bikeshare_analysis.r

root

2025-10-21

```
# BikeShare Analysis: Regression Models for Predicting Number of Bikers
# Using ISLR2 package dataset

# Load required packages
install.packages("ISLR2", quiet = TRUE)
library(ISLR2)

# Load the BikeShare dataset
data(Bikeshare)

# (a) Summarize the dataset considering only the variables needed for the regression
# Variables needed: bikers, mnth, hr, workingday, weathersit, temp

# Select only the required variables
bikeshare_subset <- Bikeshare[, c("bikers", "mnth", "hr", "workingday", "weathersit", "temp")]

# Display basic information about the dataset
cat("Dataset dimensions:", dim(bikeshare_subset), "\n")
```

```
## Dataset dimensions: 8645 6
```

```
cat("Number of observations:", nrow(bikeshare_subset), "\n")
```

```
## Number of observations: 8645
```

```
cat("Number of variables:", ncol(bikeshare_subset), "\n")
```

```
## Number of variables: 6
```

```
# Summary statistics
print("Summary of selected variables:")
```

```
## [1] "Summary of selected variables:"
```

```
print(summary(bikeshare_subset))
```

```
##      bikers           mnth            hr         workingday
##  Min.   :  1.0   May    : 744   16     : 365   Min.   :0.0000
##  1st Qu.: 31.0   July   : 744   17     : 365   1st Qu.:0.0000
##  Median :109.0   Oct    : 743   12     : 364   Median :1.0000
##  Mean   :143.8   Dec    : 741   13     : 364   Mean   :0.6837
##  3rd Qu.:211.0   Aug    : 731   14     : 364   3rd Qu.:1.0000
##  Max.   :651.0   March  : 730   15     : 364   Max.   :1.0000
##                  (Other):4212   (Other):6459
##            weathersit        temp
##  clear          :5645   Min.   :0.0200
```

```
##  cloudy/misty   :2218   1st Qu.:0.3200
##  light rain/snow: 781   Median :0.5000
##  heavy rain/snow:   1   Mean   :0.4891
##                         3rd Qu.:0.6600
##                         Max.   :0.9600
##
```

```
# Check structure and levels of categorical variables
print("Structure of categorical variables:")
```

```
## [1] "Structure of categorical variables:"
```

```
print("Month levels:")
```

```
## [1] "Month levels:"
```

```
print(levels(bikeshare_subset$mnth))
```

```
##  [1] "Jan"   "Feb"   "March" "April" "May"   "June"  "July"  "Aug"   "Sept"
## [10] "Oct"   "Nov"   "Dec"
```

```
print("Hour levels:")
```

```
## [1] "Hour levels:"
```

```
print(levels(bikeshare_subset$hr))
```

```
##  [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14"
## [16] "15" "16" "17" "18" "19" "20" "21" "22" "23"
```

```
print("Working day levels:")
```

```
## [1] "Working day levels:"
```

```
print(levels(bikeshare_subset$workingday))
```

```
## NULL
```

```
print("Weather situation levels:")
```

```
## [1] "Weather situation levels:"
```

```
print(levels(bikeshare_subset$weathersit))
```

```
## [1] "clear"           "cloudy/misty"    "light rain/snow" "heavy rain/snow"
```

```
# Transform workingday to create a categorical variable
# workingday is numeric (0/1), convert to categorical
bikeshare_subset$workingday_cat <- ifelse(bikeshare_subset$workingday == 1, "Working Day", "Non-Working
bikeshare_subset$workingday_cat <- as.factor(bikeshare_subset$workingday_cat)

# Transform weathersit to combine rain/snow levels
# Original levels: Clear, Misty, Light Rain/Snow, Heavy Rain/Snow
# Combine "Light Rain/Snow" and "Heavy Rain/Snow" into "Rain/Snow"
bikeshare_subset$weathersit_transformed <- as.character(bikeshare_subset$weathersit)
bikeshare_subset$weathersit_transformed[bikeshare_subset$weathersit_transformed %in% c("Light Rain/Snow
bikeshare_subset$weathersit_transformed <- as.factor(bikeshare_subset$weathersit_transformed)

# Remove original variables
bikeshare_subset$workingday <- NULL
bikeshare_subset$weathersit <- NULL
```

```r
# Display transformed variable levels
print("Transformed weather situation levels:")
```

```
## [1] "Transformed weather situation levels:"
```

```r
print(levels(bikeshare_subset$weathersit_transformed))
```

```
## [1] "clear"           "cloudy/misty"    "heavy rain/snow" "light rain/snow"
```

```r
print("Transformed working day levels:")
```

```
## [1] "Transformed working day levels:"
```

```r
print(levels(bikeshare_subset$workingday_cat))
```

```
## [1] "Non-Working Day" "Working Day"
```

```r
# Final dataset summary
print("Final dataset summary:")
```

```
## [1] "Final dataset summary:"
```

```r
print(summary(bikeshare_subset))
```

```
##      bikers           mnth             hr             temp
##  Min.   :  1.0   May    : 744   16     : 365   Min.   :0.0200
##  1st Qu.: 31.0   July   : 744   17     : 365   1st Qu.:0.3200
##  Median :109.0   Oct    : 743   12     : 364   Median :0.5000
##  Mean   :143.8   Dec    : 741   13     : 364   Mean   :0.4891
##  3rd Qu.:211.0   Aug    : 731   14     : 364   3rd Qu.:0.6600
##  Max.   :651.0   March  : 730   15     : 364   Max.   :0.9600
##                  (Other):4212   (Other):6459
##          workingday_cat     weathersit_transformed
##  Non-Working Day:2734   clear          :5645
##  Working Day    :5911   cloudy/misty   :2218
##                         heavy rain/snow:   1
##                         light rain/snow: 781
##
##
##
```

```r
# (b) Fit three regression models
# Model 1: Linear regression model to predict total number of bikers
model_linear <- lm(bikers ~ mnth + hr + workingday_cat + weathersit_transformed + temp,
                   data = bikeshare_subset)

print("Linear Regression Model Summary:")
```

```
## [1] "Linear Regression Model Summary:"
```

```r
print(summary(model_linear))
```

```
##
## Call:
## lm(formula = bikers ~ mnth + hr + workingday_cat + weathersit_transformed +
##     temp, data = bikeshare_subset)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -299.00  -45.70   -6.23   41.08  425.29
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                               -68.632      5.307 -12.932  < 2e-16 ***
## mnthFeb                                     6.845      4.287   1.597 0.110398
## mnthMarch                                  16.551      4.301   3.848 0.000120 ***
## mnthApril                                  41.425      4.972   8.331  < 2e-16 ***
## mnthMay                                    72.557      5.641  12.862  < 2e-16 ***
## mnthJune                                   67.819      6.544  10.364  < 2e-16 ***
## mnthJuly                                   45.324      7.081   6.401 1.63e-10 ***
## mnthAug                                    53.243      6.640   8.019 1.21e-15 ***
## mnthSept                                   66.678      5.925  11.254  < 2e-16 ***
## mnthOct                                    75.834      4.950  15.319  < 2e-16 ***
## mnthNov                                    60.310      4.610  13.083  < 2e-16 ***
## mnthDec                                    46.458      4.271  10.878  < 2e-16 ***
## hr1                                       -14.579      5.699  -2.558 0.010536 *
## hr2                                       -21.579      5.733  -3.764 0.000168 ***
## hr3                                       -31.141      5.778  -5.389 7.26e-08 ***
## hr4                                       -36.908      5.802  -6.361 2.11e-10 ***
## hr5                                       -24.135      5.737  -4.207 2.61e-05 ***
## hr6                                        20.600      5.704   3.612 0.000306 ***
## hr7                                       120.093      5.693  21.095  < 2e-16 ***
## hr8                                       223.662      5.690  39.310  < 2e-16 ***
## hr9                                       120.582      5.693  21.182  < 2e-16 ***
## hr10                                       83.801      5.705  14.689  < 2e-16 ***
## hr11                                      105.423      5.722  18.424  < 2e-16 ***
## hr12                                      137.284      5.740  23.916  < 2e-16 ***
## hr13                                      136.036      5.760  23.617  < 2e-16 ***
## hr14                                      126.636      5.776  21.923  < 2e-16 ***
## hr15                                      132.087      5.780  22.852  < 2e-16 ***
## hr16                                      178.521      5.772  30.927  < 2e-16 ***
## hr17                                      296.267      5.749  51.537  < 2e-16 ***
## hr18                                      269.441      5.736  46.976  < 2e-16 ***
## hr19                                      186.256      5.714  32.596  < 2e-16 ***
## hr20                                      125.549      5.704  22.012  < 2e-16 ***
## hr21                                       87.554      5.693  15.378  < 2e-16 ***
## hr22                                       59.123      5.689  10.392  < 2e-16 ***
## hr23                                       26.838      5.688   4.719 2.41e-06 ***
## workingday_catWorking Day                   1.270      1.784   0.711 0.476810
## weathersit_transformedcloudy/misty        -12.890      1.964  -6.562 5.60e-11 ***
## weathersit_transformedheavy rain/snow    -109.745     76.667  -1.431 0.152341
## weathersit_transformedlight rain/snow     -66.494      2.965 -22.425  < 2e-16 ***
## temp                                      157.209     10.261  15.321  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.5 on 8605 degrees of freedom
## Multiple R-squared:  0.6745, Adjusted R-squared:  0.6731
## F-statistic: 457.3 on 39 and 8605 DF,  p-value: < 2.2e-16
```

```
# Model 2: Linear regression model to predict logarithmized total number of bikers
# Add small constant to avoid log(0) issues
```

```r
bikeshare_subset$log_bikers <- log(bikeshare_subset$bikers + 1)

model_log_linear <- lm(log_bikers ~ mnth + hr + workingday_cat + weathersit_transformed + temp,
                       data = bikeshare_subset)

print("Log-Linear Regression Model Summary:")
```

```
## [1] "Log-Linear Regression Model Summary:"
```

```r
print(summary(model_log_linear))
```

```
##
## Call:
## lm(formula = log_bikers ~ mnth + hr + workingday_cat + weathersit_transformed +
##     temp, data = bikeshare_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6272 -0.2964  0.0378  0.3675  2.4218
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         2.49105    0.04120  60.457  < 2e-16 ***
## mnthFeb                             0.21278    0.03329   6.392 1.72e-10 ***
## mnthMarch                           0.29039    0.03340   8.695  < 2e-16 ***
## mnthApril                           0.58981    0.03861  15.278  < 2e-16 ***
## mnthMay                             0.85320    0.04380  19.479  < 2e-16 ***
## mnthJune                            0.80063    0.05081  15.759  < 2e-16 ***
## mnthJuly                            0.70531    0.05498  12.828  < 2e-16 ***
## mnthAug                             0.71312    0.05155  13.833  < 2e-16 ***
## mnthSept                            0.80276    0.04600  17.450  < 2e-16 ***
## mnthOct                             0.85745    0.03844  22.309  < 2e-16 ***
## mnthNov                             0.75172    0.03579  21.002  < 2e-16 ***
## mnthDec                             0.59763    0.03316  18.023  < 2e-16 ***
## hr1                                -0.57071    0.04425 -12.898  < 2e-16 ***
## hr2                                -1.00746    0.04451 -22.633  < 2e-16 ***
## hr3                                -1.49877    0.04486 -33.407  < 2e-16 ***
## hr4                                -1.78980    0.04505 -39.729  < 2e-16 ***
## hr5                                -0.96909    0.04454 -21.758  < 2e-16 ***
## hr6                                 0.19566    0.04429   4.418 1.01e-05 ***
## hr7                                 1.13062    0.04420  25.579  < 2e-16 ***
## hr8                                 1.77996    0.04418  40.293  < 2e-16 ***
## hr9                                 1.49722    0.04420  33.875  < 2e-16 ***
## hr10                                1.19169    0.04430  26.903  < 2e-16 ***
## hr11                                1.31591    0.04443  29.619  < 2e-16 ***
## hr12                                1.50493    0.04457  33.767  < 2e-16 ***
## hr13                                1.49798    0.04472  33.495  < 2e-16 ***
## hr14                                1.43455    0.04485  31.987  < 2e-16 ***
## hr15                                1.47502    0.04488  32.868  < 2e-16 ***
## hr16                                1.72562    0.04482  38.504  < 2e-16 ***
## hr17                                2.13736    0.04463  47.887  < 2e-16 ***
## hr18                                2.04523    0.04453  45.926  < 2e-16 ***
## hr19                                1.75522    0.04436  39.563  < 2e-16 ***
## hr20                                1.46066    0.04428  32.984  < 2e-16 ***
```

```
## hr21                                    1.19904    0.04420  27.125  < 2e-16 ***
## hr22                                    0.96222    0.04417  21.782  < 2e-16 ***
## hr23                                    0.56496    0.04416  12.793  < 2e-16 ***
## workingday_catWorking Day              -0.01233    0.01385  -0.890    0.374
## weathersit_transformedcloudy/misty     -0.06026    0.01525  -3.951 7.83e-05 ***
## weathersit_transformedheavy rain/snow -0.82256    0.59526  -1.382    0.167
## weathersit_transformedlight rain/snow -0.66024    0.02302 -28.678  < 2e-16 ***
## temp                                    1.04155    0.07967  13.073  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.594 on 8605 degrees of freedom
## Multiple R-squared:  0.8135, Adjusted R-squared:  0.8126
## F-statistic: 962.4 on 39 and 8605 DF,  p-value: < 2.2e-16
```

```
# Model 3: Poisson regression model with log-link
# Check for potential issues before fitting
cat("Checking data for Poisson regression:\n")
```

```
## Checking data for Poisson regression:
```

```
cat("Minimum bikers value:", min(bikeshare_subset$bikers), "\n")
```

```
## Minimum bikers value: 1
```

```
cat("Maximum bikers value:", max(bikeshare_subset$bikers), "\n")
```

```
## Maximum bikers value: 651
```

```
cat("Mean bikers value:", mean(bikeshare_subset$bikers), "\n")
```

```
## Mean bikers value: 143.7944
```

```
cat("Variance bikers value:", var(bikeshare_subset$bikers), "\n")
```

```
## Variance bikers value: 17901.87
```

```
cat("Dispersion ratio (variance/mean):", var(bikeshare_subset$bikers)/mean(bikeshare_subset$bikers), "\n")
```

```
## Dispersion ratio (variance/mean): 124.4962
```

```
# Check for overdispersion
if(var(bikeshare_subset$bikers)/mean(bikeshare_subset$bikers) > 1.5) {
  cat("WARNING: Overdispersion detected! Consider negative binomial regression.\n")
}
```

```
## WARNING: Overdispersion detected! Consider negative binomial regression.
```

```
# Try Poisson regression with error handling
tryCatch({
  model_poisson <- glm(bikers ~ mnth + hr + workingday_cat + weathersit_transformed + temp,
                       data = bikeshare_subset, family = poisson(link = "log"))

  print("Poisson Regression Model Summary:")
  print(summary(model_poisson))

  # Check for convergence
  if(!model_poisson$converged) {
    cat("WARNING: Poisson model did not converge!\n")
  }
```

```
}, error = function(e) {
  cat("ERROR in Poisson regression:", e$message, "\n")
  cat("This is likely due to overdispersion or convergence issues.\n")

  # Try negative binomial regression instead
  cat("Attempting negative binomial regression...\n")

  # Install and load MASS package for negative binomial
  if(!require(MASS, quietly = TRUE)) {
    install.packages("MASS", quiet = TRUE)
    library(MASS)
  }

  model_poisson <<- glm.nb(bikers ~ mnth + hr + workingday_cat + weathersit_transformed + temp,
                           data = bikeshare_subset)

  print("Negative Binomial Regression Model Summary:")
  print(summary(model_poisson))
})
```

```
## [1] "Poisson Regression Model Summary:"
##
## Call:
## glm(formula = bikers ~ mnth + hr + workingday_cat + weathersit_transformed +
##     temp, family = poisson(link = "log"), data = bikeshare_subset)
##
## Coefficients:
##                                    Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                        2.693688   0.009720  277.124  < 2e-16
## mnthFeb                            0.226046   0.006951   32.521  < 2e-16
## mnthMarch                          0.376437   0.006691   56.263  < 2e-16
## mnthApril                          0.691693   0.006987   98.996  < 2e-16
## mnthMay                            0.910641   0.007436  122.469  < 2e-16
## mnthJune                           0.893405   0.008242  108.402  < 2e-16
## mnthJuly                           0.773787   0.008806   87.874  < 2e-16
## mnthAug                            0.821341   0.008332   98.573  < 2e-16
## mnthSept                           0.903663   0.007621  118.578  < 2e-16
## mnthOct                            0.937743   0.006744  139.054  < 2e-16
## mnthNov                            0.820433   0.006494  126.334  < 2e-16
## mnthDec                            0.686850   0.006317  108.724  < 2e-16
## hr1                               -0.471593   0.012999  -36.278  < 2e-16
## hr2                               -0.808761   0.014646  -55.220  < 2e-16
## hr3                               -1.443918   0.018843  -76.631  < 2e-16
## hr4                               -2.076098   0.024796  -83.728  < 2e-16
## hr5                               -1.060271   0.016075  -65.957  < 2e-16
## hr6                                0.324498   0.010610   30.585  < 2e-16
## hr7                                1.329567   0.009056  146.822  < 2e-16
## hr8                                1.831313   0.008653  211.630  < 2e-16
## hr9                                1.336155   0.009016  148.191  < 2e-16
## hr10                               1.091238   0.009261  117.831  < 2e-16
## hr11                               1.248507   0.009093  137.304  < 2e-16
## hr12                               1.434028   0.008936  160.486  < 2e-16
## hr13                               1.427951   0.008951  159.529  < 2e-16
```

```
## hr14                                   1.379296   0.008999  153.266  < 2e-16
## hr15                                   1.408149   0.008977  156.862  < 2e-16
## hr16                                   1.628688   0.008805  184.979  < 2e-16
## hr17                                   2.049021   0.008565  239.221  < 2e-16
## hr18                                   1.966668   0.008586  229.065  < 2e-16
## hr19                                   1.668409   0.008743  190.830  < 2e-16
## hr20                                   1.370588   0.008973  152.737  < 2e-16
## hr21                                   1.118568   0.009215  121.383  < 2e-16
## hr22                                   0.871879   0.009536   91.429  < 2e-16
## hr23                                   0.481387   0.010207   47.164  < 2e-16
## workingday_catWorking Day              0.014665   0.001955    7.502 6.27e-14
## weathersit_transformedcloudy/misty    -0.075231   0.002179  -34.528  < 2e-16
## weathersit_transformedheavy rain/snow -0.926287   0.166782   -5.554 2.79e-08
## weathersit_transformedlight rain/snow -0.575800   0.004058 -141.905  < 2e-16
## temp                                   0.785292   0.011475   68.434  < 2e-16
##
## (Intercept)                           ***
## mnthFeb                               ***
## mnthMarch                             ***
## mnthApril                             ***
## mnthMay                               ***
## mnthJune                              ***
## mnthJuly                              ***
## mnthAug                               ***
## mnthSept                              ***
## mnthOct                               ***
## mnthNov                               ***
## mnthDec                               ***
## hr1                                   ***
## hr2                                   ***
## hr3                                   ***
## hr4                                   ***
## hr5                                   ***
## hr6                                   ***
## hr7                                   ***
## hr8                                   ***
## hr9                                   ***
## hr10                                  ***
## hr11                                  ***
## hr12                                  ***
## hr13                                  ***
## hr14                                  ***
## hr15                                  ***
## hr16                                  ***
## hr17                                  ***
## hr18                                  ***
## hr19                                  ***
## hr20                                  ***
## hr21                                  ***
## hr22                                  ***
## hr23                                  ***
## workingday_catWorking Day             ***
## weathersit_transformedcloudy/misty    ***
## weathersit_transformedheavy rain/snow ***
```

```
## weathersit_transformedlight rain/snow ***
## temp                                  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1052921  on 8644  degrees of freedom
## Residual deviance:  228041  on 8605  degrees of freedom
## AIC: 281159
##
## Number of Fisher Scoring iterations: 5
```

```
# (c) Extract regression coefficients for temp and interpret them
# Extract temp coefficients safely
temp_coeff_linear <- coef(model_linear)["temp"]
temp_coeff_log_linear <- coef(model_log_linear)["temp"]

# Handle Poisson model coefficient extraction
if(exists("model_poisson") && !is.null(model_poisson)) {
  tryCatch({
    temp_coeff_poisson <- coef(model_poisson)["temp"]
  }, error = function(e) {
    cat("Error extracting Poisson coefficient:", e$message, "\n")
    temp_coeff_poisson <<- 0  # Fallback to zero
  })
} else {
  temp_coeff_poisson <- 0
}

print("Temperature Coefficients:")
```

```
## [1] "Temperature Coefficients:"
```

```
print(paste("Linear model:", round(temp_coeff_linear, 4)))
```

```
## [1] "Linear model: 157.2094"
```

```
print(paste("Log-linear model:", round(temp_coeff_log_linear, 4)))
```

```
## [1] "Log-linear model: 1.0415"
```

```
print(paste("Poisson model:", round(temp_coeff_poisson, 4)))
```

```
## [1] "Poisson model: 0.7853"
```

```
# Interpretation
cat("\nInterpretation of Temperature Coefficients:\n")
```

```
##
## Interpretation of Temperature Coefficients:
```

```
cat("Linear model: A 1-unit increase in temperature is associated with",
    round(temp_coeff_linear, 2), "additional bikers on average.\n")
```

```
## Linear model: A 1-unit increase in temperature is associated with 157.21 additional bikers on averag
```

```
cat("Log-linear model: A 1-unit increase in temperature is associated with a",
    round(temp_coeff_log_linear * 100, 2), "% change in the number of bikers.\n")
```

```
## Log-linear model: A 1-unit increase in temperature is associated with a 104.15 % change in the numbe
```

```
cat("Poisson model: A 1-unit increase in temperature multiplies the expected number of bikers by",
    round(exp(temp_coeff_poisson), 4), ".\n")
```

```
## Poisson model: A 1-unit increase in temperature multiplies the expected number of bikers by 2.193 .
```

```
# (d) Compare model fit
# Calculate predictions for all models
pred_linear <- predict(model_linear)
pred_log_linear <- exp(predict(model_log_linear)) - 1  # Transform back from log scale

# Handle Poisson model predictions safely
if(exists("model_poisson") && !is.null(model_poisson)) {
  tryCatch({
    pred_poisson <- predict(model_poisson, type = "response")
  }, error = function(e) {
    cat("Error predicting from Poisson model:", e$message, "\n")
    pred_poisson <<- rep(0, nrow(bikeshare_subset))  # Fallback to zeros
  })
} else {
  cat("Poisson model not available, using zeros for comparison\n")
  pred_poisson <- rep(0, nrow(bikeshare_subset))
}

# Calculate Poisson log-likelihood based on predicted number of bikers
# For Poisson distribution: log-likelihood = sum(y * log(lambda) - lambda - log(y!))
# where lambda is the predicted value and y is the observed value

# Function to calculate Poisson log-likelihood
poisson_loglik <- function(observed, predicted) {
  # Add small constant to avoid log(0)
  predicted_adj <- pmax(predicted, 1e-10)
  observed_adj <- pmax(observed, 1e-10)

  # Calculate log-likelihood
  loglik <- sum(observed * log(predicted_adj) - predicted_adj - log(factorial(observed)))
  return(loglik)
}

# Calculate log-likelihoods
ll_linear <- poisson_loglik(bikeshare_subset$bikers, pred_linear)
ll_log_linear <- poisson_loglik(bikeshare_subset$bikers, pred_log_linear)
ll_poisson <- poisson_loglik(bikeshare_subset$bikers, pred_poisson)

print("Poisson Log-Likelihoods:")
```

```
## [1] "Poisson Log-Likelihoods:"
```

```
print(paste("Linear model:", round(ll_linear, 2)))
```

```
## [1] "Linear model: -Inf"
```

```
print(paste("Log-linear model:", round(ll_log_linear, 2)))
```

```
## [1] "Log-linear model: -Inf"
```

```r
print(paste("Poisson model:", round(ll_poisson, 2)))
```

```
## [1] "Poisson model: -Inf"
```

```r
# Calculate Mean Squared Error
mse_linear <- mean((bikeshare_subset$bikers - pred_linear)^2)
mse_log_linear <- mean((bikeshare_subset$bikers - pred_log_linear)^2)
mse_poisson <- mean((bikeshare_subset$bikers - pred_poisson)^2)

print("Mean Squared Errors:")
```

```
## [1] "Mean Squared Errors:"
```

```r
print(paste("Linear model:", round(mse_linear, 2)))
```

```
## [1] "Linear model: 5825.8"
```

```r
print(paste("Log-linear model:", round(mse_log_linear, 2)))
```

```
## [1] "Log-linear model: 5261.73"
```

```r
print(paste("Poisson model:", round(mse_poisson, 2)))
```

```
## [1] "Poisson model: 4764.23"
```

```r
# Create comparison table
comparison_table <- data.frame(
  Model = c("Linear", "Log-Linear", "Poisson"),
  Temperature_Coefficient = c(temp_coeff_linear, temp_coeff_log_linear, temp_coeff_poisson),
  Poisson_LogLikelihood = c(ll_linear, ll_log_linear, ll_poisson),
  Mean_Squared_Error = c(mse_linear, mse_log_linear, mse_poisson),
  R_squared = c(summary(model_linear)$r.squared,
                summary(model_log_linear)$r.squared,
                NA)  # R-squared not directly available for Poisson
)

print("Model Comparison Table:")
```

```
## [1] "Model Comparison Table:"
```

```r
print(comparison_table)
```

```
##        Model Temperature_Coefficient Poisson_LogLikelihood Mean_Squared_Error
## 1     Linear             157.2093663                  -Inf           5825.796
## 2 Log-Linear               1.0415469                  -Inf           5261.726
## 3    Poisson               0.7852919                  -Inf           4764.230
##   R_squared
## 1 0.6745328
## 2 0.8134891
## 3        NA
```

```r
# Assessment of issues
cat("\nAssessment of Evaluation Criteria:\n")
```

```
##
## Assessment of Evaluation Criteria:
```

```r
cat("1. Poisson Log-Likelihood:\n")
```

```
## 1. Poisson Log-Likelihood:
```

```
cat("   - Assumes Poisson distribution, which may not be appropriate for all models\n")
```

```
##    - Assumes Poisson distribution, which may not be appropriate for all models
```

```
cat("   - Linear model may predict negative values, causing issues with log-likelihood\n")
```

```
##    - Linear model may predict negative values, causing issues with log-likelihood
```

```
cat("   - Log-linear model predictions may be negative after transformation\n\n")
```

```
##    - Log-linear model predictions may be negative after transformation
```

```
cat("2. Mean Squared Error:\n")
```

```
## 2. Mean Squared Error:
```

```
cat("   - Sensitive to outliers\n")
```

```
##    - Sensitive to outliers
```

```
cat("   - Doesn't account for the count nature of the data\n")
```

```
##    - Doesn't account for the count nature of the data
```

```
cat("   - All models are compared on the same scale (original bikers count)\n\n")
```

```
##    - All models are compared on the same scale (original bikers count)
```

```
# Check for negative predictions
cat("Negative Predictions Check:\n")
```

```
## Negative Predictions Check:
```

```
cat("Linear model negative predictions:", sum(pred_linear < 0), "\n")
```

```
## Linear model negative predictions: 833
```

```
cat("Log-linear model negative predictions:", sum(pred_log_linear < 0), "\n")
```

```
## Log-linear model negative predictions: 0
```

```
cat("Poisson model negative predictions:", sum(pred_poisson < 0), "\n")
```

```
## Poisson model negative predictions: 0
```

```
# Summary comparison
cat("\nModel Performance Summary:\n")
```

```
##
## Model Performance Summary:
```

```
cat("Best Poisson Log-Likelihood:", c("Linear", "Log-Linear", "Poisson")[which.max(c(ll_linear, ll_log_l
```

```
## Best Poisson Log-Likelihood: Linear
```

```
cat("Best MSE:", c("Linear", "Log-Linear", "Poisson")[which.min(c(mse_linear, mse_log_linear, mse_poisso
```

```
## Best MSE: Poisson
```

```
cat("Best R-squared:", c("Linear", "Log-Linear", "Poisson")[which.max(c(summary(model_linear)$r.squared
```

```
## Best R-squared: Log-Linear
```

```
# Diagnostic information about Poisson regression issues
cat("\n=== DIAGNOSTIC: Why Poisson Regression Shows 0 Values ===\n")
```

```
##
## === DIAGNOSTIC: Why Poisson Regression Shows 0 Values ===
```

```
cat("1. Overdispersion Check:\n")
```

```
## 1. Overdispersion Check:
```

```
dispersion_ratio <- var(bikeshare_subset$bikers)/mean(bikeshare_subset$bikers)
cat("   Variance/Mean ratio:", round(dispersion_ratio, 2), "\n")
```

```
##    Variance/Mean ratio: 124.5
```

```
if(dispersion_ratio > 1.5) {
  cat("   → OVERDISPERSION detected! Poisson assumes variance = mean\n")
  cat("   → Solution: Use Negative Binomial regression\n")
} else {
  cat("   → No overdispersion detected\n")
}
```

```
##    → OVERDISPERSION detected! Poisson assumes variance = mean
##    → Solution: Use Negative Binomial regression
```

```
cat("\n2. Data Range Check:\n")
```

```
##
## 2. Data Range Check:
```

```
cat("   Min bikers:", min(bikeshare_subset$bikers), "\n")
```

```
##    Min bikers: 1
```

```
cat("   Max bikers:", max(bikeshare_subset$bikers), "\n")
```

```
##    Max bikers: 651
```

```
cat("   Range:", max(bikeshare_subset$bikers) - min(bikeshare_subset$bikers), "\n")
```

```
##    Range: 650
```

```
cat("\n3. Convergence Issues:\n")
```

```
##
## 3. Convergence Issues:
```

```
if(exists("model_poisson") && !is.null(model_poisson)) {
  if(model_poisson$converged) {
    cat("   → Poisson model converged successfully\n")
  } else {
    cat("   → Poisson model failed to converge\n")
    cat("   → This causes coefficients to be 0 or NaN\n")
  }
} else {
  cat("   → Poisson model was not fitted due to errors\n")
}
```

```
##    → Poisson model converged successfully
```

```
cat("\n4. Recommendations:\n")
```

```
##
## 4. Recommendations:
```

```
cat("   → For count data with overdispersion: Use Negative Binomial\n")
```

```
##    → For count data with overdispersion: Use Negative Binomial
```

```
cat("   → For count data without overdispersion: Poisson is appropriate\n")
```

```
##    → For count data without overdispersion: Poisson is appropriate
```

```
cat("   → For continuous data: Linear regression is better\n")
```

```
##    → For continuous data: Linear regression is better
```

```
cat("   → For skewed data: Log-linear regression helps\n")
```

```
##    → For skewed data: Log-linear regression helps
```