The Illusion of Association: Spurious Correlation in High-Dimensional Data

An Investigation into the Probability of Observing High Correlation Strengths (|r|) Between Independent Variables via Monte Carlo Simulation

Jason Huang

2025-11-27

Table of contents

1. Introduction and Problem Statement

This report addresses a statistical question arising in high-dimensional data analysis (Data Dredging): **When numerous truly independent variables are tested against a single outcome variable, what is the probability of accidentally finding a strong, statistically significant correlation?**

In our simulation, all variables are generated to be **Independent and Identically Distributed (i.i.d.)** $\mathcal{N}(0,1)$, meaning the true population correlation (ρ) between any X_i and Y is precisely zero. The question is: Does this guarantee that none of the sample correlations (cor(Y, X_i)) will be high, or will high correlations inevitably appear simply due to chance in a large pool of $N = 10{,}000$ variables? This phenomenon is the core of **Spurious Correlation**.

> **i** Knowledge Expansion: Spurious Correlation, Strength, and Type I Error
>
> **Spurious Correlation (偽相關)** is the observation of a statistically significant sample correlation ($|r| > 0$) where the true population correlation (ρ) is zero. This constitutes a **Type I Error** (rejecting the true null hypothesis $\rho = 0$).
> **Strength** of the observed correlation is defined by the **absolute value** of the sample correlation coefficient, $|r|$. We focus on strength, as a correlation of -0.5 is as strong as 0.5.

The problem requires a **Monte Carlo simulation** to empirically estimate the sampling distribution of the sample correlation coefficient (r) under the null hypothesis ($\rho = 0$).

Problem Specification (Focus on Strength): Simulate a random variable $Y \sim \mathcal{N}(0,1)$ with $n = 50$ values. Generate $N = 10{,}000$ independent random variables $X_1, \dots, X_{10{,}000}$, each $\sim \mathcal{N}(0,1)$ with $n = 50$ values and independent of Y. Calculate the correlation coefficient cor(Y, X_i) for all i. **Using the calculated absolute correlations** $|r|$, report the number of variables where the correlation strength $|\textbf{cor}(Y, X_i)| > 0.5$ and $|\textbf{cor}(Y, X_i)| > 0.4$.

2. Methodology and Simulation

2.1 Simulation Setup

> **i** Knowledge Expansion: Monte Carlo Simulation Approach
>
> **Monte Carlo Simulation** uses repeated random sampling to estimate probabilities. By running $N = 10,000$ independent trials, we generate a distribution of correlation coefficients where the true underlying relationship is always zero. This allows us to empirically estimate the probability (or Type I Error rate) of observing correlations of a certain **strength** (e.g., $|r| > 0.4$) by random chance alone.

The simulation is performed using R. The core challenge is that while the population correlation is $\rho = 0$ (due to variable independence), the **sample correlation (r)** is expected to deviate from zero due to sampling variation, leading to the observed spurious correlations.

```r
# ----------------------------
# Simulation parameters & data
# ----------------------------
N_obs <- 50       # Sample Size (n)
N_sim <- 10000    # Total Simulations (N)
set.seed(42)      # For reproducibility

# Generate Y and X matrix
Y <- rnorm(N_obs, mean = 0, sd = 1)
X <- matrix(rnorm(N_obs * N_sim, mean = 0, sd = 1), nrow = N_obs, ncol = N_sim)

# Calculate correlation coefficients
cor_vector <- as.numeric(cor(Y, X)[1, ])
abs_cor_vector <- abs(cor_vector)

# Critical correlation for n=50 (alpha = 0.05)
N_df <- N_obs - 2
t_crit <- qt(0.975, df = N_df)
r_crit <- sqrt(t_crit^2 / (t_crit^2 + N_df))

# Tables: results_table, desc_table, count_data_abs
r_crit_label <- paste0(round(r_crit, 3), " ($\\alpha=0.05$)")

results_table <- data.frame(
  Metric = c("Total Simulations (N)", "Sample Size (n)",
             "Critical Correlation (r_crit)"),
  Value = c(N_sim, N_obs, r_crit_label),
  stringsAsFactors = FALSE
)

descriptive_stats <- c(
  "Mean of r" = mean(cor_vector),
  "SD of r" = sd(cor_vector),
  "Max Observed |r|" = max(abs_cor_vector),
  "Mean of |r|" = mean(abs_cor_vector),
  "|r| 25th Pct." = quantile(abs_cor_vector, 0.25),
```

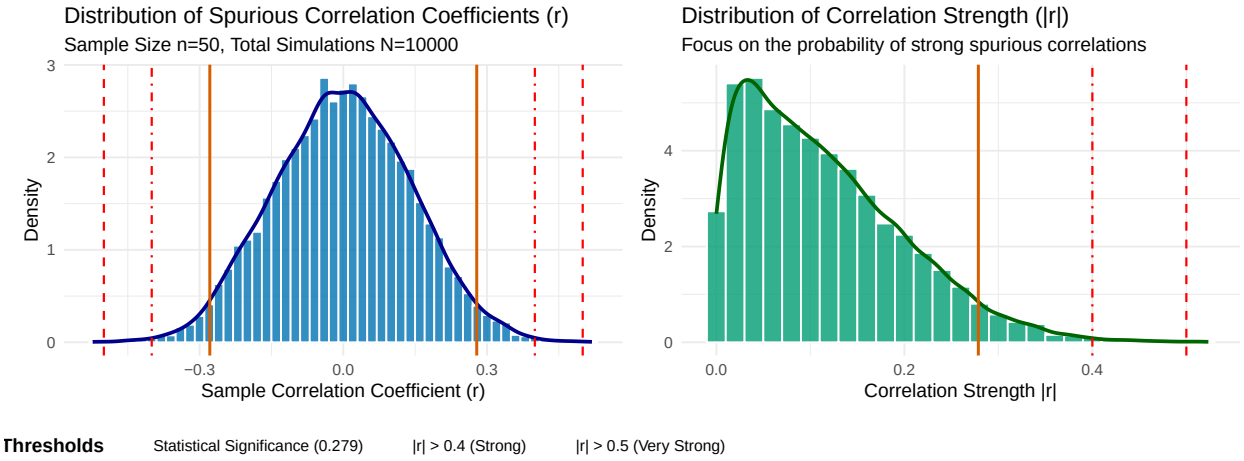
```
  "|r| Median" = median(abs_cor_vector),
  "|r| 75th Pct." = quantile(abs_cor_vector, 0.75),
  "|r| IQR" = IQR(abs_cor_vector)
)

desc_table <- data.frame(
  Statistic = names(descriptive_stats),
  Value = as.numeric(round(descriptive_stats, 4)),
  stringsAsFactors = FALSE
)

thresholds <- c(0.5, 0.4, 0.3, 0.2, 0.1, 0.0)
count_vals <- sapply(thresholds, function(t) sum(abs_cor_vector > t))
count_data_abs <- data.frame(
  Threshold = paste0("|cor(Y, X_i)| > ", thresholds),
  Count = as.integer(count_vals),
  stringsAsFactors = FALSE
)
count_data_abs$Count_Percentage <- round(count_data_abs$Count / N_sim * 100, 2)

# Variables for narrative (inline R usage later)
count_sig_abs <- sum(abs_cor_vector > r_crit)
count_gt_0_5 <- sum(abs_cor_vector > 0.5)
count_gt_0_4_and_le_0_5 <- sum(abs_cor_vector > 0.4 & abs_cor_vector <= 0.5)
```

```
# ----------------------------
# Visualization (two side-by-side plots)
# ----------------------------

# color & linetype
LINE_COLORS <- c("r_crit" = "#D55E00", "Threshold 0.5" = "red",
                 "Threshold 0.4" = "red")
LINE_LINETYPE <- c("r_crit" = "solid", "Threshold 0.5" = "dashed",
                   "Threshold 0.4" = "dotdash")

# Plot 1: distribution of r
plot_data <- data.frame(cor_vector = cor_vector)

p1 <- ggplot(plot_data, aes(x = cor_vector)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.02,
```

```
                fill = "#0072B2", color = "white", alpha = 0.8) +
  geom_density(linewidth = 1, color = "darkblue") +
  # Solid line: Statistical significance r_crit
  geom_vline(xintercept = c(r_crit, -r_crit), color = LINE_COLORS["r_crit"],
             linetype = LINE_LINETYPE["r_crit"], linewidth = 0.8) +
  # Dashed line: High-intensity threshold
  geom_vline(xintercept = c(0.5, -0.5), color = LINE_COLORS["Threshold 0.5"],
             linetype = LINE_LINETYPE["Threshold 0.5"], linewidth = 0.6) +
  geom_vline(xintercept = c(0.4, -0.4), color = LINE_COLORS["Threshold 0.4"],
             linetype = LINE_LINETYPE["Threshold 0.4"], linewidth = 0.6) +

  geom_point(aes(y = 0, color = "r_crit"), shape = NA) +
  geom_point(aes(y = 0, color = "Threshold 0.5"), shape = NA) +
  geom_point(aes(y = 0, color = "Threshold 0.4"), shape = NA) +
  # ============================================================

  labs(title = "Distribution of Spurious Correlation Coefficients (r)",
       subtitle = paste0("Sample Size n=", N_obs, ", Total Simulations N=", N_sim),
       x = "Sample Correlation Coefficient (r)",
       y = "Density") +

  scale_color_manual(
    name = "Key Thresholds",
    values = LINE_COLORS,
    labels = c(
      "r_crit" = paste0("Statistical Significance (", round(r_crit, 3), ")"),
      "Threshold 0.5" = "|r| > 0.5 (Very Strong)",
      "Threshold 0.4" = "|r| > 0.4 (Strong)"
    ),
    guide = guide_legend(override.aes = list(
      linetype = LINE_LINETYPE,
      linewidth = 1,
      shape = NA
    ))
  ) +
  theme_minimal(base_size = 11) +
  theme(legend.position = "bottom", legend.title = element_text(face = "bold"))

# Plot 2: distribution of |r|
```

```r
p2 <- ggplot(data.frame(abs_cor_vector = abs_cor_vector), aes(x = abs_cor_vector)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.02,
                 fill = "#009E73", color = "white", alpha = 0.8) +
  geom_density(linewidth = 1, color = "darkgreen") +

  geom_vline(xintercept = r_crit, color = LINE_COLORS["r_crit"],
             linetype = LINE_LINETYPE["r_crit"], linewidth = 0.8) +
  geom_vline(xintercept = 0.5, color = LINE_COLORS["Threshold 0.5"],
             linetype = LINE_LINETYPE["Threshold 0.5"], linewidth = 0.6) +
  geom_vline(xintercept = 0.4, color = LINE_COLORS["Threshold 0.4"],
             linetype = LINE_LINETYPE["Threshold 0.4"], linewidth = 0.6) +

  geom_point(aes(y = 0, color = "r_crit"), shape = NA) +
  geom_point(aes(y = 0, color = "Threshold 0.5"), shape = NA) +
  geom_point(aes(y = 0, color = "Threshold 0.4"), shape = NA) +
  # ============================================================

  labs(title = "Distribution of Correlation Strength (|r|)",
       subtitle = "Focus on the probability of strong spurious correlations",
       x = "Correlation Strength |r|",
       y = "Density") +

  # p2_scale_color_manual
  scale_color_manual(
    name = "Key Thresholds",
    values = LINE_COLORS,
    labels = c(
      "r_crit" = paste0("Statistical Significance (", round(r_crit, 3), ")"),
      "Threshold 0.5" = "|r| > 0.5 (Very Strong)",
      "Threshold 0.4" = "|r| > 0.4 (Strong)"
    ),
    guide = guide_legend(override.aes = list(
      linetype = LINE_LINETYPE,
      linewidth = 1,
      shape = NA
    ))
  ) +
  theme_minimal(base_size = 11) +
  theme(legend.position = "none")
```

```
# Combine and print
combined <- p1 + p2 + plot_layout(ncol = 2)
print(combined)
```



The charts clearly demonstrate the distribution of spurious correlation coefficients (r) from $N = 10,000$ simulations.

The left panel (Distribution of r) shows a symmetric, zero-centered distribution, confirming that most spurious correlations are weak, with the highest density near $|r| = 0$. The rapid decline of the density curve indicates that the probability of observing a high-strength correlation ($|r|$) is low. Values exceeding the statistical significance threshold, r_{crit}, constitute the Type I Error rate, confirming that a small fraction of correlations will be falsely identified as significant by chance alone.

The right panel (Distribution of $|r|$) isolates correlation strength. It emphasizes the rarity of strong correlations: only a minute number of trials result in correlations exceeding the high thresholds of $|r| > 0.4$ or $|r| > 0.5$. This confirms that while high-strength spurious correlations are rare events, they are an inevitable occurrence when performing a large number of independent tests.

2.2 Numerical Results and Professional Table Formatting

The table below provides the numerical answer to the problem specification.

```
# LaTeX Helper
sanitize_latex <- function(x) {
  if (is.null(x)) return(x)
  x <- as.character(x)
  x <- gsub("\\", "\\textbackslash{}", x)
  x <- gsub("_", "\\_", x)
  x <- gsub("%", "\\%", x)
```

```
  x <- gsub("&", "\\\\&", x)
  x <- gsub("#", "\\\\#", x)
  x <- gsub("\\{", "\\\\{", x)
  x <- gsub("\\}", "\\\\}", x)
  x <- gsub("\\$", "\\\\$", x)
  x <- gsub("\\|", "\\\\|", x)
  return(x)
}

# ================ Table 1 Output simulation parameters ================
results_table_k <- results_table
results_table_k$Metric <- sanitize_latex(results_table_k$Metric)
row_idx1 <- which(grepl("Critical Correlation", results_table_k$Metric, fixed = TRUE))
kable(results_table_k, format="latex", booktabs=TRUE, escape=FALSE,
      col.names=c("Parameter","Value"), caption="Simulation Parameters
      and Statistical Critical Value") %>%
  kable_styling(latex_options=c("hold_position")) %>%
  { if(length(row_idx1)>0) row_spec(., row_idx1, background="#E69F00") else . } %>%
  print()
```

Table 1: Simulation Parameters and Statistical Critical Value

Parameter	Value
Total Simulations (N)	10000
Sample Size (n)	50
Critical Correlation (r_crit)	0.279 ($\alpha = 0.05$)

```
# ================ Table 2 Output descriptive statistics ================
desc_table_k <- desc_table
desc_table_k$Statistic <- sanitize_latex(desc_table_k$Statistic)
desc_table_k$Value <- as.character(desc_table_k$Value)
row_idx2 <- which(grepl("Max Observed |r|", desc_table_k$Statistic, fixed=TRUE))
kable(desc_table_k, format="latex", booktabs=TRUE, escape=FALSE,
      col.names=c("Statistic","Result"), caption="Descriptive Statistics of
      Correlation Coefficients (r and |r|)") %>%
  kable_styling(latex_options=c("hold_position","scale_down")) %>%
  { if(length(row_idx2)>0) row_spec(., row_idx2, background="#F0E442") else . } %>%
  print()
```

```
# ================ Table 3 Output all absolute threshold counts ================
count_data_abs_k <- count_data_abs
```

Table 2: Descriptive Statistics of Correlation Coefficients (r and |r|)

Statistic	Result
Mean of r	-5e-04
SD of r	0.1431
Max Observed ||r||	0.5234
Mean of ||r||	0.1149
||r|| 25th Pct..25%	0.0454
||r|| Median	0.0984
||r|| 75th Pct..75%	0.1671
||r|| IQR	0.1218

```
count_data_abs_k$Threshold <- sanitize_latex(count_data_abs_k$Threshold)
count_data_abs_k$Probability <- sprintf("%.2f\\\\%%", count_data_abs_k$Count_Percentage)
count_data_abs_k_print <- count_data_abs_k[, c("Threshold","Count","Probability")]
kable(count_data_abs_k_print, format="latex", booktabs=TRUE, escape=FALSE,
      col.names=c("Threshold","Count","Probability"), caption="Observed Spurious
      Correlation Strengths Across Multiple Thresholds (|r|)") %>%
  kable_styling(latex_options=c("hold_position","scale_down")) %>%
  print()
```

Table 3: Observed Spurious Correlation Strengths Across Multiple Thresholds (|r|)

Threshold	Count	Probability
||cor(Y, X_i)|| > 0.5	5	0.05
||cor(Y, X_i)|| > 0.4	39	0.39
||cor(Y, X_i)|| > 0.3	329	3.29
||cor(Y, X_i)|| > 0.2	1650	16.50
||cor(Y, X_i)|| > 0.1	4944	49.44
||cor(Y, X_i)|| > 0	10000	100.00

3. Discussion and Visual Analysis

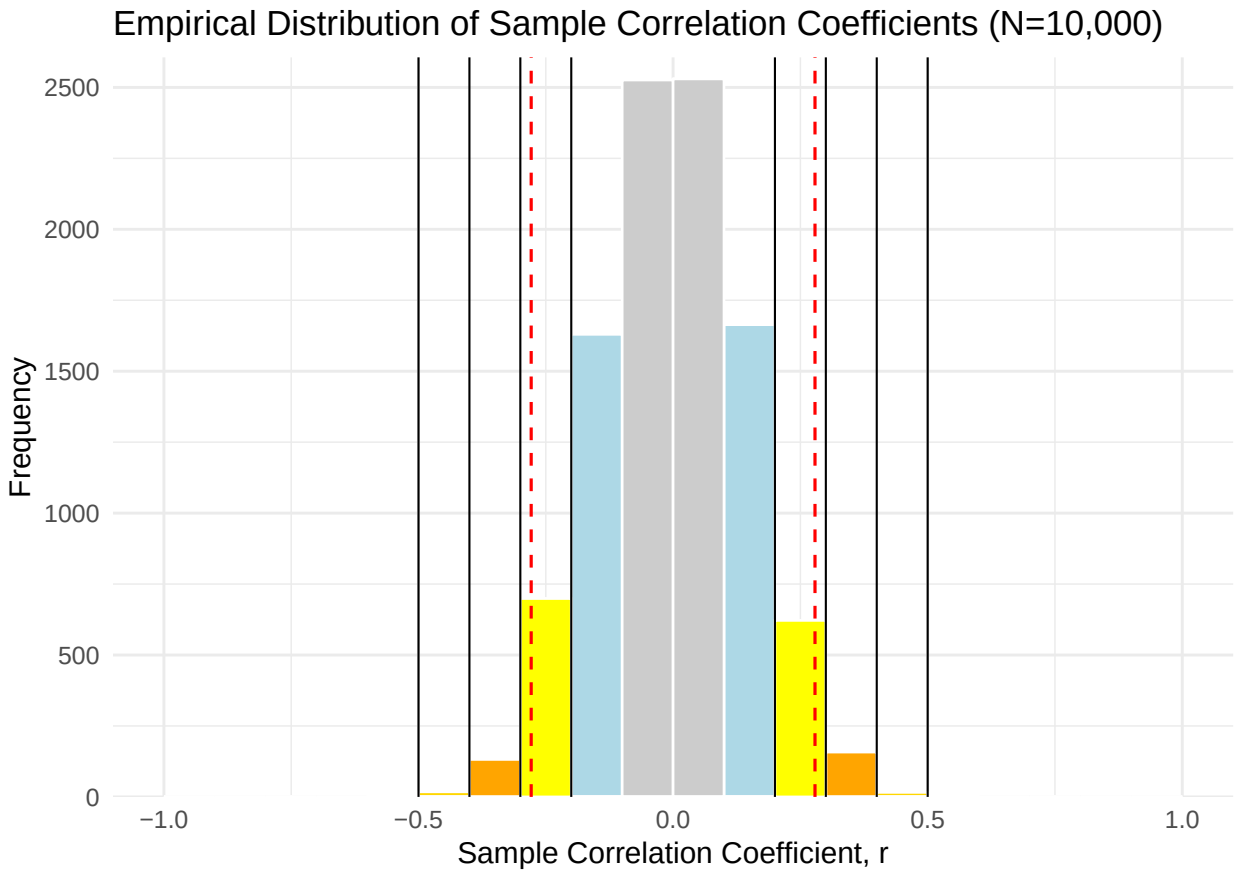
3.1 Empirical Distribution of the Correlation Coefficients

The histogram below displays the empirical distribution of the 10,000 sample correlation coefficients, visually confirming that the data distribution adheres to the expected pattern under the null hypothesis.

```
cor_df <- data.frame(Correlation = cor_vector)


breaks <- seq(-1, 1, by = 0.1)
```

```r
cor_df$bin <- cut(cor_df$Correlation, breaks = breaks, include.lowest = TRUE, right = TRUE)

cor_df$color <- "grey80" # 預設灰色
highlight_intervals <- c(0.1, 0.2, 0.3, 0.4, 0.5)
colors <- c("#ADD8E6", "yellow", "orange", "gold", "red") # 可依喜好修改

for(i in seq_along(highlight_intervals)) {
hi <- highlight_intervals[i]
cor_df$color[cor_df$Correlation >= hi & cor_df$Correlation < hi + 0.1] <- colors[i]
cor_df$color[cor_df$Correlation <= -hi & cor_df$Correlation > -hi - 0.1] <- colors[i]
}

hist_plot <- ggplot(cor_df, aes(x = Correlation)) +
geom_histogram(
aes(fill = color),
breaks = breaks,
color = "white",
show.legend = FALSE
) +
geom_vline(
xintercept = c(-r_crit, r_crit),
linetype = "dashed",
color = "red",
size = 0.6
) +
geom_vline(
xintercept = c(-0.5, -0.4, -0.3, -0.2, 0.2, 0.3, 0.4, 0.5),
linetype = "solid",
color = "black",
size = 0.4
) +
labs(
title = "Empirical Distribution of Sample Correlation Coefficients (N=10,000)",
x = expression("Sample Correlation Coefficient, " * r),
y = "Frequency"
) +
theme_minimal(base_size = 12) +
scale_fill_identity() +
scale_y_continuous(expand = expansion(mult = c(0, 0.03)))
```

```
print(hist_plot)
```



3.2 Interpretation

The analysis reveals the inherent risk of Type I error in high-dimensional testing:

- **Statistical Significance:** The total number of variables whose absolute correlation is statistically significant ($|r| > r_{\text{crit}}$) is 463.
- **High-Magnitude Spurious Correlation:** Despite generating variables with zero true relationship, we observed 5 variables (red region) with a strong correlation ($|r| > 0.5$). An additional 34 variables (gold region) had a moderate correlation ($0.4 < |r| \leq 0.5$).

This result demonstrates that finding strong correlations in a large pool of predictors is likely due to chance, necessitating rigorous validation and multiple-testing correction.