Data Visualization

Concepts and Plotting Using R

Kamarul Imran Musa

2025-08-03

Table of contents

1 Concepts and Plotting Using R

1.1 Learning Objectives

By the end of this document, readers will be able to:

- Understand the fundamental concepts and principles of effective data visualization
- Apply best practices for designing clear and informative plots
- Create various types of plots using the **ggplot2** package in R
- Combine data manipulation with visualization techniques
- Develop publication-ready graphics for epidemiological and public health research

---

2 Part 1: Concepts of Data Visualization

2.1 Introduction to Data Visualization

Data visualization is the graphical representation of information and data. In epidemiology and public health, effective visualization serves multiple critical purposes: exploring data patterns, communicating findings to diverse audiences, supporting evidence-based decision making, and revealing insights that might be hidden in raw numbers.

As Edward Tufte eloquently stated in his seminal work *The Visual Display of Quantitative Information* (1983): "Excellence in statistical graphics consists of complex ideas communicated with clarity, precision and efficiency."

2.2 The Grammar of Graphics

The theoretical foundation underlying modern data visualization, particularly the **ggplot2** package, is based on Leland Wilkinson's "Grammar of Graphics" (1999). This framework treats plots as composed of distinct layers and components that can be systematically combined:

- **Data**: The dataset being visualized
- **Aesthetics**: How variables map to visual properties (position, color, size, shape)
- **Geometries**: The visual elements representing the data (points, lines, bars)
- **Statistics**: Statistical transformations applied to the data
- **Scales**: How aesthetic mappings translate to visual values
- **Coordinate systems**: The plotting space (Cartesian, polar, etc.)
- **Facets**: Subplots based on categorical variables

2.3 Principles of Effective Data Visualization

2.3.1 1. Clarity and Purpose

Every visualization should have a clear purpose and message. Before creating any plot, ask yourself:

- What story does the data tell?
- Who is the intended audience?
- What action or understanding should result from viewing this visualization?

2.3.2 2. Maximize Data-Ink Ratio

Tufte's principle emphasizes that the majority of ink should be devoted to displaying data, not decorative elements. Remove unnecessary:

- Grid lines (unless essential)
- Excessive borders
- Redundant legends
- "Chart junk" or decorative elements

2.3.3 3. Choose Appropriate Chart Types

Different data types require different visualization approaches:

- **Continuous vs. Continuous**: Scatter plots, line graphs
- **Categorical vs. Continuous**: Box plots, violin plots, bar charts
- **Time series**: Line graphs, area charts
- **Distributions**: Histograms, density plots
- **Proportions**: Pie charts (sparingly), stacked bar charts
- **Geographic data**: Maps, choropleth plots

2.3.4 4. Use Color Strategically

- Use color to encode meaningful differences
- Ensure accessibility for colorblind individuals
- Limit the number of colors (typically 3-7 for categorical data)
- Consider cultural associations with colors
- Use sequential colors for ordered data, diverging colors for data with a meaningful center

2.3.5 5. Maintain Visual Hierarchy

Guide the viewer's attention through:

- Size variations for emphasis
- Strategic use of color saturation
- Positioning of key elements
- Appropriate font weights and sizes

2.4 Examples of Good vs. Bad Visualizations

2.4.1 Characteristics of Good Visualizations

Good visualizations exhibit:

- Clear, descriptive titles and axis labels
- Appropriate scales that don't distort relationships
- Consistent formatting and styling
- Logical ordering of categorical variables
- Appropriate use of white space
- Accessibility considerations (color choices, font sizes)

Example: A well-designed epidemiological curve showing COVID-19 cases over time with clear date labels, appropriate scale, and distinct colors for different case types (confirmed, probable, deaths).

2.4.2 Common Visualization Pitfalls

Poor visualizations often include:

- Misleading scales (truncated y-axes, inappropriate aspect ratios)
- Excessive decorative elements that distract from data
- Poor color choices that don't convey meaning
- Overcrowded plots with too much information
- Missing or unclear labels and legends
- Inappropriate chart types for the data structure

Example: A pie chart with too many small slices, making it impossible to distinguish between categories, when a horizontal bar chart would be more effective.

2.5 Best Practices for Public Health Visualizations

2.5.1 1. Consider Your Audience

- **Policymakers**: Focus on key trends and actionable insights
- **General public**: Use simple, intuitive designs with clear interpretations
- **Scientific community**: Include appropriate detail and statistical precision
- **Media**: Ensure visualizations are self-explanatory and newsworthy

2.5.2 2. Handle Uncertainty Appropriately

- Display confidence intervals when relevant
- Use error bars or shaded regions for uncertainty
- Clearly indicate data quality limitations
- Consider multiple scenarios or sensitivity analyses

2.5.3 3. Respect Privacy and Ethics

- Aggregate small numbers to protect individual privacy
- Consider the potential for misinterpretation or stigmatization
- Ensure accurate representation without sensationalism

2.6 Further Reading and Resources

2.6.1 Essential Books

1. **Tufte, E. R. (2001)**. *The Visual Display of Quantitative Information* (2nd ed.). Graphics Press.
2. **Cairo, A. (2016)**. *The Truthful Art: Data, Charts, and Maps for Communication*. New Riders.
3. **Healy, K. (2018)**. *Data Visualization: A Practical Introduction*. Princeton University Press.
4. **Wilke, C. O. (2019)**. *Fundamentals of Data Visualization*. O'Reilly Media.

2.6.2 Online Resources

- **R Graphics Cookbook** - Comprehensive online resource for ggplot2
- **Data Visualization Catalogue** - Chart type selection guide
- **ColorBrewer 2.0** - Color scheme selection for maps and charts
- **From Data to Viz** - Decision tree for chart selection
- **The Functional Art** - Alberto Cairo's blog on information design
- **Storytelling with Data** - Cole Nussbaumer Knaflic's resources

2.6.3 Public Health Specific Resources

- **CDC Data Visualization Guidelines**
- **WHO Health Data Visualization Toolkit**
- **Epidemiological Curve Guidelines**

---

3 Part 2: Making Plots Using ggplot2

3.1 Setup and Data Preparation

Let's begin by loading the necessary packages and exploring our dataset.

```
# Load required packages
library(tidyverse)    # Includes ggplot2, dplyr, and other tidyverse packages
library(gapminder)    # Contains the gapminder dataset
library(patchwork)    # For combining multiple plots
library(scales)       # For better axis formatting

# Set a clean theme as default
theme_set(theme_minimal())
```

3.2 About the Gapminder Dataset

The Gapminder dataset provides longitudinal data on countries' socioeconomic indicators. This cleaned excerpt contains observations for 142 countries from 1952 to 2007, measured every 5 years.

```r
# Load and examine the gapminder dataset
glimpse(gapminder)
```

```
Rows: 1,704
Columns: 6
$ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
$ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
$ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
$ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
$ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
$ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

```r
# Display first few rows
head(gapminder)
```

```
# A tibble: 6 x 6
  country     continent  year lifeExp      pop gdpPercap
  <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
1 Afghanistan Asia       1952    28.8  8425333      779.
2 Afghanistan Asia       1957    30.3  9240934      821.
3 Afghanistan Asia       1962    32.0 10267083      853.
4 Afghanistan Asia       1967    34.0 11537966      836.
5 Afghanistan Asia       1972    36.1 13079460      740.
6 Afghanistan Asia       1977    38.4 14880372      786.
```

```r
# Summary statistics
summary(gapminder)
```

```
        country        continent        year         lifeExp
 Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
 Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
 Algeria    :  12   Asia    :396   Median :1980   Median :60.71
 Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
 Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
 Australia  :  12                  Max.   :2007   Max.   :82.60
 (Other)    :1632
      pop              gdpPercap
 Min.   :6.001e+04   Min.   :   241.2
 1st Qu.:2.794e+06   1st Qu.:  1202.1
 Median :7.024e+06   Median :  3531.8
```

```
 Mean   :2.960e+07   Mean   :  7215.3
 3rd Qu.:1.959e+07   3rd Qu.:  9325.5
 Max.   :1.319e+09   Max.   :113523.1
```

The dataset contains six variables:

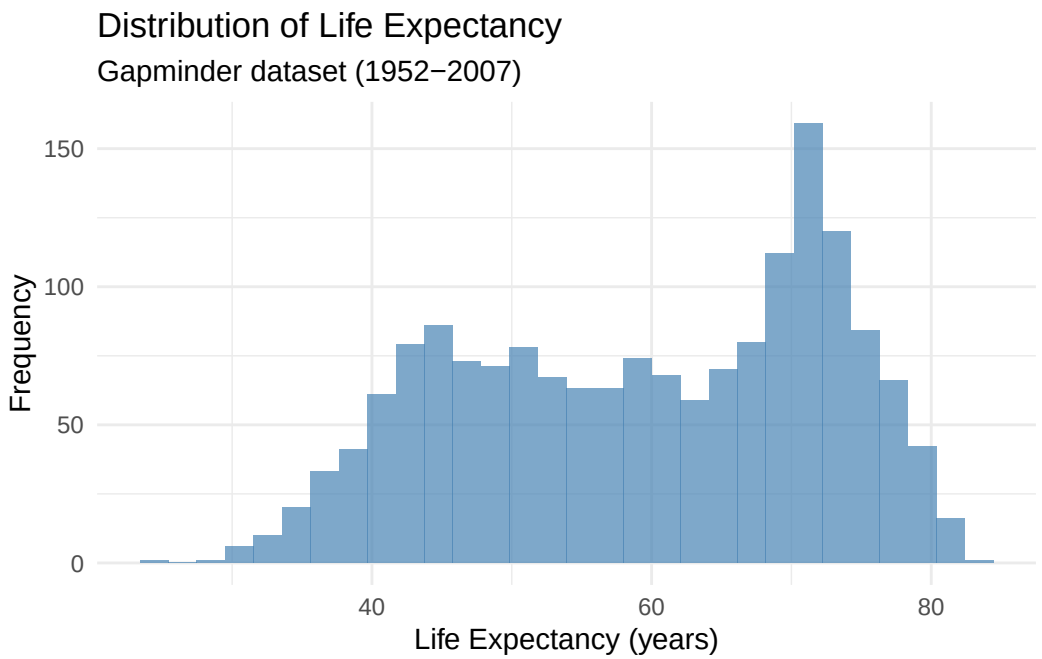
- **country**: Character variable with 142 countries
- **continent**: Factor with 5 levels (Africa, Americas, Asia, Europe, Oceania)
- **year**: Integer from 1952 to 2007 (every 5 years)
- **lifeExp**: Life expectancy at birth (years)
- **pop**: Population
- **gdpPercap**: GDP per capita (inflation-adjusted US dollars)

3.3 Building Plots Layer by Layer

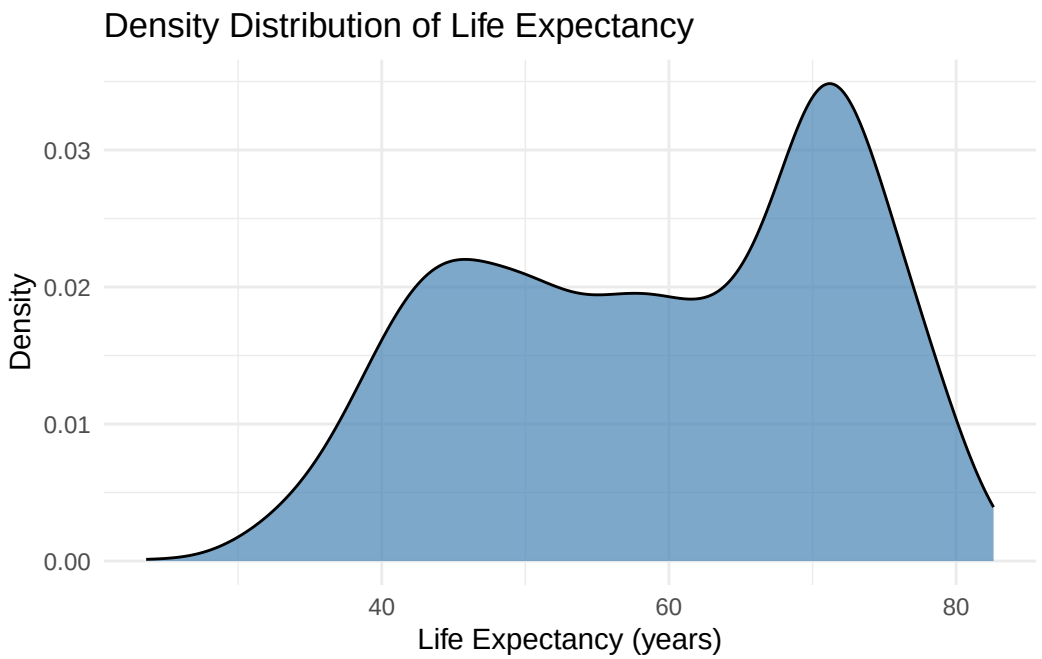
3.3.1 Single Aesthetic Mapping

Let's start with the simplest case - plotting one variable.

```
# Basic histogram of life expectancy
gapminder |>
  ggplot(aes(x = lifeExp)) +
  geom_histogram(bins = 30, fill = "steelblue", alpha = 0.7) +
  labs(
    title = "Distribution of Life Expectancy",
    subtitle = "Gapminder dataset (1952-2007)",
    x = "Life Expectancy (years)",
    y = "Frequency"
  )
```



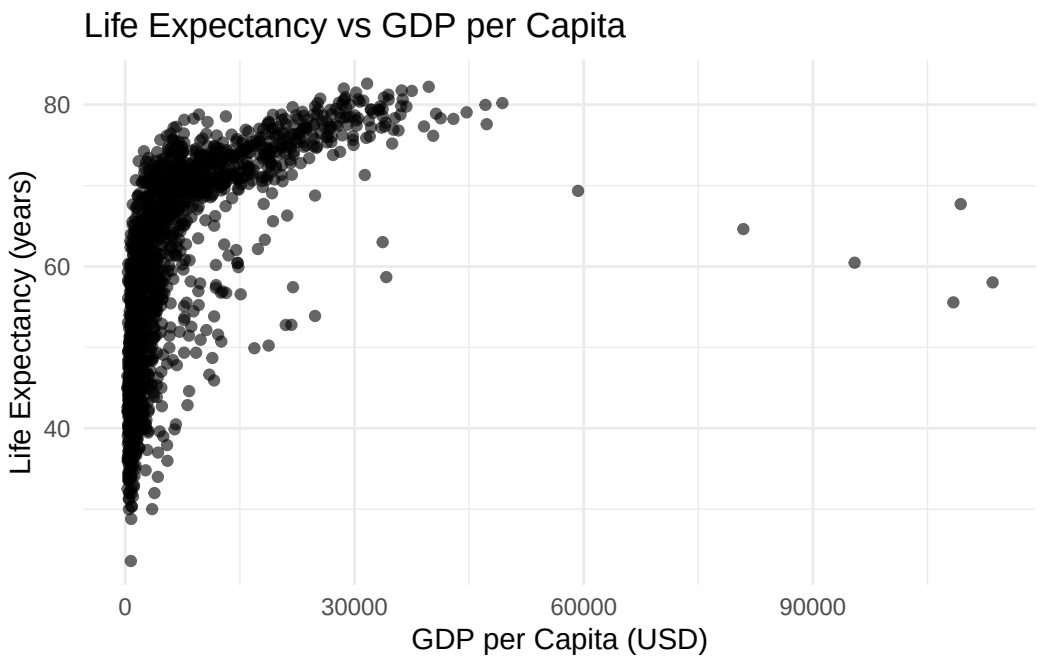

```
# Density plot for smoother distribution
gapminder |>
  ggplot(aes(x = lifeExp)) +
  geom_density(fill = "steelblue", alpha = 0.7) +
  labs(
    title = "Density Distribution of Life Expectancy",
    x = "Life Expectancy (years)",
    y = "Density"
  )
```



3.3.2 Two Aesthetic Mappings

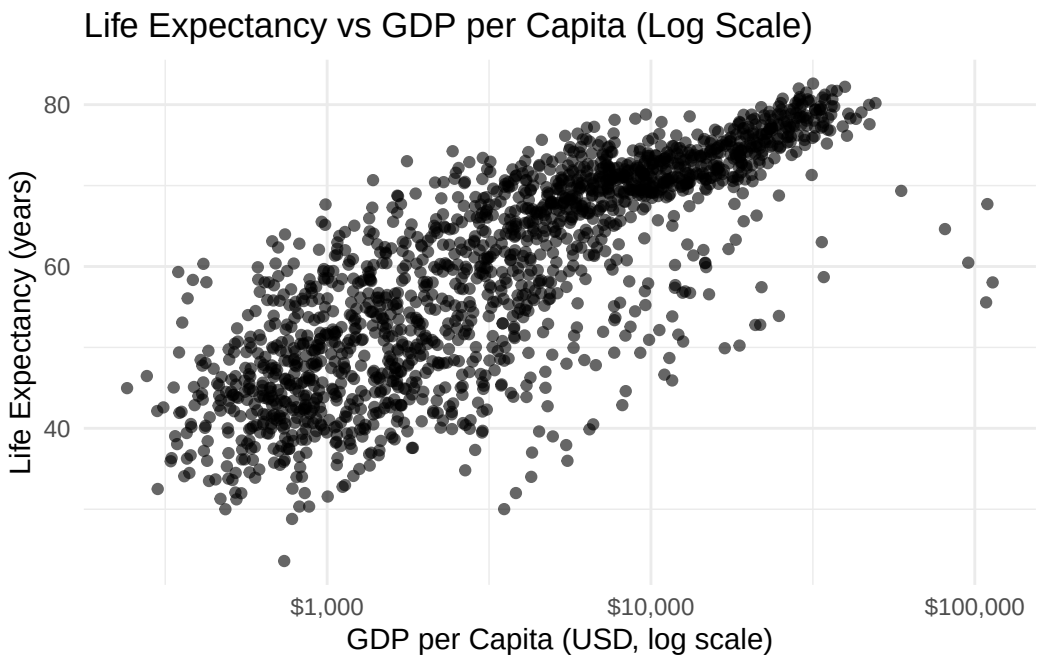
Now let's explore relationships between two variables.

```
# Basic scatter plot: GDP per capita vs Life Expectancy
gapminder |>
  ggplot(aes(x = gdpPercap, y = lifeExp)) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Life Expectancy vs GDP per Capita",
    x = "GDP per Capita (USD)",
    y = "Life Expectancy (years)"
  )
```



The relationship is clearer with a log transformation:

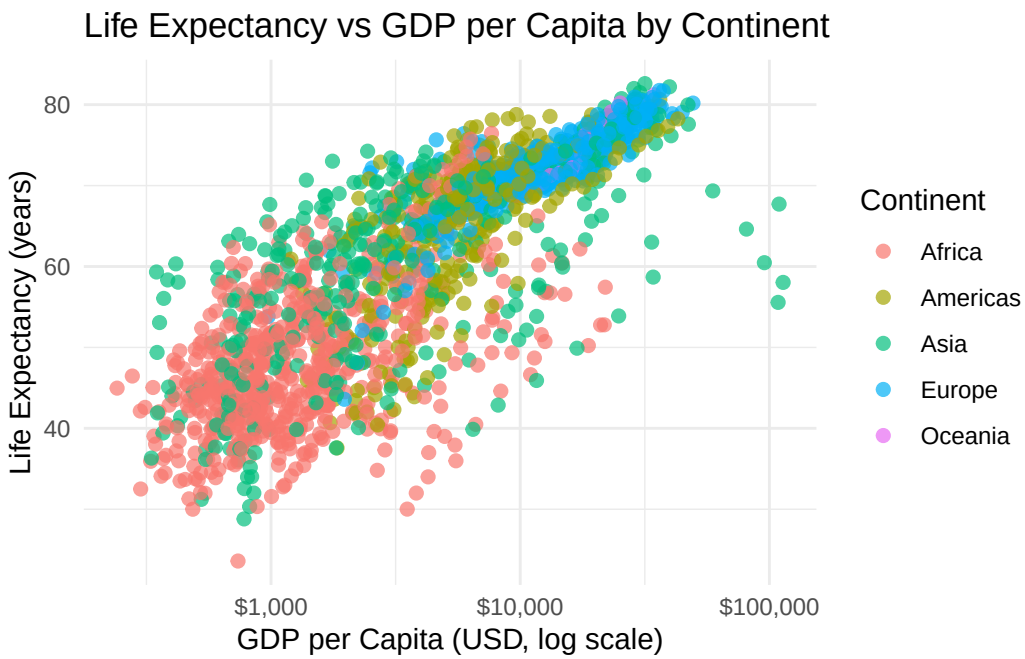
```
# Improved scatter plot with log scale
gapminder |>
  ggplot(aes(x = gdpPercap, y = lifeExp)) +
  geom_point(alpha = 0.6) +
  scale_x_log10(labels = scales::label_dollar()) +
  labs(
    title = "Life Expectancy vs GDP per Capita (Log Scale)",
    x = "GDP per Capita (USD, log scale)",
    y = "Life Expectancy (years)"
  )
```



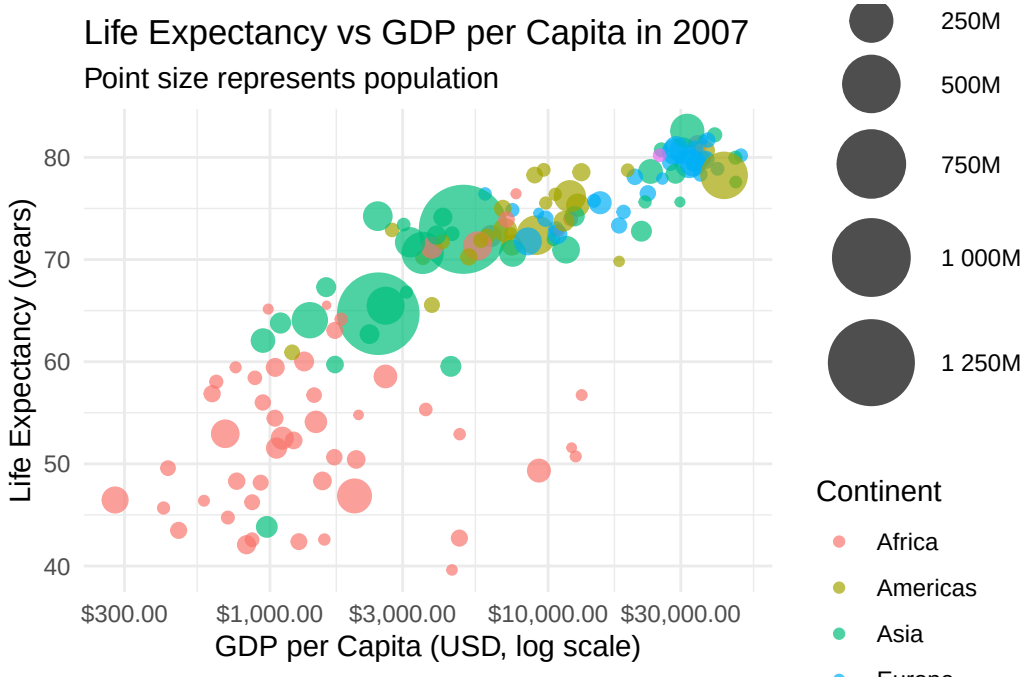
3.3.3 Adding Third Variables Through Aesthetics

We can encode additional variables using color, size, or shape:

```
# Add continent as color
gapminder |>
  ggplot(aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(alpha = 0.7, size = 2) +
  scale_x_log10(labels = scales::label_dollar()) +
  labs(
    title = "Life Expectancy vs GDP per Capita by Continent",
    x = "GDP per Capita (USD, log scale)",
    y = "Life Expectancy (years)",
    color = "Continent"
  )
```

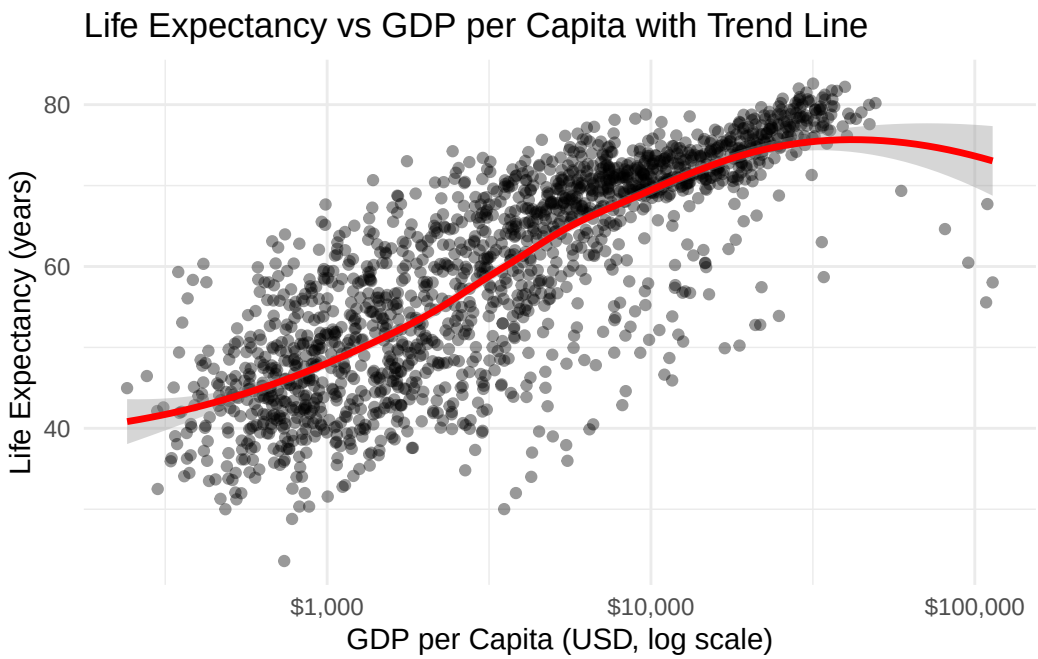



```
# Add population as size
gapminder |>
  filter(year == 2007) |>  # Focus on most recent year
  ggplot(aes(x = gdpPercap, y = lifeExp, size = pop, color = continent)) +
  geom_point(alpha = 0.7) +
  scale_x_log10(labels = scales::label_dollar()) +
  scale_size_continuous(
    name = "Population",
    labels = scales::label_number(scale = 1e-6, suffix = "M"),
    range = c(1, 15)
  ) +
  labs(
    title = "Life Expectancy vs GDP per Capita in 2007",
    subtitle = "Point size represents population",
    x = "GDP per Capita (USD, log scale)",
    y = "Life Expectancy (years)",
    color = "Continent"
  )
```

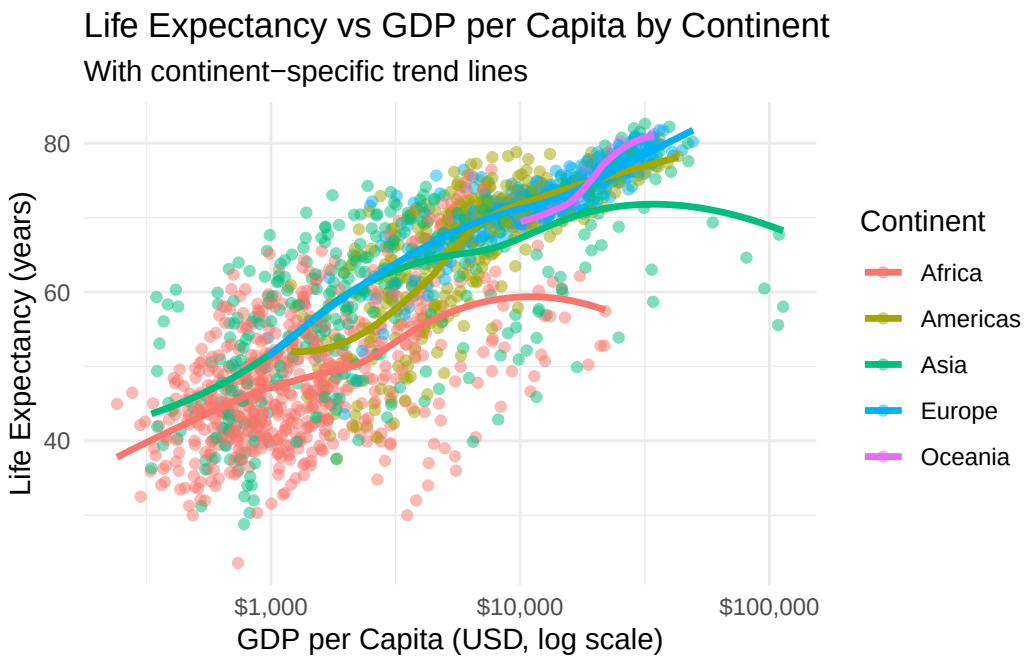


3.3.4 Adding Trend Lines

```
# Add smoothed trend lines
gapminder |>
  ggplot(aes(x = gdpPercap, y = lifeExp)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = TRUE, color = "red", linewidth = 1.2) +
  scale_x_log10(labels = scales::label_dollar()) +
  labs(
    title = "Life Expectancy vs GDP per Capita with Trend Line",
    x = "GDP per Capita (USD, log scale)",
    y = "Life Expectancy (years)"
  )
```




```
# Separate trend lines by continent
gapminder |>
  ggplot(aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE, linewidth = 1.2) +
  scale_x_log10(labels = scales::label_dollar()) +
  labs(
    title = "Life Expectancy vs GDP per Capita by Continent",
    subtitle = "With continent-specific trend lines",
    x = "GDP per Capita (USD, log scale)",
    y = "Life Expectancy (years)",
    color = "Continent"
  )
```

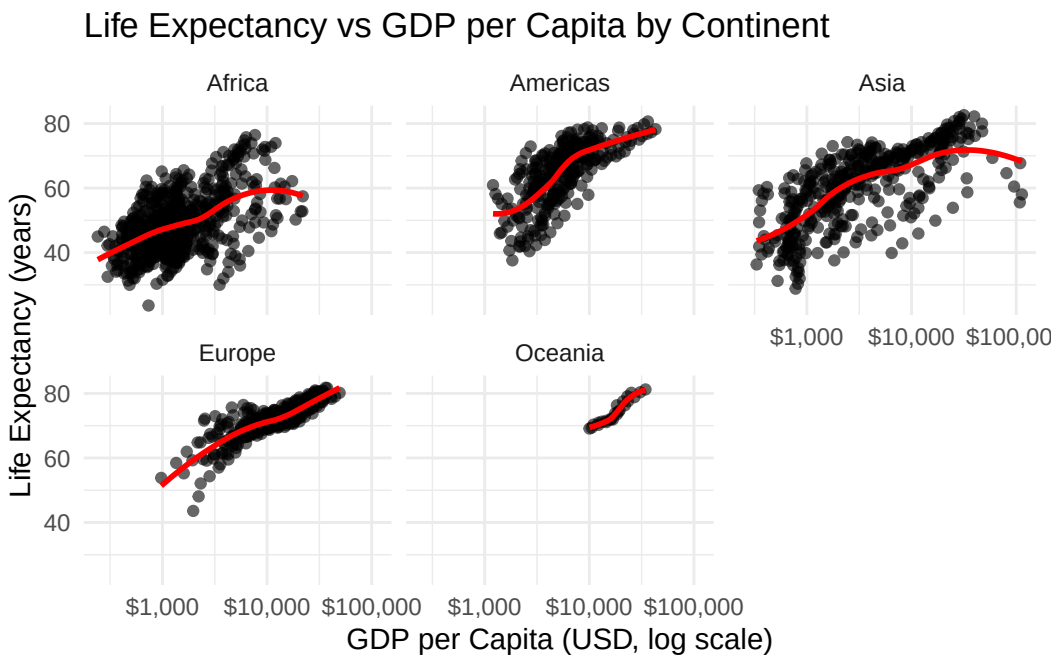


3.4 Using Faceting to Split Plots

Faceting allows us to create multiple subplots based on categorical variables.

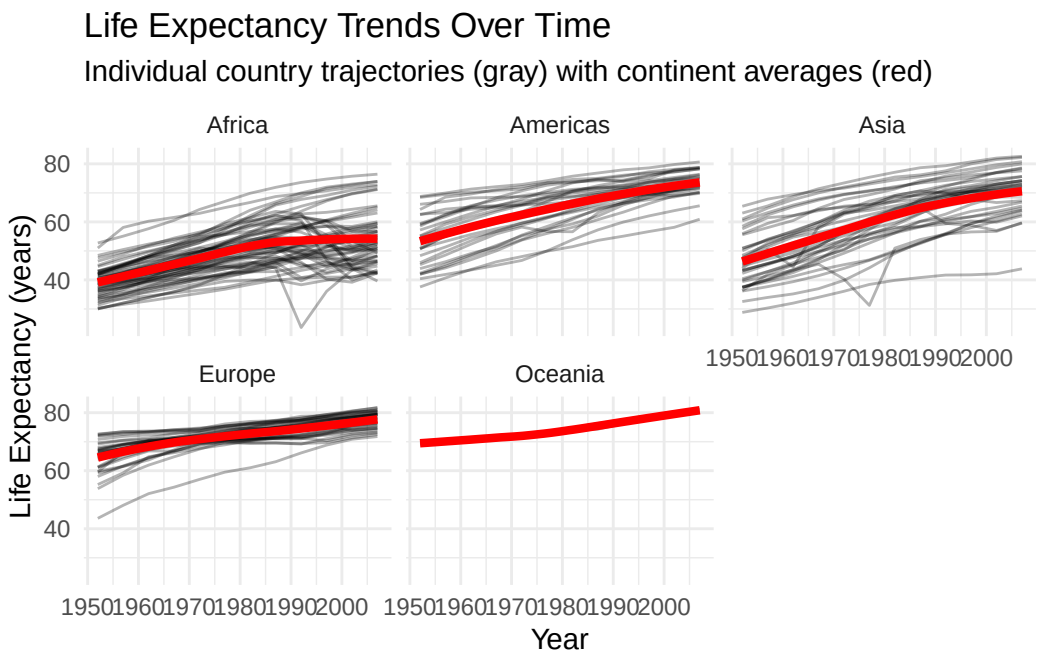
3.4.1 Facet Wrap

```
# Facet by continent
gapminder |>
  ggplot(aes(x = gdpPercap, y = lifeExp)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  scale_x_log10(labels = scales::label_dollar()) +
  facet_wrap(~continent, nrow = 2) +
  labs(
    title = "Life Expectancy vs GDP per Capita by Continent",
    x = "GDP per Capita (USD, log scale)",
    y = "Life Expectancy (years)"
  )
```



3.4.2 Time Series with Faceting

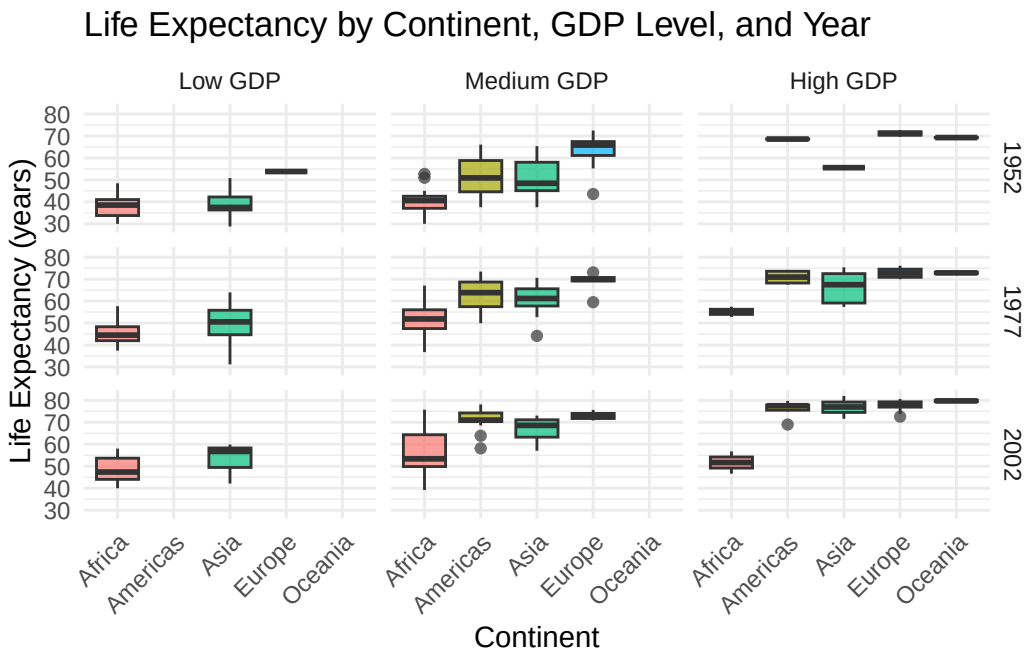
```
# Life expectancy trends over time by continent
gapminder |>
  ggplot(aes(x = year, y = lifeExp)) +
  geom_line(aes(group = country), alpha = 0.3) +
  geom_smooth(method = "loess", se = TRUE, color = "red", linewidth = 1.5) +
  facet_wrap(~continent, nrow = 2) +
  labs(
    title = "Life Expectancy Trends Over Time",
    subtitle = "Individual country trajectories (gray) with continent averages (red)",
    x = "Year",
    y = "Life Expectancy (years)"
  )
```



3.4.3 Facet Grid for Two Variables

```
# Create categorical variables for faceting
gapminder_cat <- gapminder |>
  filter(year %in% c(1952, 1977, 2002)) |>
  mutate(
    gdp_level = case_when(
      gdpPercap < 1000 ~ "Low GDP",
      gdpPercap < 10000 ~ "Medium GDP",
      TRUE ~ "High GDP"
    ),
    gdp_level = factor(gdp_level, levels = c("Low GDP", "Medium GDP", "High GDP"))
  )

gapminder_cat |>
  ggplot(aes(x = continent, y = lifeExp)) +
  geom_boxplot(aes(fill = continent), alpha = 0.7) +
  facet_grid(year ~ gdp_level) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Life Expectancy by Continent, GDP Level, and Year",
```

```
    x = "Continent",
    y = "Life Expectancy (years)",
    fill = "Continent"
  ) +
  guides(fill = "none")  # Remove redundant legend
```

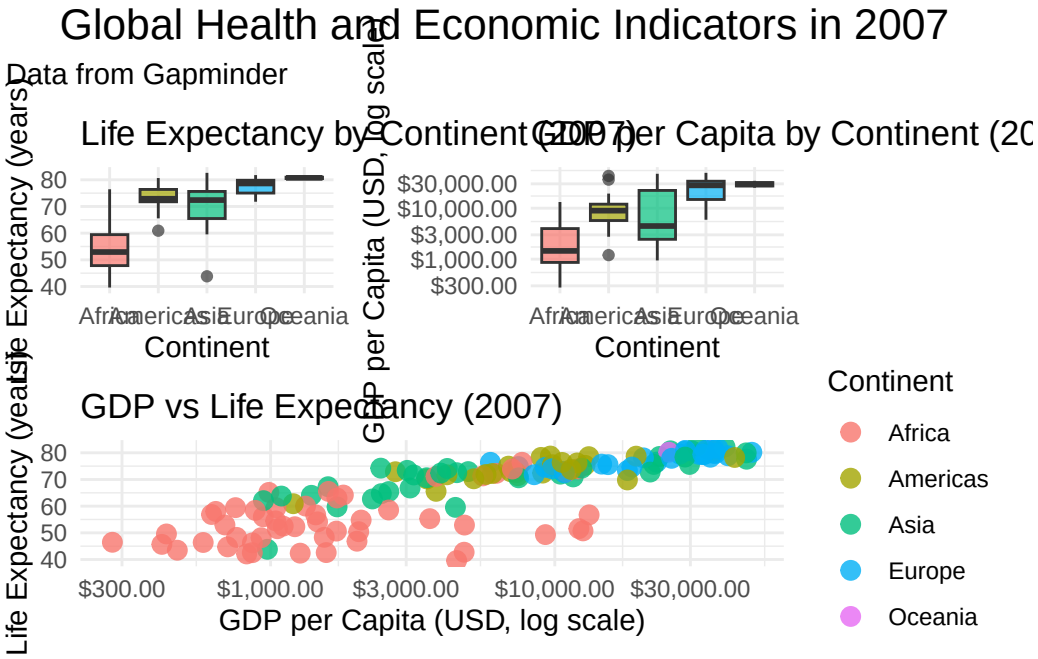


3.5 Combining Multiple Plots

Using the **patchwork** package to combine different visualizations:

```
# Create individual plots
p1 <- gapminder |>
  filter(year == 2007) |>
  ggplot(aes(x = continent, y = lifeExp, fill = continent)) +
  geom_boxplot(alpha = 0.7) +
  labs(title = "Life Expectancy by Continent (2007)",
       x = "Continent", y = "Life Expectancy (years)") +
  theme(legend.position = "none")

p2 <- gapminder |>
  filter(year == 2007) |>
```

```r
  ggplot(aes(x = continent, y = gdpPercap, fill = continent)) +
  geom_boxplot(alpha = 0.7) +
  scale_y_log10(labels = scales::label_dollar()) +
  labs(title = "GDP per Capita by Continent (2007)",
       x = "Continent", y = "GDP per Capita (USD, log scale)") +
  theme(legend.position = "none")

p3 <- gapminder |>
  filter(year == 2007) |>
  ggplot(aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_x_log10(labels = scales::label_dollar()) +
  labs(title = "GDP vs Life Expectancy (2007)",
       x = "GDP per Capita (USD, log scale)",
       y = "Life Expectancy (years)",
       color = "Continent")

# Combine plots using patchwork
(p1 + p2) / p3 +
  plot_annotation(
    title = "Global Health and Economic Indicators in 2007",
    subtitle = "Data from Gapminder",
    theme = theme(plot.title = element_text(size = 16, hjust = 0.5))
  )
```

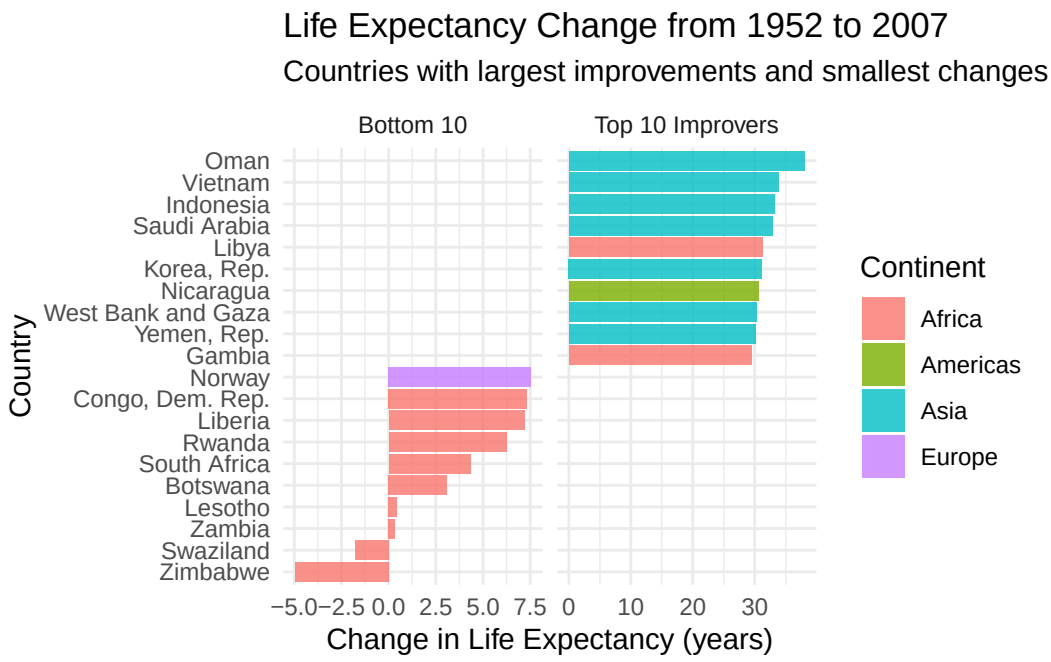


3.6 Data Wrangling with dplyr and Visualization

3.6.1 Filtering and Summarizing for Specific Analyses

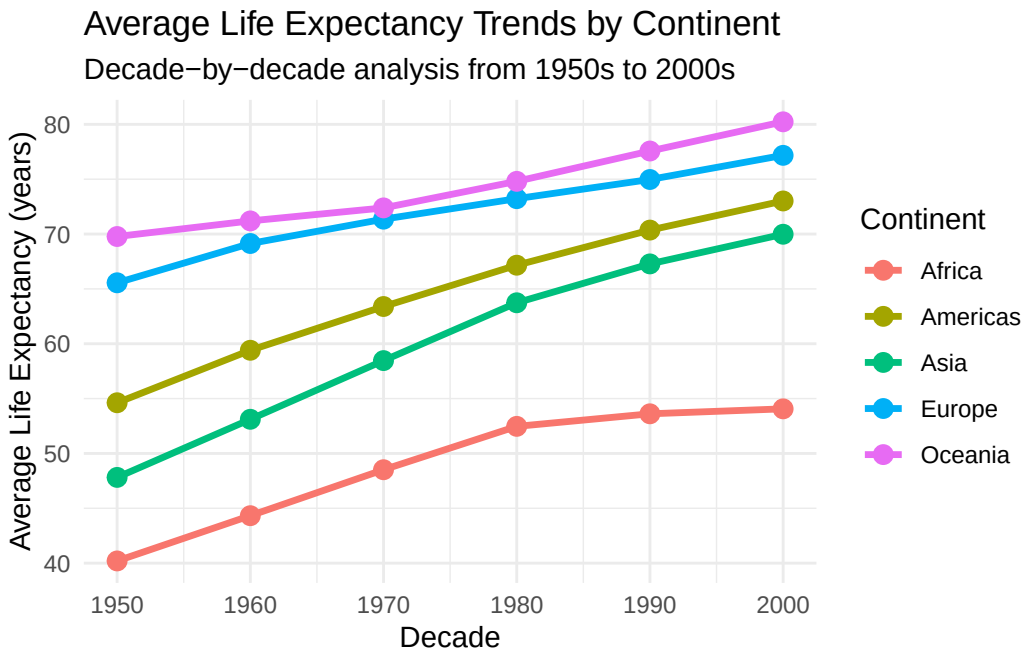
```
# Analyze top and bottom performers in life expectancy improvement
life_exp_change <- gapminder |>
  filter(year %in% c(1952, 2007)) |>
  select(country, continent, year, lifeExp) |>
  pivot_wider(names_from = year, values_from = lifeExp) |>
  mutate(
    life_exp_change = `2007` - `1952`,
    life_exp_1952 = `1952`,
    life_exp_2007 = `2007`
  ) |>
  arrange(desc(life_exp_change))

# Top 10 improvers
top_improvers <- life_exp_change |>
  slice_head(n = 10)

# Bottom 10 (least improvement or decline)
```

```r
bottom_improvers <- life_exp_change |>
  slice_tail(n = 10)

# Visualize the changes
bind_rows(
  top_improvers |> mutate(group = "Top 10 Improvers"),
  bottom_improvers |> mutate(group = "Bottom 10")
) |>
  mutate(country = fct_reorder(country, life_exp_change)) |>
  ggplot(aes(x = country, y = life_exp_change, fill = continent)) +
  geom_col(alpha = 0.8) +
  facet_wrap(~group, scales = "free_x") +
  coord_flip() +
  labs(
    title = "Life Expectancy Change from 1952 to 2007",
    subtitle = "Countries with largest improvements and smallest changes",
    x = "Country",
    y = "Change in Life Expectancy (years)",
    fill = "Continent"
  )
```



3.6.2 Regional Analysis with Grouped Operations

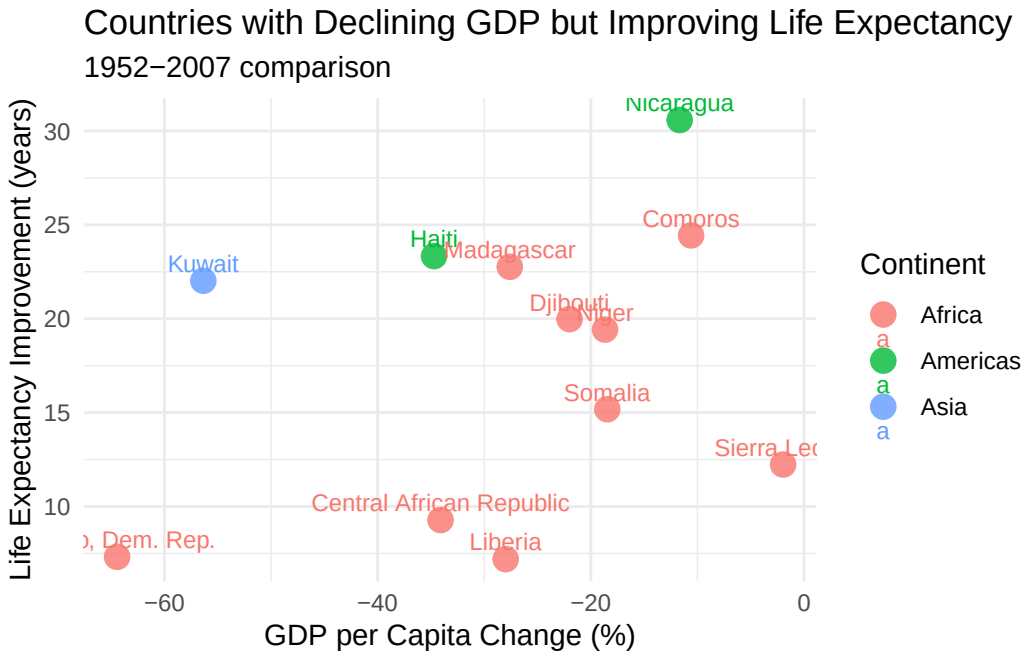
```
# Calculate regional statistics by decade
regional_trends <- gapminder |>
  mutate(decade = (year %/% 10) * 10) |>
  group_by(continent, decade) |>
  summarise(
    avg_life_exp = mean(lifeExp),
    avg_gdp = mean(gdpPercap),
    total_pop = sum(pop),
    n_countries = n_distinct(country),
    .groups = "drop"
  )

# Visualize regional trends
regional_trends |>
  ggplot(aes(x = decade, y = avg_life_exp, color = continent)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  scale_x_continuous(breaks = seq(1950, 2000, 10)) +
  labs(
    title = "Average Life Expectancy Trends by Continent",
    subtitle = "Decade-by-decade analysis from 1950s to 2000s",
    x = "Decade",
    y = "Average Life Expectancy (years)",
    color = "Continent"
  )
```



3.6.3 Advanced Filtering and Custom Calculations

```
# Identify countries that experienced GDP decline while life expectancy improved
paradox_countries <- gapminder |>
  filter(year %in% c(1952, 2007)) |>
  select(country, continent, year, lifeExp, gdpPercap) |>
  pivot_wider(
    names_from = year,
    values_from = c(lifeExp, gdpPercap),
    names_sep = "_"
  ) |>
  mutate(
    life_exp_change = lifeExp_2007 - lifeExp_1952,
    gdp_change = gdpPercap_2007 - gdpPercap_1952,
    gdp_change_pct = (gdpPercap_2007 - gdpPercap_1952) / gdpPercap_1952 * 100
  ) |>
  filter(life_exp_change > 0 & gdp_change < 0) |>
  arrange(desc(life_exp_change))

# Visualize these paradoxical cases
if(nrow(paradox_countries) > 0) {
```

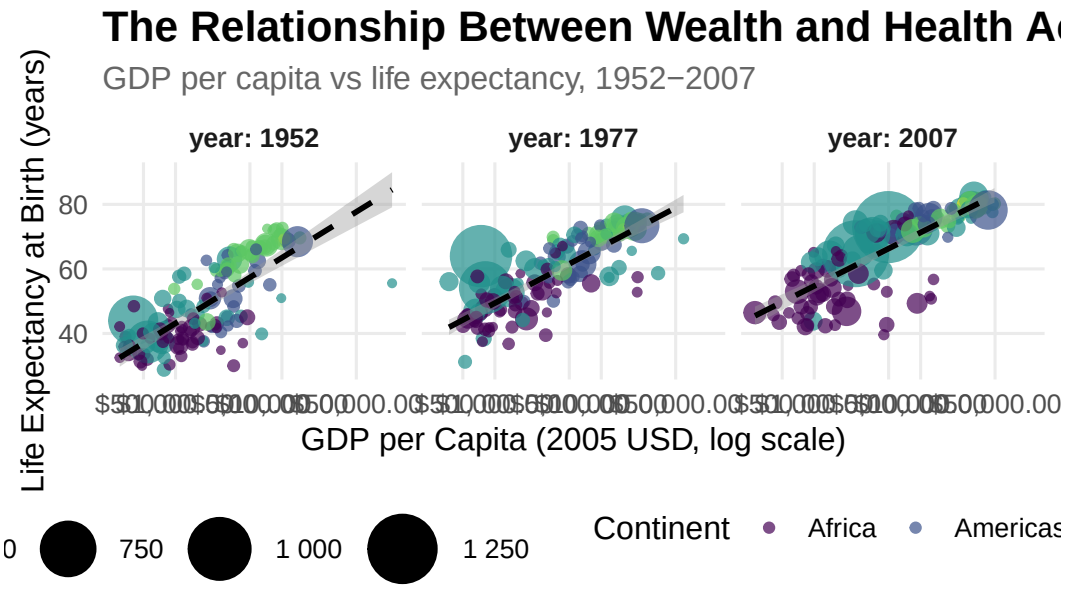
```
  paradox_countries |>
    ggplot(aes(x = gdp_change_pct, y = life_exp_change, color = continent)) +
    geom_point(size = 4, alpha = 0.8) +
    geom_text(aes(label = country), vjust = -0.5, size = 3) +
    labs(
      title = "Countries with Declining GDP but Improving Life Expectancy",
      subtitle = "1952-2007 comparison",
      x = "GDP per Capita Change (%)",
      y = "Life Expectancy Improvement (years)",
      color = "Continent"
    )
} else {
  print("No countries showed declining GDP with improving life expectancy")
}
```



3.7 Creating Publication-Ready Plots

```
# Create a comprehensive, publication-ready visualization
final_plot <- gapminder |>
  filter(year %in% c(1952, 1977, 2007)) |>
```

```r
  ggplot(aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes(size = pop, color = continent), alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  scale_x_log10(
    labels = scales::label_dollar(),
    breaks = c(500, 1000, 5000, 10000, 50000)
  ) +
  scale_size_continuous(
    name = "Population\n(millions)",
    labels = scales::label_number(scale = 1e-6, accuracy = 1),
    range = c(1, 12),
    guide = guide_legend(override.aes = list(alpha = 1))
  ) +
  scale_color_viridis_d(name = "Continent") +
  facet_wrap(~year, labeller = label_both) +
  labs(
    title = "The Relationship Between Wealth and Health Across Time",
    subtitle = "GDP per capita vs life expectancy, 1952-2007",
    caption = "Data source: Gapminder Foundation | Point size represents population",
    x = "GDP per Capita (2005 USD, log scale)",
    y = "Life Expectancy at Birth (years)"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray40"),
    plot.caption = element_text(size = 10, color = "gray50"),
    legend.position = "bottom",
    strip.text = element_text(face = "bold"),
    panel.grid.minor = element_blank()
  )

print(final_plot)
```



4 Summary

This document has covered both the theoretical foundations and practical implementation of data visualization for epidemiological and public health research. Key takeaways include:

1. **Conceptual Understanding**: Effective visualization requires clear purpose, appropriate chart selection, and attention to design principles.
2. **Technical Skills**: The **ggplot2** package provides a powerful framework for creating layered, publication-quality graphics.
3. **Integration with Data Workflow**: Combining **dplyr** data manipulation with **ggplot2** visualization enables sophisticated analytical workflows.
4. **Best Practices**: Always consider your audience, ensure accessibility, and maintain scientific integrity in your visualizations.

The combination of solid theoretical understanding and practical **R** skills will enable you to create compelling, accurate, and actionable visualizations that support evidence-based decision-making in public health practice and research.

---

*For additional practice and examples, explore the extensive **ggplot2** documentation at https://ggplot2.tidyverse.org/ and the **R Graphics Cookbook** at https://r-graphics.org/.*