2425–SSS0004.R

r2963279

2025-01-02

```
# Load required libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(e1071)  # For skewness and kurtosis
library(cluster)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(dplyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(RColorBrewer)

# Read the CSV file
FakeNewsNet <- read_csv("FakeNewsNet.csv")
```

```
## Rows: 23196 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): title, news_url, source_domain
## dbl (2): tweet_num, real
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

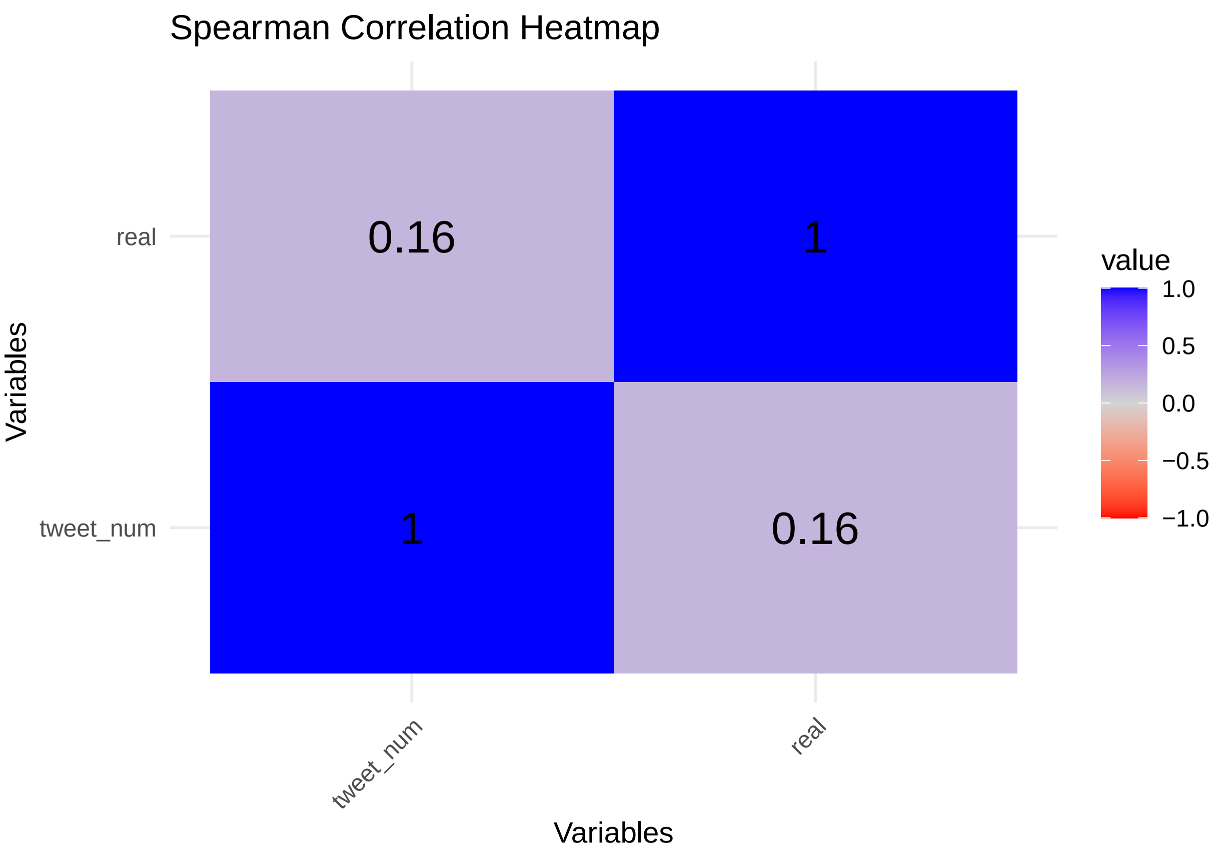
```r
# Check the column names
colnames(FakeNewsNet)
```

```
## [1] "title"         "news_url"      "source_domain" "tweet_num"
## [5] "real"
```

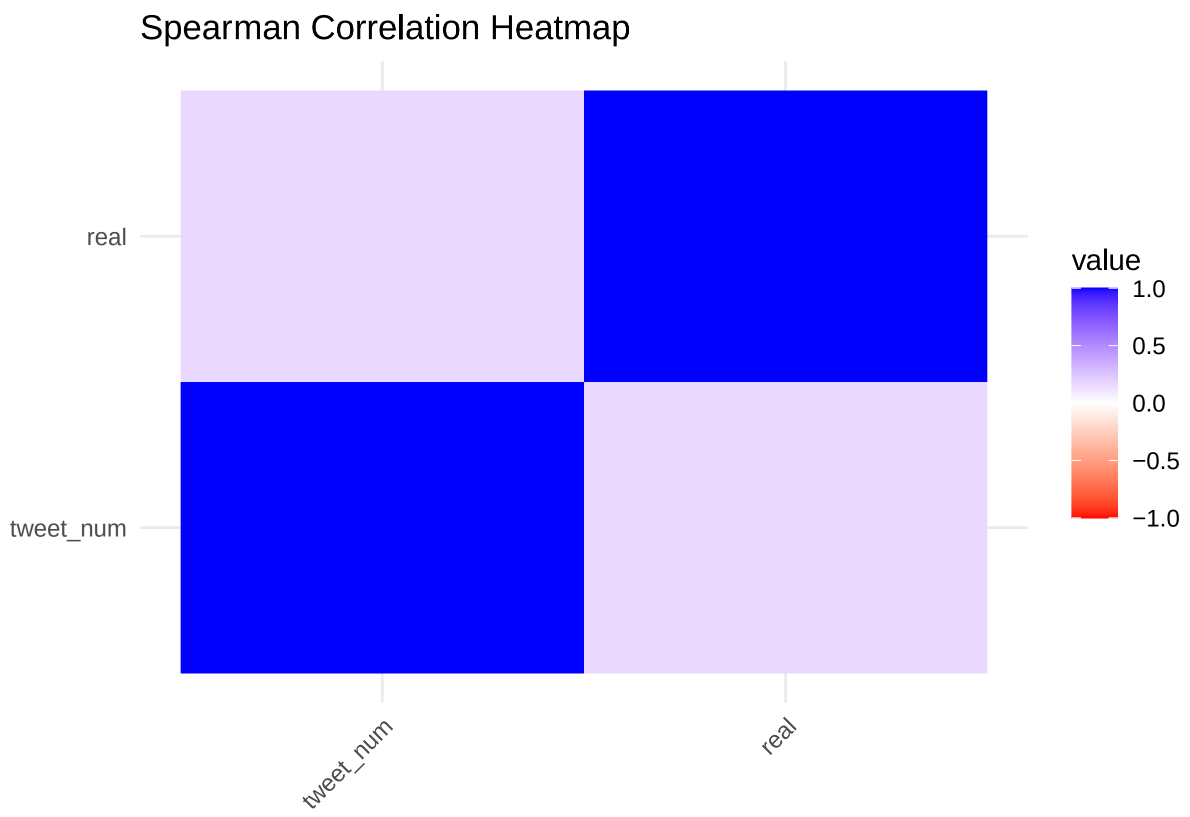
```r
# 1. Correlation

# Calculate Spearman correlation between tweet_num and real

spearman_correlation <- cor(FakeNewsNet$tweet_num, FakeNewsNet$real, method = "spearman")

# Print the Spearman correlation

cat("The Spearman correlation value is:", spearman_correlation, "\n")
```

```
## The Spearman correlation value is: 0.1625129
```

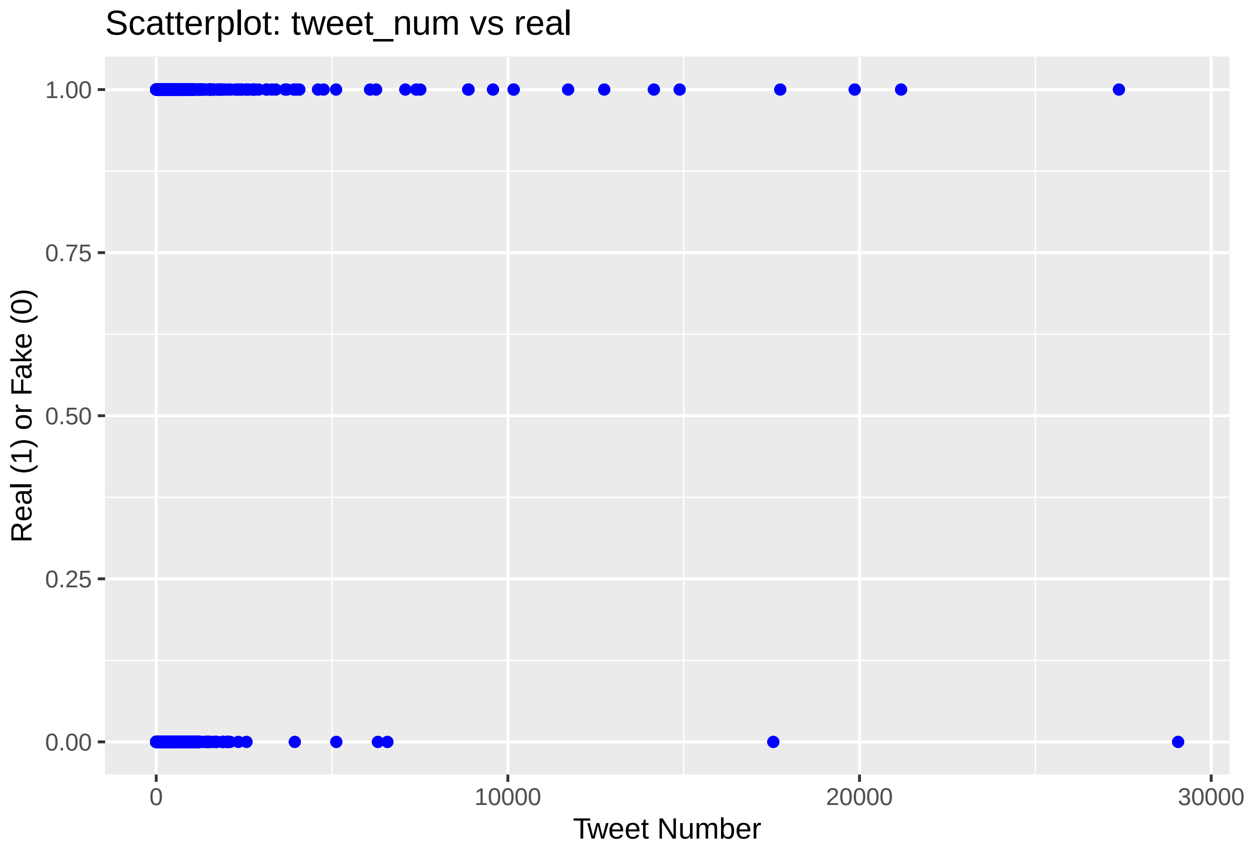
```r
spearman_matrix <- matrix(c(1, spearman_correlation, spearman_correlation, 1),
                          nrow = 2, ncol = 2,
                          dimnames = list(c("tweet_num", "real"), c("tweet_num", "real")))

# Melt the matrix to long format for ggplot
spearman_melted <- melt(spearman_matrix)

# Plot heatmap with variable names
ggplot(spearman_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "red", high = "blue", mid = "lightgray", midpoint = 0, limit = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spearman Correlation Heatmap", x = "Variables", y = "Variables") +
  geom_text(aes(label = round(value, 2)), color = "black", size = 6)
```



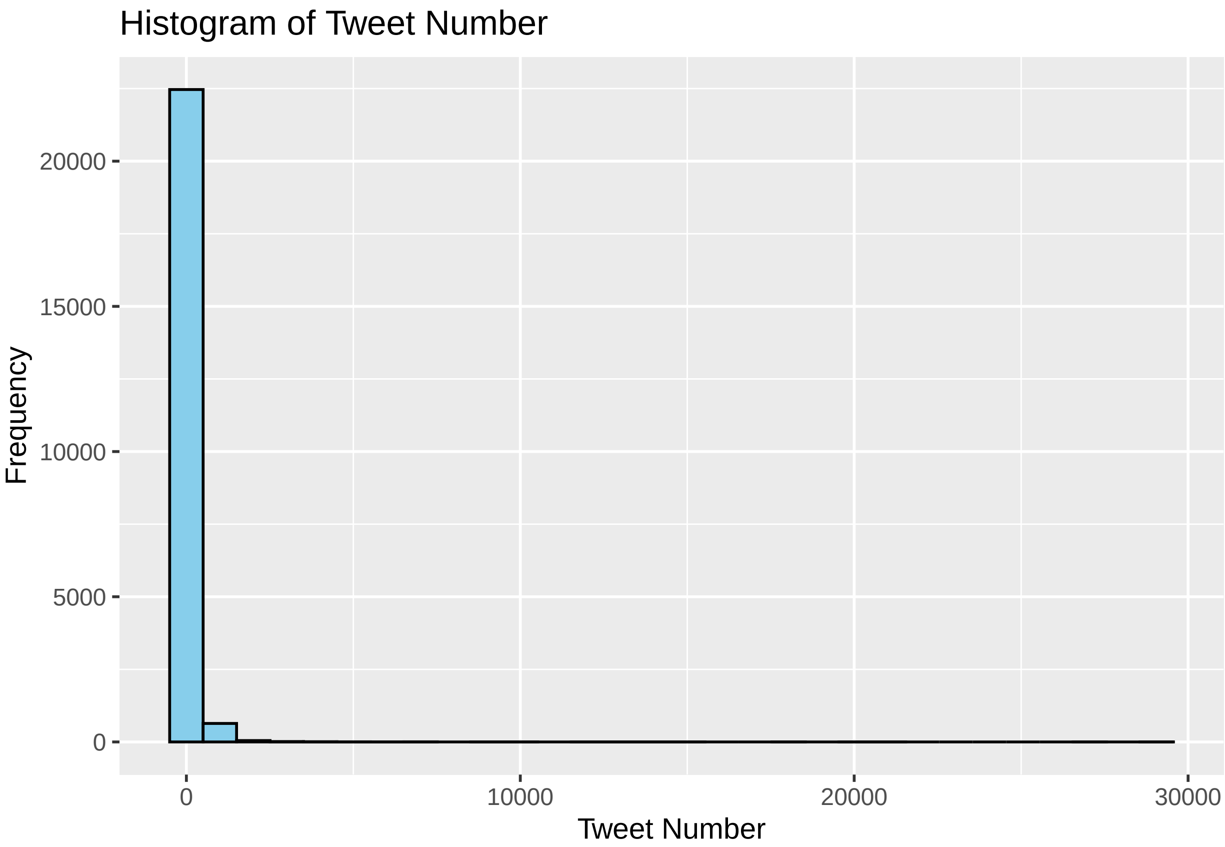
```
# Plot heatmap
ggplot(spearman_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 0, limit = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spearman Correlation Heatmap", x = "", y = "")
```



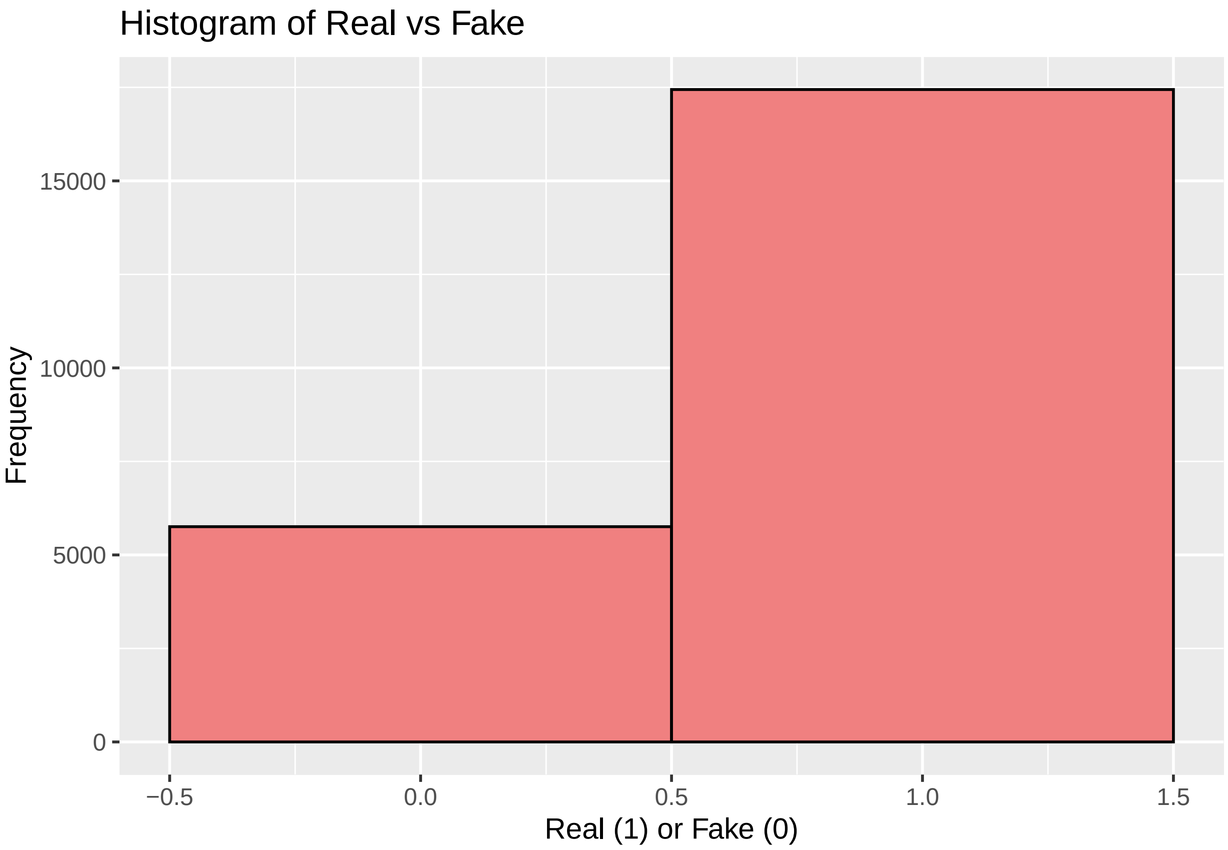
```r
# Scatterplot for tweet_num vs real
ggplot(FakeNewsNet, aes(x = tweet_num, y = real)) +
  geom_point(color = 'blue') +
  labs(title = "Scatterplot: tweet_num vs real", x = "Tweet Number", y = "Real (1) or Fake (0)")
```



```r
# Histogram for tweet_num
ggplot(FakeNewsNet, aes(x = tweet_num)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  labs(title = "Histogram of Tweet Number", x = "Tweet Number", y = "Frequency")
```



```
# Histogram for real (binary distribution)
ggplot(FakeNewsNet, aes(x = real)) +
  geom_histogram(bins = 2, fill = "lightcoral", color = "black") +
  labs(title = "Histogram of Real vs Fake", x = "Real (1) or Fake (0)", y = "Frequency")
```


Histogram of Real vs Fake
Frequency
15000
10000
5000
0
−0.5
0.0
0.5
1.0
1.5
Real (1) or Fake (0)