Data Visualization II

Distributions and ggplot2

Yen-Yi Ho

Department of Statistics, University of South Carolina

Section 1

Part I: Visualizing Data Distributions

Why Visualize Distributions?

Numerical summaries (mean, SD) **oversimplify** complex datasets.

> *"The most basic statistical summary of a list of objects or numbers is its distribution."*

Visualization helps us understand:

- Center, spread, and shape
- Symmetry and skewness
- Outliers and unexpected patterns
- Data quality issues

Types of Variables

Categorical

- **Unordered**: sex, region, race
- **Ordinal**: small / medium / large
- Summary: proportions per category

Numerical

- **Discrete**: count data
- **Continuous**: height, weight, rates
- Summary: requires richer tools

The **heights** dataset: 812 males, 238 females from a statistics class

The Heights Dataset

```
library(dslabs)
data(heights)

# Separate by sex
male   <- heights$height[heights$sex == "Male"]
female <- heights$height[heights$sex == "Female"]

# Basic summaries
summary(male)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  50.00   67.00   69.00   69.31   72.00   82.68
```

Histograms: Counting Within Bins

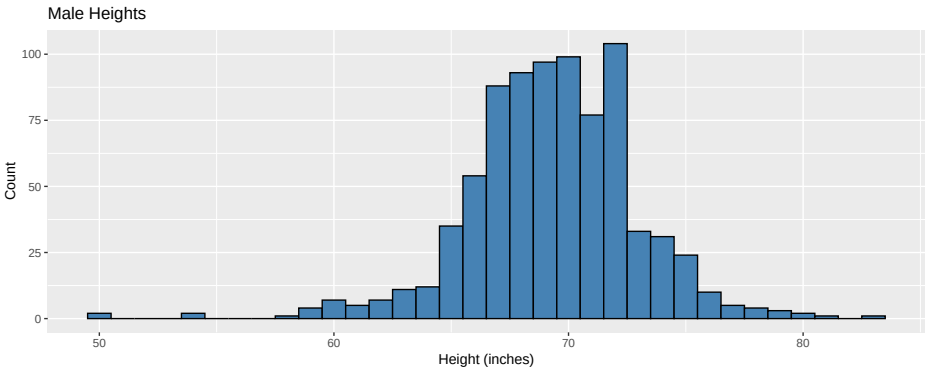
Idea: Divide the range into non-overlapping bins of equal width; count observations in each.

Histograms: The Code

```
heights |>
  filter(sex == "Male") |>
  ggplot(aes(height)) +
  geom_histogram(binwidth = 1, fill = "steelblue",
                 color = "black") +
  labs(title = "Male Heights",
       x = "Height (inches)", y = "Count")
```

Key parameter: `binwidth` — too wide loses detail, too narrow adds noise.

Histograms: The Result



Histograms: What They Reveal

From the male height histogram (1-inch bins, 63–81 in.):

- Distribution is **close to symmetric** around 69 inches
- About **95%** of males fall between 63 and 75 inches
- Roughly **bell-shaped** — suggesting a Normal distribution

Limitation: Bin choice can mislead. Try several binwidths.

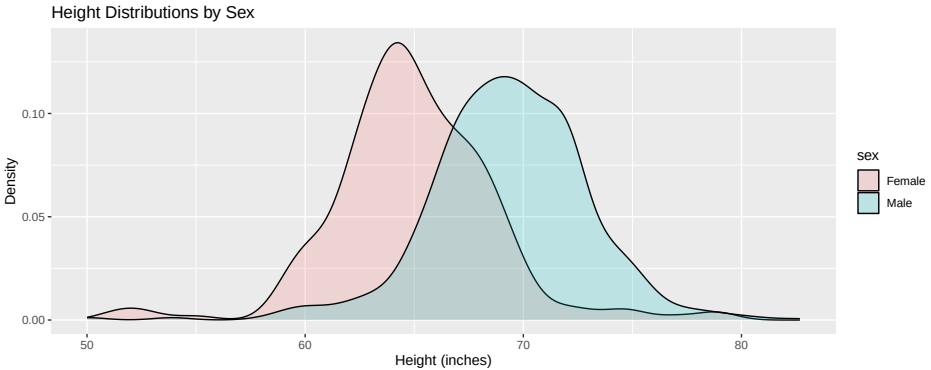
Smooth Density Plots

Density plots draw a **smooth curve** through histogram tops.

- Y-axis = **density** (not count): area under curve = 1
- Proportions ⇒ areas between two values
- Easier to **compare multiple groups**

```
heights |>
  ggplot(aes(height, fill = sex)) +
  geom_density(alpha = 0.2) +
  labs(title = "Height Distributions by Sex",
       x = "Height (inches)", y = "Density")
```

The Results


Height Distributions by Sex
Density
0.10
0.05
0.00
50
60
70
80
Height (inches)
sex
Female
Male

The Normal Distribution

Many real-world continuous variables follow (approximately) a **Normal** distribution.

Fully described by **two parameters**:

- μ = mean (center)
- σ = standard deviation (spread)

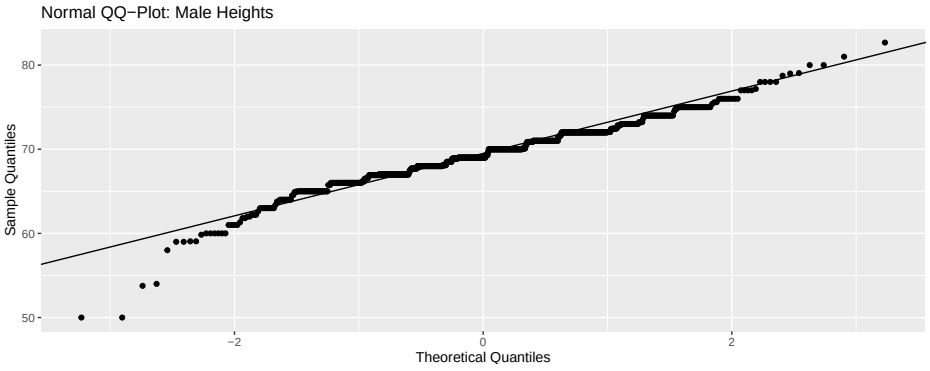
```
m <- mean(male)   # approx 69.31 inches
s <- sd(male)     # approx  3.61 inches
c(average = m, sd = s)
```

The **68–95–99.7 rule**: 68% within 1 SD, 95% within 2 SD, 99.7% within 3 SD.

Checking Normality: The Normal QQ-Plot

```
heights |>
  filter(sex == "Male") |>
  ggplot(aes(sample = height)) +
  geom_qq() +
  geom_qq_line() +
  labs(title = "Normal QQ-Plot: Male Heights",
       x = "Theoretical Quantiles",
       y = "Sample Quantiles")
```

Normal QQ Plot



Points falling on the line $\Rightarrow$ data are approximately Normal.

Percentiles and Quantiles

```
# Selected percentiles for male heights
quantile(male, c(0.10, 0.25, 0.50, 0.75, 0.90))
```

```
     10%      25%      50%      75%      90%
65.00000 67.00000 69.00000 72.00000 73.22751
```

```
# Proportion between 69 and 72 inches
mean((male > 69) & (male <= 72))
```

```
[1] 0.3337438
```

Quartiles divide data into four equal parts:

- **Q1** = 25th percentile
- **Q2** = 50th percentile (median)
- **Q3** = 75th percentile

Boxplots: Five-Number Summary

A boxplot encodes:

- Minimum (whisker end)
- Q1 (box bottom)
- Median (center line)
- Q3 (box top)
- Maximum (whisker end)
- Outliers (individual points)

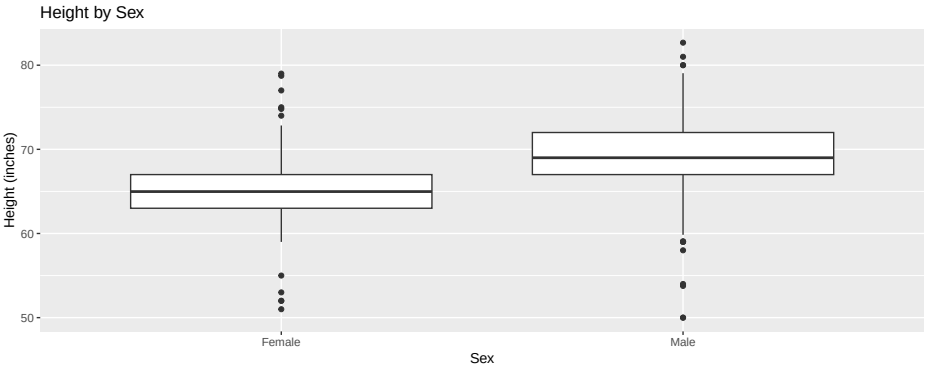
IQR $= Q3 - Q1$

- Box spans Q1 to Q3
- Whiskers extend $1.5\times$ IQR
- Points beyond whiskers $=$ outliers
- Compact: shows shape **and** spread
- Easy to compare across groups

Boxplots: R Code

```
heights |>
  ggplot(aes(sex, height)) +
  geom_boxplot() +
  labs(title = "Height by Sex",
       x = "Sex",
       y = "Height (inches)")
```

Boxplots: Results



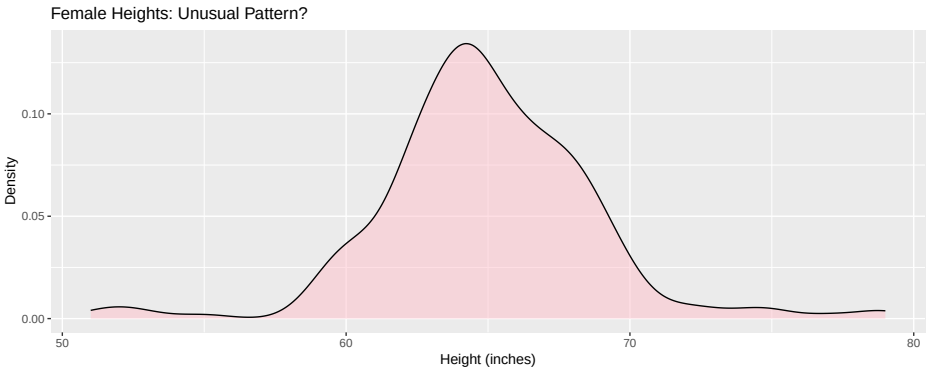
Case Study: Unexpected Patterns

Female height density showed a **second bump** — an anomaly!

```
heights |>
  filter(sex == "Female") |>
  ggplot(aes(height)) +
  geom_density(fill = "pink", alpha = 0.5) +
  labs(title = "Female Heights: Unusual Pattern?",
       x = "Height (inches)", y = "Density")
```

Likely cause: Males selecting "Female" by default $\Rightarrow$ a **data quality issue**.

Result


Female Heights: Unusual Pattern?
Density
0.10
0.05
0.00
50
60
70
80
Height (inches)

Stratification: Comparing Groups

Stratify = split observations into groups defined by a variable.

```
heights |>
  ggplot(aes(sex, height, fill = sex)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Heights Stratified by Sex",
       x = NULL, y = "Height (inches)")
```

Males average **3.5 inches taller** than females in this dataset.

Summary: Choosing a Distribution Plot

Goal	Plot Type
Overview of shape	Histogram
Smooth shape, compare groups	Density plot
Five-number summary, outliers	Boxplot
Check Normality	QQ-plot
One categorical variable	Bar chart

Section 2

Part II: Introduction to ggplot2

What is ggplot2?

ggplot2 is R's most popular visualization package, based on the *Grammar of Graphics*.

Key idea: A plot is built by combining independent components — like sentences built from grammar rules.

> *"Master a handful of verbs, nouns, and adjectives and you can construct many different sentences."*

Install once: `install.packages("ggplot2")` or via `tidyverse`.

The Grammar of Graphics

Every ggplot2 figure has these core components:

Component	Function	Example
Data	the dataset	`murders`
Geometry	visual mark type	`geom_point()`
Aesthetic mapping	data $\rightarrow$ visual property	`aes(x, y, color)`
Scale	axis/color transformation	`scale_x_log10()`
Labels	titles, axis labels	`labs()`
Theme	non-data styling	`theme_bw()`

The Basic Syntax

```
# General pattern:
data |>
  ggplot(aes(x = var1, y = var2)) +
  GEOM_FUNCTION() +
  SCALE_FUNCTION() +
  LABS_FUNCTION() +
  THEME_FUNCTION()
```

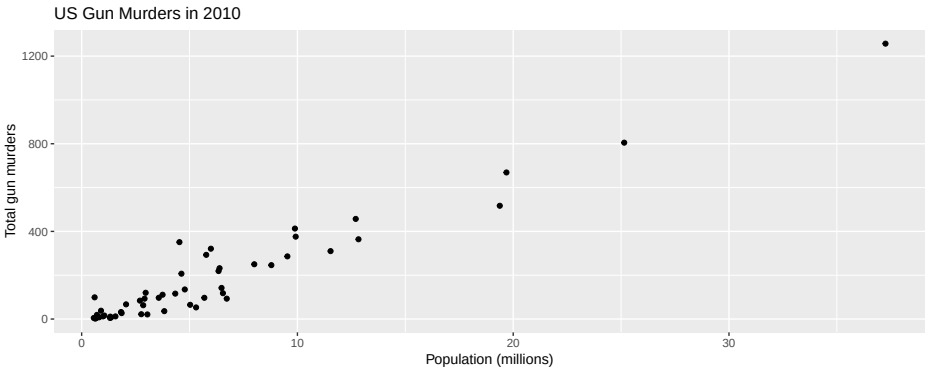
- `ggplot()` initializes the plot
- `aes()` maps variables to visual properties
- Layers are **added** with +
- Each layer is independent and composable

Your First ggplot2 Plot

```
data(murders)

murders |>
  ggplot(aes(x = population / 10^6,
             y = total)) +
  geom_point() +
  labs(x = "Population (millions)",
       y = "Total gun murders",
       title = "US Gun Murders in 2010")
```

murders columns: state, abb, region, population, total.

Result


US Gun Murders in 2010
Total gun murders
1200
800
400
0
0
10
20
30
Population (millions)

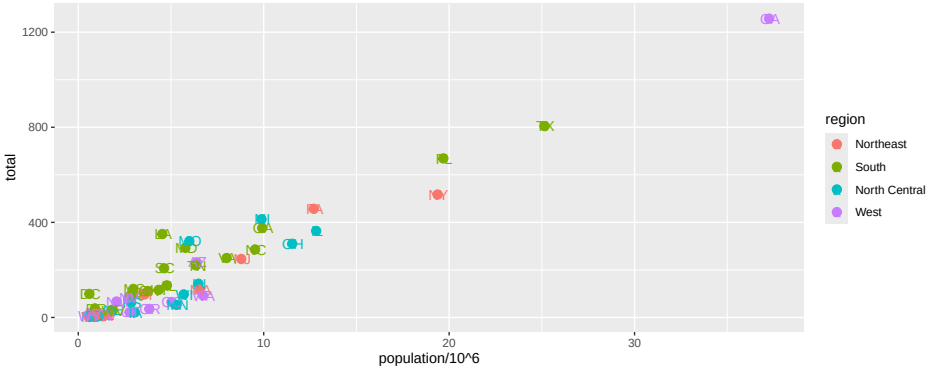
Aesthetic Mappings with `aes()`

Map **data variables** to visual properties inside `aes()`:

```
murders |>
  ggplot(aes(population / 10^6, total,
             color = region,
             label = abb)) +
  geom_point(size = 3) +
  geom_text(nudge_x = 0.05)
```

Rule: Data-driven properties ⇒ inside `aes()`. Fixed properties ⇒ outside `aes()`.

Results

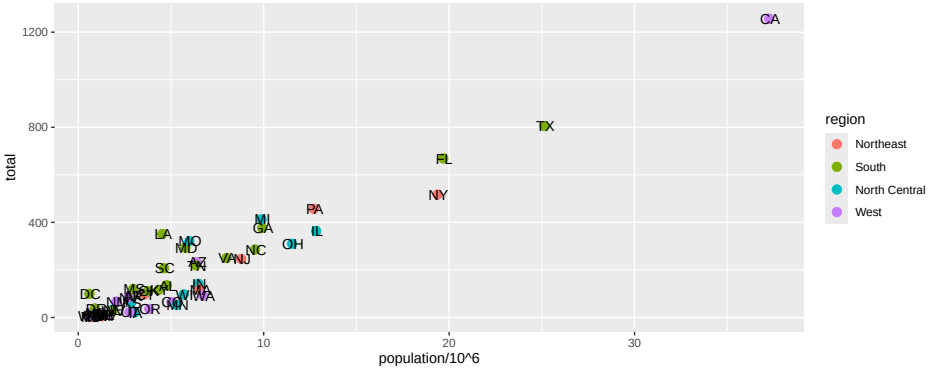

1200
800
400
0
total
0
10
20
30
population/10^6
region
Northeast
South
North Central
West

Global vs. Local Mappings

```
# Global: applies to ALL layers
p <- murders |>
  ggplot(aes(population / 10^6, total))

# Local: applies to ONE layer only
p +
  geom_point(aes(color = region), size = 3) +
  geom_text(aes(label = abb), nudge_x = 0.05)
```

Global mappings reduce repetition. Local mappings override for specific layers.

Results


total
1200
800
400
0
0
10
20
30
population/10^6
CA
TX
FL
NY
PA
IL
OH
NC
GA
NJ
MI
LA
MO
MD
VA
TN
SC
IN
WA
region
Northeast
South
North Central
West

Log Scale Transformations

Gun murders data spans orders of magnitude ⇒ use **log scale**.

```
murders |>
  ggplot(aes(population / 10^6, total,
             color = region, label = abb)) +
  geom_point(size = 3) +
  geom_text_repel(aes(label = abb)) +
  scale_x_log10() +
  scale_y_log10() +
  labs(title = "US Gun Murders in 2010",
       x = "Population (log scale, millions)",
       y = "Total murders (log scale)",
       color = "Region")
```

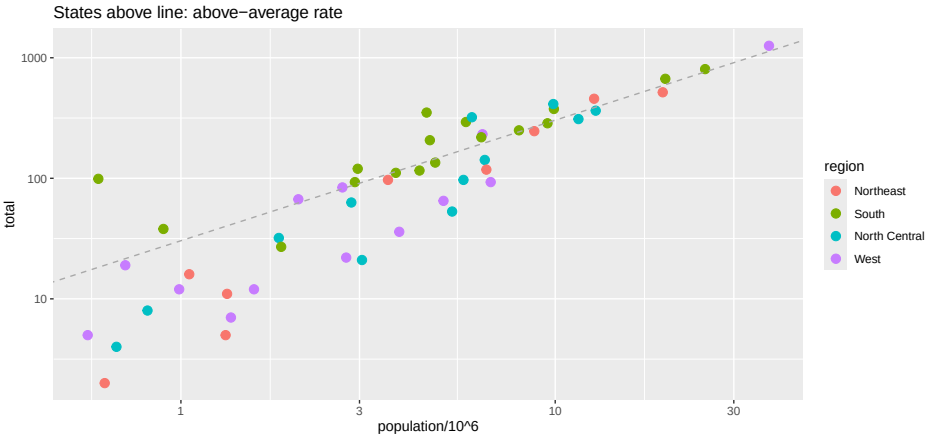
Results


US Gun Murders in 2010
Total murders (log scale)
1000
100
10
Population (log scale, millions)
1
3
10
30
CA
TX
FL
NY
PA
IL
MI
GA
LA
MO
MD
VA
NC
OH
AZ
NJ
SC
TN
IN
MS
AL
OK
MA
AR
KY
CT
WI
WA
DC
NV
NM
KS
CO
MN
OR
DE
NE
WV
UT
IA
AK
RI
ID
MT
ME
HI
SD
WY
NH
ND
VT
Region
Northeast
South
North Central
West

Adding Reference Lines

```
r <- murders |>
  summarize(rate = sum(total) /
              sum(population) * 10^6) |>
  pull(rate)

murders |>
  ggplot(aes(population / 10^6, total,
             color = region)) +
  geom_point(size = 3) +
  scale_x_log10() + scale_y_log10() +
  geom_abline(intercept = log10(r),
              lty = 2, color = "darkgrey") +
  labs(title = "States above line: above-average rate")
```

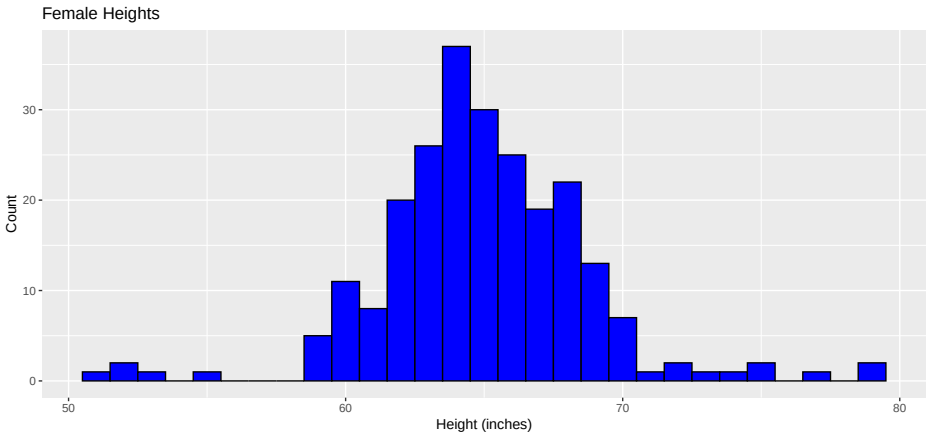
Results


States above line: above−average rate
total
1000
100
10
1
3
10
30
population/10^6
region
Northeast
South
North Central
West

Geom Gallery: Histograms

```
heights |>
  filter(sex == "Female") |>
  ggplot(aes(height)) +
  geom_histogram(binwidth = 1,
                 fill = "blue",
                 color = "black") +
  labs(title = "Female Heights",
       x = "Height (inches)", y = "Count")
```

Result



Key arguments: `binwidth`, `fill` (interior color), `color` (border color).

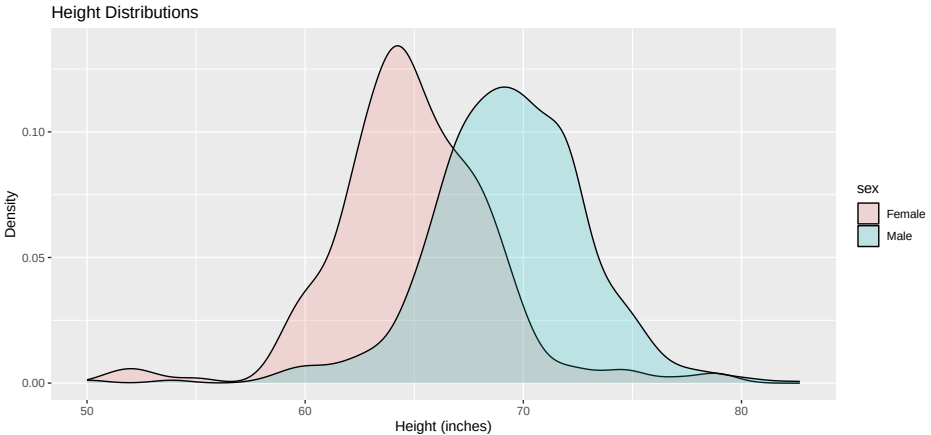
Geom Gallery: Density Plots

```
heights |>
  ggplot(aes(height, fill = sex)) +
  geom_density(alpha = 0.2) +
  labs(title = "Height Distributions",
       x = "Height (inches)", y = "Density")
```

`alpha` controls transparency (0 = invisible, 1 = opaque).

`fill = sex` inside `aes()` maps sex to fill color automatically.

Result

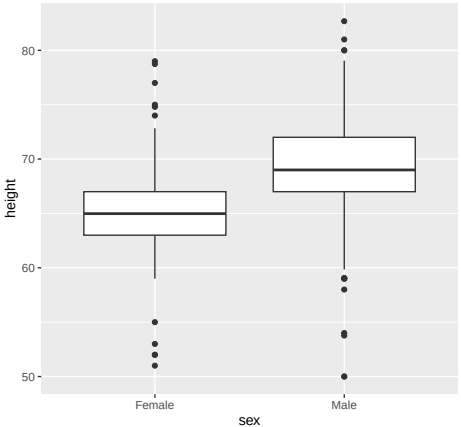

Height Distributions
Density
0.10
0.05
0.00
50
60
70
80
Height (inches)
sex
Female
Male

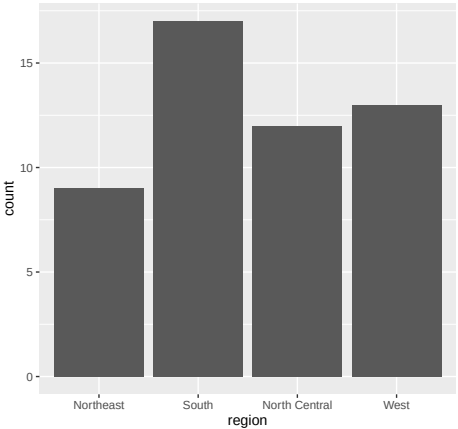
Geom Gallery: Boxplots and Bar Charts

```
p1 <- heights |>
  ggplot(aes(sex, height)) +
  geom_boxplot()

p2 <- murders |>
  ggplot(aes(region)) +
  geom_bar()

library(gridExtra)
grid.arrange(p1, p2, ncol = 2)
```

Use `geom_bar()` for counts; `geom_col()` for pre-computed values.

Result


80
70
60
50
height
Female
Male
sex


15
10
5
0
count
Northeast
South
North Central
West
region

Themes and Styling

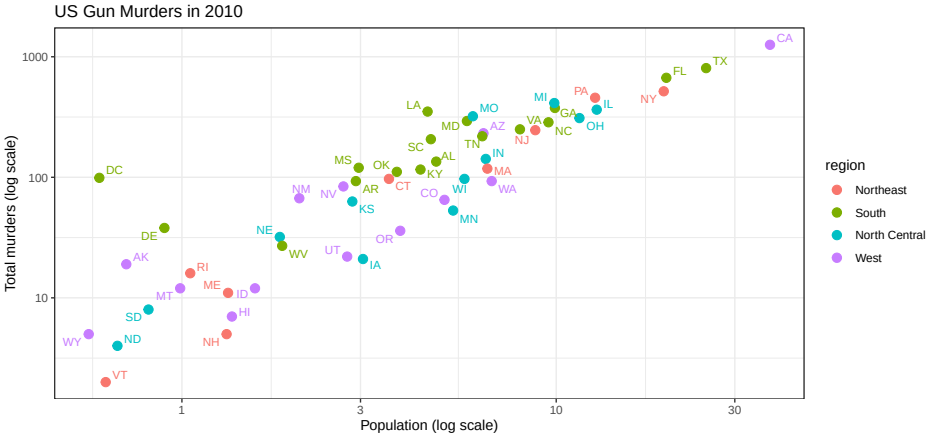
```
library(ggthemes)

p <- murders |>
  ggplot(aes(population / 10^6, total,
             color = region)) +
  geom_point(size = 3) +
  scale_x_log10() + scale_y_log10()

p + theme_bw()
p + theme_economist()
p + theme_fivethirtyeight()
p + theme_minimal()
```

Result: economist

Avoiding Label Overlap with ggrepel

```
library(ggrepel)

murders |>
  ggplot(aes(population / 10^6, total,
             color = region)) +
  geom_point(size = 3) +
  geom_text_repel(aes(label = abb), size = 3) +
  scale_x_log10() + scale_y_log10() +
  theme_bw() +
  labs(title = "US Gun Murders in 2010",
       x = "Population (log scale)",
       y = "Total murders (log scale)")
```

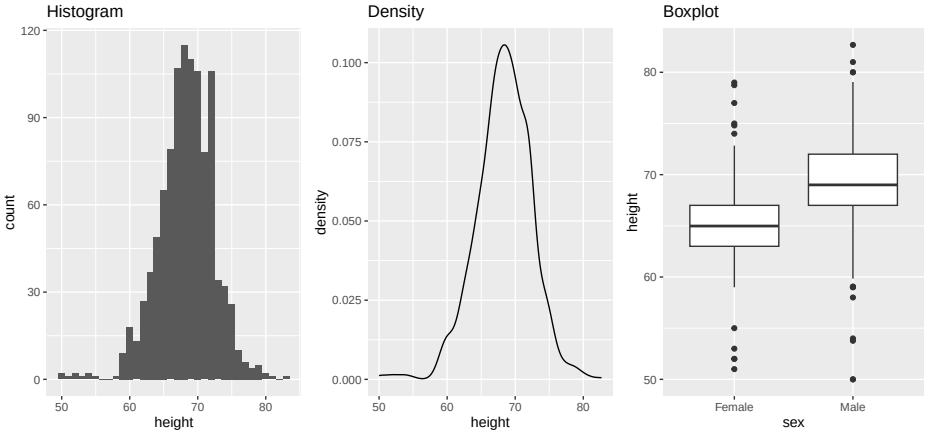
Result


US Gun Murders in 2010
Total murders (log scale)
1000
100
10
Population (log scale)
1
3
10
30
CA
TX
FL
NY
PA
IL
MI
GA
NC
OH
NJ
VA
AZ
MD
MO
LA
SC
TN
IN
AL
MS
OK
KY
MA
DC
NM
NV
AR
CT
CO
WI
WA
KS
MN
NE
WV
OR
UT
IA
DE
AK
RI
ME
ID
MT
HI
SD
ND
NH
WY
VT
region
Northeast
South
North Central
West

Multiple Plots with gridExtra

```
library(gridExtra)

p1 <- heights |>
  ggplot(aes(height)) +
  geom_histogram(binwidth = 1) +
  ggtitle("Histogram")
p2 <- heights |>
  ggplot(aes(height)) +
  geom_density() +
  ggtitle("Density")
p3 <- heights |>
  ggplot(aes(sex, height)) +
  geom_boxplot() +
  ggtitle("Boxplot")
grid.arrange(p1, p2, p3, ncol = 3)
```

Result

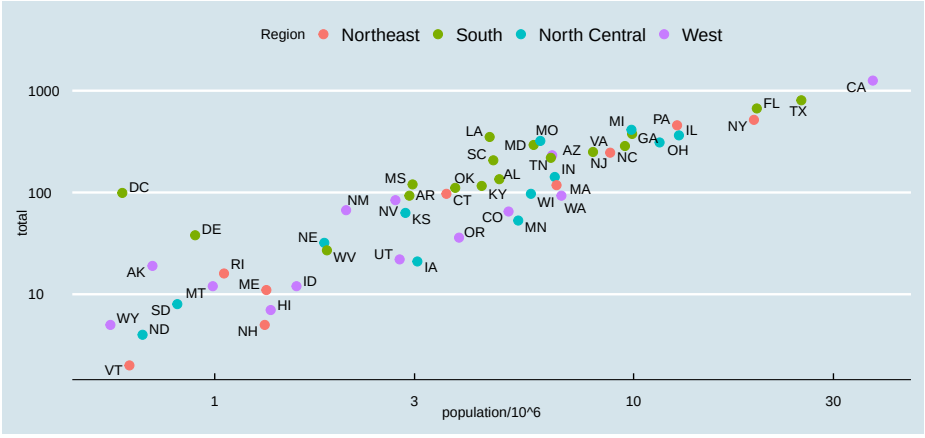

Histogram
count
height
Density
density
height
Boxplot
height
sex
Female
Male

Building Plots Incrementally

A good workflow — assign to objects and add layers:

```
p <- murders |>
  ggplot(aes(population / 10^6, total))

p <- p + geom_point(aes(color = region), size = 3)
p <- p + geom_text_repel(aes(label = abb))
p <- p + scale_x_log10() + scale_y_log10()
p <- p + theme_economist() + labs(color = "Region")
p  # display
```

Result

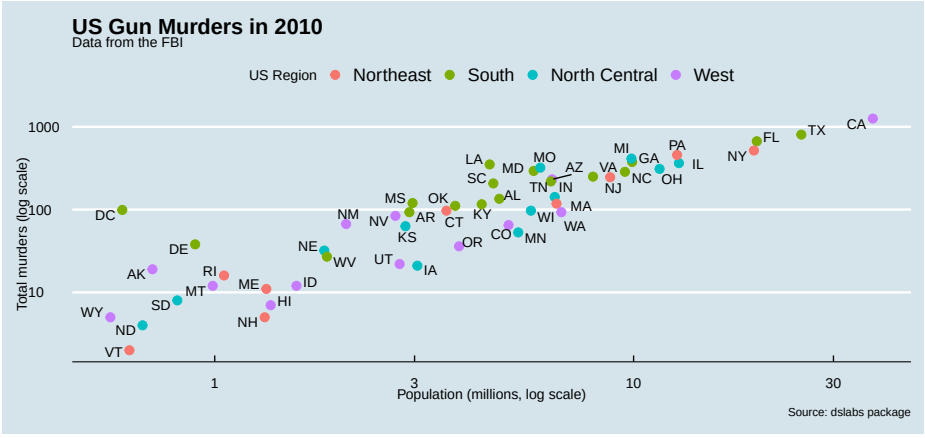

Region
Northeast
South
North Central
West
1000
100
10
total
DC
DE
AK
RI
ME
MT
ID
SD
HI
WY
ND
NH
VT
NE
WV
NM
NV
KS
MS
AR
OK
CT
KY
AL
LA
MO
MD
SC
AZ
TN
IN
NJ
VA
NC
GA
OH
MI
PA
IL
NY
FL
TX
CA
UT
IA
OR
CO
MN
WI
MA
WA
1
3
10
30
population/10^6

The `labs()` Function

Control all text labels from one place:

```
p +
  labs(
    title    = "US Gun Murders in 2010",
    subtitle = "Data from the FBI",
    caption  = "Source: dslabs package",
    x        = "Population (millions, log scale)",
    y        = "Total murders (log scale)",
    color    = "US Region"
  )
```

Result


US Gun Murders in 2010
Data from the FBI
US Region
Northeast
South
North Central
West
Total murders (log scale)
1000
100
10
Population (millions, log scale)
1
3
10
30
CA
TX
FL
NY
PA
IL
MI
GA
VA
NC
OH
NJ
MO
AZ
LA
MD
SC
TN
IN
MS
OK
AL
KY
NM
NV
AR
CT
WI
MA
WA
KS
CO
MN
OR
NE
WV
UT
IA
DC
DE
AK
RI
MT
ME
ID
SD
HI
WY
ND
NH
VT
Source: dslabs package

ggplot2 Best Practices

1. **Start simple** — get the basic geom right, then refine
2. **Use global `aes()`** for mappings shared across all layers
3. **Log-transform** when data spans orders of magnitude
4. **Use ggrepel** to prevent label overlap
5. **Choose themes** that match your publication/presentation style
6. **Separate data wrangling from plotting** — pipe clean data in
7. **Assign to objects** and build incrementally for reproducibility

Reference: https://ggplot2.tidyverse.org

Summary: Core ggplot2 Functions

Task	Function
Initialize	`ggplot(data, aes(...))`
Scatter plot	`geom_point()`
Histogram	`geom_histogram(binwidth=)`
Density	`geom_density(alpha=)`
Boxplot	`geom_boxplot()`
Bar chart	`geom_bar()` / `geom_col()`
Log axis	`scale_x_log10()`
Labels	`labs(title=, x=, y=)`
Reference line	`geom_abline()`
Themes	`theme_bw()`, `theme_economist()`

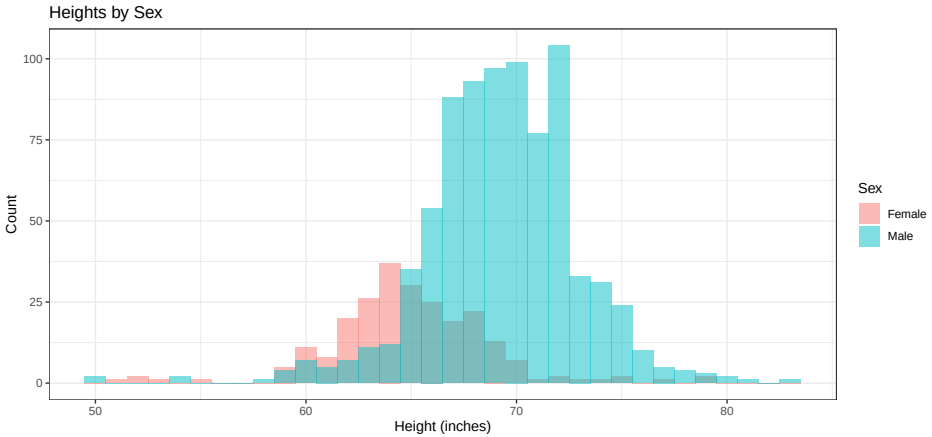
Section 3

Putting It All Together

A Complete Workflow

```
data(heights)

heights |>
  ggplot(aes(height, fill = sex)) +
  geom_histogram(binwidth = 1,
                 position = "identity",
                 alpha = 0.5) +
  labs(title = "Heights by Sex",
       x = "Height (inches)", y = "Count",
       fill = "Sex") +
  theme_bw()
```

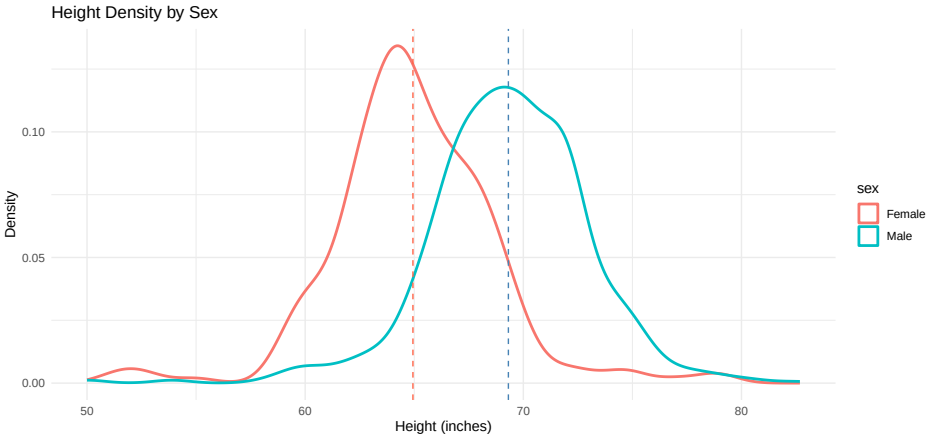
Result


Heights by Sex
Count
100
75
50
25
0
50
60
70
80
Height (inches)
Sex
Female
Male

Overlaid Density Comparison

```
heights |>
  ggplot(aes(height, color = sex)) +
  geom_density(linewidth = 1) +
  geom_vline(
    xintercept = c(mean(male), mean(female)),
    linetype = "dashed",
    color = c("steelblue", "tomato")) +
  labs(title = "Height Density by Sex",
       x = "Height (inches)", y = "Density") +
  theme_minimal()
```

Result


Height Density by Sex
Density
0.10
0.05
0.00
50
60
70
80
Height (inches)
sex
Female
Male

Key Takeaways

Distribution Visualization

- Histograms: raw shape
- Density plots: smooth comparison
- Boxplots: five-number summary
- QQ-plots: check normality
- Visualization reveals what summaries hide

ggplot2 Grammar

- Data + Geom + Aesthetics
- Layers added with +
- `aes()` for data-driven mappings
- Scales transform axes
- Themes control appearance
- Build incrementally

References and Further Reading

- Irizarry, R.A. *Introduction to Data Science*, Part 1
  - Visualizing Distributions: https://rafalab.dfci.harvard.edu/dsbook-part-1/dataviz/distributions.html
  - Introduction to ggplot2: https://rafalab.dfci.harvard.edu/dsbook-part-1/dataviz/ggplot2.html
- Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*, 3rd ed.
  - https://ggplot2-book.org
- ggplot2 Cheat Sheet: https://ggplot2.tidyverse.org
- Wilkinson, L. *The Grammar of Graphics* (2005)