Simulations to Evaluate Model Misspecification

Yuanrong Liu

2024-09-18

Packages Loaded

```
library(ggplot2)
```

Introduction

- We are tasked with evaluating how violations of the assumption of conditional normality affect the `bias`, root mean squared error (`RMSE`), `power`, and `coverage` of linear regression models.
- Specifically, we will generate data where the errors are not normally distributed by sampling from a t-distribution with varying degrees of freedom (`df`).
- We will assess the effect of different sample sizes (`n = 10, 20, 100`) and degrees of freedom (`df = seq(2, 50, by = 6)`) on model performance.
- We will also evaluate the power of the **Shapiro-Wilk** test for detecting non-normality.

Simulating Data with t-distribution

```
# simulate data for a linear model with **t-distributed** errors
sim_fun_t <- function(n = 100,
                      slope = 1,
                      sd = 1,
                      intercept = 0,
                      df = 10) {
  x <- runif(n)
  errors <- sd * rt(n, df = df)
  y <- intercept + slope * x + errors
  data.frame(x, y)
}
```

- This function generates `n` data points with a linear relationship between `x` and `y`, but with errors sampled from a **t-distribution** rather than a normal distribution. The degrees of freedom `df` control the extent of deviation from normality.

Evaluating Bias, RMSE, Power, and Coverage

- We define a function `run_simulation()` to run a specified number of simulations (`n_sim`) and calculate the `bias`, `RMSE`, `power`, and `coverage` of the linear regression mode

```r
run_simulation <- function(n = 100,
                           true_slope = 1,
                           sd = 1,
                           intercept = 0,
                           df = 10,
                           alpha = 0.05,
                           n_sim = 1000) {
  slopes <- numeric(n_sim)
  p_values <- numeric(n_sim)
  coverage <- numeric(n_sim)

  for (i in 1:n_sim) {
    # Simulate data with t-distributed errors
    data <- sim_fun_t(n, slope = true_slope, sd = sd, intercept = intercept, df = df)

    # Fit a linear regression model
    m <- lm(y ~ x, data = data)

    # Extract the estimated slope, p-value, and confidence interval for the slope
    slopes[i] <- coef(m)[2]
    p_values[i] <- coef(summary(m))[2, "Pr(>|t|)"]
    conf_int <- confint(m)[2, ]

    # Check whether the confidence interval contains the true slope (for coverage calculation)
    coverage[i] <- (conf_int[1] < true_slope & true_slope < conf_int[2])
  }

  # # Compute the bias (average difference between estimated and true slope)
  bias <- mean(slopes - true_slope)

  # Compute the root mean squared error (RMSE) of the slope estimates
  rmse <- sqrt(mean((slopes - true_slope)^2))

  # Compute the power (proportion of times p-value is less than alpha)
  power <- mean(p_values < alpha)

  # Compute the coverage (proportion of times the confidence interval contains the true slope)
  coverage_prob <- mean(coverage)

  # Return a data frame with the results for the current simulation
  data.frame(df = df, n = n, bias = bias, rmse = rmse, power = power, coverage = coverage_prob)
}
```

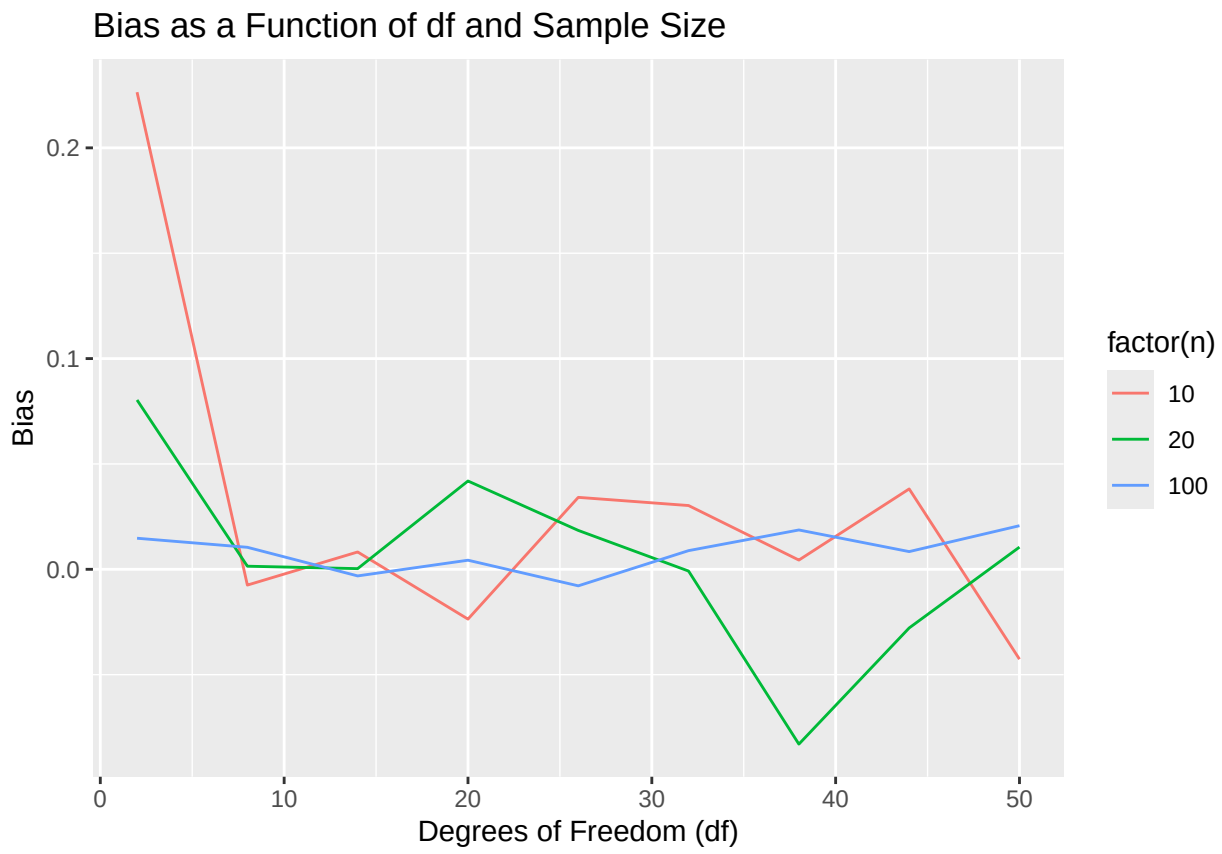
Simulations Running

```r
df_values <- seq(2, 50, by = 6)
n_values <- c(10, 20, 100)

# Run the simulation for all combinations of df and n
results <- do.call(rbind, lapply(n_values, function(n) {
  do.call(rbind, lapply(df_values, function(df) {
```

```
    run_simulation(n = n, df = df)
  }))
}))
```

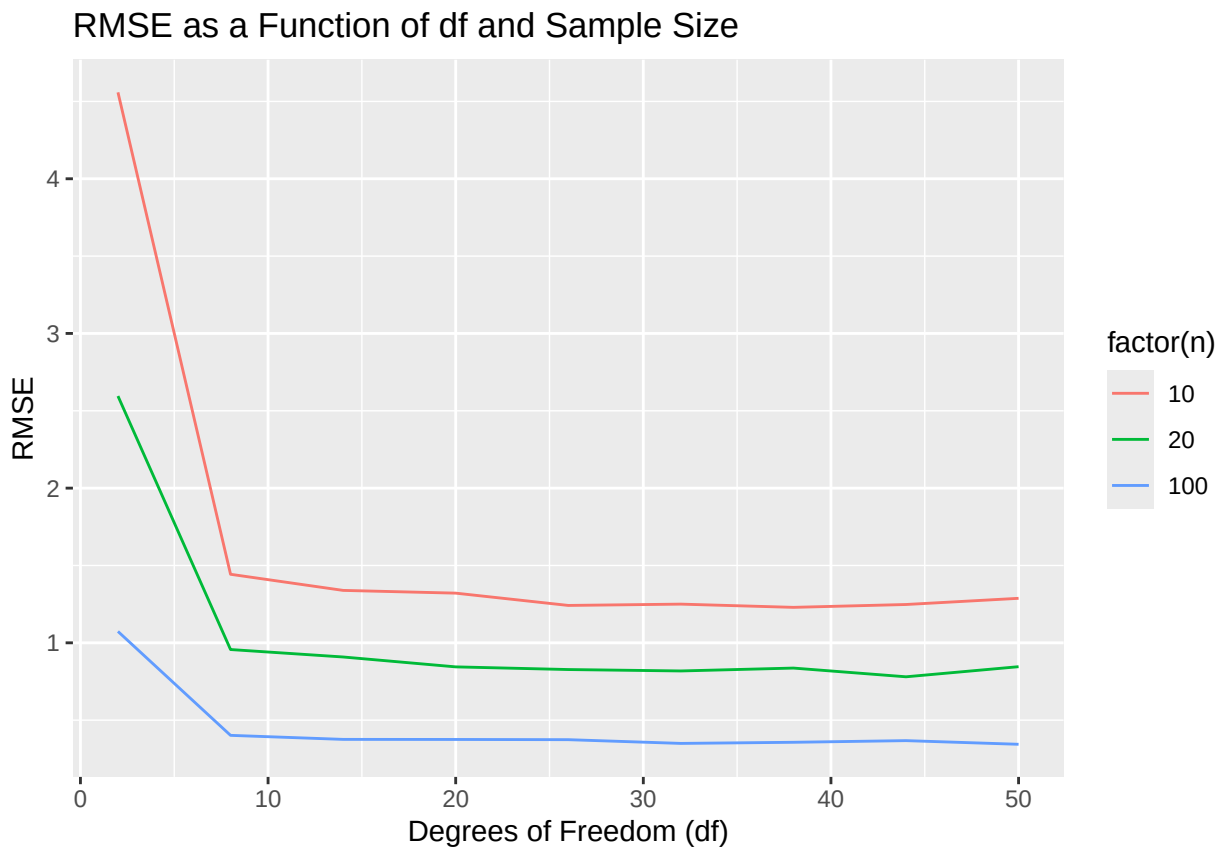
Visualization

```
# Bias as a Function of Degrees of Freedom and Sample Size
ggplot(results, aes(x = df, y = bias, color = factor(n))) +
  geom_line() +
  labs(title = "Bias as a Function of df and Sample Size", x = "Degrees of Freedom (df)", y = "Bias")
```



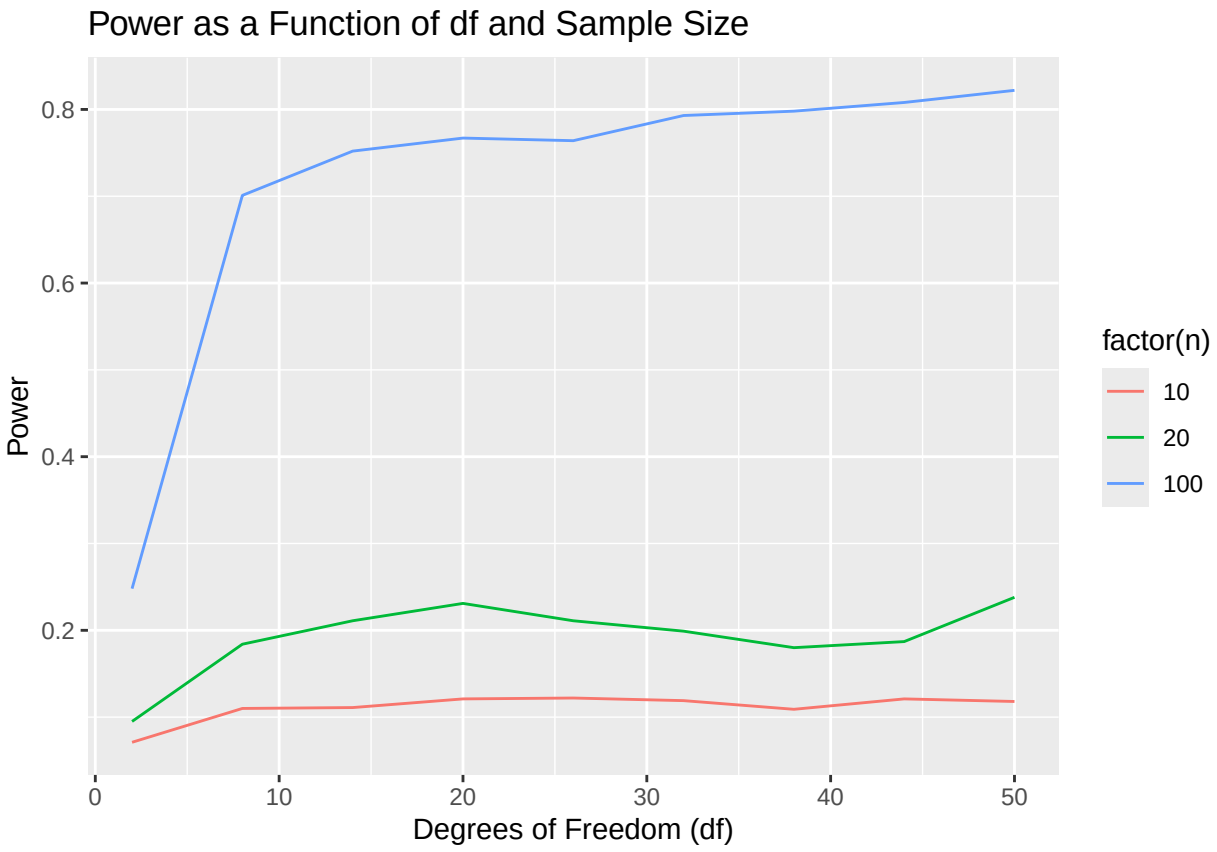
- The bias tends to fluctuate more for small sample sizes (`n = 10`, red line), and the bias is notably higher for very small `df` values (less than `10`).
- As the sample size increases (`n = 100`, blue line), the bias becomes more stable and closer to zero, indicating more reliable slope estimates.

```
# RMSE as a Function of Degrees of Freedom and Sample Size
ggplot(results, aes(x = df, y = rmse, color = factor(n))) +
  geom_line() +
  labs(title = "RMSE as a Function of df and Sample Size", x = "Degrees of Freedom (df)", y = "RMSE")
```



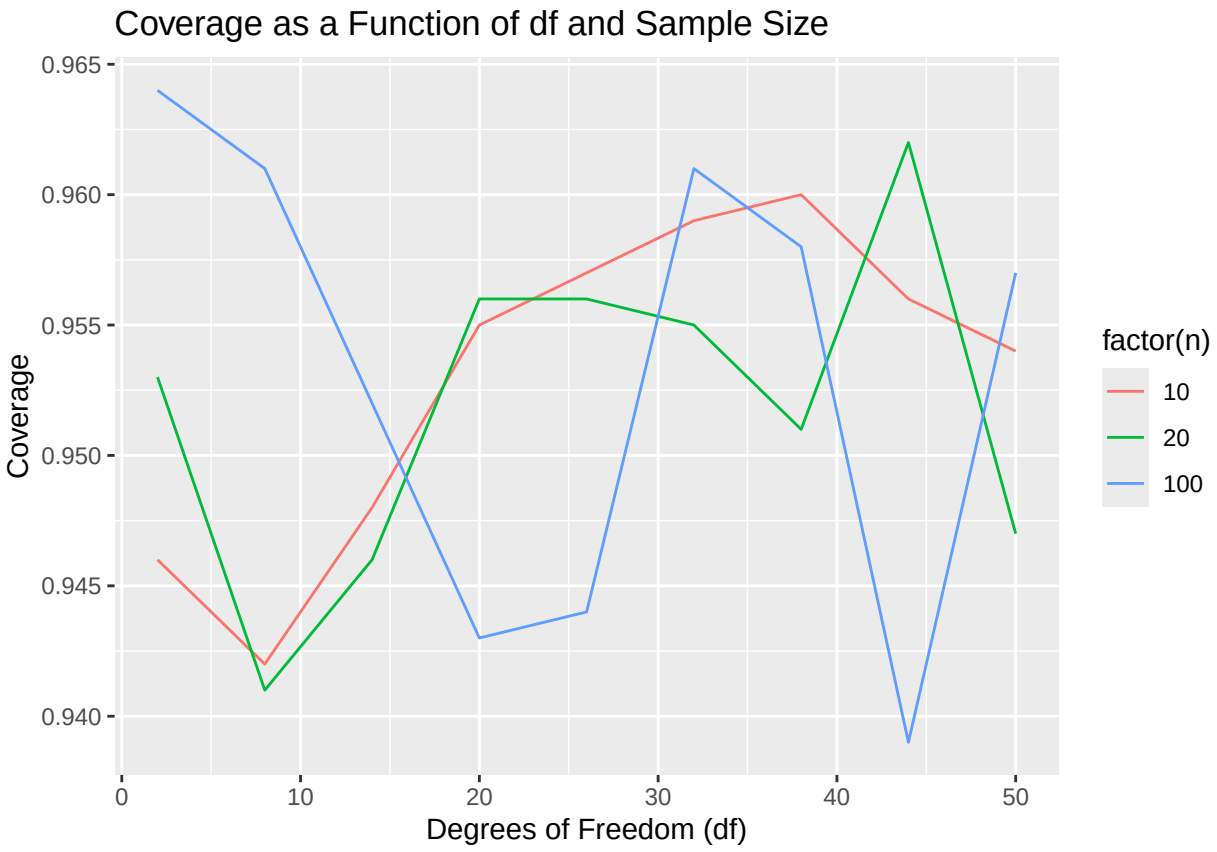
- RMSE (root mean squared error) is significantly higher for small sample sizes (`n = 10`) and low degrees of freedom (`df < 10`). This is likely due to the larger variability in the estimates when the errors are more heavily tailed (small df).
- As df increases, RMSE stabilizes for all sample sizes, though RMSE remains lower for larger sample sizes (`n = 100`).

```
# Power as a Function of Degrees of Freedom and Sample Size
ggplot(results, aes(x = df, y = power, color = factor(n))) +
  geom_line() +
  labs(title = "Power as a Function of df and Sample Size", x = "Degrees of Freedom (df)", y = "Power")
```



- Power is significantly lower for smaller sample sizes (`n = 10` and `20`) and remains fairly low across all df values.
- For larger sample sizes (`n = 100`), power increases substantially, especially as df increases, approaching 0.8, which indicates high ability to detect a true effect.

```
# Coverage as a Function of Degrees of Freedom and Sample Size
ggplot(results, aes(x = df, y = coverage, color = factor(n))) +
  geom_line() +
  labs(title = "Coverage as a Function of df and Sample Size", x = "Degrees of Freedom (df)", y = "Cove
```



- Coverage fluctuates around 95% (from 93.5% to 96.5%) for all sample sizes and df values, with some variability.

Shapiro-Wilk Test

```
run_simulation_with_shapiro <- function(n = 100,
                                        true_slope = 1,
                                        sd = 1,
                                        intercept = 0,
                                        df = 10,
                                        alpha = 0.05,
                                        n_sim = 1000) {
    slopes <- numeric(n_sim)
    p_values <- numeric(n_sim)
    coverage <- numeric(n_sim)
    shapiro_p_values <- numeric(n_sim)

    for (i in 1:n_sim) {
        data <- sim_fun_t(n,
                          slope = true_slope,
                          sd = sd,
                          intercept = intercept,
                          df = df)
        m <- lm(y ~ x, data = data)
```

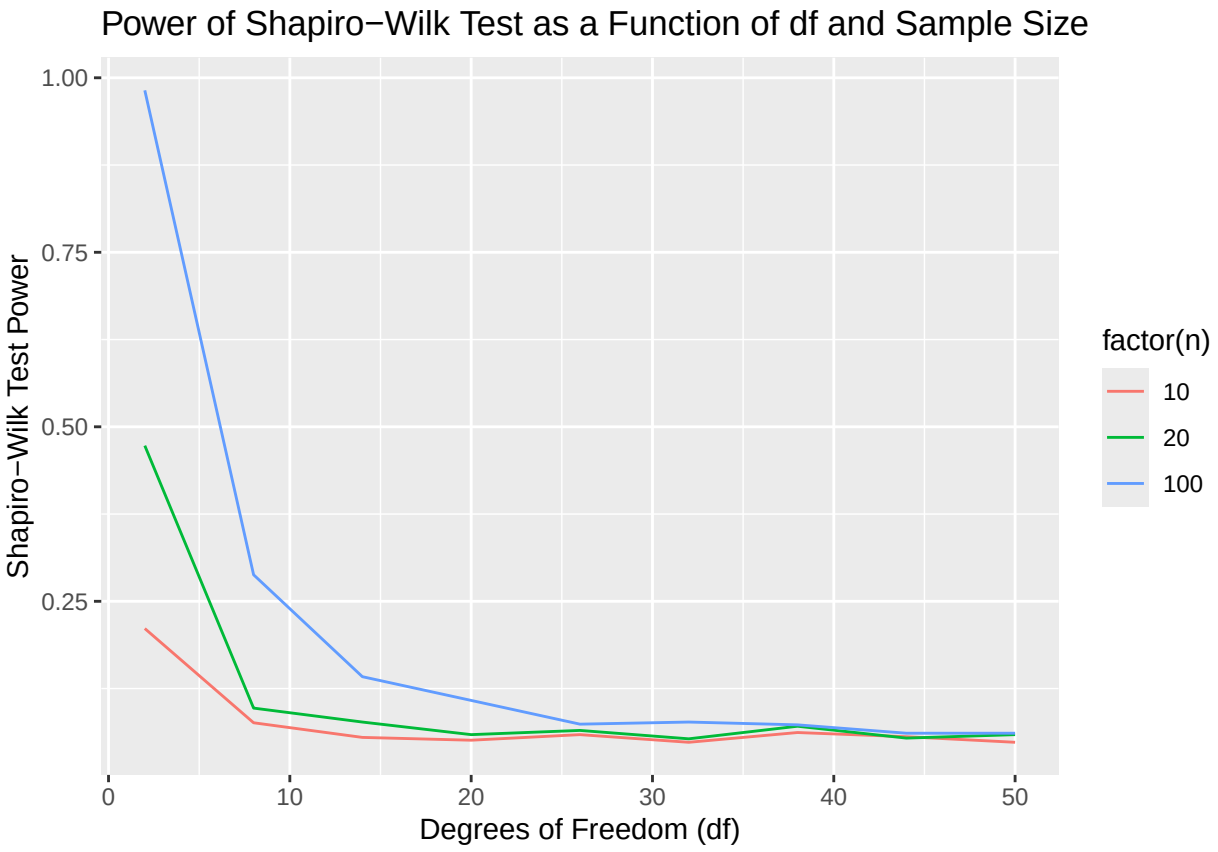
```
        slopes[i] <- coef(m)[2]
        p_values[i] <- coef(summary(m))[2, "Pr(>|t|)"]
        conf_int <- confint(m)[2, ]
        coverage[i] <- (conf_int[1] < true_slope & true_slope < conf_int[2])

        # Shapiro-Wilk test for normality of residuals
        shapiro_p_values[i] <- shapiro.test(resid(m))$p.value
    }

    bias <- mean(slopes - true_slope)
    rmse <- sqrt(mean((slopes - true_slope)^2))
    power <- mean(p_values < alpha)
    coverage_prob <- mean(coverage)
    shapiro_power <- mean(shapiro_p_values < alpha)

    data.frame(df = df, n = n, bias = bias, rmse = rmse, power = power, coverage = coverage_prob, shapi
}
```

Simulation

```
results_SW <- do.call(rbind, lapply(n_values, function(n) {
  do.call(rbind, lapply(df_values, function(df) {
    run_simulation_with_shapiro(n = n, df = df)
  }))
}))
```

Visualization

```
# Plotting the power of the Shapiro-Wilk test
ggplot(results_SW, aes(x = df,
                       y = shapiro_power,
                       color = factor(n))) +
  geom_line() +
  labs(title = "Power of Shapiro-Wilk Test as a Function of df and Sample Size",
       x = "Degrees of Freedom (df)", y = "Shapiro-Wilk Test Power")
```



High Power for Low df (Heavy-Tailed Distributions):

- For small degrees of freedom (`df < 10`), the Shapiro-Wilk test shows high power, especially for larger sample sizes (`n = 100`, blue line). This is because when the degrees of freedom are small, the t-distribution has heavier tails, making the residuals far from normal, which the Shapiro-Wilk test is detecting well.
- For small sample sizes (`n = 10`, red line), the power of the Shapiro-Wilk test is much lower, but still manages to capture some non-normality.

Decreasing Power as df Increases:

- As the degrees of freedom increase, the t-distribution becomes more similar to the normal distribution. Correspondingly, the power of the Shapiro-Wilk test decreases.
- Around `df > 20`, the power drops to nearly zero across all sample sizes. This is because the t-distribution starts resembling the normal distribution, making it harder for the Shapiro-Wilk test to reject the null hypothesis of normality.

Effect of Sample Size:

- Larger sample sizes (`n = 100`, blue line) consistently show better power to detect non-normality across all degrees of freedom. Even for `df` values between 5 and 10, the power is quite high (approaching 1.0), showing that the test performs better with more data points.
- Smaller sample sizes (`n = 10`, red line) struggle more to detect non-normality, particularly when `df` is moderately large (between `10` and `20`). This is because with smaller sample sizes, the residuals might not exhibit strong evidence of non-normality, making the test less sensitive.

Conclusion

- Shapiro-Wilk Test Sensitivity: The Shapiro-Wilk test is very sensitive to deviations from normality when the degrees of freedom are small, especially for larger sample sizes. As the degrees of freedom increase (i.e., the distribution becomes more normal), the power of the test drops significantly.
- Sample Size Impact: Larger sample sizes improve the test's ability to detect non-normality. The Shapiro-Wilk test performs much better with `n = 100` compared to smaller sample sizes (`n = 10` or `n = 20`).

Final Conclusion

Impact of Non-Normality:

- For smaller degrees of freedom (heavier-tailed distributions), the linear model shows higher bias and RMSE, especially with small sample sizes.
- Power of the linear model improves with larger sample sizes, but is affected by non-normal errors when sample sizes are small.

Shapiro-Wilk Test:

- The Shapiro-Wilk test is effective at detecting non-normality when the degrees of freedom are small (indicating heavy tails).
- Its power decreases as the degrees of freedom increase (data becomes more normal).
- The test is more powerful with larger sample sizes, but struggles with smaller samples.

Reference

OpenAI, ChatGPT (2024). Assistance with R programming and output analysis. Accessed on September 18, 2024.