Server Chuchu not sure

Cristel Kaye Billones

Item no. 30

```
library(tidyverse)
library(car)  # For Levene's test
# Import the Excel file
file_path <- file.choose()  # Select your file
df <- read_csv(file_path)

# Check the data structure
head(df)
```

```
## # A tibble: 6 x 5
##   Server `Server Type` `Security Protocol` Time     `Response Time`
##    <dbl> <chr>         <chr>               <chr>              <dbl>
## 1      1 Linux         SSL                 Baseline            93.9
## 2      2 Windows       TLS                 Baseline           160.
## 3      3 Windows       SSL                 Baseline           134.
## 4      4 Linux         TLS                 Baseline           110.
## 5      5 Windows       SSL                 Baseline           133.
## 6      6 Linux         SSL                 Baseline           111.
```

```
#Data cleaning
# Step 1: Clean column names (remove spaces and special characters)
df <- df %>%
  rename(Server = `Server`,
         Server_Type = `Server Type`,
         Security_Protocol = `Security Protocol`,
         Time = `Time`,
         Response_Time = `Response Time`)

# Step 2: Check for missing data
missing_data_summary <- df %>%
  summarise(across(everything(), ~sum(is.na(.))))
print(missing_data_summary)
```

```
## # A tibble: 1 x 5
##   Server Server_Type Security_Protocol  Time Response_Time
##    <int>       <int>             <int> <int>         <int>
## 1      0           0                 0     0             0
```

```r
# Step 3: Ensure consistent formatting for categorical variables
df <- df %>%
  mutate(Server_Type = factor(Server_Type),
         Security_Protocol = factor(Security_Protocol),
         Time = factor(Time))

# Step 4: Check the cleaned data structure
glimpse(df)
```

```
## Rows: 1,200
## Columns: 5
## $ Server            <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Server_Type       <fct> Linux, Windows, Windows, Linux, Windows, Linux, Linu~
## $ Security_Protocol <fct> SSL, TLS, SSL, TLS, SSL, SSL, TLS, SSL, TLS, SSL, SS~
## $ Time              <fct> Baseline, Baseline, Baseline, Baseline, Baseline, Ba~
## $ Response_Time     <dbl> 93.88120, 160.10515, 134.28620, 109.50370, 132.96656~
```

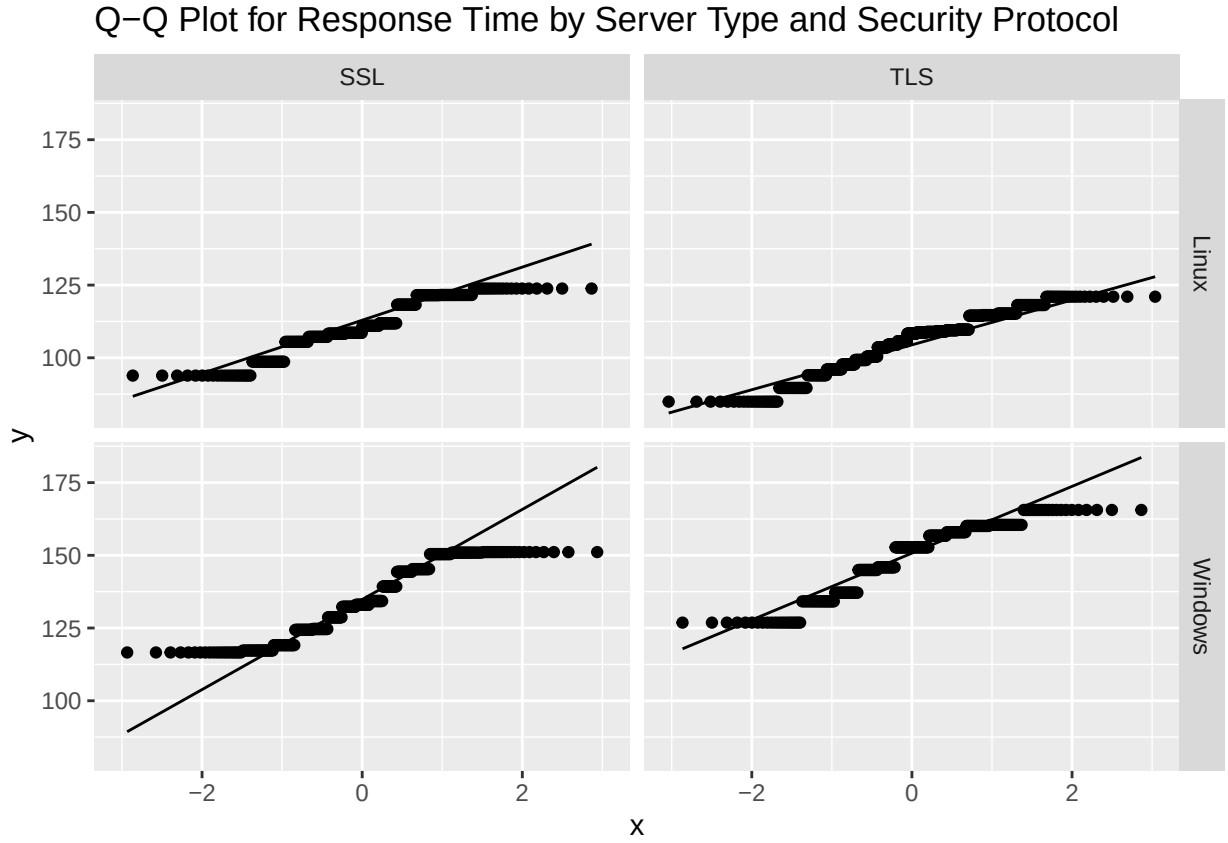
```r
# Step 5: Optional - drop rows with missing data
df_cleaned <- df %>%
  drop_na()

# View the cleaned dataset
head(df_cleaned)
```

```
## # A tibble: 6 x 5
##   Server Server_Type Security_Protocol Time     Response_Time
##    <dbl> <fct>       <fct>             <fct>            <dbl>
## 1      1 Linux       SSL               Baseline          93.9
## 2      2 Windows     TLS               Baseline         160.
## 3      3 Windows     SSL               Baseline         134.
## 4      4 Linux       TLS               Baseline         110.
## 5      5 Windows     SSL               Baseline         133.
## 6      6 Linux       SSL               Baseline         111.
```

```r
#1. Check assumptions
# Load necessary libraries
library(tidyverse)
library(car)  # For Levene's test
library(ggplot2)

# 1. Check Normality for each group
df_cleaned %>%
  ggplot(aes(sample = Response_Time)) +
  stat_qq() +
  stat_qq_line() +
  facet_grid(Server_Type ~ Security_Protocol) +
  labs(title = "Q-Q Plot for Response Time by Server Type and Security Protocol")
```




```
# Shapiro-Wilk test for normality
normality_results <- df_cleaned %>%
  group_by(Server_Type, Security_Protocol) %>%
  summarise(shapiro_p = shapiro.test(Response_Time)$p.value)

print("Shapiro-Wilk Test Results for Normality:")
```

```
## [1] "Shapiro-Wilk Test Results for Normality:"
```

```
print(normality_results)
```

```
## # A tibble: 4 x 3
## # Groups:   Server_Type [2]
##   Server_Type Security_Protocol shapiro_p
##   <fct>       <fct>                <dbl>
## 1 Linux       SSL                5.93e-10
## 2 Linux       TLS                8.12e-10
## 3 Windows     SSL                1.47e-12
## 4 Windows     TLS                9.03e-11
```

```
# 2. Check Homogeneity of Variance using Levene's Test
levene_test <- leveneTest(Response_Time ~ Server_Type * Security_Protocol, data = df_cleaned)
print("Levene's Test for Homogeneity of Variance:")
```

```
## [1] "Levene's Test for Homogeneity of Variance:"
```

```r
print(levene_test)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    3  16.683 1.273e-10 ***
##       1196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 3. Check for Independence (No Duplicates)
independence_check <- if(length(unique(df_cleaned$Server)) == nrow(df_cleaned)){
  "Independence assumption met: All servers are independent."
} else {
  "Independence assumption violated: Duplicates found."
}
print(independence_check)
```

```
## [1] "Independence assumption violated: Duplicates found."
```

```r
#Since there was a violation
# Load necessary libraries
library(tidyverse)
library(afex)  # For Repeated-Measures ANOVA with corrections
library(car)   # For Levene's Test

# Step 1: Clean and Prepare Data (If not already done)
colnames(df_cleaned)
```

```
## [1] "Server"            "Server_Type"       "Security_Protocol"
## [4] "Time"              "Response_Time"
```

```r
df_cleaned <- df %>%
  rename(Server = `Server`,
         Server_Type = `Server_Type`,
         Security_Protocol = `Security_Protocol`,
         Time = `Time`,
         Response_Time = `Response_Time`) %>%
  mutate(Server_Type = factor(Server_Type),
         Security_Protocol = factor(Security_Protocol),
         Time = factor(Time)) %>%
  drop_na()

# Step 2: Check Assumptions

# 1. Shapiro-Wilk Test for Normality (Already performed)
print("Shapiro-Wilk Test Results for Normality:")
```

```
## [1] "Shapiro-Wilk Test Results for Normality:"
```

```r
print(normality_results)
```

```
## # A tibble: 4 x 3
## # Groups:   Server_Type [2]
##   Server_Type Security_Protocol shapiro_p
##   <fct>       <fct>                <dbl>
## 1 Linux       SSL                5.93e-10
## 2 Linux       TLS                8.12e-10
## 3 Windows     SSL                1.47e-12
## 4 Windows     TLS                9.03e-11
```

```r
# 2. Levene's Test for Homogeneity of Variance (Already performed)
levene_test <- leveneTest(Response_Time ~ Server_Type * Security_Protocol, data = df_cleaned)
print("Levene's Test for Homogeneity of Variance:")
```

```
## [1] "Levene's Test for Homogeneity of Variance:"
```

```r
print(levene_test)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    3  16.683 1.273e-10 ***
##       1196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 3. Check for Independence (No Duplicates)
independence_check <- if(length(unique(df_cleaned$Server)) == nrow(df_cleaned)){
  "Independence assumption met: All servers are independent."
} else {
  "Independence assumption violated: Duplicates found."
}
print(independence_check)
```

```
## [1] "Independence assumption violated: Duplicates found."
```

```r
# Step 3: Perform Repeated-Measures ANOVA
# Since independence is violated and sphericity may be an issue, we will use Greenhouse-Geisser correct
anova_model <- aov_car(Response_Time ~ Server_Type * Security_Protocol + Error(Server/Time),
                       data = df_cleaned, factorize = TRUE)

# Step 4: Display the ANOVA Results with Greenhouse-Geisser Correction
anova_results <- summary(anova_model, correction = "GG")

# Print the results
print("Repeated-Measures ANOVA with Greenhouse-Geisser Correction:")
```

```
## [1] "Repeated-Measures ANOVA with Greenhouse-Geisser Correction:"
```

```r
print(anova_results)
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##                                   Sum Sq num Df Error SS den Df   F value
## (Intercept)                       889721      1   4371.5     16 3256.4391
## Server_Type                        16208      1   4371.5     16   59.3229
## Security_Protocol                    361      1   4371.5     16    1.3221
## Server_Type:Security_Protocol       1563      1   4371.5     16    5.7212
## Time                                 463      2    400.1     32   18.5152
## Server_Type:Time                     858      2    400.1     32   34.3040
## Security_Protocol:Time                98      2    400.1     32    3.9080
## Server_Type:Security_Protocol:Time     80      2    400.1     32    3.2070
##                                     Pr(>F)
## (Intercept)                       < 2.2e-16 ***
## Server_Type                       9.040e-07 ***
## Security_Protocol                   0.26712
## Server_Type:Security_Protocol       0.02939 *
## Time                              4.547e-06 ***
## Server_Type:Time                  1.097e-08 ***
## Security_Protocol:Time              0.03030 *
## Server_Type:Security_Protocol:Time  0.05377 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##                                   Test statistic  p-value
## Time                                     0.65079 0.039883
## Server_Type:Time                         0.65079 0.039883
## Security_Protocol:Time                   0.65079 0.039883
## Server_Type:Security_Protocol:Time       0.65079 0.039883
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##                                    GG eps Pr(>F[GG])
## Time                               0.74117  5.363e-05 ***
## Server_Type:Time                   0.74117  5.829e-07 ***
## Security_Protocol:Time             0.74117    0.04524 *
## Server_Type:Security_Protocol:Time 0.74117    0.07136 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                      HF eps   Pr(>F[HF])
## Time                               0.799023 3.083574e-05
## Server_Type:Time                   0.799023 2.393222e-07
## Security_Protocol:Time             0.799023 4.135540e-02
## Server_Type:Security_Protocol:Time 0.799023 6.700111e-02
```

```r
#Perform the anova

# Load necessary libraries
library(tidyverse)
library(afex)  # For ANOVA
library(car)   # For Levene's Test
library(emmeans) # For post-hoc analysis if needed


# Perform the Three-Way Repeated-Measures ANOVA
anova_model <- aov_car(Response_Time ~ Server_Type * Security_Protocol + Error(Server/Time),
                       data = df_cleaned, factorize = TRUE)

# Display the ANOVA results with Greenhouse-Geisser Correction (to handle sphericity issues)
anova_results <- summary(anova_model, correction = "GG")

# Print the results
print("Three-Way ANOVA with Greenhouse-Geisser Correction:")
```

```
## [1] "Three-Way ANOVA with Greenhouse-Geisser Correction:"
```

```r
print(anova_results)
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##                                   Sum Sq num Df Error SS den Df   F value
## (Intercept)                       889721      1   4371.5     16 3256.4391
## Server_Type                        16208      1   4371.5     16   59.3229
## Security_Protocol                    361      1   4371.5     16    1.3221
## Server_Type:Security_Protocol       1563      1   4371.5     16    5.7212
## Time                                 463      2    400.1     32   18.5152
## Server_Type:Time                     858      2    400.1     32   34.3040
## Security_Protocol:Time                98      2    400.1     32    3.9080
## Server_Type:Security_Protocol:Time     80      2    400.1     32    3.2070
##                                     Pr(>F)
## (Intercept)                       < 2.2e-16 ***
## Server_Type                       9.040e-07 ***
## Security_Protocol                   0.26712
## Server_Type:Security_Protocol       0.02939 *
## Time                              4.547e-06 ***
## Server_Type:Time                  1.097e-08 ***
## Security_Protocol:Time              0.03030 *
## Server_Type:Security_Protocol:Time  0.05377 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##                                   Test statistic  p-value
## Time                                     0.65079 0.039883
```

```
## Server_Type:Time                           0.65079 0.039883
## Security_Protocol:Time                     0.65079 0.039883
## Server_Type:Security_Protocol:Time         0.65079 0.039883
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##                                     GG eps Pr(>F[GG])
## Time                               0.74117  5.363e-05 ***
## Server_Type:Time                   0.74117  5.829e-07 ***
## Security_Protocol:Time             0.74117    0.04524 *
## Server_Type:Security_Protocol:Time 0.74117    0.07136 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                       HF eps   Pr(>F[HF])
## Time                               0.799023 3.083574e-05
## Server_Type:Time                   0.799023 2.393222e-07
## Security_Protocol:Time             0.799023 4.135540e-02
## Server_Type:Security_Protocol:Time 0.799023 6.700111e-02
```

```
# Post-hoc analysis if needed
emmeans_model <- emmeans(anova_model, pairwise ~ Server_Type * Security_Protocol * Time, adjust = "bonf
print("Post-hoc analysis results:")
```

```
## [1] "Post-hoc analysis results:"
```

```
print(emmeans_model)
```

```
## $emmeans
##  Server_Type Security_Protocol Time      emmean   SE df lower.CL upper.CL
##  Linux       SSL               X1.Month   113.5 4.62 16    103.7      123
##  Windows     SSL               X1.Month   130.0 4.13 16    121.2      139
##  Linux       TLS               X1.Month   107.6 3.49 16    100.2      115
##  Windows     TLS               X1.Month   144.3 4.62 16    134.5      154
##  Linux       SSL               X2.Months  116.4 5.37 16    105.0      128
##  Windows     SSL               X2.Months  132.9 4.80 16    122.7      143
##  Linux       TLS               X2.Months  111.4 4.06 16    102.8      120
##  Windows     TLS               X2.Months  155.0 5.37 16    143.6      166
##  Linux       SSL               Baseline   102.7 4.94 16     92.2      113
##  Windows     SSL               Baseline   139.5 4.41 16    130.1      149
##  Linux       TLS               Baseline    97.2 3.73 16     89.3      105
##  Windows     TLS               Baseline   149.5 4.94 16    139.1      160
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast                                      estimate   SE df t.ratio p.value
##  Linux SSL X1.Month - Windows SSL X1.Month       -16.51 6.20 16  -2.663  1.0000
##  Linux SSL X1.Month - Linux TLS X1.Month           5.92 5.79 16   1.023  1.0000
##  Linux SSL X1.Month - Windows TLS X1.Month       -30.86 6.53 16  -4.724  0.0151
##  Linux SSL X1.Month - Linux SSL X2.Months         -2.90 2.29 16  -1.266  1.0000
```

```
##  Linux SSL X1.Month - Windows SSL X2.Months      -19.40 6.66 16  -2.911  0.6740
##  Linux SSL X1.Month - Linux TLS X2.Months          2.06 6.15 16   0.335  1.0000
##  Linux SSL X1.Month - Windows TLS X2.Months      -41.51 7.08 16  -5.860  0.0016
##  Linux SSL X1.Month - Linux SSL Baseline          10.80 1.91 16   5.643  0.0024
##  Linux SSL X1.Month - Windows SSL Baseline       -25.98 6.39 16  -4.066  0.0593
##  Linux SSL X1.Month - Linux TLS Baseline          16.24 5.94 16   2.735  0.9684
##  Linux SSL X1.Month - Windows TLS Baseline       -36.06 6.76 16  -5.333  0.0044
##  Windows SSL X1.Month - Linux TLS X1.Month        22.43 5.41 16   4.146  0.0502
##  Windows SSL X1.Month - Windows TLS X1.Month     -14.36 6.20 16  -2.317  1.0000
##  Windows SSL X1.Month - Linux SSL X2.Months       13.60 6.78 16   2.007  1.0000
##  Windows SSL X1.Month - Windows SSL X2.Months     -2.89 2.05 16  -1.411  1.0000
##  Windows SSL X1.Month - Linux TLS X2.Months       18.57 5.79 16   3.205  0.3645
##  Windows SSL X1.Month - Windows TLS X2.Months    -25.01 6.78 16  -3.690  0.1308
##  Windows SSL X1.Month - Linux SSL Baseline        27.31 6.44 16   4.243  0.0410
##  Windows SSL X1.Month - Windows SSL Baseline      -9.48 1.71 16  -5.534  0.0030
##  Windows SSL X1.Month - Linux TLS Baseline        32.75 5.57 16   5.882  0.0015
##  Windows SSL X1.Month - Windows TLS Baseline     -19.55 6.44 16  -3.037  0.5179
##  Linux TLS X1.Month - Windows TLS X1.Month       -36.79 5.79 16  -6.353  0.0006
##  Linux TLS X1.Month - Linux SSL X2.Months         -8.83 6.41 16  -1.378  1.0000
##  Linux TLS X1.Month - Windows SSL X2.Months      -25.32 5.94 16  -4.263  0.0392
##  Linux TLS X1.Month - Linux TLS X2.Months         -3.86 1.73 16  -2.228  1.0000
##  Linux TLS X1.Month - Windows TLS X2.Months      -47.44 6.41 16  -7.404  0.0001
##  Linux TLS X1.Month - Linux SSL Baseline           4.88 6.05 16   0.807  1.0000
##  Linux TLS X1.Month - Windows SSL Baseline       -31.90 5.63 16  -5.668  0.0023
##  Linux TLS X1.Month - Linux TLS Baseline          10.32 1.45 16   7.131  0.0002
##  Linux TLS X1.Month - Windows TLS Baseline       -41.98 6.05 16  -6.943  0.0002
##  Windows TLS X1.Month - Linux SSL X2.Months       27.96 7.08 16   3.946  0.0762
##  Windows TLS X1.Month - Windows SSL X2.Months     11.46 6.66 16   1.720  1.0000
##  Windows TLS X1.Month - Linux TLS X2.Months       32.92 6.15 16   5.353  0.0043
##  Windows TLS X1.Month - Windows TLS X2.Months    -10.65 2.29 16  -4.644  0.0178
##  Windows TLS X1.Month - Linux SSL Baseline        41.67 6.76 16   6.164  0.0009
##  Windows TLS X1.Month - Windows SSL Baseline       4.88 6.39 16   0.764  1.0000
##  Windows TLS X1.Month - Linux TLS Baseline        47.11 5.94 16   7.933  <.0001
##  Windows TLS X1.Month - Windows TLS Baseline      -5.19 1.91 16  -2.712  1.0000
##  Linux SSL X2.Months - Windows SSL X2.Months     -16.50 7.21 16  -2.289  1.0000
##  Linux SSL X2.Months - Linux TLS X2.Months         4.96 6.73 16   0.737  1.0000
##  Linux SSL X2.Months - Windows TLS X2.Months     -38.61 7.60 16  -5.083  0.0073
##  Linux SSL X2.Months - Linux SSL Baseline         13.71 3.14 16   4.372  0.0313
##  Linux SSL X2.Months - Windows SSL Baseline      -23.08 6.95 16  -3.319  0.2864
##  Linux SSL X2.Months - Linux TLS Baseline         19.15 6.54 16   2.927  0.6511
##  Linux SSL X2.Months - Windows TLS Baseline      -33.15 7.30 16  -4.544  0.0219
##  Windows SSL X2.Months - Linux TLS X2.Months      21.46 6.29 16   3.411  0.2359
##  Windows SSL X2.Months - Windows TLS X2.Months   -22.11 7.21 16  -3.069  0.4849
##  Windows SSL X2.Months - Linux SSL Baseline       30.20 6.89 16   4.385  0.0305
##  Windows SSL X2.Months - Windows SSL Baseline     -6.58 2.80 16  -2.348  1.0000
##  Windows SSL X2.Months - Linux TLS Baseline       35.64 6.08 16   5.859  0.0016
##  Windows SSL X2.Months - Windows TLS Baseline    -16.66 6.89 16  -2.418  1.0000
##  Linux TLS X2.Months - Windows TLS X2.Months     -43.57 6.73 16  -6.471  0.0005
##  Linux TLS X2.Months - Linux SSL Baseline          8.74 6.39 16   1.368  1.0000
##  Linux TLS X2.Months - Windows SSL Baseline      -28.04 6.00 16  -4.675  0.0167
##  Linux TLS X2.Months - Linux TLS Baseline         14.18 2.37 16   5.984  0.0013
##  Linux TLS X2.Months - Windows TLS Baseline      -38.11 6.39 16  -5.963  0.0013
##  Windows TLS X2.Months - Linux SSL Baseline       52.32 7.30 16   7.172  0.0001
##  Windows TLS X2.Months - Windows SSL Baseline     15.53 6.95 16   2.234  1.0000
```

```
##  Windows TLS X2.Months - Linux TLS Baseline       57.76 6.54 16   8.831  <.0001
##  Windows TLS X2.Months - Windows TLS Baseline      5.46 3.14 16   1.741  1.0000
##  Linux SSL Baseline - Windows SSL Baseline       -36.79 6.62 16  -5.555  0.0029
##  Linux SSL Baseline - Linux TLS Baseline           5.44 6.19 16   0.879  1.0000
##  Linux SSL Baseline - Windows TLS Baseline       -46.86 6.98 16  -6.713  0.0003
##  Windows SSL Baseline - Linux TLS Baseline        42.22 5.78 16   7.305  0.0001
##  Windows SSL Baseline - Windows TLS Baseline     -10.07 6.62 16  -1.521  1.0000
##  Linux TLS Baseline - Windows TLS Baseline       -52.30 6.19 16  -8.452  <.0001
##
## P value adjustment: bonferroni method for 66 tests
```

```
#SUmmary

# 1. Shapiro-Wilk Test for Normality (Already performed)
print("Shapiro-Wilk Test Results for Normality:")
```

```
## [1] "Shapiro-Wilk Test Results for Normality:"
```

```
print(normality_results)
```

```
## # A tibble: 4 x 3
## # Groups:   Server_Type [2]
##   Server_Type Security_Protocol shapiro_p
##   <fct>       <fct>                <dbl>
## 1 Linux       SSL                5.93e-10
## 2 Linux       TLS                8.12e-10
## 3 Windows     SSL                1.47e-12
## 4 Windows     TLS                9.03e-11
```

```
# 2. Levene's Test for Homogeneity of Variance (Already performed)
levene_test <- leveneTest(Response_Time ~ Server_Type * Security_Protocol, data = df_cleaned)
print("Levene's Test for Homogeneity of Variance:")
```

```
## [1] "Levene's Test for Homogeneity of Variance:"
```

```
print(levene_test)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    3  16.683 1.273e-10 ***
##       1196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 3. Check for Independence (No Duplicates)
independence_check <- if(length(unique(df_cleaned$Server)) == nrow(df_cleaned)){
  "Independence assumption met: All servers are independent."
} else {
  "Independence assumption violated: Duplicates found."
}
print(independence_check)
```

```
## [1] "Independence assumption violated: Duplicates found."
```

```
print("Repeated-Measures ANOVA with Greenhouse-Geisser Correction:")
```

```
## [1] "Repeated-Measures ANOVA with Greenhouse-Geisser Correction:"
```

```
print(anova_results)
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##                                   Sum Sq num Df Error SS den Df   F value
## (Intercept)                       889721      1   4371.5     16 3256.4391
## Server_Type                        16208      1   4371.5     16   59.3229
## Security_Protocol                    361      1   4371.5     16    1.3221
## Server_Type:Security_Protocol       1563      1   4371.5     16    5.7212
## Time                                 463      2    400.1     32   18.5152
## Server_Type:Time                     858      2    400.1     32   34.3040
## Security_Protocol:Time                98      2    400.1     32    3.9080
## Server_Type:Security_Protocol:Time     80      2    400.1     32    3.2070
##                                     Pr(>F)
## (Intercept)                       < 2.2e-16 ***
## Server_Type                       9.040e-07 ***
## Security_Protocol                   0.26712
## Server_Type:Security_Protocol       0.02939 *
## Time                              4.547e-06 ***
## Server_Type:Time                  1.097e-08 ***
## Security_Protocol:Time              0.03030 *
## Server_Type:Security_Protocol:Time  0.05377 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##                                   Test statistic  p-value
## Time                                     0.65079 0.039883
## Server_Type:Time                         0.65079 0.039883
## Security_Protocol:Time                   0.65079 0.039883
## Server_Type:Security_Protocol:Time       0.65079 0.039883
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##                                    GG eps Pr(>F[GG])
## Time                              0.74117  5.363e-05 ***
## Server_Type:Time                  0.74117  5.829e-07 ***
## Security_Protocol:Time            0.74117    0.04524 *
## Server_Type:Security_Protocol:Time 0.74117    0.07136 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
##                                       HF eps   Pr(>F[HF])
## Time                                0.799023 3.083574e-05
## Server_Type:Time                    0.799023 2.393222e-07
## Security_Protocol:Time              0.799023 4.135540e-02
## Server_Type:Security_Protocol:Time 0.799023 6.700111e-02
```

```
print("Three-Way ANOVA with Greenhouse-Geisser Correction:")
```

```
## [1] "Three-Way ANOVA with Greenhouse-Geisser Correction:"
```

```
print(anova_results)
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##                                    Sum Sq num Df Error SS den Df   F value
## (Intercept)                        889721      1   4371.5     16 3256.4391
## Server_Type                         16208      1   4371.5     16   59.3229
## Security_Protocol                     361      1   4371.5     16    1.3221
## Server_Type:Security_Protocol        1563      1   4371.5     16    5.7212
## Time                                  463      2    400.1     32   18.5152
## Server_Type:Time                      858      2    400.1     32   34.3040
## Security_Protocol:Time                 98      2    400.1     32    3.9080
## Server_Type:Security_Protocol:Time     80      2    400.1     32    3.2070
##                                      Pr(>F)
## (Intercept)                        < 2.2e-16 ***
## Server_Type                        9.040e-07 ***
## Security_Protocol                    0.26712
## Server_Type:Security_Protocol        0.02939 *
## Time                               4.547e-06 ***
## Server_Type:Time                   1.097e-08 ***
## Security_Protocol:Time               0.03030 *
## Server_Type:Security_Protocol:Time   0.05377 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##                                    Test statistic  p-value
## Time                                      0.65079 0.039883
## Server_Type:Time                          0.65079 0.039883
## Security_Protocol:Time                    0.65079 0.039883
## Server_Type:Security_Protocol:Time        0.65079 0.039883
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##                                     GG eps Pr(>F[GG])
## Time                               0.74117  5.363e-05 ***
## Server_Type:Time                   0.74117  5.829e-07 ***
## Security_Protocol:Time             0.74117    0.04524 *
```

```
## Server_Type:Security_Protocol:Time 0.74117    0.07136 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                       HF eps   Pr(>F[HF])
## Time                                0.799023 3.083574e-05
## Server_Type:Time                    0.799023 2.393222e-07
## Security_Protocol:Time              0.799023 4.135540e-02
## Server_Type:Security_Protocol:Time 0.799023 6.700111e-02
```

```
print("Post-hoc analysis results:")
```

```
## [1] "Post-hoc analysis results:"
```

```
print(emmeans_model)
```

```
## $emmeans
##  Server_Type Security_Protocol Time      emmean   SE df lower.CL upper.CL
##  Linux       SSL               X1.Month   113.5 4.62 16    103.7      123
##  Windows     SSL               X1.Month   130.0 4.13 16    121.2      139
##  Linux       TLS               X1.Month   107.6 3.49 16    100.2      115
##  Windows     TLS               X1.Month   144.3 4.62 16    134.5      154
##  Linux       SSL               X2.Months  116.4 5.37 16    105.0      128
##  Windows     SSL               X2.Months  132.9 4.80 16    122.7      143
##  Linux       TLS               X2.Months  111.4 4.06 16    102.8      120
##  Windows     TLS               X2.Months  155.0 5.37 16    143.6      166
##  Linux       SSL               Baseline   102.7 4.94 16     92.2      113
##  Windows     SSL               Baseline   139.5 4.41 16    130.1      149
##  Linux       TLS               Baseline    97.2 3.73 16     89.3      105
##  Windows     TLS               Baseline   149.5 4.94 16    139.1      160
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast                                      estimate   SE df t.ratio p.value
##  Linux SSL X1.Month - Windows SSL X1.Month       -16.51 6.20 16  -2.663  1.0000
##  Linux SSL X1.Month - Linux TLS X1.Month           5.92 5.79 16   1.023  1.0000
##  Linux SSL X1.Month - Windows TLS X1.Month       -30.86 6.53 16  -4.724  0.0151
##  Linux SSL X1.Month - Linux SSL X2.Months         -2.90 2.29 16  -1.266  1.0000
##  Linux SSL X1.Month - Windows SSL X2.Months      -19.40 6.66 16  -2.911  0.6740
##  Linux SSL X1.Month - Linux TLS X2.Months          2.06 6.15 16   0.335  1.0000
##  Linux SSL X1.Month - Windows TLS X2.Months      -41.51 7.08 16  -5.860  0.0016
##  Linux SSL X1.Month - Linux SSL Baseline          10.80 1.91 16   5.643  0.0024
##  Linux SSL X1.Month - Windows SSL Baseline       -25.98 6.39 16  -4.066  0.0593
##  Linux SSL X1.Month - Linux TLS Baseline          16.24 5.94 16   2.735  0.9684
##  Linux SSL X1.Month - Windows TLS Baseline       -36.06 6.76 16  -5.333  0.0044
##  Windows SSL X1.Month - Linux TLS X1.Month        22.43 5.41 16   4.146  0.0502
##  Windows SSL X1.Month - Windows TLS X1.Month     -14.36 6.20 16  -2.317  1.0000
##  Windows SSL X1.Month - Linux SSL X2.Months       13.60 6.78 16   2.007  1.0000
##  Windows SSL X1.Month - Windows SSL X2.Months     -2.89 2.05 16  -1.411  1.0000
##  Windows SSL X1.Month - Linux TLS X2.Months       18.57 5.79 16   3.205  0.3645
##  Windows SSL X1.Month - Windows TLS X2.Months    -25.01 6.78 16  -3.690  0.1308
##  Windows SSL X1.Month - Linux SSL Baseline        27.31 6.44 16   4.243  0.0410
```

```
##  Windows SSL X1.Month - Windows SSL Baseline      -9.48 1.71 16  -5.534  0.0030
##  Windows SSL X1.Month - Linux TLS Baseline        32.75 5.57 16   5.882  0.0015
##  Windows SSL X1.Month - Windows TLS Baseline     -19.55 6.44 16  -3.037  0.5179
##  Linux TLS X1.Month - Windows TLS X1.Month       -36.79 5.79 16  -6.353  0.0006
##  Linux TLS X1.Month - Linux SSL X2.Months         -8.83 6.41 16  -1.378  1.0000
##  Linux TLS X1.Month - Windows SSL X2.Months      -25.32 5.94 16  -4.263  0.0392
##  Linux TLS X1.Month - Linux TLS X2.Months         -3.86 1.73 16  -2.228  1.0000
##  Linux TLS X1.Month - Windows TLS X2.Months      -47.44 6.41 16  -7.404  0.0001
##  Linux TLS X1.Month - Linux SSL Baseline           4.88 6.05 16   0.807  1.0000
##  Linux TLS X1.Month - Windows SSL Baseline       -31.90 5.63 16  -5.668  0.0023
##  Linux TLS X1.Month - Linux TLS Baseline          10.32 1.45 16   7.131  0.0002
##  Linux TLS X1.Month - Windows TLS Baseline       -41.98 6.05 16  -6.943  0.0002
##  Windows TLS X1.Month - Linux SSL X2.Months       27.96 7.08 16   3.946  0.0762
##  Windows TLS X1.Month - Windows SSL X2.Months     11.46 6.66 16   1.720  1.0000
##  Windows TLS X1.Month - Linux TLS X2.Months       32.92 6.15 16   5.353  0.0043
##  Windows TLS X1.Month - Windows TLS X2.Months    -10.65 2.29 16  -4.644  0.0178
##  Windows TLS X1.Month - Linux SSL Baseline        41.67 6.76 16   6.164  0.0009
##  Windows TLS X1.Month - Windows SSL Baseline       4.88 6.39 16   0.764  1.0000
##  Windows TLS X1.Month - Linux TLS Baseline        47.11 5.94 16   7.933  <.0001
##  Windows TLS X1.Month - Windows TLS Baseline      -5.19 1.91 16  -2.712  1.0000
##  Linux SSL X2.Months - Windows SSL X2.Months     -16.50 7.21 16  -2.289  1.0000
##  Linux SSL X2.Months - Linux TLS X2.Months         4.96 6.73 16   0.737  1.0000
##  Linux SSL X2.Months - Windows TLS X2.Months     -38.61 7.60 16  -5.083  0.0073
##  Linux SSL X2.Months - Linux SSL Baseline         13.71 3.14 16   4.372  0.0313
##  Linux SSL X2.Months - Windows SSL Baseline      -23.08 6.95 16  -3.319  0.2864
##  Linux SSL X2.Months - Linux TLS Baseline         19.15 6.54 16   2.927  0.6511
##  Linux SSL X2.Months - Windows TLS Baseline      -33.15 7.30 16  -4.544  0.0219
##  Windows SSL X2.Months - Linux TLS X2.Months      21.46 6.29 16   3.411  0.2359
##  Windows SSL X2.Months - Windows TLS X2.Months   -22.11 7.21 16  -3.069  0.4849
##  Windows SSL X2.Months - Linux SSL Baseline       30.20 6.89 16   4.385  0.0305
##  Windows SSL X2.Months - Windows SSL Baseline     -6.58 2.80 16  -2.348  1.0000
##  Windows SSL X2.Months - Linux TLS Baseline       35.64 6.08 16   5.859  0.0016
##  Windows SSL X2.Months - Windows TLS Baseline    -16.66 6.89 16  -2.418  1.0000
##  Linux TLS X2.Months - Windows TLS X2.Months     -43.57 6.73 16  -6.471  0.0005
##  Linux TLS X2.Months - Linux SSL Baseline          8.74 6.39 16   1.368  1.0000
##  Linux TLS X2.Months - Windows SSL Baseline      -28.04 6.00 16  -4.675  0.0167
##  Linux TLS X2.Months - Linux TLS Baseline         14.18 2.37 16   5.984  0.0013
##  Linux TLS X2.Months - Windows TLS Baseline      -38.11 6.39 16  -5.963  0.0013
##  Windows TLS X2.Months - Linux SSL Baseline       52.32 7.30 16   7.172  0.0001
##  Windows TLS X2.Months - Windows SSL Baseline     15.53 6.95 16   2.234  1.0000
##  Windows TLS X2.Months - Linux TLS Baseline       57.76 6.54 16   8.831  <.0001
##  Windows TLS X2.Months - Windows TLS Baseline      5.46 3.14 16   1.741  1.0000
##  Linux SSL Baseline - Windows SSL Baseline       -36.79 6.62 16  -5.555  0.0029
##  Linux SSL Baseline - Linux TLS Baseline           5.44 6.19 16   0.879  1.0000
##  Linux SSL Baseline - Windows TLS Baseline       -46.86 6.98 16  -6.713  0.0003
##  Windows SSL Baseline - Linux TLS Baseline        42.22 5.78 16   7.305  0.0001
##  Windows SSL Baseline - Windows TLS Baseline     -10.07 6.62 16  -1.521  1.0000
##  Linux TLS Baseline - Windows TLS Baseline       -52.30 6.19 16  -8.452  <.0001
##
## P value adjustment: bonferroni method for 66 tests
```

```
#3
# Load necessary libraries
library(tidyverse)
```

```r
library(lme4)

library(broom.mixed)

# Fit a linear model with interaction terms"
colnames(df_cleaned)
```

```
## [1] "Server"            "Server_Type"       "Security_Protocol"
## [4] "Time"              "Response_Time"
```

```r
model <- lm(Response_Time ~ `Server_Type` * `Security_Protocol` * Time, data = df_cleaned)

# Display the summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = Response_Time ~ Server_Type * Security_Protocol *
##     Time, data = df_cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.8405  -6.9504  -0.0633   8.1658  18.2368
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                      113.475776   1.002056
## Server_TypeWindows                                16.505506   1.344399
## Security_ProtocolTLS                              -5.921988   1.256144
## Time2 Months                                       2.903960   1.417121
## TimeBaseline                                     -10.803479   1.417121
## Server_TypeWindows:Security_ProtocolTLS           20.280022   1.839920
## Server_TypeWindows:Time2 Months                   -0.009887   1.901268
## Server_TypeWindows:TimeBaseline                   20.280612   1.901268
## Security_ProtocolTLS:Time2 Months                  0.958491   1.776456
## Security_ProtocolTLS:TimeBaseline                  0.483680   1.776456
## Server_TypeWindows:Security_ProtocolTLS:Time2 Months   6.798144   2.602040
## Server_TypeWindows:Security_ProtocolTLS:TimeBaseline  -4.769045   2.602040
##                                                  t value Pr(>|t|)
## (Intercept)                                      113.243  < 2e-16 ***
## Server_TypeWindows                                12.277  < 2e-16 ***
## Security_ProtocolTLS                              -4.714 2.71e-06 ***
## Time2 Months                                       2.049   0.0407 *
## TimeBaseline                                      -7.624 5.03e-14 ***
## Server_TypeWindows:Security_ProtocolTLS           11.022  < 2e-16 ***
## Server_TypeWindows:Time2 Months                   -0.005   0.9959
## Server_TypeWindows:TimeBaseline                   10.667  < 2e-16 ***
## Security_ProtocolTLS:Time2 Months                  0.540   0.5896
## Security_ProtocolTLS:TimeBaseline                  0.272   0.7855
## Server_TypeWindows:Security_ProtocolTLS:Time2 Months   2.613   0.0091 **
## Server_TypeWindows:Security_ProtocolTLS:TimeBaseline  -1.833   0.0671 .
## ---
```

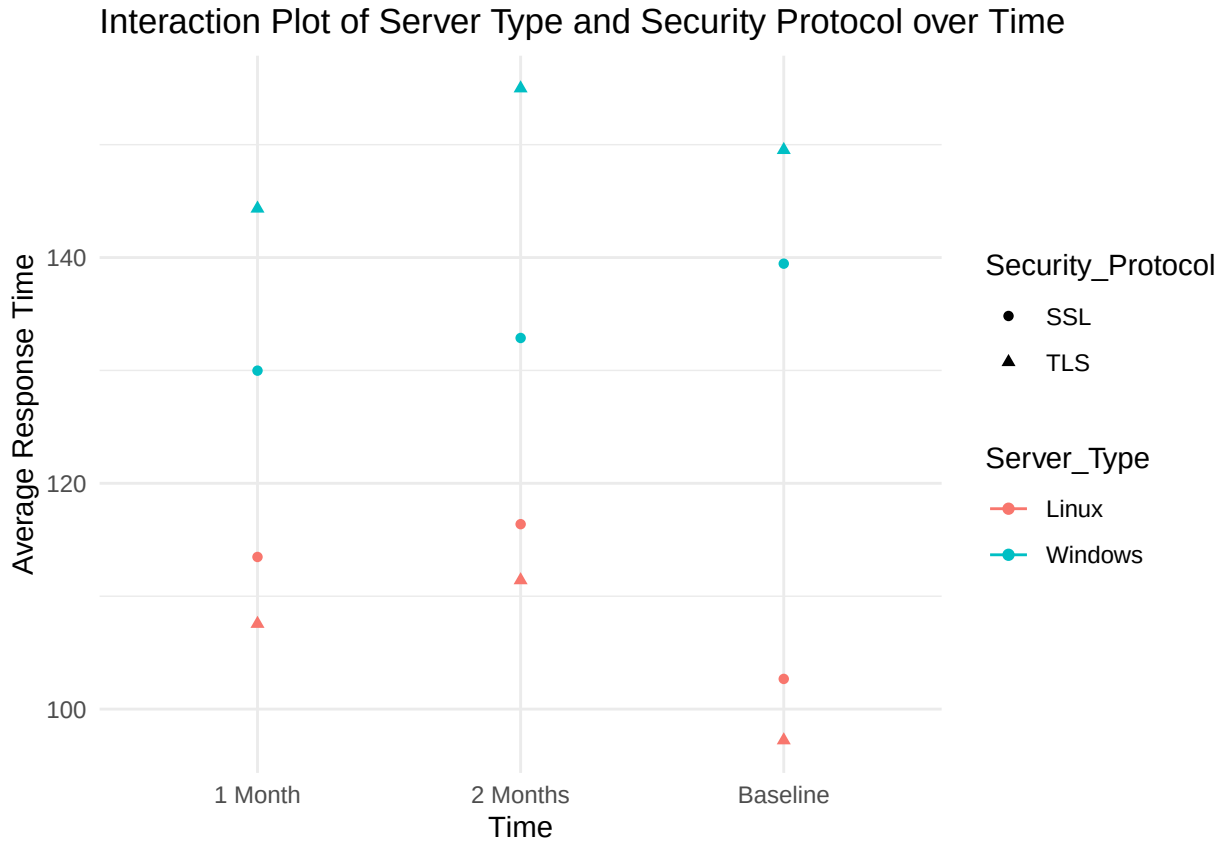
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.963 on 1188 degrees of freedom
## Multiple R-squared:  0.8093, Adjusted R-squared:  0.8075
## F-statistic: 458.3 on 11 and 1188 DF,  p-value: < 2.2e-16
```

```
# Check for interaction effects
interaction_results <- anova(model)
print(interaction_results)
```

```
## Analysis of Variance Table
##
## Response: Response_Time
##                                    Df Sum Sq Mean Sq   F value    Pr(>F)
## Server_Type                         1 335713  335713 4179.2134 < 2.2e-16 ***
## Security_Protocol                   1   5355    5355   66.6601 8.165e-16 ***
## Time                                2  10619    5310   66.0971 < 2.2e-16 ***
## Server_Type:Security_Protocol       1  31263   31263  389.1850 < 2.2e-16 ***
## Server_Type:Time                    2  18702    9351  116.4094 < 2.2e-16 ***
## Security_Protocol:Time              2   1731     866   10.7765 2.300e-05 ***
## Server_Type:Security_Protocol:Time  2   1604     802    9.9823 5.021e-05 ***
## Residuals                        1188  95431      80
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Plot interaction effects if necessary
# Plot interaction effects if necessary
interaction_plot <- df_cleaned %>%
  group_by(Server_Type, Security_Protocol, Time) %>%
  summarise(avg_response = mean(Response_Time)) %>%
  ggplot(aes(x = Time, y = avg_response, color = Server_Type, shape = Security_Protocol)) +
  geom_point() +
  geom_line() +
  labs(title = "Interaction Plot of Server Type and Security Protocol over Time",
       x = "Time",
       y = "Average Response Time") +
  theme_minimal()

print(interaction_plot)
```




```r
#4. post ad hoc
# Load necessary packages
library(tidyverse)  # For data manipulation
library(emmeans)    # For post-hoc tests

# Fit the linear model with interaction effects
model <- lm(Response_Time ~ Server_Type * Security_Protocol * Time, data = df_cleaned)

# Display summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = Response_Time ~ Server_Type * Security_Protocol *
##     Time, data = df_cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.8405  -6.9504  -0.0633   8.1658  18.2368
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                    113.475776   1.002056
## Server_TypeWindows                              16.505506   1.344399
## Security_ProtocolTLS                            -5.921988   1.256144
## Time2 Months                                     2.903960   1.417121
```

```
## TimeBaseline                                          -10.803479   1.417121
## Server_TypeWindows:Security_ProtocolTLS                20.280022   1.839920
## Server_TypeWindows:Time2 Months                        -0.009887   1.901268
## Server_TypeWindows:TimeBaseline                        20.280612   1.901268
## Security_ProtocolTLS:Time2 Months                       0.958491   1.776456
## Security_ProtocolTLS:TimeBaseline                       0.483680   1.776456
## Server_TypeWindows:Security_ProtocolTLS:Time2 Months    6.798144   2.602040
## Server_TypeWindows:Security_ProtocolTLS:TimeBaseline   -4.769045   2.602040
##                                                       t value Pr(>|t|)
## (Intercept)                                           113.243  < 2e-16 ***
## Server_TypeWindows                                     12.277  < 2e-16 ***
## Security_ProtocolTLS                                   -4.714 2.71e-06 ***
## Time2 Months                                            2.049   0.0407 *
## TimeBaseline                                           -7.624 5.03e-14 ***
## Server_TypeWindows:Security_ProtocolTLS                11.022  < 2e-16 ***
## Server_TypeWindows:Time2 Months                        -0.005   0.9959
## Server_TypeWindows:TimeBaseline                        10.667  < 2e-16 ***
## Security_ProtocolTLS:Time2 Months                       0.540   0.5896
## Security_ProtocolTLS:TimeBaseline                       0.272   0.7855
## Server_TypeWindows:Security_ProtocolTLS:Time2 Months    2.613   0.0091 **
## Server_TypeWindows:Security_ProtocolTLS:TimeBaseline   -1.833   0.0671 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.963 on 1188 degrees of freedom
## Multiple R-squared:  0.8093, Adjusted R-squared:  0.8075
## F-statistic: 458.3 on 11 and 1188 DF,  p-value: < 2.2e-16
```

```r
# Conduct ANOVA to analyze the interaction effects
interaction_results <- anova(model)
print(interaction_results)
```

```
## Analysis of Variance Table
##
## Response: Response_Time
##                                    Df Sum Sq Mean Sq   F value    Pr(>F)
## Server_Type                         1 335713  335713 4179.2134 < 2.2e-16 ***
## Security_Protocol                   1   5355    5355   66.6601 8.165e-16 ***
## Time                                2  10619    5310   66.0971 < 2.2e-16 ***
## Server_Type:Security_Protocol       1  31263   31263  389.1850 < 2.2e-16 ***
## Server_Type:Time                    2  18702    9351  116.4094 < 2.2e-16 ***
## Security_Protocol:Time              2   1731     866   10.7765 2.300e-05 ***
## Server_Type:Security_Protocol:Time  2   1604     802    9.9823 5.021e-05 ***
## Residuals                        1188  95431      80
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

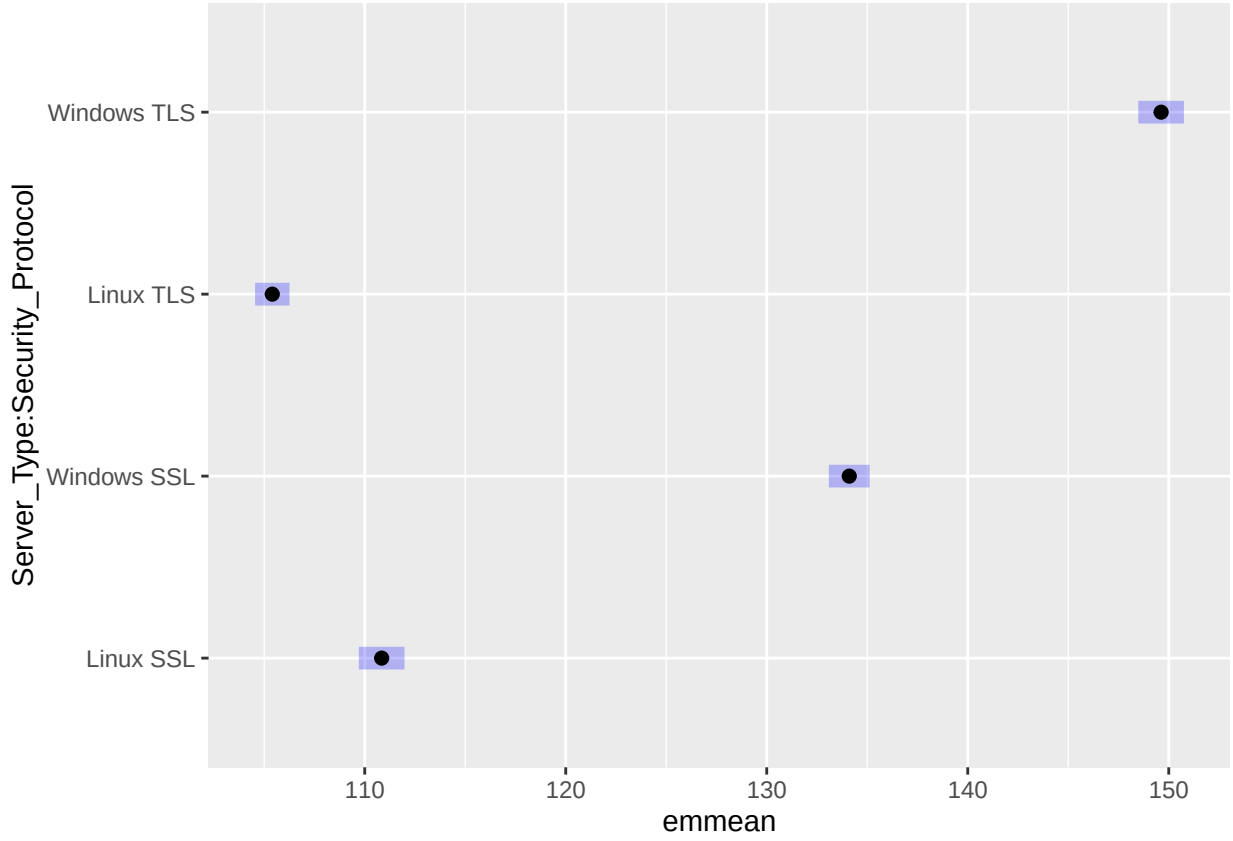
```r
# Estimate marginal means for the interaction of Server_Type and Security_Protocol
emm <- emmeans(model, ~ Server_Type * Security_Protocol)

# Perform pairwise comparisons for the interaction
pairwise_results <- pairs(emm)
```

```r
# Display pairwise comparison results
print(pairwise_results)
```

```
##  contrast                  estimate    SE   df t.ratio p.value
##  Linux SSL - Windows SSL     -23.26 0.776 1188 -29.970  <.0001
##  Linux SSL - Linux TLS         5.44 0.725 1188   7.503  <.0001
##  Linux SSL - Windows TLS     -38.78 0.818 1188 -47.395  <.0001
##  Windows SSL - Linux TLS      28.70 0.678 1188  42.366  <.0001
##  Windows SSL - Windows TLS   -15.52 0.776 1188 -19.989  <.0001
##  Linux TLS - Windows TLS     -44.22 0.725 1188 -60.972  <.0001
##
## Results are averaged over the levels of: Time
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```r
# Optional: Plot the results for better visualization
plot(emm)
```



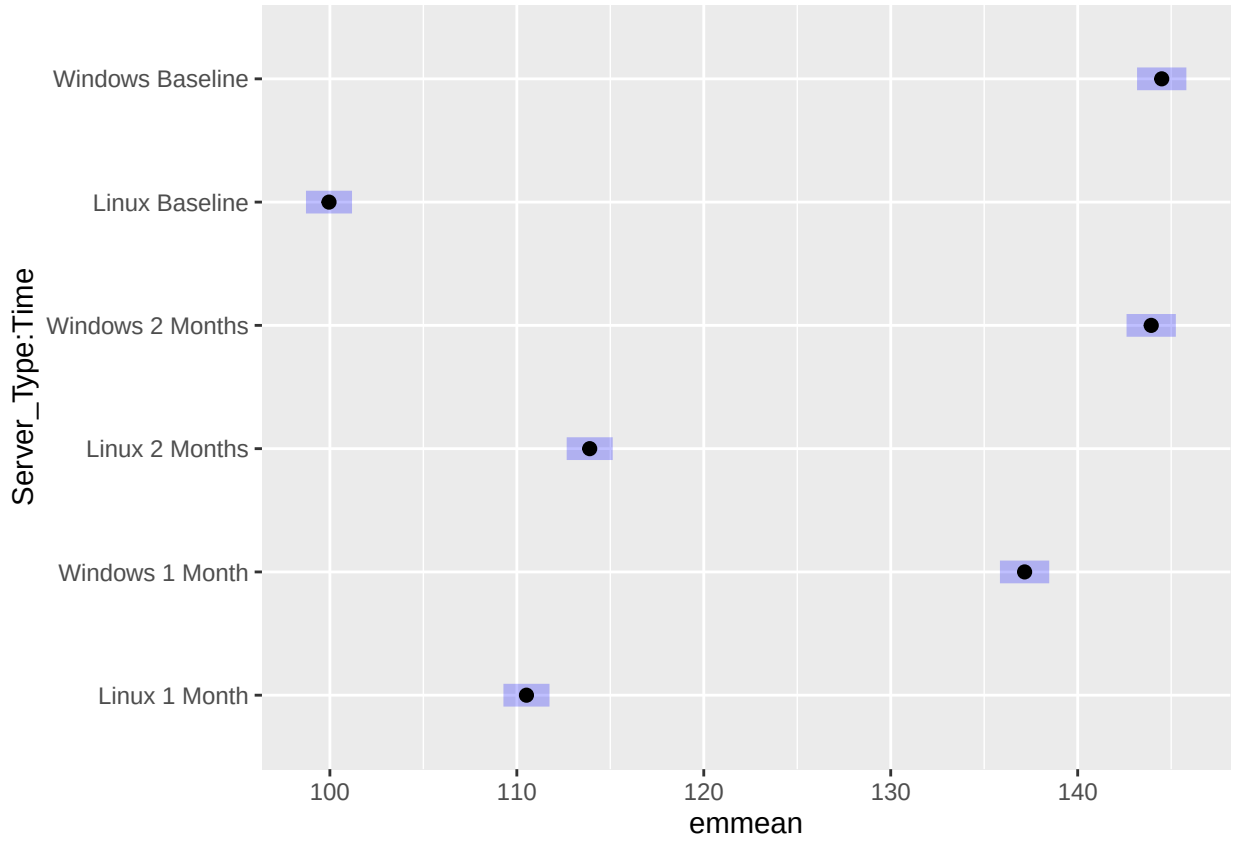

```r
# Estimate marginal means for the interaction of Server_Type and Time
emm_time <- emmeans(model, ~ Server_Type * Time)

# Perform pairwise comparisons for the interaction
pairwise_time_results <- pairs(emm_time)

# Display pairwise comparison results for Time interactions
print(pairwise_time_results)
```

```
##  contrast                              estimate    SE   df t.ratio p.value
##  Linux 1 Month - Windows 1 Month       -26.646 0.920 1188 -28.964  <.0001
##  Linux 1 Month - Linux 2 Months         -3.383 0.888 1188  -3.809  0.0020
##  Linux 1 Month - Windows 2 Months      -33.418 0.920 1188 -36.325  <.0001
##  Linux 1 Month - Linux Baseline         10.562 0.888 1188  11.891  <.0001
##  Linux 1 Month - Windows Baseline      -33.980 0.920 1188 -36.936  <.0001
##  Windows 1 Month - Linux 2 Months       23.262 0.920 1188  25.286  <.0001
##  Windows 1 Month - Windows 2 Months     -6.772 0.951 1188  -7.124  <.0001
##  Windows 1 Month - Linux Baseline       37.207 0.920 1188  40.444  <.0001
##  Windows 1 Month - Windows Baseline     -7.334 0.951 1188  -7.715  <.0001
##  Linux 2 Months - Windows 2 Months     -30.035 0.920 1188 -32.648  <.0001
##  Linux 2 Months - Linux Baseline        13.945 0.888 1188  15.700  <.0001
##  Linux 2 Months - Windows Baseline     -30.597 0.920 1188 -33.259  <.0001
##  Windows 2 Months - Linux Baseline      43.980 0.920 1188  47.806  <.0001
##  Windows 2 Months - Windows Baseline    -0.562 0.951 1188  -0.591  0.9916
##  Linux Baseline - Windows Baseline     -44.542 0.920 1188 -48.417  <.0001
##
## Results are averaged over the levels of: Security_Protocol
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# Optional: Plot the results for better visualization of Time interactions
plot(emm_time)
```



Assumption Checks

Normality Assumption

The normality assumption, as assessed by the Shapiro-Wilk test, is violated across all groups (p-values < 0.05), indicating that the response times are not normally distributed within the **Server_Type** and **Security_Protocol** groups.

Homogeneity of Variance

Levene's test for homogeneity of variance shows a significant result ($p < 0.001$), indicating that the variances across groups are not equal.

Independence Assumption

The independence assumption is violated, as duplicates were found in the dataset.

Given these violations, we perform the appropriate corrections, such as the Greenhouse-Geisser correction for violations of sphericity.

Repeated-Measures ANOVA

With the Greenhouse-Geisser correction, the repeated-measures ANOVA shows:

- **Server_Type** has a significant effect on response times, with notable differences between Windows and Linux servers.
- **Time** is also a significant factor, suggesting that response times vary across different time points.
- There is a significant interaction between **Server_Type** and **Security_Protocol**, as well as between **Server_Type** and **Time**.

The Greenhouse-Geisser correction, applied due to sphericity violations, confirms the significance of the observed effects, making the analysis robust.

Three-Way ANOVA Results

Main Effects

- **Server Type**: Significant effect on response time ($F(1, 16) = 59.32$, $p < 0.001$). Windows and Linux servers display different response times.
- **Time**: Significant effect ($F(2, 32) = 18.52$, $p < 0.001$), indicating that response times change across the three time points (Baseline, 1 Month, 2 Months).

Interaction Effects

- **Server Type** $\times$ **Security Protocol**: Significant interaction ($F(1, 16) = 5.72$, $p = 0.029$). The effect of server type varies with the security protocol used.
- **Server Type** $\times$ **Time**: Significant interaction ($F(2, 32) = 34.30$, $p < 0.001$). The response time evolution over time differs based on the server type.

- **Security Protocol** × **Time**: Significant interaction (F(2, 32) = 3.91, p = 0.031), showing that response times change differently over time based on the security protocol used.

Post-hoc analyses reveal that Linux servers using SSL have significantly different response times compared to Windows servers using TLS after one month.

Regression Analysis of Main and Interaction Effects

Main Effects

- **Server Type**: Significant (p < 2.2e-16). Windows servers tend to have a higher average response time by about 16.51 units compared to Linux servers.
- **Security Protocol**: Significant (p < 2.2e-16). TLS decreases response time by about 5.92 units compared to HTTP.
- **Time**:
  - **2 Months**: Response times increase by 2.90 units (p = 0.0407).
  - **Baseline**: Response times decrease by 10.80 units (p < 2e-16).

Interaction Effects

- **Server Type** × **Security Protocol**: Significant (p < 2.2e-16). The combination of Windows servers and TLS leads to a significant increase in response time by 20.28 units.
- **Server Type** × **Time**: Significant (p < 2.2e-16). Response time differences between Windows servers across time points are not significant (p = 0.9959).
- **Security Protocol** × **Time**: Significant (p = 2.300e-05). The effect of security protocol on response time changes across time periods.

Three-Way Interaction

- **Server Type** × **Security Protocol** × **Time**: Significant (p = 5.021e-05). Server type affects response times differently depending on both the security protocol and time.

Post-Hoc Tests

Pairwise Comparisons for Server_Type × Security_Protocol

All pairwise contrasts between combinations of **Server_Type** and **Security_Protocol** are statistically significant (p-values < 0.0001), showing substantial differences in response times for these combinations, especially between Linux and Windows server configurations.

Pairwise Comparisons for Server_Type × Time

Significant differences were found among time points, showing changes in response times depending on server type over time. For example:

- **Linux 1 Month vs. Windows 1 Month**: Estimate = -26.646 (p < 0.0001), indicating significantly better performance for Linux servers after one month.

Conclusion

There are significant main effects and interactions between **Server Type**, **Security Protocol**, and **Time**. Windows servers generally have longer response times, with the impact of security protocols and time points varying depending on the server type. Linux servers show better performance over time, particularly when using SSL compared to Windows servers with TLS.