STAT 3355
Introduction to Data Analysis

Lecture 06: Summaries for Univariate Data I

Created by: Qiwei Li
Assistant Professor of Statistics
Presented by: Octavious Smiley
Assistant Professor of Instruction

Department of Mathematical Sciences
The University of Texas at Dallas

Learning Goals

- Summarize a univariate data in two ways
  - Center
  - Spread
- Numerical summaries for discrete data
  - Center: The mode
- Graphical summaries for discrete data
  - Bar chart

Discrete Data

- Samples share a finite number of values (have ties)
- Data type
  - Integer (if the number of possible values is small)
  - Categorical data
  - Logical data
- Examples
  - The whole number of age of people in this class
  - Adult or nonadult of a person
  - The blood type of a person: A, B, AB, or O
  - The political party that a person vote for: Democratic, republican, etc.

Data Tabulating

- Denote a univariate discrete dataset by

$$\boldsymbol{x} = [x_1, \ldots, x_i, \ldots, x_n], \text{ where } x_i \in \{0, 1, \ldots, K-1\}$$

- Tabulating data is to obtain a frequency table, which is an integer vector

$$\boldsymbol{f} = [n_0, \ldots, n_k, \ldots, n_{K-1}], \text{ where } n_k = \sum_{i=1}^{n} I(x_i = k)$$

  Here $I(\cdot)$ is an indicator function

- Interpretation
  - The frequency of each possible value

Data Tabulating

- Implementation in R
  - $\boldsymbol{x}$ is a integer/factor/logical vector
    - `table(`x`)`
  - $\boldsymbol{x}$ is a integer/factor/logical variable in a data frame $\boldsymbol{X}$
    - `table(`X`$x_name)`
  - `NA` will be omitted
  - Output

Data type	Order
Integer	From smallest to largest
Factor	Alphabetical
Ordered factor	Self-defined levels
Logical	`FALSE`, `TRUE`

Data Tabulating

- Examples

```
library(UsingR)

# Load data babies
data("babies")

# Smoke variable
x <- babies$smoke
table(x)

# Turn x into a factor vector
x <- factor(x, labels = c("Never", "Now", "
    Until pregnancy", "Once but quit", "
    Unknown"))
table(x)
```

The Mode

- Denote a univariate discrete dataset by

$$\boldsymbol{x} = [x_1, \ldots, x_i, \ldots, x_n], \text{ where } x_i \in \{0, 1, \ldots, K-1\}$$

and the resulting frequency table by

$$\boldsymbol{f} = [n_0, \ldots, n_k, \ldots, n_{K-1}], \text{ where } n_k = \sum_{i=1}^{n} I(x_i = k)$$

- The mode

$$m = \text{argmax}_k \boldsymbol{f}$$

- Interpretation
  - The unique value in $\boldsymbol{x}$ that occurs most often

The Mode

- Implementation in R
  - $\boldsymbol{x}$ is a integer/factor/logical vector
    - `names(which.max(table(`x`)))`
  - $\boldsymbol{x}$ is a integer/factor/logical variable in a data frame $\boldsymbol{X}$
    - `names(which.max(table(`X`$x_name)))`

The Diversity

- Denote a univariate discrete dataset by

$$\boldsymbol{x} = [x_1, \ldots, x_i, \ldots, x_n], \text{ where } x_i \in \{0, 1, \ldots, K-1\}$$

and the resulting frequency table by

$$\boldsymbol{f} = [f_0, \ldots, f_k, \ldots, f_{K-1}], \text{ where } f_k = \sum_{i=1}^{n} I(x_i = k)$$

- The relative frequency table is defined by

$$\boldsymbol{p} = [p_0, \ldots, p_k, \ldots, p_{K-1}], \text{ where } p_k = \frac{f_k}{\sum_{j=0}^{K-1} f_j}$$

The Shannon Index

- The Shannon index

$$H_{\text{shannon}} = -\sum_{k=0}^{K-1} p_k \log p_k$$

- Originates from information science (Shannon, 1948)
- Interpretation
  - All the values have the same frequency, then $H_{\text{shannon}} = \log K$
  - The data has only one value, then $H_{\text{shannon}} = 0$
  - The more unequal the distribution of the types, the smaller the H_{shannon}
- Implementation in R: $p =$ `table(`x`)/sum(table(`x`))` and $H_{\text{shannon}} =$ `-sum(`p `* log(`p`))`

The Simpson Index

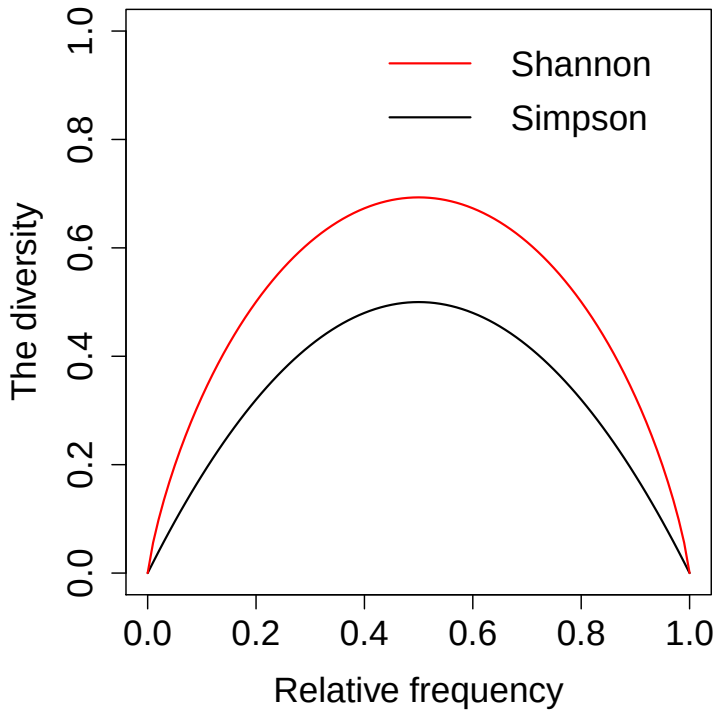
- The Simpson index

$$H_{\text{simpson}} = 1 - \sum_{k=0}^{K-1} p_k^2$$

- Originates from economics (Gini, 1912)
- Interpretation
  - All the values have the same frequency, then $H_{\text{simpson}} = 1 - 1/K$
  - The data has only one value, then $H_{\text{simpson}} = 0$
  - The more unequal the distribution of the types, the smaller the H
- Implementation in R: $p =$ `table(`x`)/sum(table(`x`))` and $H_{\text{simpson}} =$ `1 - sum(`p `* `p`))`

The Diversity

- Suppose there are only $K = 2$ categories
- If the relative frequency for one category is p, then the one for the other is $1 - p$



Your Turn

- Apply the function `table()` to the age variable in the dataset babies
- Identify the missing values and change them to `NA`
- Turn the integer vector to a factor vector by truncating values into 10s, 20s, 30s, and 40s
- What is the mode of the factor vector?
- What is the proportion of pregnancy women whose age were above 40?

Your Turn

- Solutions

```
# Load data
library(UsingR)
data("babies")
x <- babies$age

# Tabulate data
table(x)

# Change the value of 99 to NA
x[which(x == 99)] <- NA

# Turn x to a factor vector
x <- cut(x, breaks = c(0, 19, 29, 39, 49),
    labels = c("10s", "20s", "30s", "40s"))
```

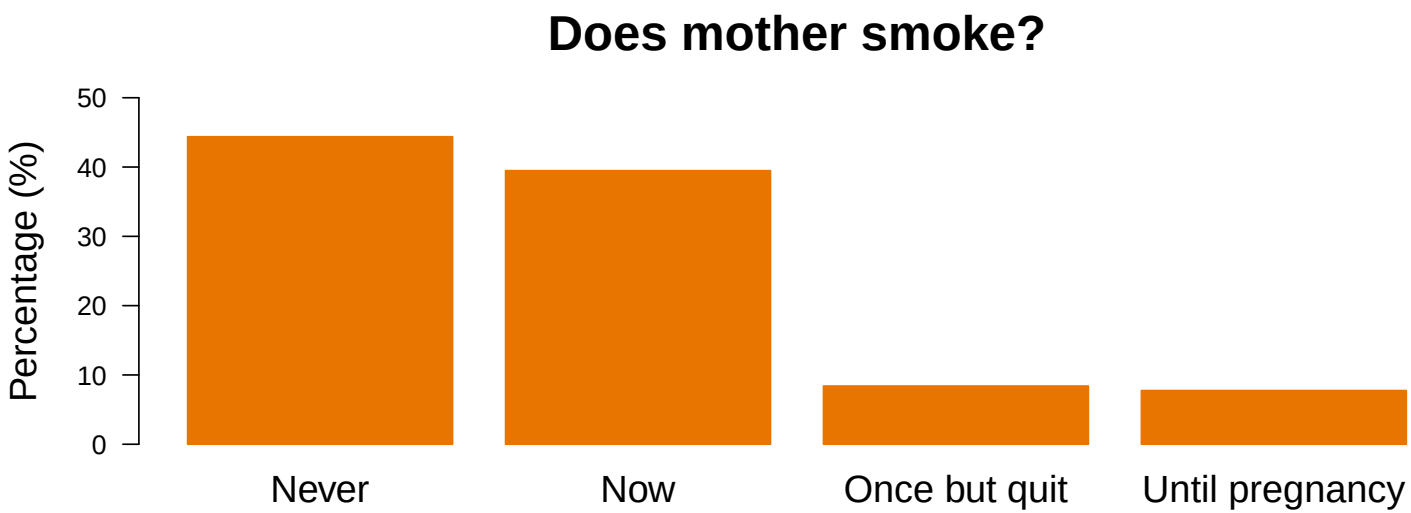
Your Turn

- Solutions

```
table(x)

# What is the mode
names(which.max(table(x)))

# What is the proportion of 40s women
round(table(x)["40s"]/sum(table(x)), 3)
```

Bar Chart

- Also known as barplots
  - One axis arranges the levels of a discrete univariate data in some order
  - The other axis represents their frequency with a bar of a height proportional to the frequency
- The input is the output of the function `table()`



Bar Chart

- Implementation in R
  - x is a integer/factor/logical vector
    - `barplot(table(`x`))`
  - x is a integer/factor/logical variable in a data frame X
    - `barplot(table(`X`$x_name))`

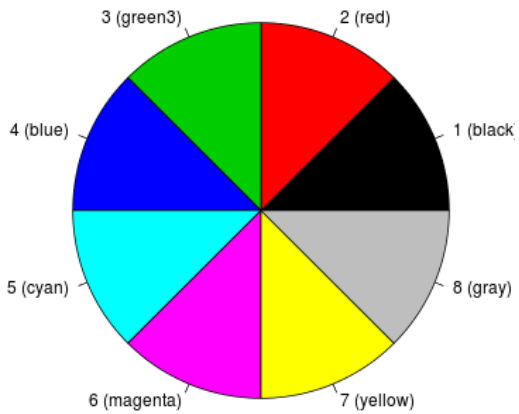
Bar Chart

- Examples

```
# Smoke variable
x <- babies$smoke
table(x)

# Change the value of 9 to NA
index_9 <- which(x == 9)
x[index_9] <- NA

# Turn x to a factor vector
x <- factor(x, labels = c("Never", "Now", "
    Until pregnancy", "Once but quit"))

# Plot the bar chart
barplot(table(x))
```

The Function `barplot()`

- Important arguments controlling the bars
  - `horiz`: A logical value for the orientation of the bars
  - `col`: A vector of colors for each bar



  - `border`: A vector of colors for the boarder of each bar
  - `ylim`: A numerical vector of two values indicating the limits for the axis that represents the frequency
  - `xpd`: A logical value indicating if bars go outside region

The Function `barplot()`

- Important arguments controlling the labels
  - `main`: A string of title
  - `names.arg`: A character vector of names for each bar
  - `xlab` and `ylab`: A string of label for the axis names
  - `las`: A numeric value of $\{0, 1, 2, 3\}$ for the orientation of axis labels

The Function `barplot()`

- Important arguments controlling the label size
  - `cex.main`: A numeric value for the title size
  - `cex.lab`: A numeric value for the size of axis labels
  - `cex.axis`: A numeric value for the size of x axis label
  - `cex.names`: A numeric value for the size of axis names

Discussions

- Sort the levels in terms of their frequencies
  - `sort(table(`x`))`
  - `sort(table(`x`), decreasing = TRUE)`
- Transfer frequency to relative frequency
  - `table(`x`)/sum(table(`x`))`
  - ~~`table(`x`)/length(`x`)`~~
- Mislead audience by truncating the y axis
  - `barplot(table(`x`), ylim = c( , ))`
- Barplots can be used to illustrate time-series data
  - `barplot(x)`

Time-series Data

- Examples

```
# Load data
data("central.park")

# Plot the average temperature in May 2003
    at Central Park, NYC
barplot(central.park$AVG)
```

Example

- Make the barplot of the average daily temperature in May 2003 at Central Park more informative and pretty
  - Name the bars from day 1 to 31
  - Name the x axis as "Days in May 2003" and name the y axis as "Temp. (Fahrenheit)"
  - Set the title as "Average Temperature at Central Park, NYC"
  - Limit the bottom of y axis to 32, which corresponds to freezing point of water
  - Color those days above the average temperature in May in red; otherwise in blue

Example

- Solutions

```
x <- central.park$AVG

# Name the bars from day 1 to 31
barplot(x, names.arg = 1:31)

# Name the x and y axis
barplot(x, names.arg = 1:31, xlab = "Days in
    May 2003", ylab = "Temp. (Fahrenheit)")

# Set the title
barplot(x, names.arg = 1:31, xlab = "Days in
    May 2003", ylab = "Temp. (Fahrenheit)",
   main = "Average Temperature at Central
   Park, NYC")
```

Example

- Solutions

```
# Limit the bottom of y axis to freezing
    point
barplot(x, names.arg = 1:31, xlab = "Days in
     May 2003", ylab = "Temp. (Fahrenheit)",
    main = "Average Temperature at Central
    Park, NYC", ylim = c(32, 75), xpd = FALSE
    )
```

Example

- Solutions

```
# Color each bar with respect to above or
    below the mean
cr <- rep("blue", 31)
index <- which(central.park$AVG > mean(
    central.park$AVG))
cr[index] <- "red"
```

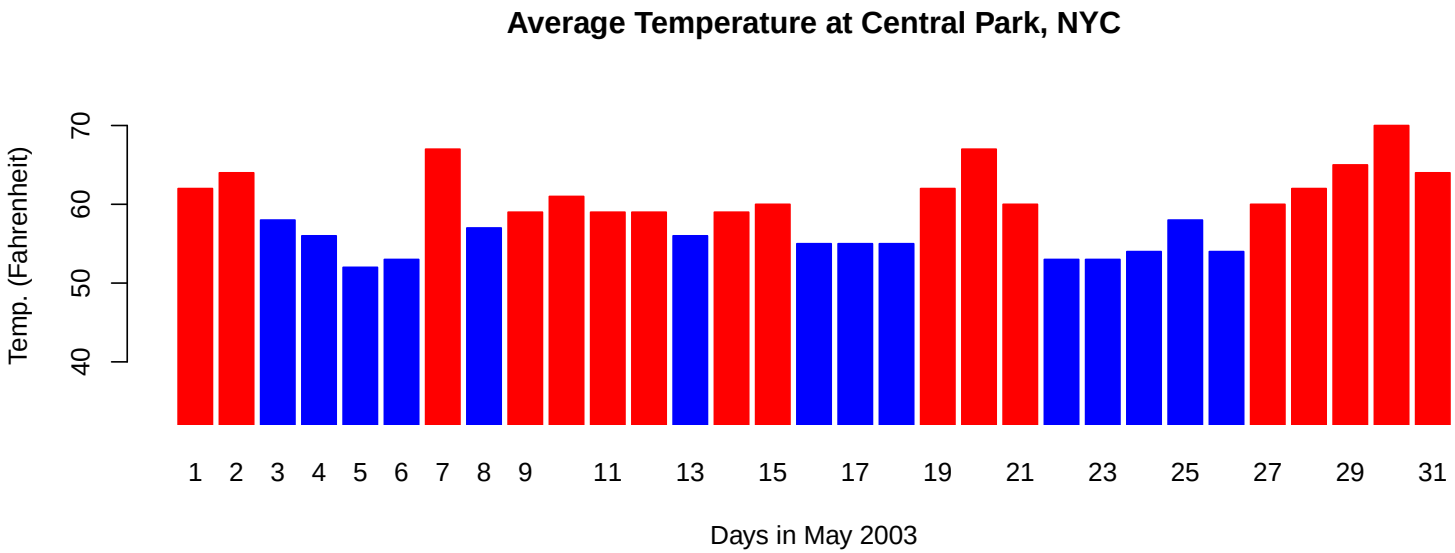
Example

- Solutions

```
barplot(x, names.arg = 1:31, xlab = "Days in
    May 2003", ylab = "Temp. (Fahrenheit)",
   main = "Average Temperature at Central
   Park, NYC", ylim = c(32, 75), xpd = FALSE
   , col = cr, border = cr)
```

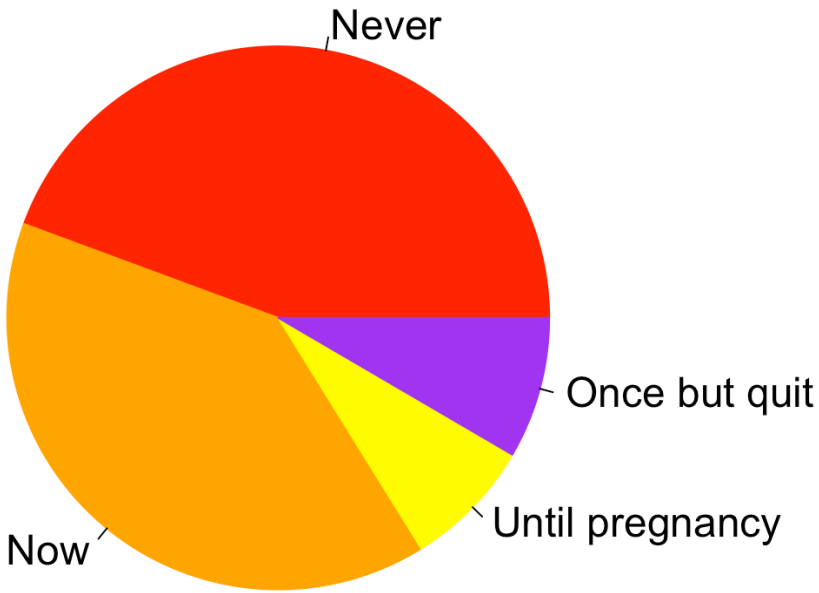
Example

- Final display



Pie Chart

- A common graphic to represent proportions
  - Wedges of a circle indicate the relative frequencies of each unique value in a discrete univariate data
- Fails at discerning differences
- Not suitable when the number of unique values is large



Pie Chart

- Implementation in R
  - x is a factor/logical vector
    - `pie(table(`x`))`
  - x is a factor/logical variable in a data frame X
    - `pie(table(`X`$x_name))`

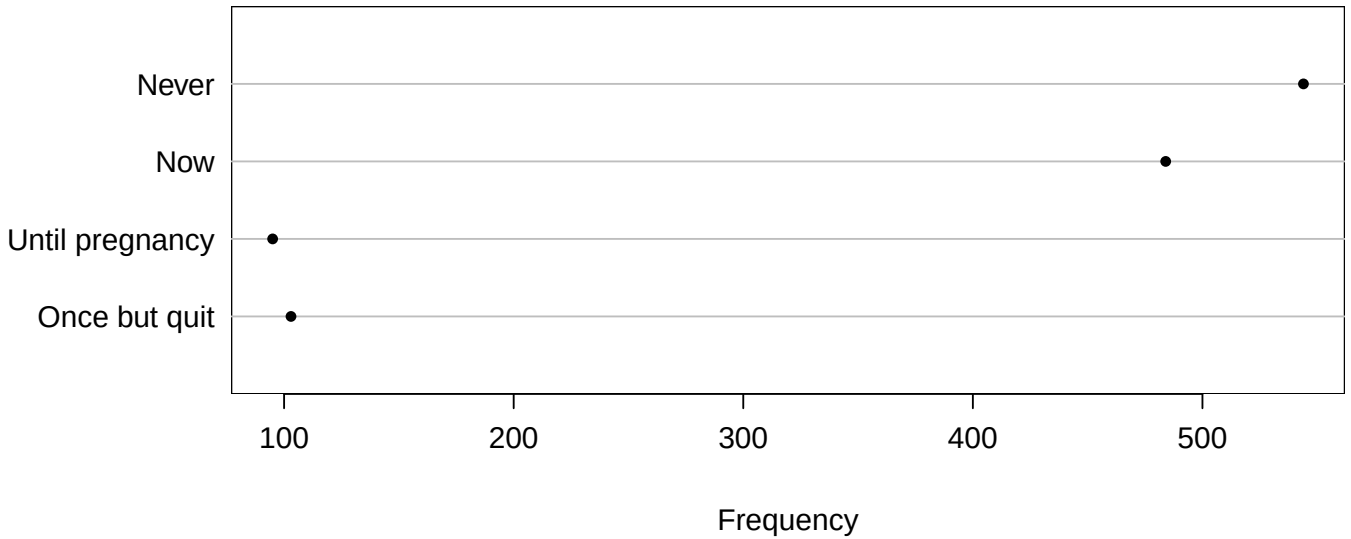
Pie Chart

- Example

```
# Load data
library(UsingR)
data("babies")
x <- babies$smoke

# Clean data
x[which(x == 9)] <- NA
x <- factor(x, labels = c("Never", "Now", "
    Until pregnancy", "Once but quit"))

# Plot data
pie(table(x), radius = 1, col = c("red", "
    orange", "yellow", "purple"), border = c(
    "red", "orange", "yellow", "purple"))
```

Dot Chart

- Also known as Cleveland dotplots
  - y axis arranges the levels of a discrete univariate data in some order
  - x axis represents their frequency over the range of the data
- Difference from the largest to the smallest is very obvious



Dot Chart

- Implementation in R
  - x is a factor/logical vector
    - `dotchart(table(`x`))` or `dotchart2(table(`x`))`
  - x is a factor/logical variable in a data frame X
    - `dotchart(table(`X`$x_name))` and `dotchart2(table(`X`$x_name))`

```
dotchart2(table(x), ylab = "Frequency")
```

After-class Reading

- *Using R for Introductory Statistics (1st Ed.)* by John Verzani
- Chapter 2 Univariate data
  - Section 2.1 Categorical data
    - Subsection 2.1.1 Tables
    - Subsection 2.1.2 Barplots
    - Subsection 2.1.3 Pie charts
    - Subsection 2.1.4 Dot charts
    - Subsection 2.1.5 Factors

After-class Reading

- *Using R for Introductory Statistics (2nd Ed.)* by John Verzani
- Chapter 2 Univariate data
  - Section 2.4 Categorical data