Graphical and Markov Models: Assignment 1

Simon Schmetz

2025-04-23

```
library(readr)
library(dplyr)
library(knitr)
library(graph)
library(gRim)
```

Tasks

1) Write a brief introduction to the data (where it comes from, what are the variables, etc.)
2) Try to fit different types of graphical log linear models to the data and select the most appropriate model. Draw the graphs of the models you fit and comment on whether the independencies in the data make sense.
3) Carry out a goodness of fit test to see whether the selected model fits the data.
4) Write some brief conclusions of your analysis.

1) Introduction

The dataset used in the following work on Graphical loglinear models is the "Student Alcohol Consumption" dataset (https://www.kaggle.com/datasets/uciml/student-alcohol-consumption?select=student-por.csv) with a wide array of variables and the rows corresponding to students behavior and background. From this dataset, the following variables are selected for the analysis from the Portuguese language course data:

sex - student's sex (binary: 'F' - female or 'M' - male) traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour) studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours) failures - number of past class failures (numeric: n if 1<=n<3, else 4) schoolsup - extra educational support (binary: yes or no) health - current health status (numeric: from 1 - very bad to 5 - very good)

These originally categorical variables with values 1:n are transformed in binary categorical with values corresponding to "high" and "low" with values <= 4 being low. A frequency column is then added and the dataset is used to create a contingency table. In these contingency tables we find many entries to be zero, corresponding to the dificulty to find a purely categorical dataset with even distribution across all variables. We however proceed with a lack of alternatives and keep this in mind when interpreting the results of models trained on this data.

```
# Load the data
data <- read_csv("data/student-por.csv")
```

```
## Rows: 649 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr (17): school, sex, address, famsize, Pstatus, Mjob, Fjob, reason, guardi...
## dbl (16): age, Medu, Fedu, traveltime, studytime, failures, famrel, freetime...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
data <- data %>%
  rename_all(~ gsub(" ", "_", tolower(.)))

data = data[c("traveltime", "studytime", "failures", "sex", "schoolsup", "health")]

data <- data %>%
  mutate(
    sex = ifelse(sex == "F", 1, 0),          # 1 for Female, 0 for Male
    schoolsup = ifelse(schoolsup == "yes", 1, 0)
  )


# transform categorical variables
data <- data %>%
  mutate(across(all_of(c("traveltime","studytime","health","failures")), ~ case_when(
    . >= 3 ~ "high",
    . <= 2 ~ "low",
    TRUE ~ as.character(.)
  )))

# Add a frequency column and remove duplicate rows
data <- data %>%
  group_by(across(everything())) %>%
  mutate(Freq = n()) %>%
  ungroup()
data <- data %>%
  distinct(across(everything()), .keep_all = TRUE)

# Generate Contingency table
xtab_result <- xtabs(Freq ~ ., data = data)
print(xtab_result)
```

```
## , , failures = high, sex = 0, schoolsup = 0, health = high
##
##           studytime
## traveltime high low
##       high    0   1
##       low     0   5
##
## , , failures = low, sex = 0, schoolsup = 0, health = high
##
##           studytime
## traveltime high low
##       high    3  21
##       low    23 147
##
## , , failures = high, sex = 1, schoolsup = 0, health = high
##
##           studytime
```

```
## traveltime high low
##       high    0   1
##       low     0   2
##
## , , failures = low, sex = 1, schoolsup = 0, health = high
##
##           studytime
## traveltime high low
##       high    3  16
##       low    64 142
##
## , , failures = high, sex = 0, schoolsup = 1, health = high
##
##           studytime
## traveltime high low
##       high    0   0
##       low     0   1
##
## , , failures = low, sex = 0, schoolsup = 1, health = high
##
##           studytime
## traveltime high low
##       high    0   2
##       low     3   7
##
## , , failures = high, sex = 1, schoolsup = 1, health = high
##
##           studytime
## traveltime high low
##       high    0   0
##       low     0   0
##
## , , failures = low, sex = 1, schoolsup = 1, health = high
##
##           studytime
## traveltime high low
##       high    1   2
##       low    10  27
##
## , , failures = high, sex = 0, schoolsup = 0, health = low
##
##           studytime
## traveltime high low
##       high    0   0
##       low     0   2
##
## , , failures = low, sex = 0, schoolsup = 0, health = low
##
##           studytime
## traveltime high low
##       high    1   5
##       low     6  35
##
## , , failures = high, sex = 1, schoolsup = 0, health = low
```

```
##
##           studytime
## traveltime high low
##       high    0   0
##       low     0   1
##
## , , failures = low, sex = 1, schoolsup = 0, health = low
##
##           studytime
## traveltime high low
##       high    2  11
##       low    14  76
##
## , , failures = high, sex = 0, schoolsup = 1, health = low
##
##           studytime
## traveltime high low
##       high    0   0
##       low     0   0
##
## , , failures = low, sex = 0, schoolsup = 1, health = low
##
##           studytime
## traveltime high low
##       high    0   1
##       low     0   3
##
## , , failures = high, sex = 1, schoolsup = 1, health = low
##
##           studytime
## traveltime high low
##       high    0   0
##       low     0   1
##
## , , failures = low, sex = 1, schoolsup = 1, health = low
##
##           studytime
## traveltime high low
##       high    0   0
##       low     2   8
```

With the dataset set up, we begin with performing a chi square test of independence for each pair of variables. The resulting p-Values show some cases with the p-Values equal to zero or one, indicating there might be problems with performing the test, possibly due to the many zero frequencies as discussed above. We further observe some expected dependences like sex and school support, and some somewhat surprising indipendences like study time and failures for a significance level .05.

```
### Chi square test of indip

# get var pairs
vars <- names(data)[names(data) != "Freq"]
pairs <- combn(vars, 2, simplify = FALSE)

# perform Chi-square test for each pair
results <- lapply(pairs, function(pair) {
```

```
  xtab <- xtabs(Freq ~ ., data = data[, c(pair, "Freq")])
  test <- chisq.test(xtab)
  list(
    variables = pair,
    p_value = test$p.value,
    statistic = test$statistic,
    df = test$parameter,
    expected = test$expected
  )
})
```

```
## Warning in chisq.test(xtab): Chi-squared approximation may be incorrect
## Warning in chisq.test(xtab): Chi-squared approximation may be incorrect
## Warning in chisq.test(xtab): Chi-squared approximation may be incorrect
## Warning in chisq.test(xtab): Chi-squared approximation may be incorrect
```

```
# Print
for (res in results) {
  cat("\n---\n")
  cat("Variables:", paste(res$variables, collapse = " x "), " | p-value =", round(res$p_value, 3))
}
```

```
##
## ---
## Variables: traveltime x studytime  | p-value = 0.24
## ---
## Variables: traveltime x failures  | p-value = 1
## ---
## Variables: traveltime x sex  | p-value = 0.216
## ---
## Variables: traveltime x schoolsup  | p-value = 0.73
## ---
## Variables: traveltime x health  | p-value = 0.69
## ---
## Variables: studytime x failures  | p-value = 0.115
## ---
## Variables: studytime x sex  | p-value = 0
## ---
## Variables: studytime x schoolsup  | p-value = 0.595
## ---
## Variables: studytime x health  | p-value = 0.054
## ---
## Variables: failures x sex  | p-value = 0.129
## ---
## Variables: failures x schoolsup  | p-value = 0.977
## ---
## Variables: failures x health  | p-value = 1
## ---
## Variables: sex x schoolsup  | p-value = 0.007
## ---
## Variables: sex x health  | p-value = 0.005
## ---
## Variables: schoolsup x health  | p-value = 0.538
```

2) Fitting different types of graphical log linear models

With the initial set up of the data and the frequency table complete, we proceed with fitting different types of graphical log linear models to the data. We begin by defining a full model and a graphical model corresponding to the dependences of variables as shown by the chi-squared test with significance level .005.

```
# Define full model
model_formular_full <- ~traveltime*studytime*failures*sex*schoolsup*health

# Define reduced model
model_formular_reduced  <- ~ traveltime*studytime+
  studytime*sex+
  studytime*schoolsup+
  sex*schoolsup+
  sex*health
```

We begin by visualizing the dependence structure in the full model vs the reduced model and checking if they both indeed are graphical models.

```
par(mfrow=c(1,2))

# full model
m_full <- dmod(model_formular_full, data=xtab_result)
isGraphical(model_formular_full)
```

```
## [1] TRUE
```

```
plot(m_full)
summary(m_full)
```

```
##           Length Class  Mode
## modelinfo  4     -none- list
## varNames   6     -none- character
## datainfo   1     -none- list
## fitinfo   10     -none- list
## isFitted   1     -none- logical
## call       3     -none- call
```

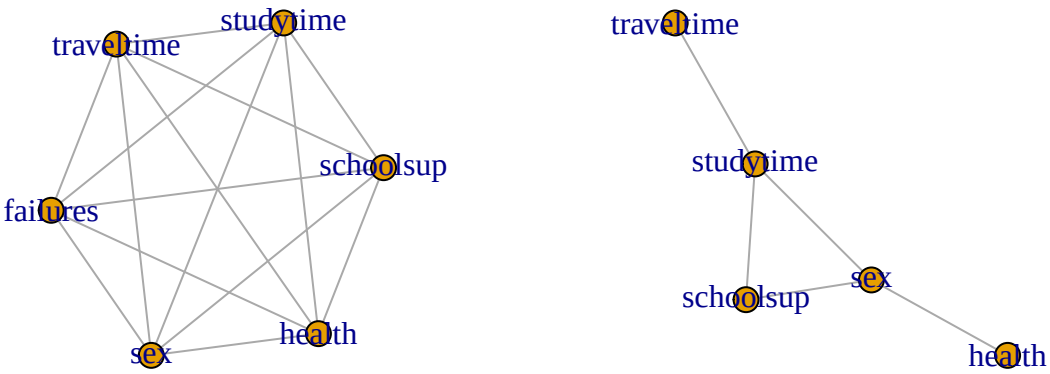
```
ug_full <- ug(model_formular_full)
rip(ug_full)
```

```
## cliques
##   1 : studytime traveltime failures sex schoolsup health
## separators
##   1 :
## parents
##   1 : 0
```

```
# reduced model
m_red <- dmod(model_formular_reduced, data=xtab_result)
isGraphical(model_formular_reduced) #
```

```
## [1] TRUE
```

```
plot(m_red)
```




```
summary(m_red)
```

```
##           Length Class  Mode
## modelinfo  4     -none- list
## varNames   5     -none- character
## datainfo   1     -none- list
## fitinfo   10     -none- list
## isFitted   1     -none- logical
## call       3     -none- call
```

```
ug_red <- ug(model_formular_reduced)
rip(ug_red)
```

```
## cliques
##   1 : traveltime studytime
##   2 : studytime sex schoolsup
##   3 : health sex
## separators
##   1 :
##   2 : studytime
##   3 : sex
## parents
##   1 : 0
##   2 : 1
##   3 : 2
```

We then evaluate AIC and BIC for both models and find the reduced model to shot better (lower) AIC/BIC

```
dmod(model_formular_full, data=xtab_result)
```

```
## Model: A dModel with 6 variables
##  -2logL    :        3222.28 mdim :   63 aic :      3348.28
##  ideviance :          68.53 idf  :   57 bic :      3630.23
##  deviance  :           0.00 df   :    0
## Notice: Table is sparse
##   Asymptotic chi2 distribution may be questionable.
##   Degrees of freedom can not be trusted.
##   Model dimension adjusted for sparsity : 34
```

```
dmod(model_formular_reduced, data=xtab_result)
```

```
## Model: A dModel with 5 variables
##  -2logL    :        3123.31 mdim :   11 aic :      3145.31
```

```
##  ideviance :          32.39 idf  :    5 bic :      3194.54
##  deviance  :          20.37 df   :   20
```

We then perform forward/backwards selection both restricted and unrestricted to find the optimal model. Overall, we find forward/backwards selection to yield the same results except for AIC-unrestricted, but different Results between AIC and BIC based selection. The results optained for restricted/unrestricted appear to be the same with the exception of backwards AIC unrestricted. Overall, travel time is shown to be indipendent in all the resulting models, which is somewhat surprising as the chi-square test showed dependence between travel time and study time. Likewise, some models show the failure variable to be indipendent, also somewhat counter intuitive but in line with the chi-squared test results.

```
# AIC - restricted
par(mfrow=c(1,2))

m_full_selection = dmod(~.^.,data=xtab_result)
mbaic <- backward(m_full_selection)
```

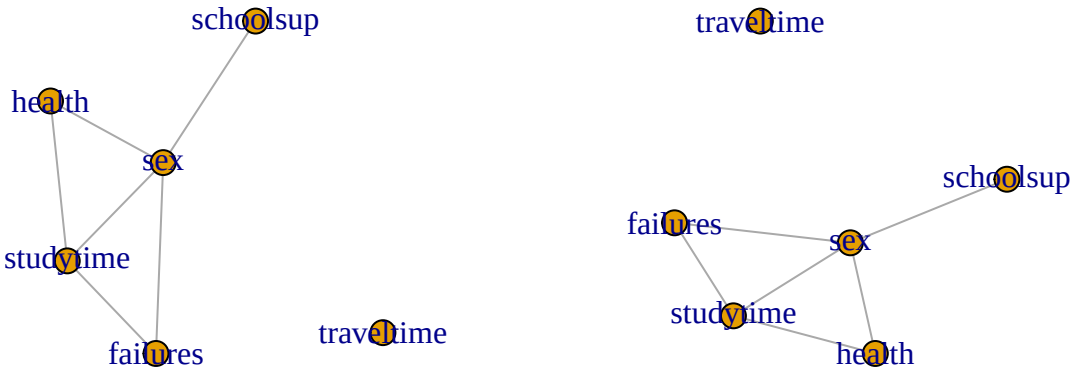
```
##   change.AIC  -13.7085 Edge deleted: traveltime,health
##   change.AIC   -8.1539 Edge deleted: traveltime,schoolsup
##   change.AIC   -5.4792 Edge deleted: studytime,schoolsup
##   change.AIC   -4.0045 Edge deleted: sex,traveltime
##   change.AIC   -3.8780 Edge deleted: failures,schoolsup
##   change.AIC   -3.8866 Edge deleted: failures,health
##   change.AIC   -1.9051 Edge deleted: failures,traveltime
##   change.AIC   -1.7433 Edge deleted: health,schoolsup
##   change.AIC   -0.0872 Edge deleted: studytime,traveltime
```

```
plot(mbaic)

m_zero_selection = dmod(~.^1,data=xtab_result)
mfaic <- forward(m_zero_selection)
```

```
##   change.AIC   11.3883 Edge added: studytime,sex
##   change.AIC    6.5254 Edge added: sex,health
##   change.AIC    6.4800 Edge added: sex,schoolsup
##   change.AIC    4.4453 Edge added: studytime,failures
##   change.AIC    4.2628 Edge added: studytime,health
##   change.AIC    0.2777 Edge added: sex,failures
```

```
plot(mfaic)
```




```
# AIC - unrestricted
par(mfrow=c(1,2))

m_full_selection = dmod(~.^.,data=xtab_result)
```

```
mbaic_unrestr <- backward(m_full_selection,type="unrestricted")
```

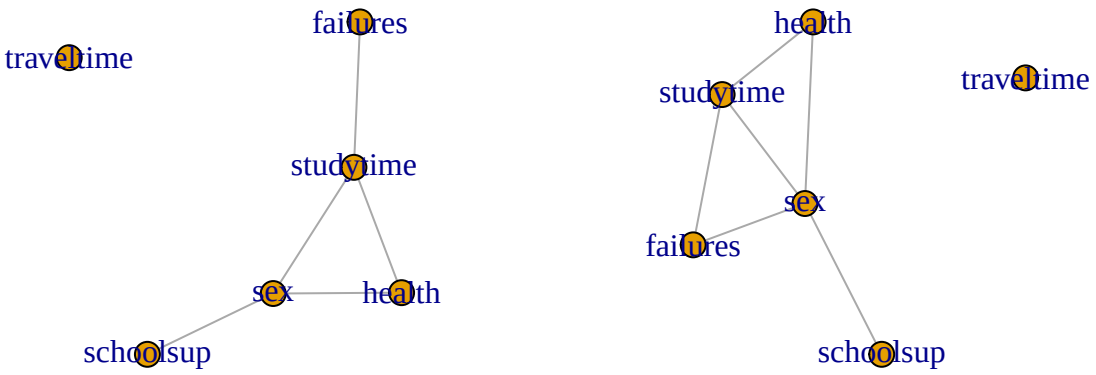
```
##    change.AIC  -13.7085 Edge deleted: traveltime,health
##    change.AIC  -19.6535 Edge deleted: studytime,schoolsup
##    change.AIC  -11.9245 Edge deleted: failures,sex
##    change.AIC   -5.3307 Edge deleted: traveltime,failures
##    change.AIC   -5.3096 Edge deleted: health,failures
##    change.AIC   -3.3460 Edge deleted: traveltime,sex
##    change.AIC   -1.7524 Edge deleted: schoolsup,failures
##    change.AIC   -1.7534 Edge deleted: health,schoolsup
##    change.AIC   -1.7028 Edge deleted: schoolsup,traveltime
##    change.AIC   -0.0872 Edge deleted: studytime,traveltime
```

```
plot(mbaic_unrestr)


m_zero_selection = dmod(~.^1,data=xtab_result)
mfaic_unrestr <- forward(m_zero_selection,type="unrestricted")
```

```
##    change.AIC   11.3883 Edge added: studytime,sex
##    change.AIC    6.5254 Edge added: sex,health
##    change.AIC    6.4800 Edge added: sex,schoolsup
##    change.AIC    4.4453 Edge added: studytime,failures
##    change.AIC    4.2628 Edge added: studytime,health
##    change.AIC    0.2777 Edge added: sex,failures
```

```
plot(mfaic_unrestr)
```




```
# BIC - restricted
par(mfrow=c(1,2))

m_full_selection = dmod(~.^.,data=xtab_result)
mbbic <- backward(m_full_selection,k=log(sum(xtab_result)))
```

```
##    change.AIC  -53.9874 Edge deleted: traveltime,health
##    change.AIC  -35.0065 Edge deleted: traveltime,schoolsup
##    change.AIC  -30.7230 Edge deleted: schoolsup,health
##    change.AIC  -17.4308 Edge deleted: sex,traveltime
##    change.AIC  -12.8374 Edge deleted: failures,health
##    change.AIC  -12.6453 Edge deleted: studytime,schoolsup
##    change.AIC  -12.4866 Edge deleted: failures,schoolsup
##    change.AIC   -6.3805 Edge deleted: failures,traveltime
##    change.AIC   -7.3769 Edge deleted: studytime,failures
```

```
##    change.AIC   -4.6880 Edge deleted: studytime,health
##    change.AIC   -4.5626 Edge deleted: studytime,traveltime
##    change.AIC   -3.3264 Edge deleted: sex,failures
```

```
plot(mbbic)


m_zero_selection = dmod(~.^1,data=xtab_result)
mfbic <- forward(m_zero_selection,k=log(sum(xtab_result)))
```

```
##    change.AIC    6.9128 Edge added: studytime,sex
##    change.AIC    2.0500 Edge added: sex,health
##    change.AIC    2.0046 Edge added: sex,schoolsup
```

```
plot(mfbic)
```


```
# BIC - unrestricted
par(mfrow=c(1,2))

m_full_selection = dmod(~.^.,data=xtab_result)
mbbic_unrestr <- backward(m_full_selection,type="unrestricted",k=log(sum(xtab_result)))
```

```
##    change.AIC  -53.9874 Edge deleted: traveltime,health
##    change.AIC  -73.3587 Edge deleted: studytime,schoolsup
##    change.AIC  -52.2034 Edge deleted: failures,sex
##    change.AIC  -18.7570 Edge deleted: traveltime,failures
##    change.AIC  -18.7359 Edge deleted: health,failures
##    change.AIC  -16.7723 Edge deleted: traveltime,sex
##    change.AIC  -10.6898 Edge deleted: health,schoolsup
##    change.AIC   -6.2422 Edge deleted: schoolsup,failures
##    change.AIC   -6.1782 Edge deleted: schoolsup,traveltime
##    change.AIC   -4.6880 Edge deleted: studytime,health
##    change.AIC   -4.5626 Edge deleted: studytime,traveltime
##    change.AIC   -0.0301 Edge deleted: studytime,failures
```

```
plot(mbbic_unrestr)


m_zero_selection = dmod(~.^1,data=xtab_result)
mfbic_unrestr <- forward(m_zero_selection,type="unrestricted",k=log(sum(xtab_result)))
```

```
##    change.AIC    6.9128 Edge added: studytime,sex
##    change.AIC    2.0500 Edge added: sex,health
##    change.AIC    2.0046 Edge added: sex,schoolsup
```

```
plot(mfbic_unrestr)
```

3) Model Selection and Goodness of fit test

With the Models trained, the resulting models are compared based on AIC/BIC and the chi-squared test. The results show that the unrestricted models have a equal or better fit than the restricted models, while the differences are small. The AIC and BIC values are close to each other, indicating that the models are similar in terms of fit. The chi-squared test shows that all models selected via BIC fit the data well, with p-values greater than 0.05 while AIC selection based models barely not passing the threshhold. Overall, the key for the significant increase in goodness of fit for the BIC based models appears to be identifying the independence of the variable failures.

```
# select bases on AIC/BIC

#
aic <- rep(NA,8)
aic[1] <- mbaic$fitinfo$aic
aic[2] <- mbaic_unrestr$fitinfo$aic
aic[3] <- mfaic$fitinfo$aic
aic[4] <- mfaic_unrestr$fitinfo$aic
aic[5] <- mbbic$fitinfo$aic
aic[6] <- mbbic_unrestr$fitinfo$aic
aic[7] <- mfbic$fitinfo$aic
aic[8] <- mfbic_unrestr$fitinfo$aic


bic <- rep(NA,8)
bic[1] <- mbaic$fitinfo$bic
bic[2] <- mbaic_unrestr$fitinfo$bic
bic[3] <- mfaic$fitinfo$bic
bic[4] <- mfaic_unrestr$fitinfo$bic
bic[5] <- mbbic$fitinfo$bic
bic[6] <- mbbic_unrestr$fitinfo$bic
bic[7] <- mfbic$fitinfo$bic
bic[8] <- mfbic_unrestr$fitinfo$bic


chisq_pval <- rep(NA,8)
chisq_pval[1] <- pchisq(mbaic$fitinfo$dev,mbaic$fitinfo$dimension[4])
chisq_pval[2] <- pchisq(mbaic_unrestr$fitinfo$dev,mbaic$fitinfo$dimension[4])
chisq_pval[3] <- pchisq(mfaic$fitinfo$dev,mfaic$fitinfo$dimension[4])
```

```
chisq_pval[4] <- pchisq(mfaic_unrestr$fitinfo$dev,mfaic$fitinfo$dimension[4])
chisq_pval[5] <- pchisq(mbbic$fitinfo$dev,mbbic$fitinfo$dimension[4])
chisq_pval[6] <- pchisq(mbbic_unrestr$fitinfo$dev,mbbic$fitinfo$dimension[4])
chisq_pval[7] <- pchisq(mfbic$fitinfo$dev,mfbic$fitinfo$dimension[4])
chisq_pval[8] <- pchisq(mfbic_unrestr$fitinfo$dev,mfbic$fitinfo$dimension[4])

selection_type = c("mbaic","mbaic_unrestr","mfaic","mfaic_unrestr","mbbic","mbbic_unrestr","mfbic","mfb

model_seelction_results <- data.frame(
  Model = selection_type,
  AIC = aic,
  BIC = bic,
  Chi_Sq_pval = chisq_pval
)

model_seelction_results
```

```
##           Model      AIC      BIC Chi_Sq_pval
## 1         mbaic 3271.435 3334.091  0.04186113
## 2 mbaic_unrestr 3269.713 3323.418  0.08658915
## 3         mfaic 3271.435 3334.091  0.04186113
## 4 mfaic_unrestr 3271.435 3334.091  0.04186113
## 5         mbbic 3278.421 3318.700  0.71332998
## 6 mbbic_unrestr 3278.421 3318.700  0.71332998
## 7         mfbic 3278.421 3318.700  0.71332998
## 8 mfbic_unrestr 3278.421 3318.700  0.71332998
```

Choosing a model with all BIC selection based models identical and the best AIC best model as mbaic_unrestr, but with the BIC models showing significantly better goodness of fit results while only marginal increase in AIC, the BIC models are heavily favoured in the choice of the model.

```
# Best models
best_aic_index <- which.min(model_seelction_results$AIC)
best_bic_index <- which.min(model_seelction_results$BIC)

best_aic <- model_seelction_results[best_aic_index, ]
best_bic <- model_seelction_results[best_bic_index, ]

# Create a single text block
output <- paste0(
  "====================================\n",
  " Best Model Based on AIC\n",
  "====================================\n",
  sprintf("Model:           %s\n", best_aic$Model),
  sprintf("AIC:             %.2f\n", best_aic$AIC),
  sprintf("BIC:             %.2f\n", best_aic$BIC),
  sprintf("Chi-Sq p-value:  %.4f\n", best_aic$Chi_Sq_pval),

  "\n====================================\n",
  " Best Model Based on BIC\n",
  "====================================\n",
  sprintf("Model:           %s\n", best_bic$Model),
  sprintf("AIC:             %.2f\n", best_bic$AIC),
  sprintf("BIC:             %.2f\n", best_bic$BIC),
  sprintf("Chi-Sq p-value:  %.4f\n", best_bic$Chi_Sq_pval)
```

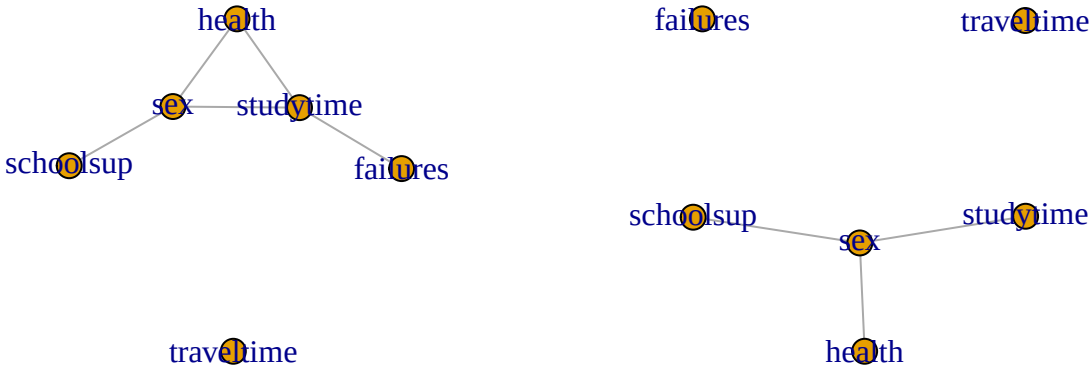
```
)

asis_output(paste0("```\n", output, "\n```"))
```

```
====================================
 Best Model Based on AIC
====================================
Model:            mbaic_unrestr
AIC:              3269.71
BIC:              3323.42
Chi-Sq p-value:   0.0866

====================================
 Best Model Based on BIC
====================================
Model:            mbbic
AIC:              3278.42
BIC:              3318.70
Chi-Sq p-value:   0.7133
```

```
par(mfrow=c(1,2))
plot(mbaic_unrestr)
plot(mbbic)
```



4) Conclusions

For prediction tasks, the BIC based models are preferred as for good AIC, BIC and goodness-of-fit test results. However, when obtaining results from these models, the zero values in the initial frequency tables have to be taken into account. Bearing that in might, the models appear to represent the available data well.