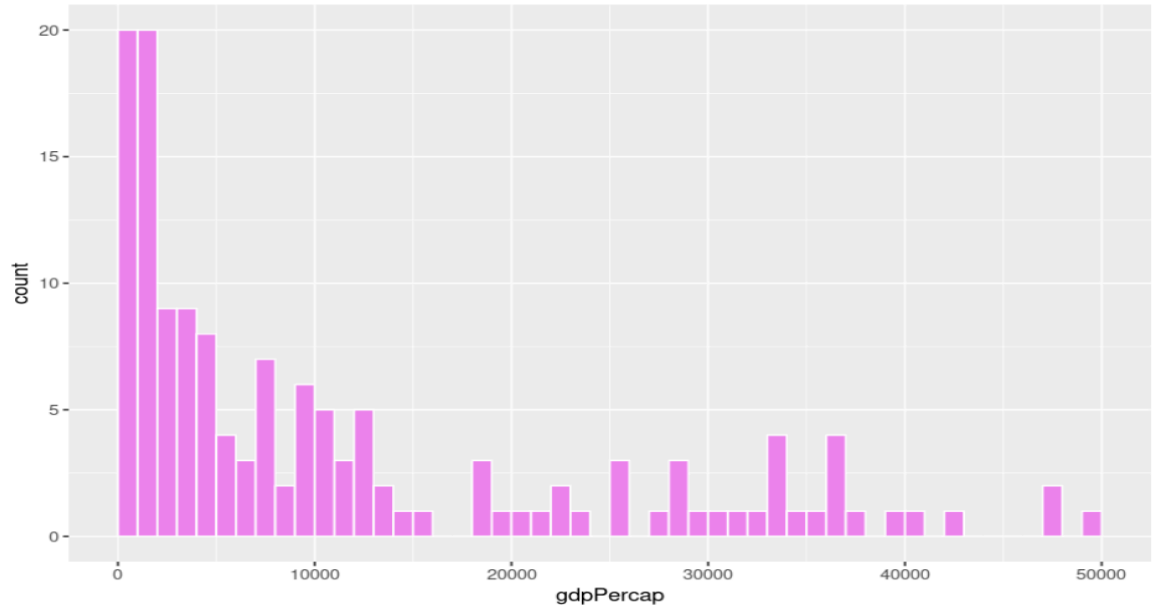
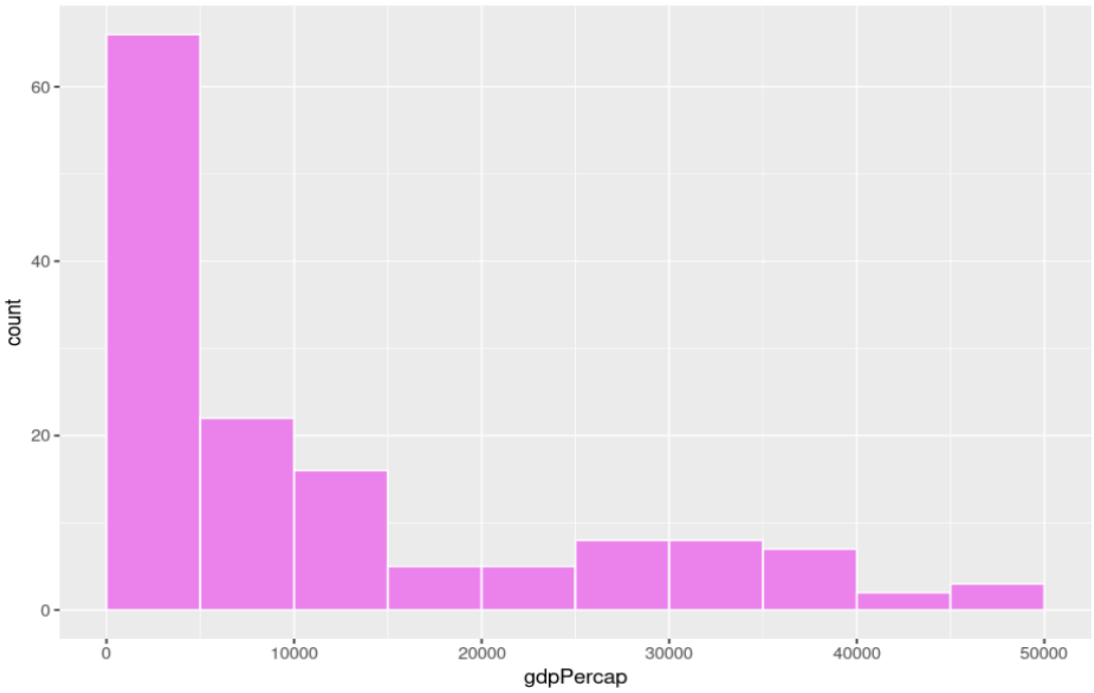
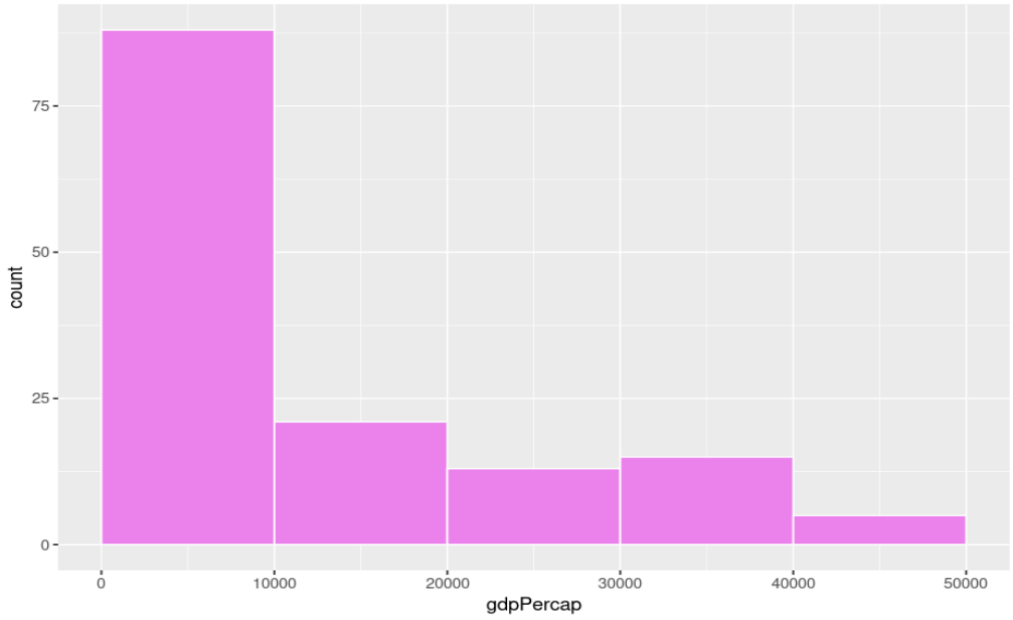
Data Visualisation DA332

Assignment-05

1. a)



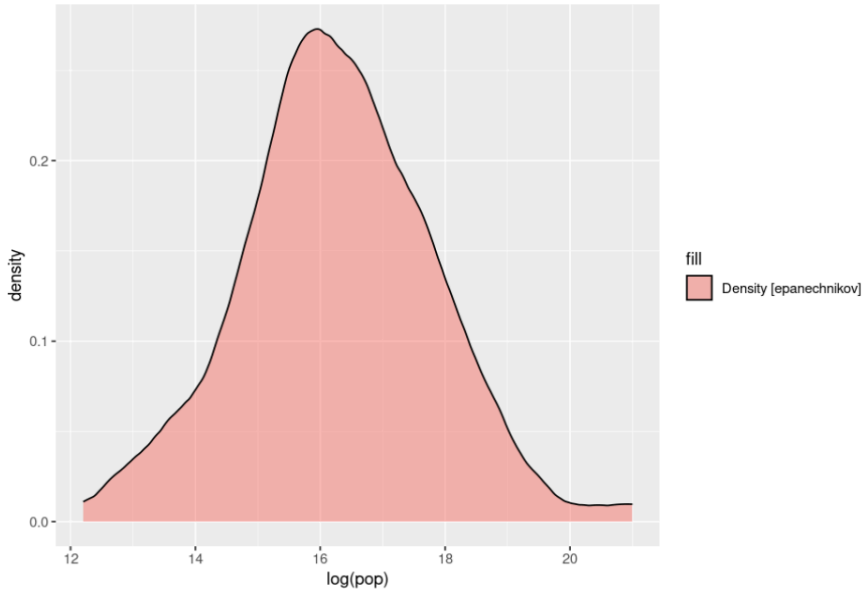


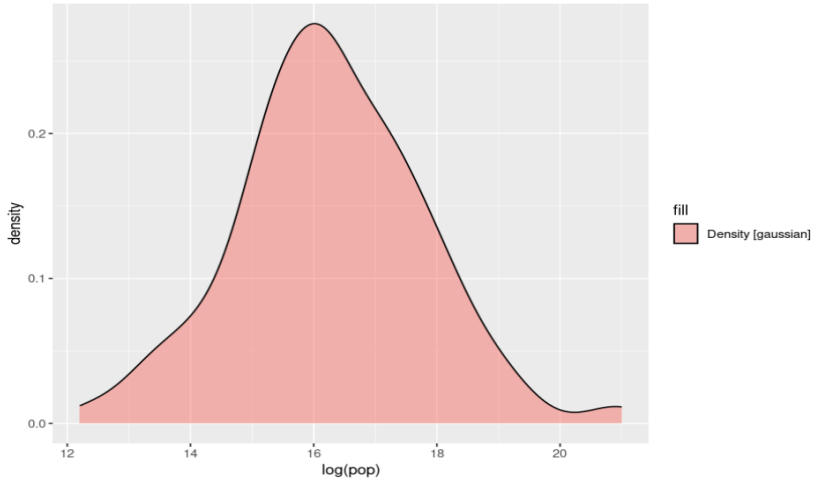


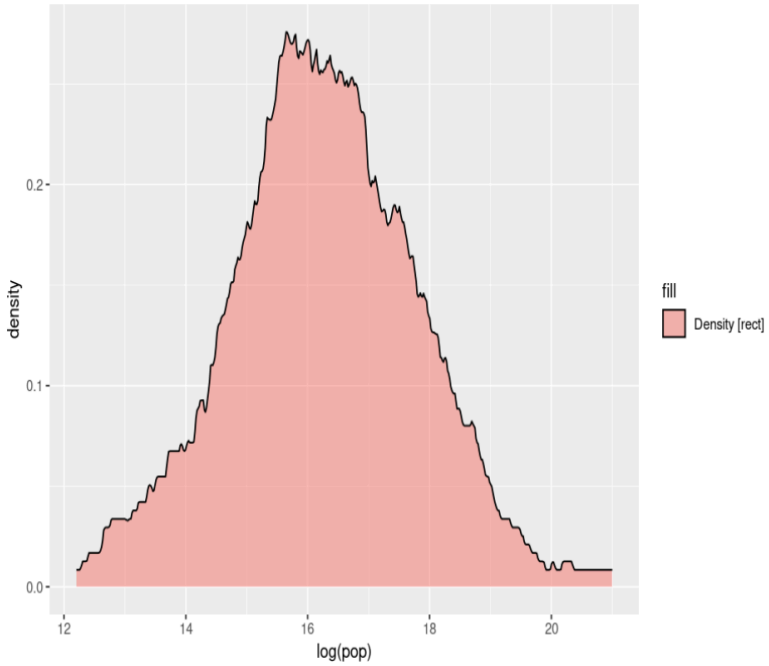
When I choose a bin width for a histogram, it has a big impact on how the distribution looks. If the bins are too wide, I might miss important details and patterns. On the other hand, if the bins are too narrow, the histogram can look too noisy and complicated. The goal is to find a balanced bin width that clearly shows the data's structure without overwhelming me with too much detail. I usually experiment with different widths to see which one provides the best insights.

In this case binwidth = 5000 looks optimal.

b)


density
0.2
0.1
0.0
12
14
16
18
20
log(pop)
fill
Density [epanechnikov]

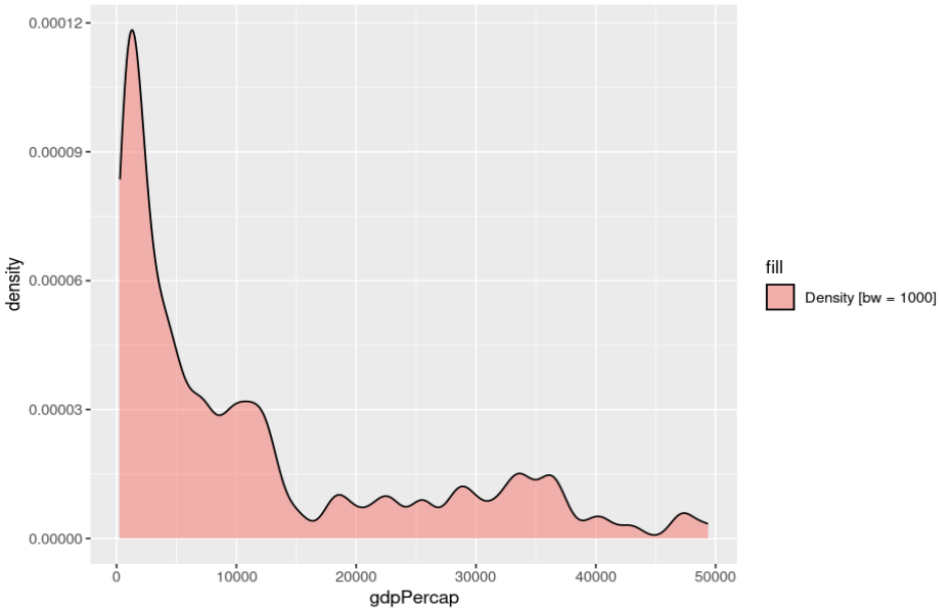

density
0.2
0.1
0.0
12
14
16
18
20
log(pop)
fill
Density [gaussian]

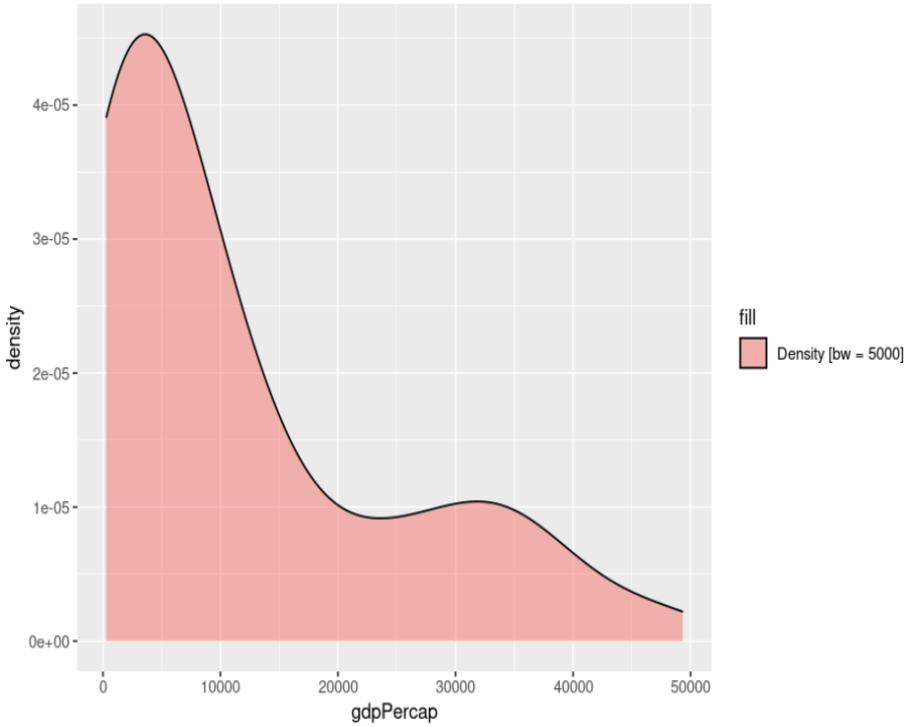


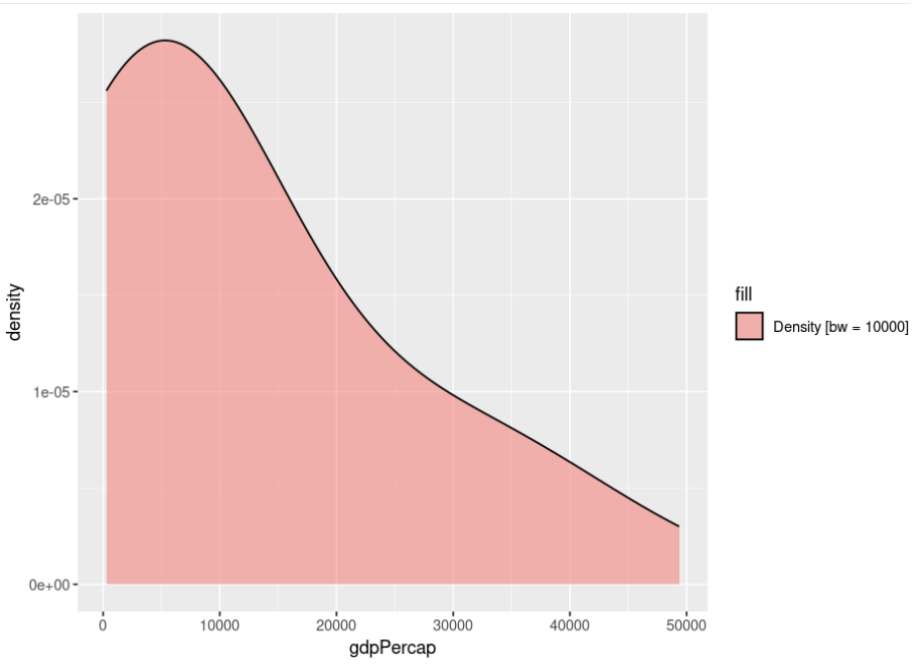
When we use different kernel functions for density estimation, they each affect the shape of the distribution in unique ways:

1. **Gaussian Kernel**: This kernel produces a smooth and continuous density curve. It's ideal for data with a normal distribution and helps in capturing long tails of the distribution.
2. **Rectangular Kernel**: Also known as the uniform kernel, it provides a blocky and less smooth density estimation. It's simple and fast but may not capture finer details of the distribution.
3. **Epanechnikov Kernel**: This kernel provides a balance between smoothness and sensitivity to the data. It's efficient and reduces the effect of outliers, often providing a more accurate representation of the true distribution

c)

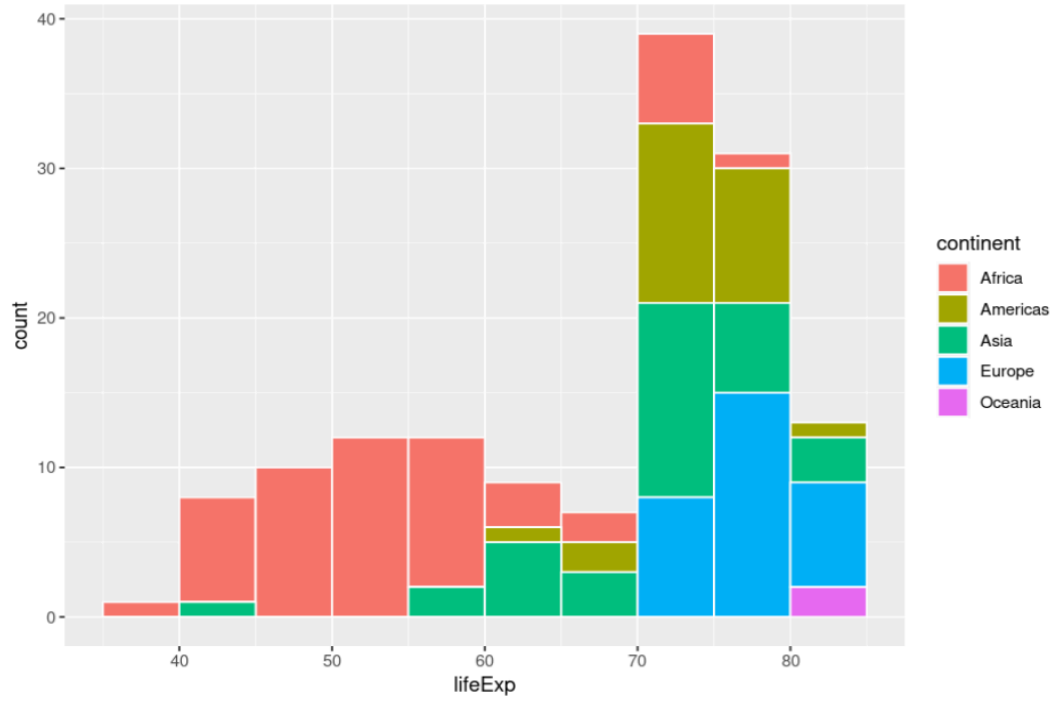

0.00012
0.00009
0.00006
0.00003
0.00000
density
0
10000
20000
30000
40000
50000
gdpPercap
fill
Density [bw = 1000]

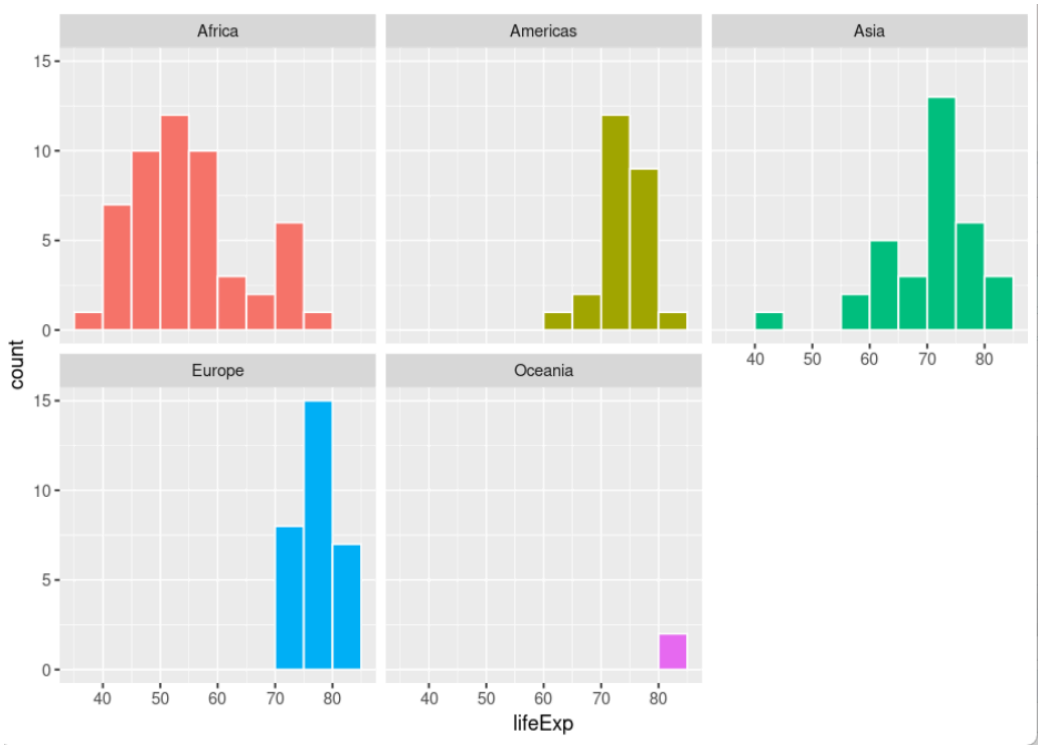

4e-05
3e-05
2e-05
1e-05
0e+00
density
0
10000
20000
30000
40000
50000
gdpPercap
fill
Density [bw = 5000]



When I create density plots of GDP per capita using bandwidths of 1000, 5000, and 10000, I notice the following: With a bandwidth of 1000, the plot is very detailed but noisy. At 5000, there's a nice balance between smoothness and detail. At 10000, the plot is smooth but loses finer details. So, the bandwidth choice directly affects the level of detail and smoothness in the distribution.

d)

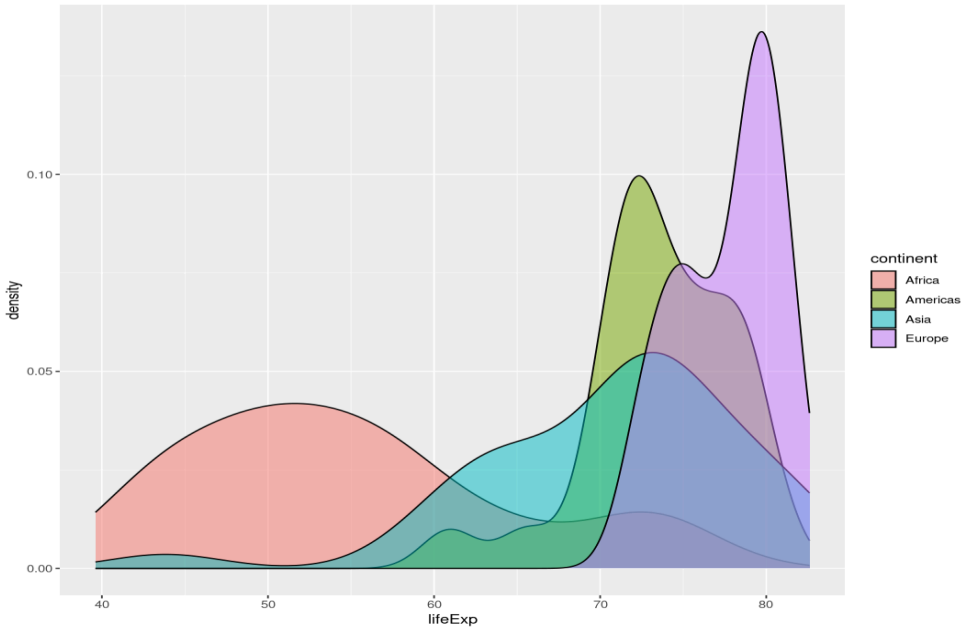


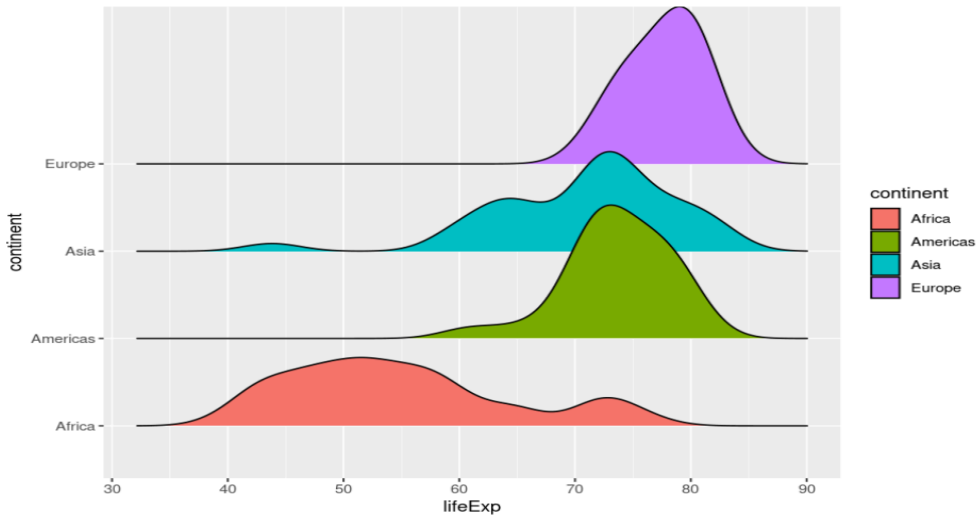


Overlapping Histograms vs. Faceted Small Multiples

- **Overlapping Histograms**: Different continents' life expectancy distributions are shown on one plot with unique colors. It can get cluttered and hard to interpret.
- **Faceted Small Multiples**: Separate histograms for each continent in a grid. It's cleaner and easier to compare distributions across continents.

e)



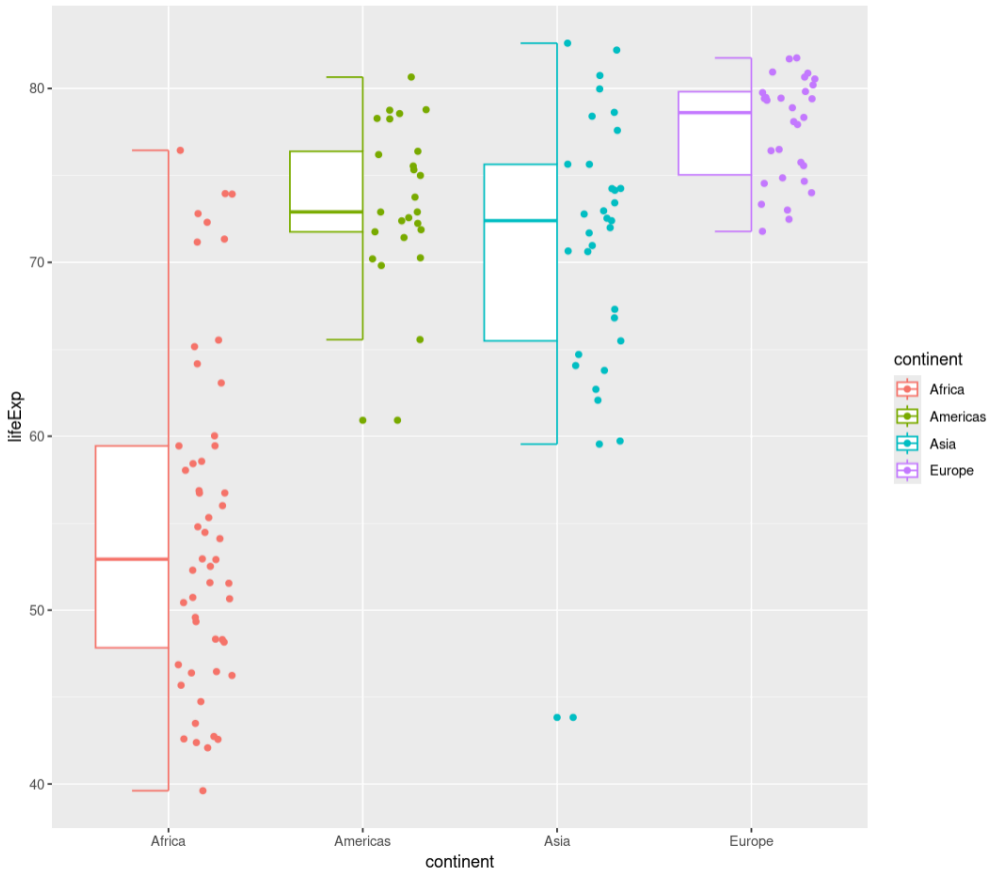


Overlapping Densities with Transparency vs. Ridge Plots

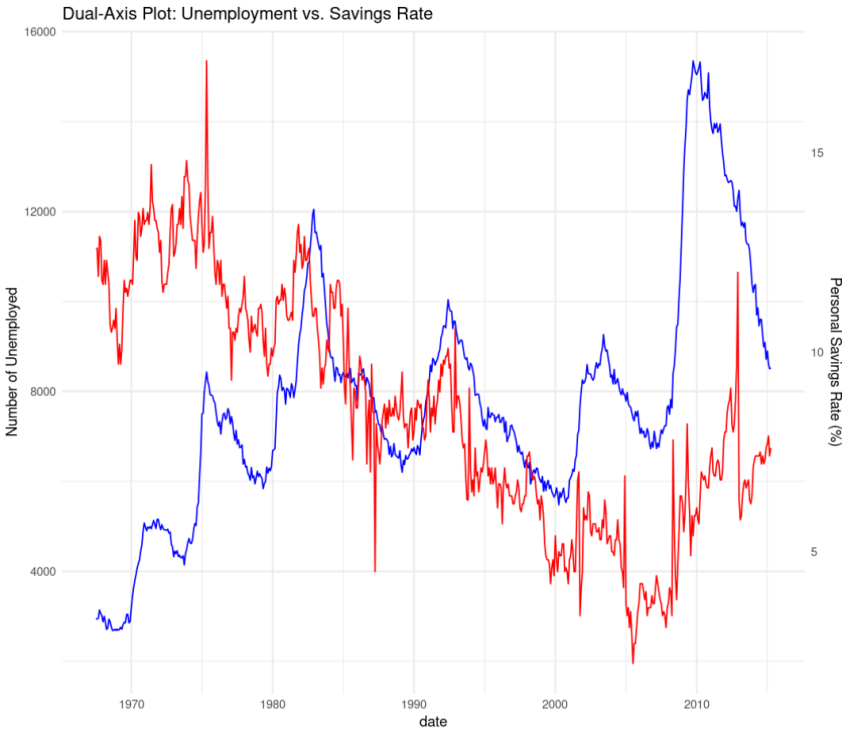
- **Overlapping Densities**: Density plots for each continent on one graph with transparency. It provides a smooth view but can be cluttered.
- **Ridge Plots**: Stacked density plots for each continent, reducing clutter and making comparisons easier.

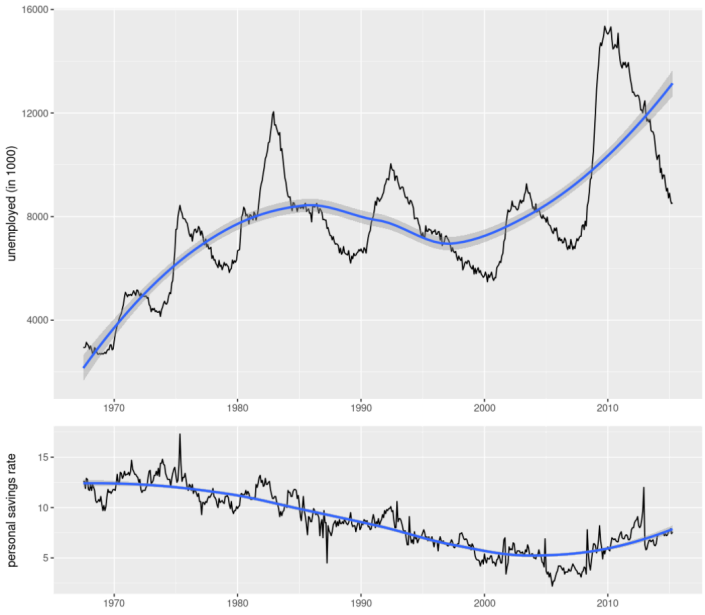
Ridge plots and **faceted small multiples** are better because they provide clearer, more organized visual representations and facilitate easier comparisons

f)



2)

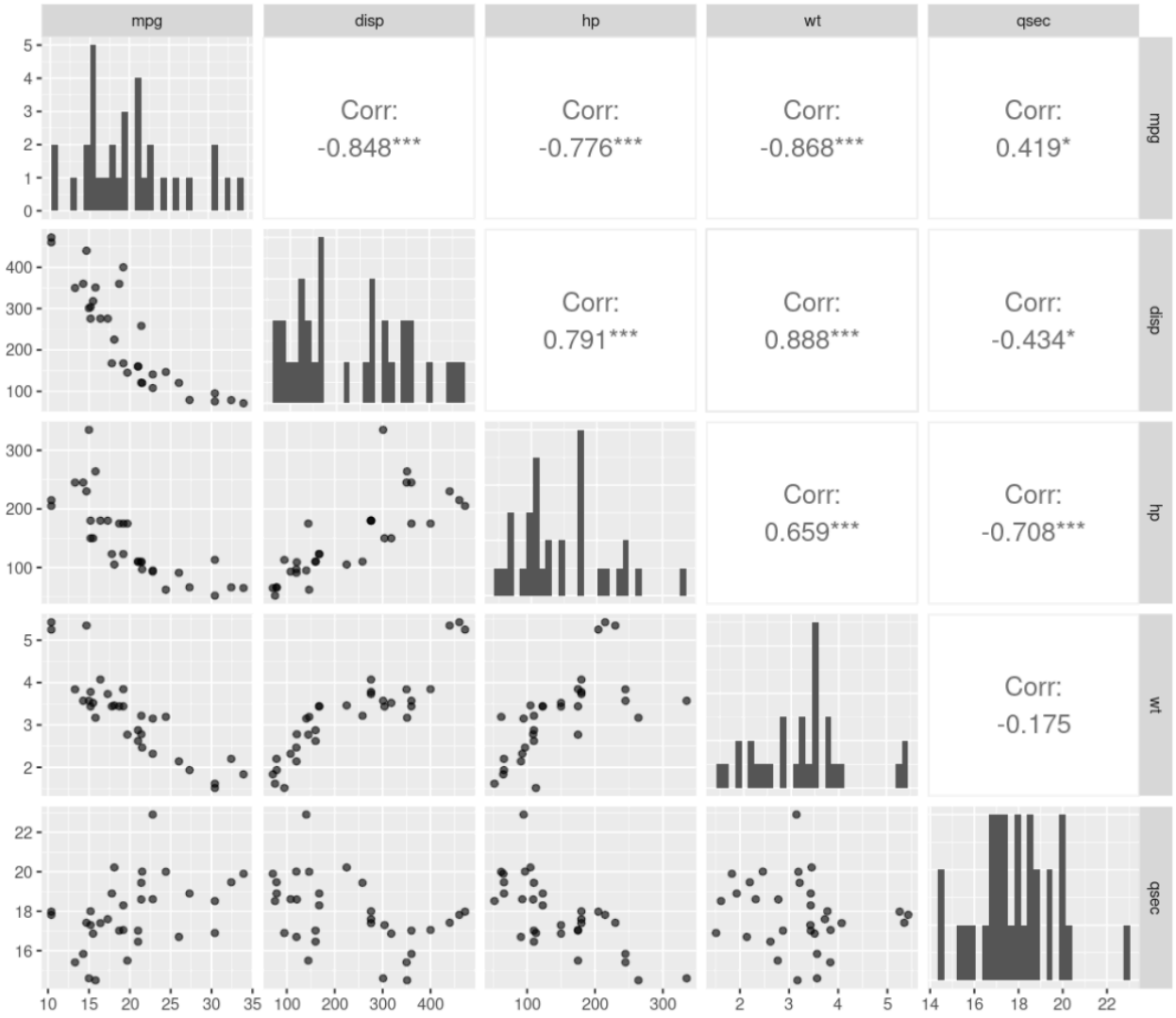




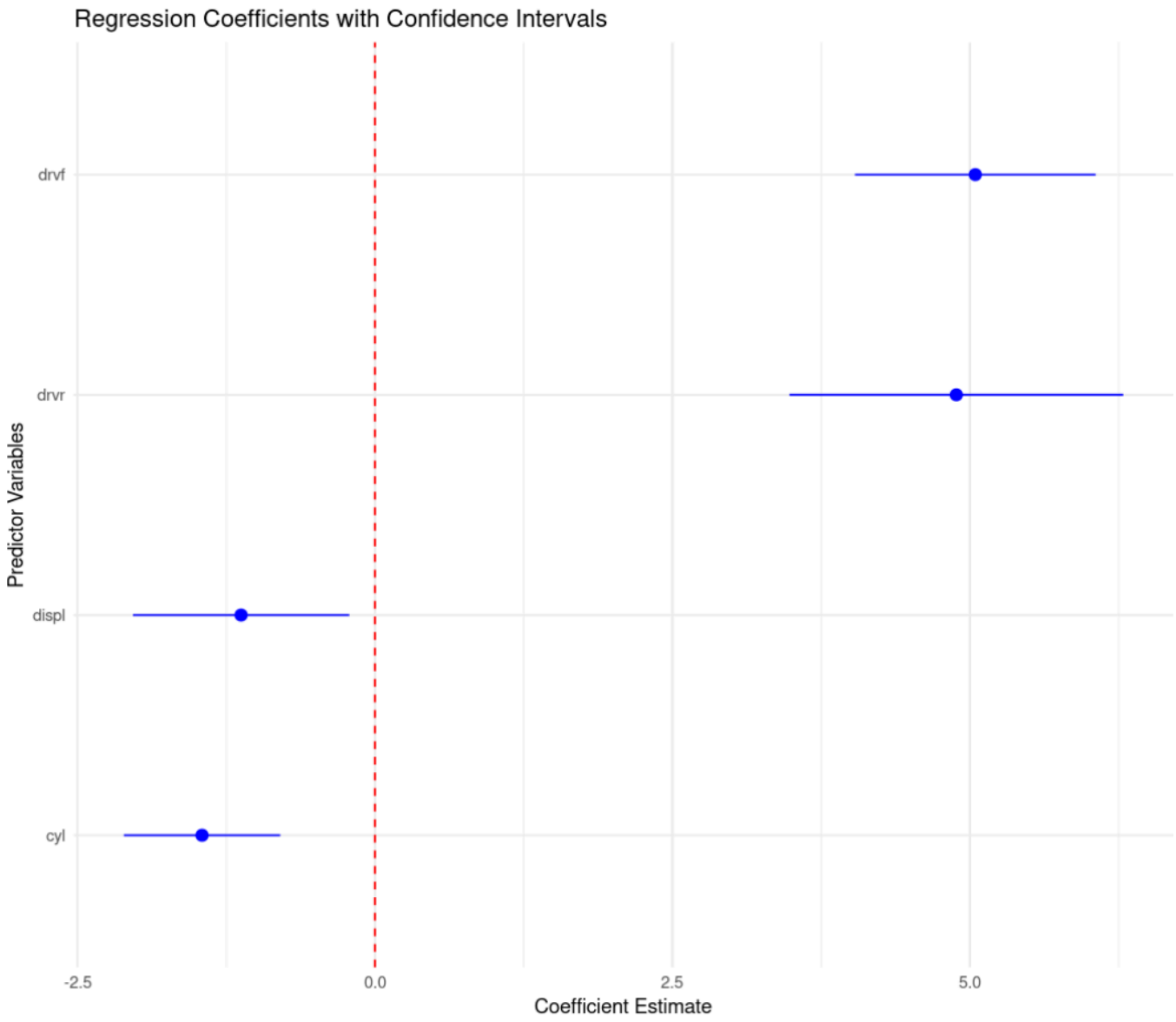
When I create a dual-axis plot, it's challenging to accurately compare the two variables due to different scales, which can be misleading. Using the secondary plot method, both

variables are plotted separately but share a common x-axis (date), making it much clearer. This method avoids confusion by allowing me to see each variable's trend without the risk of misinterpreting their relationship. It's a more transparent and effective way to visualize data.

3)



4)


Regression Coefficients with Confidence Intervals
drvf
drvr
displ
cyl
Predictor Variables
-2.5
0.0
2.5
5.0
Coefficient Estimate

References:

[1] https://rkabacoff.github.io/datavis/index.html

[2] https://clauswilke.com/dataviz/index.html

[3] https://datavizs21.classes.andrewheiss.com/slides/06-slides.pdf

[4] https://datavizs21.classes.andrewheiss.com/slides/07-slides.pdf

[5] https://datavizs21.classes.andrewheiss.com/slides/08-slides.pdf

R Code:

```
library (ggplot2)

library (viridis)

library (RColorBrewer)

library (readr)

library (scales)

library (dplyr)

library (readxl)

library(tidyverse)



# Reference:

# https://datavizs21.classes.andrewheiss.com/slides/06-slides.pdf

# https://datavizs21.classes.andrewheiss.com/slides/07-slides.pdf

# https://datavizs21.classes.andrewheiss.com/slides/08-slides.pdf



# Q1.

install.packages("gapminder")

library(gapminder)



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

```
# data$country

ggplot(data,

       aes(x = gdpPercap)) +

  geom_histogram(binwidth = 1000, color="white",fill="violet",boundary = 10000)

ggplot(data,

       aes(x = gdpPercap)) +

  geom_histogram(binwidth = 5000, color="white",fill="violet",boundary = 10000)

ggplot(data,

       aes(x = gdpPercap)) +

  geom_histogram(binwidth = 10000, color="white",fill="violet",boundary = 10000)
```

```
ggplot(data, aes(x = log(pop), y = ..density..)) +

  geom_density(aes(fill = "Density [rect]"), alpha = 0.5,

               kernel = "rectangular")

ggplot(data, aes(x = log(pop), y = ..density..)) +

  geom_density(aes(fill = "Density [gaussian]"), alpha = 0.5,

               kernel = "gaussian")

ggplot(data, aes(x = log(pop), y = ..density..)) +

  geom_density(aes(fill = "Density [epanechnikov]"), alpha = 0.5,

               kernel = "epanechnikov")
```

```
ggplot(data, aes(x = gdpPercap , y = ..density..)) +

  geom_density(aes(fill = "Density [bw = 1000]"), alpha = 0.5,bw = 1000)

ggplot(data, aes(x = gdpPercap , y = ..density..)) +

  geom_density(aes(fill = "Density [bw = 5000]"), alpha = 0.5,bw = 5000)
```

```
ggplot(data, aes(x = gdpPercap , y = ..density..)) +
  geom_density(aes(fill = "Density [bw = 10000]"), alpha = 0.5,bw = 10000)


ggplot(data,
       aes(x = lifeExp,
           fill = continent)) + geom_histogram(binwidth = 5, color = "white",
                                               boundary = 50)
ggplot(data,
       aes(x = lifeExp,
           fill = continent)) + geom_histogram(binwidth = 5, color = "white",
                                               boundary = 50)+
  guides(fill = "none") +
  facet_wrap(vars(continent))


ggplot(filter(data,
              continent != "Oceania"),
       aes(x = lifeExp,
           fill = continent)) +
  geom_density(alpha = 0.5)


library(ggridges)
ggplot(filter(data,
              continent != "Oceania"),
       aes(x = lifeExp,
           fill = continent,
           y = continent)) +
  geom_density_ridges()
```

```
library(gghalves)

ggplot(filter(data,

              continent != "Oceania"),

       aes(y = lifeExp,

           x = continent,

           color = continent)) +

  geom_half_boxplot(side = "l") +

  geom_half_point(side = "r")




# Q2.

data <- economics


# Rescale psavert to match the unemploy range

scale_factor <- max(economics$unemploy) / max(economics$psavert)


ggplot(economics, aes(x = date)) +

  geom_line(aes(y = unemploy), color = "blue") +

  geom_line(aes(y = psavert * scale_factor), color = "red") +

  scale_y_continuous(

    name = "Number of Unemployed",

    sec.axis = sec_axis(~ . / scale_factor, name = "Personal Savings Rate (%)")

  ) +

  labs(title = "Dual-Axis Plot: Unemployment vs. Savings Rate") +

  theme_minimal()
```

```
library(patchwork)
g1 <- ggplot(data,
                    aes(x = date,
                        y = unemploy))+
geom_line() + geom_smooth() +
  labs(x = NULL, y = "unemployed (in 1000)")
g2 <- ggplot(data,
                     aes(x = date,
                         y = psavert)) +
  geom_line() + geom_smooth() +
  labs(x = NULL, y = "personal savings rate")
g1 + g2 +
  plot_layout(ncol = 1,
              heights = c(0.7, 0.3))


# Q3.
library(GGally)
library(ggplot2)
# Select relevant variables from the mtcars dataset
data <- mtcars[, c("mpg", "disp", "hp", "wt", "qsec")]
# Generate the pairwise correlation matrix plot
ggpairs(
  data,
  lower = list(continuous = wrap("points", alpha = 0.6)),  # Scatter plots in lower triangle
  diag = list(continuous = wrap("densityDiag")),           # Density plots on the diagonal
  upper = list(continuous = wrap("cor", size = 5))         # Correlation coefficients in upper triangle
)
```

```
# Qustion no - 4

library(broom)



# Fit the regression model

model <- lm(hwy ~ displ + cyl + drv, data = mpg)



# Extract coefficients along with confidence intervals

coeff_data <- tidy(model, conf.int = TRUE) %>%

  filter(term != "(Intercept)")  # Exclude intercept



# Create the coefficient plot

ggplot(coeff_data, aes(x = estimate, y = reorder(term, estimate))) +

  geom_pointrange(aes(xmin = conf.low, xmax = conf.high), color = "blue") +  # Confidence intervals

  geom_vline(xintercept = 0, color = "red", linetype = "dashed") +           # Reference line at x = 0

  labs(

    title = "Regression Coefficients with Confidence Intervals",

    x = "Coefficient Estimate",

    y = "Predictor Variables"

  ) +  theme_minimal()
```