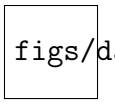


```
> summary(glm(y ~ 0 + X, family = poisson))

Call:
glm(formula = y ~ 0 + X, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0568  -0.8302  -0.3072   0.9279   1.7310

Coefficients:
   Estimate Std. Error z value Pr(>|z|)
X1  0.99600    0.16971   5.869 4.39e-09 ***
X2  1.32661    0.06463  20.525  < 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: 9062.600  on 20  degrees of freedom
Residual deviance:   21.755  on 18  degrees of freedom
AIC: 138.05

Number of Fisher Scoring iterations: 4
```

R Code

```r
dat = read.table("~/files/data/637/aids.txt", header = TRUE)
y = as.numeric(t(matrix(unlist(dat[,-1]), 5, 4)))
X = cbind(1, log(1:length(y)))


### part a
pdf("figs/data_plot.pdf", width = 9, height = 9)
plot(y, pch=20, xaxt = "n", ylab = "Number_of_Cases_of_AIDS",
    xlab = "Time_period", lwd=5, ylim=c(0,max(y)+10))
abline(v=c(4.5, 8.5, 12.5, 16.5), lty=2)
axis(1, at = 1:20, labels = paste0(rep("Q", 20), 1:4))
for (i in 4:8)
    text(4*(i-3)-2+0.5, max(y)+10, paste0("198", i), cex = 2.5)
dev.off()

### part b
pdf("figs/log_data.pdf", width = 9, height = 9)
plot(log(1:20), log(y), pch=20, xaxt = "n", ylab = "log_Number_of_Cases_of_AIDS",
    xlab = "log_Time_period")
abline(v=log(c(4.5, 8.5, 12.5, 16.5)), lty=2)
axis(1, at = log(1:20), labels = paste0(rep("Q", 20), 1:4))
dev.off()

### part c
beta = c(0, 1)  # starting point
eps = 1e-6      # stopping rule
diff = eps + 1
iter = 0

while (diff > eps){
    iter = iter + 1
    oldbeta = beta
    xbeta = X %*% oldbeta
    W = diag(as.numeric(exp(xbeta)), length(y))
    z = xbeta + y * exp(-xbeta) - 1
    beta = solve(t(X) %*% W %*% X) %*% t(X) %*% W %*% z
    diff = sqrt(sum((beta - oldbeta)^2))
    if (TRUE){
        print(iter)
        print(beta)
        cat("\n")
        }
    }

### part d
summary(glm(y ~ 0 + X, family = poisson))

### part e
# covariance matrix (using the latest W matrix)
solve(t(X) %*% W %*% X)

# standard errors
(se = sqrt(diag(solve(t(X) %*% W %*% X))))

# z-statistics for beta[2] (slope parameter, for x_i = log(i))
beta[2] / se[2]

# confidence intervals
cbind(beta - qnorm(0.975) * se, beta + qnorm(0.975) * se)
```