

Ejercicio 3.2

Silvia Pineda

Carga de Datos y Librerías

```
library(naniar)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.1      v stringr    1.5.2
v ggplot2    4.0.0      v tibble     3.3.0
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.1.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(rio)

data <- read.csv("students_FP.csv",
  na.strings = c("", "NA", "NaN", "NULL"),
  stringsAsFactors = TRUE
)
```

1.Imputación de las variables MCAR

La variable `program` y `study_mode` tienen $< 5\%$ y ambas son cualitativas, por tanto la forma más simple de imputarlas sería por la moda.

```
#Comprobar la categoría más frecuente
prop.table(table(data$program,useNA = "always"))
```

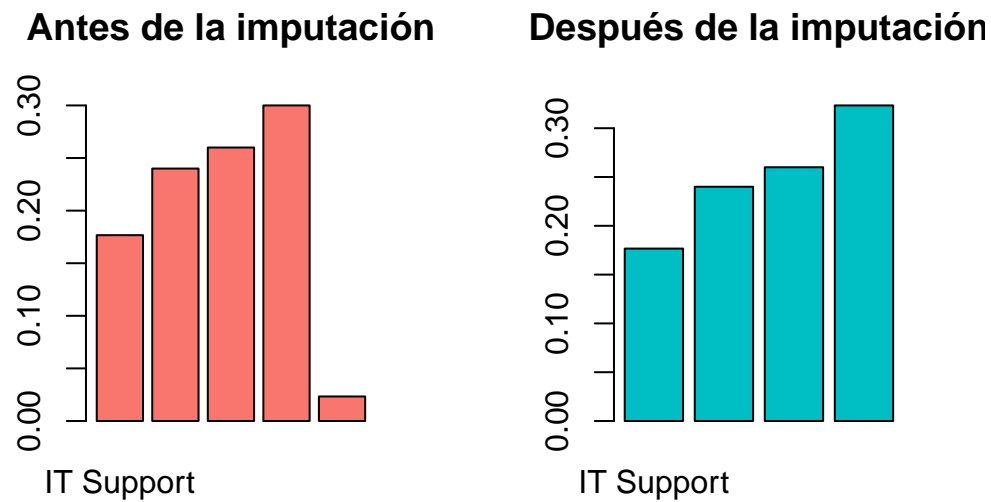
	IT Support Network Administration	Software Engineering
	0.1766667	0.2400000
Web Development	<NA>	0.2600000
	0.3000000	0.0233333

```
## Sustituir los NA con la categoría más frecuente
data$program_imp <-data$program
data$program_imp[is.na(data$program_imp)] <- "Web Development"

## Volvemos a comprobar
prop.table(table(data$program_imp,useNA = "always"))
```

	IT Support Network Administration	Software Engineering
	0.1766667	0.2400000
Web Development	<NA>	0.2600000
	0.3233333	0.0000000

```
# Graficar la distribución de la variable categórica antes y después de la imputación
par(mfrow = c(1, 2)) # Organizar las gráficas en una fila de 2 columnas
barplot(prop.table(table(data$program, useNA = "ifany")),
        main = "Antes de la imputación", col = "#F8766D" )
barplot(prop.table(table(data$program_imp)),
        main = "Después de la imputación",col = "#00BFC4")
```



El % de datos missing de la variable program es muy bajo y al imputar por la moda, no cambia la distribución en absoluto, por tanto, lo asumiremos como una buena imputación

```
#Comprobar la categoría más frecuente
prop.table(table(data$study_mode,useNA = "always"))
```

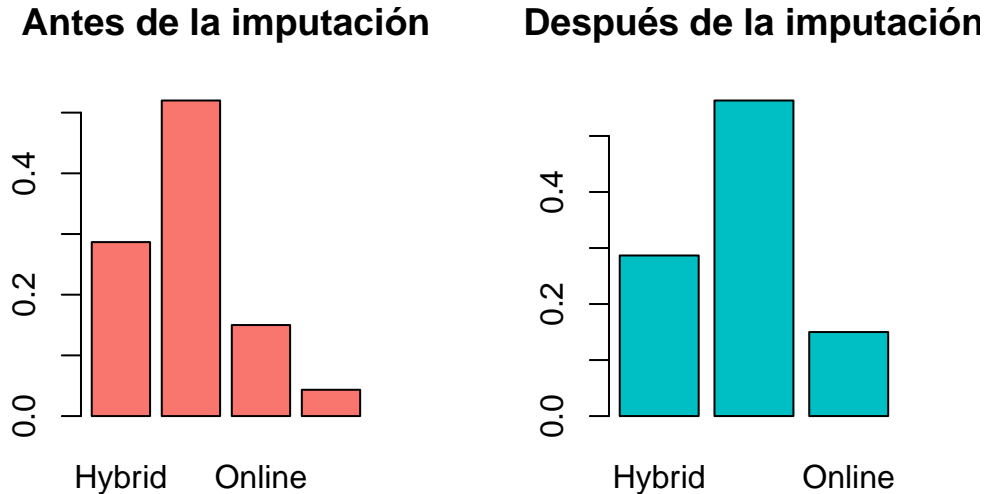
```
      Hybrid On-campus      Online      <NA>
0.28666667 0.52000000 0.15000000 0.04333333
```

```
## Sustituir los NA con la categoría más frecuente
data$study_mode_imp <-data$study_mode
data$study_mode_imp[is.na(data$study_mode_imp)] <- "On-campus"
```

```
## Volvemos a comprobar
prop.table(table(data$study_mode_imp,useNA = "always"))
```

```
      Hybrid On-campus      Online      <NA>
0.28666667 0.56333333 0.15000000 0.00000000
```

```
# Graficar la distribución de la variable categórica antes y después de la imputación
par(mfrow = c(1, 2)) # Organizar las gráficas en una fila de 2 columnas
barplot(prop.table(table(data$study_mode, useNA = "ifany")),
        main = "Antes de la imputación", col = "#F8766D" )
barplot(prop.table(table(data$study_mode_imp)),
        main = "Después de la imputación", col = "#00BFC4")
```



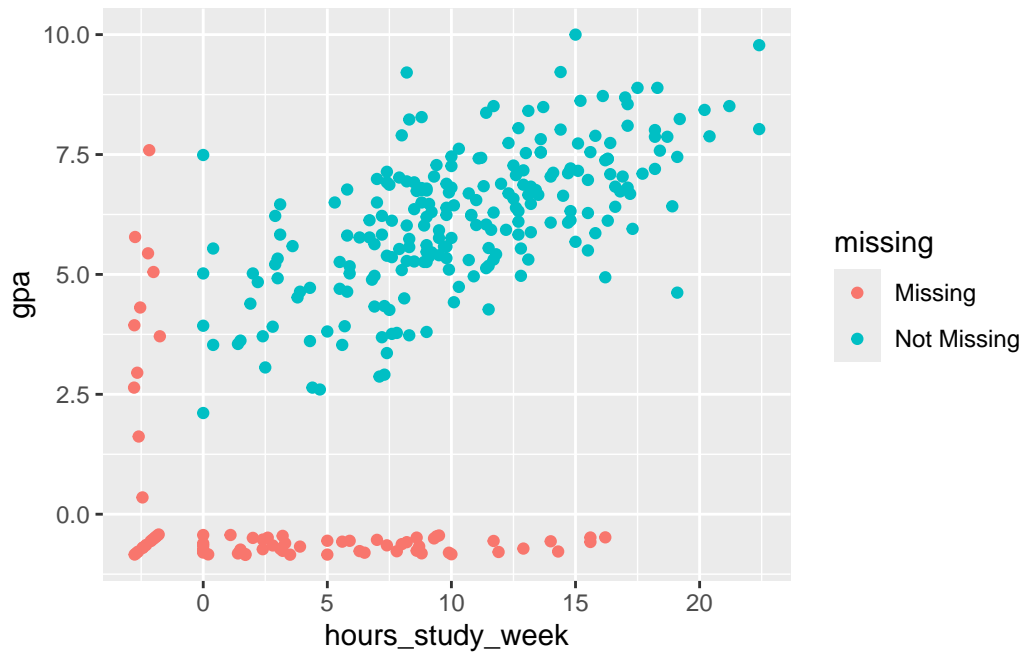
El % de datos missing de la variable `study_program` es muy bajo y al imputar por la moda, no cambia la distribución en absoluto, por tanto, lo asumiremos como una buena imputación

2.Imputación de las variables MAR/MNAR

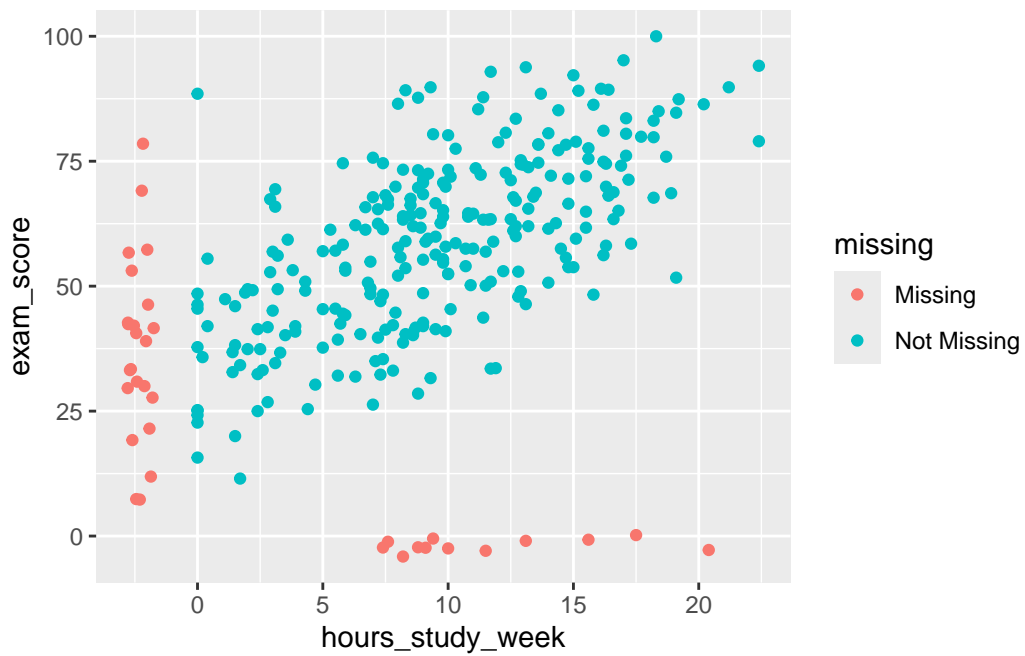
- Variable `hours_study_week`

Los NA de esta variable son MAR y corresponden a un 8%, además la variable está asociada con `gpa` y `exam_score`:

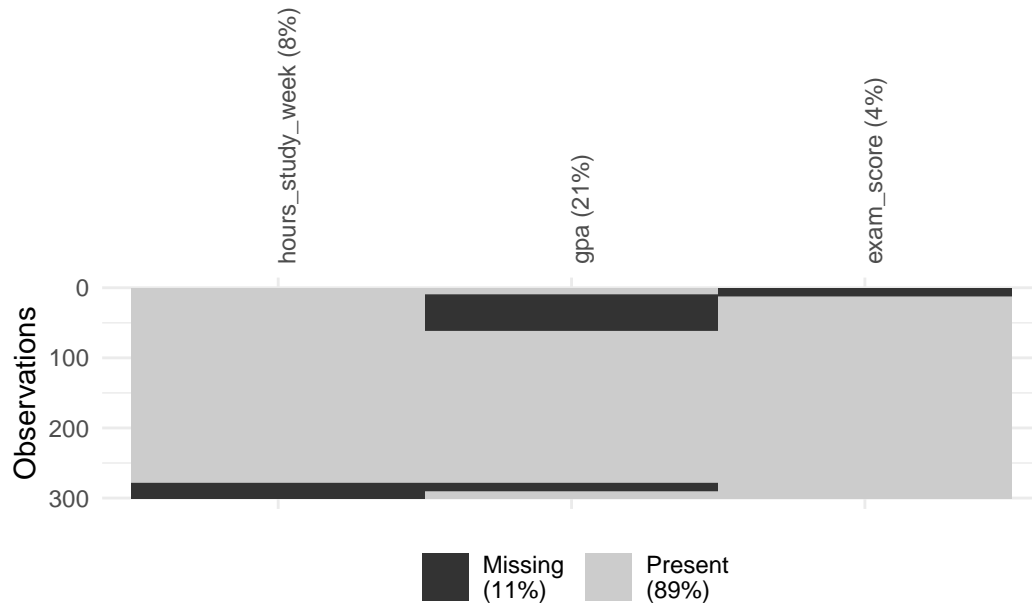
```
ggplot(data = data, aes (x = hours_study_week ,
                        y =gpa )) + geom_miss_point()
```



```
ggplot(data = data, aes (x = hours_study_week ,
                          y =exam_score )) + geom_miss_point()
```



```
vis_miss(select(data, hours_study_week, gpa, exam_score), cluster=TRUE) +
  theme(axis.text.x = element_text(angle = 90))
```



Como solo están completos los datos de `exam_score` para los NA de `hours_study_week`, la imputaremos mediante una regresión:

```
#Ajustar un modelo de regresión lineal
model1 <- lm(hours_study_week ~ exam_score, data = data)

#Predecir los valores solo para las observaciones faltantes
predictions <- predict(model1, newdata = data [is.na(data$hours_study_week),])

##Crear una nueva variable imputada
data$hours_study_week_imp_model1 <- data$hours_study_week
data$hours_study_week_imp_model1[is.na(data$hours_study_week)] <- predictions

summary(model1)
```

Call:

```
lm(formula = hours_study_week ~ exam_score, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.154	-2.908	0.145	2.647	10.931

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.64250	0.84511	-1.944	0.053 .
exam_score	0.18979	0.01368	13.871	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.908 on 263 degrees of freedom

(35 observations deleted due to missingness)

Multiple R-squared: 0.4225, Adjusted R-squared: 0.4203

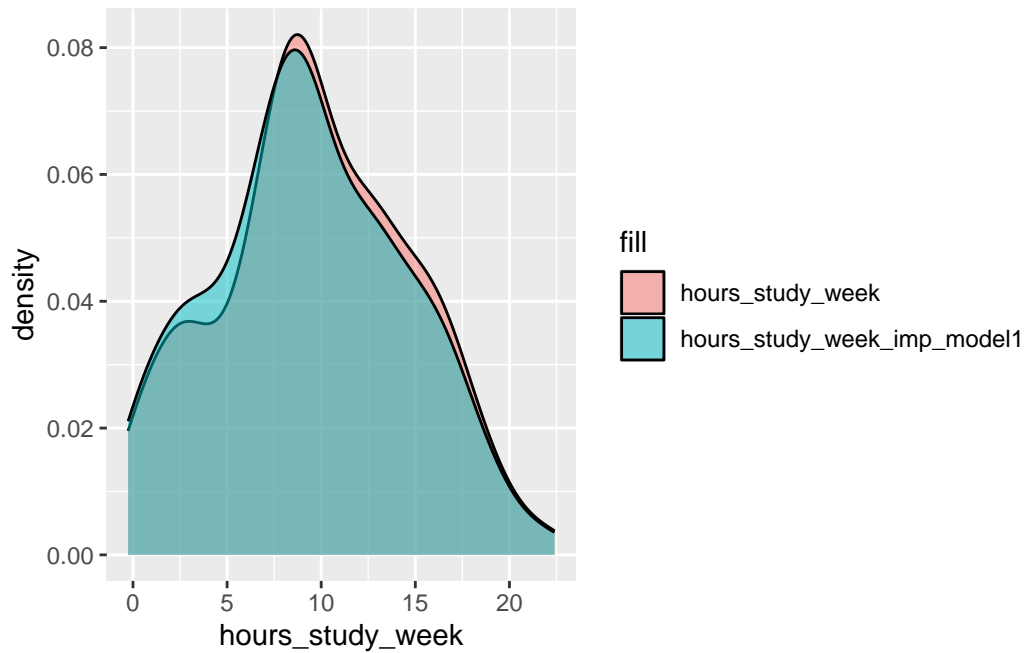
F-statistic: 192.4 on 1 and 263 DF, p-value: < 2.2e-16

```
#Hacer un gráfico para comparar las observaciones
```

```
ggplot(data, aes(x = hours_study_week, fill = "hours_study_week")) +  
  geom_density(alpha = 0.5) +
```

```
geom_density(aes(x = hours_study_week_imp_model1, fill = "hours_study_week_imp_model1"), alp
```

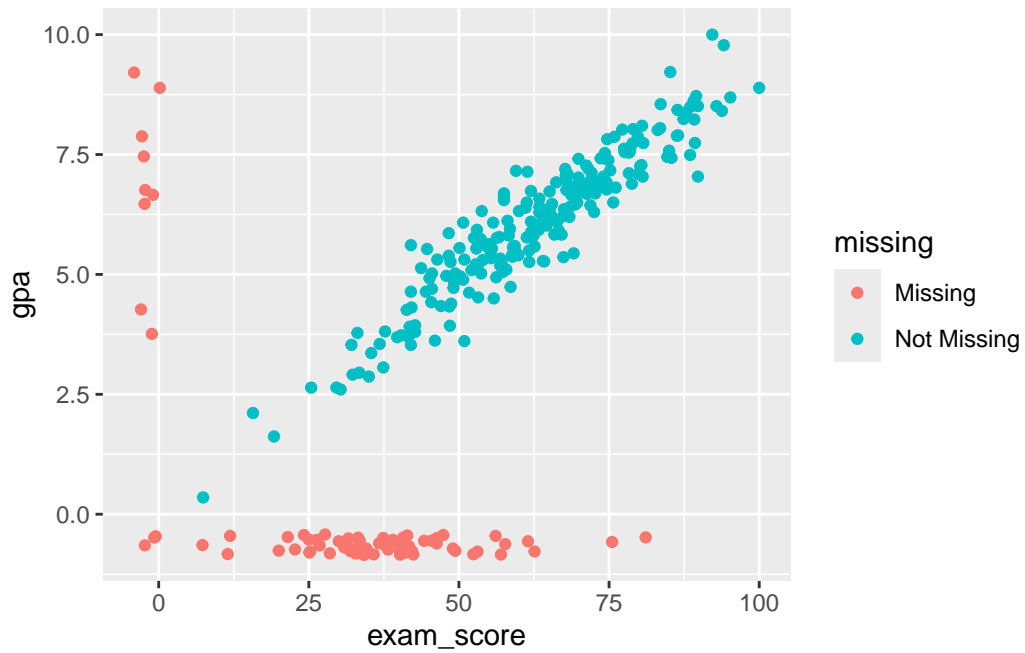
Warning: Removed 23 rows containing non-finite outside the scale range
(`stat_density()`).



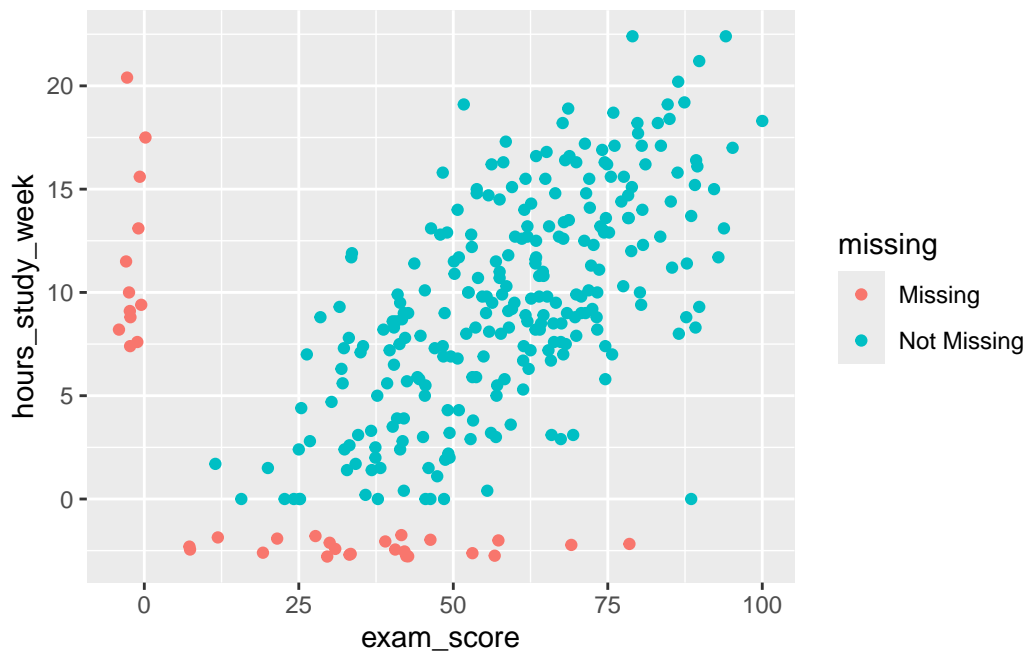
- Variable exam_score

Los NA de esta variable podrían ser MAR si corresponden a un profesor que no ha subido las notas o MNAR si correspondieran a aquellos que han suspendido.

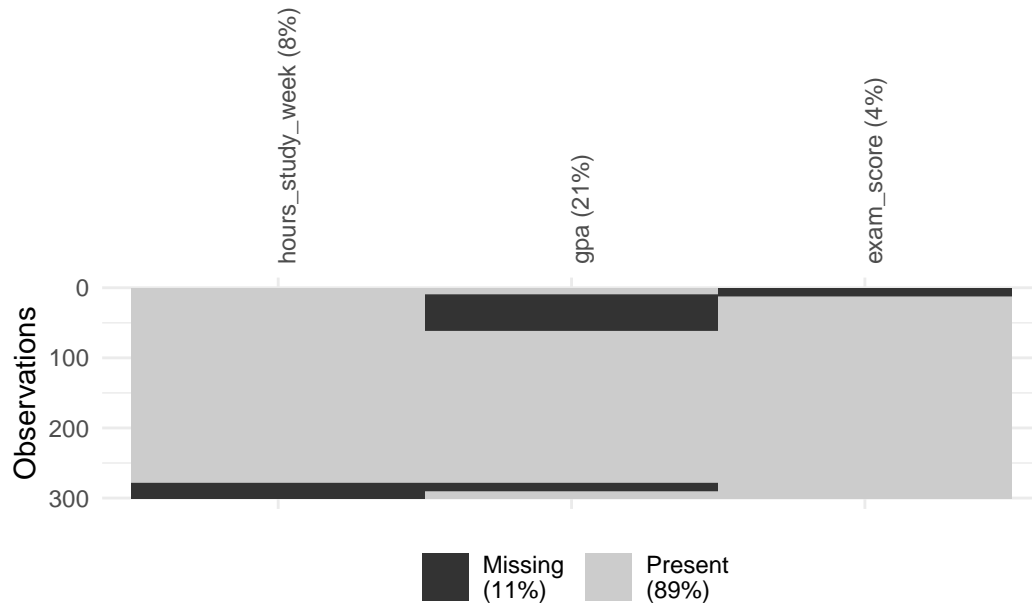
```
ggplot(data = data, aes (x = exam_score ,  
                          y =gpa )) + geom_miss_point()
```

```
ggplot(data = data, aes (x = exam_score ,
                          y =hours_study_week )) + geom_miss_point()
```



```
vis_miss(select(data, hours_study_week, gpa, exam_score), cluster=TRUE) +
  theme(axis.text.x = element_text(angle = 90))
```



Como solo están completos los datos de hours_study_week para los NA de exam_score, la imputaremos mediante una regresión:

```
#Ajustar un modelo de regresión lineal
model1 <- lm(exam_score ~ hours_study_week, data = data)

#Predecir los valores solo para las observaciones faltantes
predictions <- predict(model1, newdata = data [is.na(data$exam_score),])

##Crear una nueva variable imputada
data$exam_score_imp_model1 <- data$exam_score
data$exam_score_imp_model1[is.na(data$exam_score_imp_model1)] <- predictions

summary(model1)
```

Call:

```
lm(formula = exam_score ~ hours_study_week, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-30.748	-10.156	0.564	8.443	50.643

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	37.8574	1.7460	21.68	<2e-16 ***
hours_study_week	2.2261	0.1605	13.87	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.39 on 263 degrees of freedom

(35 observations deleted due to missingness)

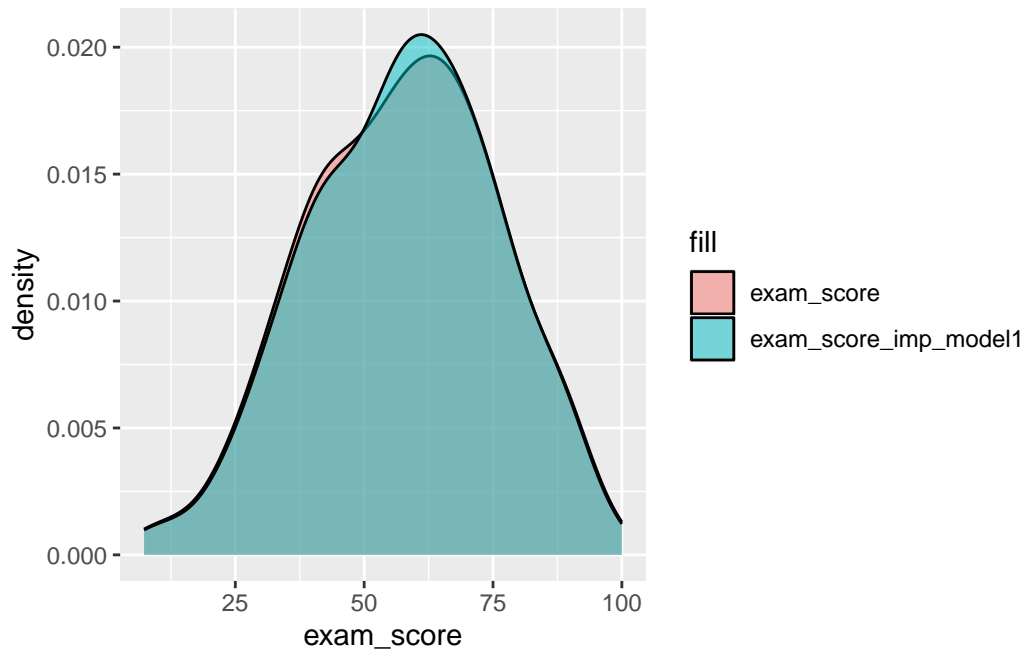
Multiple R-squared: 0.4225, Adjusted R-squared: 0.4203

F-statistic: 192.4 on 1 and 263 DF, p-value: < 2.2e-16

```
#Hacer un gráfico para comparar las observaciones
ggplot(data, aes(x = exam_score, fill = "exam_score")) +
  geom_density(alpha = 0.5) +

geom_density(aes(x = exam_score_imp_model1,
                 fill = "exam_score_imp_model1"), alpha = 0.5)
```

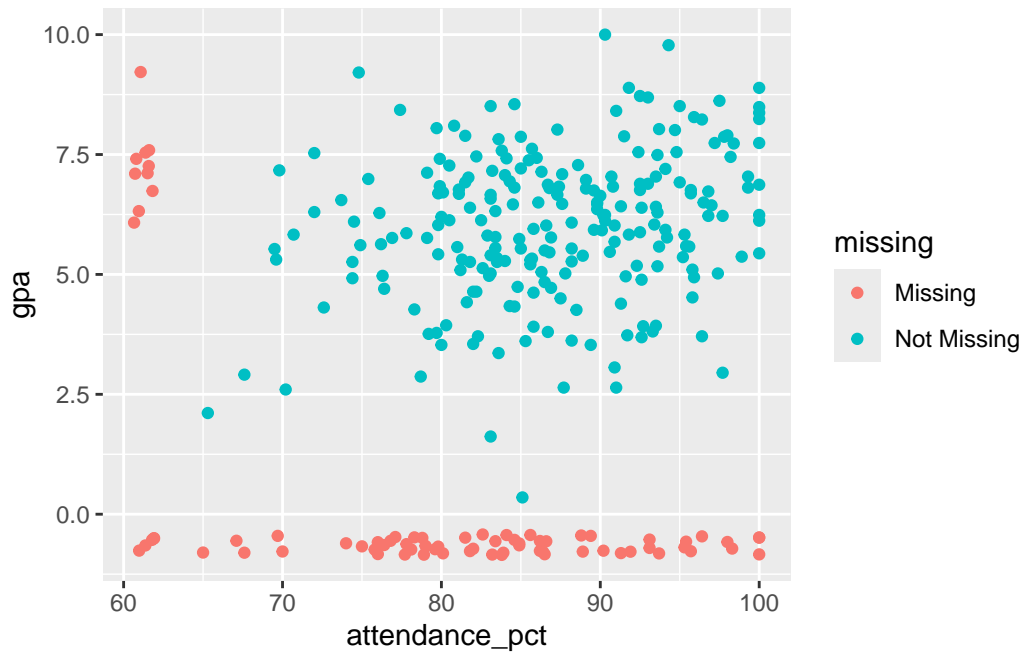
Warning: Removed 12 rows containing non-finite outside the scale range
(`stat_density()`).



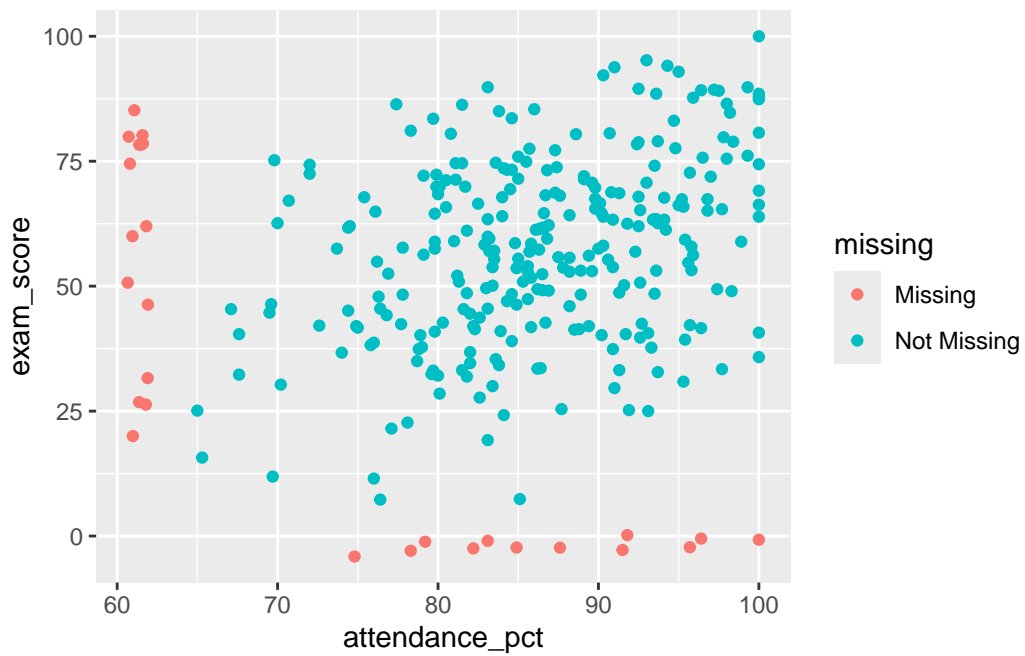
- Variable attendance_pct

Los NA de `attendance_pct` son MNAR, pero con un % pequeño y asociados a valores altos de `gpa`, además esta variable está asociada a `exam_score` y todos sus missing están completos

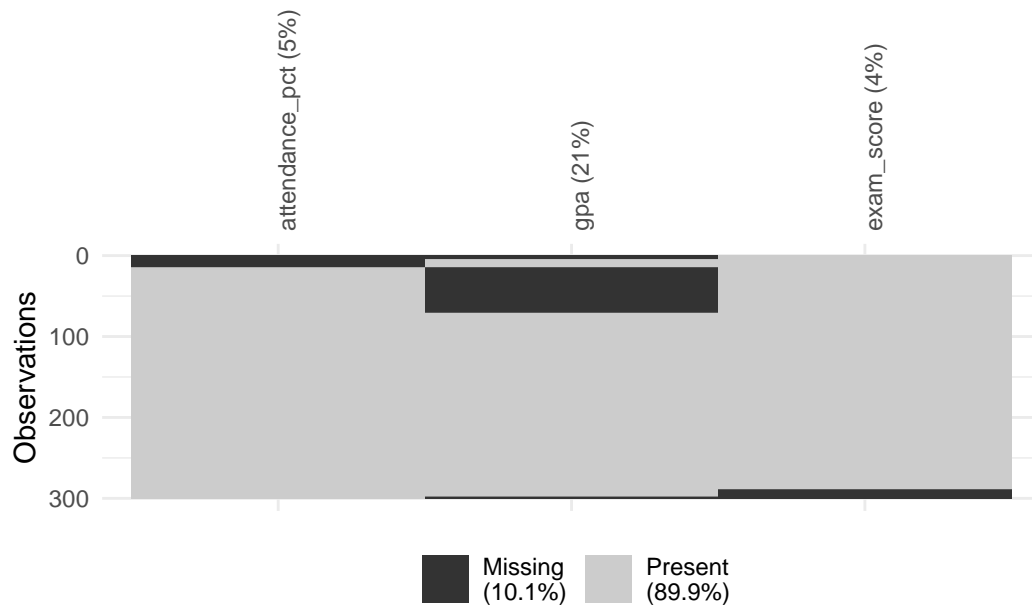
```
ggplot(data = data, aes (x = attendance_pct ,  
                        y =gpa )) + geom_miss_point()
```



```
ggplot(data = data, aes (x = attendance_pct ,
                          y = exam_score )) + geom_miss_point()
```



```
vis_miss(select(data, attendance_pct, gpa, exam_score), cluster=TRUE) +  
  theme(axis.text.x = element_text(angle = 90))
```



Como solo están completos los datos de exam_score para los NA de attendance_pct, la imputaremos mediante una regresión:

```
#Ajustar un modelo de regresión lineal  
model1 <- lm(attendance_pct ~ exam_score, data = data)  
summary(model1)
```

Call:

```
lm(formula = attendance_pct ~ exam_score, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-19.4057	-4.3153	0.0505	5.7861	16.9649

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	77.42832	1.45920	53.062	< 2e-16 ***
exam_score	0.15661	0.02421	6.468	4.61e-10 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.315 on 271 degrees of freedom

(27 observations deleted due to missingness)

Multiple R-squared: 0.1337, Adjusted R-squared: 0.1305

F-statistic: 41.84 on 1 and 271 DF, p-value: 4.614e-10

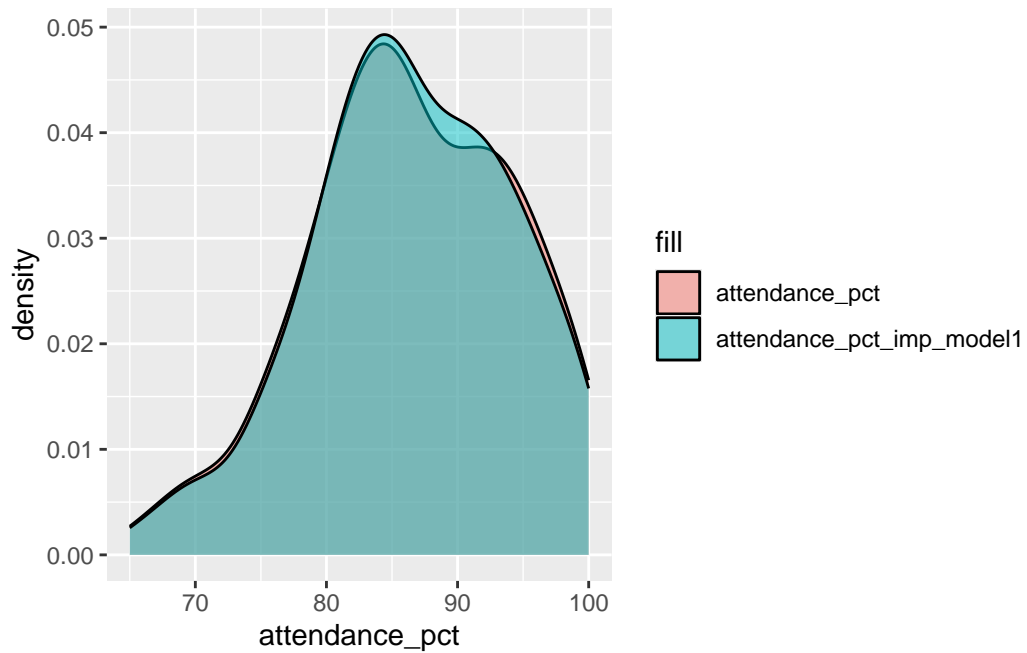
```
#Predecir los valores solo para las observaciones faltantes
predictions <- predict(model1,newdata = data [is.na(data$attendance_pct),])

##Crear una nueva variable imputada
data$attendance_pct_imp_model1 <- data$attendance_pct
data$attendance_pct_imp_model1[is.na(data$attendance_pct_imp_model1)]<- predictions

#Hacer un gráfico para comparar las observaciones
ggplot(data, aes(x = attendance_pct, fill = "attendance_pct")) +
  geom_density(alpha = 0.5) +

  geom_density(aes(x = attendance_pct_imp_model1,
                  fill = "attendance_pct_imp_model1"), alpha = 0.5)
```

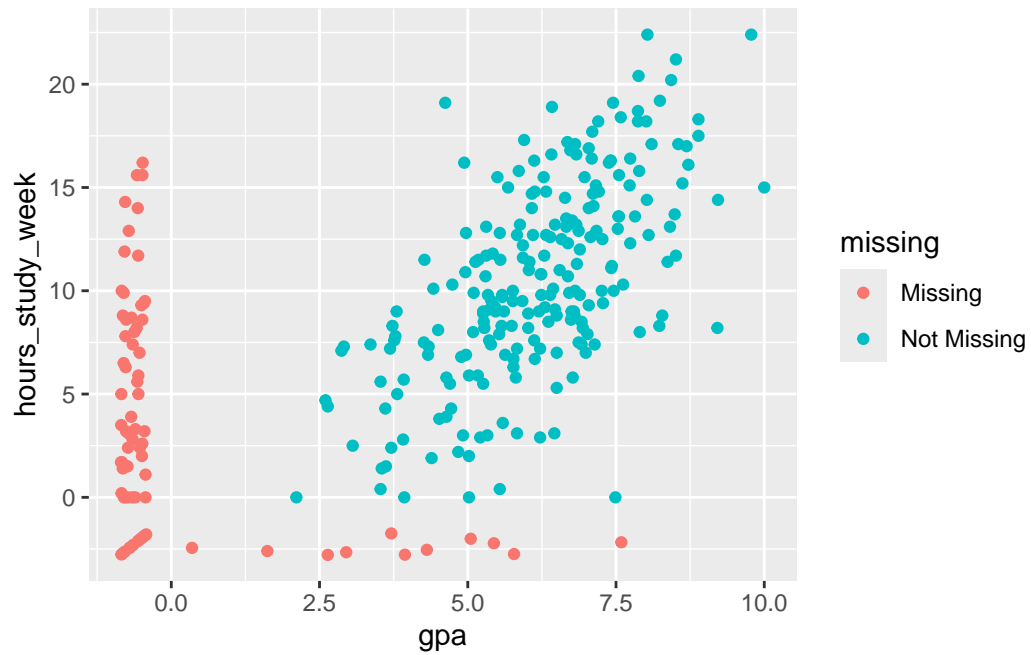
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_density()`).



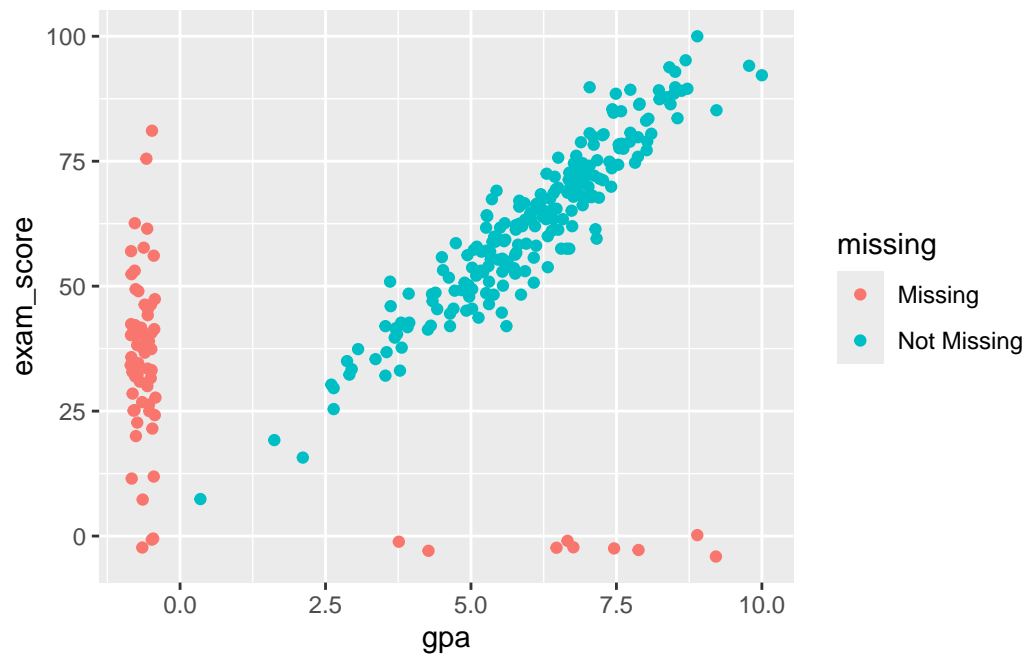
- Variable gpa

Los NA de **gpa** corresponden mayoritariamente a valores con datos inferiores de **hours_study_week** y con datos menores de **exam_score**. Por tanto, este patrón podría corresponder a un patrón **MNAR** ya que quizás corresponda a valores bajos de **gpa**.

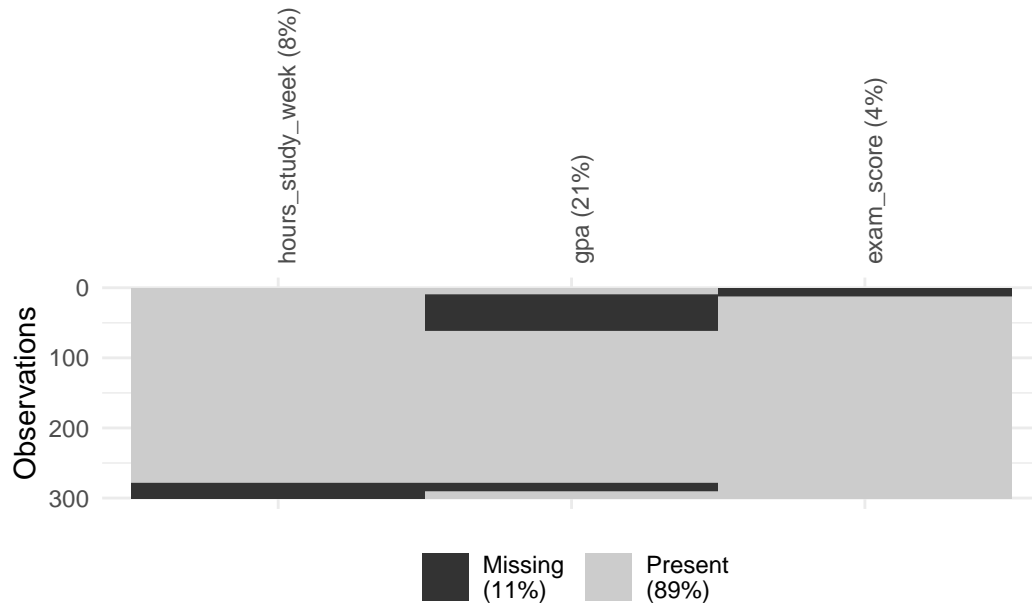
```
ggplot(data = data, aes (x = gpa ,  
                        y =hours_study_week )) + geom_miss_point()
```

```
ggplot(data = data, aes (x = gpa ,
                          y =exam_score )) + geom_miss_point()
```



```
vis_miss(select(data, hours_study_week, gpa, exam_score), cluster=TRUE) +  
  theme(axis.text.x = element_text(angle = 90))
```



En este caso, no tenemos ninguna de las dos variables asociadas como completas. Por tanto una opción es usar una de las ya imputadas. En este caso, usaremos exam_score que tiene el menor número de datos faltantes y una asociación muy clara

```
#Ajustar un modelo de regresión lineal  
model1 <- lm(gpa ~ exam_score_imp_model1, data = data)  
summary(model1)
```

Call:

```
lm(formula = gpa ~ exam_score_imp_model1, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.8735	-0.3485	-0.0369	0.2798	3.7261

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.445945	0.160162	2.784	0.0058 **

```
exam_score_imp_model1 0.089785    0.002476  36.264    <2e-16 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.6056 on 234 degrees of freedom
```

```
(64 observations deleted due to missingness)
```

```
Multiple R-squared:  0.8489,    Adjusted R-squared:  0.8483
```

```
F-statistic: 1315 on 1 and 234 DF,  p-value: < 2.2e-16
```

```
#Predecir los valores solo para las observaciones faltantes
predictions <- predict(model1,newdata = data [is.na(data$gpa),])
```

```
##Crear una nueva variable imputada
```

```
data$gpa_imp_model1 <- data$gpa
```

```
data$gpa_imp_model1[is.na(data$gpa_imp_model1)]<- predictions
```

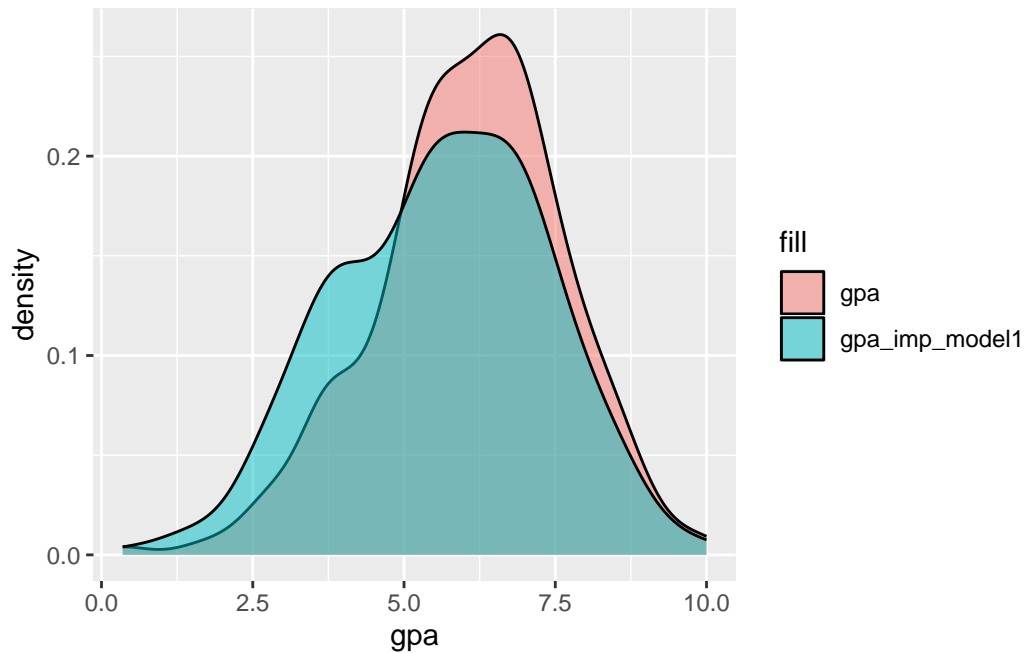
```
#Hacer un gráfico para comparar las observaciones
```

```
ggplot(data, aes(x = gpa, fill = "gpa")) +
```

```
  geom_density(alpha = 0.5) +
```

```
  geom_density(aes(x = gpa_imp_model1, fill = "gpa_imp_model1"),
               alpha = 0.5)
```

```
Warning: Removed 64 rows containing non-finite outside the scale range
(`stat_density()`).
```



En este caso vemos que la imputación no es tan buena como en las anteriores, se puede ver si añadiendo incertidumbre mejora

```
summary(model1) ##Cogemos el residual standard error
```

Call:

```
lm(formula = gpa ~ exam_score_imp_model1, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.8735	-0.3485	-0.0369	0.2798	3.7261

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.445945	0.160162	2.784	0.0058 **
exam_score_imp_model1	0.089785	0.002476	36.264	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6056 on 234 degrees of freedom

(64 observations deleted due to missingness)

Multiple R-squared: 0.8489, Adjusted R-squared: 0.8483

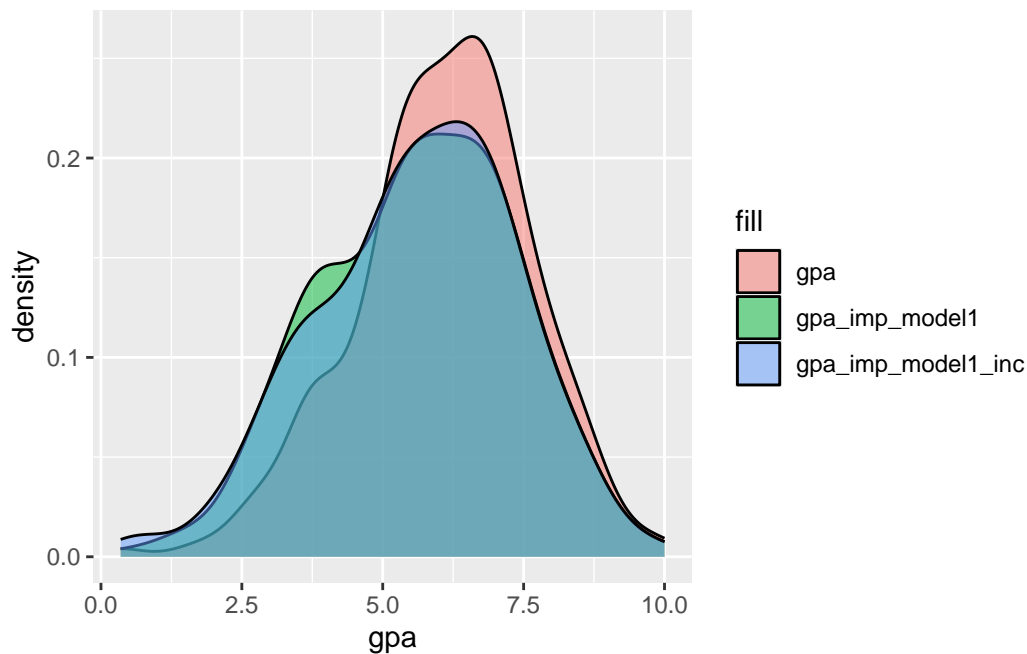
F-statistic: 1315 on 1 and 234 DF, p-value: < 2.2e-16

```
set.seed(3)
inc<-rnorm(sum(is.na(data$gpa)), 0, sd = 0.6056)

data$gpa_imp_model1_inc <- data$gpa
data$gpa_imp_model1_inc[is.na(data$gpa)]<- predictions + inc

#Hacer un gráfico para comparar las observaciones con la media y la regresión
ggplot(data, aes(x = gpa, fill = "gpa")) +
  geom_density(alpha = 0.5) +
  geom_density(aes(x = gpa_imp_model1, fill = "gpa_imp_model1"),
    alpha = 0.5) +
  geom_density(aes(x = gpa_imp_model1_inc, fill = "gpa_imp_model1_inc"),
    alpha = 0.5)
```

Warning: Removed 64 rows containing non-finite outside the scale range
(`stat_density()`).



tampoco parece muy acertado, las razones son que el % de faltantes es muy elevado y que estamos bajo el supuesto de MNAR, así que habrá que ver imputaciones más sofisticadas.