Analysis of Fertility vs. PPgdp

Sai Mulagan

2023-10-20

Data visualization and pre-processing

1. Scatterplot of Fertility vs. PPgdp

```
library(dplyr)
```

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

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

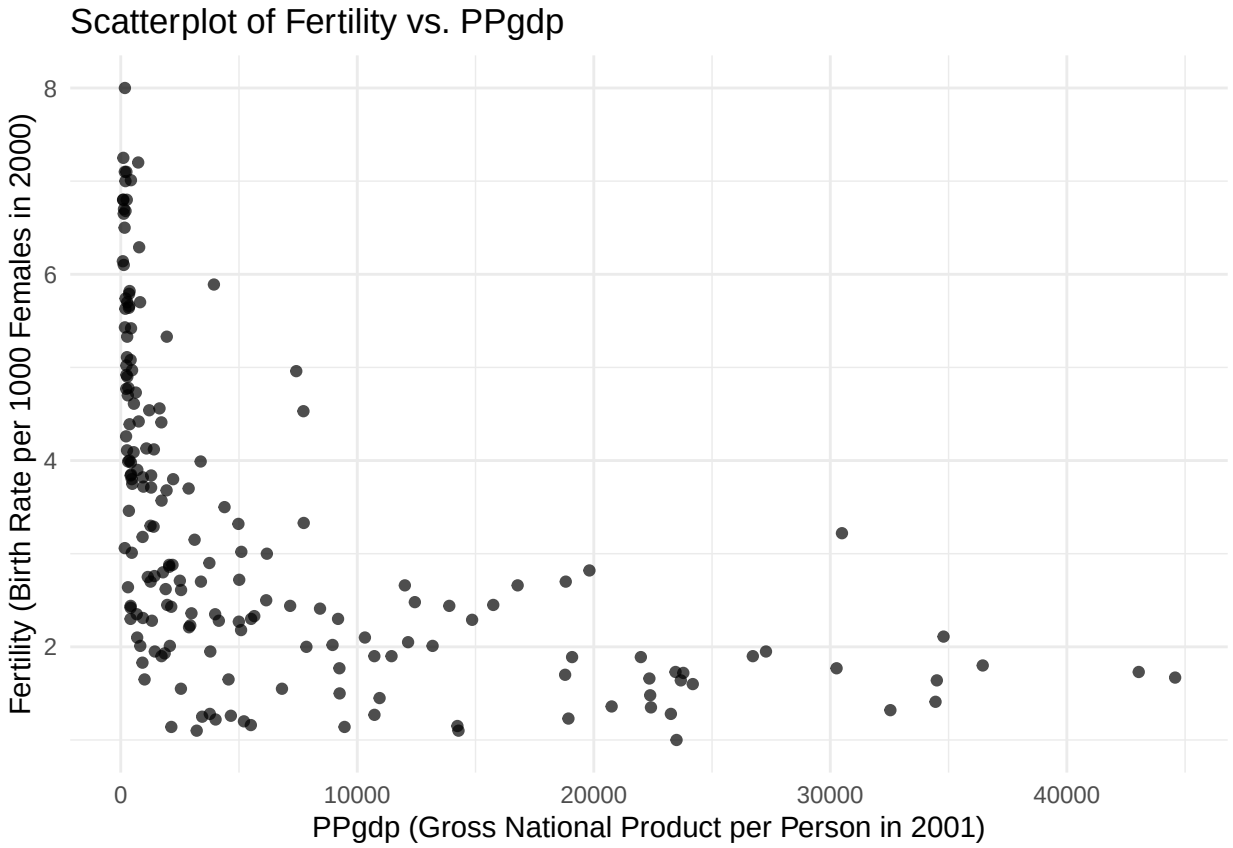
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(Matrix)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
# Load the data
data <- read.table("/Users/sai/Downloads/UN.txt", header=TRUE, sep="")

# Plotting the scatterplot
ggplot(data, aes(x=PPgdp, y=Fertility)) +
  geom_point(alpha=0.7) +
  theme_minimal() +
  labs(title="Scatterplot of Fertility vs. PPgdp",
       x="PPgdp (Gross National Product per Person in 2001)",
       y="Fertility (Birth Rate per 1000 Females in 2000)")
```



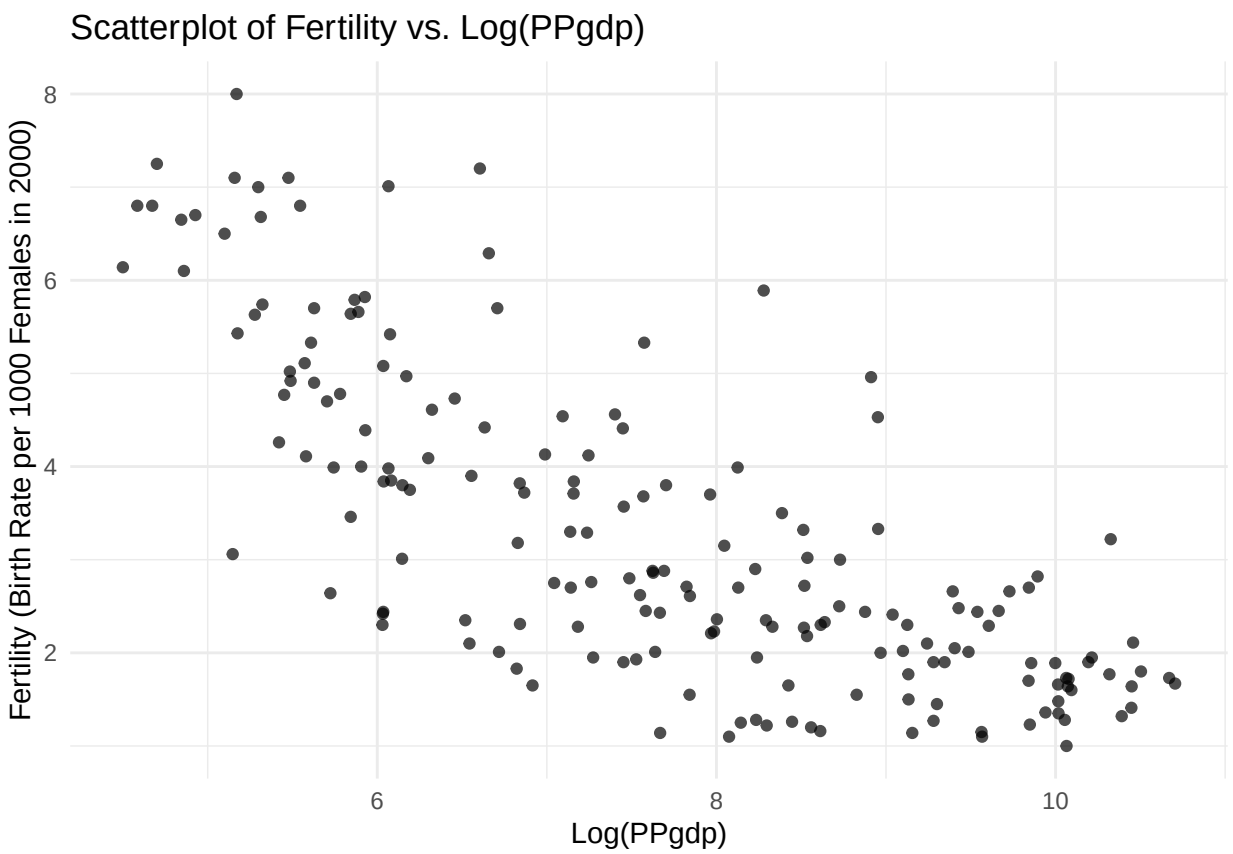
From the graph, we can see a general trend where countries with higher PPgdp values tend to have lower fertility rates. The relationship appears to be nonlinear with a sharp decline in fertility as PPgdp increases, followed by a plateau. Therefore, a simple linear regression model may not be the best fit for this data in its current form.

2. Transformation of Variables

To capture the nonlinear relationship, we can consider a logarithmic transformation of the PPgdp variable. This transformation can help linearize the relationship and make it more suitable for simple linear regression.

```
# Apply logarithmic transformation to PPgdp
data$log_PPgdp <- log(data$PPgdp)

# Plotting the scatterplot of transformed variables
ggplot(data, aes(x=log_PPgdp, y=Fertility)) +
  geom_point(alpha=0.7) +
  theme_minimal() +
  labs(title="Scatterplot of Fertility vs. Log(PPgdp)",
       x="Log(PPgdp)",
       y="Fertility (Birth Rate per 1000 Females in 2000)")
```



After applying the logarithmic transformation to the PPgdp variable, the relationship between Fertility and Log(PPgdp) appears to be more linear. This transformation makes it easier to capture the relationship using a simple linear regression model.

Model fitting and diagnostics

3. Fit the simple linear model on the transformed data

(a) Plain coding (not using the lm function or matrix manipulation)

```
data$log_PPgdp <- log(data$PPgdp)

# Calculate means
mean_log_PPgdp <- mean(data$log_PPgdp)
mean_Fertility <- mean(data$Fertility)

# Compute the slope (beta1) and intercept (beta0) for the regression line
beta1 <- sum((data$log_PPgdp - mean_log_PPgdp) * (data$Fertility - mean_Fertility)) / sum((data$log_PPg
beta0 <- mean_Fertility - beta1 * mean_log_PPgdp

# Compute R-squared value
predicted_Fertility <- beta0 + beta1 * data$log_PPgdp
residuals <- data$Fertility - predicted_Fertility
SSE <- sum(residuals^2)
SST <- sum((data$Fertility - mean_Fertility)^2)
R2 <- 1 - SSE/SST
```

```r
print(paste("Intercept (beta0):", beta0))
```

```
## [1] "Intercept (beta0): 9.25161137752739"
```

```r
print(paste("Slope (beta1):", beta1))
```

```
## [1] "Slope (beta1): -0.778881659463683"
```

```r
print(paste("R-squared:", R2))
```

```
## [1] "R-squared: 0.583260867786204"
```

Using plain coding, the results for the simple linear regression on the transformed data are:

beta0=9.25 beta1=-0.78 R2=0.583

(b) Using the lm function

```r
# Fit the model using lm function (equivalent of the plain coding approach in R)
model <- lm(Fertility ~ log_PPgdp, data=data)

# Extract coefficients and R-squared value
beta0 <- coef(model)[1]
beta1 <- coef(model)[2]
R2 <- summary(model)$r.squared
c(beta0, beta1, R2)
```

```
## (Intercept)   log_PPgdp
##   9.2516114  -0.7788817   0.5832609
```

Using the lm function, the results for the simple linear regression on the transformed data are:

beta0=9.25 beta1=-0.78 R2=0.583

(c) Matrix manipulation

```r
# Prepare the input matrix and output vector
X_matrix <- cbind(1, data$log_PPgdp)
y_vector <- data$Fertility

# Calculate coefficients using matrix operations
beta_matrix <- solve(t(X_matrix) %*% X_matrix) %*% t(X_matrix) %*% y_vector
beta0_matrix <- beta_matrix[1]
beta1_matrix <- beta_matrix[2]

# Calculate the R-squared value using matrix operations
y_pred_matrix <- X_matrix %*% beta_matrix
residuals_matrix <- y_vector - y_pred_matrix
SSE_matrix <- sum(residuals_matrix^2)
SST_matrix <- sum((y_vector - mean(y_vector))^2)
R2_matrix <- 1 - SSE_matrix/SST_matrix
c(beta0_matrix, beta1_matrix, R2_matrix)
```

```
## [1]  9.2516114 -0.7788817  0.5832609
```

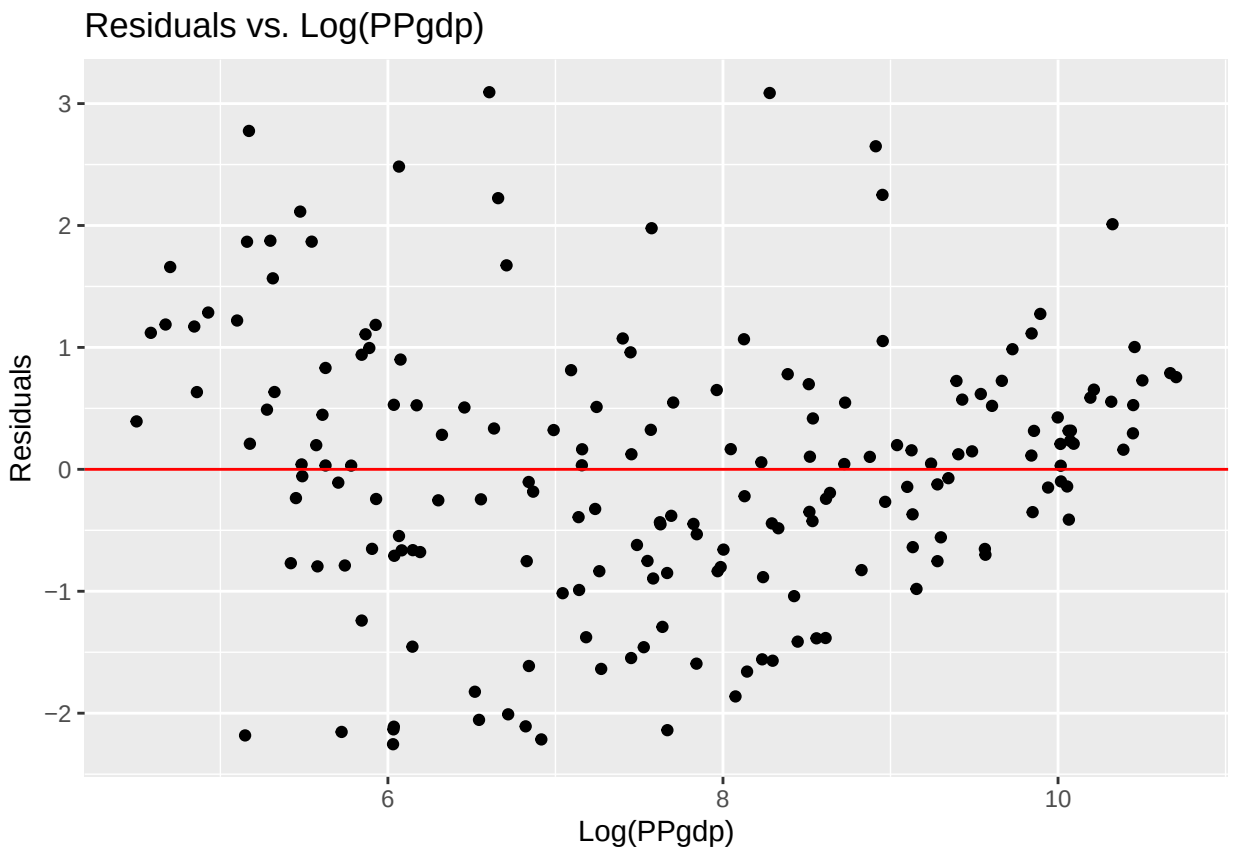
Using matrix manipulation, the results for the simple linear regression on the transformed data are:

beta0=9.25 beta1=-0.78 R2=0.583

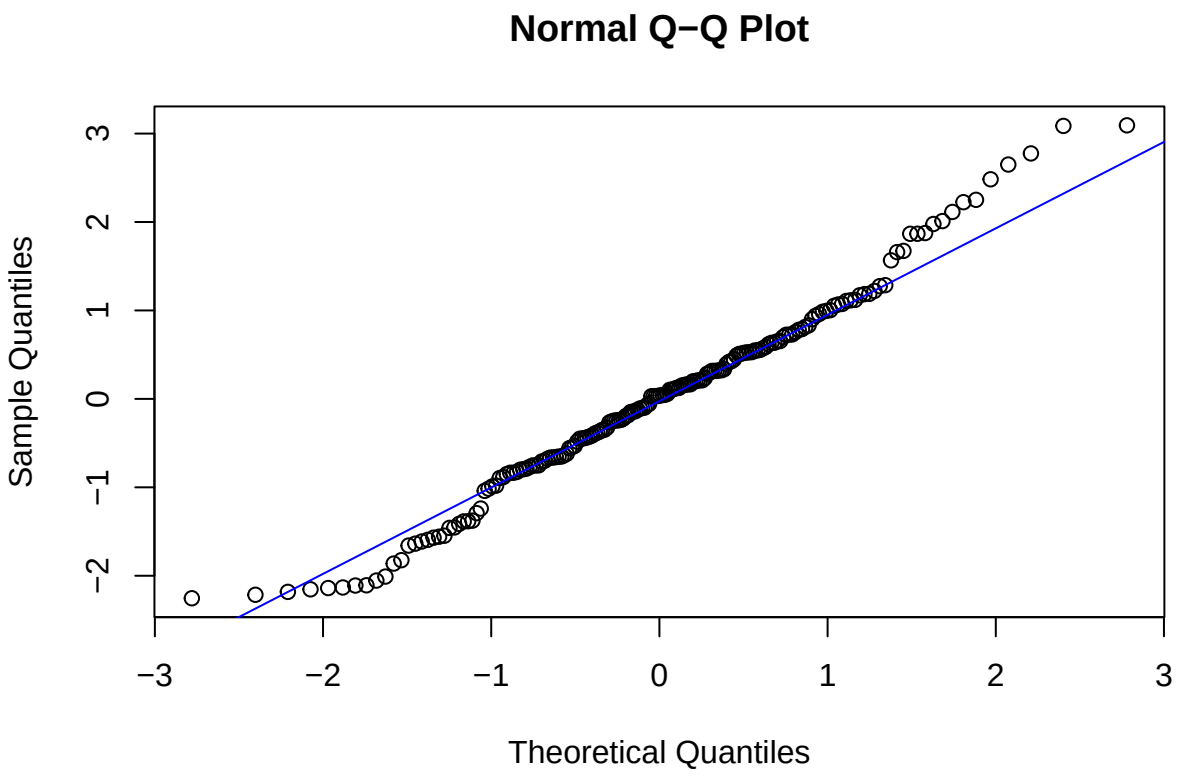
4. Diagnostic Plots

```
model <- lm(Fertility ~ log_PPgdp, data = data)

# Diagnostics
# 1. Identifying Outliers: Using residuals
data$resid = model$residuals
ggplot(data, aes(x = log_PPgdp, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0, col = "red") +
  ggtitle("Residuals vs. Log(PPgdp)") +
  xlab("Log(PPgdp)") +
  ylab("Residuals")
```



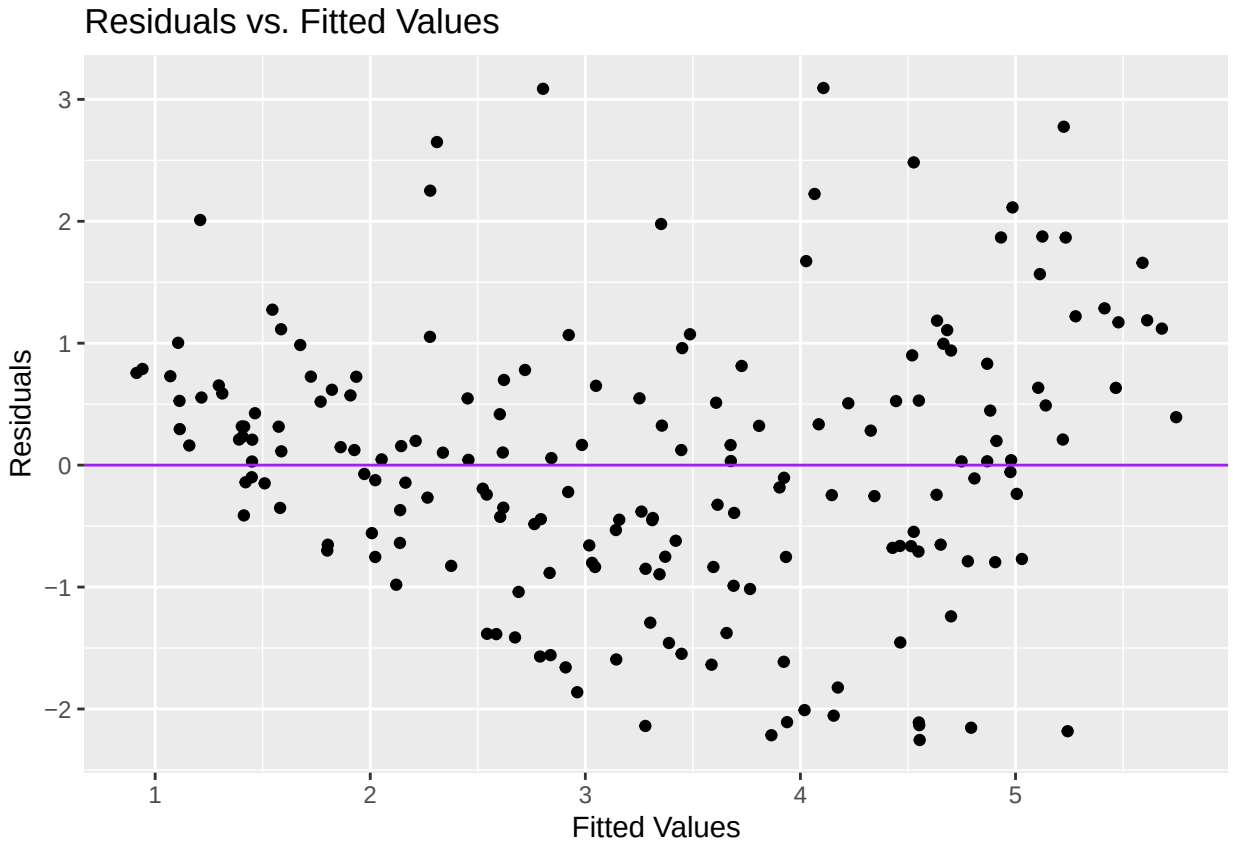

```
# 2. Assessing Normality: QQ plot
qqnorm(model$residuals)
qqline(model$residuals, col = "blue")
```




```
# Shapiro-Wilks test for normality
shapiro.test(model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals
## W = 0.9849, p-value = 0.04512
```

```
# 3. Assessing constant variance (homoscedasticity)
ggplot(data, aes(x = model$fitted.values, y = model$residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, col = "purple") +
  ggtitle("Residuals vs. Fitted Values") +
  xlab("Fitted Values") +
  ylab("Residuals")
```



Residuals vs. log(PPgdp): The residuals should show no systematic pattern and be scattered randomly around the horizontal axis. The plot shows no pattern. We can also see that there are some outliers in the data.

Normal Q-Q Plot: The points should ideally lie on the reference line. The plot shows some deviation, especially at the tails, suggesting potential non-normality in the residuals.

Residuals vs. Fitted Values: The residuals should be randomly scattered around the horizontal axis. In the plot, there's a slight pattern, indicating potential non-linearity in the data that hasn't been fully captured by our model.

Given these diagnostic plots and the Shapiro-Wilks test result, it's evident that our model has certain limitations, especially regarding the assumption of normally distributed residuals

Inference

5. Test for linear relationship

We can conduct a hypothesis test to determine if a linear relationship exists between the transformed variables. The null hypothesis is that the slope (Beta1) is zero (no relationship), and the alternative hypothesis is that the slope is not zero.

```
summary(model)$coefficients[2,4]
```

```
## [1] 1.969498e-36
```

The p-value for the slope (Beta1) is 1.97*10^-36, which is extremely close to zero. Given this very low p-value, we reject the null hypothesis that the slope is zero. This suggests that there is a statistically significant linear relationship between the transformed variables.

6. 99% Confidence Interval

The 99% confidence interval for the expected Fertility for a region with a PPgdp of 20,000 US dollars in 2001.

```
new_data <- data.frame(log_PPgdp=log(20000))
predict(model, newdata=new_data, interval="confidence", level=0.99)
```

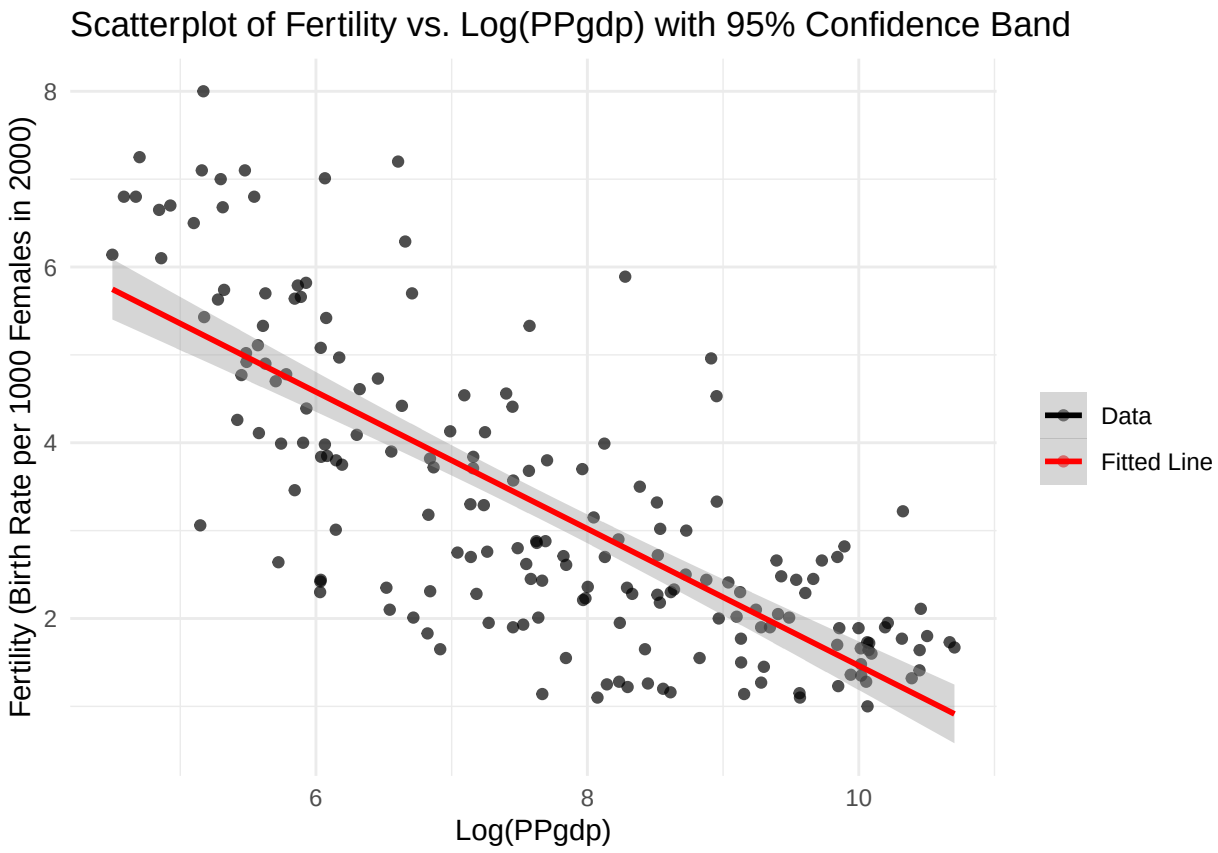
```
##        fit      lwr      upr
## 1 1.537967 1.183211 1.892722
```

7. 95% Confidence Band

```
model <- lm(Fertility ~ log(PPgdp), data=data)

# Plot using ggplot2
ggplot(data, aes(x=log(PPgdp), y=Fertility)) +
  geom_point(aes(color="Data"), alpha=0.7) +
  geom_smooth(method="lm", se=TRUE, aes(color="Fitted Line"), level=0.95) +
  labs(title="Scatterplot of Fertility vs. Log(PPgdp) with 95% Confidence Band",
       x="Log(PPgdp)", y="Fertility (Birth Rate per 1000 Females in 2000)") +
  theme_minimal() +
  scale_color_manual(name="", values=c("Data"="black", "Fitted Line"="red"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



8. 99% Prediction Interval

```r
new_data <- data.frame(PPgdp=25000)
predicted_fertility_25000 <- predict(model, newdata=data.frame(PPgdp=log(25000)), interval="prediction"
predicted_fertility_25000
```

```
##        fit      lwr      upr
## 1 7.448369 4.490107 10.40663
```

For a region with a PPgdp of 25,000 US dollars in 2018, the linear regression model predicts a Fertility value of approximately 7.45. Based on the 99% prediction interval, we are 99% confident that the Fertility for such a region would fall within the interval [4.49,10.41].

9 Concerns on Hypothesis Testing and Inferences

There was a slight pattern in the residuals, suggesting potential non-linearity or other systematic effects not captured by our model. There was also some deviation from the 45-degree line, especially at the tails, suggesting potential non-normality of residuals.The model might benefit from considering additional variables, interactions, or a different functional form to better capture the underlying relationship. The assumption of normally distributed residuals, which is foundational for linear regression, may be violated. This can affect the validity of hypothesis tests and confidence/prediction intervals.The potential heteroscedasticity suggests that the model's efficiency might be an issue.