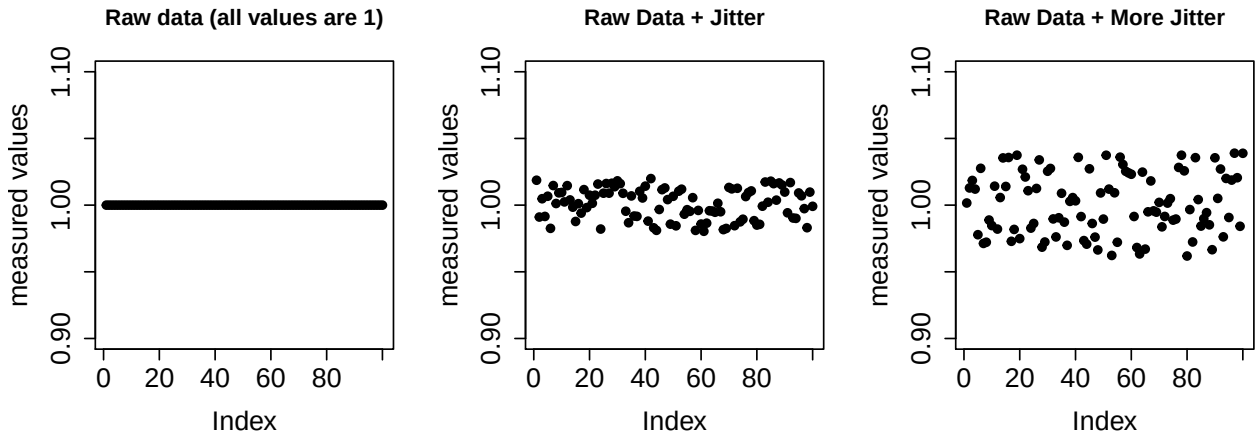



- Note the use of `par` to display four figures in one plot
  - `mfrow=c(2,2)` makes 2x2 panel of figures.
  - `mar=c(4,4,1,1)` changes the spacing so that the plots are more compact.
  - Try changing or removing the `mar` and/or `mfrow` calls to see what kind of control you have.
- Another way to deal with discreteness (multiple points at the same co-ordinate) is to add **jitter** to the population values.
  - Jitter seperates duplicate points slightly (provided it makes sense to do so).

$$y_u^\star = y_u + noise$$

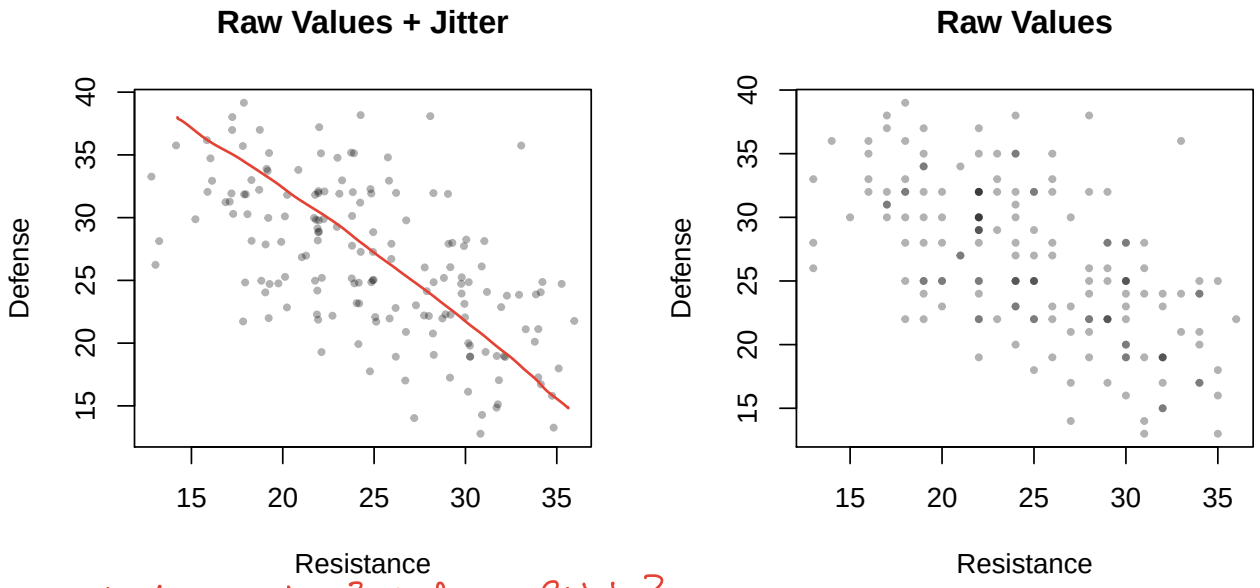
- The amount of jitter can vary:

```
y.example <- rep(1,100)
par(mfrow=c(1,3)) #dividing the panel into 1 row and 3 columns for 3 plots

title.seq = c('Raw data (all values are 1)', 'Raw Data + Jitter', 'Raw Data + More Jitter')
fact.seq  = c(0, 1, 2)

for (i in 1:3) {
 plot(jitter(y.example, factor= fact.seq[i]),
      main=title.seq[i],
      ylim=c(0.9,1.1), ylab='measured values',
      cex.lab=1.5, cex.axis=1.5, pch=19)
}
```




- Check the help documentation of the function `jitter` by typing `?jitter` or `help(jitter)` in R.
- The scatter-plot of Fire Emblem Heroes data with jitter is:

```
par(mfrow=c(1,2))

plot( jitter(feh$RES,factor=1.5), jitter(feh$DEF,factor=1.5),
      main="Raw Values + Jitter",  pch = 19, cex=0.5,
      col=adjustcolor("black", alpha = 0.3),
      xlab="Resistance",  ylab="Defense", type="p" )

plot(feh$RES, feh$DEF,
     main= "Raw Values",
     pch = 19,  cex=0.5,
     col=adjustcolor("black", alpha = 0.3 ),
     xlab="Resistance", ylab="Defense" )
```




Which is better? Left or Right?

- Advantage of Left: more clearly illustrates how much data we have
- Advantage of Right: more clearly indicates the observed values in the dataset

## 2.2.5 Power Transformations

For any variate $y$, it is sometimes helpful to re-express the values in a non-linear way via a transformation $T(y)$ so that on the transformed scale location/scale attributes are easier to define, to understand, or simply to determine.

- A commonly used transformation when $y > 0$ is the family of **power transformations** which is indexed by a power $\alpha$. The general form is

$$T_\alpha(y) = \begin{cases} y^\alpha & \alpha \neq 0 \\ \log(y) & \alpha = 0 \end{cases}$$

- These transformations are monotonic, in the sense that

$$y_u < y_v \iff T_\alpha(y_u) < T_\alpha(y_v)$$

  That is, they preserve the order of the variate values associated with the units $u$ and $v$.
  - What does change, often dramatically, is the relative positions of the variate values.
- The effect of the power transformation on number of farms in 1987
- What is the effect of varying the power transformation?
  - Different values of $\alpha$ change the spacing between observations. Changing this spacing serves to symmetrize the data
- **Note:** the most common purpose of a transformation is to change the shape of the histogram so that it is more symmetric.
  - We mentioned that if $y > 0$, the family of **power transformations** indexed by a power $\alpha$ is defined as

$$T_\alpha(y) = \begin{cases} y^\alpha & \alpha \neq 0 \\ \log(y) & \alpha = 0 \end{cases}$$

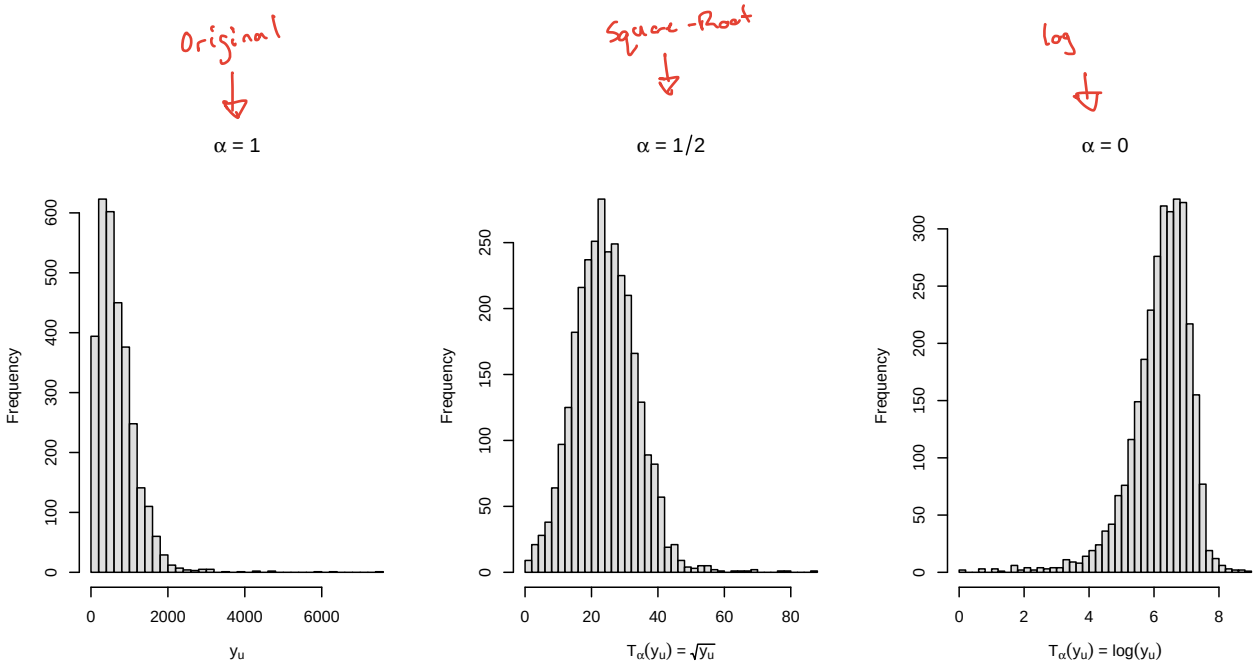



Figure 1: Effect of power transformation on number of farms in 1987

– a more convenient mathmatical form is

$$T_\alpha(y) = \frac{y^\alpha - 1}{\alpha} \quad \forall\ \alpha$$

Box-Cox Transformation

Note that the following limit gives rise to the $\alpha = 0$ case above:

$$\lim_{\alpha \to 0} T_\alpha(y) = \log(y)$$

– Finally, an even more computationally efficient power transformation (with minimal potential for calculation errors) is the following:

$$T_\alpha(y) = \begin{cases} y^\alpha & \alpha > 0 \\ \log(y) & \alpha = 0 \\ -(y^\alpha) & \alpha < 0 \end{cases}$$

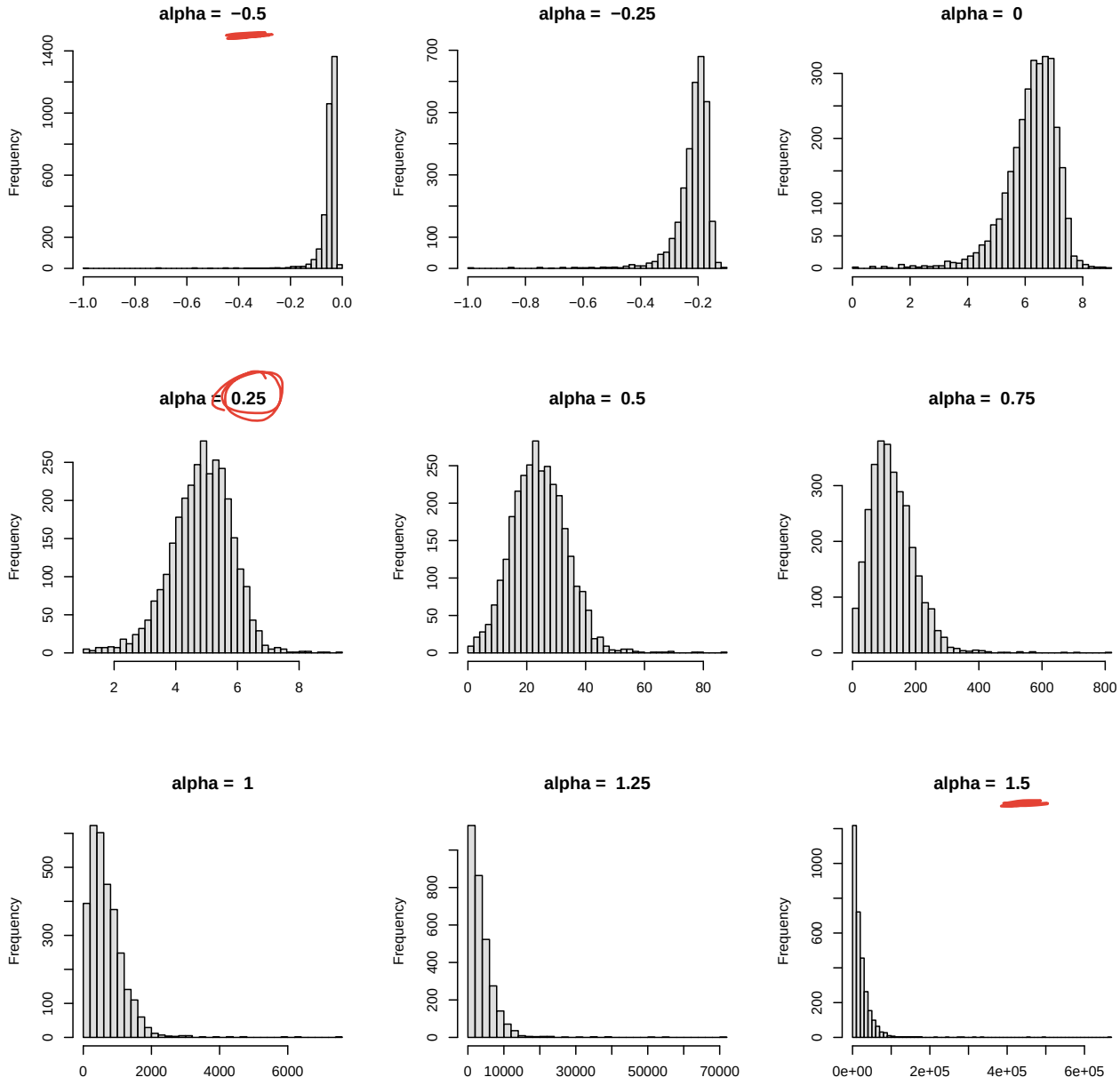
– A general-purpose R function to perform the power transformation is

the data — the power

```
powerfun <- function(x, alpha) {
  if(sum(x <= 0) > 0) stop("x must be positive")
  if (alpha == 0)
    log(x)
  else if (alpha > 0) {
    x^alpha
  } else -x^alpha
}
```

- Varying the power on the number of farms in 1987

```
par(mfrow=c(3,3))
a = seq(-1/2, 1.5, length.out=9)

for (i in 1:9) {
hist( powerfun(agpop$farms87 + 1, a[i]), col=adjustcolor("grey", alpha = 0.5),
     main= paste("alpha = ", a[i] ), xlab="", breaks=50 )
}
```


alpha = −0.5
alpha = −0.25
alpha = 0
alpha = 0.25
alpha = 0.5
alpha = 0.75
alpha = 1
alpha = 1.25
alpha = 1.5
Frequency

- Notice how the "bump" changes as $\alpha$ changes.
- Which values of $\alpha$ make the histogram or population symmetric?
  - Should we find an optimal $\alpha$? Optimize $\alpha$ by minimizing a skewness attribute
- The $\alpha$ values can take any real value in principle,
  - but in practice, we restrict the values to a small set.
  - The powers should be restricted to those which are easily interpretable.
  - John Tukey suggested (Tukey 1977) imagining that the set of powers were arranged in a "ladder" with the smallest powers on the bottom and the largest on the top.

Now one simply moves "up" or "down" on Tukey's ladder of powers to arrive at a re-expression that achieves the desired effect on the data values.

| alpha | ladder |
|---|---|
| ... | up |
| 2 | \| |
| 1 | <- original values |
| 1/2 | \| |
| 1/3 | \| |
| 0 | \| |
| -1/3 | \| |
| -1/2 | \| |
| -1 | \| |
| -2 | \| |
| ... | down |

## How to pick $\alpha$?

Two different, but related, effects of transformation are often of interest:

- First, producing a more symmetric looking histogram
- Second, producing roughly linear scatter-plots
  - Imagine (for all $u \in \mathcal{P}$) a scatter-plot of all pairs $(x_u, y_u)$.
  - Can we change the powers $\alpha_x$ and $\alpha_y$ for each such that the scatter-plot of the re-expressed pairs $(T_{\alpha_x}(x), T_{\alpha_y}(y))$ lie nearly on a straight line?
- Fortunately, for each of these effects there is a corresponding "bump rule" that indicates the direction (up or down) to move on Tukey's ladder to achieve it.

**Bump Rule 1: Making histograms more symmetric**

- The rule is that the location of the "bump" in the histogram (where the points are concentrated) tells you which way to "move" on the ladder.
  - If the bump is on "lower" values, then move the power "lower" on the ladder;
  - If it is on the "higher" values, then move the power "higher" on the ladder.
- Does this rule agree with the 1987 farming data we just plotted above?

Yes.

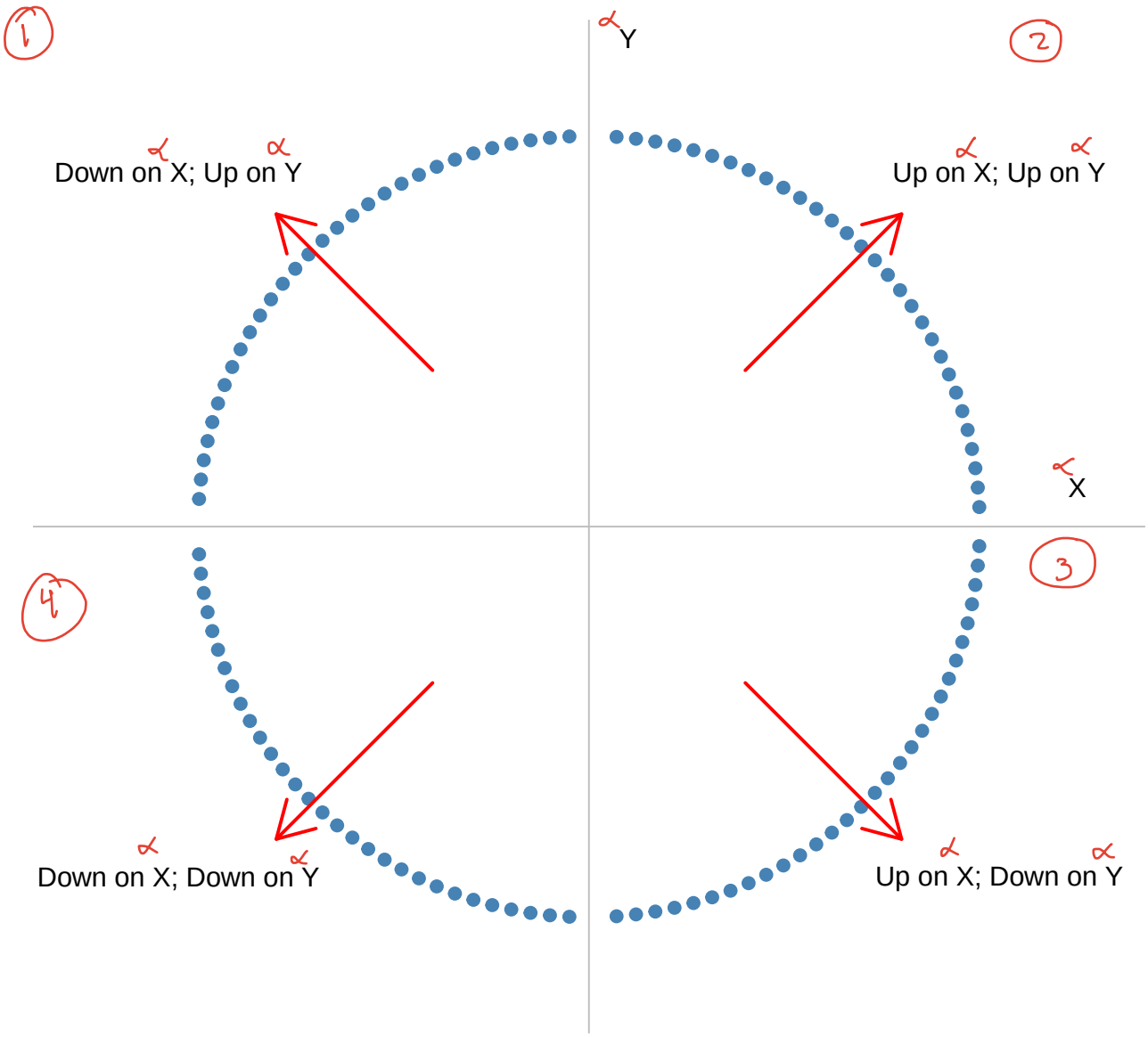
**Bump Rule 2: Straightening Scatter-plots**

i.e. make the scatter-points appear more linear.

A scatter-plot of $(x_u, y_u)$ for $u \in \mathcal{P}$ may be "straightened" by applying (possibly) different power transforms to each coordinate to give a new (hopefully straighter looking) scatter-plot of the re-expressed data $(T_{\alpha_x}(x_u), T_{\alpha_y}(y_u))$.

- Because each of the coordinates has its own power transformation, there will be two different ladders of transformation
  - the $x$ ladder and
  - the $y$ ladder.
- As with histograms, there is a "bump rule" to tell you how to move on the ladder.
  - In the case of scatter-plots, the "bump" corresponds to the curvature appearing in the scatter-plot.
  - This is only approximate in practice, but reduces to one of four different possibilities:

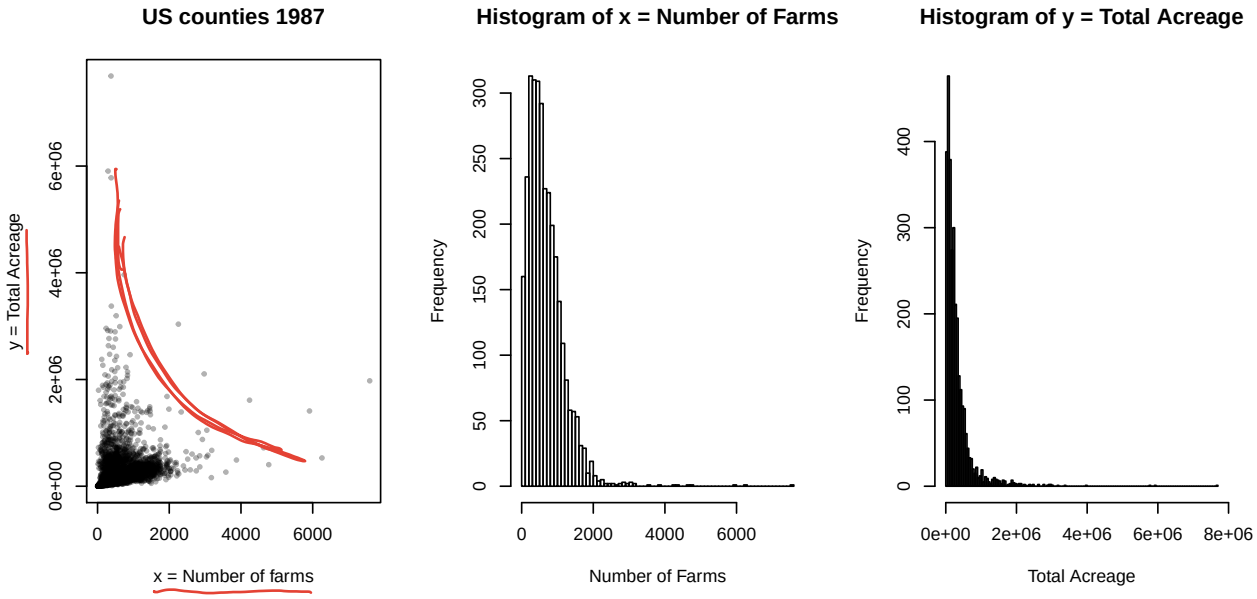
## Each quadrant shows a monotonic curved relation




Direction of the bump suggests ladder moves

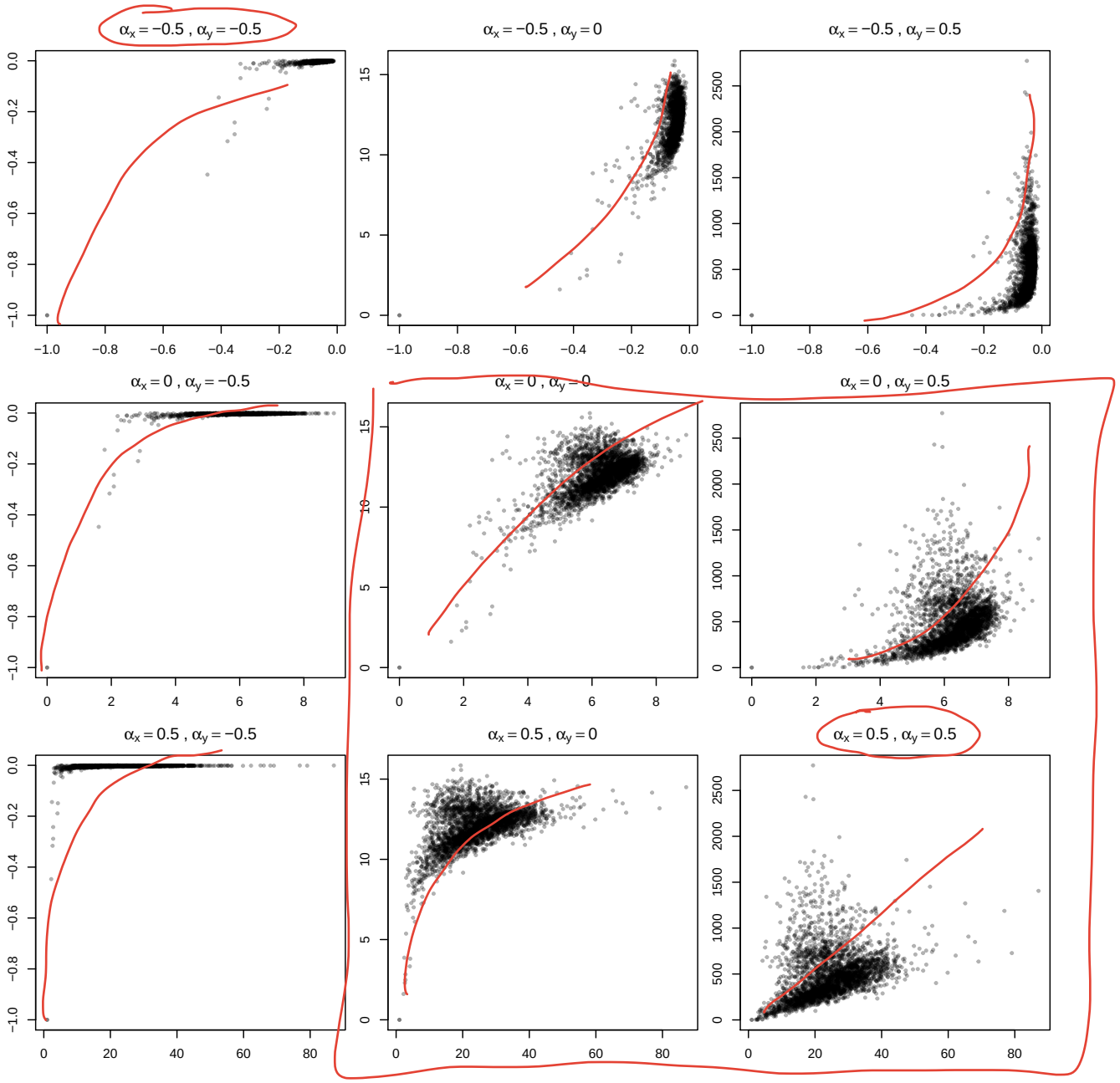
- **Example: Agriculture Data**
  - 1987: Number of Farms vs. Total Acreage of Farming

```
par(mfrow=c(1,3))

plot( agpop$farms87, agpop$acres87, pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3),
     xlab = "x = Number of farms", ylab = "y = Total Acreage",
     main = "US counties 1987")

hist( agpop$farms87, breaks = "FD",
      xlab="Number of Farms",
      main = "Histogram of x = Number of Farms")
hist( agpop$acres87, breaks = "FD",
      xlab="Total Acreage",
      main = "Histogram of y = Total Acreage")
```




- We can apply a different power transformation to each variate ($x$ = Number of Farms; $y$ = Total Acreage of Farming).

```
par(mfrow=c(3,3), mar=2.5*c(1,1,1,0.1))
a = rep(c(-1/2,0,1/2),each=3)
b = rep(c(-1/2,0,1/2),times=3)
subdata = agpop[,c('farms87', 'acres87')]
subdata  = na.omit(subdata )

for (i in 1:9) {
plot( powerfun(subdata$farms87+1, a[i]), powerfun(subdata$acres87+1, b[i]), pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3), xlab = "", ylab = "",
     main = bquote(alpha[x] == .(a[i]) ~ "," ~  alpha[y] == .(b[i])))
}
```

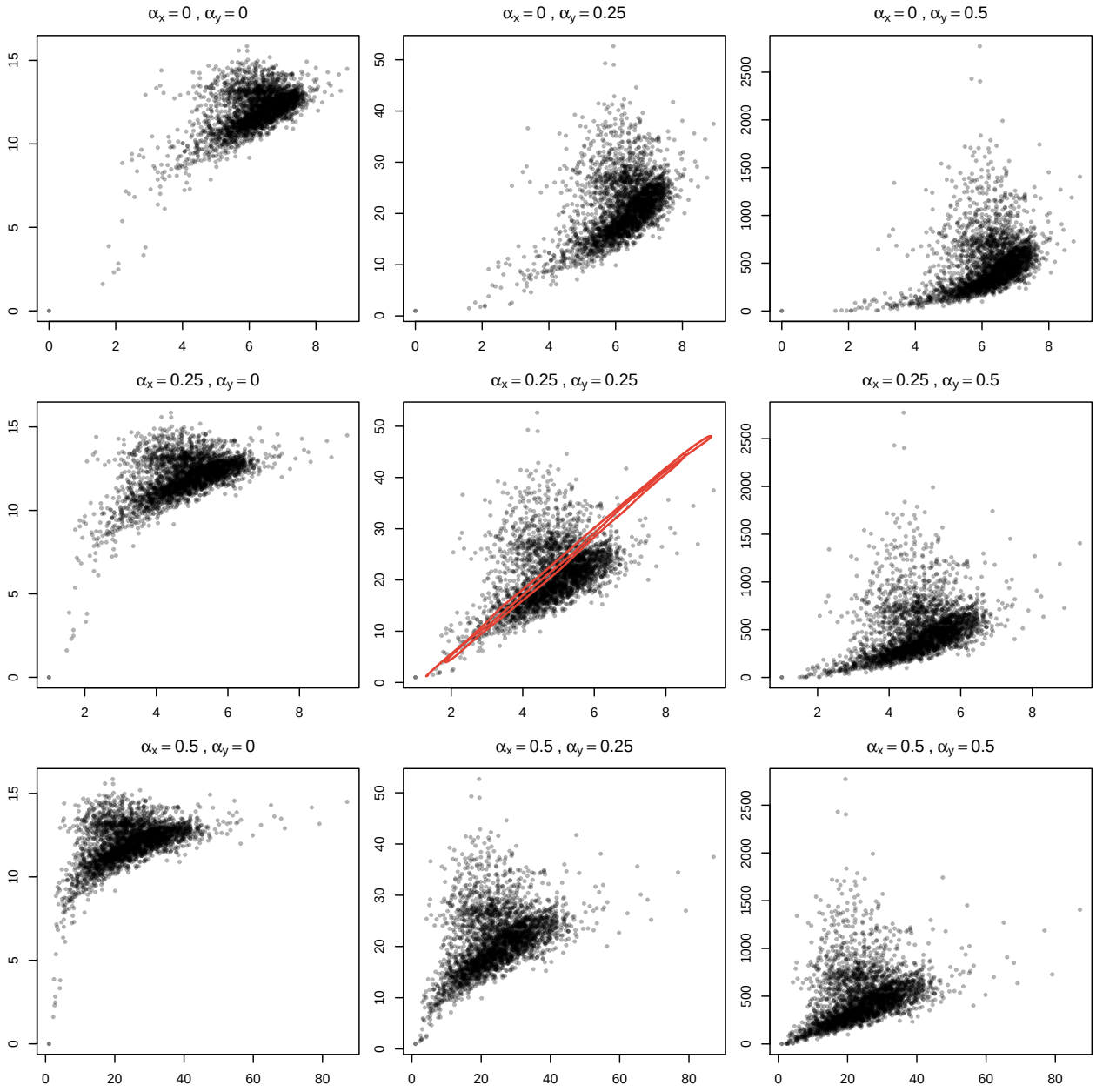



- Which $\alpha_x$ and $\alpha_y$ help straighten the relationship?

$\alpha_x$ and $\alpha_y$ in $[0, 0.5]$ range seems okay

- Refining the search grid, we have

```
par(mfrow=c(3,3), mar=2.5*c(1,1,1,0.1))
a = rep(c(0,1/4,1/2),each=3)
b = rep(c(0,1/4,1/2),times=3)
subdata = agpop[,c('farms87', 'acres87')]
subdata  = na.omit(subdata )
for (i in 1:9) {
plot( powerfun(subdata$farms87+1, a[i]), powerfun(subdata$acres87+1, b[i]), pch = 19, cex=0.5,
     col=adjustcolor("black", alpha = 0.3), xlab = "", ylab = "",
     main = bquote(alpha[x] == .(a[i]) ~ "," ~  alpha[y] == .(b[i])))
}
```




- We can perform this task (choosing the approperiate power transformation) dynamically using the 'loon' packcage in 'R'.

```
### This requires that the loon package be installed.
### install.package("loon") will install the package from CRAN
### (requires R >= 3.4)
###
library(loon)
###
power <- function(x, y,
                  linkingGroup="linkingGroup",
                  from=-5, to=5, ...) {
  ## Create histograms
  histX <- l_hist(x, linkingGroup = linkingGroup,
```



* I'm never going test you on loon or ask you to use it

```
                yshows="density")
histY <- l_hist(y, linkingGroup = linkingGroup,
                yshows="density", swapAxes = TRUE
                )
## Now we build an interactive scatter-plot
## with sliders for power transformations
## on each of x and y
tt <- tktoplevel()
tktitle(tt) <- "Power Transformation"
p <- l_plot(x=x, y=y, parent=tt,
                linkingGroup=linkingGroup,
            ...)
## Alpha values
alpha_x <- tclVar('1')
alpha_y <- tclVar('1')
## Sliders to change the alphas
sx <- tkscale(tt, orient='horizontal',
              variable=alpha_x,
              from=from, to=to, resolution=0.1)
sy <- tkscale(tt, orient='vertical',
              variable=alpha_y,
              from=to, to=from, resolution=0.1)
## Laying out the pieces in one window
tkgrid(sy, row=0, column=0, sticky="ns")
tkgrid(p, row=0, column=1, sticky="nswe")
tkgrid(sx, row=1, column=1, sticky="we")
tkgrid.columnconfigure(tt, 1, weight=1)
tkgrid.rowconfigure(tt, 0, weight=1)

## This function redraws the plots with the alphas
## from the slider values whenever it is called.
##
update <- function(...) {
  ### get transformed x and y
  transformedX <- powerfun(x, as.numeric(tclvalue(alpha_x)))
  transformedY <- powerfun(y, as.numeric(tclvalue(alpha_y)))

  ## First the scatter-plot
  l_configure(p,
              x = transformedX,
              y = transformedY)
  l_scaleto_world(p)
  ## Now the histograms
  l_configure(histX, x = transformedX)
  l_scaleto_world(histX)
  l_configure(histY, x = transformedY)
  l_scaleto_world(histY)
}
## Set the function update to be called
## whenever the slider values are changed
tkconfigure(sx, command=update)
tkconfigure(sy, command=update)
## Return the scatter-plot if assigned
```

```
  invisible(p)
}

###
### Here's putting everything together
p <- with(agpop,
          power((farms87+1), (acres87+1),
                xlabel="# farms" ,
                ylabel="acres",
                title=
                  "",
                linkingGroup = "agpop",
                itemLabel=rownames(agpop),
                showItemLabels=TRUE)
)
```

- Another example: using the `mammals` data set from the `MASS` package, the code helps you use `loon` and apply bump rules 1 and 2.

```
library(MASS)
data("mammals")
p <- with(mammals,
          power(body, brain,
                xlabel="body weight",
                ylabel="brain weight",
                title=
                  "Brain and Body Weights for 62 Species of Land Mammals",
                linkingGroup = "mammals",
                itemLabel=rownames(mammals),
                showItemLabels=TRUE)
)
```

## 2.2.6 Order and Rank Statistics

Population attributes can also be an indexed collection of values. For example, consider the following different attributes

$y_{(k)} = k^{th}$ smallest observation

- Recall the **order statistic**:

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(N)}$$

  which is the ordered values (including ties) of the variate values $y_u \in \mathcal{P}$

- The **rank statistic**:

$$r_1, r_2, \ldots, r_N$$

  which is the **ranks** of the variate values $y_1, y_2, \ldots, y_N$ from the $y_u \in \mathcal{P}$.

- For example, if $y_i = y_{(k)}$ then $y_i$ is the $k^{\text{th}}$ smallest value and so $y_i$ has rank $r_i = k$. This means that

$$y_{(r_u)} = y_u \quad \forall \ u \in \mathcal{P}$$

### Example

A small population with five units and a single variate $y$:

```
y <-  c(22,  3, 12,  1, 42)
y
```

```
## [1] 22  3 12  1 42
```

The order statistics:

```
y.ordered <- sort(y)
y.ordered
```

```
## [1]  1  3 12 22 42
```

The rank statistic (Note, no ties to worry about):

```
y.rank <- rank(y)
y.rank
```

```
## [1] 4 2 3 1 5
```

The connection between them:

```
y.ordered[y.rank] == y
```

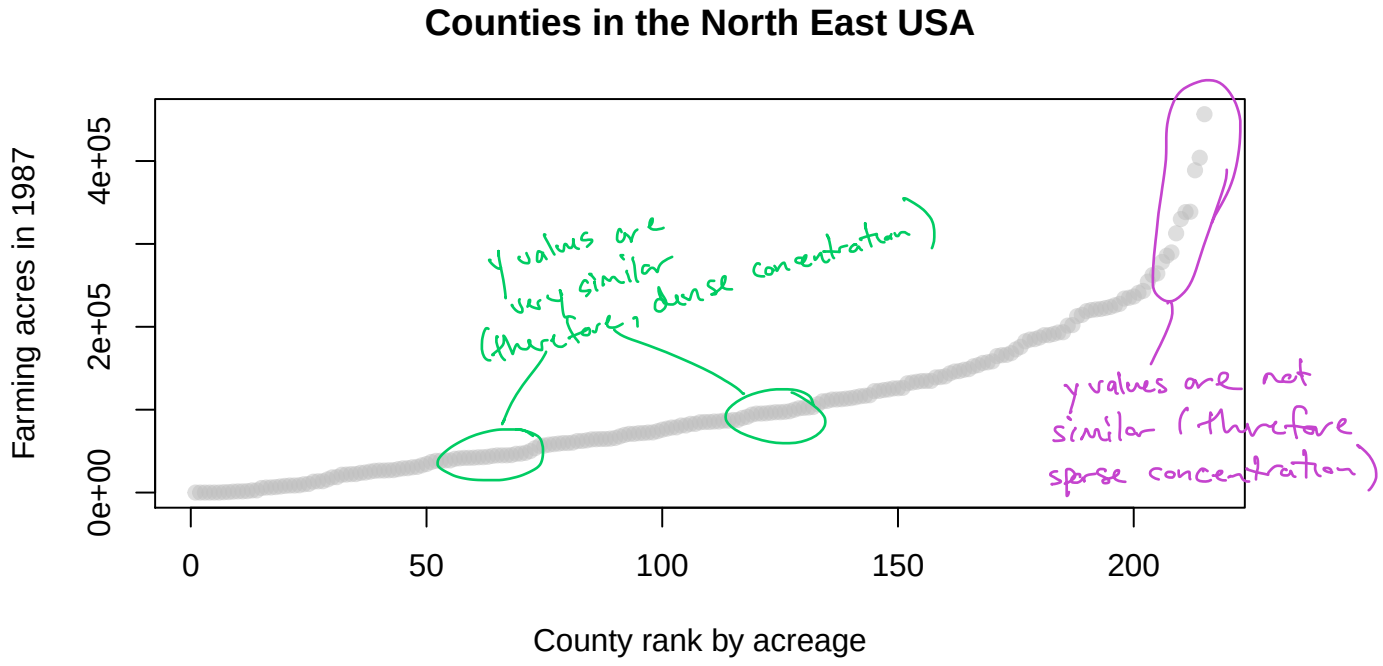
```
## [1] TRUE TRUE TRUE TRUE TRUE
```

### Scatter-plot for $y_u$ vs. rank

These two attributes are often combined as a single graphical attribute by plotting the pairs $(r_u, y_u)$ or, equivalently, $(u, y_{(u)})$ for all $u \in \mathcal{P}$.

- For example, the variate `acres87` from the agricultural census data:

```
### read the data from wherever it is stored, e.g. in some directory named Data
y <- agpop$acres87[agpop$region == "NE"]
y <- na.omit(y)
yrank <- rank(y, ties.method = "first")  # Now ensure ties appear in data set order

plot(yrank, y, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlab = "County rank by acreage",
     ylab = "Farming acres in 1987",
     main = "Counties in the North East USA")
```




- **Notes**
  - the height at any point tells the location of the value of $y$
  - horizontal location identifies where in the order of the variate values that unit appears
  - the plot is monotonically non-decreasing from left to right.
  - flat spots indicate tied values of $y$; nearly flat spots are counties where the number of acres under farming are nearly the same.
  - rapidly rising spots are counties which, though near each other in order (rank), are very different in the actual values of $y$ (acreage).

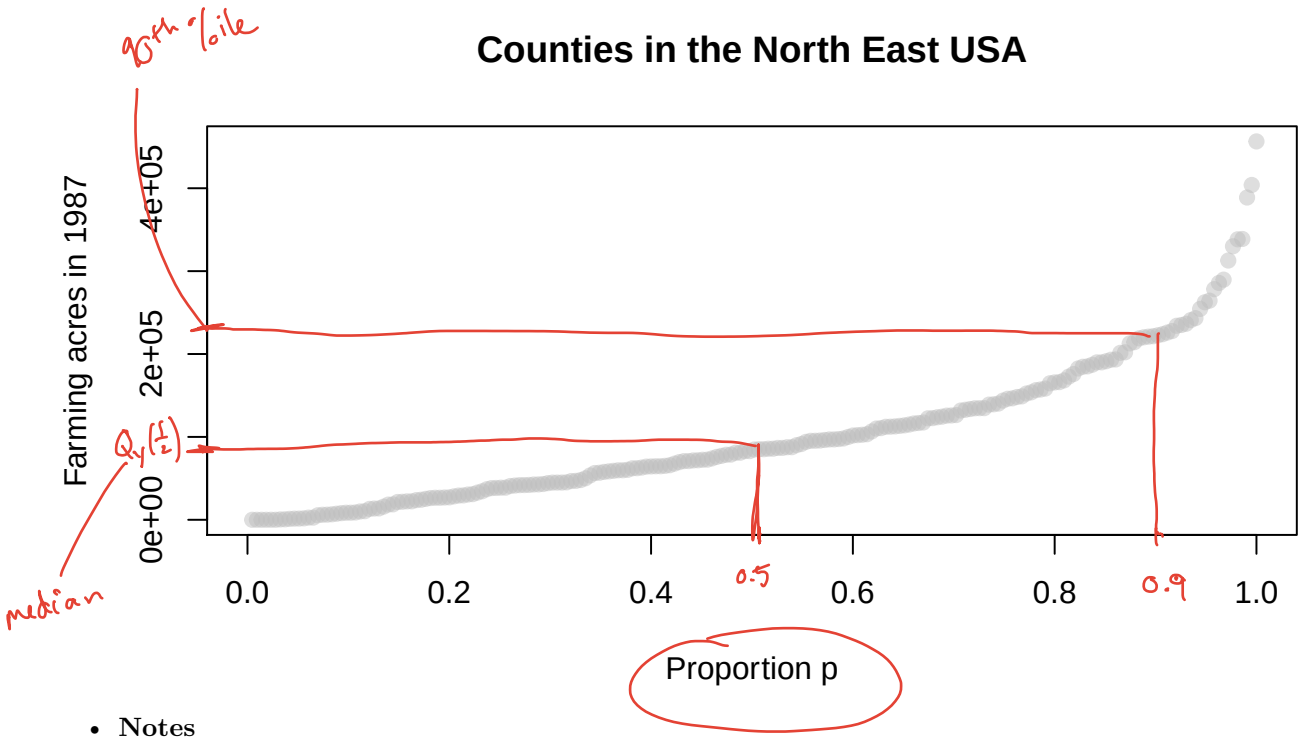
## 2.2.7 Quantiles

- Rather than using ranks, it can be more convenient to use the proportion of units in the population having a value less-than-or-equal-to $y$.
  - So instead of plotting the pairs $(r_u, y_u)$, we could equivalently plot the pairs $(p_u, y_u)$ where

$$p_u = \frac{r_u}{N}$$

    is the proportion of the units $i \in \mathcal{P}$ whose value $y_i \leq y_u$

```
N <- length(y)
p <- yrank/N
plot(p, y, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlim=c(0,1),
     xlab = "Proportion p",
     ylab = "Farming acres in 1987",
     main = "Counties in the North East USA")
```




- **Notes**
  - The middle value or proportion equal to 1/2 corresponds to the median.
  - The values on the $y$-axis are the quantiles.

- Strictly speaking, the plotted points are $(p, Q_y(p))$ where
  - $p \in \{\frac{1}{N}, \frac{2}{N}, \ldots, 1\}$ and
  - $Q_y(p)$ is the $p^{\text{th}}$ **quantile** of $y$

    $$Q_y(p) = y_{(N \times p)}$$

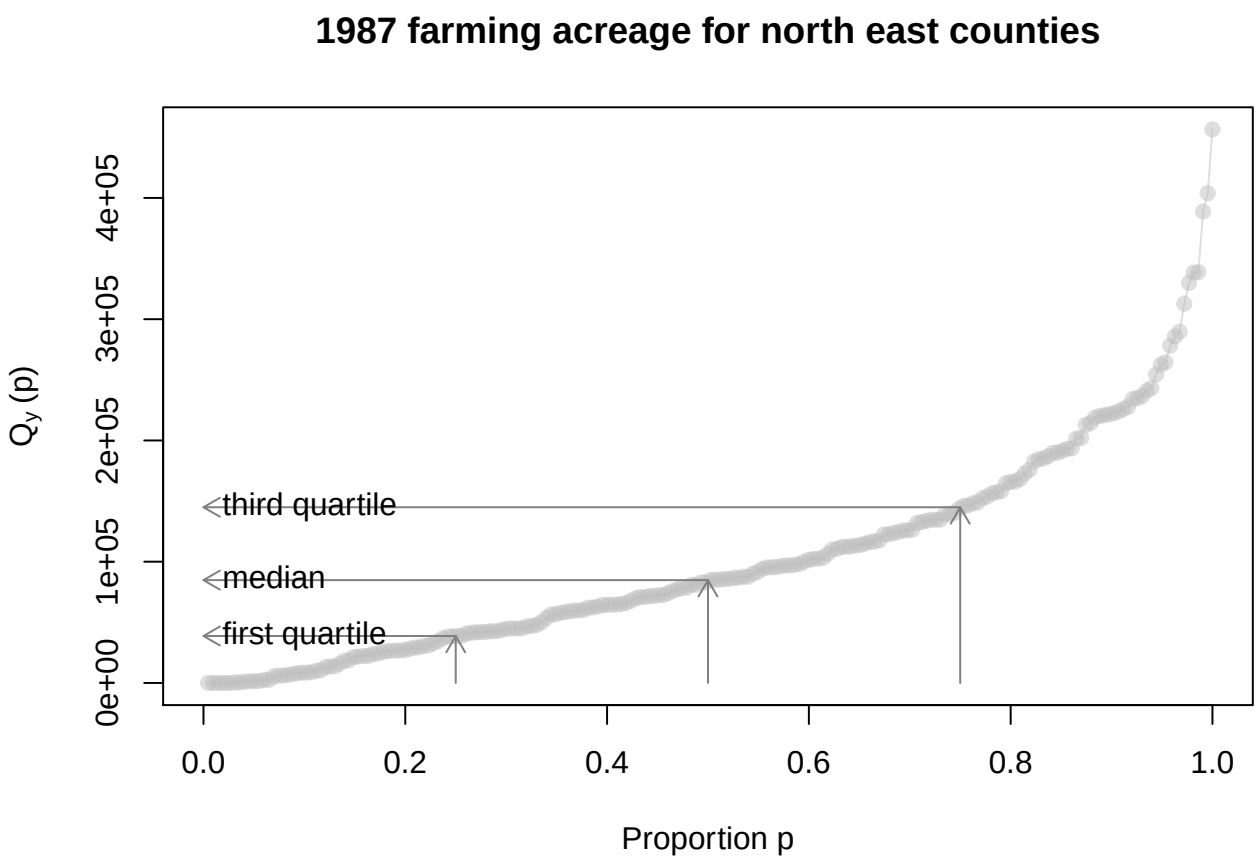
    and is sometimes called the **quantile function** of $y$ for all $p \in [\frac{1}{N}, 1]$.
- The quantile function is a population attribute which can be used to generate a number of other interesting population attributes:
  - the quantile $Q_y(p)$ for any $p$ locates the variate values in the population, and is thus a **measure of location**.
  - most (but not all) location measures try to capture **central** tendency.

## Quantiles that measure center

- the **median**: $Q_y(1/2)$
- the **mid-hinge** (average of the first and third quartiles): $\frac{Q_y(1/4)+Q_y(3/4)}{2}$
- the **mid-range** (average of the minimum and maximum): $\frac{Q_y(1/N)+Q_y(1)}{2}$
- the **trimean**: $\frac{Q_y(1/4)+2\times Q_y(1/2)+Q_y(3/4)}{4}$

These can be readily obtained from the quantile plot.

- Reading off the vertical location of $Q_y(p)$ for any pre-determined $p$ provides some measure of location.




### Quantiles that measure Spread

- The quantile function can also be used to provide some natural measures of **spread** for the variate $y$:
  - the **range**: $Q_y(1) - Q_y(1/N)$
  - the **inter-quartile range**: $IQR_y = Q_y(3/4) - Q_y(1/4)$
  - the **central 100** $\times p$**% range**
- Alternatively, the difference between any two quantiles might be divided by the difference in the corresponding $p$ values.
  - That is, the **slope** of the line segment joining any two points $(p_1, Q_y(p_1))$ and $(p_2, Q_y(p_2))$ for $p_1 < p_2$ provides a measure of spread.

## Concentration in Quantile Plots

Flatter regions in a quantile plot indicate areas where the variate values appear to be concentrated.

- To quantify this we could draw a box with **fixed height** and see how many elements are within the box.

```
# Here's an R function that draws a single
# box between the pair of points (x[1],y[1]) and (x[2],y[2])
#
drawbox <- function(x,y, ...) {
  rect(xleft = x[1], ybottom = y[1], xright = x[2], ytop = y[2], ...)
}
```

```
### Quantiles:
qvals <- sort(y)
pvals <- ppoints(length(qvals))
plot(pvals, qvals, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlim=c(0,1),
     xlab = "Proportion p",
     ylab = bquote("Q"["y"]~"(p)"),
     main = "1987 farming acreage for north east counties")


# Need some boundaries for the qvals range
qrange <- extendrange(qvals)
bins <- seq(qrange[1], qrange[2], length.out=15)
col <- adjustcolor("steelblue", 0.2)
border <- adjustcolor("black", 0.7)

# Draw one
i <- 1
drawbox(c(min(pvals),
          pvals[sum(qvals <= bins[i+1])]),
        bins[i:(i+1)],
        lty=1,
        lwd=2,
        col= col, border = border)

# Now the rest
for (i in c(3,7,12) ) {
  biny <- c(sum(qvals <= bins[i]),
            sum(qvals <= bins[i+1]))
  drawbox(pvals[biny],
          bins[c(i, i+1)],
          lty=1,
          lwd=2,
          col= col, border = border)
}
```

- The width of the box is proportional to the number of elements it contains.
  - The greater the width, the greater the concentration.
- We can produce all such boxes, with **fixed height**, to see how the concentration changes with $p$.

```
plot(pvals, qvals, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlim=c(0,1),
     xlab = "Proportion p",
```

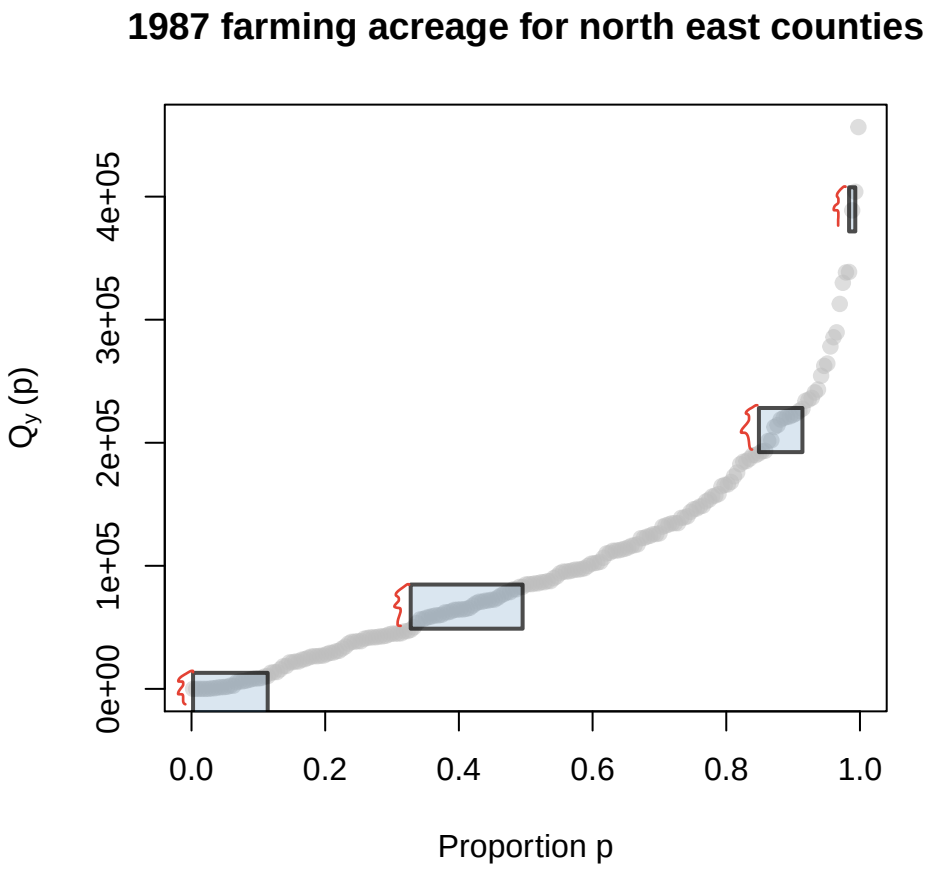



Figure 2: Concentration box on quantile plot

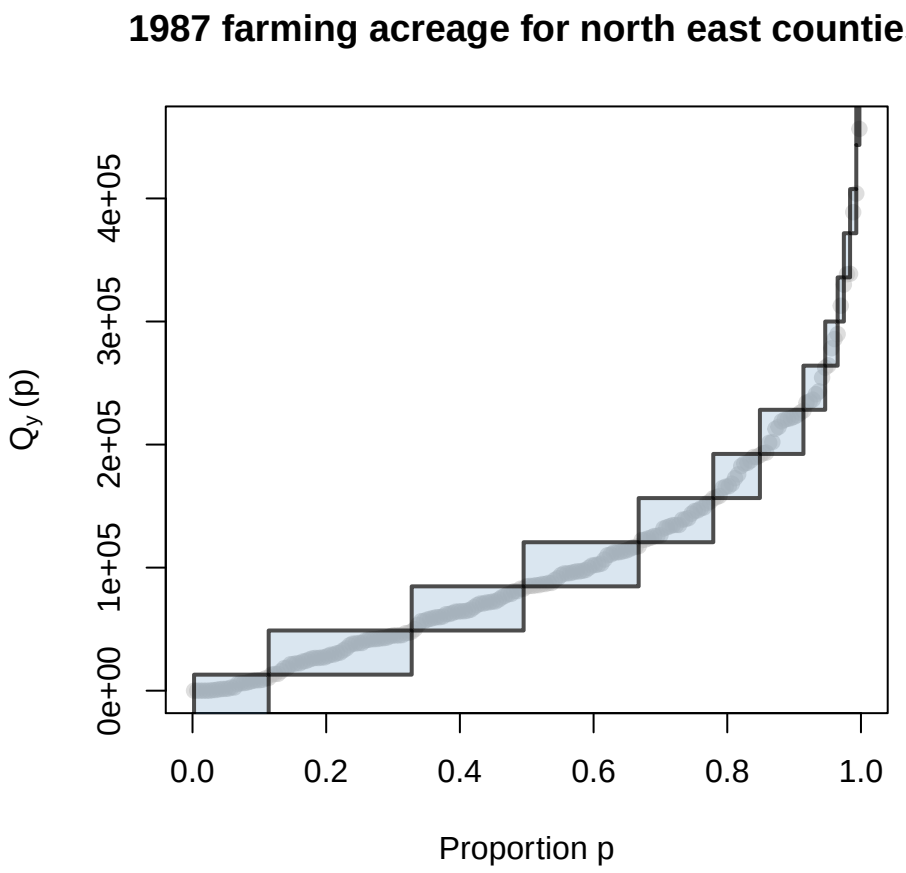



Figure 3: Contiguous concentration boxes on quantile plot

```
     ylab = bquote("Q"["y"]~"(p)"),
     main = "1987 farming acreage for north east counties")

# Draw first one
i <- 1
drawbox(c(min(pvals),
          pvals[sum(qvals <= bins[i+1])]),
        bins[i:(i+1)],
        lty=1,
        lwd=2,
        col= col, border = border)

# Now the rest
for (i in 2:length(bins)) {
  biny <- c(sum(qvals <= bins[i]),
            sum(qvals <= bins[i+1]))
  drawbox(pvals[biny],
          bins[c(i, i+1)],
          lty=1,
          lwd=2,
          col= col, border = border)
}
```

- So how do we interpret these boxes?
  - What happens if we move them all to the left edge of the plot?

We get a histogram!

```
# par(mfrow=c(1,2))
#
# plot(pvals, qvals, pch = 19, col=adjustcolor("grey", alpha = 0.5),
#      xlim=c(0,1),
#      xlab = "Proportion p",
#      ylab = bquote("Q"["y"]~"(p)"),
#      main = "1987 farming acreage for north east counties")
#
# # Draw first one
# i <- 1
# drawbox(c(min(pvals),
#           pvals[sum(qvals <= bins[i+1])]),
#         bins[i:(i+1)],
#         lty=1,
#         lwd=2,
#         col= col, border = border)
#
# # Now the rest
# for (i in 2:length(bins)) {
#   biny <- c(sum(qvals <= bins[i]),
#             sum(qvals <= bins[i+1]))
#   drawbox(pvals[biny],
#           bins[c(i, i+1)],
#           lty=1,
#           lwd=2,
#           col= col, border = border)
# }

plot(pvals, qvals, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlim=c(0,1),
     xlab = "Proportion p",
     ylab = bquote("Q"["y"]~"(p)"),
     main = "1987 farming acreage for north east counties")

# Draw first one
i <- 1
drawbox(c(0,
          pvals[sum(qvals <= bins[i+1])]),
        bins[i:(i+1)],
        lty=1,
        lwd=2,
        col= col, border = border)

# Now the rest
for (i in 2:length(qvals)) {
  biny <- c(sum(qvals <= bins[i]),
            sum(qvals <= bins[i+1]))
```

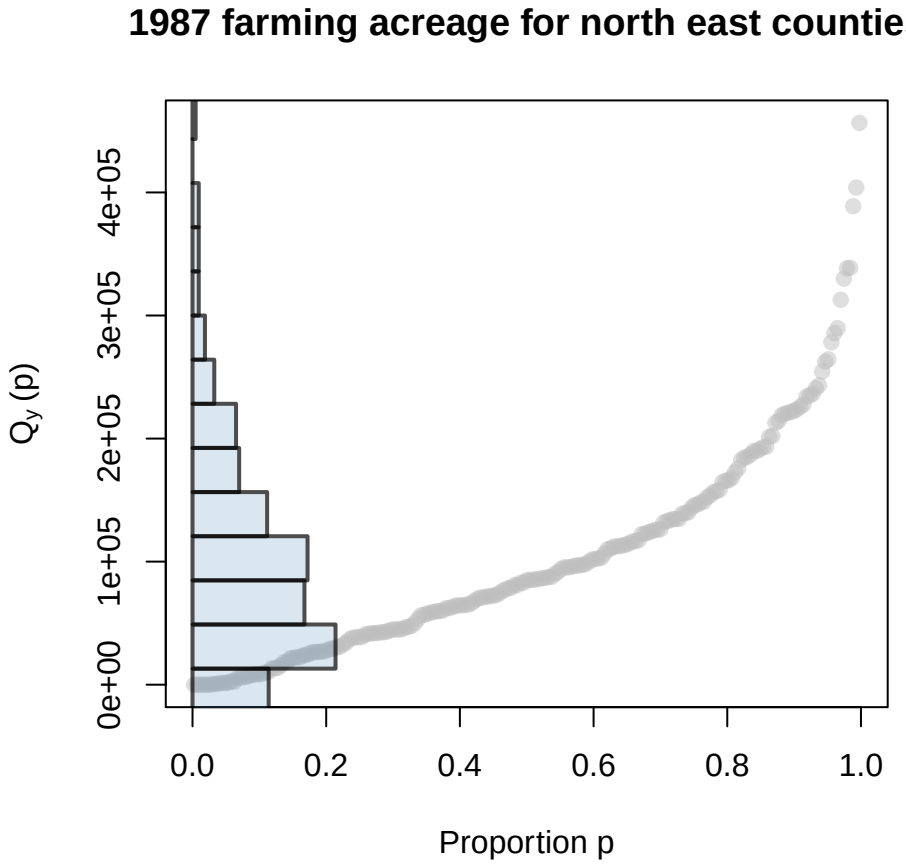



Figure 4: Contiguous concentration boxes on quantile plot

```
  drawbox(c(0, diff(abs(pvals[biny]))),
          bins[c(i, i+1)],
          lty=1,
          lwd=2,
          col= col, border = border)
}
```

- A histogram of the acreage (or any $y$ variate) is formed from the boxes that identify concentrations on the quantile plot!