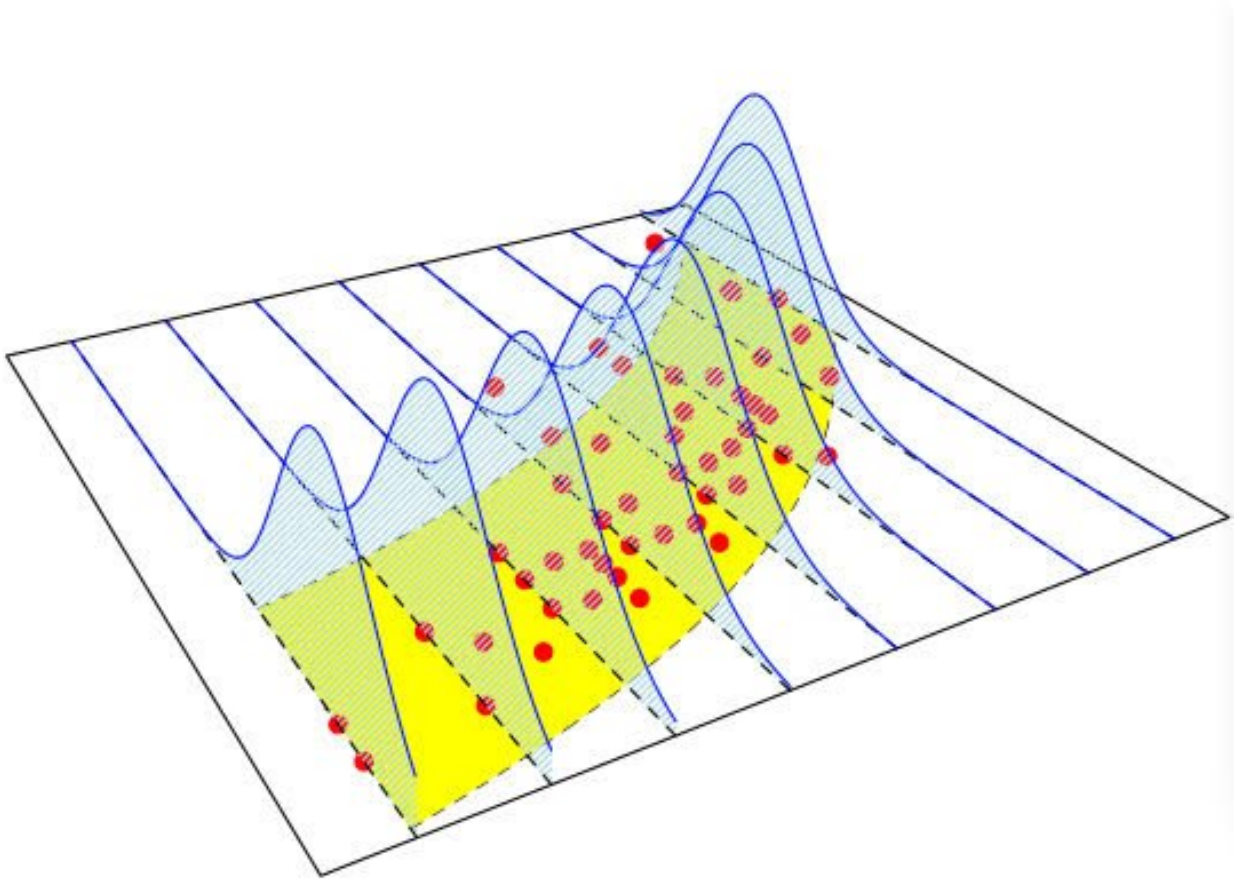
Week 3 - AYU - Pod

Contents



GLM in R

```r
library(tidyverse)
d <- read_csv("data/TermLife.csv")
d <- d[d$FACE>0, ]
modelMLR <- glm(FACE ~ EDUCATION+NUMHH+INCOME, data=d)
summary(modelMLR)
```

```
##
## Call:
## glm(formula = FACE ~ EDUCATION + NUMHH + INCOME, data = d)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2655152   -651472   -339712    -31468  13039540
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.773e+06  6.107e+05  -2.904 0.003987 **
## EDUCATION    1.463e+05  3.849e+04   3.801 0.000178 ***
## NUMHH        1.098e+05  6.552e+04   1.675 0.095001 .
## INCOME       3.392e-01  1.201e-01   2.825 0.005077 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.547961e+12)
##
##     Null deviance: 7.6816e+14  on 274  degrees of freedom
## Residual deviance: 6.9050e+14  on 271  degrees of freedom
## AIC: 8642.1
##
## Number of Fisher Scoring iterations: 2
```

Logistic Regression

We will use the [Wisconsin Hospital Data] again for this example. In the data, our response variable is the total charge, which is a numeric variable, so we cannot use logistic regression for this variable. We will create a binary variable from the total charge. Instead of the exact charge, we are interested in the charge is small (less than the median) or large (more than the median). Create a variable `TOTCHG2` that takes the below value

- `small` if `TOTCHG` is smaller than the average of `TOTCHG`
- `large` otherwise

```r
library(tidyverse)
d <- read_csv('data/frees/HospitalCosts.csv')
d$TOTCHG2 = ifelse(d$TOTCHG > median(d$TOTCHG), 1, 0)
```

Now that `TOTCHG2` is binary, we can regress it using the logistic regression.

```r
model <- glm(TOTCHG2 ~ AGE + factor(GENDER)  + LOS  + factor(RACE) + APRDRG, data=d, family = binomial(
summary(model)
```

```
##
## Call:
## glm(formula = TOTCHG2 ~ AGE + factor(GENDER) + LOS + factor(RACE) +
##     APRDRG, family = binomial(link = "logit"), data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9326  -0.5560   0.0000   0.5374   3.5948
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.067e-01  7.824e-01   0.392  0.69508
## AGE               1.608e-01  2.626e-02   6.123 9.18e-10 ***
## factor(GENDER)1 -8.395e-01  2.822e-01  -2.975  0.00293 **
## LOS               2.636e+00  2.603e-01  10.127  < 2e-16 ***
## factor(RACE)2    -1.253e+00  1.466e+00  -0.855  0.39266
## factor(RACE)3     1.194e+01  1.455e+03   0.008  0.99345
## factor(RACE)4    -4.095e-01  1.718e+00  -0.238  0.81156
## factor(RACE)5    -4.230e+00  1.501e+00  -2.818  0.00484 **
## factor(RACE)6    -1.494e+01  1.020e+03  -0.015  0.98831
## APRDRG           -1.020e-02  1.421e-03  -7.181 6.90e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 691.76  on 498  degrees of freedom
## Residual deviance: 355.38  on 489  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 375.38
##
## Number of Fisher Scoring iterations: 14
```

Prediction

Find the probability that a person get charged a `large` amount.

```
predict(model, list(AGE = 15, GENDER = 1,  LOS = 1, RACE = 1, APRDRG = 600), type = 'response')
```

```
##         1
## 0.1672514
```

```
# Checking the accuracy of the model

predicted_value = ifelse(predict(model, d , type = 'response')>=.5, 1, 0)
true_value = d$TOTCHG2
library(caret)
confusion_matrix = confusionMatrix(data=factor(predicted_value), reference = factor(true_value))
confusion_matrix
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction   0   1
##          0 236  37
##          1  13 213
##
##                Accuracy : 0.8998
##                  95% CI : (0.87, 0.9247)
##     No Information Rate : 0.501
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7996
##
##  Mcnemar's Test P-Value : 0.001143
##
##             Sensitivity : 0.9478
##             Specificity : 0.8520
##          Pos Pred Value : 0.8645
##          Neg Pred Value : 0.9425
##              Prevalence : 0.4990
##          Detection Rate : 0.4729
##    Detection Prevalence : 0.5471
##       Balanced Accuracy : 0.8999
##
##        'Positive' Class : 0
##
```

Poisson Regression

```
p = read_csv('data/poisson_sim.csv')
p$prog <- factor(p$prog, levels=1:3, labels=c("General", "Academic",
                                              "Vocational"))
summary(m1 <- glm(num_awards ~ prog + math, family="poisson", data=p))
```

```
##
## Call:
## glm(formula = num_awards ~ prog + math, family = "poisson", data = p)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -5.24712    0.65845  -7.969 1.60e-15 ***
## progAcademic    1.08386    0.35825   3.025  0.00248 **
## progVocational  0.36981    0.44107   0.838  0.40179
## math            0.07015    0.01060   6.619 3.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```

```
d <- read_csv('data/poisson_sim.csv')

model = glm(num_awards ~ factor(prog) + math, data = d, family = 'poisson')
summary(model)
```

```
##
## Call:
## glm(formula = num_awards ~ factor(prog) + math, family = "poisson",
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.24712    0.65845  -7.969 1.60e-15 ***
## factor(prog)2  1.08386    0.35825   3.025  0.00248 **
## factor(prog)3  0.36981    0.44107   0.838  0.40179
## math           0.07015    0.01060   6.619 3.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```

```
# Coefficients

exp(coef(model))
```

```
##   (Intercept) factor(prog)2 factor(prog)3          math
##    0.00526263    2.95606545    1.44745846    1.07267164
```

```
# Goodness-of-fit test
gof.pvalue = 1 - pchisq(model$deviance, model$df.residual)
gof.pvalue
```

```
## [1] 0.6182274
```

Questions

1. With your group find datasets that suitable for logistic regression to

- Specify the response variable and the input variables to build logistic regression.
- Compute the confusion matrix and report the accuracy of the model

2. With your group find datasets that suitable for poisson regression to

- Specify the response variable and the input variables to build poisson regression.
- Evaluate the quality of the poisson model.