# STAT 3355
# Introduction to Data Analysis

## Lecture 09: Summaries for Univariate Data III

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

# Last Class

- Summarize a univariate data

| | | Continuous | |
|---|---|---|---|
| | Discrete | Value-based | Position-based |
| Numeric | `table(`$x$`)` | `mean(`$x$`)` | `median(`$x$`)` |
| | | `var(`$x$`)` | `IQR(`$x$`)` |
| | | | `quantile(`$x$`)` |
| | | | `min(`$x$`)` |
| | | | `max(`$x$`)` |
| | | | `range(`$x$`)` |
| Graphic | `barplot(table(`$x$`))` | | |

# Quiz 4

- Load the dataset `mpg` in the library `ggplot2`. Create a new variable $\texttt{cmb} = (\texttt{cty} + \texttt{hwy})/2$.
  - Problem 1: What is the mode of variable `cyl`?
  - Problem 2: What is the difference of the median combined mpg between front-wheel drive and four-wheel drive cars?
  - Problem 3: What is the sample standard deviation of combined mpg of toyota cars?

# Quiz 4

- Answers

```
# Load data
library(ggplot2)
data(mpg)

# Create the new variable
mpg$cmb <- (mpg$hwy + mpg$cty)/2

# Problem 1
names(which.max(table(mpg$cyl)))
```

# Quiz 4

- Answers

```
# Problem 2
fwd_index <- which(mpg$drv == "f")
awd_index <- which(mpg$drv == "4")
median(mpg$cmb[fwd_index]) - median(mpg$cmb[
    awd_index])

# Problem 3
toyota_index <- which(mpg$manufacturer == "
    toyota")
sd(mpg$cmb[toyota_index])
```

# Learning Goals

- Graphical summaries for continuous data
  - Histogram
  - Boxplot

# Table of Contents

# Histogram

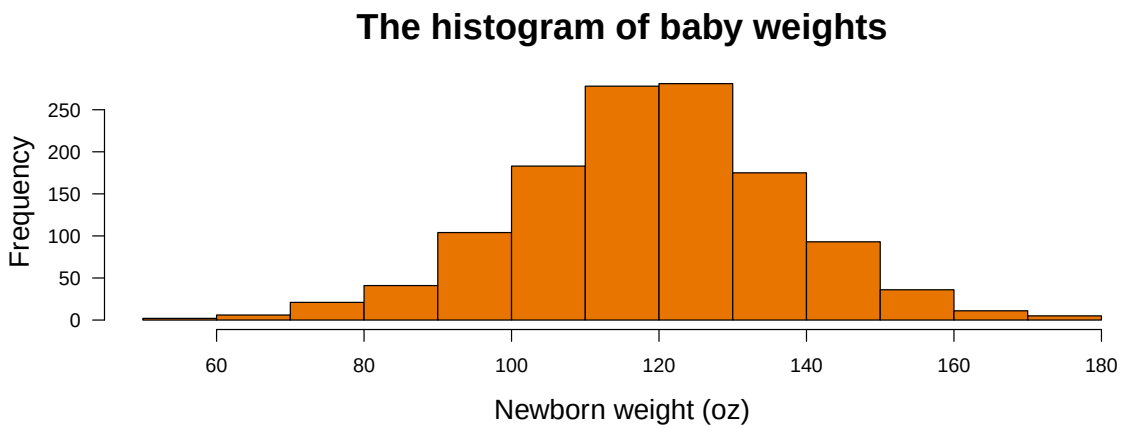
- A special bar chart that turns a numeric data into a ordinal data by binning

# Histogram

- A special bar chart that turns a numeric data into a ordinal data by binning
  - Break up an interval that covers all the values in $x$ into several bins (consecutive, non-overlapping, equally-sized)
  - Count the number of $x$ entries in each bin

# Histogram

- A special bar chart that turns a numeric data into a ordinal data by binning
  - Break up an interval that covers all the values in $x$ into several bins (consecutive, non-overlapping, equally-sized)
  - Count the number of $x$ entries in each bin
  - x axis arranges the bins
  - y axis represents their frequency with a bar of a height proportional to the frequency



# Histogram

- Implementation in R
  - $x$ is a numeric vector
    - `hist(`$x$`)`
  - $x$ is a numeric variable in a data frame $X$
    - `hist(`$X$`$x_name)`

# Histogram

- Examples

```
# Load data babies
library(UsingR)
data("babies")

# Baby weight variable
x <- babies$wt

# Draw the histogram
hist(x)
```

# The Function `hist()`

- Important arguments controlling the bars
  - `breaks`:
    - A number that gives the approximate number of bins
    - A numeric vector that gives the breakpoints between bins, e.g. via the function `seq(min(`$x$`) , max(`$x$`), length.out =)`

# The Function `hist()`

- Important arguments controlling the bars
  - `breaks`:
    - A number that gives the approximate number of bins
    - A numeric vector that gives the breakpoints between bins, e.g. via the function `seq(min(`$x$`) , max(`$x$`), length.out =)`
  - `col`: A number/string of a color for all bars
    - Colors in `R`
    - `rgb(red = , green = , blue = , alpha =)`
  - `border`: A number/string of a color for all bar borders

# The Function `hist()`

- Important arguments controlling the bars
  - `breaks`:
    - A number that gives the approximate number of bins
    - A numeric vector that gives the breakpoints between bins, e.g. via the function `seq(min(`$x$`) , max(`$x$`), length.out =)`
  - `col`: A number/string of a color for all bars
    - Colors in `R`
    - `rgb(red = , green = , blue = , alpha =)`
  - `border`: A number/string of a color for all bar borders
  - `xlim` and `ylim`: A numerical vector of two values indicating the limits of the axis

# The Function `hist()`

- Important arguments controlling the labels
  - `freq`: A logical value for a representation of frequencies or probability densities
  - `main`: A string of title
  - `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 tick labels

# The Function `hist()`

- 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 axis tick labels

# The Choices of Bin Number

$$k = \left\lceil \frac{x_{[n]} - x_{[1]}}{h} \right\rceil$$

- The look depends on the bin number $k$ or bin width $h$
  - Small $k$: blocky
  - Large $k$: spiky

# The Choices of Bin Number

$$k = \left\lceil \frac{x_{[n]} - x_{[1]}}{h} \right\rceil$$

- The look depends on the bin number $k$ or bin width $h$
  - Small $k$: blocky
  - Large $k$: spiky
- Existing methods make strong assumptions about the data
  - Square-root choice: $k = \lceil \sqrt{n} \rceil$
  - Sturges' formula: $k = 1 + \lceil \log_2 n \rceil$
  - Rice rule: $k = \lceil 2\sqrt[3]{n} \rceil$

# The Choices of Bin Number

$$k = \left\lceil \frac{x_{[n]} - x_{[1]}}{h} \right\rceil$$

- The look depends on the bin number $k$ or bin width $h$
  - Small $k$: blocky
  - Large $k$: spiky
- Existing methods make strong assumptions about the data
  - Square-root choice: $k = \lceil \sqrt{n} \rceil$
  - Sturges' formula: $k = 1 + \lceil \log_2 n \rceil$
  - Rice rule: $k = \lceil 2\sqrt[3]{n} \rceil$
  - Scott's normal reference rule: $h = \frac{3.5s}{\sqrt[3]{n}}$
  - Freedman-Diaconi's choice: $h = 2\frac{\text{IQR}}{\sqrt[3]{n}}$

# The Choices of Bin Number

- Examples

```
x <- babies$wt
n <- length(x)

# Square-root choice
k <- ceiling(sqrt(n))

# Sturges' formula
k <- 1 + ceiling(log2(n))

# Rice rule
k <- ceiling(2*n^(1/3))

hist(x, breaks = seq(min(x), max(x), length.
    out = k + 1), xlab = "Weight", main = "")
```

# Your Turn

- Continue to work on the wt variable in the babies dataset
  - Calculate the Scott's normal reference for the bin width $h$ and the corresponding bin number $k$ via the formula $k = \left\lceil \frac{x_{[n]} - x_{[1]}}{h} \right\rceil$
  - Draw the resulting histogram in UTD Eco Green color ('#154734')
  - Repeat the above steps for the Freedman-Diaconi's choice

# Your Turn

- Solutions

```
# Scott's normal reference
h <- 3.5 * sqrt(var(x)) / n^(1/3)
k <- ceiling((max(x) - min(x)) / h)

# Plot the histogram
hist(x, breaks = seq(min(x), max(x), length.
    out = k + 1), xlab = "Weight", main = "",
     col = "#008542", las = 1)
```

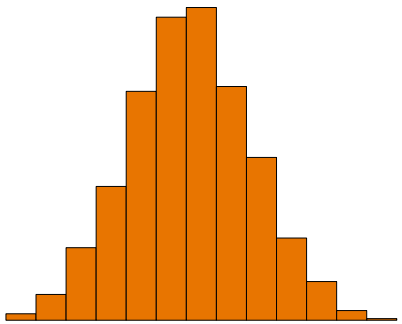
# Your Turn

- Solutions

```
# Freedman-Diaconi's choice
h <- 2 * IQR(x) / n^(1/3)
k <- ceiling((max(x) - min(x)) / h)

# Plot the histogram
hist(x, breaks = seq(min(x), max(x), length.
    out = k + 1), xlab = "Weight", main = "",
     col = "#008542", las = 1)
```

# Discussions

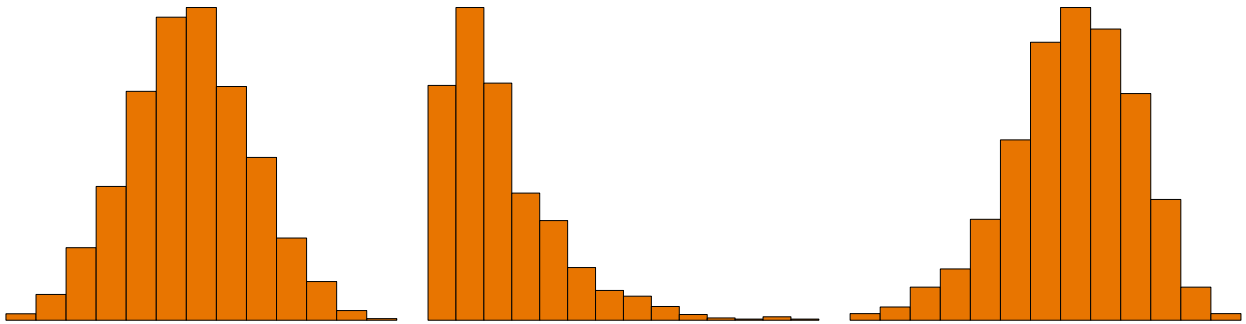
- Easy to identity the center
  - The peak is the mode
  - The balancing point is the mean
  - The point splitting the area into half is the median
- Easy to identity the spread
- Easy to identity the shape



Symmetric

# Discussions

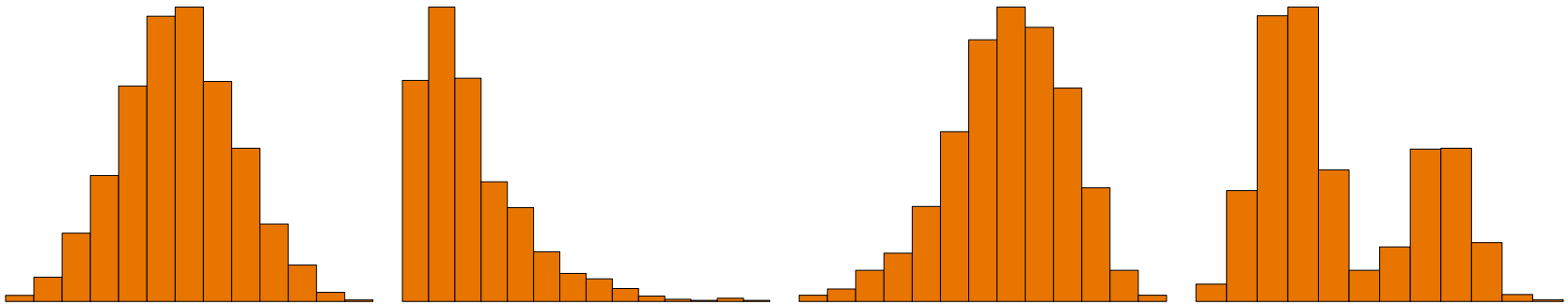
- Easy to identity the center
  - The peak is the mode
  - The balancing point is the mean
  - The point splitting the area into half is the median
- Easy to identity the spread
- Easy to identity the shape



Symmetric Right-skewed Left-skewed

# Discussions

- Easy to identity the center
  - The peak is the mode
  - The balancing point is the mean
  - The point splitting the area into half is the median
- Easy to identity the spread
- Easy to identity the shape



Symmetric Right-skewed Left-skewed Bi-modal

# Histogram

- Examples

```
data("exec.pay")
x <- exec.pay
n <- length(x)

# Freedman-Diaconi's choice
h <- 2*IQR(x)/(n^(1/3))
k <- ceiling((max(x) - min(x))/h)

hist(x, breaks = seq(min(x), max(x), length.
    out = k + 1), xlab = "Compensation (10k)"
    )
hist(x, breaks = seq(min(x), max(x), length.
    out = k + 1), xlim = c(0, 200), xlab = "
    Compensation (10k)")
```

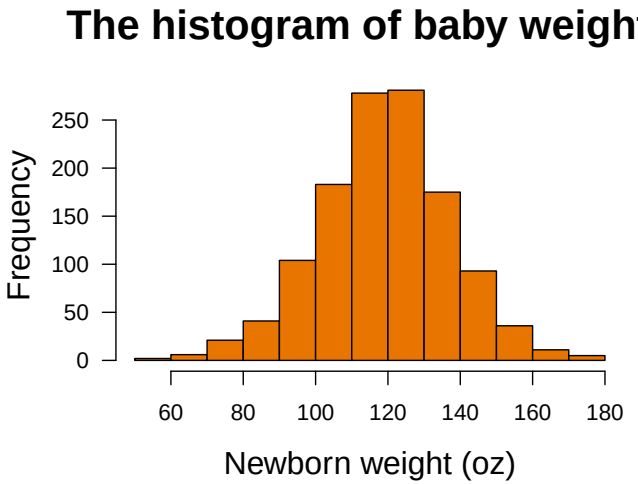
# Discussions

- Difficult for comparing samples from different groups

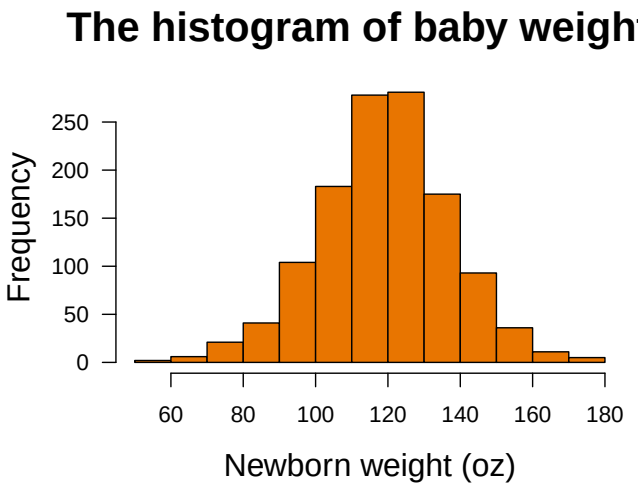
# Discussions

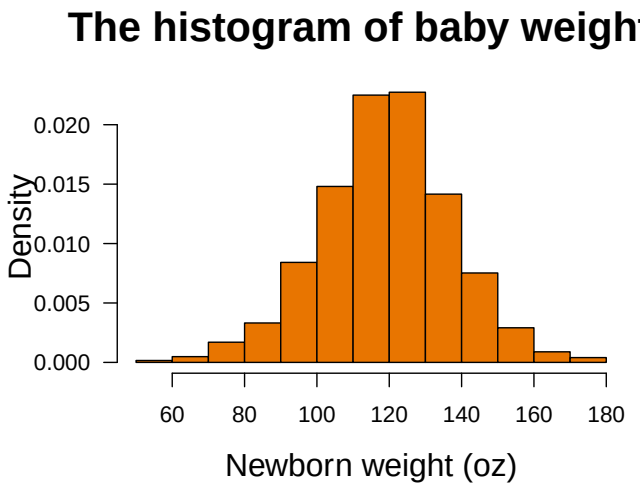
- Difficult for comparing samples from different groups
  - Combine multiple plots into one graph via the function `par(mfrow = c( , ))`
  - Density plot via the function `plot(density())`

# Density Plot

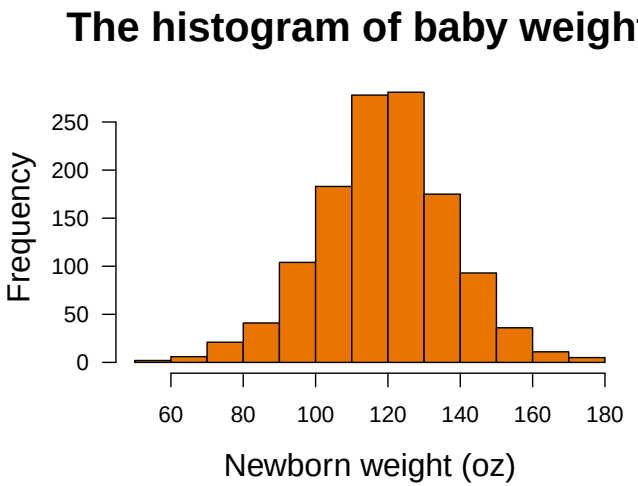


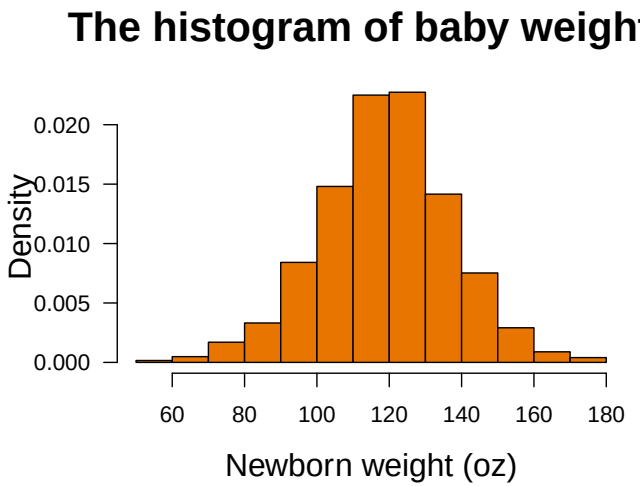
# Density Plot

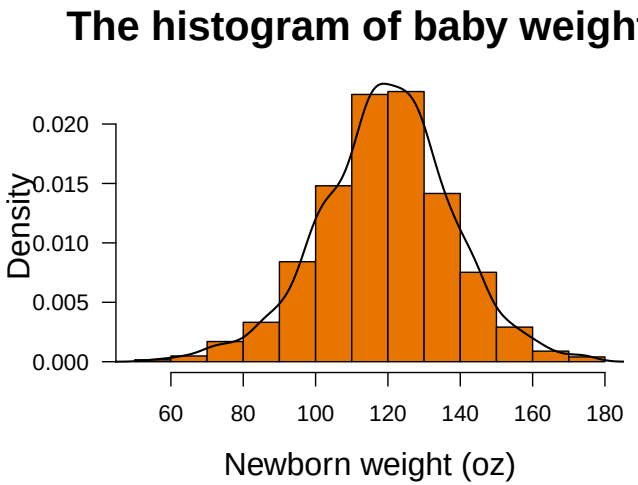


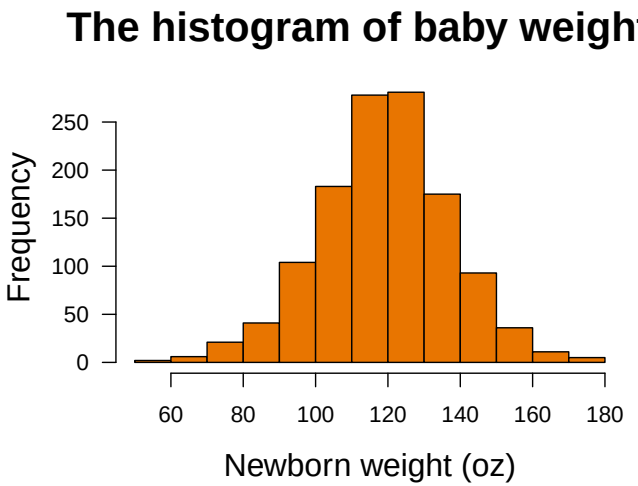
# Density Plot

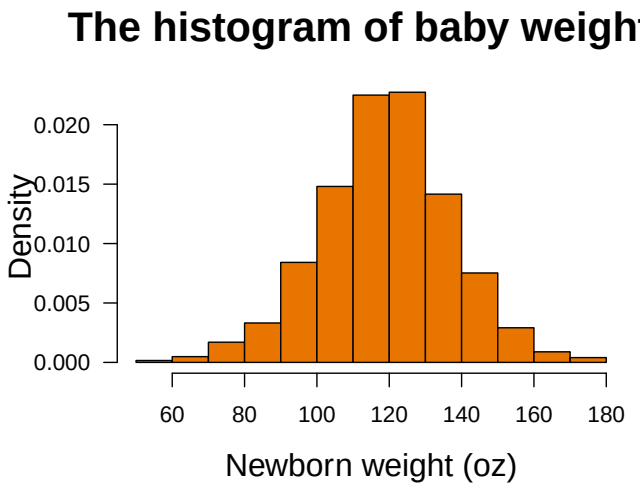


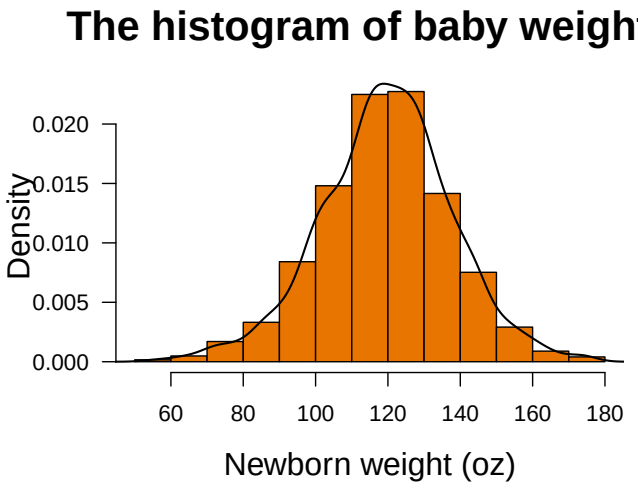



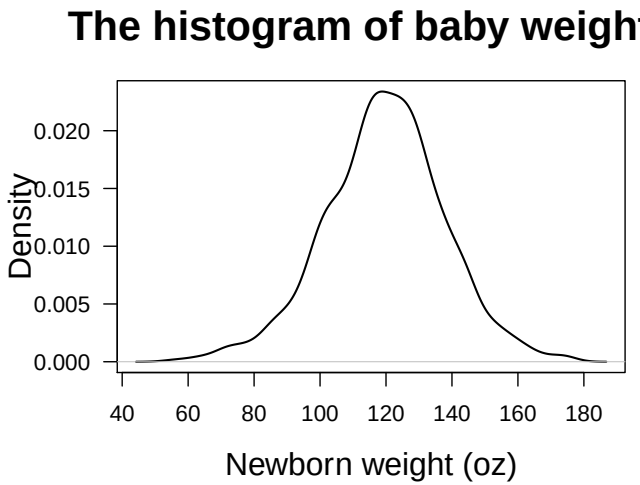



# Density Plot



# Density Plot

- Visualizes the probability distribution of the data by drawing a continuous curve
  - x axis represents the value of the data
  - y axis represents the probability density
  - The height of the curve is scaled such that the area under the curve equals one

# Density Plot

- Implementation in R
  - $\boldsymbol{x}$ is a numeric vector
    - `plot(density(`$x$`))`
    - `hist(`$x$`, freq = FALSE)` and `lines(density(`$x$`))`
  - $\boldsymbol{x}$ is a numeric variable in a data frame $\boldsymbol{X}$
    - `plot(density(`$X$`$x_name))`
    - `hist(`$X$`$x_name, freq = FALSE)` and `lines(density(`$X$`$x_name))`

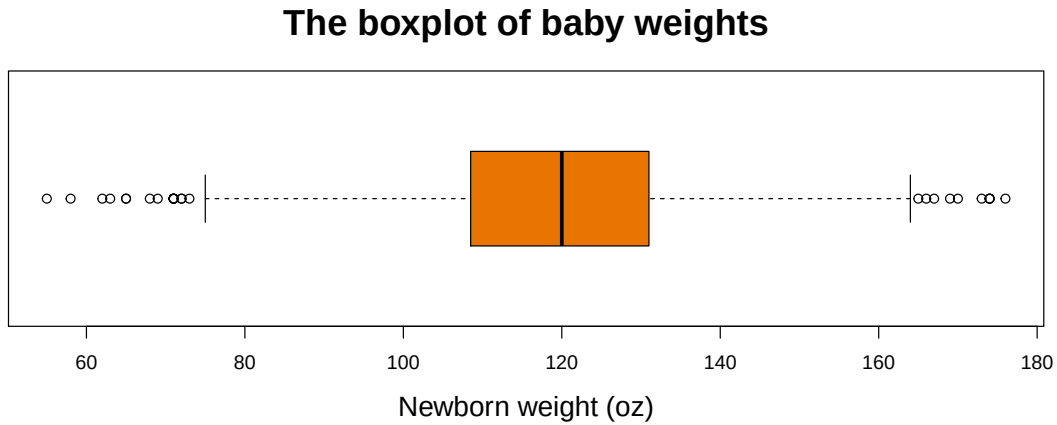
# Density Plot

- Examples

```
# Baby weight variable
x <- babies$wt

# Histogram
hist(x, xlab = "", freq = TRUE, las = 1)

# Histogram and density plot
hist(x, xlab = "", freq = FALSE, las = 1)
lines(density(x))

# Density plot only
plot(density(x), xlab = "", las = 1, main =
    "Density plot of x")
```

# Boxplot

- Displays the five-number summary (the output of the function `summary()`)
  - One axis represents the value of the data
  - A box is drawn from $Q(0.25)$ to $Q(0.75)$, representing the IQR
  - A thick line through the box indicates $Q(0.5)$, i.e. the median
  - Whiskers are drawn from $\max(Q(0.25) - 1.5\text{IQR}, Q(0))$ and $\max(Q(0.75) + 1.5\text{IQR}, Q(1))$ to the box
  - Points are the outliers



# Boxplot

- Implementation in R
  - $\boldsymbol{x}$ is a numeric vector
    - `boxplot(`$x$`)`
  - $\boldsymbol{x}$ is a numeric variable in a data frame $\boldsymbol{X}$
    - `boxplot(`$X$`$x_name)`

# Boxplot

- Examples

```
# Baby weight variable
x <- babies$wt

# Get the quantile summary
summary(x)

# Draw the boxplot
boxplot(x)
boxplot(x, horizontal = TRUE)
```

# The Function `boxplot()`

- Important arguments controlling the box
  - `horizontal`: A logical value for the orientation of the box
  - `range`: A number that determines how far the whiskers extend out from the box
  - `outline`: A logical value for drawing the outlines defined by the `range`
  - `col`: A vector of colors for each box
  - `border`: A vector of colors for the boarder of each box
  - `ylim`: A numerical vector of two values indicating the limits for the axis that represents the values

# The Function `boxplot()`

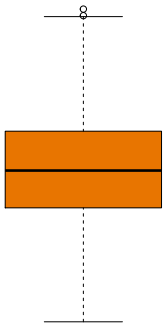
- Important arguments controlling the labels
  - `main`: A string of title
  - `xlab` and `ylab`: A string of label for the axis names
  - `las`: A numeric value of $\{0, 1, 2, 3\}$ for the orientation of axis labels

# The Function `boxplot()`

- 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

# Discussions

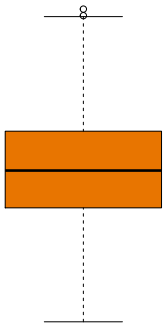
- Easy to identity the center: The median
- Easy to identity the spread: The IQR
- Easy to identity the shape: The location of the median within the box and the lengths of the two whiskers.



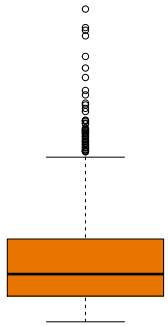
Symmetric

# Discussions

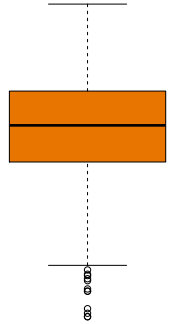
- Easy to identity the center: The median
- Easy to identity the spread: The IQR
- Easy to identity the shape: The location of the median within the box and the lengths of the two whiskers.



Symmetric



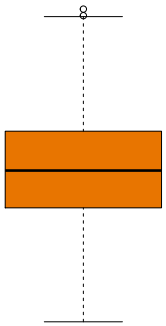
Right-skewed



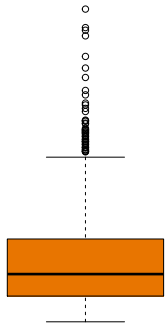
Left-skewed

# Discussions

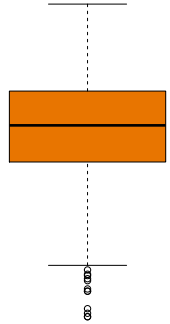
- Easy to identity the center: The median
- Easy to identity the spread: The IQR
- Easy to identity the shape: The location of the median within the box and the lengths of the two whiskers.



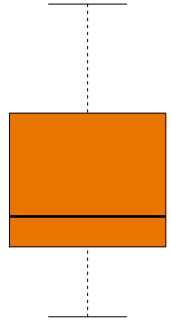
Symmetric



Right-skewed



Left-skewed



Bi-modal

# Combine Multiple Plots

- Stack multiple plots into one graph via the function `par(mfrow = c( , ))`
- The graphics device will remain divided until changing it back with `par(mfrow = c(1, 1))`

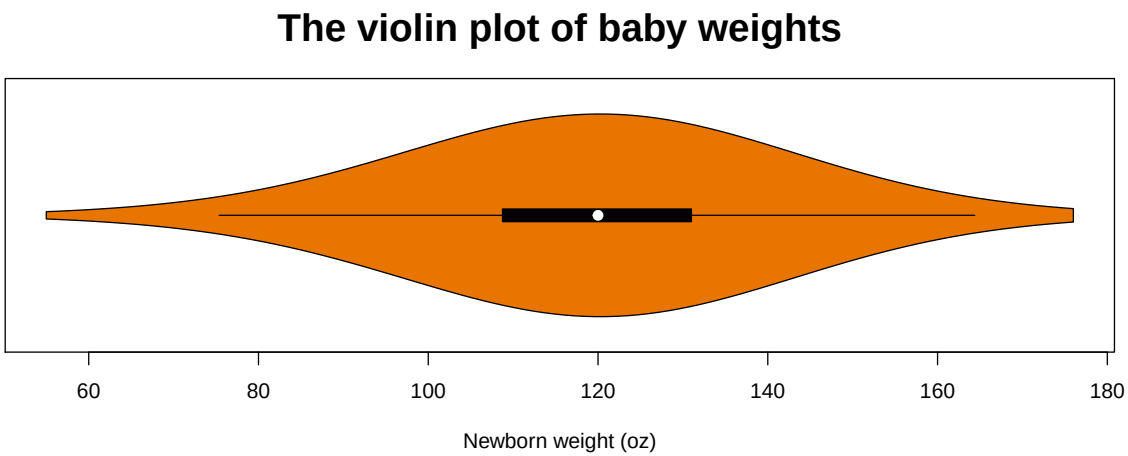
# Combine Multiple Plots

- Examples

```
par(mfrow = c(2, 1))
hist(x, main = "Histogram", xlab = "", ylab
    = "", xlim = c(60, 180))
boxplot(x, main = "Boxplot", horizontal =
    TRUE, ylim = c(60, 180))
par(mfrow = c(1, 1))
```

# Violin Plot

- A hybrid of boxplot and histogram
  - A boxplot within
  - A histogram smoothed by a kernel density estimator on each side
- More informative, such as the mode and the full distribution of the data
- Harder to grasp the meanings due to the unpopularity



# Violin Plot

- Implementation in R
  - Don't use the function `violinplot()` in the package `UsingR`
  - Install the package `vioplot`
  - $\boldsymbol{x}$ is a numeric vector
    - `vioplot(`$x$`)`
  - $\boldsymbol{x}$ is a numeric variable in a data frame $\boldsymbol{X}$
    - `vioplot(`$X$`$x_name)`

# Violin Plot

- Examples

```
violinplot(x, col = "orange")

library(vioplot)
vioplot(x, horizontal = TRUE, xlab = "
    Newborn weight (oz)", col = "orange",
    main = "The violin plot of baby weights",
     cex.main = 1.8)
```

# After-class Reading

- *Using R for Introductory Statistics (1st Ed.)* by John Verzani
- Chapter 2 Univariate data
  - Section 2.3 Shape of a distribution
    - Subsection 2.3.1 Histogram
    - Subsection 2.3.2 Modes, symmetry, and skew
    - Subsection 2.3.3 Boxplots

# After-class Reading

- *Using R for Introductory Statistics (2nd Ed.)* by John Verzani
- Chapter 2 Univariate data
  - Section 2.3 Numeric summaries