EDA

Ignacio Vellido

11/13/2020

Intro

Para este trabajo contamos con dos datasets distintos: **habermanMPG6** para aplicar Regresión y **haberman** para aplicar Clasificación.

Descripciones de los problemas

haberman

http://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival https://sci2s.ugr.es/keel/dataset.php?cod=62

Este dataset codifica el ratio de supervivencia de pacientes operados de cáncer de pecho en el Hospital Universitario de Chicago, en base a las siguientes características:

1. Age: Indica la edad del paciente en el momento de la operación.
2. Year: Los dos últimas cifras del año en el que se operó el paciente.
3. Positive: Número de nodos auxiliares positivos detectados. Esta variable hace referencia a los ganglios linfáticos que dan positivos como presentes de cáncer. A mayor número de nodos detectados, mayor es la gravedad del cáncer. Aunque normalmente la primera zona de propagación del cáncer son estos nodos, no es la única medida de la seriedad, pues este puede propagarse a otras zonas del cuerpo. En principio deberíamos suponer la posibilidad de que puede haber cosas de no supervivencia con bajo número de positivos, pero la bibliografía nos asegura que la probabilidad es baja.

Viendo que solo tenemos esta medida del cáncer en el dataset es posible que la operación que recibieron los pacientes sea algún tipo de cirugía de ganglios linfáticos, donde el cirujano intenta extraer los nodos afectados por el tumor. Por consiguiente, cuanto mayor es la cantidad de nodos detectados, más complicaciones pueden acarrearse de la operación. (poner referencias)

https://www.cancer.org/cancer/breast-cancer/treatment/surgery-for-breast-cancer/lymph-node-surgery-for-breast-cancer.html https://en.wikipedia.org/wiki/Lymph_node#:~:text=A%20lymph%20node%2C%20or%20lymph,include%20B%20and%20T%20cells.

El objetivo es poder clasificar, en base a los tres atributos, si los pacientes pueden sobrevir 5 años o más:

4. Survival: Sí/No indicando la supervivencia del paciente tras 5 años.

BUSCAR POSIBLES COMPLICACIONES

Contamos por tanto con un problema de clasificación binario en base a tres características, y con un número total de 306 instancias.

Análisis Estadístico de Datos

haberman

La descripción del problema nos da alguna información adicional sobre las variables:

1. Age: Variable numérica discreta, contamos con valores enteros en el rango [30,83].
2. Year: Variable numérica discreta, contamos con valores enteros en el rango [58,69].
3. Positive: Variable numérica discreta, contamos con valores enteros en el rango [0,52].
4. Survival: Variable binaria

Hipótesis de partida

- H.1: Habrá menor ratio de supervivencia cuanto mayor sea el número de nodos positivos encontrados: Por los razonamientos explicados en la introducción del problema.
- H.2: Habrá mayor ratio de supervivencia cuanto más joven sea el paciente.
- H.3: El rango de Year es pequeño. La influencia de esta variable creemos que podría darse solo si durante ese período se hubieran descubierto técnicas mejores de cirugía. Este razonamiento va orientado de cara a la población y no a la muestra. Puesto que contamos con datos de un solo hospital durante pocos años, es posible que el equipo de cirugía hubiera sido el mismo para la mayoría de pacientes.
- H.4: Podría haber relación entre la edad y el número de positivos, posiblemente indicando lo tardío que se descubre el cáncer.
- H.5: La bibliografía nos dice que el cáncer puede aparecer a diferentes edades con diferentes factores de riesgo (alcoholismo, herencia genética...). Podría ser que el número de variables con las que contamos sea insuficiente para la clasificación. (Hipótesis no correspondiente al EDA).

Cargamos los datos:

```
names <- c("Age", "Year", "Positive", "Survival")

haberman <- read_csv("Data/haberman/haberman.dat", comment = "@", col_names = names)
```

```
-- Column specification --------------------------------------------------------
cols(
  Age = col_double(),
  Year = col_double(),
  Positive = col_double(),
  Survival = col_character()
)
```

R por defecto nos carga las variables Age, Year y Positive como numéricas y Survival como carácter.

Vamos a transformar Survival a Factor

```
haberman$Survival <- haberman$Survival %>% factor(levels = c("negative", "positive"), labels = c("No",
```

El resto de variables las mantenemos como numéricas

Análisis univariable Los datos nos quedan por tanto de la siguiente manera:

```
head(haberman)
```

Age	Year	Positive	Survival
38	59	2	No
39	63	4	No
49	62	1	No
53	60	2	No
47	68	4	No
56	67	0	No

Hacemos summary para sacar datos de relevancia

```
summary(haberman)
```

```
      Age             Year          Positive      Survival
 Min.   :30.00   Min.   :58.00   Min.   : 0.000   No :225
 1st Qu.:44.00   1st Qu.:60.00   1st Qu.: 0.000   Yes: 81
 Median :52.00   Median :63.00   Median : 1.000
 Mean   :52.46   Mean   :62.85   Mean   : 4.026
 3rd Qu.:60.75   3rd Qu.:65.75   3rd Qu.: 4.000
 Max.   :83.00   Max.   :69.00   Max.   :52.000
```

En las distribuciones de los clasificadores nos fijaremos más adelante. Aquí hacemos notar que los valores de salida en nuestros datos están bastante desbalanceados, solo un 26.5% de los paciente sobrevivieron a los 5 años.

El dataset cuenta con valores repetidos

```
sum(duplicated(haberman))
```

```
[1] 17
```

Mostramos estas ocurrencias:

```
ind <- duplicated(haberman) | duplicated(haberman, fromLast = TRUE)
haberman[ind,] %>% arrange(Age)
```

Age	Year	Positive	Survival
37	63	0	No
37	63	0	No
38	60	0	No
38	60	0	No
41	65	0	No
41	65	0	No
43	64	0	Yes
43	64	0	Yes
44	61	0	No
44	61	0	No
48	58	11	Yes
48	58	11	Yes
50	61	0	No
50	61	0	No
54	62	0	No
54	62	0	No
55	58	1	No
55	58	1	No
56	60	0	No
56	60	0	No
57	64	0	No
57	64	0	No
61	59	0	No
61	59	0	No
61	59	0	No
62	66	0	No
62	66	0	No
63	63	0	No
63	63	0	No
65	64	0	No
65	64	0	No
67	66	0	No
67	66	0	No

Existen dos posibilidades para el origen de estos datos:

1. Errores en la introducción de los datos, entradas repetidas por error.
2. Sean entradas de pacientes distintos casualmente con las mismas características.

Como en este caso tenemos muy pocas variables (y un número moderado de entradas, 306), es probable que los pacientes coincidan en las características. Además, podemos ver que las entradas en la mayoría de los casos las variables solo están duplicadas (solo hay una entrada triplicada).

Por tanto proseguimos sin eliminar estas instancias duplicadas.

No contamos con missing values

```
sum(is.na(haberman))
```

```
[1] 0
```

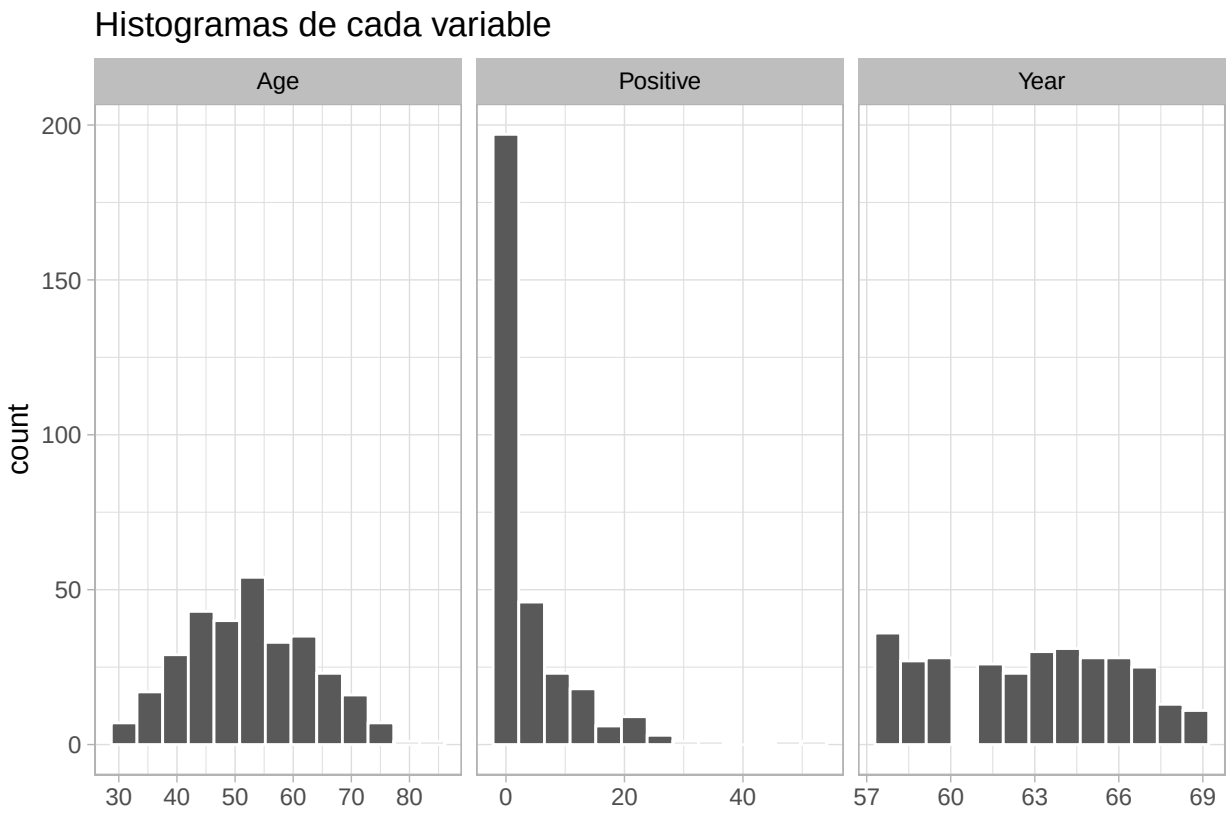
Separamos los datos de las etiquetas

```
labels <- haberman[4]
haberman <- haberman[-4]
```

```
names <- colnames(haberman)
```

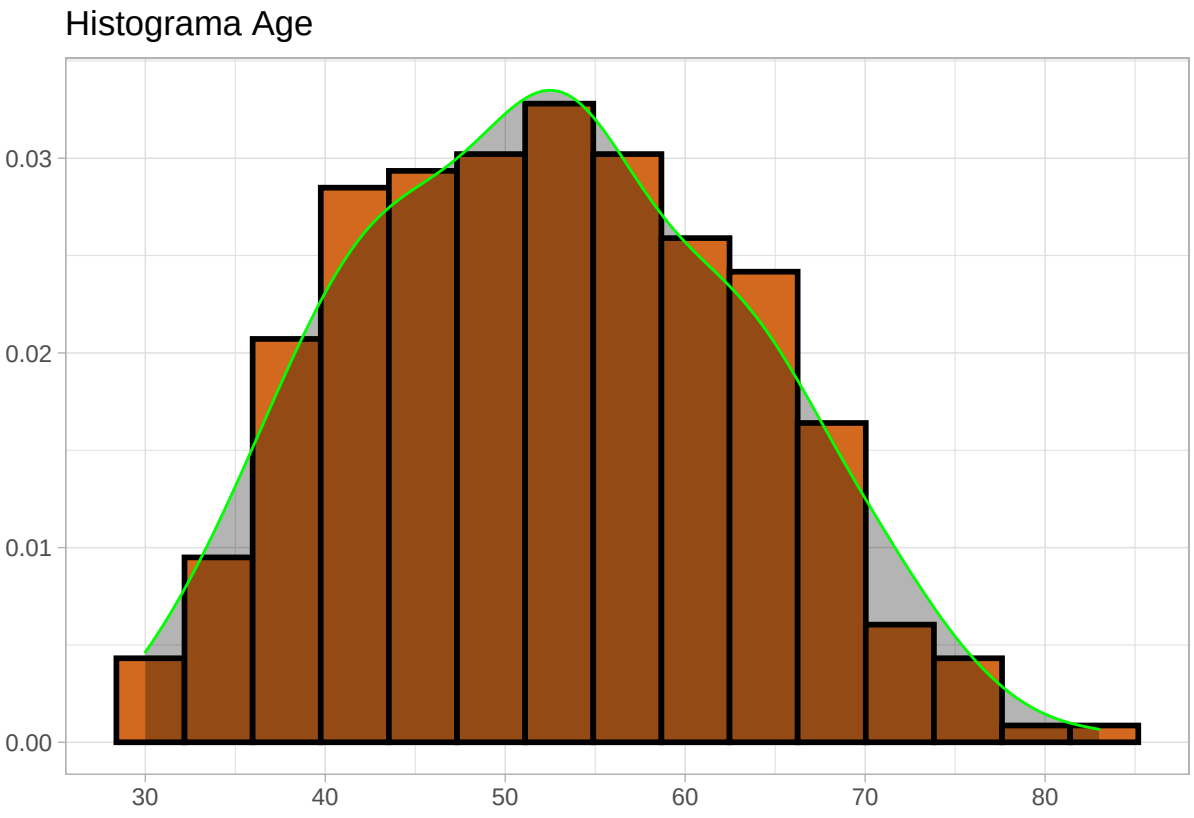
Vamos a sacar plots de cada variable para verlo mejor

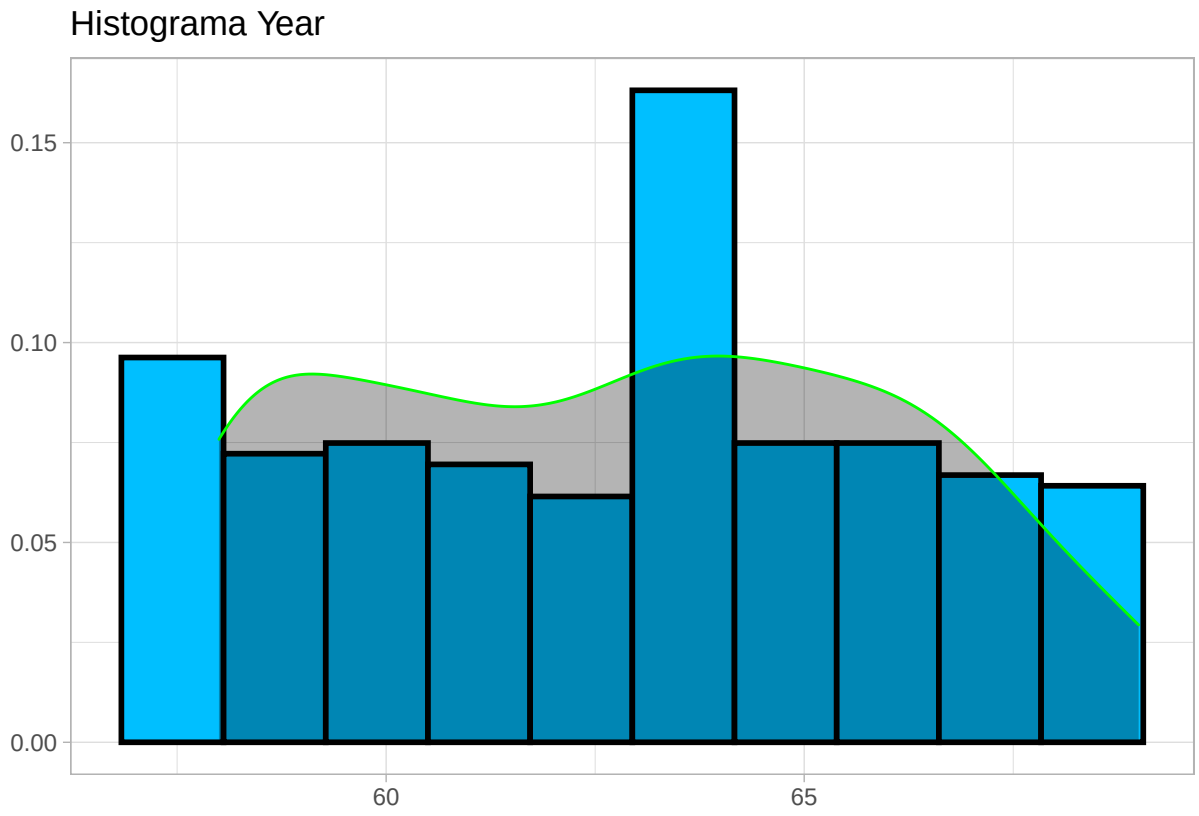
```
ggplot(gather(haberman), aes(value)) +
  geom_histogram(bins = 13, color="white") +
  facet_wrap(~key, scales = 'free_x') +
  theme_light() +
  theme(strip.background = element_rect(fill="grey", size=2))+
  theme(strip.text = element_text(colour = 'black')) +
  labs(title="Histogramas de cada variable", x = "")
```

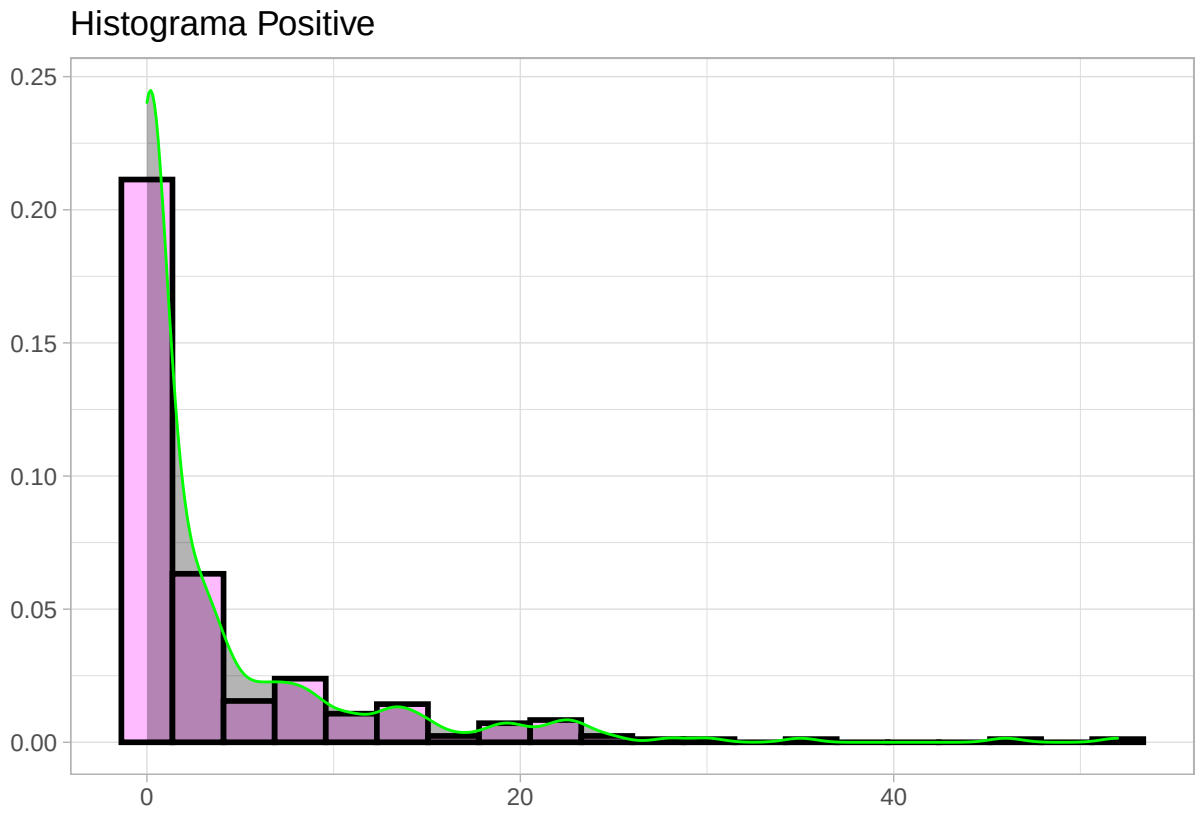


Una a una

```
colors <- c("chocolate", "deepskyblue1", "plum1")
bins <- c(15,10,20)
plt <- list(length = length(names))

for (i in 1:length(names)) {
  ggplot(haberman, aes_string(x=names[i])) +
    geom_histogram(aes(y=..density..), size=1, bins=bins[i], color="black", fill=colors[i]) +
    geom_density(alpha=.3, fill="black", color="green", size=.5) +
    labs(title="", x="", y="") +
    theme_light() -> plt[[i]]

  print(plt[[i]] + labs(title=sprintf("Histograma %s", names[i]), x=""))
}
```

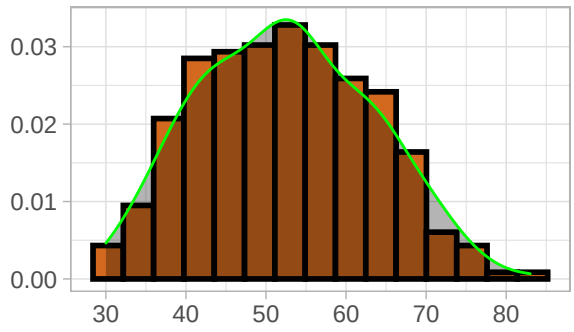

Histograma Age
0.03
0.02
0.01
0.00
30
40
50
60
70
80


Histograma Year
0.15
0.10
0.05
0.00
60
65

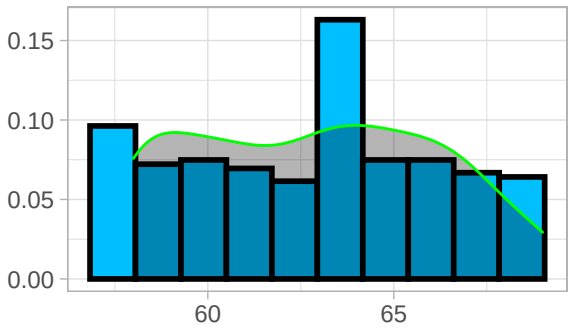



```
plot_grid(plotlist=plt, ncol=2, labels = names, label_size = 8)
```

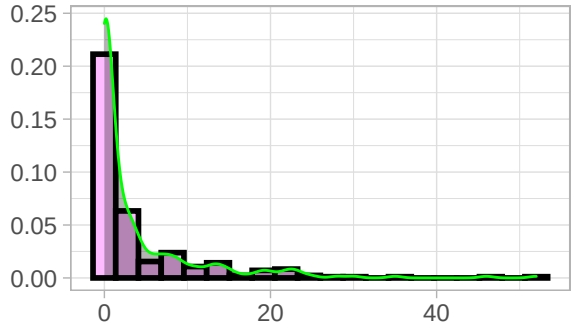
Age



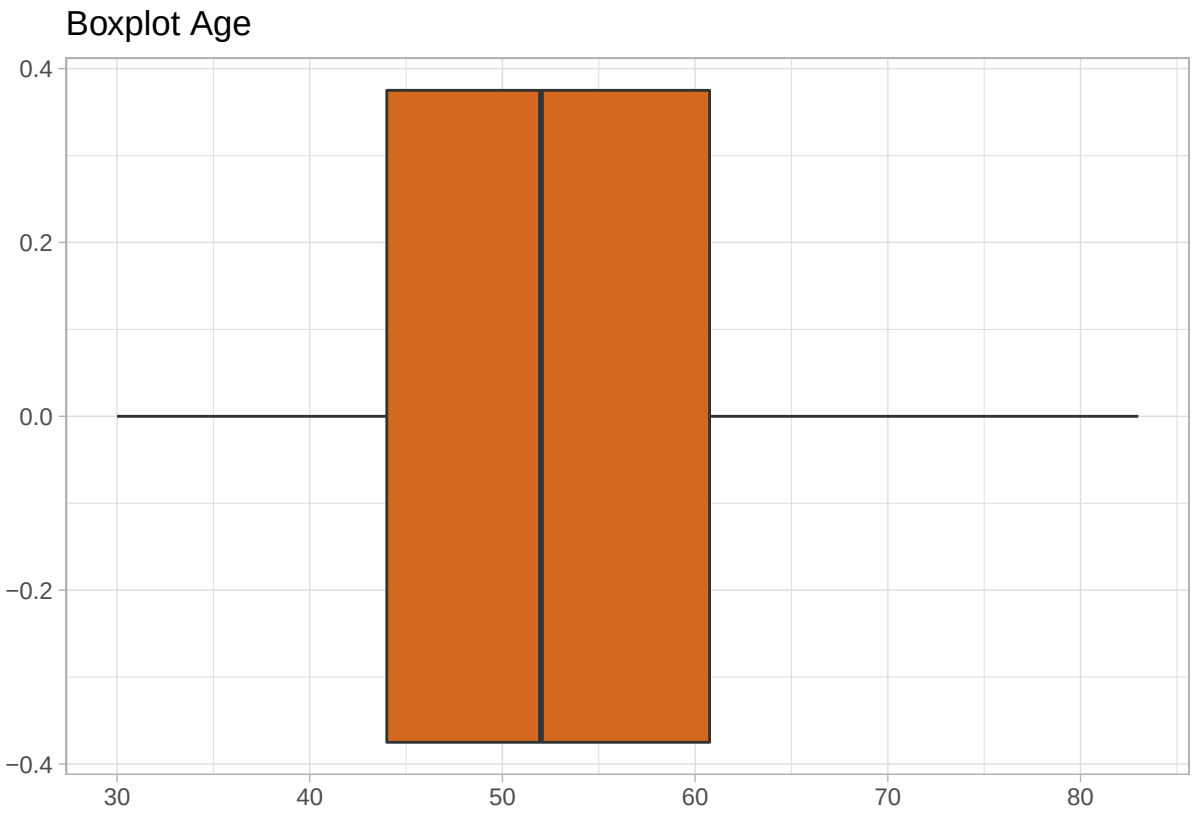
Year

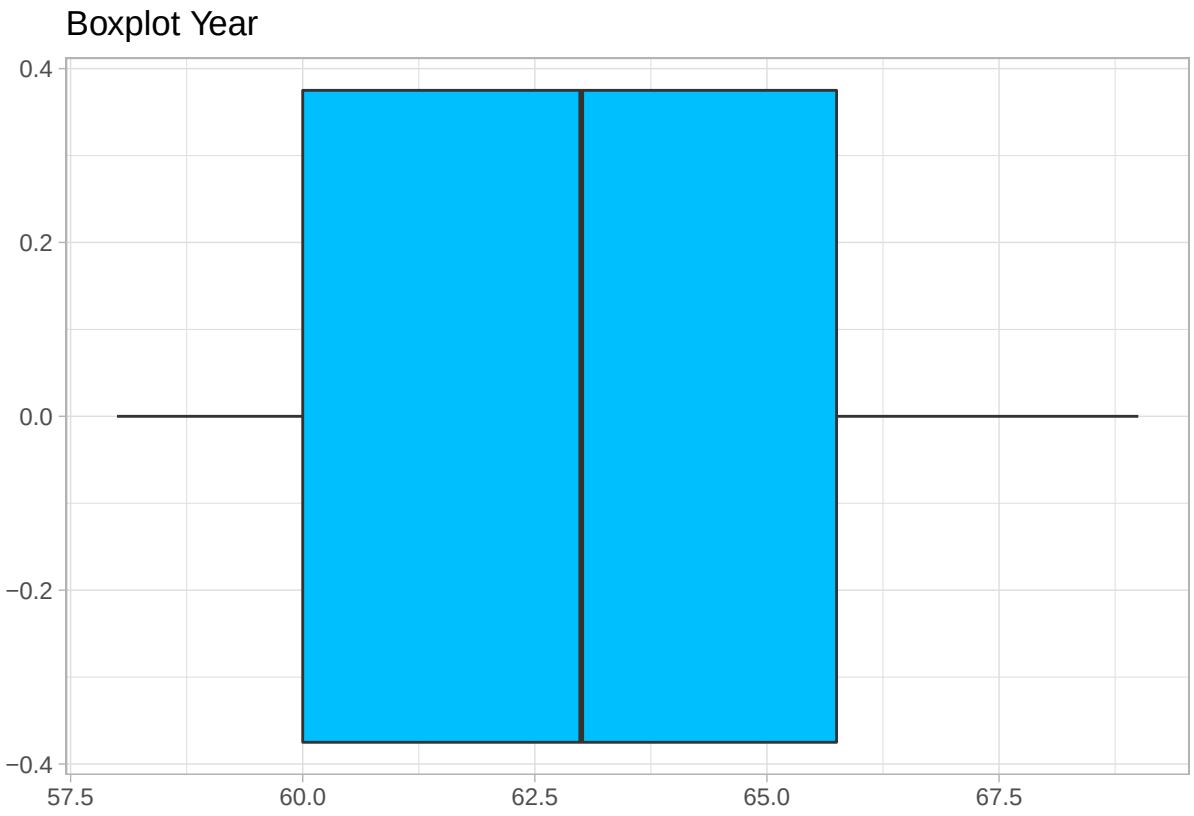


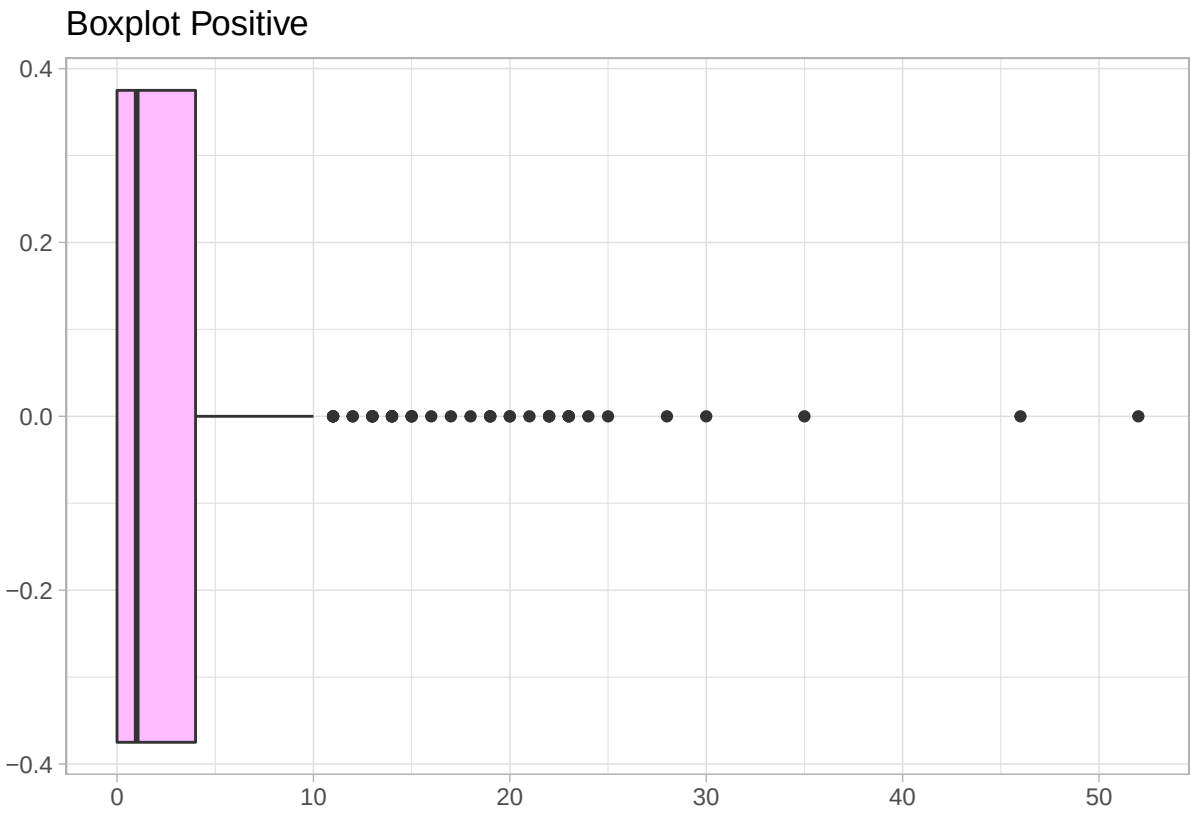
Positive




```
colors <- c("chocolate", "deepskyblue1", "plum1")
plt <- list(length = length(names))

for (i in 1:length(names)) {
  ggplot(haberman, aes_string(x=names[i])) +
    geom_boxplot(fill = colors[i]) +
    labs(title="", x="", y="") +
    theme_light() -> plt[[i]]

  print(plt[[i]] + labs(title=sprintf("Boxplot %s", names[i]), x=""))
}
```

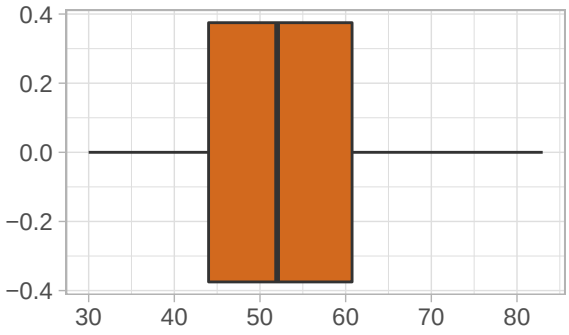

Boxplot Age
0.4
0.2
0.0
−0.2
−0.4
30
40
50
60
70
80


Boxplot Year
0.4
0.2
0.0
−0.2
−0.4
57.5
60.0
62.5
65.0
67.5

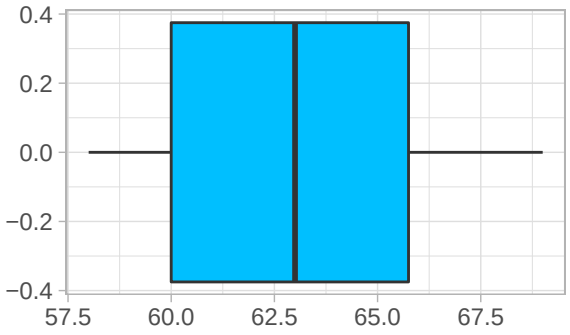



```
plot_grid(plotlist=plt, ncol=2, labels = names, label_size = 8)
```

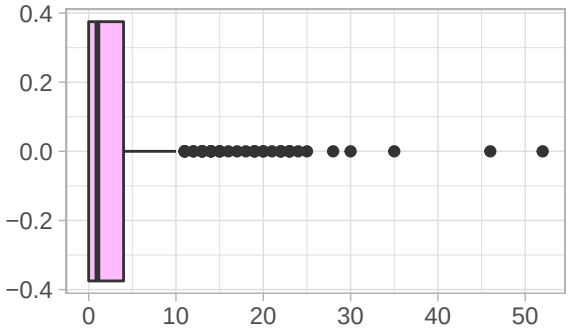
Age



Year

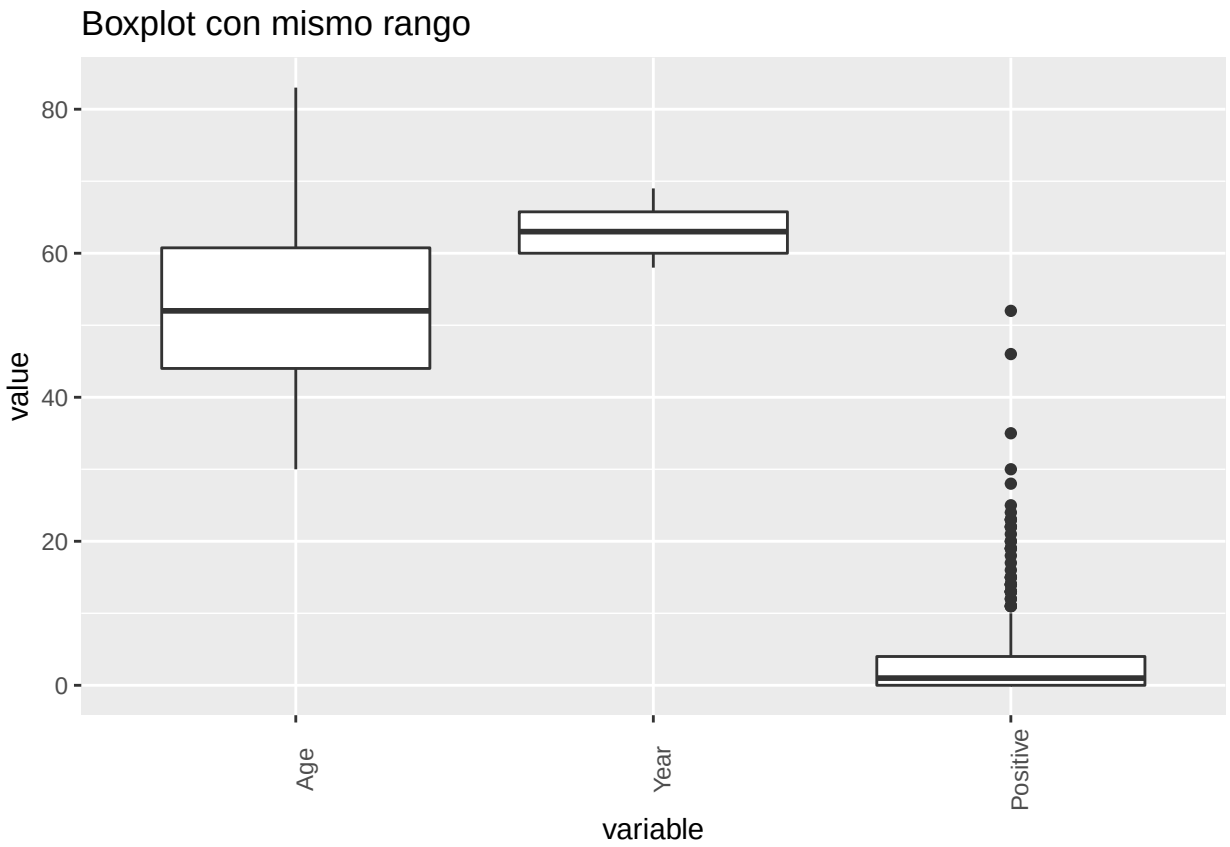


Positive




```
ggplot(melt(haberman), aes(x=variable, y=value)) +
  geom_boxplot() +
  labs(title="Boxplot con mismo rango") +
  theme(axis.text.x = element_text(angle = 90))
```

```
No id variables; using all as measure variables
```



Ya la descripción del problema nos lo decía, los rangos en los que se distribuyen los datos son muy diferentes dependiendo de la variable. Es necesario aplicar un proceso de estandarización para clasificación.

Missing values

Nos cuestionamos la ocurrencia de instancias con cero en el número de positivos. Podríamos pensar que se trata de una codificación de missing values si nos aseguramos que la operación consistan en eliminar estos nodos positivos.

Si revisamos la información que tenemos, estos nodos positivos se denominan auxiliares, y una mayor investigación del problema por internet nos asegura de que estos valores de cero no se corresponden a missing values.

Si hubiéramos descubierto que sí lo son, y tras ver que una gran parte de las instancias contienen este valor, habríamos tenido que buscar algún tipo de imputación para rellenar estos valores. Teniendo un número pequeño de valores reales, probablemente habríamos optado por KNN o interpolación lineal.

Podemos comparar los rangos intercuartiles si estandarizamos antes el dataset

```
scale(haberman) %>% apply(2, IQR)
```

```
     Age     Year Positive
1.550430 1.769555 0.556355
```

También podemos ver la distancia entre mínimos y máximos

```
scale(haberman) %>% apply(2, range) %>% apply(2, dist)
```

```
     Age     Year Positive
4.905839 3.385235 7.232616
```

Age Vemos que no contamos con valores de todos los años:

```
table(haberman$Age)
table(haberman$Age) %>% length

(83-30+1) == table(haberman$Age) %>% length
```

```
30 31 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
 3  2  2  7  2  2  6 10  6  3 10  9 11  7  9  7 11  7 10 12  6 14 11 13 10  7
57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 83
11  7  8  6  9  7  8  5 10  5  6  2  4  7  1  4  2  2  1  1  1  1  1
[1] 49
[1] FALSE
```

Year Aunque no se vea bien en las gráficas, contamos con valores de todos los años, con mayor cantidad en los iniciales:

```
table(haberman$Year)
```

```
58 59 60 61 62 63 64 65 66 67 68 69
36 27 28 26 23 30 31 28 28 25 13 11
```

Positive La variable Positive parece llevar una distribución exponencial, y problablemente por ello aparezcan tantos posibles outliers.

Análisis sobre las distribuciones

Ninguna variable parece seguir una distribución semejante a una distribución normal, lo comprobamos con un test estadístico (Shapiro-Wilk test):

```
normality(haberman) %>% filter(p_value < 0.05)
```

```
Warning: `cols` is now required when using unnest().
Please use `cols = c(statistic)`
```

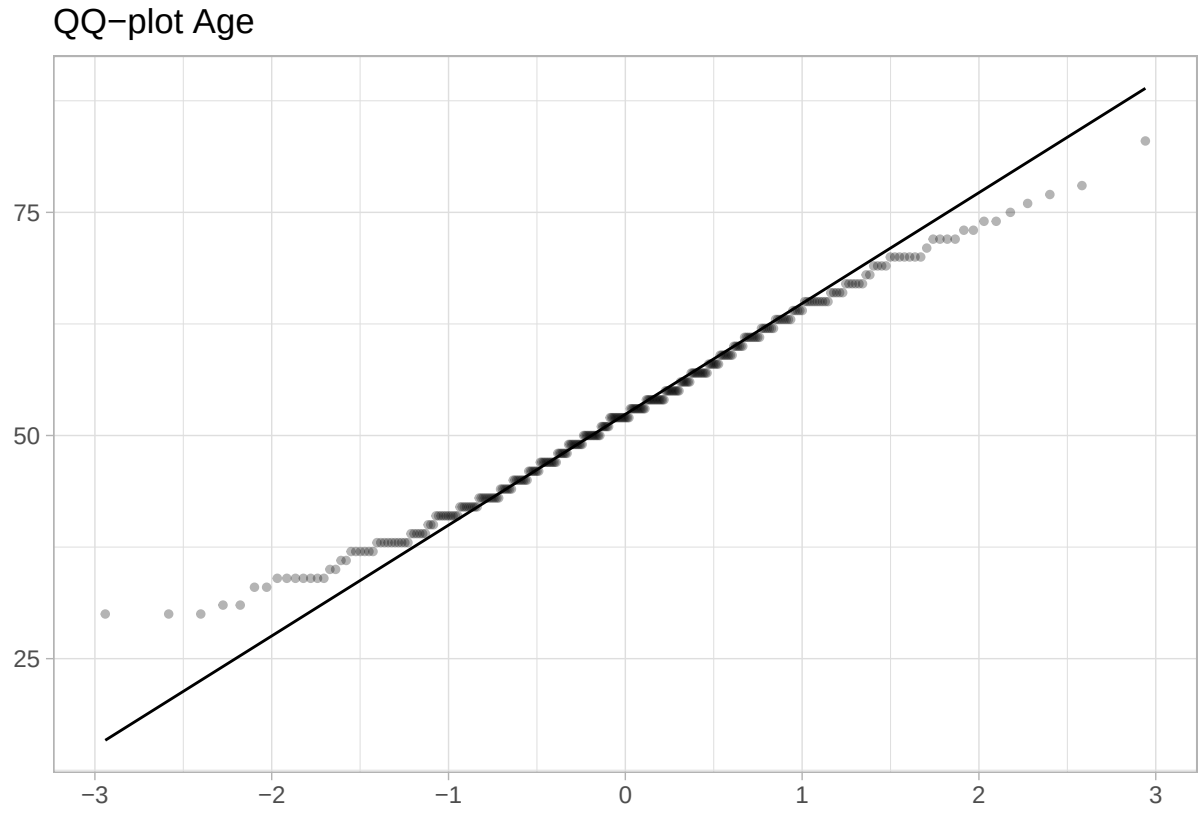
vars	statistic	p_value	sample
Age	0.9894580	0.0260466	306
Year	0.9467912	0.0000000	306
Positive	0.6153079	0.0000000	306

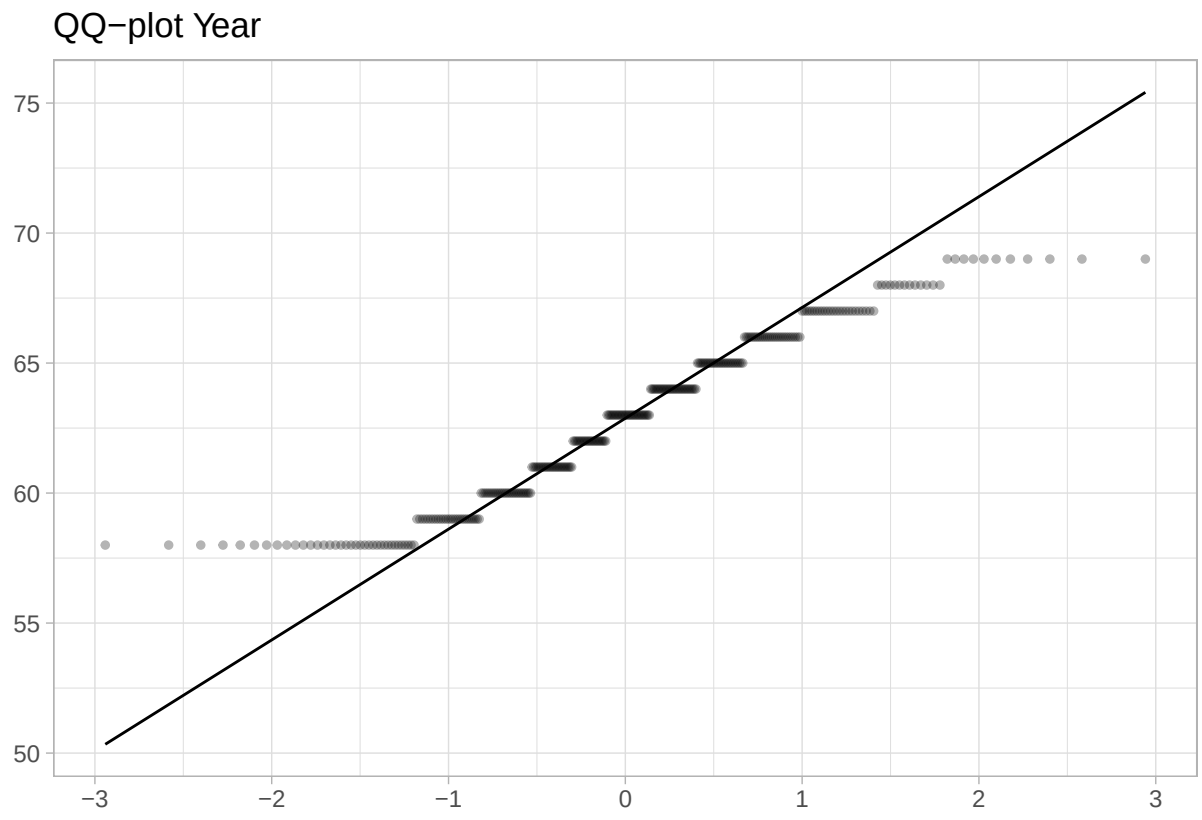
El test de Shapiro nos dice claramente que ninguna variable sigue una distribución normal.

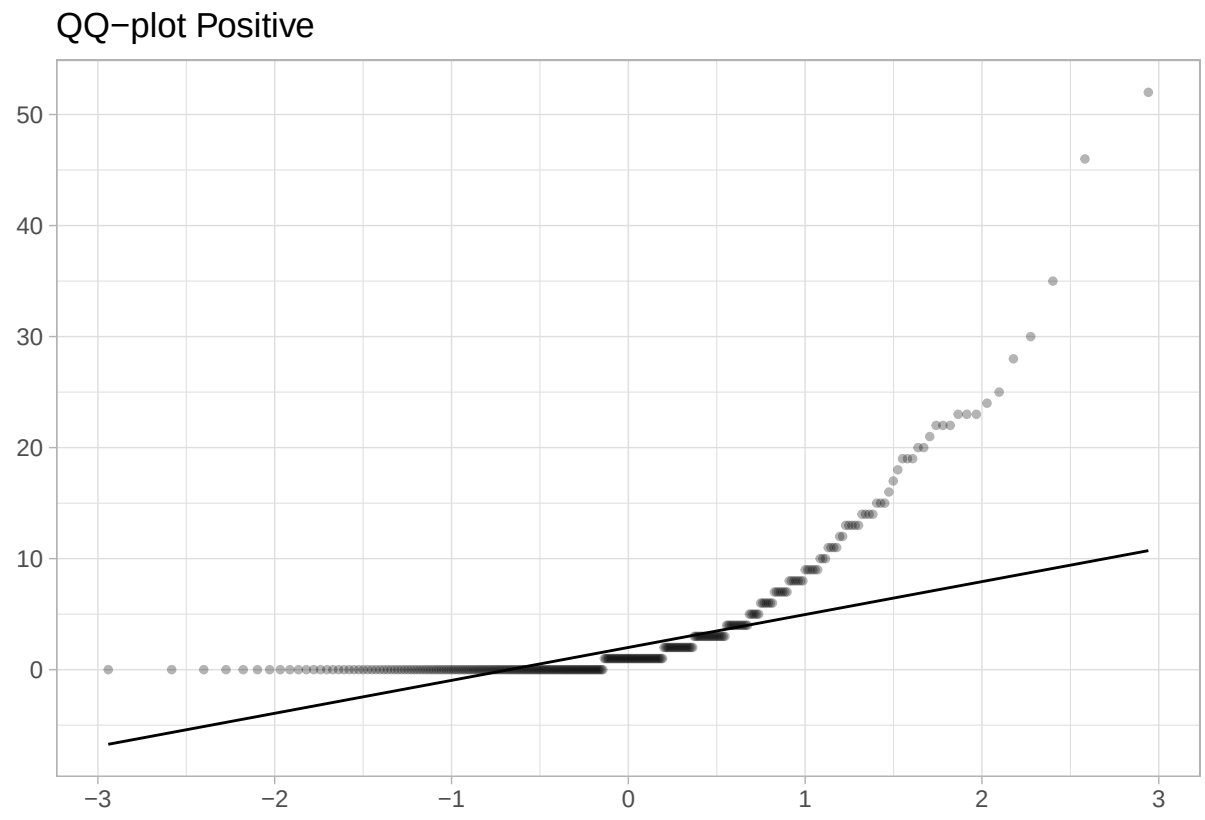
Lo mostramos gráficamente con plots Q-Q:

```
plt <- list(length = length(names))

x<-rnorm(100, mean=0, sd=1)

for (i in 1:length(names)) {
```

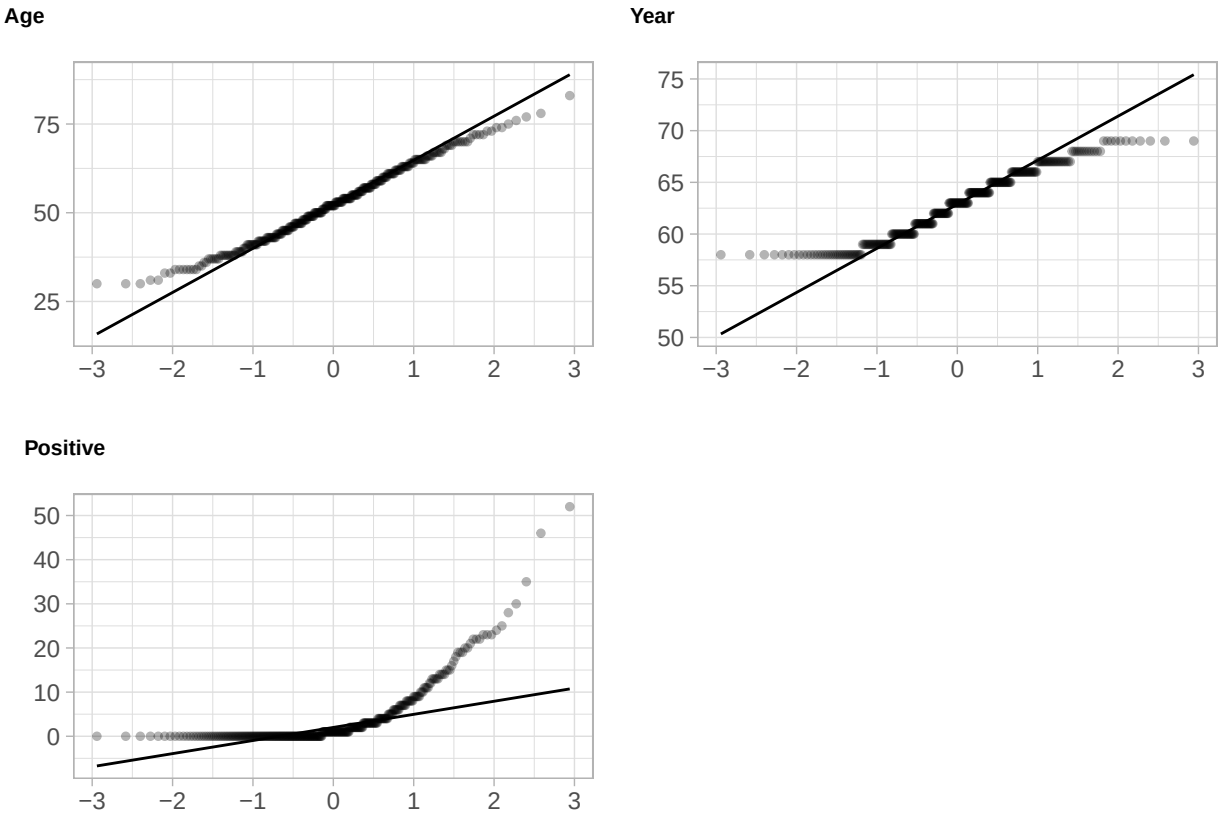
```
  ggplot(haberman, aes_string(sample=names[i])) +
    stat_qq(alpha=.3, fill=colors[i], size=1) +
    stat_qq_line() +
    labs(title="", x="", y="") +
    theme_light() -> plt[[i]]

  print(plt[[i]] + labs(title=sprintf("QQ-plot %s", names[i]), x=""))
}
```




QQ−plot Year
75
70
65
60
55
50
−3
−2
−1
0
1
2
3




```
plot_grid(plotlist=plt, ncol=2, labels = names, label_size = 8)
```



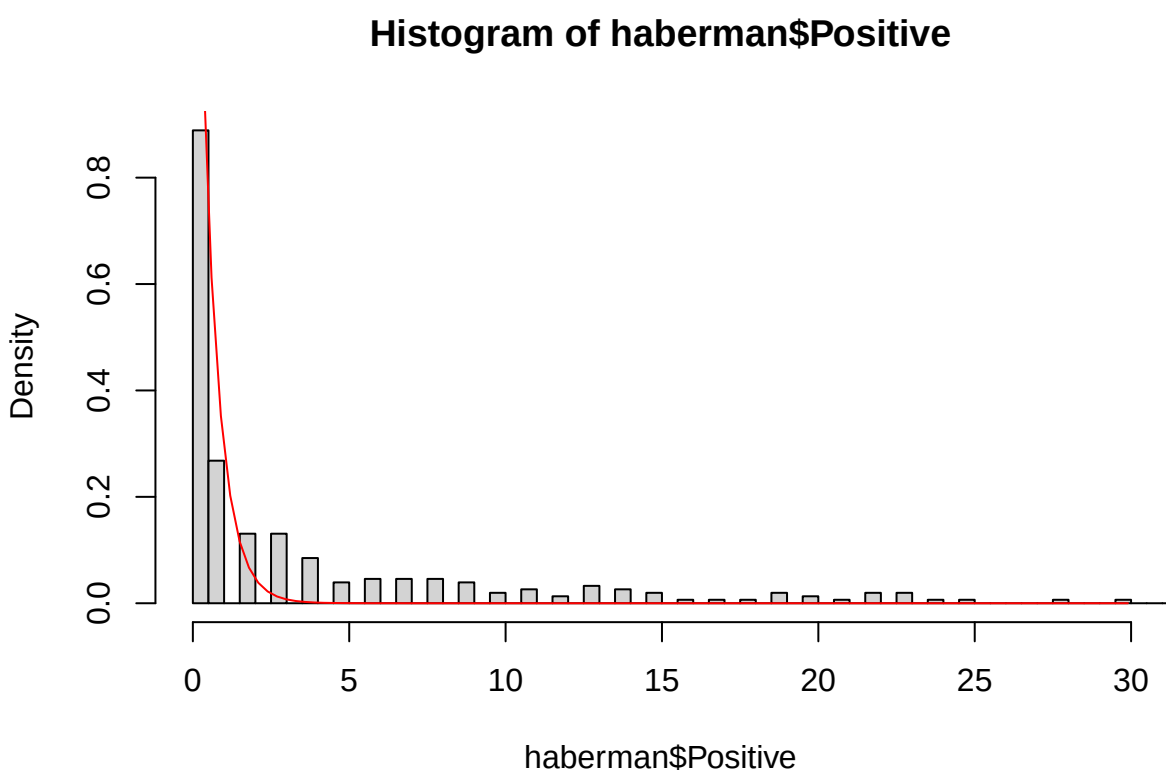
Se ve que las distribuciones no siguen los cuartiles normales, mayormente en las colas.

La variable Positive parece seguir una distribución exponencial, hacemos un test de Kolmogorov-Smirnov para corroborarlo:

```
# data generation
ex <- rexp(10000, rate = 1.85) # generate some exponential distribution
control <- abs(rnorm(10000)) # generate some other distribution

# estimate the parameters
fit1 <- fitdistr(ex, "exponential")
fit2 <- fitdistr(haberman$Positive %>% unique(), "exponential")

# goodness of fit test
ks.test(ex, "pexp", fit1$estimate) # p-value > 0.05 -> distribution not refused
ks.test(haberman$Positive, "pexp", fit2$estimate) #  significant p-value -> distribution refused
```

```
Warning in ks.test(haberman$Positive, "pexp", fit2$estimate): ties should not be
present for the Kolmogorov-Smirnov test
```

```
# plot a graph
hist(haberman$Positive, freq = FALSE, breaks = 100, xlim = c(0, quantile(haberman$Positive, 0.99)))
curve(dexp(x, rate = fit1$estimate), from = 0, col = "red", add = TRUE)
```




```
# control
```

```
    One-sample Kolmogorov-Smirnov test

data:  ex
D = 0.0053904, p-value = 0.9334
alternative hypothesis: two-sided
```

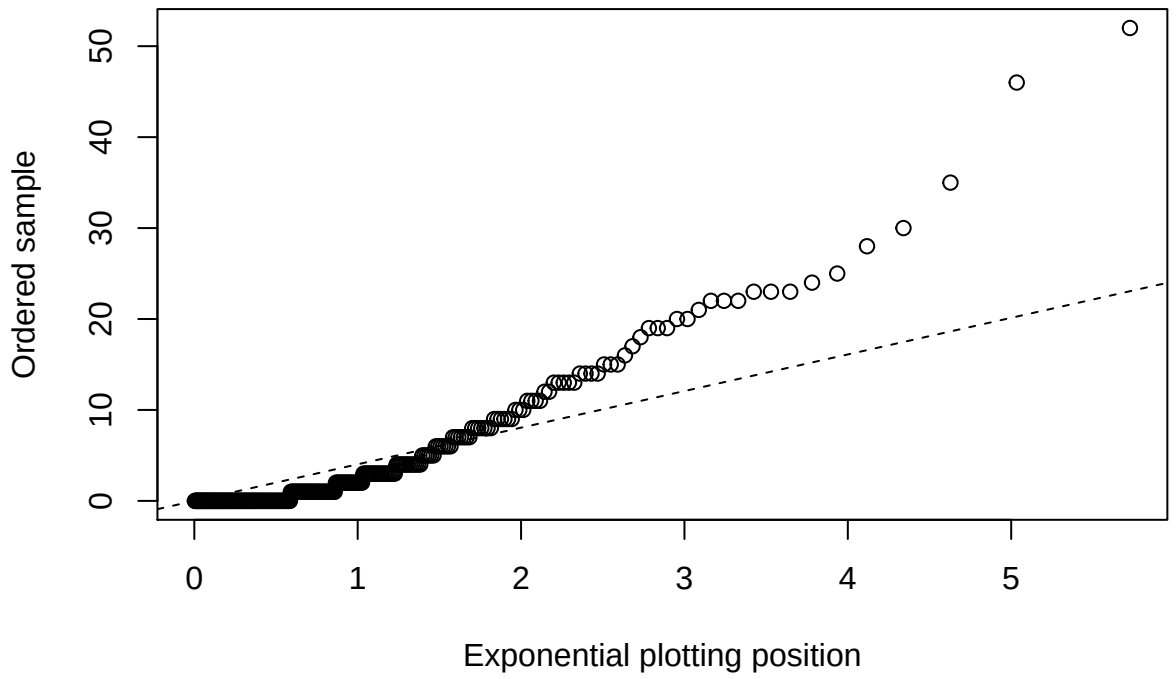
```
    One-sample Kolmogorov-Smirnov test

data:  haberman$Positive
D = 0.54423, p-value < 2.2e-16
alternative hypothesis: two-sided
```

Al ser el p-valor <0.05 el test nos lo rechaza. El plot también nos lo muestra más claramente, la forma de la cola de la distribución probablemente sea la causante de que no siga ese tipo de distribución

Podemos hacer un gráfico QQ con los cuartiles de una distribución exponencial

```
# From https://stats.stackexchange.com/questions/76994/how-do-i-check-if-my-data-fits-an-exponential-di
qqexp <-  function(y, line=FALSE, ...) {
    y <- y[!is.na(y)]
    n <- length(y)
    x <- qexp(c(1:n)/(n+1))
    m <- mean(y)
    if (any(range(y)<0)) stop("Data contains negative values")
```

```
    ylim <- c(0,max(y))
    qqplot(x, y, xlab="Exponential plotting position",ylim=ylim,ylab="Ordered sample", ...)
    if (line) abline(0,m,lty=2)
    invisible()
}

qqexp(haberman$Positive, line=TRUE)
```



Solo vemos skewness en la variable Positive, lo comprobamos:

```
skewCols <- find_skewness(haberman)
colnames(haberman)[skewCols]
```

```
[1] "Positive"
```

Calculamos el grado que tienen:

```
cat("Positive: ")
skewness(haberman$Positive)
cat("Year: ")
skewness(haberman$Year)
cat("Age: ")
skewness(haberman$Age)
```

```
Positive: [1] 2.969176
Year: [1] 0.07836828
Age: [1] 0.1457859
```

Positive tiene skewness positiva en un alto grado, las demás tienen tan poco como para poder considerarlo.

Transformaciones

Dejamos que el paquete caret nos proponga metodos de preprocesado

```
preProcess(haberman)
```

```
Created from 306 samples and 3 variables

Pre-processing:
  - centered (3)
  - ignored (0)
  - scaled (3)
```

Nos sugiere una estandarización a media cero y desviación típica 1. Para un problema de clasificación esto es totalmente necesario puesto que no queremos que los diferentes rangos de las variables hagan que haya información de más peso que otra.

Según el método utilizado la necesitad de normalidad puede ser o no necesaria. (CORROBORAR) Aplicar métodos de reducción de skewness en Positive no parece interesante puesto que está demasiado ladeado.

```
haberman_transform <- preProcess(haberman, method=c("scale", "center"))
# haberman_transform <- preProcess(haberman, method=c("YeoJohnson","scale", "center"))
haberman_norm <- predict(haberman_transform, haberman)
```

```
# colors <- c("chocolate", "deepskyblue1", "plum1")
# bins <- c(15,10,20)
# plt <- list(length = length(names))
#
# for (i in 1:length(names)) {
#   ggplot(haberman_norm, aes_string(x=names[i])) +
#     geom_histogram(aes(y=..density..), size=1, bins=bins[i], color="black", fill=colors[i]) +
#     geom_density(alpha=.3, fill="black", color="green", size=.5) +
#     labs(title="", x="", y="") +
#     theme_light() -> plt[[i]]
#
#   print(plt[[i]] + labs(title=sprintf("Histograma %s", names[i]), x=""))
# }
#
# plot_grid(plotlist=plt, ncol=2, labels = names, label_size = 8)
```

Outliers

La única variable en la que podríamos considerar outliers es Positive. Tanto para la edad como para los años no tiene sentido, además de que hemos visto en los boxplots que en ellas todos los valores caen en el 95% de la distribución.

A la hora de considerar los outliers en Positive, tal y como habíamos mencionado en la descripción del problema, un alto número de nodos detectados complica la operación y el pronóstico para el paciente.

Podemos mostrar para aquellos valores outliers la cantidad de sobrevivientes:

```
haberman %>%
  bind_cols(labels) %>%
  filter(Positive>10) %>%
```

```
  summarise(Survival) %>%
  table()
```

.

```
 No Yes
 17  23
```

Vemos que realmente está equilibrado. Aún así, si hubiera una tendencia negativa por un alto número en Positive querríamos que nuestro clasificador fuera capaz de aprenderlo, por lo que proseguimos sin eliminar outliers.

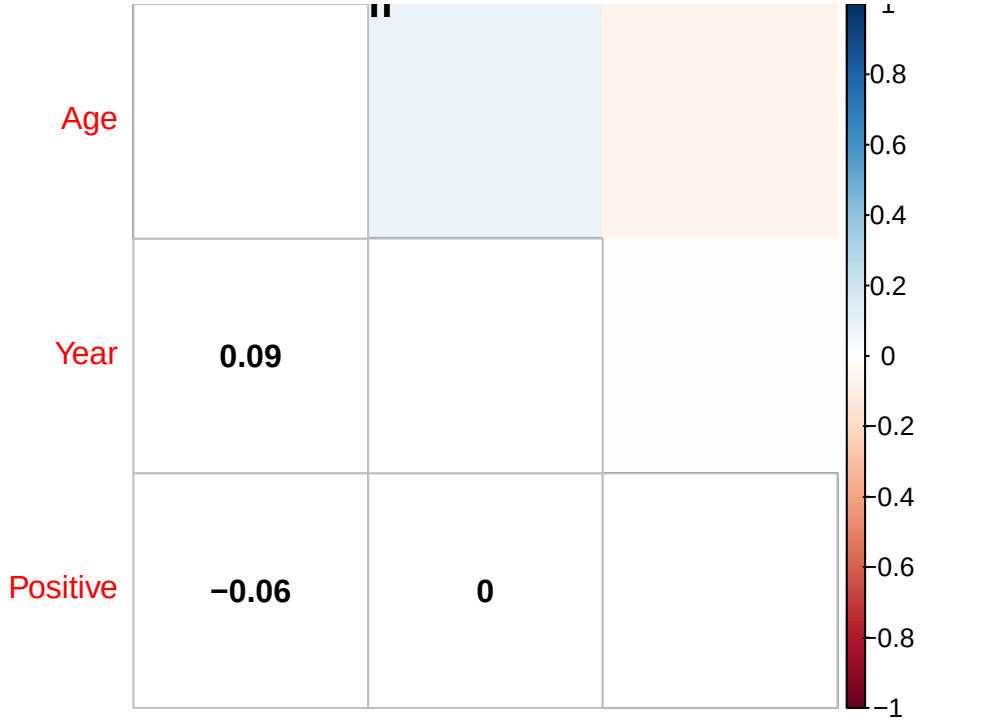
Análisis de correlación

Como este es un problema de clasificación, necesitamos eliminar aquellas variables correladas para que la información se aporte de manera equitativa. Las gráficas no nos han dado ninguna señal de una posible correlación, pero debemos asegurarnos de forma estadística.

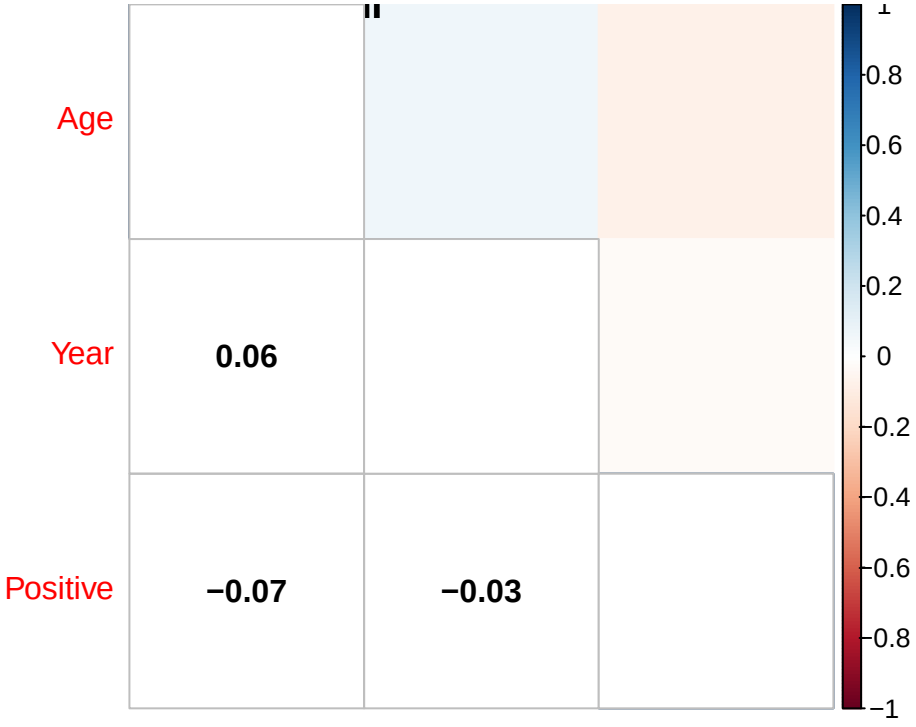
Tenemos que tener en cuenta que las variables no siguen distribuciones normales. Aunque el coeficiente de Pearson no asume normalidad (si asume varianza y covarianza finitas), podemos usar el coeficiente de Kendall para los cálculos. Independientemente del método usado vamos a obtener las mismas correlaciones en este dataset, solo varía la fuerza con la que se dan.

Corrplot

```
corrplot.mixed(cor(haberman), tl.pos="lt", upper="color", lower.col="black", title="Pearson")
```

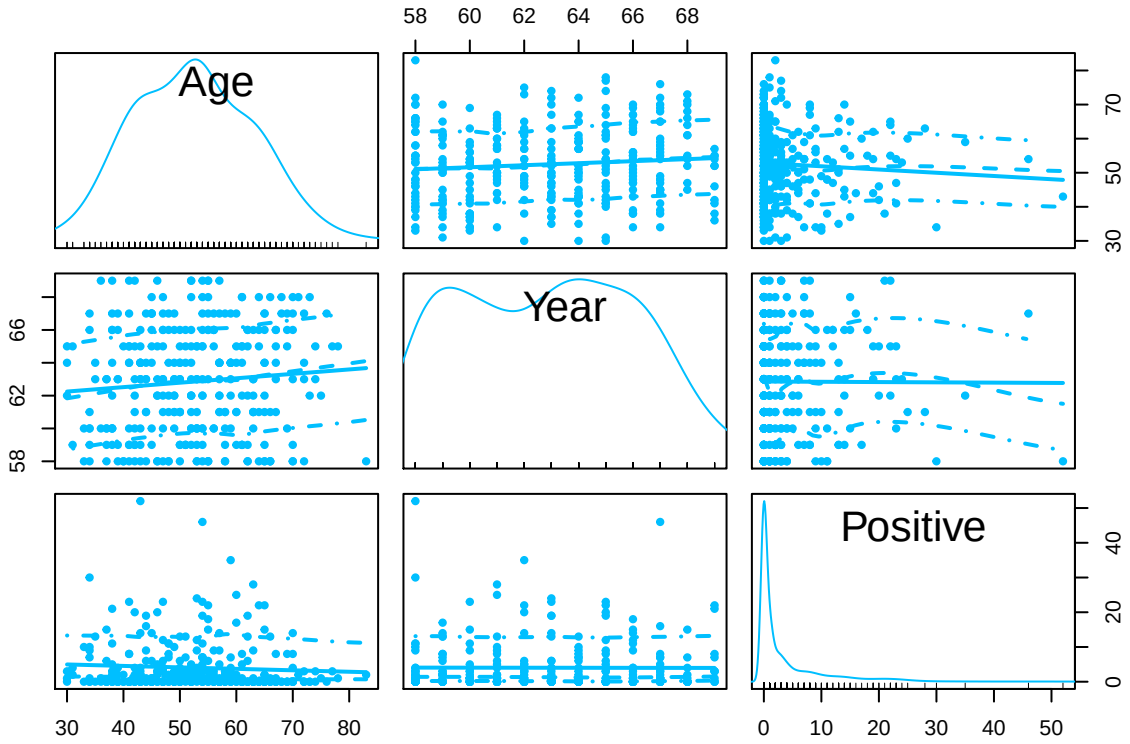


```
corrplot.mixed(cor(haberman, method="kendall"), tl.pos="lt", upper="color", lower.col="black", title="K
```



Nos muestra que no existe correlación alguna entre las variables.

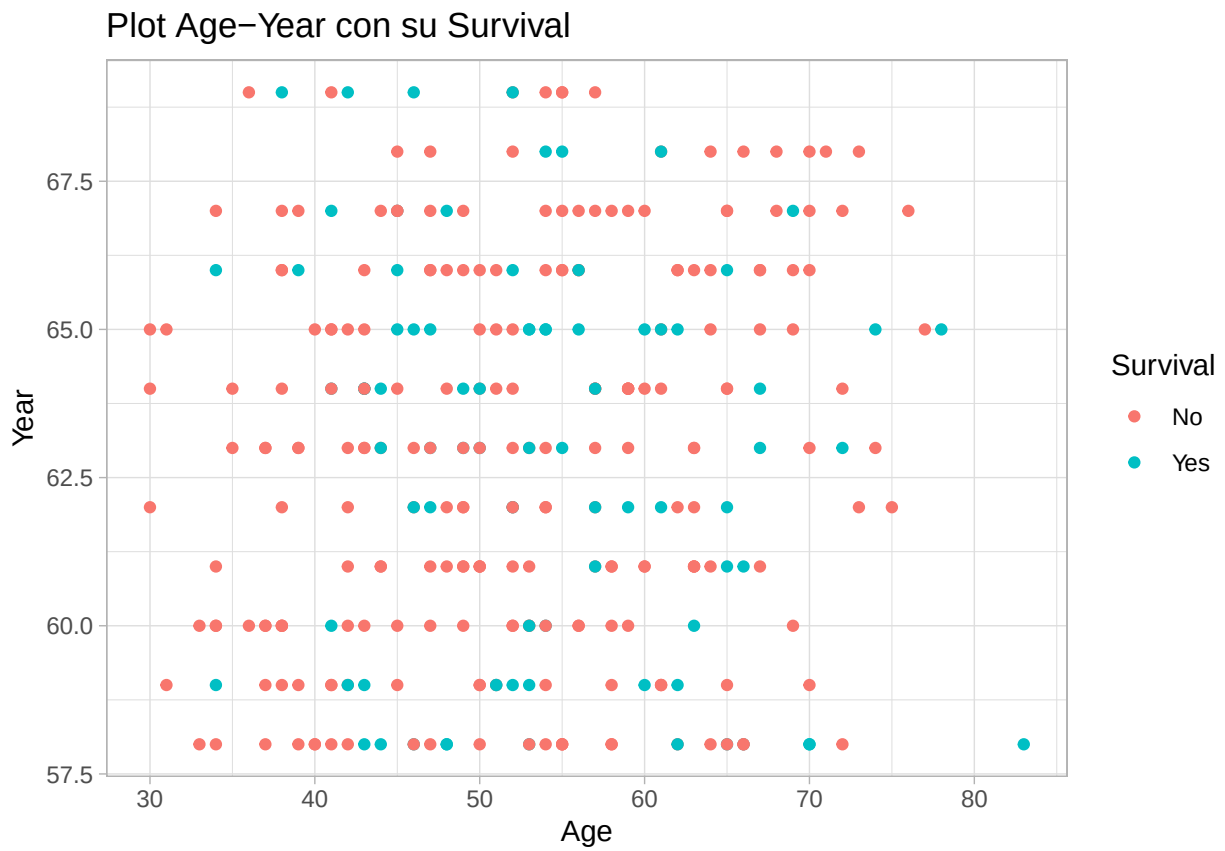
```
scatterplotMatrix(haberman, pch=20, col="deepskyblue")
```



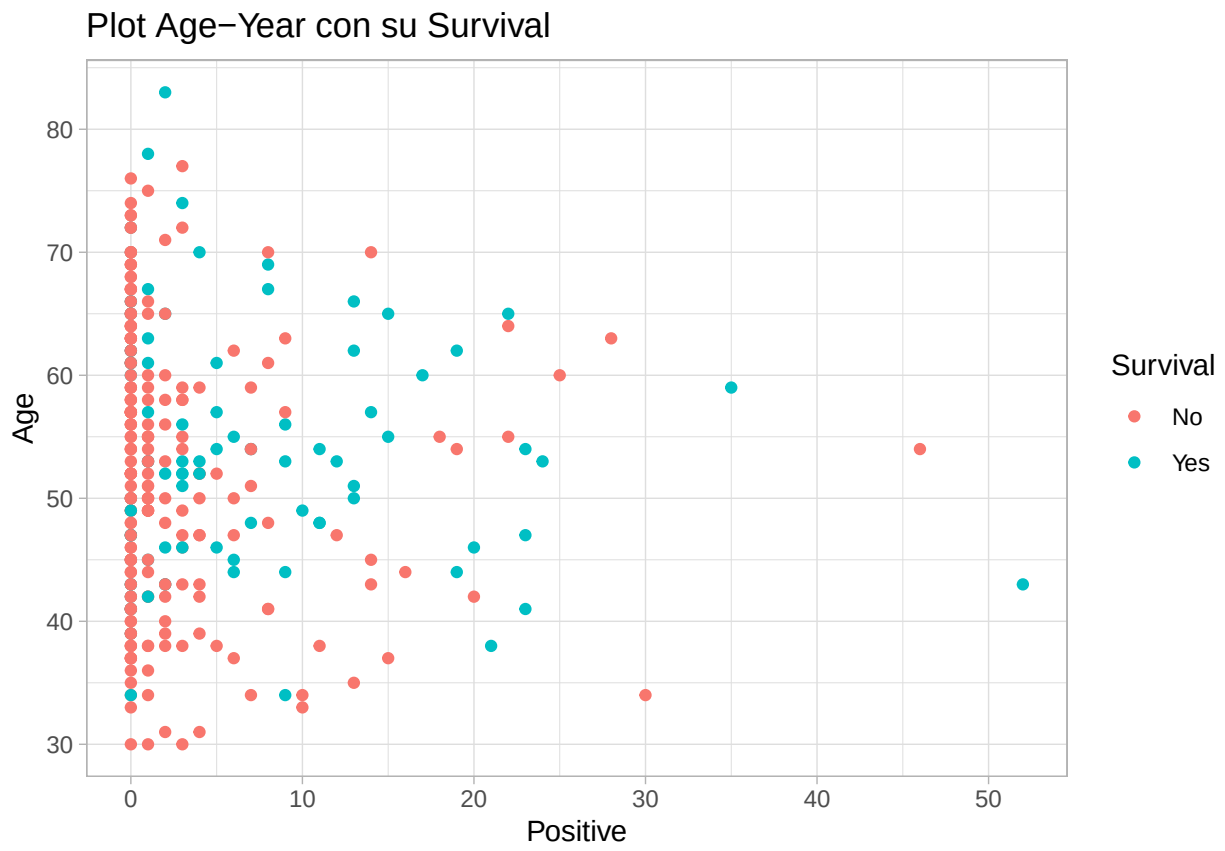
El scatterplot anterior nos muestra mejor la forma de las relaciones entre variables, vemos que no existe tendencia alguna.

Miramos la distribución de las variables con su clasificación

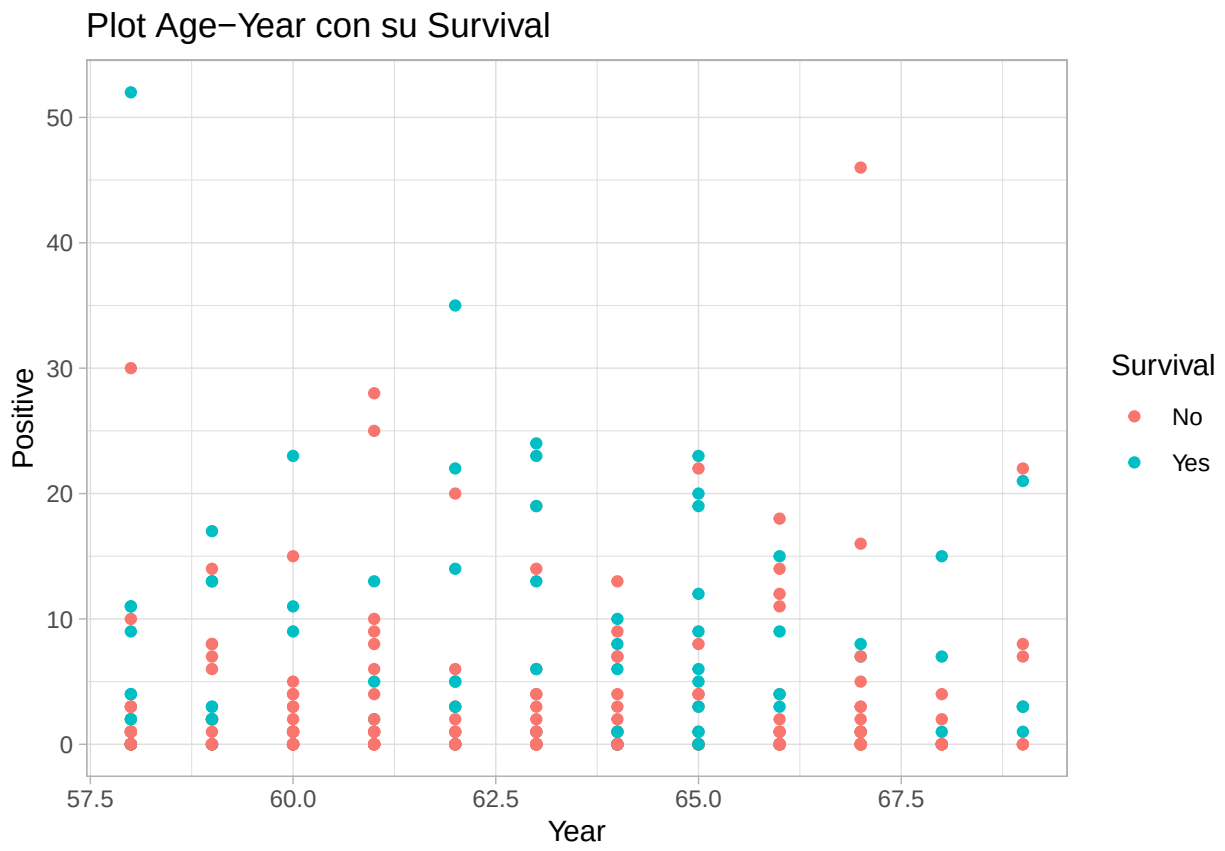
```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Age, y=Year, color=Survival)) +
    geom_point() +
    labs(title="Plot Age-Year con su Survival") +
    theme_light()
```




```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(y=Age, x=Positive, color=Survival)) +
    geom_point() +
    labs(title="Plot Age-Year con su Survival") +
    theme_light()
```




```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Year, y=Positive, color=Survival)) +
    geom_point() +
    labs(title="Plot Age-Year con su Survival") +
    theme_light()
```



No se aprecia ninguna relación visual que nos ayude a clasificar el Survival.

Tratamiento de variables y ordenaciones

Volvemos a mostrar la cabecera de los datos:

```
head(haberman)
```

Age	Year	Positive
38	59	2
39	63	4
49	62	1
53	60	2
47	68	4
56	67	0

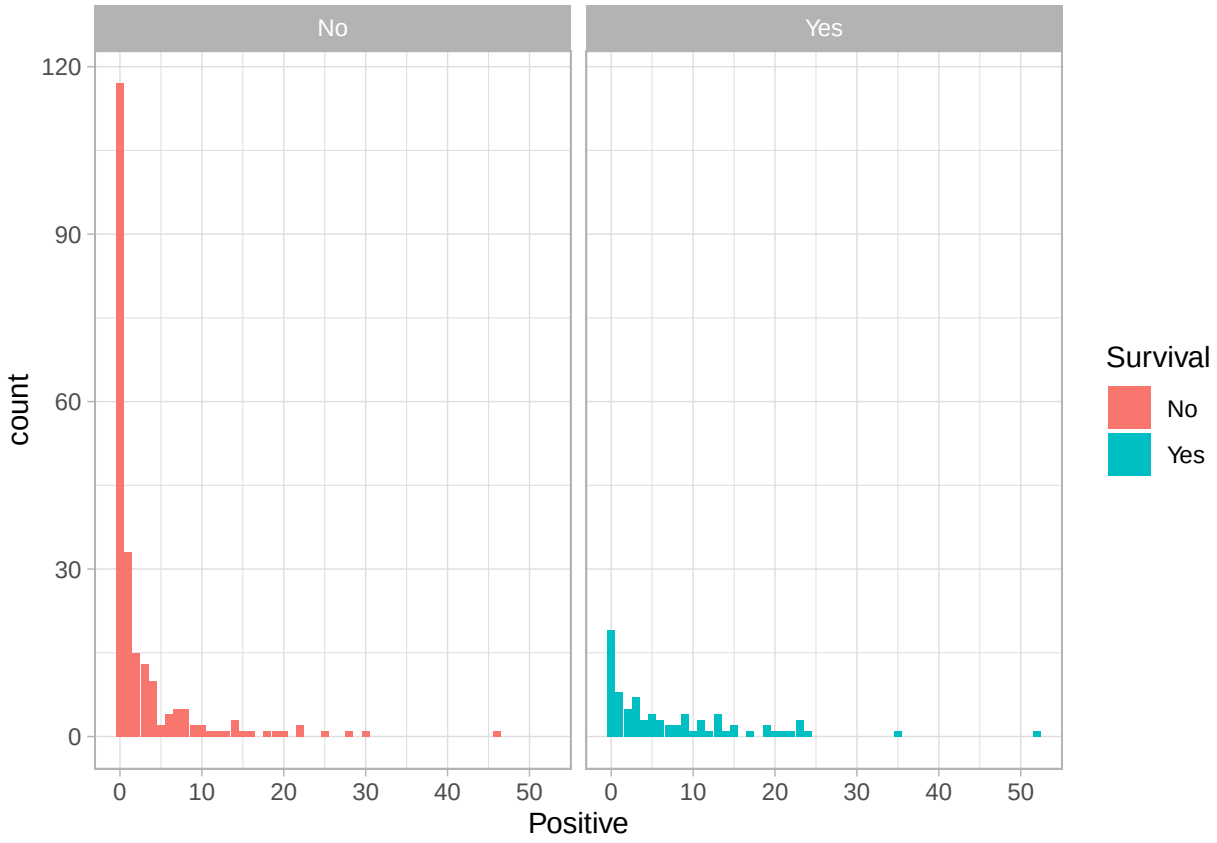
Para este dataset contamos con tres clasificadores con información de distinto tipo y bien organizada, por lo que no necesitamos hacer ningún tipo de ordenación/tratamiento. No existe ninguna relación entre variables sobre la información que codifican (en el sentido de que podrían agruparse).

La variable Year solo indica las dos últimas cifras del año de operación, pero como todas las instancias son del mismo siglo nos resulta más conveniente tenerla así

Resolución de hipótesis Nos habíamos planteado las siguientes hipótesis

- H.1: Habrá menor ratio de supervivencia cuanto mayor sea el número de nodos positivos encontrados: Por los razonamientos explicados en la introducción del problema.

```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Positive, fill=Survival)) +
    geom_bar() +
    facet_wrap(~ Survival) +
    theme_light()
```




```
haberman %>%
  bind_cols(labels) %>%
  filter(Positive>10) %>%
  summarise(Survival) %>%
  table()
```

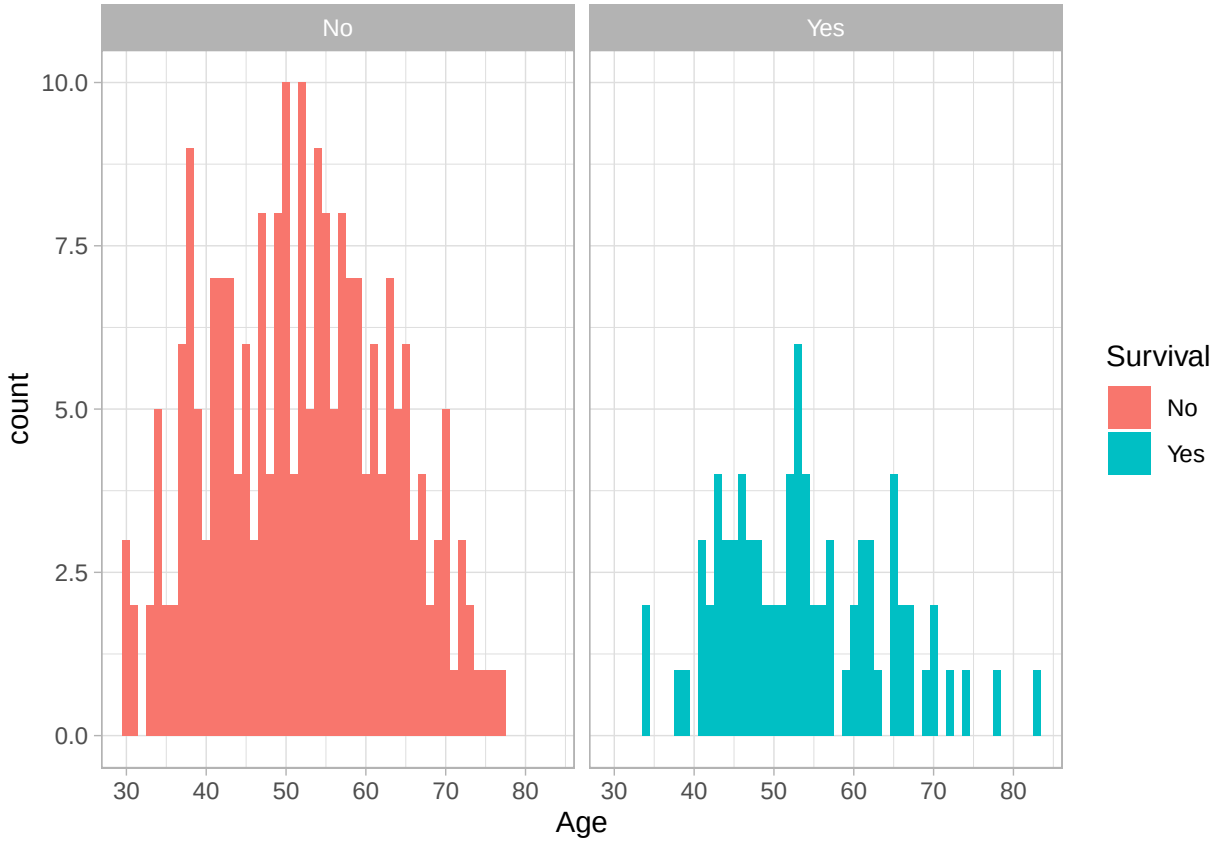
```
.
 No Yes
 17  23
```

Vemos que la hipótesis no es cierta

- H.2: Habrá mayor ratio de supervivencia cuanto más joven sea el paciente.

```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Age, fill=Survival)) +
    geom_bar() +
    facet_wrap(~ Survival) +
```

```
    theme_light()
```



Si miramos los pacientes <40

```
haberman %>%
  bind_cols(labels) %>%
  filter(Age<40) %>%
  summarise(Survival) %>%
  table()
```

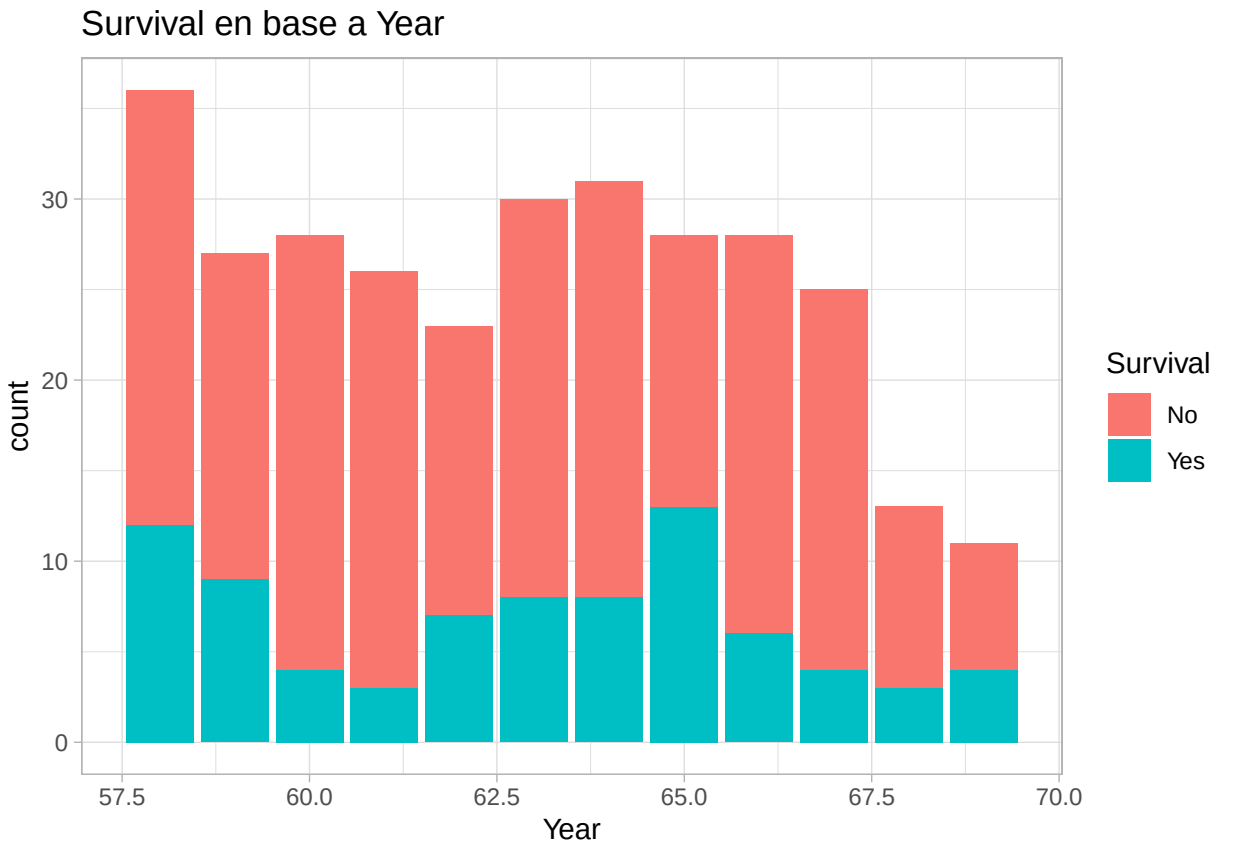
.

```
 No Yes
 36   4
```

Nos sale todo lo contrario.

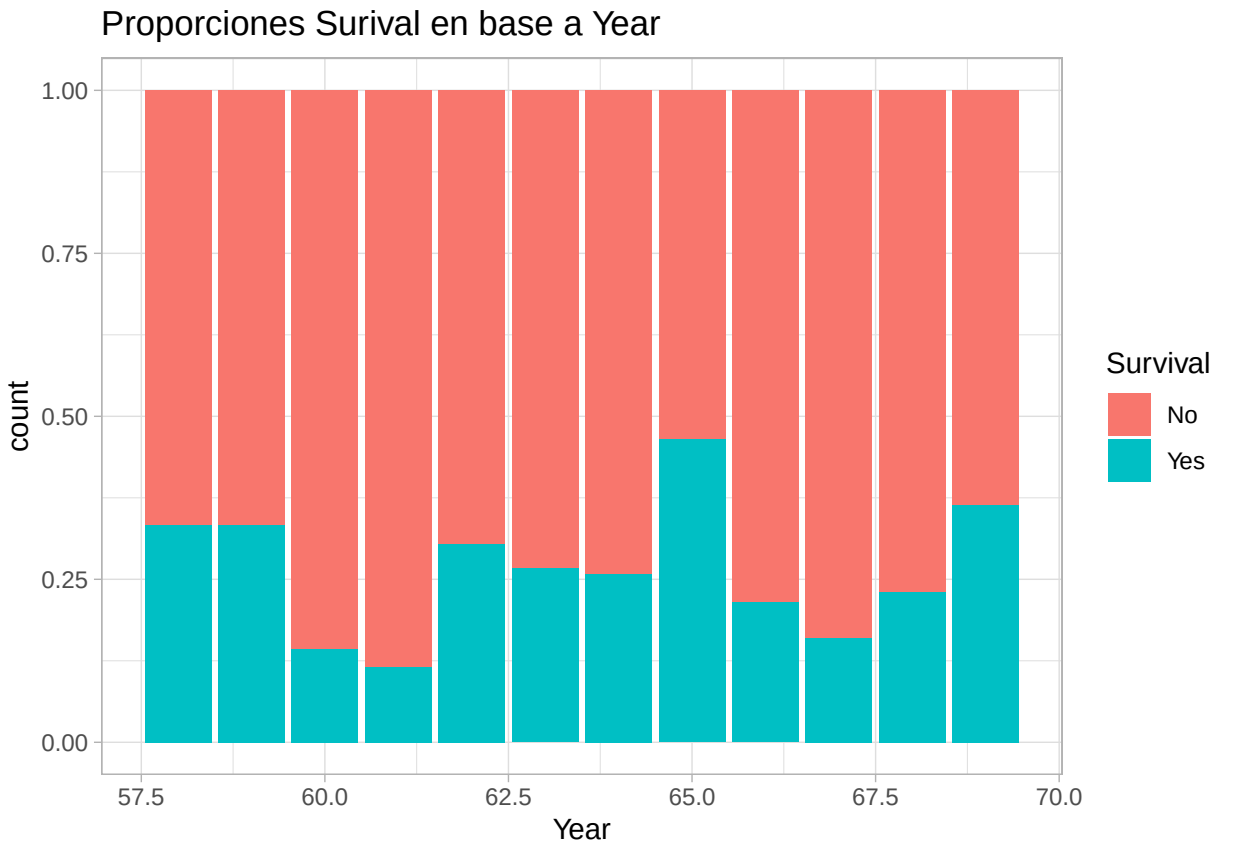
- H.3: El rango de Year es pequeño. La influencia de esta variable creemos que podría darse solo si durante ese período se hubieran descubierto técnicas mejores de cirugía. Este razonamiento va orientado de cara a la población y no a la muestra. Puesto que contamos con datos de un solo hospital durante pocos años, es posible que el equipo de cirugía hubiera sido el mismo para la mayoría de pacientes.

```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Year, fill=Survival)) +
    geom_bar() +
    labs(title="Survival en base a Year") +
    theme_light()
```



Decrementa el número de datos en años superiores, aunque las proporciones parecen aumentar. Lo mostramos:

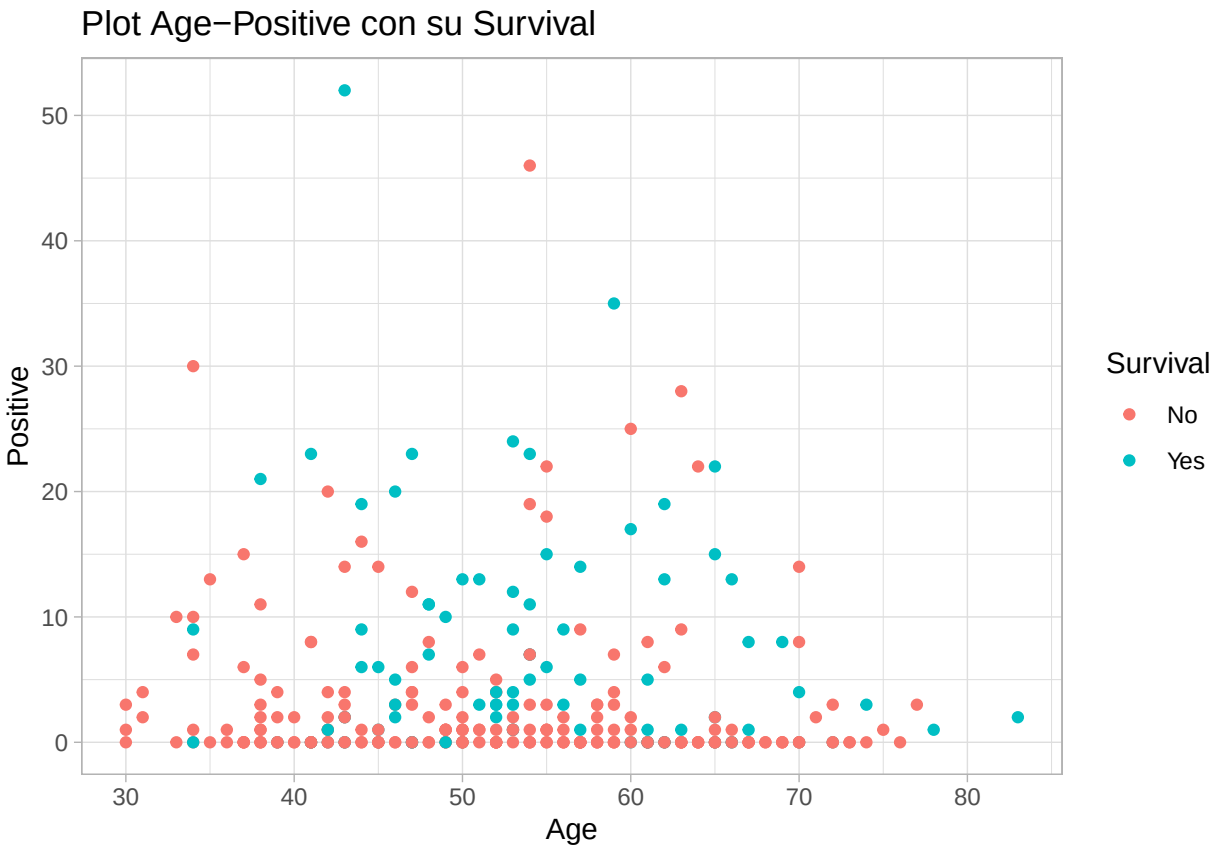
```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Year, fill=Survival)) +
    geom_bar(position="fill") +
    labs(title="Proporciones Surival en base a Year") +
    theme_light()
```



A excepción de algunos años, la porporción es bastante similar con los datos que tenemos, por lo que no podemos confirmar la hipótesis.

- H.4: Podría haber relación entre la edad y el número de positivos, posiblemente indicando lo tardío que se descubre el cáncer.

```
haberman %>%
  bind_cols(labels) %>%
  ggplot(aes(x=Age, y=Positive, color=Survival)) +
    geom_point() +
    labs(title="Plot Age-Positive con su Survival") +
    theme_light()
```



Conclusiones En resumen concluímos diciendo que tenemos un dataset con pocas variables, pero con ninguna correlación entre ellas, lo que nos favorece en el problema de clasificación que nos atañe. También hemos visto ausencia de normalidad en los clasificadores (PONER CONSECUENCIAS DE ESTO) ...

Además, contamos con más instancias de no supervivientes, algo que probablemente no nos resulte favorable para regresión logística y LDA (SEGURO?)

Únicamente hemos preprocesado los datos aplicando una estandarización, preparando el dataset a los algoritmos que se van a utilizar.