Correlation and Regression

MD MAHFUJUL KARIM SHEIKH

2025-03-10

Contents

Packages

install.packages("ggcorrplot")

install.packages("lmtest")

install.packages("lares")

install.packages("ggstatsplot")

```
library(ggplot2)
library(dplyr)
library(gridExtra)
library(ggcorrplot)

library(car)
library(lmtest)
```

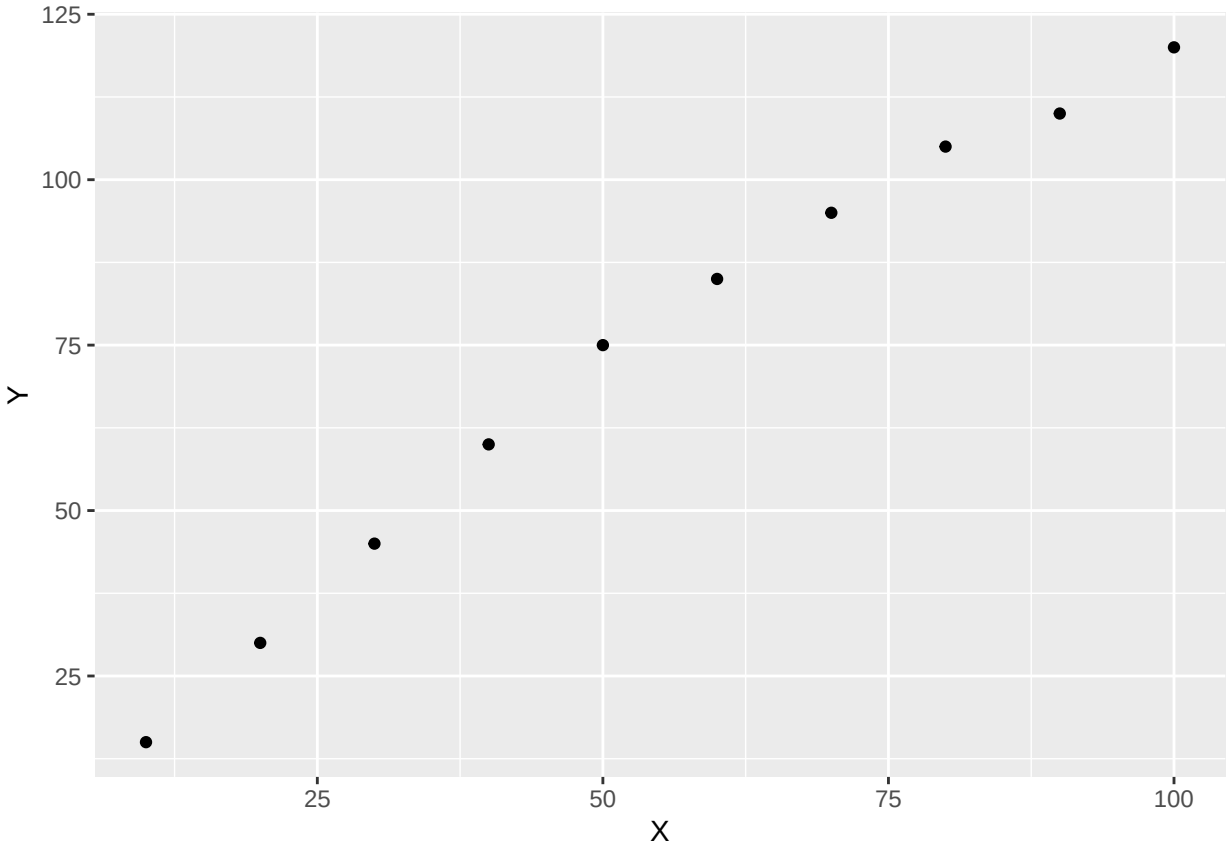
Correlation

Data

```
data <- data.frame(
  X = c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100),
  Y = c(15, 30, 45, 60, 75, 85, 95, 105, 110, 120),
  Z = c(30, NA, 40, 45, 30, 50, 30, 40, 60, 70),
  M = -c(30, NA, 40, 45, 30, 50, 30, 40, 60, 70),
  Cat = rep(c("A","B"), each = 5)
)
data
```

```
     X   Y  Z   M Cat
1   10  15 30 -30   A
2   20  30 NA  NA   A
3   30  45 40 -40   A
4   40  60 45 -45   A
5   50  75 30 -30   A
6   60  85 50 -50   B
7   70  95 30 -30   B
8   80 105 40 -40   B
9   90 110 60 -60   B
10 100 120 70 -70   B
```

```
ggplot(data, aes(x = X, y = Y)) + geom_point()
```



Functions

```
cor(data$X, data$Y)
```

```
[1] 0.9903356
```

```
cor(data$X, data$Z, use = "complete.obs")
```

```
[1] 0.6767234
```

```
cor(data$X, data$M, use = "complete.obs")
```

```
[1] -0.6767234
```

```
data %>%
  select(where(is.numeric)) %>%
  cor(use = "complete.obs") %>%
  round(3)
```

```
       X      Y      Z      M
X  1.000  0.989  0.677 -0.677
```

```
Y  0.989  1.000  0.610 -0.610
Z  0.677  0.610  1.000 -1.000
M -0.677 -0.610 -1.000  1.000
```

```
data %>%
  select(where(is.numeric)) %>%
  cor(use = "complete.obs", method = "spearman") %>%
  round(3)
```

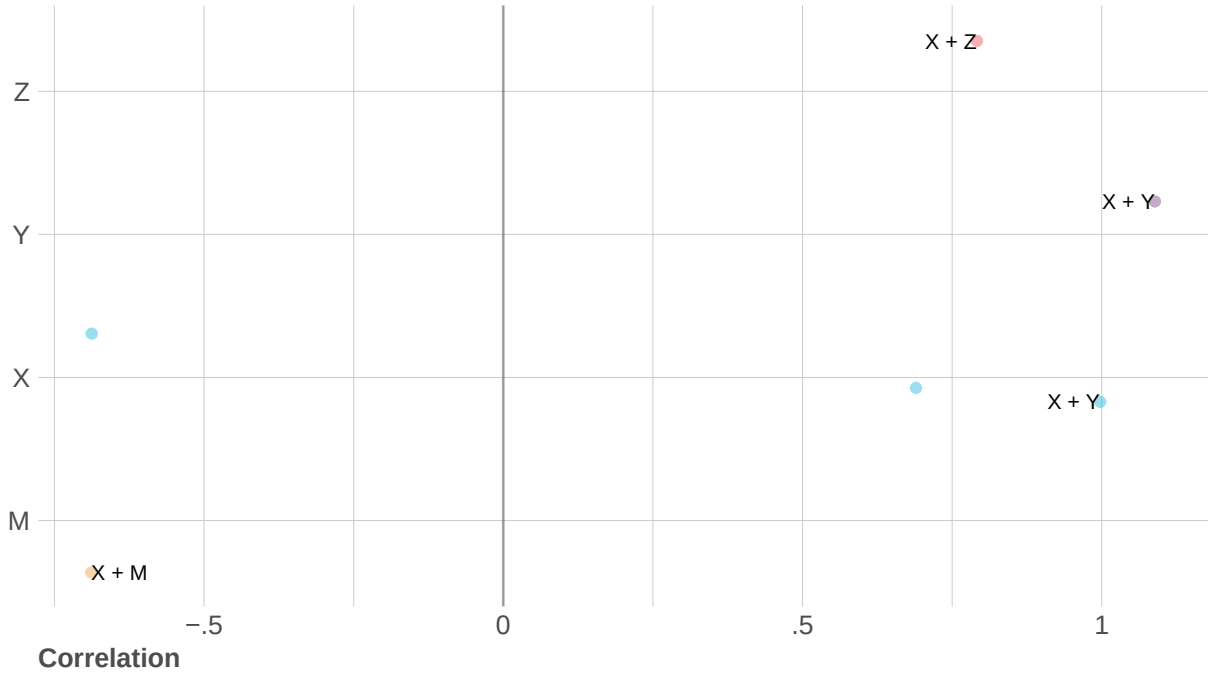
```
       X      Y      Z      M
X  1.000  1.000  0.604 -0.604
Y  1.000  1.000  0.604 -0.604
Z  0.604  0.604  1.000 -1.000
M -0.604 -0.604 -1.000  1.000
```

```
data %>%
  select(where(is.numeric)) %>%
  cor(use = "complete.obs", method = "kendall") %>%
  round(3)
```

```
      X     Y     Z     M
X  1.00  1.00  0.53 -0.53
Y  1.00  1.00  0.53 -0.53
Z  0.53  0.53  1.00 -1.00
M -0.53 -0.53 -1.00  1.00
```

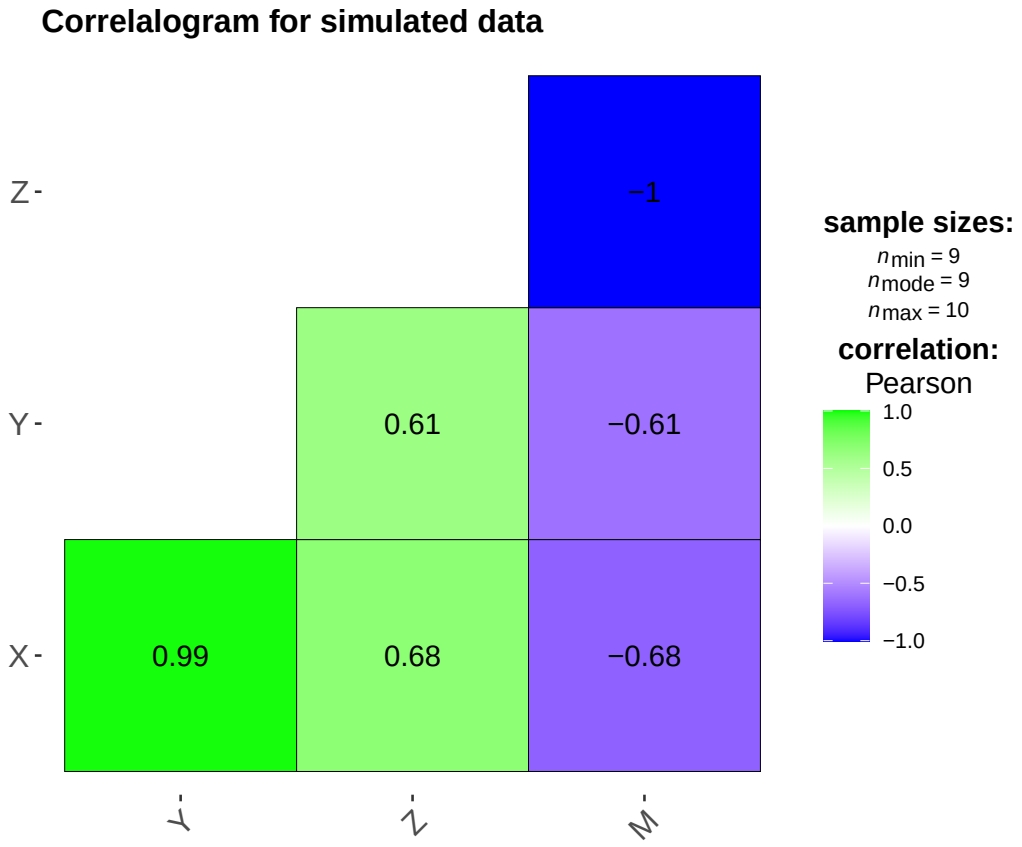
lares

```
library(lares)

corr_cross(
  df = data %>% select(where(is.numeric)),
  max_pvalue = 0.05,
  type = 2,
  top = 20,
  grid = F
)
```



ggstatsplot

```
library(ggstatsplot)

ggcorrmat(
  data = data,
  colors = c("blue", "white", "green"),
  title = "Correlalogram for simulated data",
  matrix.type = "lower",
  type = "parametric", pch = ""
)
```

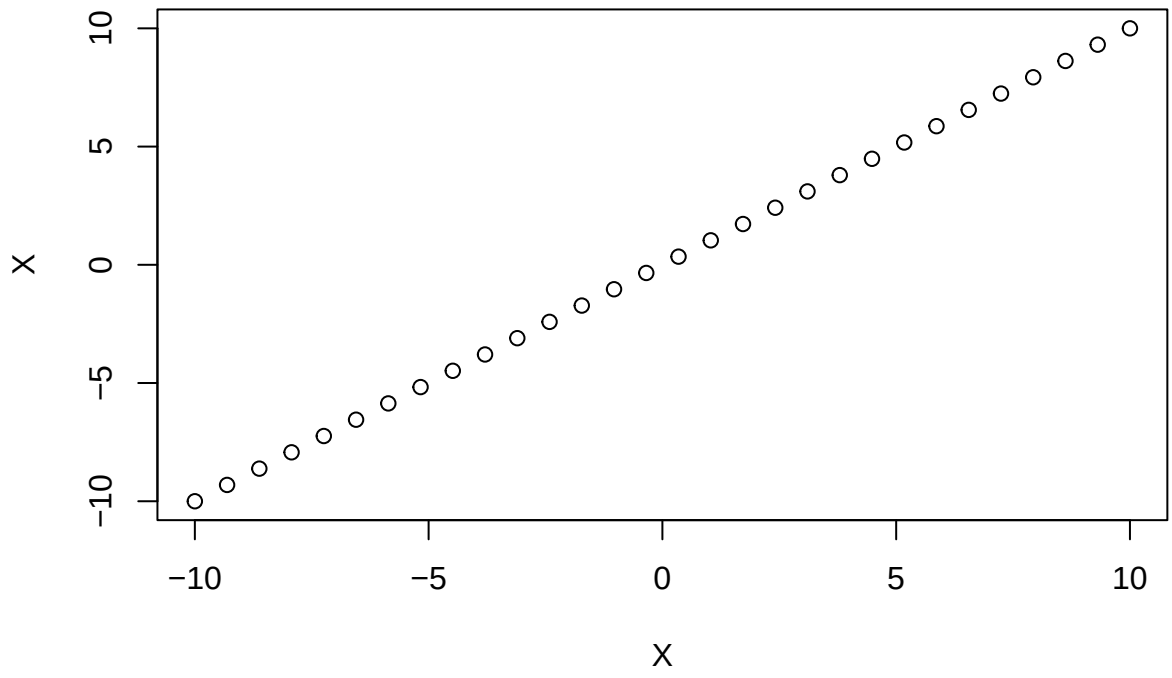


Example 1

```
set.seed(42)

# Generate 30 observations for X
X <- seq(-10, 10, length.out = 30)
U <- -X
cor(X, X + rnorm(30))
```

```
[1] 0.9785344
```

```
plot(X, X)
```

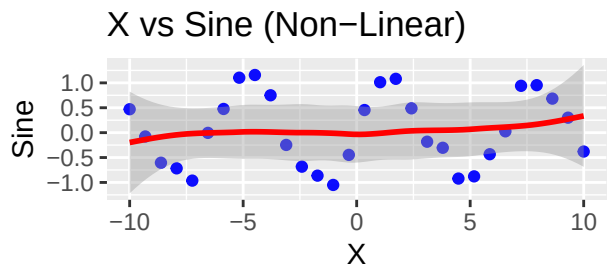


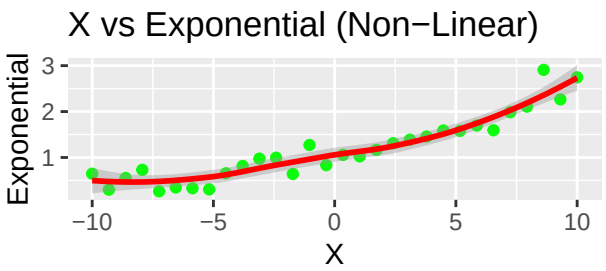

```r
# Create different relationships
Linear <- 2 * X + rnorm(30, mean = 0, sd = 1)              # Linear relationship with noise
Sine <- sin(X) + rnorm(30, mean = 0, sd = 0.2)             # Sine wave with noise
Exponential <- exp(X / 10) + rnorm(30, mean = 0, sd = 0.2) # Exponential growth
Quadratic <- X^2 + rnorm(30, mean = 0, sd = 5)             # Quadratic with noise
Logarithmic <- log(abs(X) + 1) + rnorm(30, mean = 0, sd = 0.2) # Logarithmic

df <- data.frame(X, Sine, Exponential, Quadratic, Logarithmic, Linear)
head(df)
```

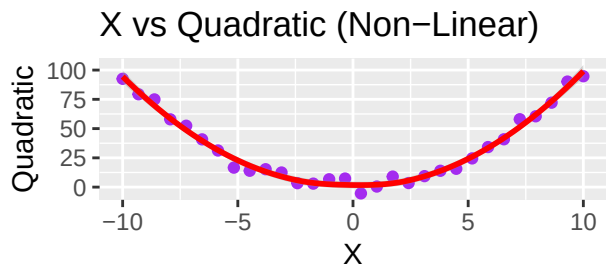
```
           X         Sine Exponential Quadratic Logarithmic    Linear
1 -10.000000  0.470574182   0.6463027  92.53187    2.389756 -19.54455
2  -9.310345 -0.077137434   0.2989110  79.33034    2.022839 -17.91585
3  -8.620690 -0.603833692   0.5523572  74.93980    2.497350 -16.20628
4  -7.931034 -0.717085532   0.7306606  57.91811    2.134803 -16.47100
5  -7.241379 -0.963612870   0.2625845  52.42846    2.015599 -13.97780
6  -6.551724 -0.004814394   0.3471940  40.78379    1.774125 -14.82046
```

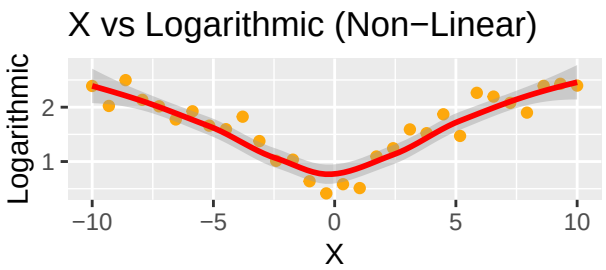
```r
# Scatter plot for Sine
p1 <- ggplot(df, aes(x = X, y = Sine)) +
  geom_point(color = "blue") +
  geom_smooth(method = "loess", color = "red") +
  ggtitle("X vs Sine (Non-Linear)")

# Scatter plot for Exponential
```

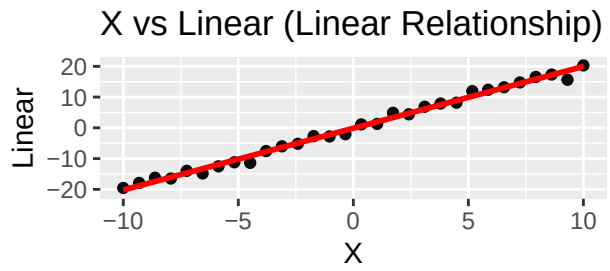
```r
p2 <- ggplot(df, aes(x = X, y = Exponential)) +
  geom_point(color = "green") +
  geom_smooth(method = "loess", color = "red") +
  ggtitle("X vs Exponential (Non-Linear)")

# Scatter plot for Quadratic
p3 <- ggplot(df, aes(x = X, y = Quadratic)) +
  geom_point(color = "purple") +
  geom_smooth(method = "loess", color = "red") +
  ggtitle("X vs Quadratic (Non-Linear)")

# Scatter plot for Logarithmic
p4 <- ggplot(df, aes(x = X, y = Logarithmic)) +
  geom_point(color = "orange") +
  geom_smooth(method = "loess", color = "red") +
  ggtitle("X vs Logarithmic (Non-Linear)")

# Scatter plot for Linear
p5 <- ggplot(df, aes(x = X, y = Linear)) +
  geom_point(color = "black") +
  geom_smooth(method = "lm", color = "red") +  # **Linear fit**
  ggtitle("X vs Linear (Linear Relationship)")

# Arrange plots in a grid
grid.arrange(p1, p2, p3, p4, p5, ncol = 2)
```





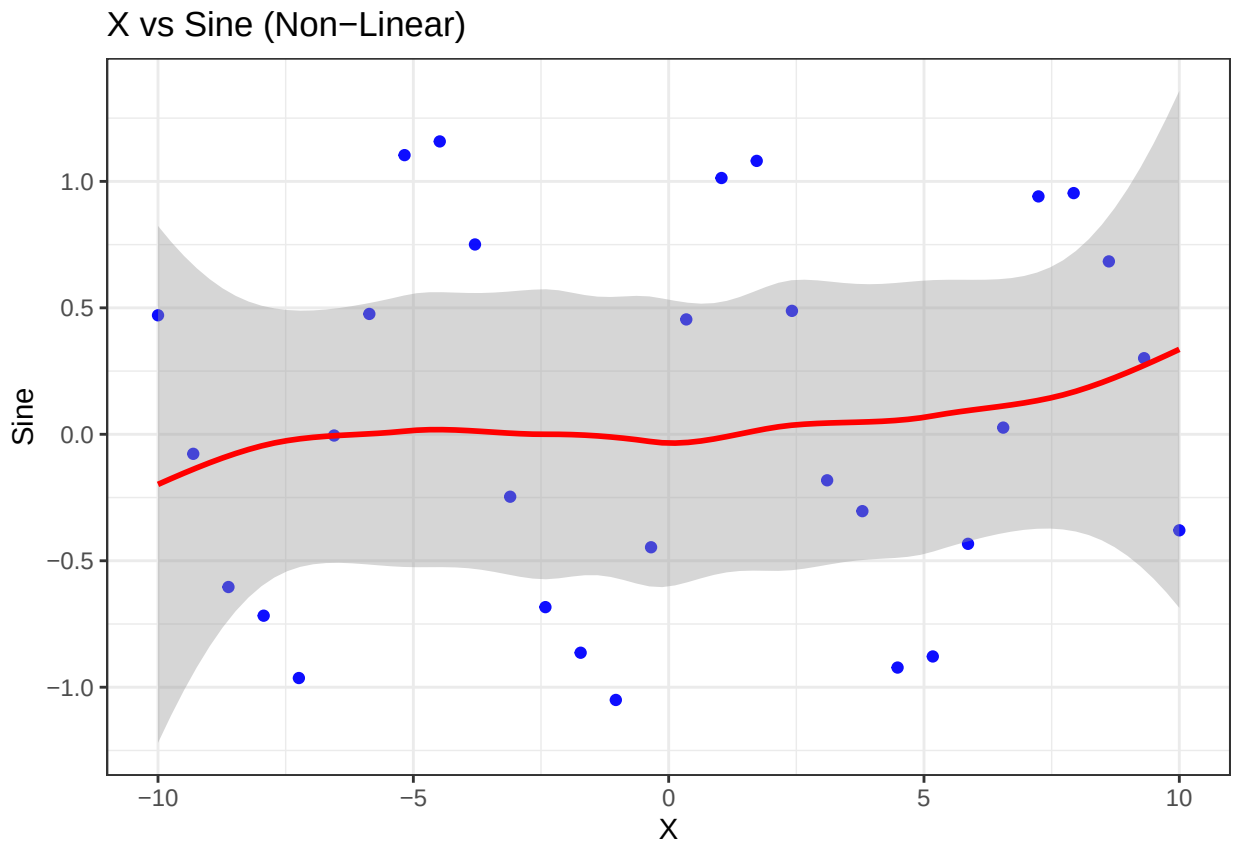




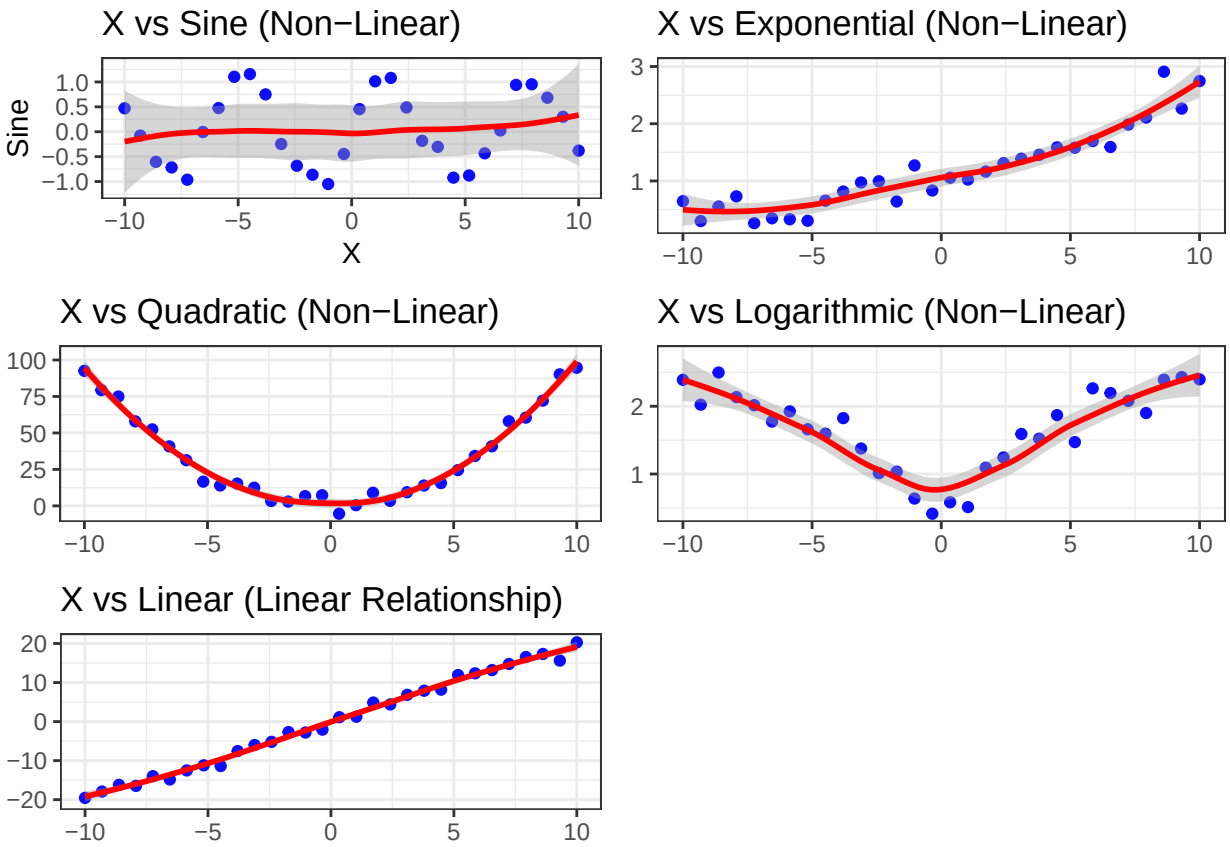


Shorten code by creating functions for ggplto:

```
scatter_plot <- function(data, x_var, y_var, title, x_lab = NULL, y_lab = NULL) {

  ggplot(data, aes(x = {x_var}, y = {y_var})) +
    geom_point(color = "blue") +
    geom_smooth(method = "loess", color = "red") +
    labs(title = title,
         x = x_lab, y = y_lab) +
    theme_bw()
}

scatter_plot(df, X, Sine, "X vs Sine (Non-Linear)", "X", "Sine")
```




```
p1 <- scatter_plot(df, X, Sine, "X vs Sine (Non-Linear)", "X", "Sine")
p2 <- scatter_plot(df, X, Exponential, "X vs Exponential (Non-Linear)")
p3 <- scatter_plot(df, X, Quadratic, "X vs Quadratic (Non-Linear)")
p4 <- scatter_plot(df, X, Logarithmic, "X vs Logarithmic (Non-Linear)")
p5 <- scatter_plot(df, X, Linear, "X vs Linear (Linear Relationship)")

# Arrange plots in a grid
grid.arrange(p1, p2, p3, p4, p5, ncol = 2)
```



Correlation Matrix:

```
# Compute correlation matrix
cor_matrix <- cor(df, method = "pearson")
cor_matrix
```

```
                    X       Sine Exponential  Quadratic Logarithmic     Linear
X           1.00000000 0.10493893  0.92022268 0.03832624  0.06142619 0.99631802
Sine        0.10493893 1.00000000  0.07949279 0.01511637  0.02869163 0.08845893
Exponential 0.92022268 0.07949279  1.00000000 0.29774862  0.26096613 0.91936183
Quadratic   0.03832624 0.01511637  0.29774862 1.00000000  0.83091855 0.04086503
Logarithmic 0.06142619 0.02869163  0.26096613 0.83091855  1.00000000 0.06656353
Linear      0.99631802 0.08845893  0.91936183 0.04086503  0.06656353 1.00000000
```

Regression

```
# Set seed for reproducibility
set.seed(42)

# Generate 50 observations
n <- 50
X1 <- runif(n, 1, 100)  # Independent variable
X2 <- rnorm(n, 0, 5) # another independent variable
```

```r
Y <- 5 + 2*X1 - 3*X2 + rnorm(n, 0, 10)  # Linear model with noise

df <- round(
  data.frame(Y, X1, X2),
  2)

head(df)
```

```
       Y    X1    X2
1 200.40 91.57 -2.15
2 204.08 93.77 -1.29
3  94.74 29.33 -8.82
4 155.67 83.21  2.30
5 132.67 64.53 -3.20
6 118.08 52.39  2.28
```

```r
cor(df)
```

```
            Y          X1          X2
Y   1.0000000  0.95634888 -0.26394709
X1  0.9563489  1.00000000 -0.01240269
X2 -0.2639471 -0.01240269  1.00000000
```

Model

```r
model <- lm(Y ~ X1 + X2, data = df)
summary(model)
```

```

Call:
lm(formula = Y ~ X1 + X2, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-17.2545  -6.1818  -0.1287   4.4273  20.2253

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.69609    2.83610   4.477 4.81e-05 ***
X1           1.86519    0.04218  44.217  < 2e-16 ***
X2          -3.05519    0.26124 -11.695 1.62e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.876 on 47 degrees of freedom
Multiple R-squared:  0.9782,    Adjusted R-squared:  0.9772
F-statistic:  1053 on 2 and 47 DF,  p-value: < 2.2e-16
```

```r
test <- data.frame(
  X1 = c(30, 40),
  X2 = c(10, 14)
)

cbind(test, predicted = predict(model, test))
```

```
  X1 X2 predicted
1 30 10  38.10002
2 40 14  44.53120
```

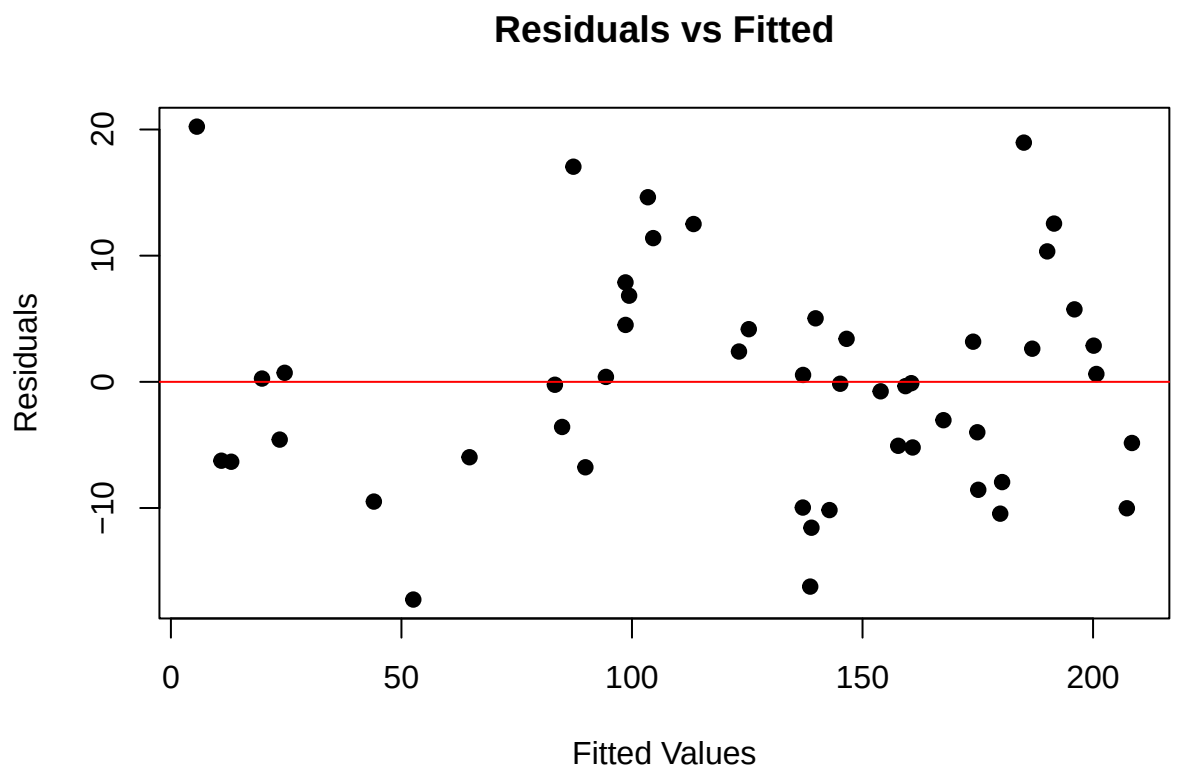
Residuals vs. fitted plot

```r
cbind(df, predicted = predict(model, df), resid(model))
```

```
        Y    X1     X2  predicted resid(model)
1  200.40 91.57  -2.15 190.060609   10.3393913
2  204.08 93.77  -1.29 191.536572   12.5434277
3   94.74 29.33  -8.82  94.349028    0.3909719
4  155.67 83.21   2.30 160.871984   -5.2019836
5  132.67 64.53  -3.20 142.833701  -10.1637007
6  118.08 52.39   2.28 103.447794   14.6322060
7  144.85 73.92   3.52 139.816994    5.0330058
8   19.02 14.33   5.18  23.598439   -4.5784392
9  145.01 66.04  -3.04 145.161314   -0.1513138
10 127.09 70.80   2.52 137.052778   -9.9627783
11 129.51 46.32  -8.59 125.335988    4.1740118
12 158.98 72.19  -3.92 159.320828   -0.3408277
13 203.00 93.53  -4.25 200.132291    2.8677094
14 103.12 26.29 -12.07  98.608207    4.5117928
15 106.21 46.77   0.18  99.381302    6.8286981
16 203.95 94.06   1.03 184.989436   18.9605640
17 201.34 97.84  -1.81 200.716613    0.6233869
18  25.39 12.63   3.79  24.674324    0.7156761
19 125.86 48.02  -3.63 113.353072   12.5069279
20 127.36 56.47  -6.84 138.921128  -11.5611280
21 170.90 90.50   2.16 174.896978   -3.9969780
22  35.32 14.73  -4.06  52.574480  -17.2544805
23 166.55 98.90   7.22 175.105346   -8.5553458
24 201.71 94.72  -2.16 195.966523    5.7434769
25  20.02  9.16   3.28  19.760246    0.2597536
26 115.99 51.91   1.61 104.599479   11.3905215
27 106.47 39.63  -3.92  98.590096    7.8799041
28 152.67 90.67   7.88 157.738369   -5.0683694
29 104.34 45.25   3.21  87.288978   17.0510220
30 164.51 83.76   0.45 167.549944   -3.0399437
31 149.95 74.02   1.38 146.541622    3.4083779
32 153.18 81.29   3.40 153.930100   -0.7501003
33  81.27 39.42   0.45  84.847221   -3.5772211
34 189.44 68.83 -14.97 186.813633    2.6263667
35   4.70  1.39   1.42  10.950340   -6.2503404
```

```
36 177.17 83.46  -1.84 173.986772    3.1832276
37   6.75  1.73   0.93  13.081550   -6.3315500
38  34.53 21.56   2.91  44.019078   -9.4890784
39 160.47 90.75   7.00 160.576153   -0.1061529
40 122.43 61.57  -3.64 138.657009  -16.2270091
41  58.79 38.58   6.51  64.766001   -5.9760011
42  83.12 44.14   1.68  89.893054   -6.7730542
43  25.85  4.71   5.19   5.624716   20.2252835
44 172.33 97.38   4.60 180.274850   -7.9448502
45  83.05 43.74   3.60  83.281010   -0.2310099
46 197.31 95.80  -5.22 207.329817  -10.0198172
47 169.42 88.89  -0.45 179.868063  -10.4480630
48 125.61 64.36   3.12 123.207811    2.4021886
49 203.59 97.13  -4.77 208.435690   -4.8456899
50 137.65 62.26  -2.71 137.102666    0.5473342
```

```r
plot(model$fitted.values, resid(model),
     main="Residuals vs Fitted", xlab="Fitted Values", ylab="Residuals", pch=19)
abline(h=0, col="red")  # Reference line
```



VIF

```
car::vif(model)
```

```
      X1       X2
1.000154 1.000154
```

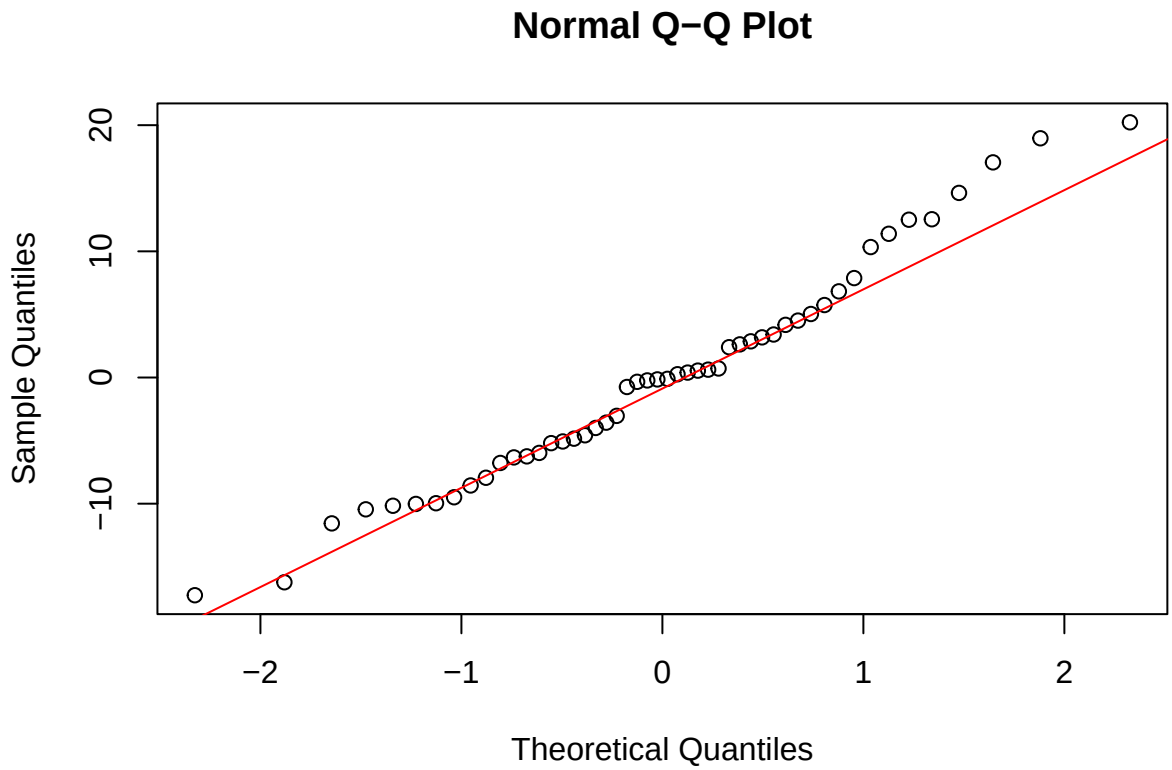
Breusch-Pagan Test

```
lmtest::bptest(model)
```

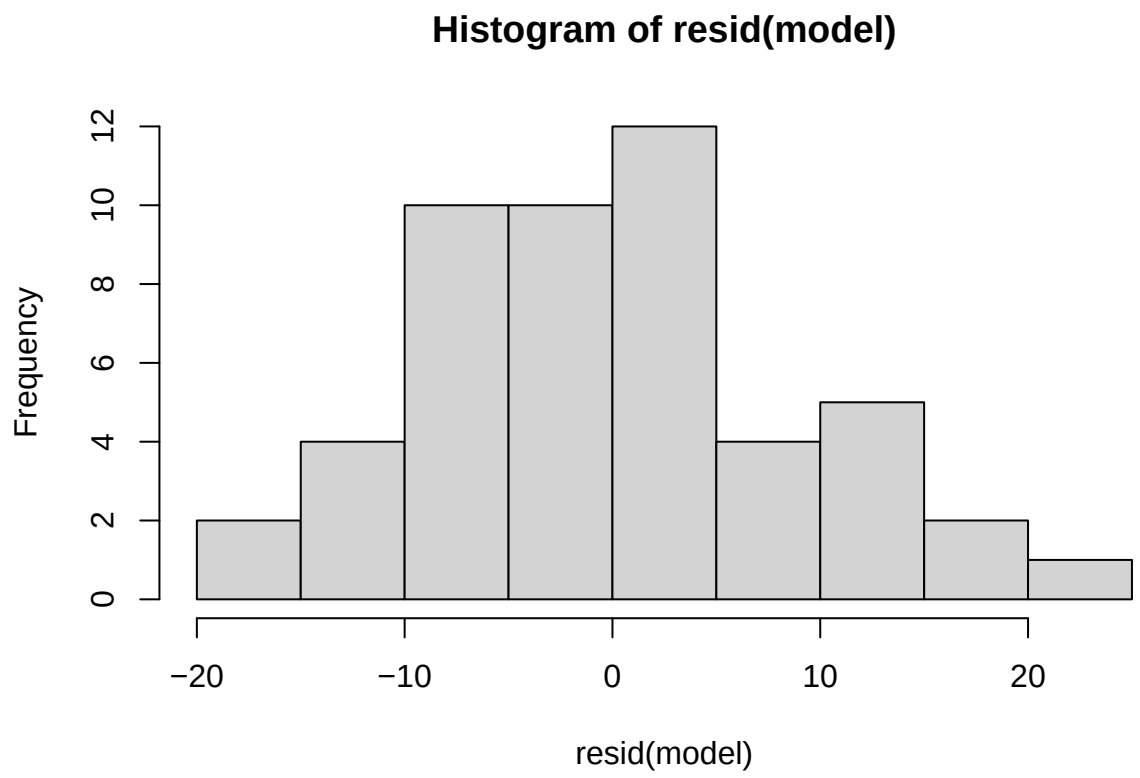
```

    studentized Breusch-Pagan test

data:  model
BP = 1.1264, df = 2, p-value = 0.5694
```

QQ plot

```
qqnorm(resid(model))
qqline(resid(model), col="red")
```



```r
hist(resid(model))
```




```r
shapiro.test(resid(model))
```

```
    Shapiro-Wilk normality test

data:  resid(model)
W = 0.97605, p-value = 0.3999
```