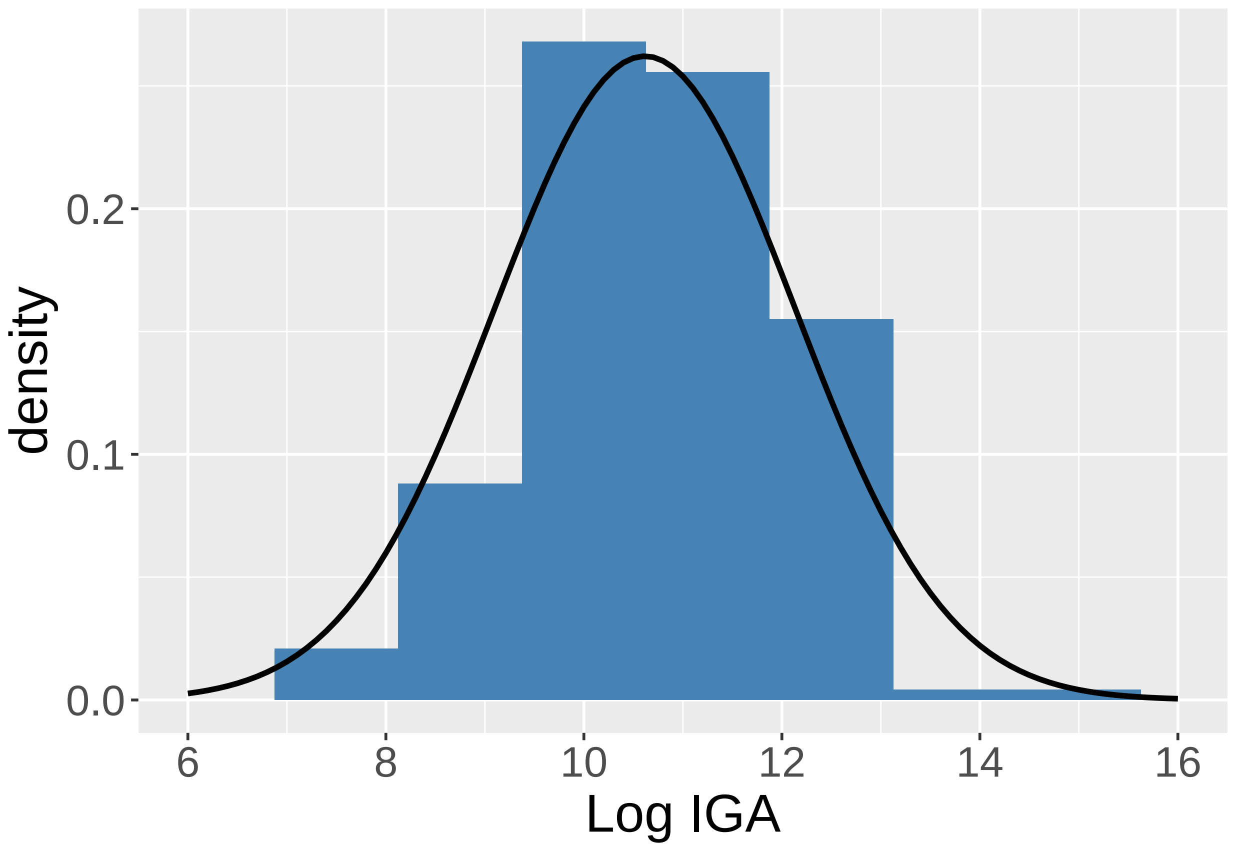
Informe

Javier Savedra - Thomas Gonzalez

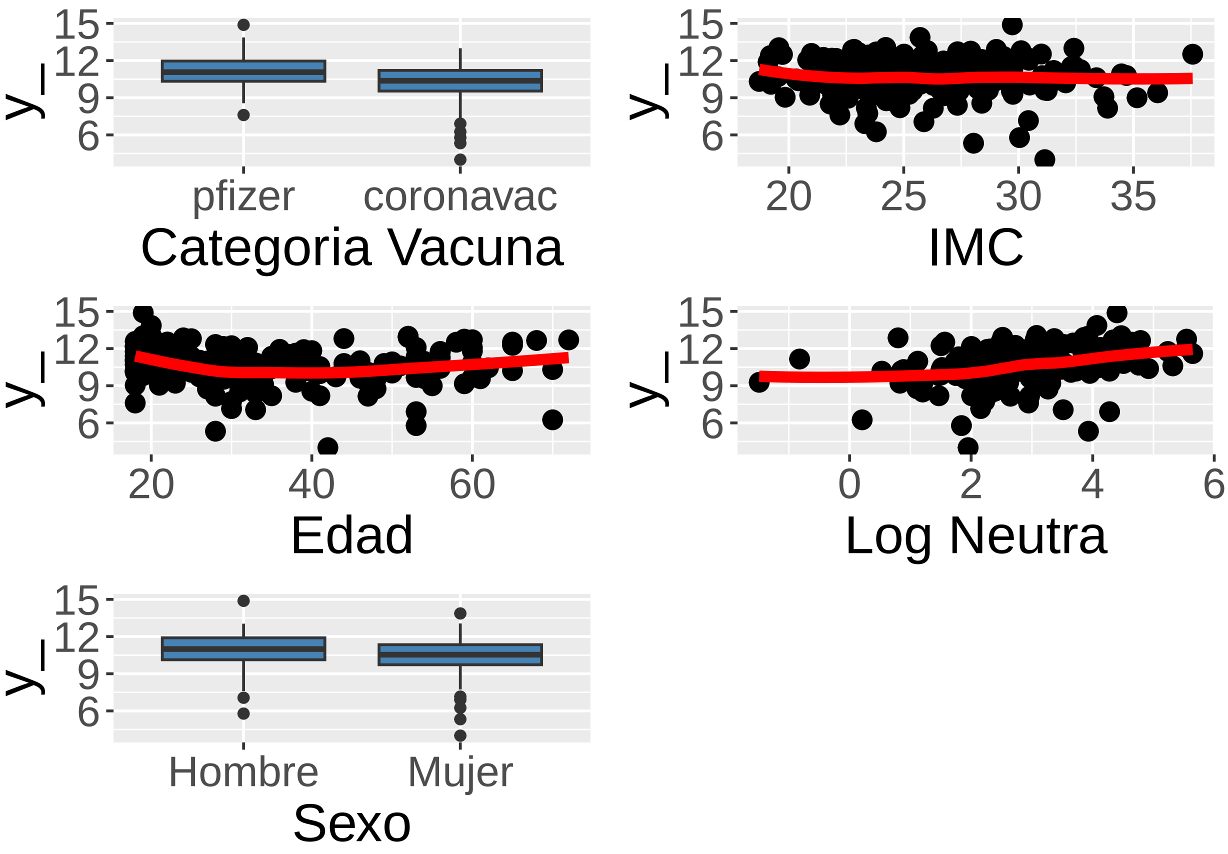
2024-03-01

```r
rm(list=ls())
#load("Aplicacion_thomas_javier_log_iga.RData")
load("~/tesis_R/Aplicacion_thomas_javier_log_iga.RData")
library(ggplot2)
library(gridExtra)
library(gamlss)

dados <- datos
y <- y_


v1 <- ggplot(dados, aes(x = y_)) +  # Agrega aes() alrededor de y_
  geom_histogram(bins = 9, aes(y = ..density..), fill = "steelblue") +
  theme(
    axis.text.x = element_text(size = 16, face = "plain"),
    axis.text.y = element_text(size = 16, face = "plain"),
    axis.title.x = element_text(size = 20, face = "plain"),
    axis.title.y = element_text(size = 20, face = "plain")
  ) +
  stat_function(
    fun = dNO,
    args = list(mu = mean(y_), sigma = sd(y_)),  # Utiliza y_ en lugar de y
    col = "black",
    size = 1
  ) +
  xlim(6,16) +
  xlab("Log IGA")


v1
```

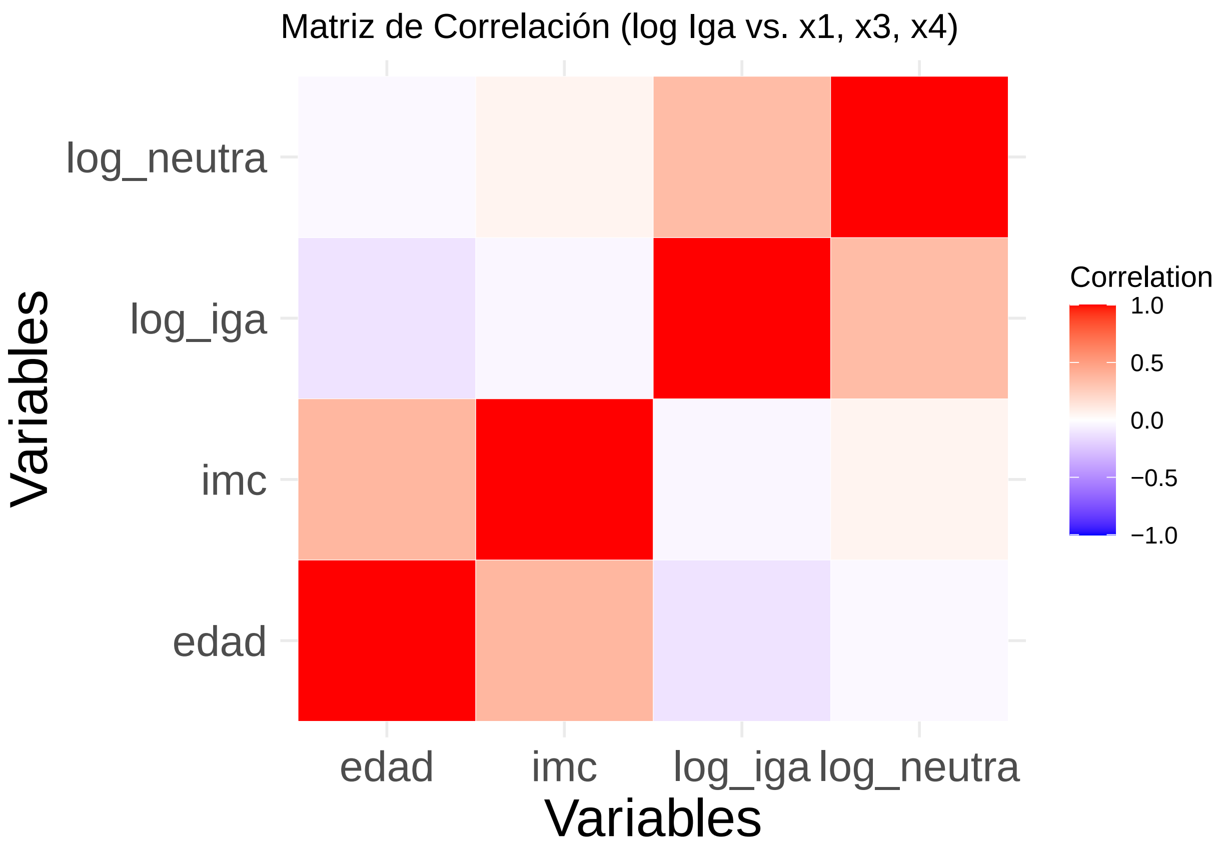


```
v2 <- ggplot(dados, aes(x2, y_)) +
  geom_boxplot(fill = "steelblue") +
  theme(
    axis.text.x = element_text(size = 16, face = "plain"),
    axis.text.y = element_text(size = 16, face = "plain"),
    axis.title.x = element_text(size = 20, face = "plain"),
    axis.title.y = element_text(size = 20, face = "plain")
  )+
  xlab("Categoria Vacuna")

# Grafico suavizado para IMC (v3)
v3 <- ggplot(dados, aes(x1, y_)) +
  geom_point(size = 3) +
  geom_smooth(se = FALSE, method = "loess", col = "red", size = 2) +
  theme(
    axis.text.x = element_text(size = 16, face = "plain"),
    axis.text.y = element_text(size = 16, face = "plain"),
    axis.title.x = element_text(size = 20, face = "plain"),
    axis.title.y = element_text(size = 20, face = "plain")
  ) +
  xlab("IMC")

# Grafico suavizado para Log Neutra (v4)
v4 <- ggplot(dados, aes(x3, y_)) +
  geom_point(size = 3) +
  geom_smooth(se = FALSE, method = "loess", col = "red", size = 2) +
```

```r
  theme(
    axis.text.x = element_text(size = 16, face = "plain"),
    axis.text.y = element_text(size = 16, face = "plain"),
    axis.title.x = element_text(size = 20, face = "plain"),
    axis.title.y = element_text(size = 20, face = "plain")
  ) +
  xlab("Log Neutra")

# Grafico suavizado para Edad (v5)
v5 <- ggplot(dados, aes(x4, y_)) +
  geom_point(size = 3) +
  geom_smooth(se = FALSE, method = "loess", col = "red", size = 2) +
  theme(
    axis.text.x = element_text(size = 16, face = "plain"),
    axis.text.y = element_text(size = 16, face = "plain"),
    axis.title.x = element_text(size = 20, face = "plain"),
    axis.title.y = element_text(size = 20, face = "plain")
  ) +
  xlab("Edad")
v6 <- ggplot(dados, aes(x5, y_)) +
  geom_boxplot(fill = "steelblue") +
  theme(
    axis.text.x = element_text(size = 16, face = "plain"),
    axis.text.y = element_text(size = 16, face = "plain"),
    axis.title.x = element_text(size = 20, face = "plain"),
    axis.title.y = element_text(size = 20, face = "plain")
  )+
  xlab("Sexo")

grid.arrange(v2, v3, v5, v4,v6)
```



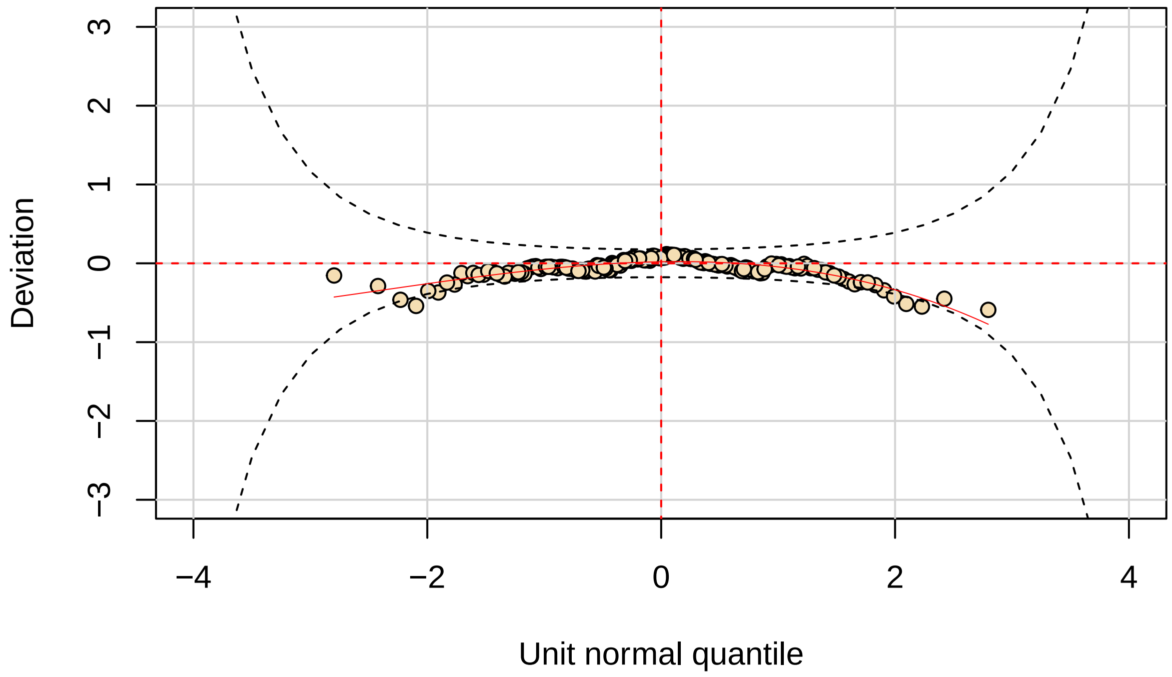
```r
# Calcular la matriz de correlación
correlation_matrix_log_iga <- cor(dados1[, c("log_iga", "imc", "edad", "log_neutra")])

# Convertir la matriz de correlación en un dataframe
correlation_df <- as.data.frame(correlation_matrix_log_iga)
correlation_df$variable1 <- rownames(correlation_matrix_log_iga)
correlation_df <- tidyr::gather(correlation_df, variable2, value, -variable1)

# Crear un gráfico de heatmap para la matriz de correlación
heatmap_plot_log_iga <- ggplot(data = correlation_df, aes(variable1, variable2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space =
  theme_minimal() +
  theme(axis.text.x = element_text(size = 16, face = "plain"),
        axis.text.y = element_text(size = 16, face = "plain"),
        axis.title.x = element_text(size = 20, face = "plain"),
        axis.title.y = element_text(size = 20, face = "plain")) +
  labs(x = "Variables", y = "Variables", title = "Matriz de Correlación (log Iga vs. x1, x3, x4)")


heatmap_plot_log_iga
```



```
GAIC(Normal_identidade,
     tStudent_identidade,
     Exponencial_potencia_identidade,
     Normal_inversa,
     tStudent_inversa,
     Exponencial_potencia_inversa,
     Normal_log,
     tStudent_log,
     Exponencial_potencia_log,
     Normal_sqrt,
     tStudent_sqrt,
     Exponencial_potencia_sqrt)
```

```
##                                       df      AIC
## tStudent_identidade             9.000750 685.3184
## tStudent_sqrt                   9.000349 685.6077
## tStudent_log                    9.000447 685.9018
## tStudent_inversa                9.000620 686.4969
## Exponencial_potencia_identidade 9.000750 687.0457
## Exponencial_potencia_sqrt       9.000366 687.2355
## Exponencial_potencia_log        9.000481 687.4457
## Exponencial_potencia_inversa    9.000659 687.9640
## Normal_identidade               8.000750 696.6358
## Normal_sqrt                     8.000330 696.8888
## Normal_log                      8.000410 697.1445
## Normal_inversa                  8.000556 697.6585
```

```
wp(tStudent_identidade,ylim.all = 3)
```



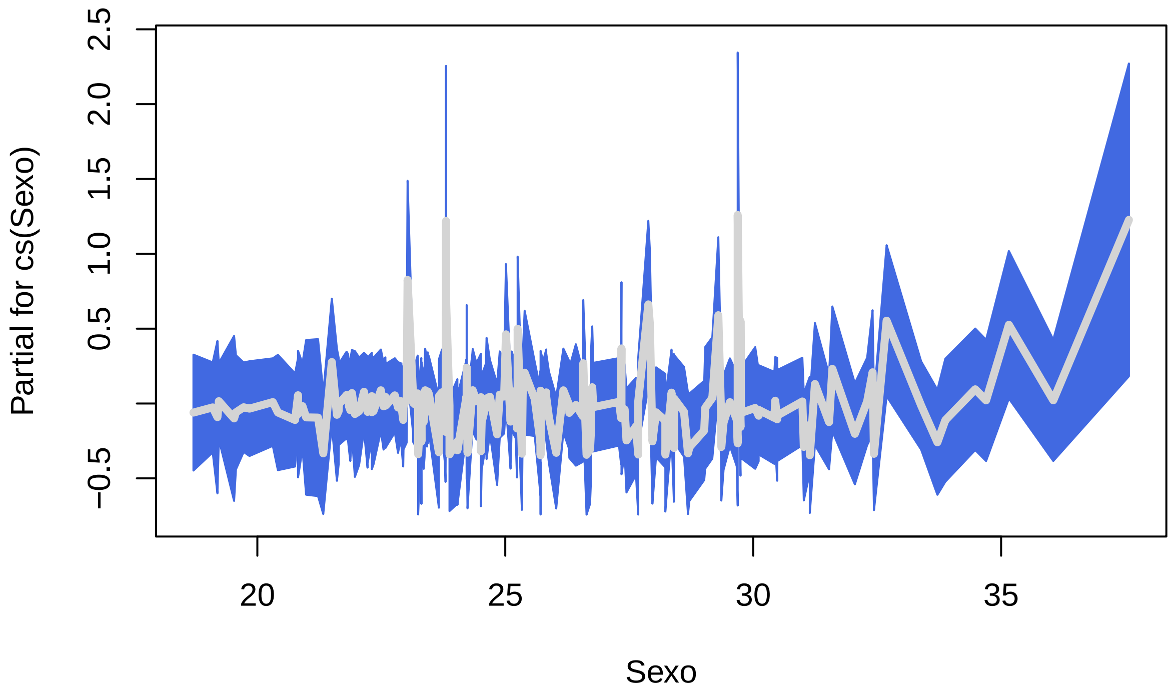
```
summary(tStudent_identidade)
```

```
## ******************************************************************
## Family:  c("TF", "t Family")
##
## Call:
## gamlss(formula = y_ ~ x2 + x5 + cs(x1 + x3 + x4), family = TF(mu.link = identity,
##     sigma.link = log), data = dados1, n.cyc = 1000,      trace = TRUE)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  identity
## Mu Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      11.110192   0.377893  29.400  < 2e-16 ***
## x2coronavac      -0.685132   0.193019  -3.550 0.000489 ***
## x5Mujer          -0.487217   0.191371  -2.546 0.011714 *
## cs(x1 + x3 + x4)  0.004880   0.006131   0.796 0.427108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## Sigma link function:  log
## Sigma Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.08828    0.08841   0.999    0.319
##
## ------------------------------------------------------------------
## Nu link function:  log
## Nu Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.587      0.353   4.497 1.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## NOTE: Additive smoothing terms exist in the formulas:
##  i) Std. Error for smoothers are for the linear effect only.
## ii) Std. Error for the linear terms maybe are not accurate.
## ------------------------------------------------------------------
## No. of observations in the fit:  194
## Degrees of Freedom for the fit:  9.00075
##       Residual Deg. of Freedom:  184.9993
##                       at cycle:  7
##
## Global Deviance:     667.3169
##             AIC:     685.3184
##             SBC:     714.7316
## ******************************************************************
```

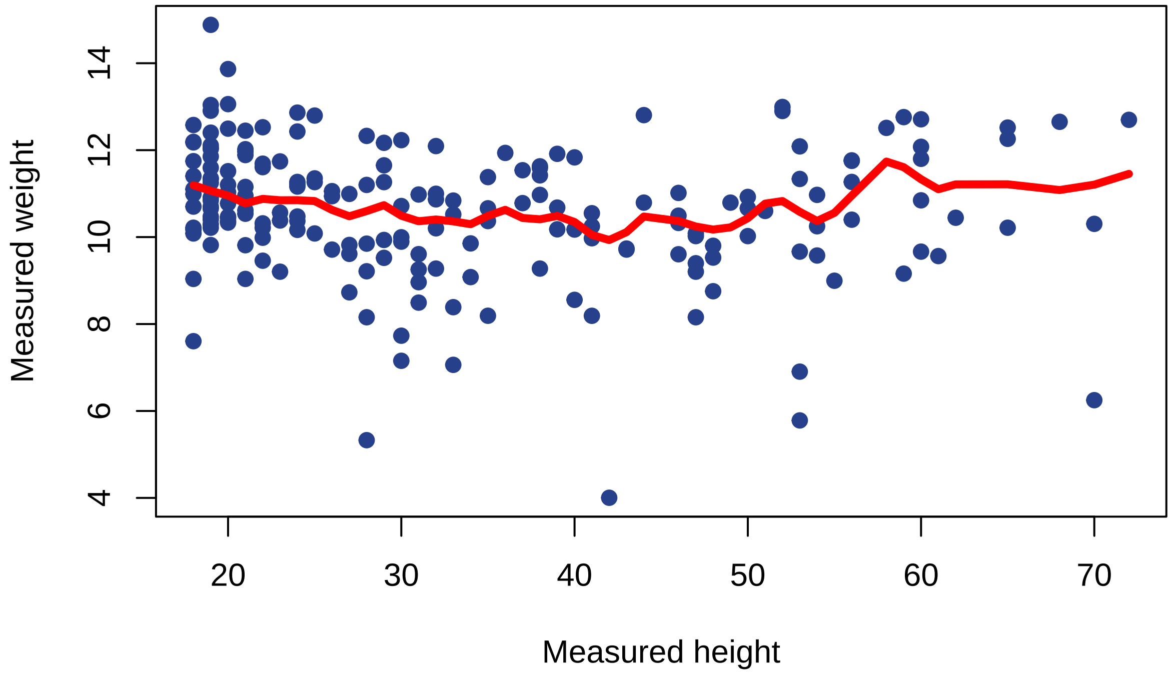
```
exp(tStudent_identidade$nu.coefficients)
```

```
## (Intercept)
##    4.891035
```

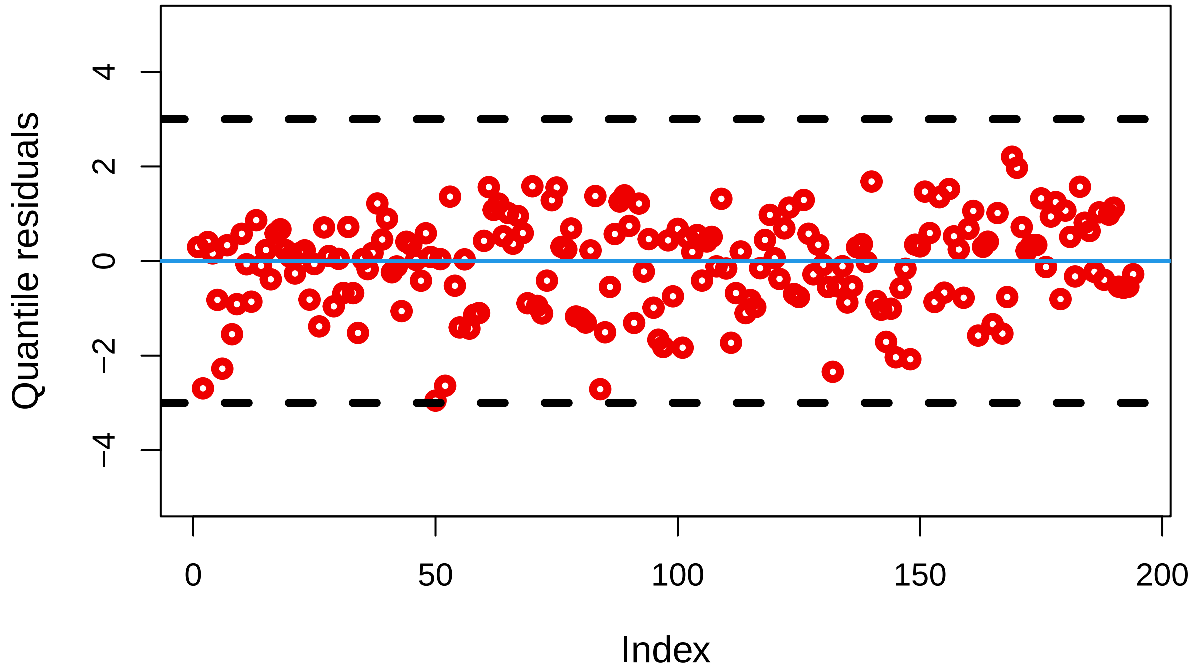
```
#x11()
term.plot(tStudent_identidade,
          what = c("mu"),
          terms = 3,main="",
          col.shaded="royalblue",
          col.term = "gray83",
          xlabs="Sexo",col.se="black",
          ylabs="Partial for cs(Sexo)",
          lwd.term = 4,
          lwd.se=2,
          lty.se = 1)
```



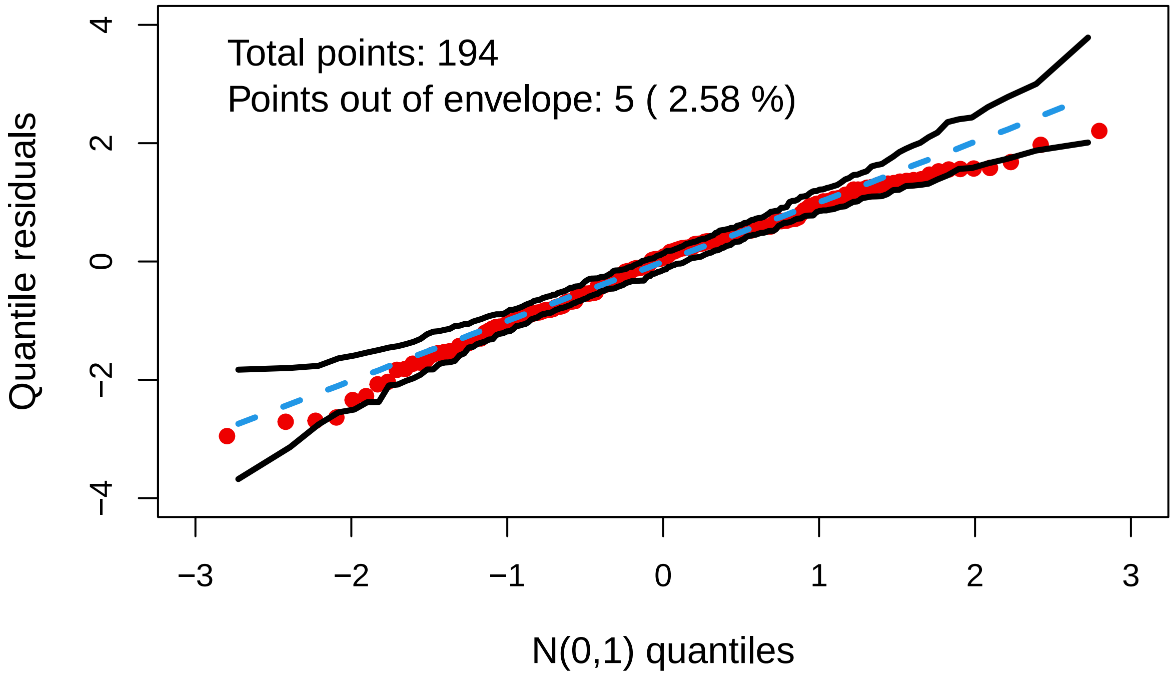
```
#______________________________________________________________________
#Tendencias y vs x's

#y vs x2
#x11()
plot(x4,y_,pch=19,main="",col="royalblue4",
     xlab="Measured height",
     ylab="Measured weight")
lines(smooth.spline(x4,fitted(tStudent_identidade)),
     type="l",col="red",lwd=4)
```



```
#_____________________________________________________________
#           Qualidade de ajuste da dist. t-Student,
#justificativa para dizer que a dist. t-Student n?o
#ajustou-se bem aos dados

#_______Gr?fico dos res?duos vs indices
Res.q2 <- tStudent_identidade$residuals
Res.qo2 <- sort(abs(Res.q2))

index <- 1:length(dados1$log_iga)

#x11()
plot(Res.q2,pch=1,col="red2",lwd=4,
     ylab="Quantile residuals",xlab="Index",
     main ="",ylim = c(-5,5),cex.lab=1.2)
abline(h=-3,lwd=4,lty=2,col="1")
abline(h=0,lwd=2,lty=1,col="4")
abline(h=3,lwd=4,lty=2,col="1")
```



```
#identify(index,Res.q2)
##x11()
qqnorm(Res.qo,pch=19,col="red2",
       xlim=c(-3,3),
       ylim=c(-4,4),
       ylab="Quantile residuals",
       xlab="N(0,1) quantiles",main = "",cex.lab=1.2)
lines(Z,rqmax,col="1",lwd=3)
lines(Z,rqm , lty = 2,col="4",lwd=3)
lines(Z,rqmin,col="1",lwd=3)
legend("topleft", c(paste("Total points:", n),
       paste("Points out of envelope:",
       res.out , "(", per.out, "%)")),
       bty="n", cex=1.2)
```


Total points: 194
Points out of envelope: 5 (2.58 %)
Quantile residuals
N(0,1) quantiles
−4
−2
0
2
4
−3
−2
−1
0
1
2
3