

Evidencia 2. Redes bayesianas caso continuo. Código

AUTHORS

Mayra Sarahí de Luna Castillo

A01635774

Juan Manuel Hernández Solano

A00572208

Alejandra Velasco Zárate A01635453

José Antonio Juárez Pacheco A0057218

José Carlos Yamuni Contreras

A01740285

PUBLISHED

August 24, 2023

Instalamos librerías

```
# Instalar Bioconductor si aún no está instalado
#if (!requireNamespace("BiocManager", quietly = TRUE))
#   install.packages("BiocManager")

# Cargar la biblioteca BiocManager
#library(BiocManager)

# Instalar paquetes desde Bioconductor
#BiocManager::install(c("graph", "Rgraphviz"))

#install.packages("mgcv")
#install.packages("gratia")
library(mgcv)
```

Loading required package: nlme

This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.

```
library(gratia)

library(bnlearn)
library(nlme)
library(ggplot2)
```

Leemos base de datos y cambiamos el nombre a 3 columnas

```
data=read.csv("/Users/tonitojuarez/Documents/RStudio/clean_data")
colnames(data)[colnames(data) == "Packed.cell.volume"] ="Hematocrit"
```

```
colnames(data)[colnames(data) == "White.blood.cell.count"] ="Cel
colnames(data)[colnames(data) == "Red.blood.cell.count"] ="Celu
head(data)
```

	Edad	Presion	Gravedad	Glucosa	Urea	Creatinina	Sodio
	Potasio	Hemoglobina					
1	48	80	1.020	121.0000	36	1.2	137.5919
4.629126			15.4				
2	7	50	1.020	148.0113	18	0.8	137.5919
4.629126			11.3				
3	62	80	1.010	423.0000	53	1.8	137.5919
4.629126			9.6				
4	48	70	1.005	117.0000	56	3.8	111.0000
2.500000			11.2				
5	51	80	1.010	106.0000	26	1.4	137.5919
4.629126			11.6				
6	60	90	1.015	74.0000	25	1.1	142.0000
3.200000			12.2				
	Hematocrito	CelulasBlancas	CelulasRojas				
1	44	7800	5.200000				
2	38	6000	4.707435				
3	31	7500	4.707435				
4	32	6700	3.900000				
5	35	7300	4.600000				
6	39	7800	4.400000				

Verificar la dimensión y si hay variables faltantes

```
sum(is.na(data))
```

```
[1] 0
```

```
dim(data)
```

```
[1] 397 12
```

Creamos una nueva variable de Gravedad con un poco de ruido. Se le agrega ruido a la variable de gravedad para que así a la hora de hacer los modelos, tengamos suficientes combinaciones con nuestras variables. Esta nueva variable se agrega a una nueva base de dato.

```
gravedad_2=jitter(data$Gravedad)
data_ruido <- cbind(data, gravedad_2)
data_ruido<-data_ruido[, -3]
head(data_ruido)
```

	Edad	Presion	Glucosa	Urea	Creatinina	Sodio	Potasio
							Hemoglobina
1	48	80	121.0000	36	1.2	137.5919	4.629126
							15.4
2	7	50	148.0113	18	0.8	137.5919	4.629126
							11.3
3	62	80	423.0000	53	1.8	137.5919	4.629126
							9.6
4	48	70	117.0000	56	3.8	111.0000	2.500000
							11.2
5	51	80	106.0000	26	1.4	137.5919	4.629126
							11.6
6	60	90	74.0000	25	1.1	142.0000	3.200000
							12.2

	Hematocrito	CelulasBlancas	CelulasRojas	gravedad_2
1	44	7800	5.200000	1.019615
2	38	6000	4.707435	1.019602
3	31	7500	4.707435	1.009904
4	32	6700	3.900000	1.004632
5	35	7300	4.600000	1.009769
6	39	7800	4.400000	1.014722

Implementamos primer DAG

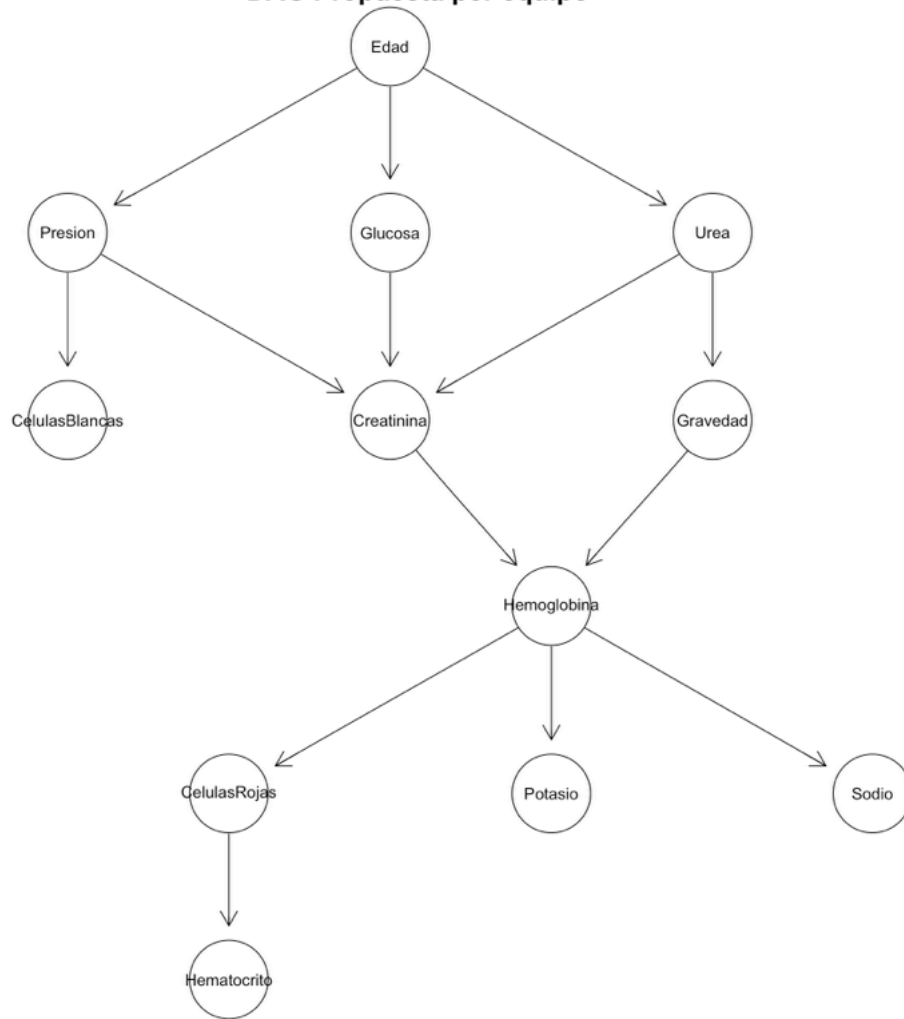
```
dag=model2network("[Edad] [Urea|Edad] [Presion|Edad] [Glucosa|Edad]
```

Representación gráfica

```
g <- Rgraphviz::layoutGraph(bnlearn::as.graphNEL(dag))
graph::nodeRenderInfo(g) <- list(fontsize=22)
Rgraphviz::renderGraph(g)
# Create the title text
title_text <- "DAG Propuesta por equipo"

# Add title using grid graphics
library(grid)
grid.text(label = title_text, x = 0.45, y = 0.99, gp = gpar(for
```

DAG Propuesta por equipo



Ajustamos una GBN (Gaussian Bayesian Network)

```
data_fit=bn.fit(dag, data = data)
```

Revisamos los parámetros ajustados, e.g., de la distribución local

```
data_fit$Creatinina
```

Parameters of node Creatinina (Gaussian distribution)

Conditional density: Creatinina | Glucosa + Presion + Urea

Coefficients:

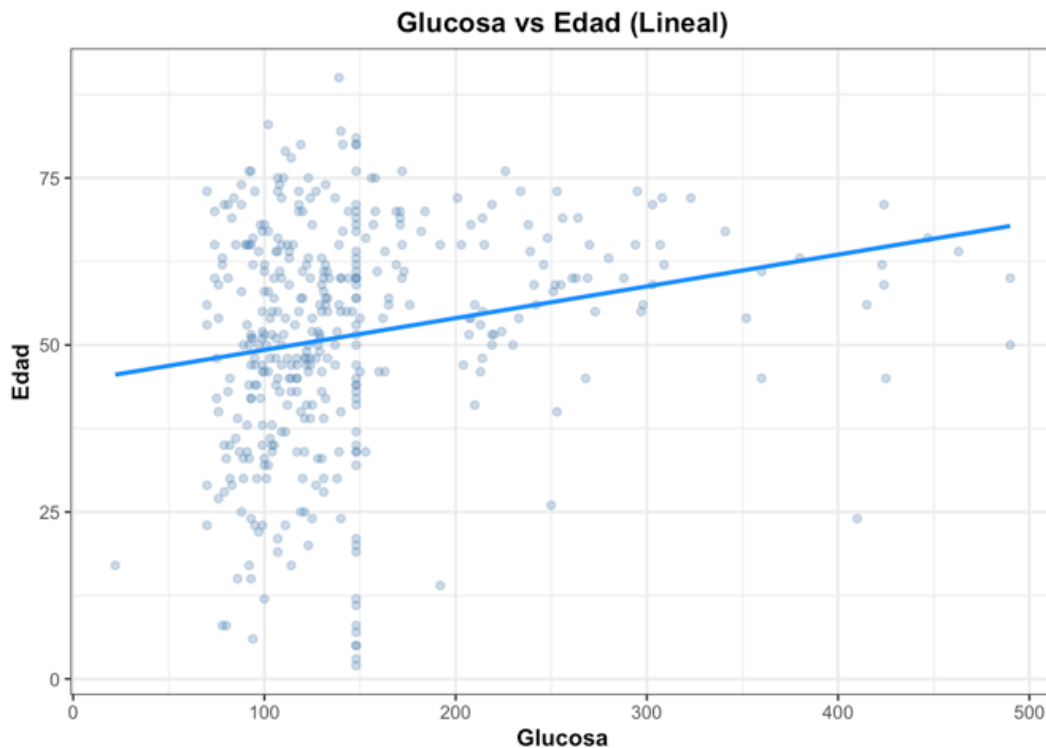
(Intercept)	Glucosa	Presion	Urea
-1.9294610394	0.0002521164	0.0158626937	0.0653914973

Standard deviation of the residuals: 4.600415

Checar si los datos del data frame se ajustan a un modelo lineal

```
ggplot(data, aes(x = Glucosa, y = Edad)) + geom_point(color = 'dodgerblue') +
  geom_smooth(method = lm, se = FALSE, lwd = 1, color = "dodgerblue") +
  theme_bw() + labs(x = "Glucosa", y = "Edad", title = "Glucosa vs Edad") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title.x = element_text(face = "bold", size = 12),
        axis.title.y = element_text(face = "bold", angle = 90, size = 12),
        legend.title = element_text(hjust = 0.5, face = "bold", size = 12),
        legend.text = element_text(hjust = 0.5, size = 10),
        strip.text = element_text(hjust = 0.5, face = "bold", size = 12),
        plot.subtitle = element_text(hjust = 0.5, face = "bold", size = 12))
```

`geom_smooth()` using formula = 'y ~ x'



BIC y AIC Scores del Modelo Lineal

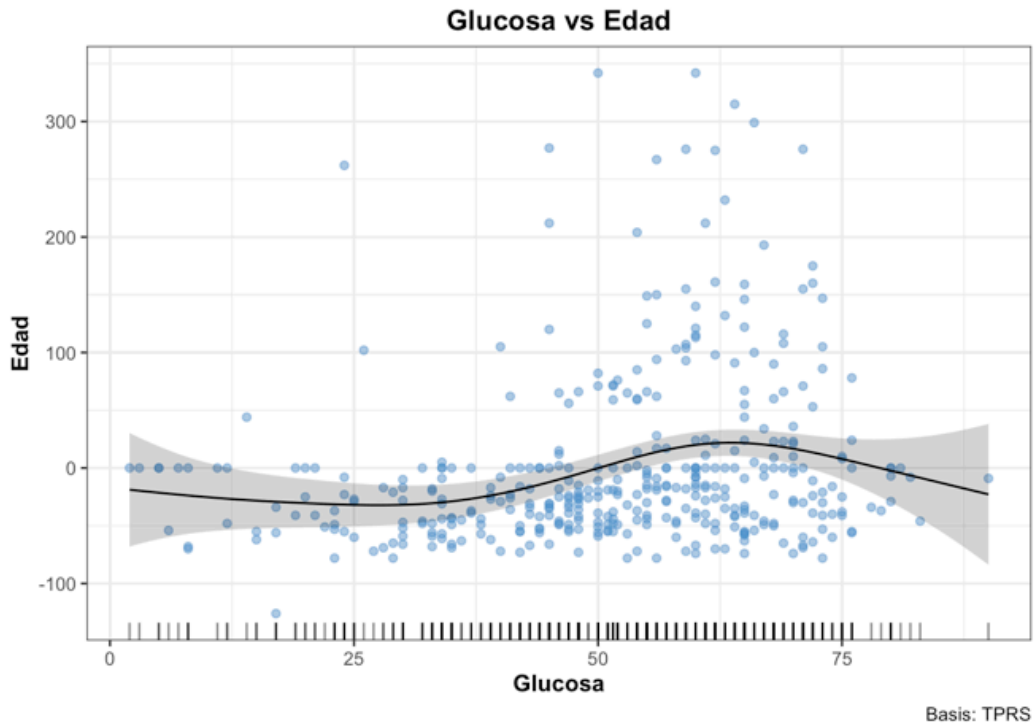
```
#BIC Y AIC DEL MODELO LINEAL
BIC1=score(dag, data = data, type = "bic-g")
#cat("BIC del modelo lineal DAG equipo:",BIC1, "\n")
AIC1=score(dag, data = data, type = "aic-g")
#cat("BIC del modelo lineal DAG equipo:",AIC1, "\n")
```

Vemos que los puntos no están bien ajustados a la gráfica lo que quiere decir que un modelo lineal no es el mejor, por lo tanto, tratamos con un modelo no paramétrico. Entonces probamos con un modelo no paramétrico

```

mod_gam<-gam(Glucosa ~ s(Edad), data = data)
draw(mod_gam, residuals = TRUE) +
  theme_bw() + labs(x = "Glucosa", y = "Edad", title = "Glucosa vs Edad") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title.x = element_text(face = "bold", size = 12),
        axis.title.y = element_text(face = "bold", angle = 90, size = 12),
        legend.title = element_text(hjust = 0.5, face = "bold", size = 12),
        legend.text = element_text(hjust = 0.5, size = 10),
        strip.text = element_text(hjust = 0.5, face = "bold", size = 10),
        plot.subtitle = element_text(hjust = 0.5, face = "bold", size = 10))

```



Ajustamos cada una de las variables a nuestro modelo no paramétrico

```

#Tuvimos que usar la gravedad_2 para que nuestras variables tuvieran el mismo nivel de gravedad
Glucosa_Edad=gam(Glucosa~s(Edad),data=data)
Presion_Edad=gam(Presion~s(Edad),data=data)
white_presion = gam(CelulasBlancas~s(Presion),data=data)
Urea_Edad=gam(Urea~s(Edad),data=data)
Creatinina_Glucosa_Presion_Urea=gam(Creatinina ~ s(Glucosa) + s(Presion) + s(Urea), data = data)
Gravedad_Urea=gam(gravedad_2 ~ s(Urea), data = data_ruido)
Hemoglobina_Creatina_Gravedad=gam(Hemoglobina ~ s(Creatinina) + s(Gravedad_Urea), data = data)
Potasio_Hemoglobina=gam(Potasio ~ s(Hemoglobina), data = data)
Sodio_Hemoglobina=gam(Sodio ~ s(Hemoglobina),data = data)
Red_Hemoglobina=gam(CelulasRojas ~ s(Hemoglobina),data = data)
Volume_Red = gam(Hematocrito ~ s(CelulasRojas),data = data)
Edad<-lm(Edad ~ 1, data = data)

```

Sacamos el BIC y el AIC del modelo no paramétrico de la DAG 1

```
BIC2=-(BIC(Edad_) + BIC(Volume_Red) + BIC(Red_Hemoglobina) + BIC(Red_Hemoglobina))
#cat("BIC del modelo no paramétrico DAG equipo:",BIC2, "\n")

AIC2=-(AIC(Edad_) + AIC(Volume_Red) + AIC(Red_Hemoglobina) + AIC(Red_Hemoglobina))
#cat("AIC del modelo no paramétrico DAG equipo:",AIC2, "\n")
```

Se crea una segunda DAG

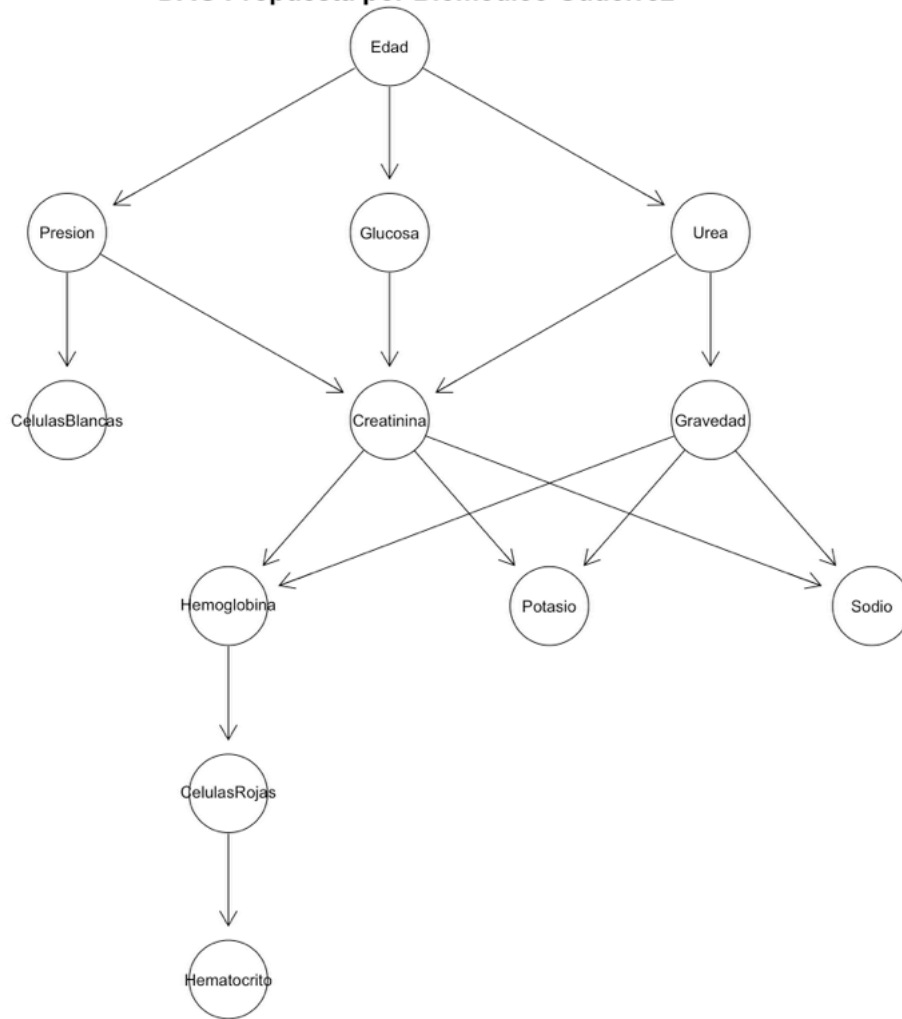
```
dag1=model2network("[Edad] [Urea|Edad] [Presion|Edad] [Glucosa|Edad]
```

Representación gráfica

```
g <- Rgraphviz::layoutGraph(bnlearn::as.graphNEL(dag1))
graph::nodeRenderInfo(g) <- list(fontsize=22)
Rgraphviz::renderGraph(g)
# Create the title text
title_text <- "DAG Propuesta por Biomédico Gutiérrez"

# Add title using grid graphics
library(grid)
grid.text(label = title_text, x = 0.45, y = 0.99, gp = gpar(for
```

DAG Propuesta por Biomédico Gutiérrez



BIC y AIC Scores del Modelo Lineal de la DAG 2

```

#BIC Y AIC DEL MODELO LINEAL
BIC3=score(dag1, data = data, type = "bic-g")
#cat("BIC del modelo lineal de DAG Biomédico Gutiérrez:",BIC3,
AIC3=score(dag1, data = data, type = "aic-g")
#cat("AIC del modelo lineal de DAG Biomédico Gutiérrez:",AIC3,

```

Ajustamos cada una de las variables a nuestro modelo no paramétrico de la DAG 2

```

#Tuvimos que usar la gravedad_2 para que nuestras variables tuv
Glucosa_Edad=gam(Glucosa~s(Edad),data=data)
Presion_Edad=gam(Presion~s(Edad),data=data)
white_presion = gam(CelulasBlancas~s(Presion),data=data)
Urea_Edad=gam(Urea~s(Edad),data=data)
Creatinina_Glucosa_Presion_Urea=gam(Creatinina ~ s(Glucosa) + s
Gravedad_Urea=gam(gravedad_2 ~ s(Urea), data = data_ruido)
Hemoglobina_Creatina_Gravedad=gam(Hemoglobina ~ s(Creatinina) +
Potasio_Creatina_Gravedad=gam(Potasio ~ s(Creatinina) + s(grave

```



```
Sodio_Creatina_Gravedad=gam(Sodio ~ s(Creatinina) + s(gravedad_
Red_Hemoglobina=gam(CelulasRojas ~ s(Hemoglobina),data = data)
Volume_Red = gam(Hematocrito ~ s(CelulasRojas),data = data)
Edad_<-lm(Edad ~ 1, data = data)
```

Sacamos el BIC y el AIC del modelo no paramétrico de la DAG 2

```
BIC4=-(BIC(Edad_) + BIC(Volume_Red) + BIC(Red_Hemoglobina) + BIC(Sodio_Creatina_Gravedad))
#cat("BIC del modelo no paramétrico de DAG Biomédico Gutiérrez: ", BIC4, "\n")

AIC4=-(AIC(Edad_) + AIC(Volume_Red) + AIC(Red_Hemoglobina) + AIC(Sodio_Creatina_Gravedad))
#cat("AIC del modelo no paramétrico de DAG Biomédico Gutiérrez: ", AIC4, "\n")
```

Quitaremos la variable de edad para las siguientes DAGs

```
df <- data[, -1]
head(df)
```

	Presion	Gravedad	Glucosa	Urea	Creatinina	Sodio	Potasio
Hemoglobina							
1	80	1.020	121.0000	36	1.2	137.5919	4.629126
15.4							
2	50	1.020	148.0113	18	0.8	137.5919	4.629126
11.3							
3	80	1.010	423.0000	53	1.8	137.5919	4.629126
9.6							
4	70	1.005	117.0000	56	3.8	111.0000	2.500000
11.2							
5	80	1.010	106.0000	26	1.4	137.5919	4.629126
11.6							
6	90	1.015	74.0000	25	1.1	142.0000	3.200000
12.2							
Hematocrito							
1	44		7800	5.200000			
2	38		6000	4.707435			
3	31		7500	4.707435			
4	32		6700	3.900000			
5	35		7300	4.600000			
6	39		7800	4.400000			

Se crea una tercera DAG

```
dag2=model2network("[Urea][Presion][Glucosa][Creatinina|Glucosa]
```

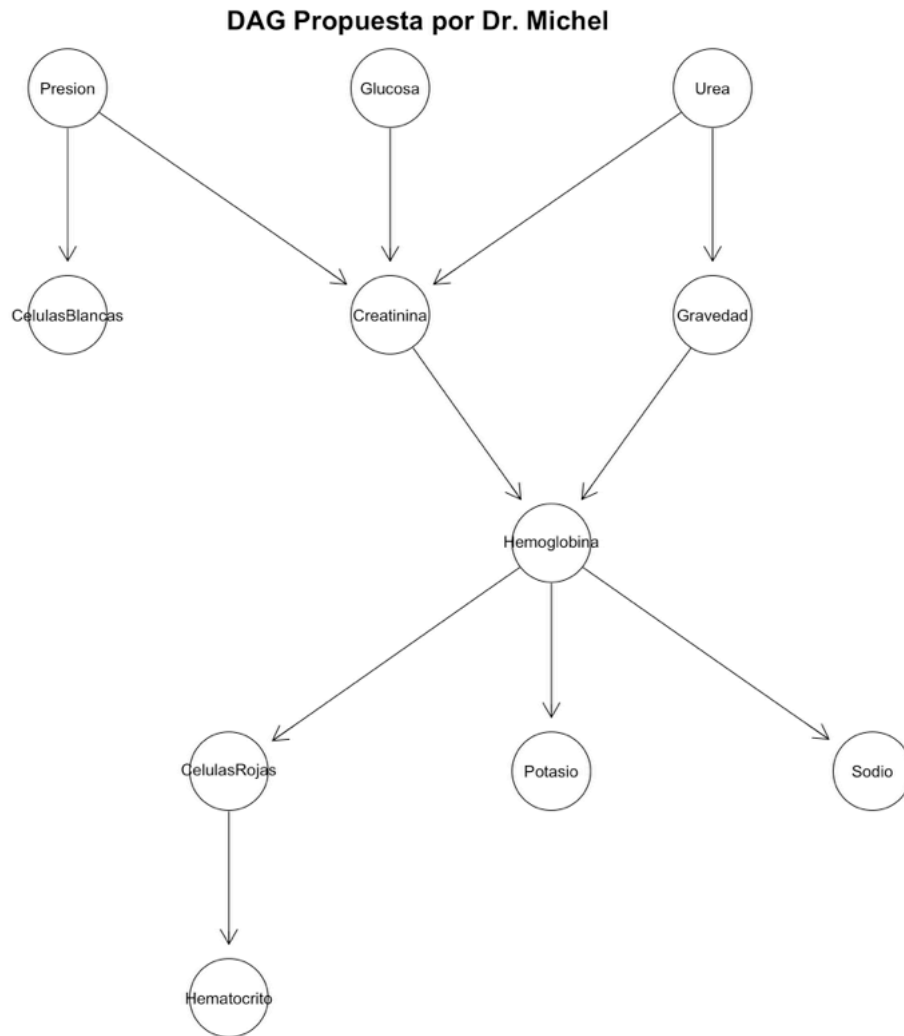
Representación gráfica

```

g <- Rgraphviz::layoutGraph(bnlearn::as.graphNEL(dag2))
graph::nodeRenderInfo(g) <- list(fontsize=22)
Rgraphviz::renderGraph(g)
# Create the title text
title_text <- "DAG Propuesta por Dr. Michel"

# Add title using grid graphics
library(grid)
grid.text(label = title_text, x = 0.45, y = 0.99, gp = gpar(for

```



BIC y AIC Scores del Modelo Líneal de la DAG 3

```

#BIC Y AIC DEL MODELO LINEAL
BIC5=score(dag2, data = df, type = "bic-g")
#cat("BIC del modelo lineal de DAG Biomédico Gutiérrez:",BIC5,
AIC5=score(dag2, data = df, type = "aic-g")
#cat("AIC del modelo lineal de DAG Biomédico Gutiérrez:",AIC5,

```

Ajustamos cada una de las variables a nuestro modelo no paramétrico de

la DAG 3

```
#Tuvimos que usar la gravedad_2 para que nuestras variables tuvieran un efecto
white_presion = gam(CelulasBlancas~s(Presion),data=df)
Creatinina_Glucosa_Presion_Urea=gam(Creatinina ~ s(Glucosa) + s(Presion) + s(Urea), data = df)
Gravedad_Urea=gam(gravedad_2 ~ s(Urea), data = data_ruido)
Hemoglobina_Creatina_Gravedad=gam(Hemoglobina ~ s(Creatinina) + s(Gravedad_2), data = df)
Potasio_Creatina=gam(Potasio ~ s(Creatinina), data = df)
Sodio_Creatina=gam(Sodio ~ s(Creatinina), data = df)
Red_Hemoglobina=gam(CelulasRojas ~ s(Hemoglobina),data = df)
Volume_Red = gam(Hematocrito ~ s(CelulasRojas),data = df)
Glucosa<-lm(Glucosa ~ 1, data = df)
Presion<-lm(Presion ~ 1, data = df)
Urea<-lm(Urea ~ 1, data = df)
```

Sacamos el BIC y el AIC del modelo no paramétrico de la DAG 3

```
BIC6=-(BIC(Urea) + BIC(Presion) + BIC(Glucosa) + BIC(Volume_Red))
#cat("BIC del modelo no paramétrico de DAG Biomédico Gutiérrez: ", BIC6)

AIC6=-(AIC(Urea) + AIC(Presion) + AIC(Glucosa) + AIC(Volume_Red))
#cat("AIC del modelo no paramétrico de DAG Biomédico Gutiérrez: ", AIC6)
```

Ahora quitamos además de edad las variables de potasio y sodio.

```
df <- df[, -6:-7]
head(df)
```

	Presion	Gravedad	Glucosa	Urea	Creatinina	Hemoglobina
Hematocrito						
1	80	1.020	121.0000	36	1.2	15.4
44						
2	50	1.020	148.0113	18	0.8	11.3
38						
3	80	1.010	423.0000	53	1.8	9.6
31						
4	70	1.005	117.0000	56	3.8	11.2
32						
5	80	1.010	106.0000	26	1.4	11.6
35						
6	90	1.015	74.0000	25	1.1	12.2
39						
CelulasBlancas						
1	7800					
2	6000					
3	7500					

4	6700	3.900000
5	7300	4.600000
6	7800	4.400000

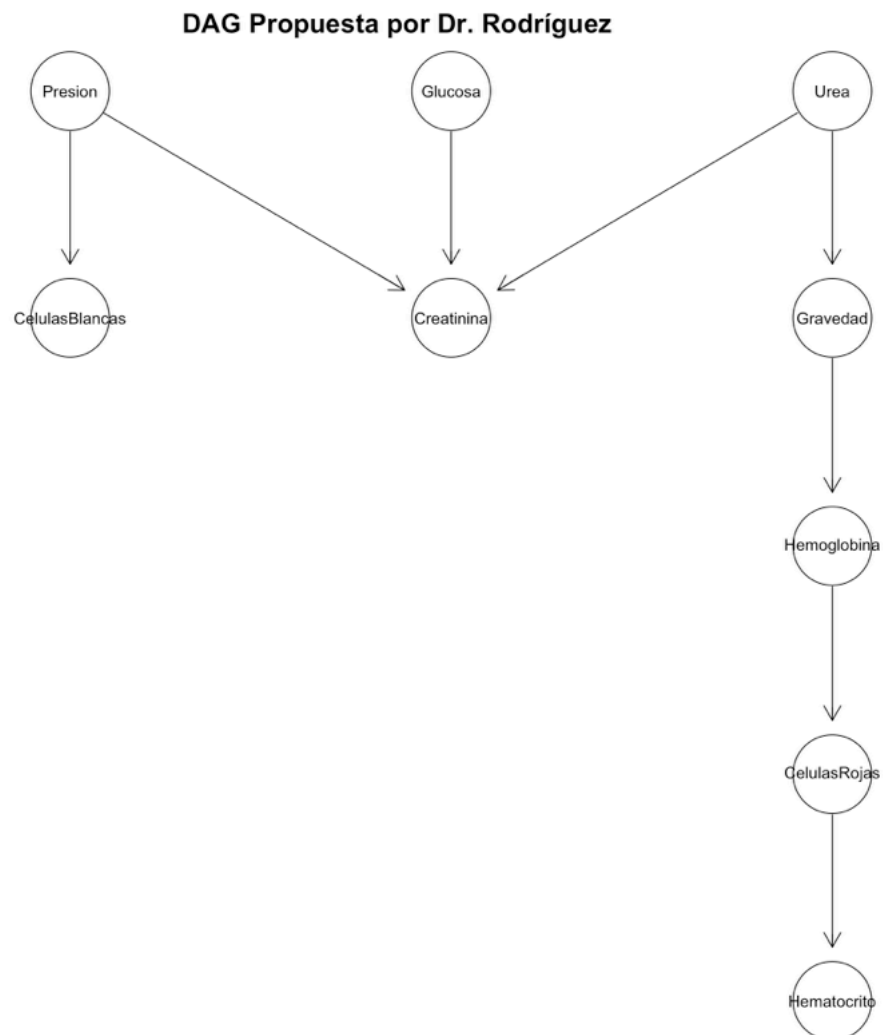
Se crea una cuarta DAG

```
dag3=model2network("[Urea] [Presion] [Glucosa] [Creatinina] [Glucosa]
```

Representación Gráfica

```
g <- Rgraphviz::layoutGraph(bnlearn::as.graphNEL(dag3))
graph::nodeRenderInfo(g) <- list(fontsize=22)
Rgraphviz::renderGraph(g)
# Create the title text
title_text <- "DAG Propuesta por Dr. Rodríguez"

# Add title using grid graphics
library(grid)
grid.text(label = title_text, x = 0.45, y = 0.99, gp = gpar(for
```



BIC y AIC Scores del Modelo Lineal de la DAG 4

```
#BIC Y AIC DEL MODELO LINEAL
BIC7=score(dag3, data = df, type = "bic-g")
#cat("BIC del modelo lineal de DAG Biomédico Gutiérrez:",BIC7,
AIC7=score(dag3, data = df, type = "aic-g")
#cat("AIC del modelo lineal de DAG Biomédico Gutiérrez:",AIC7,
```

Ajustamos cada una de las variables a nuestro modelo no paramétrico de la DAG 4

```
#Tuvimos que usar la gravedad_2 para que nuestras variables tuv
white_presion = gam(CelulasBlancas~s(Presion),data=df)
Creatinina_Glucosa_Presion_Urea=gam(Creatinina ~ s(Glucosa) + s
Gravedad_Urea=gam(gravedad_2 ~ s(Urea), data = data_ruido)
Hemoglobina_Gravedad=gam(Hemoglobina ~ s(gravedad_2), data = da
Red_Hemoglobina=gam(CelulasRojas ~ s(Hemoglobina),data = df)
Volume_Red = gam(Hematocrito ~ s(CelulasRojas),data = df)
Glucosa<-lm(Glucosa ~ 1, data = df)
Presion<-lm(Presion ~ 1, data = df)
Urea<-lm(Urea ~ 1, data = df)
```

Sacamos el BIC y el AIC del modelo no paramétrico de la DAG 4

```
BIC8=-(BIC(Urea) + BIC(Presion) + BIC(Glucosa) + BIC(Volume_Red
#cat("BIC del modelo no paramétrico de DAG Biomédico Gutiérrez:

AIC8=-(AIC(Urea) + AIC(Presion) + AIC(Glucosa) + AIC(Volume_Red
#cat("AIC del modelo no paramétrico de DAG Biomédico Gutiérrez:
```

Generamos una tabla con los scores de los modelos lineales para poder compararlos y ver el que mejor resultado tiene.

```
#install.packages("gt")
library(gt)
tab_lineal <- matrix(c(BIC1, AIC1, BIC3, AIC3, BIC5, AIC5, BIC7
colnames(tab_lineal) <- c('BIC Score','AIC Score')
rownames(tab_lineal) <- c('Modelo Lineal Equipo','Modelo Lineal
tab_lineal <- as.table(tab_lineal)
tab_lineal
```

	BIC Score	AIC Score
Modelo Lineal Equipo	-15980.29	-15904.59
Modelo Lineal IMD Gutiérrez	-15898.83	-15819.15
Modelo Lineal Dr. Michel	-14301.90	-14236.16

Modelo Lineal Dr. Rodríguez -11902.00 -11850.21

Generamos una tabla con los scores de los modelos no paramétricos para poder compararlos y ver el que mejor resultado tiene.

```
tab_nopara <- matrix(c(BIC2, AIC2, BIC4, AIC4, BIC6, AIC6, BIC8, AIC8),
  colnames(tab_nopara) <- c('BIC Score', 'AIC Score')
  rownames(tab_nopara) <- c('Modelo no paramétrico Equipo', 'Modelo no paramétrico IMD Gutiérrez', 'Modelo no paramétrico Dr. Michel', 'Modelo no paramétrico Dr. Rodríguez')
  tab_nopara <- as.table(tab_nopara)
  tab_nopara
```

	BIC Score	AIC Score
Modelo no paramétrico Equipo	-15937.66	-15782.88
Modelo no paramétrico IMD Gutiérrez	-15673.87	-15489.23
Modelo no paramétrico Dr. Michel	-13997.15	-13836.45
Modelo no paramétrico Dr. Rodríguez	-11897.50	-11788.12

Se puede observar que la DAG 4 es a que mejor BIC Score, esta se usará para la probabilidad condicional.

Ahora generaremos queries para responder probabilidad condicional usando la DAG con mejor BIC Score. Primero necesitamos entrenar la DAG con los datos para responder las queries.

```
bn<-bn.fit(dag3, data = df)
```

Revisamos el nombre de las variables

```
colnames(df)
```

```
[1] "Presion"      "Gravedad"     "Glucosa"      "Urea"
[5] "Creatinina"   "Hemoglobina"  "Hematocrito"
"CelulasBlancas"
[9] "CelulasRojas"
```

Query 1. ¿Cuál es la probabilidad de que un individuo tenga más de 11000 células blancas en la sangre dado que su presión diastólica esté entre 80 y 110?

```
cpquery(bn, event = (CelulasBlancas >= 11000) , evidence = ((PresionDiastolica >= 80) && (PresionDiastolica <= 110)))
```

```
[1] 0.1595416
```

Query 2. Si una persona tiene urea en la sangre mayor a 40,

¿cuál es la probabilidad de que la densidad de su sangre esté entre 1.005 y 1.015?

```
cpquery(bn, event = ((Gravedad >= 1.005) & (Gravedad <= 1.015)))
```

```
[1] 0.3715671
```

Query 3. ¿Cuál es la probabilidad de que un paciente de enfermedad crónica renal tenga un volumen celular entre 20 y 45, dado que su conteo de células rojas sea menor o igual a 4.3 y su hemoglobina en la sangre menor a 13?

```
cpquery(bn, event = ((Hematocrito >= 20) & (Hematocrito <= 45)))
```

```
[1] 0.9416449
```

Query 4. ¿Cuál es la probabilidad de que un paciente de enfermedad crónica renal tenga un conteo de células blancas menor a 3000, dado que su presión está entre 105 y 130?

```
cpquery(bn, event = (CelulasBlancas <= 3000) , evidence = ((Presion >= 105) & (Presion <= 130)))
```

```
[1] 0.01533568
```