Theming with bslib and thematic

Code

- Show All Code
- Hide All Code

Theming with bslib and thematic

Here's a comprehensive tutorial on how to use `dplyr` and `ggplot2` to explore the relationship between GDP per capita (GDPPC) and average life expectancy, along with a third categorical variable like region. We'll also go through creating histograms, bar plots, box plots, and scatter plots with custom aesthetics, including separated bars with dashes and a clean, minimalistic design.

We'll use the `WDI` package to load the data, `dplyr` for data manipulation, and `ggplot2` for plotting.

Step 1: Load Necessary Packages

First, install and load the necessary packages:

```
# Install required packages if you don't have them yet
install.packages(c("WDI", "dplyr", "ggplot2"))
```

```
## Installing packages into 'C:/Users/Ignacio/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## also installing the dependency 'scales'

## package 'scales' successfully unpacked and MD5 sums checked
## package 'WDI' successfully unpacked and MD5 sums checked
## package 'dplyr' successfully unpacked and MD5 sums checked
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Ignacio\AppData\Local\Temp\RtmpCSL80W\downloaded_packages
```

```
# Load the libraries
library(WDI)
```

```
## Warning: package 'WDI' was built under R version 4.3.3
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

Step 2: Load World Bank Data for GDP per Capita and Life Expectancy

We can use the `WDI` package to pull data on GDP per capita and life expectancy across countries. Additionally, we'll include region information from another source.

```r
# Load the WDI data for GDP per capita (NY.GDP.PCAP.CD) and Life Expectancy (SP.DYN.LE00.IN
data <- WDI(indicator = c("NY.GDP.PCAP.CD", "SP.DYN.LE00.IN"),
            start = 2020, end = 2020, extra = TRUE)

# Rename columns for convenience
data <- data %>%
  rename(GDPPC = NY.GDP.PCAP.CD, LifeExpectancy = SP.DYN.LE00.IN)

# Keep only relevant columns
data <- data %>%
  select(country, region, GDPPC, LifeExpectancy) %>%
  filter(!is.na(GDPPC), !is.na(LifeExpectancy), !is.na(region))

# Check the structure of the data
head(data)
```

```
##                       country                     region     GDPPC
## 1                 Afghanistan                 South Asia  512.0551
## 2 Africa Eastern and Southern                 Aggregates 1356.0889
## 3  Africa Western and Central                 Aggregates 1688.4709
## 4                     Albania      Europe & Central Asia 5343.0377
## 5                     Algeria Middle East & North Africa 3794.4095
```

```
## 6                       Angola        Sub-Saharan Africa 1450.9051
##   LifeExpectancy
## 1       62.57500
## 2       63.31386
## 3       57.22637
## 4       76.98900
## 5       74.45300
## 6       62.26100
```

Step 3: Data Exploration Using dplyr

We can start by doing some basic data exploration, like summarizing GDP per capita and life expectancy across different regions.

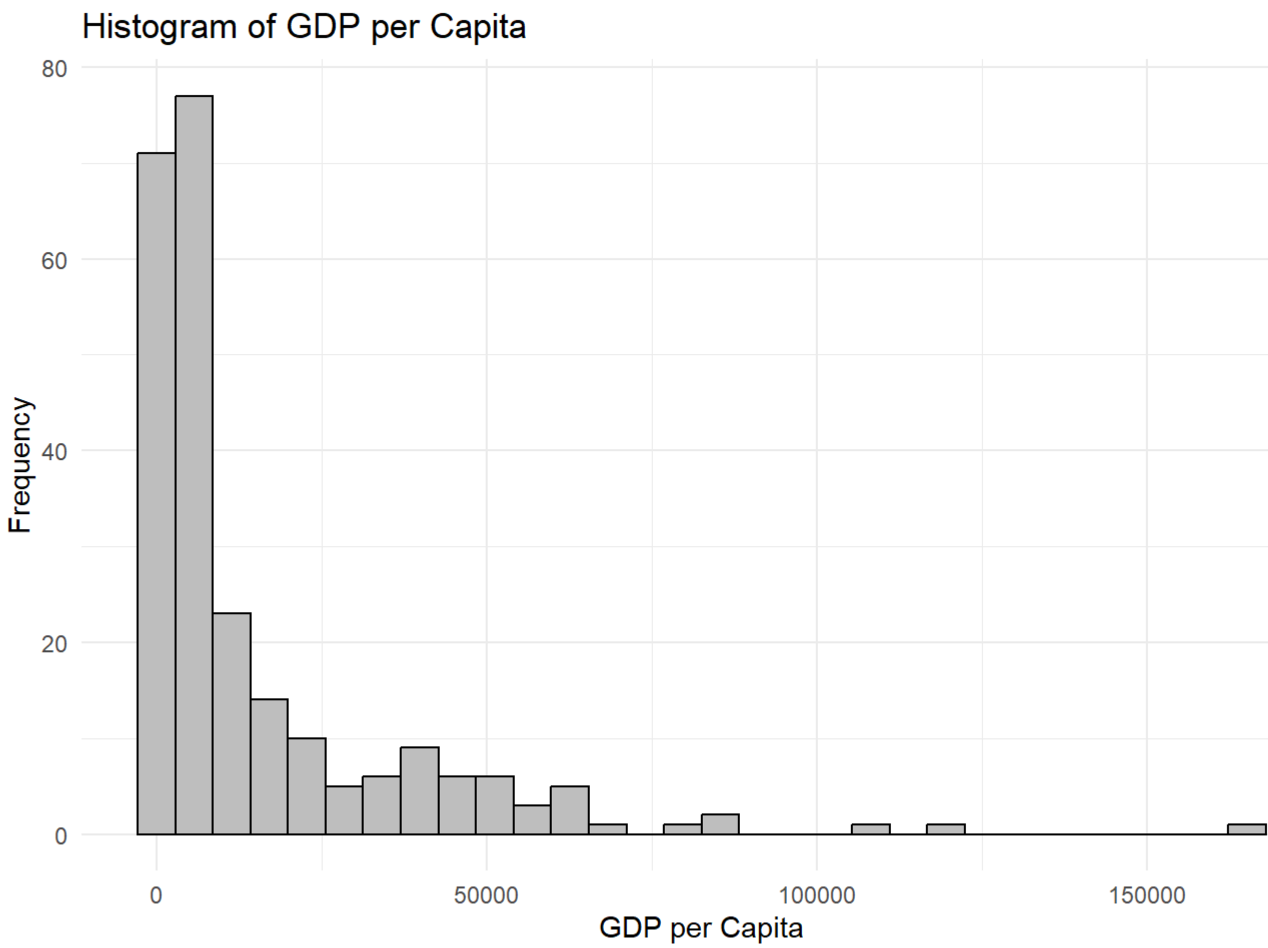
```r
# Summarize GDPPC and LifeExpectancy by region
summary_by_region <- data %>%
  group_by(region) %>%
  summarize(Avg_GDPPC = mean(GDPPC, na.rm = TRUE),
            Avg_LifeExpectancy = mean(LifeExpectancy, na.rm = TRUE))

# View summary statistics
print(summary_by_region)
```

```
## # A tibble: 8 × 3
##   region                     Avg_GDPPC Avg_LifeExpectancy
##   <chr>                          <dbl>              <dbl>
## 1 Aggregates                    10357.               70.8
## 2 East Asia & Pacific           16225.               74.1
## 3 Europe & Central Asia         31625.               77.3
## 4 Latin America & Caribbean     11362.               73.8
## 5 Middle East & North Africa    14166.               74.7
## 6 North America                 71882.               79.9
## 7 South Asia                     2671.               71.0
## 8 Sub-Saharan Africa             2136.               62.6
```

Step 4: Plot Histograms

We'll plot histograms to visualize the distribution of GDP per capita and life expectancy across countries.

```r
# GDPPC Histogram
ggplot(data, aes(x = GDPPC)) +
  geom_histogram(color = "black", fill = "grey", bins = 30) +
  labs(title = "Histogram of GDP per Capita", x = "GDP per Capita", y = "Frequency") +
  theme_minimal()
```

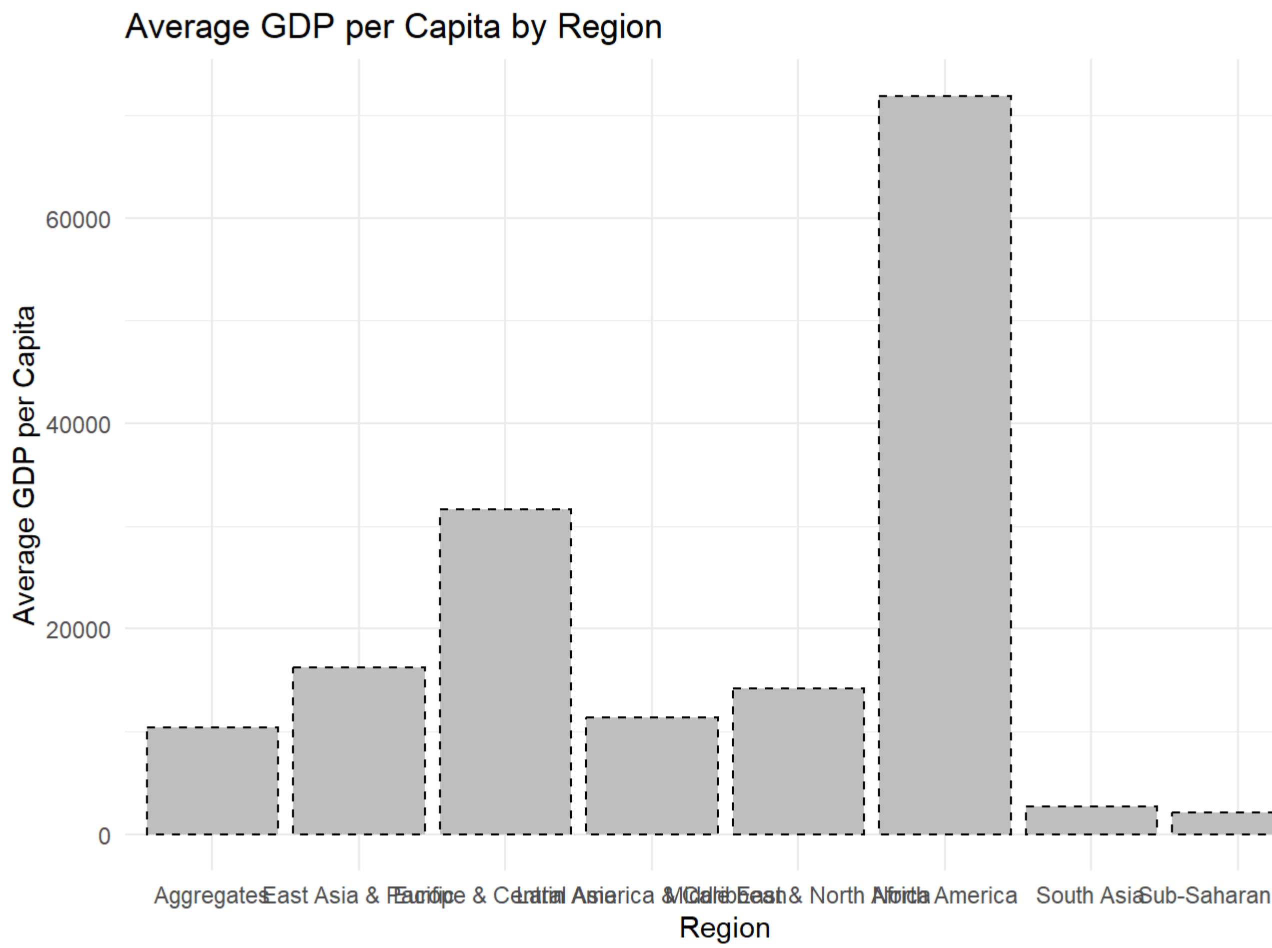



```r
# Life Expectancy Histogram
ggplot(data, aes(x = LifeExpectancy)) +
  geom_histogram(color = "black", fill = "grey", bins = 30) +
  labs(title = "Histogram of Life Expectancy", x = "Life Expectancy", y = "Frequency") +
  theme_minimal()
```

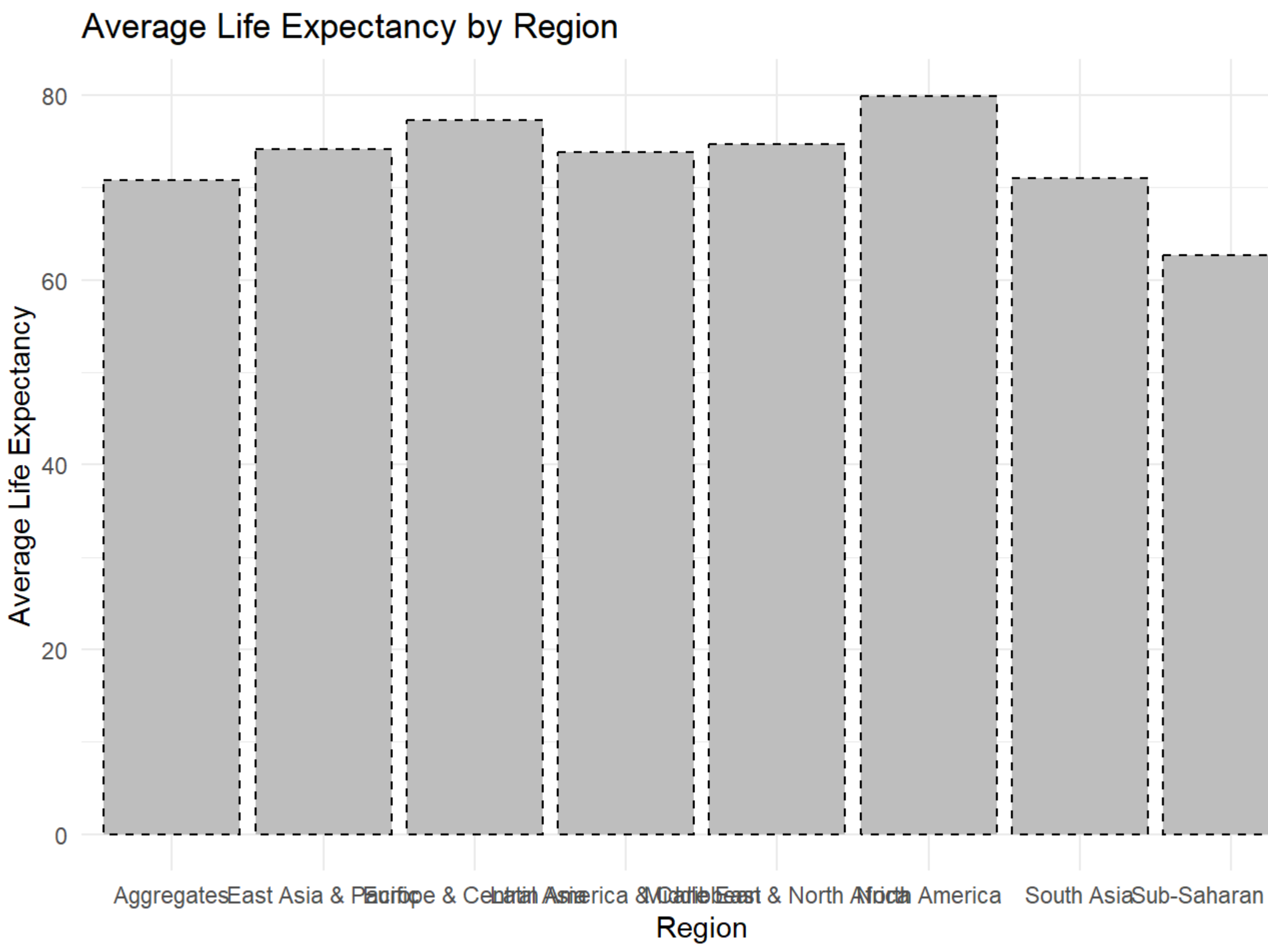
Step 5: Plot Bar Plots

We'll use bar plots to show the average GDP per capita and life expectancy for each region. We'll also customize the bars to have dashes between them and use a grey color.

```
# Bar plot for GDPPC by region
ggplot(summary_by_region, aes(x = region, y = Avg_GDPPC)) +
  geom_bar(stat = "identity", fill = "grey", color = "black", linetype = "dashed") +
  labs(title = "Average GDP per Capita by Region", x = "Region", y = "Average GDP per Capita
  theme_minimal()
```

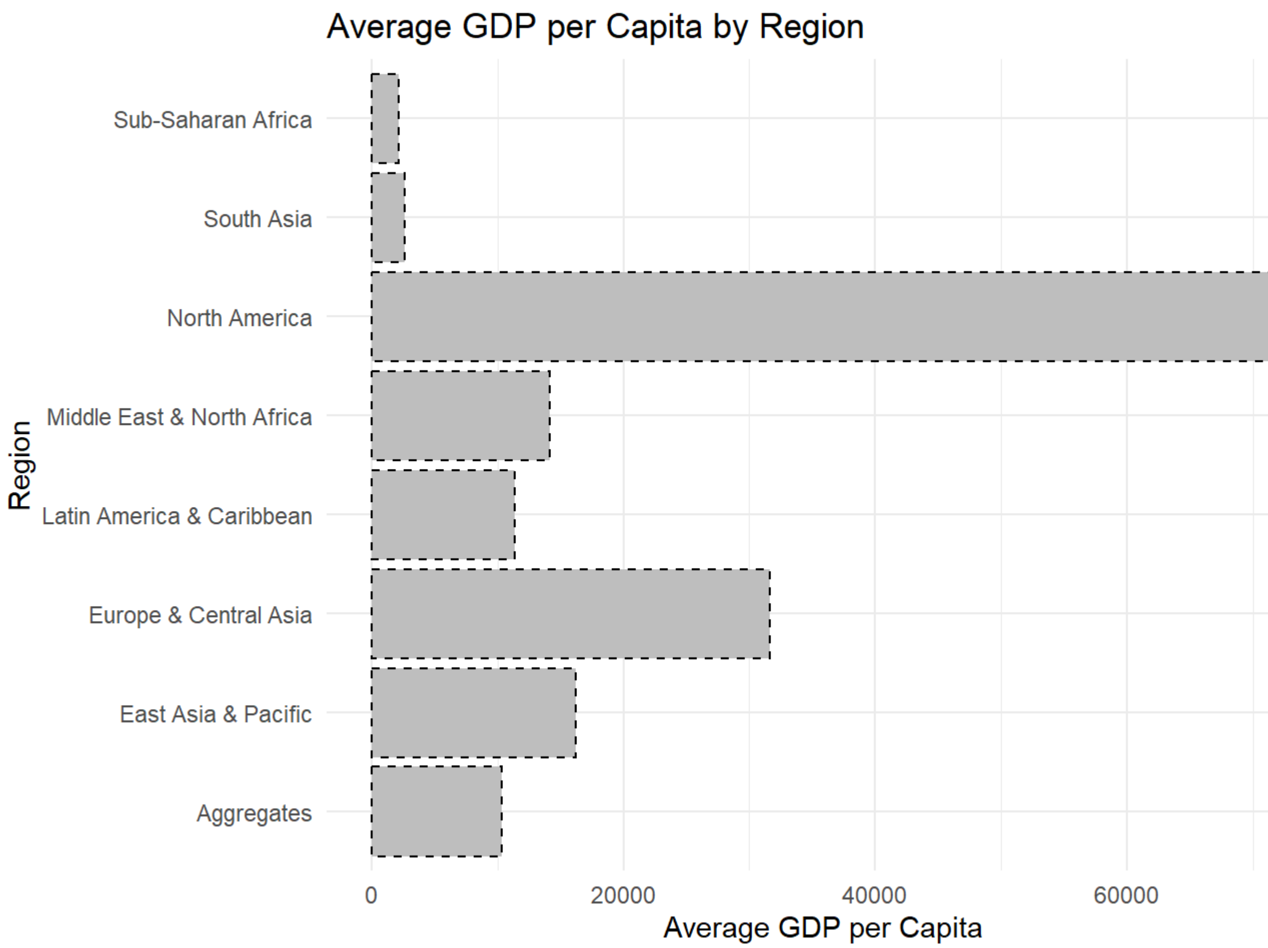



```
# Bar plot for Life Expectancy by region
ggplot(summary_by_region, aes(x = region, y = Avg_LifeExpectancy)) +
  geom_bar(stat = "identity", fill = "grey", color = "black", linetype = "dashed") +
  labs(title = "Average Life Expectancy by Region", x = "Region", y = "Average Life Expectan
  theme_minimal()
```

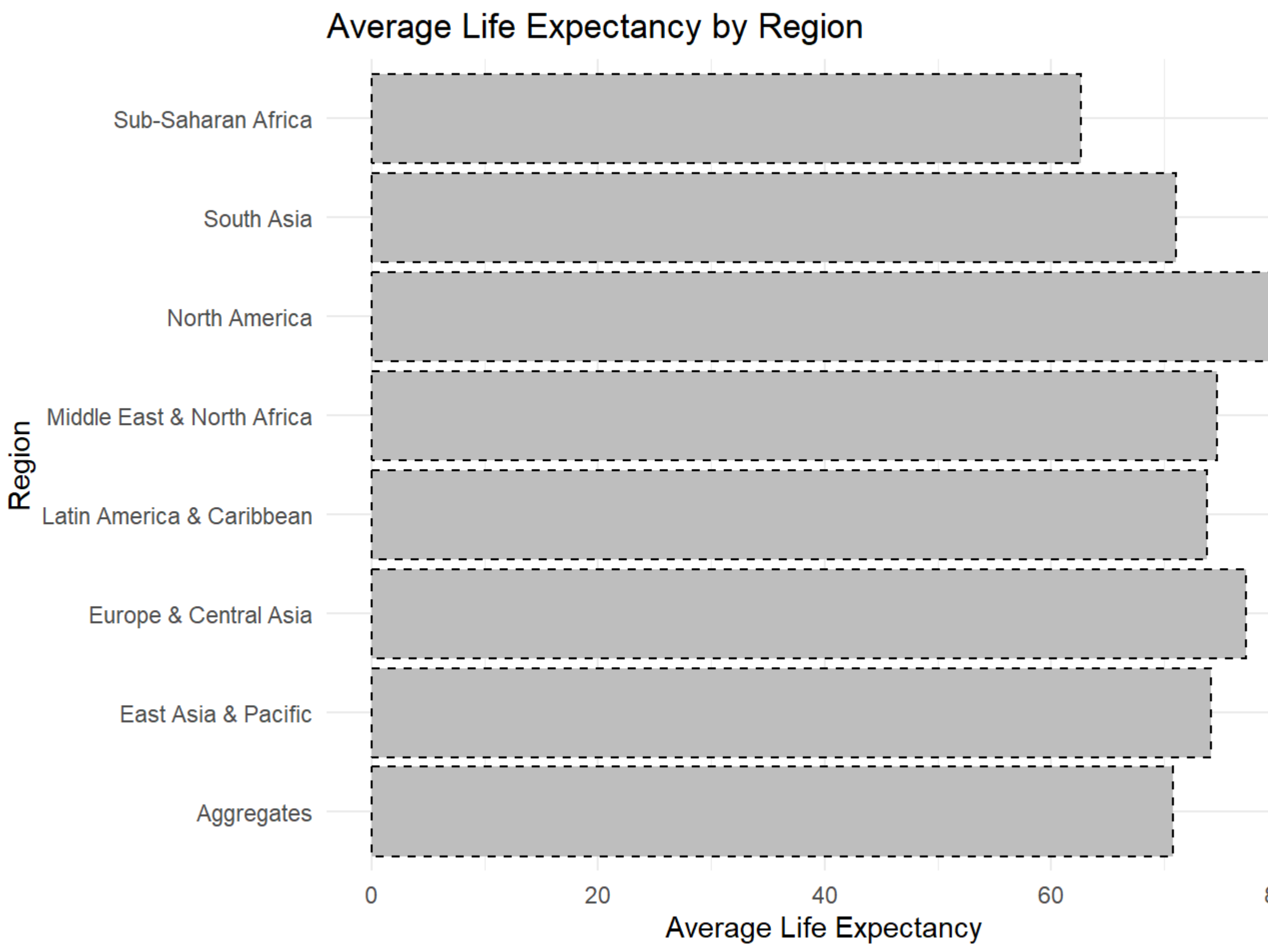


The labels are overlapping and it does not look great. We can easily fix this by adding `coord_flip()`:

```
# Bar plot for GDPPC by region (transposed)
ggplot(summary_by_region, aes(x = region, y = Avg_GDPPC)) +
  geom_bar(stat = "identity", fill = "grey", color = "black", linetype = "dashed") +
  labs(title = "Average GDP per Capita by Region", x = "Region", y = "Average GDP per Capita
  theme_minimal() +
  coord_flip()  # Transpose the bars
```



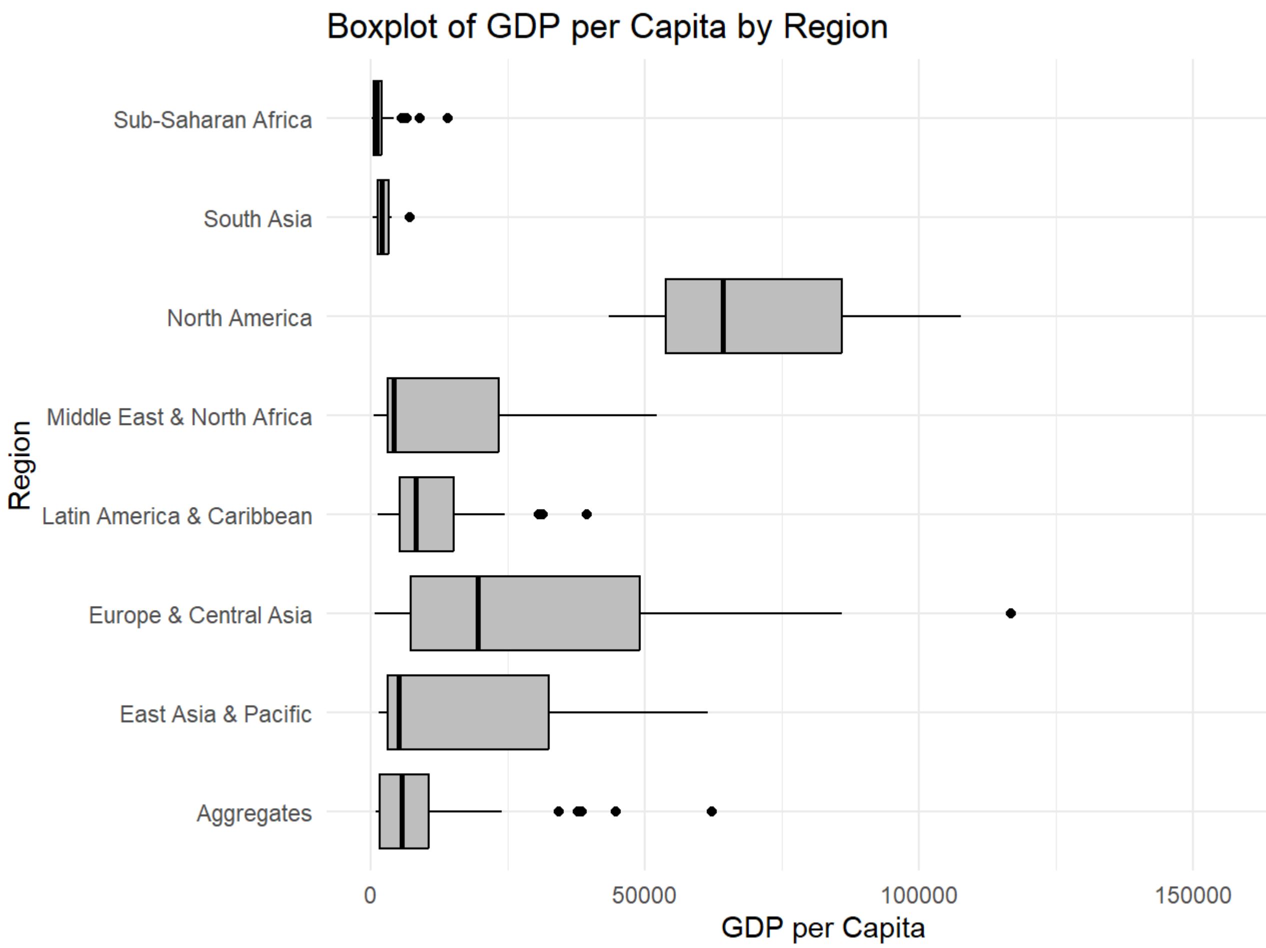

```
# Bar plot for Life Expectancy by region (transposed)
ggplot(summary_by_region, aes(x = region, y = Avg_LifeExpectancy)) +
  geom_bar(stat = "identity", fill = "grey", color = "black", linetype = "dashed") +
  labs(title = "Average Life Expectancy by Region", x = "Region", y = "Average Life Expectan
  theme_minimal() +
  coord_flip()  # Transpose the bars
```



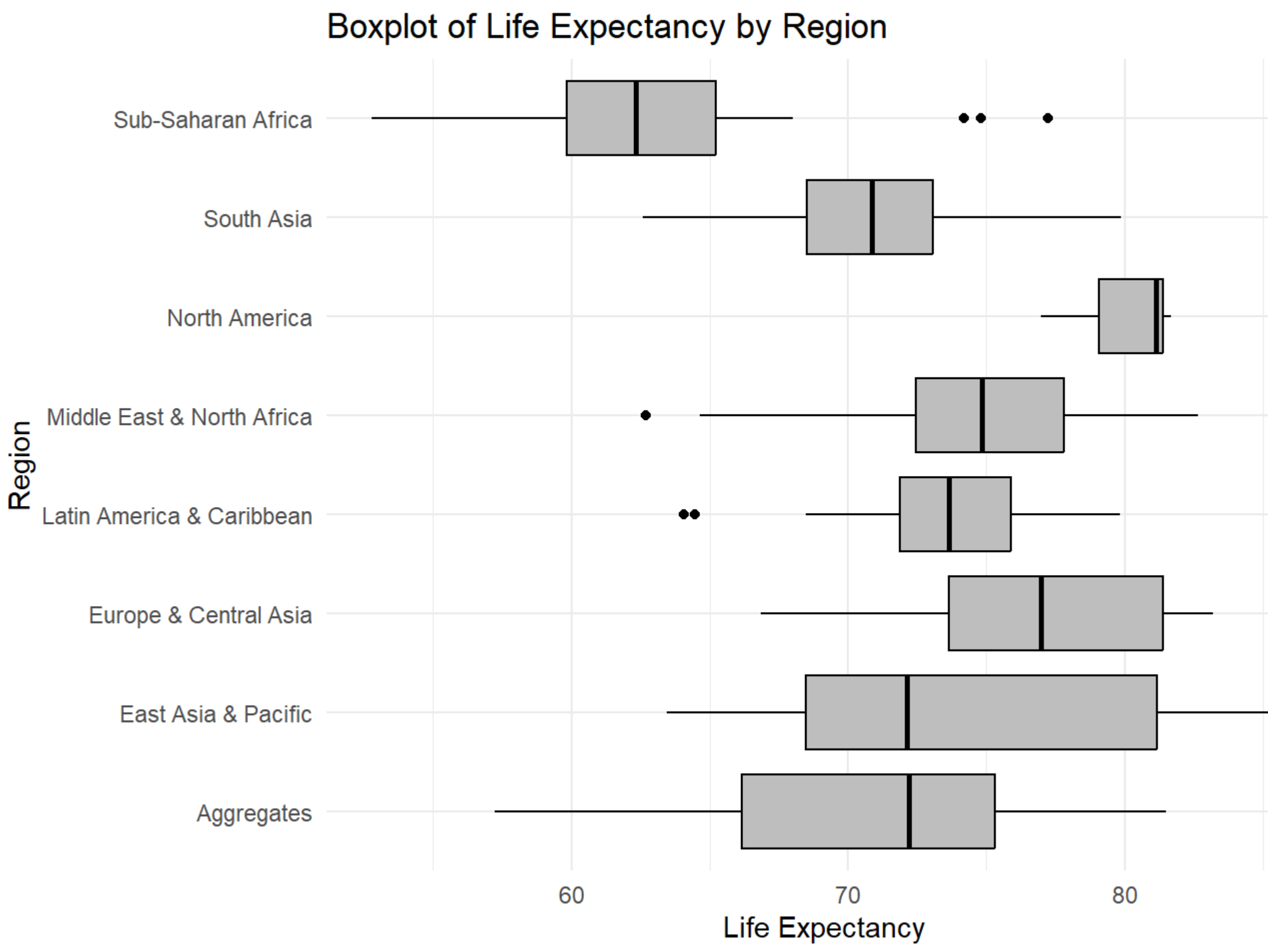
Step 6: Plot Box Plots

Box plots are useful for visualizing the distribution of GDP per capita and life expectancy across regions.

```
# Box plot for GDPPC by region
ggplot(data, aes(x = region, y = GDPPC)) +
  geom_boxplot(fill = "grey", color = "black") +
  labs(title = "Boxplot of GDP per Capita by Region", x = "Region", y = "GDP per Capita") +
  theme_minimal() +
  coord_flip()  # Transpose the bars
```



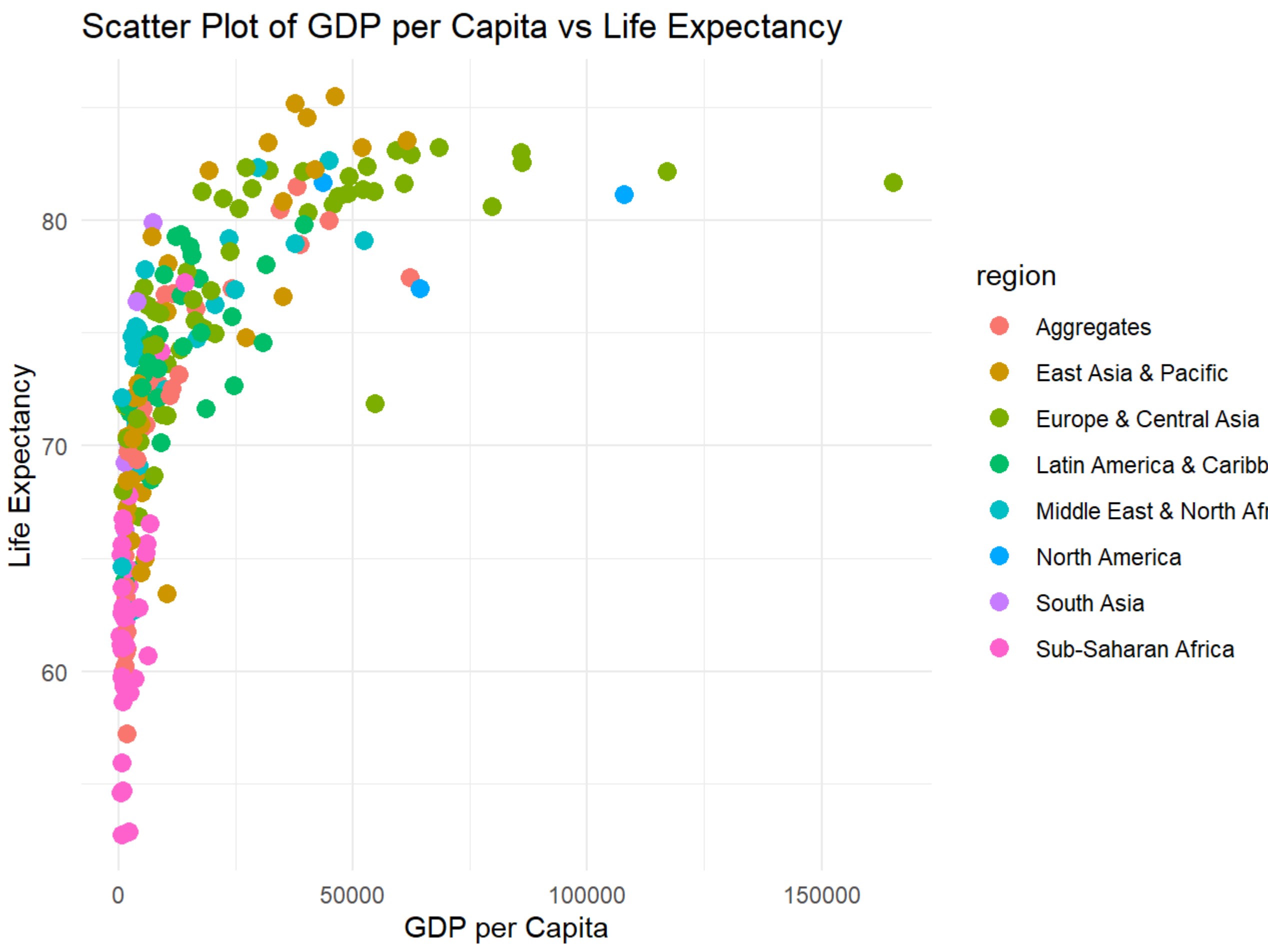

```r
# Box plot for Life Expectancy by region
ggplot(data, aes(x = region, y = LifeExpectancy)) +
  geom_boxplot(fill = "grey", color = "black") +
  labs(title = "Boxplot of Life Expectancy by Region", x = "Region", y = "Life Expectancy")
  theme_minimal() +
  coord_flip()  # Transpose the bars
```



Step 7: Plot Scatter Plot

Finally, we'll create a scatter plot to examine the relationship between GDP per capita and life expectancy. We'll color the points by region to make the plot more informative.

```r
# Scatter plot for GDPPC vs Life Expectancy, colored by region
ggplot(data, aes(x = GDPPC, y = LifeExpectancy, color = region)) +
  geom_point(size = 3) +
  labs(title = "Scatter Plot of GDP per Capita vs Life Expectancy",
       x = "GDP per Capita", y = "Life Expectancy") +
  theme_minimal()
```



Step 8: Customize Aesthetic Features

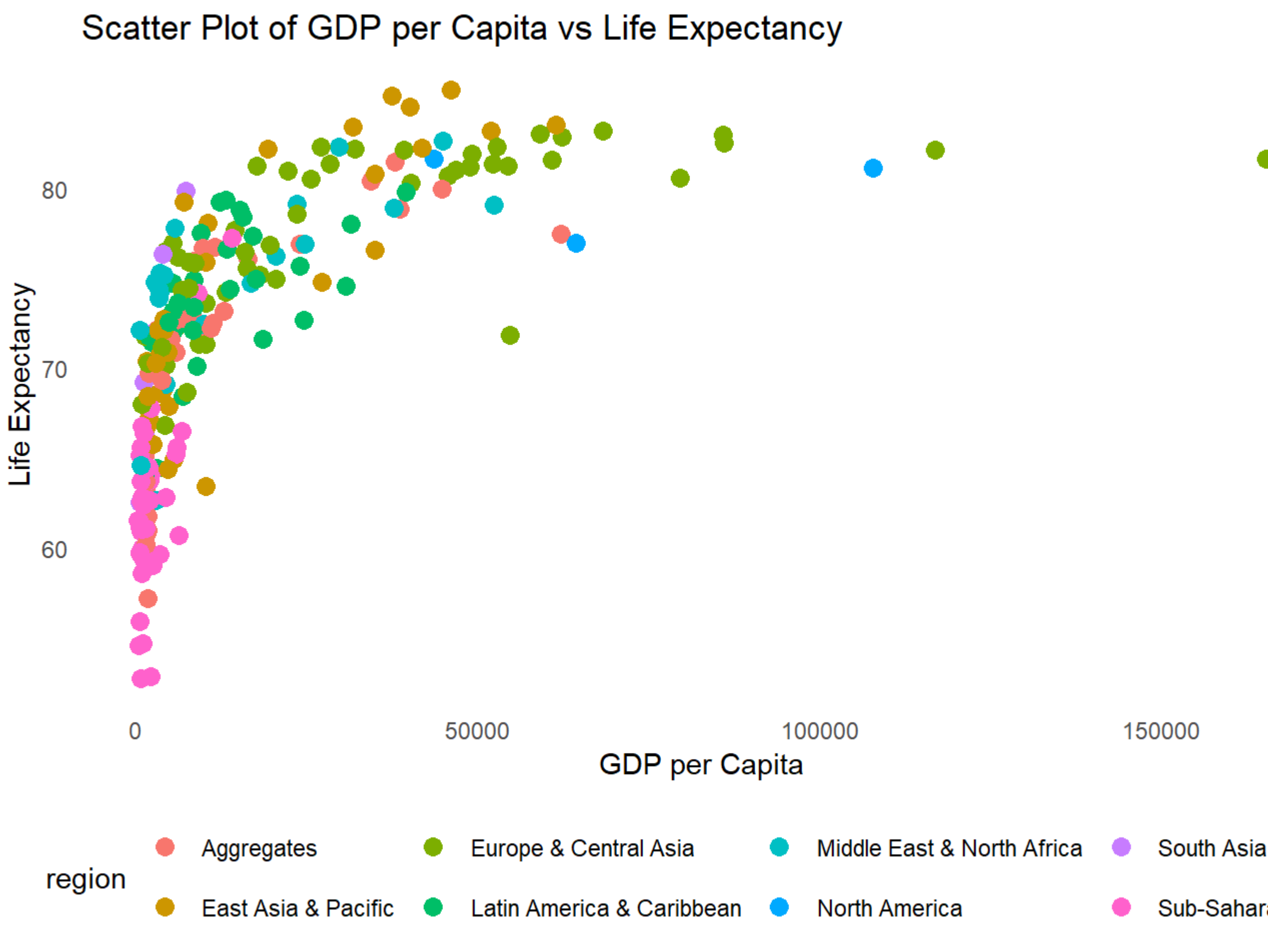
To enhance the minimalistic and clean design, we can apply a few customizations to the plots:

- Remove grid lines for a cleaner look.
- Use a simple font.
- Keep the color palette minimal and use grey tones.

This is achieved by using `theme_minimal()` and additional options to remove grid lines.

```
# Scatter plot with minimalistic aesthetics
ggplot(data, aes(x = GDPPC, y = LifeExpectancy, color = region)) +
  geom_point(size = 3) +
  labs(title = "Scatter Plot of GDP per Capita vs Life Expectancy",
```

```
       x = "GDP per Capita", y = "Life Expectancy") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = "bottom")
```



Step 9: Create a New Variable `logGDPPC`

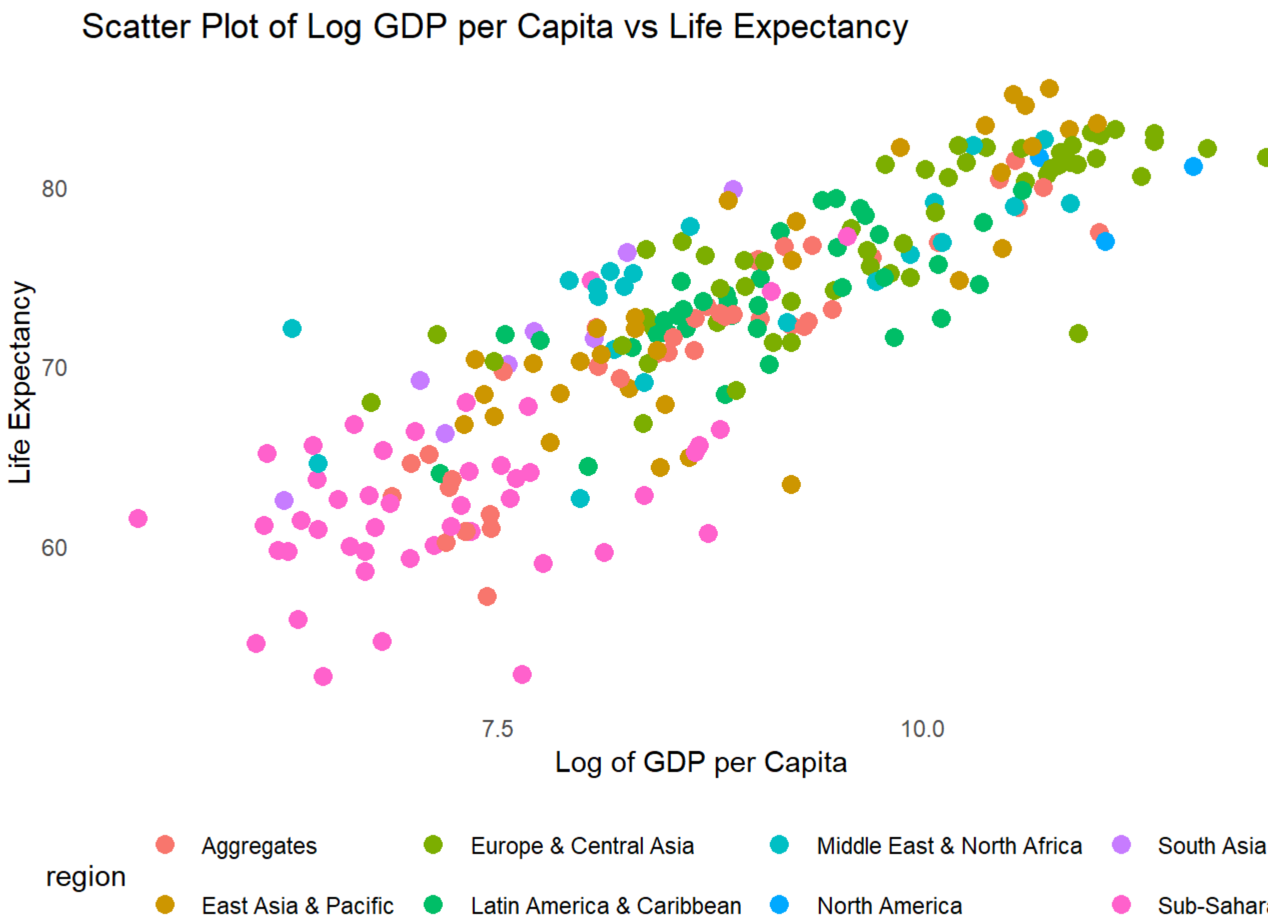
In many datasets, variables like GDP per capita tend to have a highly skewed distribution due to the vast differences in wealth between countries. For example, a few countries have very high GDP per capita, while many have significantly lower GDP. This kind of skewed distribution can make it difficult to visualize patterns or relationships, particularly in scatter plots. By transforming GDP per capita using a logarithmic scale, we can mitigate the effect of extreme values and focus on proportional differences, which often reveal clearer trends.

Why Use Log Transformation?

- **Reduce Skewness**: Log transformation compresses the range of values, which reduces the impact of extreme outliers.
- **Improves Visualization**: Relationships between variables like GDP per capita and life expectancy are often nonlinear. Log transformation helps linearize relationships, making patterns easier to visualize.
- **Proportional Comparison**: It emphasizes relative (percentage) changes rather than absolute differences, which can be more insightful when comparing countries.

Code Implementation: We will create a new variable `logGDPPC` by taking the natural logarithm of `GDPPC` and then plot it against life expectancy.

```
# Create a new variable 'logGDPPC'
data <- data %>%
  mutate(logGDPPC = log(GDPPC))

# Scatter plot of logGDPPC vs Life Expectancy
ggplot(data, aes(x = logGDPPC, y = LifeExpectancy, color = region)) +
  geom_point(size = 3) +
  labs(title = "Scatter Plot of Log GDP per Capita vs Life Expectancy",
       x = "Log of GDP per Capita", y = "Life Expectancy") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = "bottom")
```



Explanation:

1. **`mutate(logGDPPC = log(GDPPC))`**: This line creates the new `logGDPPC` variable by applying the natural logarithm to `GDPPC`.
2. **Scatter Plot of `logGDPPC` vs Life Expectancy**: We replace the `GDPPC` variable on the x-axis with `logGDPPC` to observe how life expectancy relates to the log of GDP per capita.

This transformation often reveals a clearer and more linear relationship between GDP per capita and life expectancy, improving the overall interpretability of the scatter plot.

Visual Impact of Log Transformation:

- In the original scatter plot of `GDPPC` vs Life Expectancy, countries with very high GDP per capita (e.g., oil-rich nations or advanced economies) might create a visual distortion, pulling the plot towards the high end.
- After log-transforming GDP per capita, the values are compressed, and the scatter plot will typically show a more linear and evenly distributed relationship between GDP per capita and life expectancy. This allows for better comparisons between countries with lower GDP as well.

Summary

You now have a full workflow using `dplyr` for data manipulation and `ggplot2` for creating histograms, bar plots, box plots, and scatter plots. The key steps involved:

1. **Data Import**: Used `WDI` to load GDP per capita and life expectancy data.
2. **Data Wrangling**: Used `dplyr` to clean and summarize the data.
3. **Visualizations**:
    - Histograms for distributions of GDPPC and life expectancy.
    - Bar plots for regional averages.
    - Box plots to show the spread of data by region.
    - A scatter plot to explore the relationship between GDPPC and life expectancy.

The plots are styled with a clean and minimalistic aesthetic, using grey colors, dashed lines, and a focus on simplicity. You can expand or refine these visualizations depending on your specific analysis needs.

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

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