Assignment 5

Shinjon Ghosh

2025-02-14

```
# 1. Simulate Predictor Variables
set.seed(123)
n <- 100
X <- as.data.frame(matrix(rnorm(n * 10), nrow = n, ncol = 10))
colnames(X) <- paste0("X", 1:10)

# 2. Calculate the Mean Parameter (mu_i)
mu <- 1 + 2 * X$X1 + X$X2 + 0.5 * X$X5 + 1.5 * X$X10

# 3. Generate the Count Response (Y_i)
Y <- rpois(n, lambda = exp(mu))

# 4. Split Data into Training and Testing Sets
set.seed(456)
train_index <- sample(1:n, size = 80, replace = FALSE)
test_index <- setdiff(1:n, train_index)
X_train <- X[train_index, ]
X_test <- X[test_index, ]
Y_train <- Y[train_index]
Y_test <- Y[test_index]

# 5. Fit a Poisson Regression Model
m1 <- glm(Y_train ~ X1 + X2 + X5 + X10, family = poisson(), data = X_train)
summary(m1)
```

```
##
## Call:
## glm(formula = Y_train ~ X1 + X2 + X5 + X10, family = poisson(),
##     data = X_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.02650    0.04937   20.79   <2e-16 ***
## X1           1.98425    0.02337   84.91   <2e-16 ***
## X2           1.08449    0.03863   28.07   <2e-16 ***
## X5           0.45694    0.02115   21.61   <2e-16 ***
## X10          1.47379    0.03034   48.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 17728.02  on 79  degrees of freedom
## Residual deviance:    91.47  on 75  degrees of freedom
```

```
## AIC: 340.29
##
## Number of Fisher Scoring iterations: 4
```

```r
Y_pred <- predict(m1, newdata = X_test, type = "response")
print(Y_pred)
```

```
##            7            9           11           18           21
26
## 8.561482e+00 5.764463e-01 9.457957e+02 2.006728e-03 4.668347e-01 2.750095e
-01
##           29           37           39           42           44
49
## 8.484655e+00 8.650274e-01 3.180309e+00 1.006892e+00 1.050172e+01 3.353672e
+01
##           53           55           57           60           61
72
## 5.644910e+00 4.023417e+00 5.469277e-01 1.019496e+01 8.784807e+01 1.200804e
-03
##           74           81
## 1.215115e+00 2.687758e+01
```

```r
# Compare predicted values with actual test values
comparison <- data.frame(Actual = Y_test, Predicted = Y_pred)
print(comparison)
```

```
##    Actual    Predicted
## 7       7 8.561482e+00
## 9       1 5.764463e-01
## 11   1048 9.457957e+02
## 18      0 2.006728e-03
## 21      0 4.668347e-01
## 26      0 2.750095e-01
## 29      9 8.484655e+00
## 37      0 8.650274e-01
## 39      1 3.180309e+00
## 42      2 1.006892e+00
## 44      7 1.050172e+01
## 49     32 3.353672e+01
## 53      7 5.644910e+00
## 55      8 4.023417e+00
## 57      0 5.469277e-01
## 60      6 1.019496e+01
## 61     81 8.784807e+01
## 72      0 1.200804e-03
## 74      1 1.215115e+00
## 81     25 2.687758e+01
```