

```

#importing libraries
library(aod)
library(ggplot2)
library(readr)

#importing csv as data frame
binary <- read_csv("~/Downloads/binary.csv")

#summarization
head(binary)
summary(binary)
sapply(binary, sd)

#contingency tables
xtabs(~admit + rank, data = binary)

#running logistic regression or the logic model using gem
binary$rank <- factor(binary$rank)

mylogit <- glm(admit ~ gre + gpa + rank, data = binary, family = "binomial")

summary(mylogit)

#diagnostics
# confidence intervals
# profiled log likelihood
confint(mylogit)

#standard errors
confint.default(mylogit)

#wald test
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)

#testing for difference between ranks
l <- cbind(0, 0, 0, 1, -1, 0)
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), L = l)

#constructing and displaying odds ratios
exp(coef(mylogit))
exp(cbind(OR = coef(mylogit), confint(mylogit)))

#predicted probabilities
# varying rank
newdata1 <- with(binary, data.frame(gre = mean(gre), gpa = mean(gpa), rank =
factor(1:4)))
newdata1

```

```
newdata1$rankP <- predict(mylogit, newdata = newdata1, type = "response")
newdata1
```

```
#varying gre and gpa
newdata2 <- with(binary, data.frame(gre = rep(seq(from = 200, to = 800, length.out =
100),4), gpa = mean(gpa), rank = factor(rep(1:4, each = 100))))
```

```
newdata3 <- cbind(newdata2, predict(mylogit, newdata = newdata2, type = "link", se =
TRUE))
```

```
newdata3 <- within(newdata3, {
+   PredictedProb <- plogis(fit)
+   LL <- plogis(fit - (1.96 * se.fit))
+   UL <- plogis(fit + (1.96 * se.fit))
+ })
```

```
#plotting predicted probabilities
ggplot(newdata3, aes(x = gre, y = PredictedProb)) + geom_ribbon(aes(ymin = LL, ymax
= UL, fill = rank), alpha = 0.2) + geom_line(aes(colour = rank), size = 1)
```

```
#model fit
with(mylogit, null.deviance - deviance)
with(mylogit, df.null - df.residual)
with(mylogit, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
logLik(mylogit)
```