How blood characteristics can influence an athlete

Probabilistic Modeling
Research Project Report

University of Milan
Department of Economics, Management and Quantitative Methods
Master's Degree in Data Science and Economics
Academic year 2021/2022

Abstract

Do blood hemoglobin concentrations of athletes in endurance-related events differ from those in power-related events? This is the question which we are looking answers for. Hemoglobin is an iron-containing protein in the blood of many animals (in the red blood cells, erythrocytes, of vertebrates) that transports oxygen to the tissues such as muscles as in this analysis. The major difference between power and endurance sports is that they use a different energy system as their major source of fuel. Oxygen is used by these systems, and therefore, blood hemoglobin should be present in higher amount for endurance athletes. The dataset used is the Australian Athletes Dataset which comprises 202 observations of 13 different variables. It is used to study the relationships between hemoglobin and other blood such as hematocrit as well as physical features such as body mass. After exploring a variety of models, a Bayesian network constructed with the Hill-Climbing (HC) algorithm minimizing the Bayesian Information Criterion (BIC) score is selected as final model. It is evaluated through k-fold cross-validation and bootstrap. The applications of this research are many. Among them, the athlete could take appropriate action to ensure their hemoglobin concentrations are at optimal levels when performing by modifying different aspects of his/her preparation.

"If you can't outplay them, outwork them" – Ben Hogan

Table of Contents

1. Introduction

This section presents a general and brief introduction of what is hemoglobin and how it relates to different types of sport. The outline of the paper is then laid down.

1.1 Hemoglobin and types of sport

1.1.1 Hemoglobin

Hemoglobin also spelled haemoglobin, iron-containing protein in the blood of many animals (in the red blood cells, erythrocytes, of vertebrates—that transports oxygen to the tissues. Hemoglobin forms an unstable reversible bond with oxygen. In the oxygenated state, it is called oxyhemoglobin and is bright red; in the reduced state, it is purplish blue. Hemoglobin develops in cells in the bone marrow that become red blood cells. When red cells die, hemoglobin is broken up: iron is salvaged, transported to the bone marrow by proteins called transferrins, and used again in the production of new red blood cells; the remainder of the hemoglobin forms the basis of bilirubin, a chemical that is excreted into the bile and gives the feces their characteristic yellow brown color. There are four iron atoms in each molecule of hemoglobin, which accordingly can bind four molecules of oxygen.

1.1.1 Endurance vs Power Disciplines

A study conducted by El-Rayess and his study team analyzed 743 metabolites in the blood serum samples of each of 191 elite athletes (all of which initially tested negative for illegal substances). They found stark differences between the two groups: endurance athletes, for example, had higher levels of endogenous steroids hormones. Power athletes, on the other hand, had higher levels of creatine, which helps muscles recycle the energy molecule ATP (useful for explosive bursts of muscle strength). In fact, endurance athletes rely on the aerobic system, while power athletes primarily use the phosphagen system (the fastest way for the body to resynthesize ATP). Therefore, another major difference between these two groups of athletes is the amount of energy they expend. Apart from the outward, physical differences (it's not hard to tell a powerlifter from a marathoner), endurance and power athletes rely on different types of muscle, which work under different metabolic conditions. Endurance athletes tend to have a higher proportion of slow-twitch muscle fiber, which use oxygen to produce slow and steady energy, while power athletes have a higher proportion of fast-twitch muscles, which generate spikes of energy without oxygen. Muscle fiber type is partly driven by genetics, and the rest of the gap comes from sport-specific training.

1.2 Paper Structure

The project is structured as follows:

II. *Previous Literature*
III. *Dataset and Exploratory Data Analysis (EDA)*
IV. *Model Definition*
V. *Conclusions and limitations*
VI. *References*
VII. *Appendix*

In chapter II, findings from previous studies are briefly explained. In chapter III, the dataset composition is outlined, and its characteristics are summarized. In chapter IV, the probabilistic models created using Bayesian Networks are presented along with the architecture choice and the performances. In chapter V, the conclusions of the study are exposed with the results, the inherent limitations and the recommendations for future work.

2. Previous Literature

This section displays the findings of a previous research on which this study is based.

2.1 Sex, sport, and body-size dependency of hematology in highly trained athletes

This research was conducted by *R. D. Telford* 1 and R. *B. Cunningham*. Blood hemoglobin concentration, hematocrit, red cell count, white cell count (WBC), and plasma ferritin concentration were measured on 1604 occasions from 706 nationally ranked athletes in 12 sports. The blood samples were taken from a forearm vein amidst periods of moderate to intense training but at least 6 h after a training session. A multiple regression model, accounting for correlations between variables and incorporating the categorical variables of sex and sport revealed the following. Each blood variable was found to be dependent on body mass index, (mass/height2, BMI), except for WBC in the males. As BMI increased so did the magnitude of these blood variables (P less than 0.01). Each blood variable was also dependent on the sport (P less than 0.01), significant differences being observed between several sports in each case. Furthermore, as has been previously reported, the magnitude of the blood variables was dependent on the sex of the athlete, each being significantly greater in males (P less than 0.01), except for the WBC, which was greater in females (P less than 0.01). These data indicate that the rationality of interpreting the hematology in highly trained athletes may be increased by taking BMI and sport into account, as well as gender.

As it can be seen, this approach is certainly valid from a scientific point of view. However, the analysis mainly used regression techniques to achieve such results. In this analysis, a different approach is attempted with Bayesian Network.

3. Dataset and Exploratory Data Analysis (EDA)

In this section, the original structure of the dataset *Australian Athletes* is decomposed in detail. In the meanwhile, the findings of the Exploratory Data Analysis (EDA) are shared.

3.1 Dataset Overview

These data were collected in a study of how data on various characteristics of the bloood varied with sport body size and sex of the athlete. The Australian Athletes dataset contains 202 observations of 13 features which are explained below.

3.1.1 Features of the dataset

The dataset contains 13 different features:

1. *rcc:* red blood cell count, in $10^{12}l^{-1}$
2. *wcc:* white blood cell count, in 10^{12} per liter
3. *hc:* hematocrit in percentage
4. *hg:* hemoglobin concentration, in g per decaliter
5. *ferr:* plasma ferritins, ng dl^{-1}
6. *bmi:* body mass index, kg $cm^{-2}10^{2}$
7. *ssf:* sum of skin folds
8. *pcBfat:* percent body fat
9. *lbm:* lean body mass, kg
10. *ht:* height, cm
11. *wt:* weight, kg
12. *sex:* a factor with 2 levels
13. *sport:* a factor with 11 levels

We can divide the feature into two macro groups: the first regarding blood components such as the plasma ferritins and the second regarding body components such as height and body fat. The last two variables, sex and sport, are key components of the analysis and they are expected to play an important part in the Bayesian network.

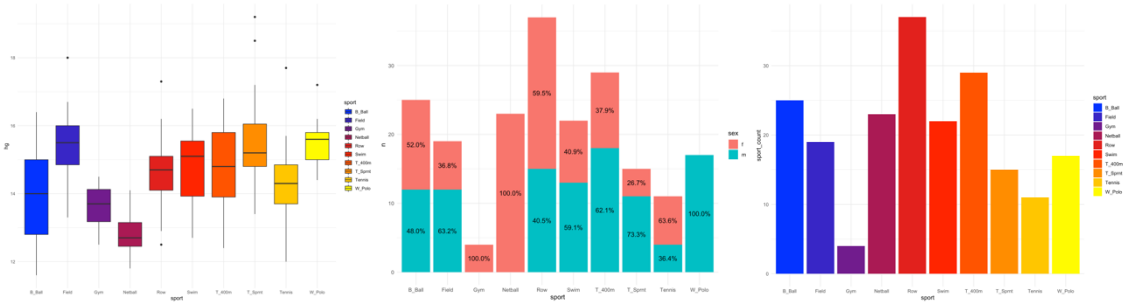
3.2 EDA

3.2.1 Data Pre-Processing

The dataset has both continuous and discrete (sex and sports) variables. As the continuous variables were measured and expressed with different value scales, standardization with the *scale* function is applied in order to correctly learn about the variable interactions. This function centered the dataset with mean approximately 0 and standard deviation 1. The dataset was already prepared to work on it without missing data and outliers.

3.2.2 Analysis over sports

To begin with we'll quickly look at a box plot comparing the distribution of hemoglobin levels for the different sports in order to start learning the features of the data.



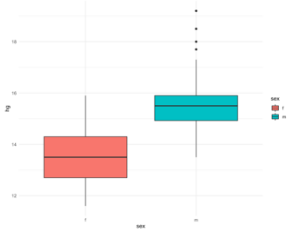
The box plots would suggest there are some differences. For instance, Netball is clearly the sport where the hemoglobin concentration is at minimum in this dataset. This could be due to the fact that Netball was created for girls and women, and it is played indoor. On the other hand, Water Polo and Field sports have the higher concentrations of hemoglobin on average. This could be attributed to the very high consumption of energy these sports require.

It can be observed that this dataset is unbalanced over the sport variable:

1. Some sports such as Gym or Water Polo have only one category of sex, thus either the candidates were only women or men.
2. The observations are not distributed equally among the different sports. For example, Row has more than 35 records while Gym has less than 5.

3.2.3 Analysis over sex

From the graph below, it is confirmed that on average female candidates had less hemoglobin in the blood than male ones. This could be explained also by the sports that the genre plays: in general males practice more energy intense sports. Another hypothesis is that this is a physiological trait.

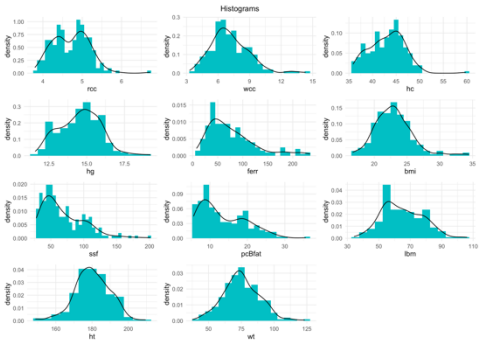


3.2.4 Analysis of variable interactions

From the scatter plots that represent the distribution of the variables against hemoglobin, it can be easily seen that hematocrit (hc) and red blood cell count (rcc) have a linear relationship with hg (hemoglobin). Hg is the protein inside red blood cell as explained above, while hc is simply the percentage by volume of red cells in your blood. This explains the observed behavior. In addition, a distinct line can be drawn between female and male observations.

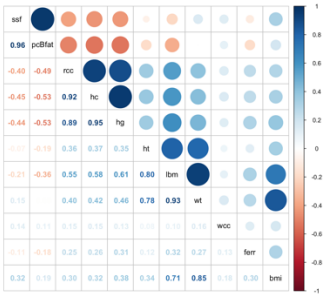
3.2.4 Analysis of variable distributions

From the histogram distribution plotted along the density function, we can see that some variables follow the Gaussian distribution while others do not either partially or entirely.



3.2.4 Analysis of variable distributions

Lastly, the correlation matrix is plotted. It can be observed that the correlation is quite present in several pairs of variables. Nevertheless, as we are dealing with already few variables, we limited the exclusion to the variables that were function of others e.g., BMI = WT/HT^2.



4. Model Definition

This section gives an overview of the models used on the dataset and the logic behind their creation and selection. After experimenting a bit with very simple models which can be found in the appendix, a variety of models was created using different parameters and approaches.

4.1 Discrete Bayesian Networks

A Bayesian Network (hereafter sometimes simply network, net or BN for brevity) is a probabilistic graphical model that encodes the conditional dependency relationships of a set of variables using a Directed Acyclic Graph (DAG). Each node of the graph represents one variable of the dataset; we will therefore interchange the terms node and variable when no confusion arises. The set of directed edges connecting the nodes forms the structure of the network, while the set of conditional probabilities associated with each variable forms the set of parameters of the net. The DAG is represented as an adjacency matrix, a n × n matrix, where n is the number of nodes, whose cells of indices (i, j) take value 1 if there is an edge going from node i to node j, and 0 otherwise. The problems of learning the structure and the parameters of a network from data define the structure learning and parameter learning tasks, respectively. Given a dataset of observations, the structure learning problem is the problem of finding the DAG of a network that may have generated the data. Several algorithms have been proposed fot this problem, but a complete search is doable only for networks with no more than 20-30 nodes. For larger networks, several heuristic strategies exist. 3 The subsequent problem of parameter learning, instead, aims to discover the conditional probabilities that relate the variables, given the dataset of observations and the structure of the network. In addition to structure learning, sometimes it is of interest to estimate a level of the confidence on the presence of an edge in the network. This is what happens when we apply bootstrap to the problem of structure learning. The result is not a DAG, but a different entity that we call weighted partially DAG, which is an adjacency matrix whose cells of indices (i, j) take the number of times that an edge going from node i to node j appear in the network obtained from each bootstrap sample. As the graph obtained when performing structure learning with bootstrap represents a measure of the confidence on the presence of each edge

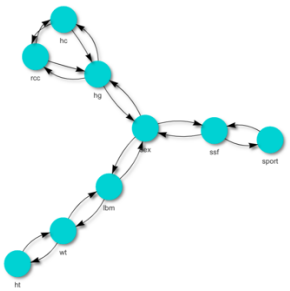
in the original network, and not a binary response on the presence of the edge, the graph is likely to contain undirected edges or cycles. As the structure learnt is not a DAG but a measure of confidence, it cannot be used to learn conditional probabilities. Therefore, parameter learning is not defined in case of network learning with bootstrap. The *bnlearn* package is used from now on.

4.1.1 Structure Learning

When constructing a network starting from a dataset, the first operation we may want to perform is to learn a network that may have generated that dataset, in particular its structure and its parameters. Here, having a hybrid dataset composed by both continuous and discrete variables, we decided to discretize the continuous ones.

4.1.1.1 Constraint-based Algorithm

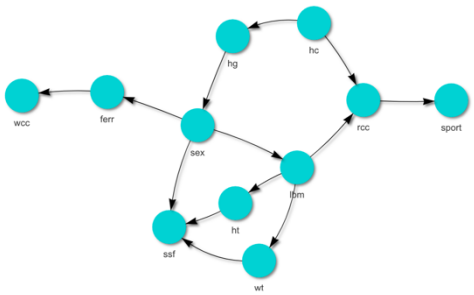
Constraint-based algorithms identify conditional independence constraints with statistical tests, and link nodes that are not found to be independent. The first used algorithm is *Hiton-PC* which is currently believed to be the most scalable choice, it uses a first pass based on marginal tests followed by a backward selection.



Clearly, the algorithm is not producing a DAG. Therefore, it must be discarded.

4.1.1.2 Score-based Algorithm

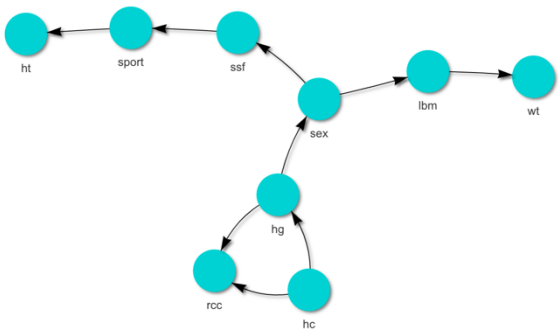
Score-based algorithms are applications of general optimisation techniques; each candidate DAG is assigned a network score maximise as the objective function. Among the different choices, we opted for the *Hill-Climbing* algorithm. In this study, the BIC is believed to be the most appropriate.



However, the quality of DAG crucially depends on whether variables are normally distributed and on whether the relationships that link them are linear; from the exploratory analysis it is not clear that is the case for all of them. We dealt with these problems using bootstrap to construct a consensus network with the arcs that appear more often.

4.1.1.3 Hybrid Algorithm

Hybrid algorithms combine constraint-based and score-based algorithms to complement the respective strengths and weaknesses; they are considered the state of the art in current literature. They work by alternating the following two steps: learn some conditional independence constraints to restrict the number of candidate networks and find the network that maximises some score function and that satisfies those constraints and define a new set of constraints to improve on. The algorithm chosen was the *Max-Min Hill-Climbing (MMHC)*.



4.1.2 Parameters Learning

Parameter learning is the operation that learns the conditional probabilities entailed by a network, given the data and the structure of the network (local distributions). There are several possibilities regarding the estimators, however we selected the *Maximum Likelihood Estimator* (also referred to as either maximum entropy or minimum divergence).

4.1.3 Model Validation

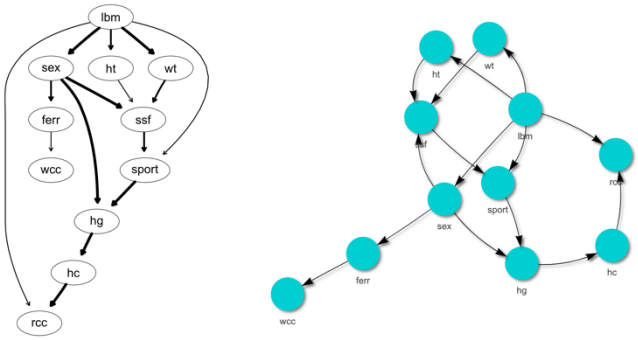
The results of both structure learning and parameter learning should be validated before using a BN for inference. In this paper, the frequentist method is implemented: generate network structures using bootstrap and model averaging (aka bagging). As an additional step, error on predicted values is computed and evaluated.

5. Conclusions

This last section presents the results obtained by using the *Directed Acyclic Graph (DAG)* from the Hill-Climbing Averaged algorithm. Then are outlined the main limitations of this paper as well as the recommendations for future works.

5.1 Results

The final model is the Bayesian Network derived from averaging the network constructed with Hill-Climbing algorithm. Initially, we wanted to learn if there was an intrinsic relationship between the variables based only on raw data. However, from the arch strengths graph, sex is the main variable along with hematocrit. On the other hand, sport is the last determined variable, and it is independent from hemoglobin. This was true for all the models explored. This is clearly not ideal for the sake of our analysis as sport was believed to be relevant in order to learn about different hemoglobin levels in the blood. The possible causes for this conclusion are discussed in the next paragraph. At this point, we whitelisted the relationship between sport, sex and hg that we know it is very likely from previous researches (aka we inserted prior knowledge into the model). The final model is presented below.

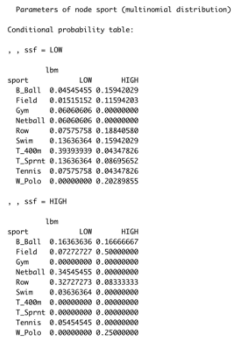


5.1.1 Inference

Probabilistic reasoning on BNs works in the framework of Bayesian statistics and focuses on the computation of posterior probabilities or densities. There are two types of inference:

1. *Exact Inference:* compute the conditional probability distribution over the variables of interest.
2. *Approximate Inference:* use of use approximation techniques based on statistical sampling.

Questions that can be asked are called queries and are typically an event of interest. The two most common queries are: *Conditional Probability (CPQ)* and *Maximum A Posteriori (MAP).*



```
Parameters of node sport (multinomial distribution)

Conditional probability table:

, , ssf = LOW

         lbm
sport           LOW       HIGH
  B_Ball  0.04545455 0.15942029
  Field   0.01515152 0.11594203
  Gym     0.06060606 0.00000000
  Netball 0.06060606 0.00000000
  Row     0.07575758 0.18840580
  Swim    0.13636364 0.15942029
  T_400m  0.39393939 0.04347826
  T_Sprnt 0.13636364 0.08695652
  Tennis  0.07575758 0.04347826
  W_Polo  0.00000000 0.20289855

, , ssf = HIGH

         lbm
sport           LOW       HIGH
  B_Ball  0.16363636 0.16666667
  Field   0.07272727 0.50000000
  Gym     0.00000000 0.00000000
  Netball 0.34545455 0.00000000
  Row     0.32727273 0.08333333
  Swim    0.03636364 0.00000000
  T_400m  0.00000000 0.00000000
  T_Sprnt 0.00000000 0.00000000
  Tennis  0.05454545 0.00000000
  W_Polo  0.00000000 0.25000000
```

```
Parameters of node sex (multinomial distribution)

Conditional probability table:

   lbm
sex         LOW       HIGH
  f 0.80165289 0.03703704
  m 0.19834711 0.96296296
```

In this study, we will make inference using the approximate inference algorithm of *Logic Sampling*.

5.1.1.1 Logic Sampling

Essentially, the algorithm is based on forward (i.e., according to the weak ordering implied by the directed graph) generation of instantiations of nodes guided by their prior probability. If a generated instantiation of an evidence node is different from its observed value, then the entire sample is discarded. This makes the algorithm inefficient if the prior probability of evidence is low. The algorithm is very efficient in cases when no evidence has been observed or the evidence is very likely.

5.1.2 Inference Findings

In accordance with the study conducted by Telford and Cunningham in 1991, also the final Bayesian Network model found that each blood variable is found to be dependent on Body Mass Index (BMI). In addition, the characteristics of the athlete determines which sport he/she practices. It is interesting to see how all the variables are encapsulated at the end in the red blood cell count. This could be why blood analysis is such an important part for the athlete's health monitoring. We also made some queries to investigate more the relationship between the variables and to confirm what was just said above:

1. How probable is it to have high hg given that is a woman?
   $P(hg = HIGH \mid sex = f) = 0.09914669$
2. How probable is that an individual with high hg, wcc, rcc, ht, and ferr, has also high lbm?
   $P(lbm = HIGH \mid hg = HIGH, wcc = HIGH, rcc = HIGH, ht = HIGH, ferr = HIGH) = 0.973913$
3. How probable is it that a woman has low hg given that she swims and is tall?
   $P(hg = LOW \mid sex = f, sport = SWIM) = 1$
4. How probable is it that an individual has high weight and is short given high hg and lbm?
   $P(wt = HIGH \ \& \ ht = LOW \mid hg = HIGH, lbm = HIGH) = 0.1643553$
5. How probable is it that an individual is male given high wcc?
   $P(sex = m \mid wcc = HIGH) = 0.554267$

5.2 Limitations and Recommendations for future work

There are several limitations to our paper. The first and foremost is the dataset itself. The dataset is in reality a small part of a much bigger research sample; therefore, it could be enlarged with more observations. In addition, the dataset we are working on is unbalanced as presented in Chapter III. This is causing the algorithms not learn properly from raw data without injecting prior knowledge. Furthermore, additional relevant variables could be taken in consideration as well as working directly with an expert of the field. Additionally, the dataset is composed by mixed type of data. In case of a deeper discrete analysis, additional levels for continuous variables could be added to get a more precise estimation. On the other hand, mixed interactions models could be involved. This kind of analysis is certainly useful because for example, the athlete could take appropriate action to ensure their hemoglobin concentrations are at optimal levels. Decisions need to be made around

- Diet – What to eat on certain days e.g. training days versus taper days
- Training – When to increase or decrease the intensity and frequency
- Rest – When to take a rest day or recover after a game/race
- Illness – How long do you need to recover?

It is when interventions such as these can be accounted for in the model the user can implement 'what if' scenarios to help make the best decision. Some of these variables can easily be observed but other cannot such as red cell count. This might be a measurement that gets taken once every 2-3 months perhaps, in which case decisions will need to be made without the knowledge of the athletes current red cell count. Fortunately, a Bayesian network can handle this type of uncertainty and missing information.

In conclusion, Bayesian Network are a valid approach to standard methods, in particular for a preliminary analysis in which the underlying patterns of the dataset are not so clear. However, in order to work at the best, prior knowledge is definitely required.

6. References

[1] Autralian Athletes Dataset. *https://vincentarelbundock.github.io/Rdatasets/datasets.html*

[2] R D Telford, R B Cunningham (1991). Sex, sport, and body-size dependency of hematology in highly trained athletes. *https://pubmed.ncbi.nlm.nih.gov/1921671/*

[3] Wikipedia. Hemoglobin. *https://en.wikipedia.org/wiki/Hemoglobin.*

[4] Wikipedia. Blood Oxigen. *https://en.wikipedia.org/wiki/Blood#Oxygen_transport.*

[5] The Difference Between Power and Endurance Athletes Is in Their Blood. *https://gizmodo.com/the-difference-between-power-and-endurance-athletes-is-1823167075.*

[6] Macronutrient needs of endurance and power athletes. *https://training-conditioning.com/article/macronutrient-needs-of-endurance-and-power-athletes/..*

[7] BLearn Package. *https://www.bnlearn.com/.*

[8] Rgraphviz. *https://www.bioconductor.org/packages/release/bioc/html/Rgraphviz.html.*

[9] gRbase. *https://rdrr.io/rforge/gRbase/.*

[10] gRim. *https://cran.rstudio.com/web/packages/gRim/index.html.*

[11] gRain. *https://cran.r-project.org/web/packages/gRain/index.html.*

[12] visNetwork. *https://datastorm-open.github.io/visNetwork/.*

[13] cleandata. *https://cran.r-project.org/web/packages/cleandata/index.html.*

7. Appendix (R Code)

```
'#
Topic: Exploring Levels of Hemoglobin for Different Sports
Name: Filippo Guardassoni
University: Unimi
Course: Probabilistic Modeling
Date: 08/2022
#'

'#
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("RBGL")
BiocManager::install("Rgraphviz")
install.packages("igraph")
install.packages("gRbase")
install.packages("ggm")
install.packages("gRim")
install.packages("mgm")
install.packages("bnlearn")
install.packages("pcalg")
install.packages("deal")
install.packages("gRain")
install.packages("GeneNet")
install.packages("RHugin")
install.packages("BayesNetBP")
install.packages("visNetwork")
install.packages("cleandata")
#'

library(pastecs)
library(ggplot2)
library(dplyr)
library(gridExtra)
library(corrplot)
library(gRim)
library(bnlearn)
library(visNetwork)
library(gRain)
library(cleandata)
```

```
#=============================================================================
# DATASET
#=============================================================================

set.seed(1234)
ais <- read.csv("/Users/amministratore/Documents/Data Science/Second
Year/Second Semester/Probabilistic
Modelling/Project/probabilistic_project/ais.csv")
attach(ais)

#=============================================================================
# EXPLORATORY DATA ANALYSIS
#=============================================================================

# First Look
summary(ais)
str(ais)
stat.desc(ais)

# Analysis over sports
ggplot(ais, aes(x=sport, y=hg, fill=sport)) +
  geom_boxplot() +
  scale_fill_manual(values=colorRampPalette(c('blue', 'red', 'yellow'))(10))
+
  theme_minimal()

sportcount <- ais %>%
  group_by(sport) %>%
  summarize(sport_count = n())

ggplot(sportcount, aes(x=sport, y=sport_count, fill=sport)) +
  geom_bar(stat="identity") + theme_minimal() +
  scale_fill_manual(values=colorRampPalette(c('blue', 'red', 'yellow'))(10))
+
  theme_minimal()

ggplot(ais %>% group_by(sport) %>% count(sport, sex) %>%
         mutate(pct=n/sum(n)),
       aes(sport, n, fill=sex)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=paste0(sprintf("%1.1f", pct*100),"%")),
            position=position_stack(vjust=0.5)) +
  theme_minimal()

# Analysis over sex
ggplot(ais, aes(x=sex, y=hg, fill=sex)) +
  geom_boxplot() + theme_minimal()

# Analysis of variable interactions
s1 <- qplot(data=ais, x=hg, y=wcc, colour=sex) + theme_minimal()
s2 <- qplot(data=ais, x=hg, y=hc, colour=sex) + theme_minimal()
s3 <- qplot(data=ais, x=hg, y=rcc, colour=sex) + theme_minimal()
s4 <- qplot(data=ais, x=hg, y=ferr, colour=sex) + theme_minimal()
s5 <- qplot(data=ais, x=hg, y=bmi, colour=sex) + theme_minimal()
s6 <- qplot(data=ais, x=hg, y=ssf, colour=sex) + theme_minimal()
s7 <- qplot(data=ais, x=hg, y=pcBfat, colour=sex) + theme_minimal()
s8 <- qplot(data=ais, x=hg, y=lbm, colour=sex) + theme_minimal()
s9 <- qplot(data=ais, x=hg, y=ht, colour=sex) + theme_minimal()
s10 <- qplot(data=ais, x=hg, y=wt, colour=sex) + theme_minimal()
```

```
grid.arrange(grobs = list(s1, s2, s3, s4, s5, s6, s7, s8, s9, s10),
             ncol = 3, top = "Scatter Plots")

# Analysis of variable distributions
h1 <- ggplot(data=ais, aes(x=rcc)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 0.1) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h2 <- ggplot(data=ais, aes(x=wcc)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 0.5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h3 <- ggplot(data=ais, aes(x=hc)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 1) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h4 <- ggplot(data=ais, aes(x=hg)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 0.5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h5 <- ggplot(data=ais, aes(x=ferr)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 10) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h6 <- ggplot(data=ais, aes(x=bmi)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 1) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h7 <- ggplot(data=ais, aes(x=ssf)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h8 <- ggplot(data=ais, aes(x=pcBfat)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 1.5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h9 <- ggplot(data=ais, aes(x=lbm)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h10 <- ggplot(data=ais, aes(x=ht)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()
h11 <- ggplot(data=ais, aes(x=wt)) +
  geom_histogram(aes(y=..density..), fill="#00BFC4", binwidth = 5) +
  geom_density(alpha = .2, color="black") +
  theme_minimal()

grid.arrange(grobs = list(h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11),
             ncol = 3, top = "Histograms")


# Correlation Matrix
ais.corr <- select(ais, -c(sex, sport))
ais.corr <- scale(ais.corr, center=TRUE, scale=TRUE)

corr <- cor(ais.corr)
round(corr, 2)
```

```
corrplot.mixed(corr, order = "hclust",
               tl.col = "black", tl.srt = 45)

#=============================================================================
# PROBABILISTIC MODELING - BAYES NETWORK
#=============================================================================

#=============================================================================
## MANUALLY INPUTTED STRUCTURES
#=============================================================================

# SIMPLE DISCRETE CASE
# Using hg, hc and sports
# Set boolean variables
ais.disc <- as.data.frame(ais)
ais.disc$high_hc <- as.factor(ais$hc > median(ais$hc))
ais.disc$high_hg <- as.factor(ais$hg > median(ais$hg))

# Create an empty graph
structure <- empty.graph(c("high_hc", "high_hg", "sport"))

# Set relationships manually
modelstring(structure) <- "[high_hc][sport][high_hg|sport:high_hc]"

# Plot network function
plot.network <- function(structure, ht = "400px"){
  nodes.uniq <- unique(c(structure$arcs[,1], structure$arcs[,2]))
  nodes <- data.frame(id = nodes.uniq,
                      label = nodes.uniq,
                      color = "darkturquoise",
                      shadow = TRUE)

  edges <- data.frame(from = structure$arcs[,1],
                      to = structure$arcs[,2],
                      arrows = "to",
                      smooth = TRUE,
                      shadow = TRUE,
                      color = "black")

  return(visNetwork(nodes, edges, height = ht, width = "100%"))
}


# Observe structure
plot.network(structure)

# Fit model and compute conditional probabilities
ais.sub <- ais.disc[ais.disc$sport %in% c("Netball", "Tennis", "W_Polo"),
               c("high_hc", "high_hg", "sport")]
ais.sub$sport <- factor(ais.sub$sport)
bn.mod <- bn.fit(structure, data = ais.sub)
bn.mod

cat("P(high hemaglobin levels) =",
    cpquery(bn.mod, (high_hg=="TRUE"), TRUE), "\n")

# Trying different queries
# 1) High hg | Play wp, high hc
cat("P(High hg | Play wp, high hc) =",
```

```
    cpquery(bn.mod, (high_hg=="TRUE"), (sport=="W_Polo"&high_hc=="TRUE")),
"\n")
# 2) Play water polo | high hg, high hc
cat("P(Play water polo | high hg, high hc) =",
    cpquery(bn.mod, (sport=="W_Polo"), (high_hg=="TRUE"&high_hc=="TRUE")),
"\n")
# 3) Play water polo | high hg
cat("P(Play water polo | high hg) =",
    cpquery(bn.mod, (high_hg=="TRUE"), (sport=="W_Polo")), "\n")



#=============================================================================
# ALGORITHMICALLY DEFINED STRUCTURES
#=============================================================================
#=============================================================================
# Structure Learning
#=============================================================================

# Remove variables that are functions of others
ais.sub.4 <- ais.fm[, c("hc", "hg", "lbm", "rcc", "wcc", "ferr", "ht", "wt",
                        "ssf", "sex", "sport")]

# Adjust input variables
ais.sub.4 <- ais.sub %>% mutate(
  across(where(is.character), as.factor),
  across(where(is.integer), as.numeric)
)

# Discretize data
ais.disc.2 <- discretize(ais.disc.2, method = "hartemink",
                    breaks = 2, ibreaks = 60, idisc = "quantile")
head(ais.disc.2)

# Compute DAG with a Constraint-based structure learning
dag.hiton = si.hiton.pc(ais.disc.2, undirected = FALSE)
dag.hiton

plot.network(dag.hiton)

# Compute DAG with a Score-based structure learning
dag.hc <- hc(ais.disc.2, score = "bic")
dag.hc

plot.network(dag.hc)

# Compute DAG with a Score-based structure learning with whitelist
dag.hc <- hc(ais.disc.2, whitelist = data.frame(from = c("sport", "sex"), to
= c("hg", "hg")) , score = "bic")
dag.hc

plot.network(dag.hc)



# Compute DAG with a Hybrid structure learning
dag.mmhc <- mmhc(ais.disc.2)
dag.mmhc

plot.network(dag.mmhc)
```

```
#=============================================================================
# Parameter Learning - Maximum Likelihood Estimate
#=============================================================================

# Addressing problems of the dataset with bootstrap
boot <- boot.strength(ais.disc.2, R = 500, algorithm = "hc", algorithm.args
= list(whitelist=data.frame(from = c("sport", "sex"), to = c("hg", "hg"))))
head(boot[(boot$strength > 0.85) & (boot$direction >= 0.5), ], n = 3)

attr(boot, "threshold")

avg.hc = averaged.network(boot)

strength.plot(avg.hc, boot, shape = "ellipse")

par(mfrow = c(1, 2))
graphviz.compare(avg.hc, dag.hc, shape = "ellipse", main = c("averaged DAG",
"single DAG"))
compare(avg.hc, dag.hc)

plot.network(avg.hc)

ais.disc.3 = ais.disc.2[, 1:11]
for (i in names(ais.disc.3[, 1:9]))
  levels(ais.disc.3[, i]) = c("LOW", "HIGH")

# Fit the model and create CPTs
fitted <- bn.fit(avg.hc, ais.disc.3, method = "mle")
fitted

fitted$sex
fitted$sport

#=============================================================================
# Model Validation
#=============================================================================
predcor = structure(numeric(9), names = c("hc", "hg", "lbm", "rcc", "wcc",
                                          "ferr", "ht", "wt", "ssf"))
for (var in names(predcor)) {
  xval = bn.cv(ais.sub.4, bn="hc", fit="mle", loss="cor-lw-cg",
               loss.args=list(target=var, n=200), method="k-fold", runs =
10)
  predcor[var] = mean(sapply(xval, function(x) attr(x, "mean")))
}

round(predcor, digits = 3)

mean(predcor)

encode_ordinal <- function(x, order = unique(x)) {
  x <- as.numeric(factor(x, levels = order, exclude = NULL))
  x
}

ais.encoded <- as.data.frame(ais.sub.4)
ais.encoded[["sport"]] <- encode_ordinal(ais.sub.4[["sport"]])
ais.encoded[["sex"]] <- encode_ordinal(ais.sub.4[["sex"]])
```

```
xval.2 = bn.cv(ais.encoded, bn = "hc", loss = "cor-lw", loss.args =
list(target = "sex", n = 200), runs = 10)

err = numeric(10)

for (i in 1:10) {
  tt = table(unlist(sapply(xval.2[[i]], '[[', "observed")),
             unlist(sapply(xval.2[[i]], '[[', "predicted")) > 0.50)
  err[i] = (sum(tt) - sum(diag(tt))) / sum(tt)
}

summary(err)

xval.3 = bn.cv(ais.encoded, bn = "hc", loss = "cor-lw", loss.args =
list(target = "sport", n = 200), runs = 10)

err.2 = numeric(10)

for (i in 1:10) {
  tt.2 = table(unlist(sapply(xval.3[[i]], '[[', "observed")),
             unlist(sapply(xval.3[[i]], '[[', "predicted")) > 0.50)
  err.2[i] = (sum(tt.2) - sum(diag(tt.2))) / sum(tt.2)
}

summary(err.2)

# Model Averaging from multiple searches
nodes <- names(ais.disc.2)
start <- random.graph(nodes = nodes, method = "ic-dag",
                      num = 500, every = 50)
netlist <- lapply(start,
                  function(net) {
                    hc(ais.disc.2, score = "bic", iss = 10, start = net)
                  }
)

start <- custom.strength(netlist, nodes = nodes)

avg.start <- averaged.network(start)
plot.network(avg.start)

avg.boot <- averaged.network(boot)
plot.network(avg.boot)

all.equal(cpdag(avg.boot), cpdag(avg.start))

#============================================================================
# Inference
#============================================================================

# P(hg = high | sex = f)
cat("P(hg = high | sex = f) =",
    cpquery(fitted, (hg == "HIGH"), (sex == "f")), "\n")

# P(sex = m | wcc = high)
cat("P(sex = m | wcc = high) =",
    cpquery(fitted, (sex == "m"), (wcc == "HIGH")), "\n")

# P(BMI = HIGH | hg = HIGH, wcc = HIGH, rcc = HIGH, ht = HIGH, ferr = HIGH)
```

```
cat("P(lbm = HIGH  | hg = HIGH, wcc = HIGH, rcc = HIGH, ht = HIGH, ferr =
HIGH) =",
    cpquery(fitted, (lbm == "HIGH"), (hg == "HIGH" & wcc == "HIGH"& rcc ==
"HIGH" & ht == "HIGH" & ferr == "HIGH")), "\n")

# P(hg = low | sex = f & sport = swim)
cat("P(hg = low | sex = f & sport = swim) =",
    cpquery(fitted, (hg == "LOW"), (sex == "f" & sport == "Swim")), "\n")

# P(wt = HIGH & ht = LOW | hg = HIGH, lbm = "HIGH")
cat("P(wt = HIGH & ht = LOW | hg = HIGH, lbm = HIGH) =",
    cpquery(fitted, (wt == "HIGH"  & ht == "LOW"), (hg == "HIGH" & lbm ==
"HIGH")), "\n")
```