6 Logistic Regression

Immanuel Klein

```r
library("tidyverse")
library("ggplot2")
```

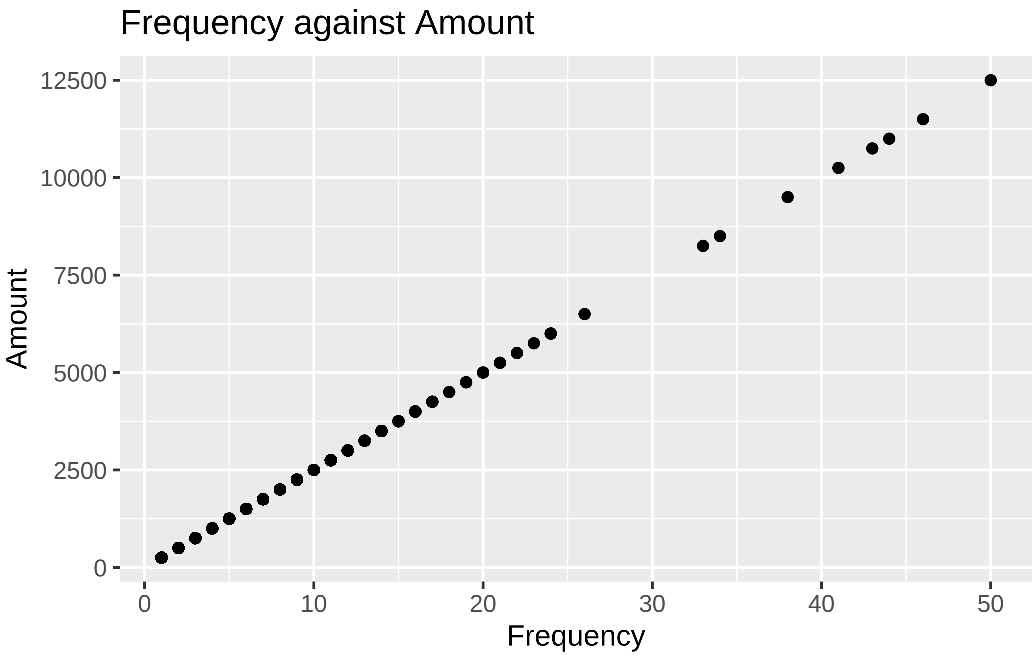
This problem is about predicting whether a donor will donate blood based on features like `recency`, `frequency`, `time`, and `amount` of blood donated. We initially check whether the use of `recency` and `frequency` together is redundant by plotting them against each other and comparing the AIC of GLMs with each of them. We then fit GLMs using various link functions to understand their predictive power. To improve prediction accuracy, we split the dataset into training and test sets and use different covariates to improve model performance. Performance is then evaluated based on the classification error, aiming to achieve a classification error lower than 0.2085561.

Exercise (a)

```r
# Read data, rename columns
blood.data <- read.csv('blood+transfusion+service+center/transfusion.data')
colnames(blood.data) <- c("recency", "frequency", "amount", "time", "donation")

# GLM with donation as response and frequency as covariate
model.frequency <- glm(donation ~ frequency,
                       data = blood.data,
                       family = binomial(link = "logit"))

# GLM with donation as response and amount as covariate
model.amount <- glm(donation ~ amount,
                    data = blood.data,
                    family = binomial(link = "logit"))

# Compare models with AIC
print(AIC(model.frequency, model.amount))
```

```
                df      AIC
model.frequency  2 793.1162
model.amount     2 793.1162
```

```
# Plot frequency against amount
ggplot(blood.data, aes(x = frequency, y = amount)) +
  geom_point() +
  labs(title = "Frequency against Amount",
       x = "Frequency",
       y = "Amount")
```



- Both models have the same AIC value (793.1162), which means that the models are equally good at explaining the variance in the `donation` variable.
- The plot of `frequency` against `amount` shows a linear relationship. This makes sense because the `amount` is directly derived from `frequency`, which means that amount is basically just a scaled version of `frequency`.
- Including both variables in the same model would be redundant as one variable is a direct linear transformation of the other. Including both could also lead to multicollinearity, so we really only should use one of them. From now on, `frequency` will be used.

Exercise (b)

```
# GLM models with different link functions
model.logit <- glm(donation ~ recency,
                   data = blood.data,
                   family = binomial(link = "logit"))
model.probit <- glm(donation ~ recency,
                    data = blood.data,
                    family = binomial(link = "probit"))
model.cloglog <- glm(donation ~ recency,
                     data = blood.data,
                     family = binomial(link = "cloglog"))
model.cauchit <- glm(donation ~ recency,
                     data = blood.data,
                     family = binomial(link = "cauchit"))

# Compare coefficients
data.frame(
  Link = c("Logit", "Probit", "Cloglog", "Cauchit"),
  Intercept = c(coef(model.logit)[1],
                coef(model.probit)[1],
                coef(model.cloglog)[1],
                coef(model.cauchit)[1]),
  Recency = c(coef(model.logit)[2],
              coef(model.probit)[2],
              coef(model.cloglog)[2],
              coef(model.cauchit)[2])
)
```

```
     Link   Intercept     Recency
1   Logit -0.20325062 -0.12497399
2  Probit -0.15962729 -0.06921594
3 Cloglog -0.47660545 -0.11089441
4 Cauchit  0.05060312 -0.17852709
```

```
# Compare AIC values
data.frame(
  Link = c("Logit", "Probit", "Cloglog", "Cauchit"),
  AIC = c(AIC(model.logit),
          AIC(model.probit),
          AIC(model.cloglog),
           AIC(model.cauchit))
)
```

```
     Link      AIC
1   Logit 747.5547
2  Probit 748.3307
3 Cloglog 747.4671
4 Cauchit 749.4959
```

- All models have a negative coefficient for `recency`. Intuitively, this makes sense, because longer gaps between donations might correlate with a lower probability of future donations.
- The cauchit model is the only model with a positive intercept and has the most negative coefficient for `recency`. The cauchit link is more sensitive to outliers, which could explain these differences.
- The cloglog model has the lowest AIC, indicating it fits the data a little better than the other models. The logit model has an AIC close to that of the cloglog, so it also fits the data quite well.
- All in all, the cloglog model slightly outperforms the others because of its AIC value. Given that differences are only small however, the decision between these models is not clear and will depend on context.

Exercise (c)

```
set.seed(1122)

# Randomly sample 374 rows for training set, use rest for test set
training.indices <- sample(1:nrow(blood.data), 374)
training.set <- blood.data[training.indices, ]
test.set <- blood.data[-training.indices, ]

# GLM model on training set with canonical link (logit)
glm.model <- glm(donation ~ recency + frequency,
                 data = training.set,
                 family = binomial(link = "logit"))

# Predict on test set
predictions.prob <- predict(glm.model,
                            newdata = test.set,
                            type = "response")
# Classification
predictions <- ifelse(predictions.prob > 0.5, 1, 0)

# Calculate classification error of first prediction
classification.error <- mean(abs(test.set$donation - predictions))
paste("Error of first prediction: ", classification.error)
```

```
[1] "Error of first prediction:  0.229946524064171"
```

```
# Extend model with time, do same steps as above
glm.model.extended <- glm(donation ~ recency + frequency + time,
                          data = training.set,
                          family = binomial(link = "logit"))

predictions.prob.extended <- predict(glm.model.extended,
                                     newdata = test.set,
                                     type = "response")
predictions.extended <-
  ifelse(predictions.prob.extended > 0.5, 1, 0)

classification.error.extended <-
  mean(abs(test.set$donation - predictions.extended))
paste("Error of extended prediction: ", classification.error.extended)
```

```
[1] "Error of extended prediction:  0.224598930481283"
```

Extending the model with the `time` covariate decreased the classification error, but only marginally. The threshold of 0.2085561 could not be surpassed.