Modelos lineales

Armando Ocampo

Librerias de trabajo

Antes de comenzar a trabajar, te recomiendo abrir las siguientes librerias. Si no cuentas con alguna de ellas es posible instalarla mediante la función *install.packages()*

```
library(readr)
library(dplyr)
library(corrplot)
library(ggplot2)
library(scales)
library(ggpubr)
library(broom)
```

Asimismo, se agregan dos conjuntos de datos. Los cuales debes descargar y guardar en la carpeta *clean_data*.

Modelos lineales

Para este apartado utilizaremos el conjunto de datos de vacunación obtenido del portal **Our World in Data**, el cual se conforma por información de 171 países, agregando las variables esperanza de vida, producto interno bruto *per capita* y porcentaje de vacunación de 13 inmunizaciones. Este dataset tiene actualización al 31 de diciembre de 2019. El formato es un archivo de valores separados por coma (*comma separate values, csv*). Antes de comenzar, lo llamaremos a nuestro ambiente de trabajo.

```
vacunas_df <- read_csv('../data/dataset_vacunas.csv')
```

Para facilitar el uso de las variables del conjunto de datos utilizaremos la función *attach()* de la paquetería *base*. Esta permite que cada variable del dataset se vuelva un vector independiente, sin saturar la memoria del ambiente.

```
attach(vacunas_df)
```

Los modelos lineales describen la relación de la variable respuesta (dependiente) y la(s) variable(s) explicativa(s) (independiente). Esta relación puede ser positiva, negativa o estar ausente. En R, este tipo de modelos se realiza mediante la función *lm()* de la paquetería *stats*. Los argumentos que se colocan son los siguientes. *formula* = y_x, describe la relación de la variable "y" con la variable "x" ($_y x$, este elemento indica, en medida que x explica y). El siguiente argumento, *data*= detalla el conjunto de datos a utilizar. En el siguiente ejemplo se creará un modelo lineal que exlique la relación de la esperanza de vida con el producto interno bruto en el conjunto de datos de vacunas.

```
gdp_vs_life <- lm(formula=Life_expectancy~GDP, data = vacunas_df)
```

Para extraer la información del modelo se utilizan las funciones *print()*, *summary()* y *tidy()*.

```
summary(gdp_vs_life)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ GDP, data = vacunas_df)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.901  -3.883   1.317   4.144   8.623
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.756e+01  5.798e-01  116.53   <2e-16 ***
## GDP         2.485e-04  1.962e-05   12.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.386 on 169 degrees of freedom
## Multiple R-squared:  0.4869, Adjusted R-squared:  0.4839
## F-statistic: 160.4 on 1 and 169 DF,  p-value: < 2.2e-16
```

```
confint(gdp_vs_life)
```

```
##                    2.5 %       97.5 %
## (Intercept) 6.641799e+01 6.870703e+01
## GDP         2.097573e-04 2.872285e-04
```

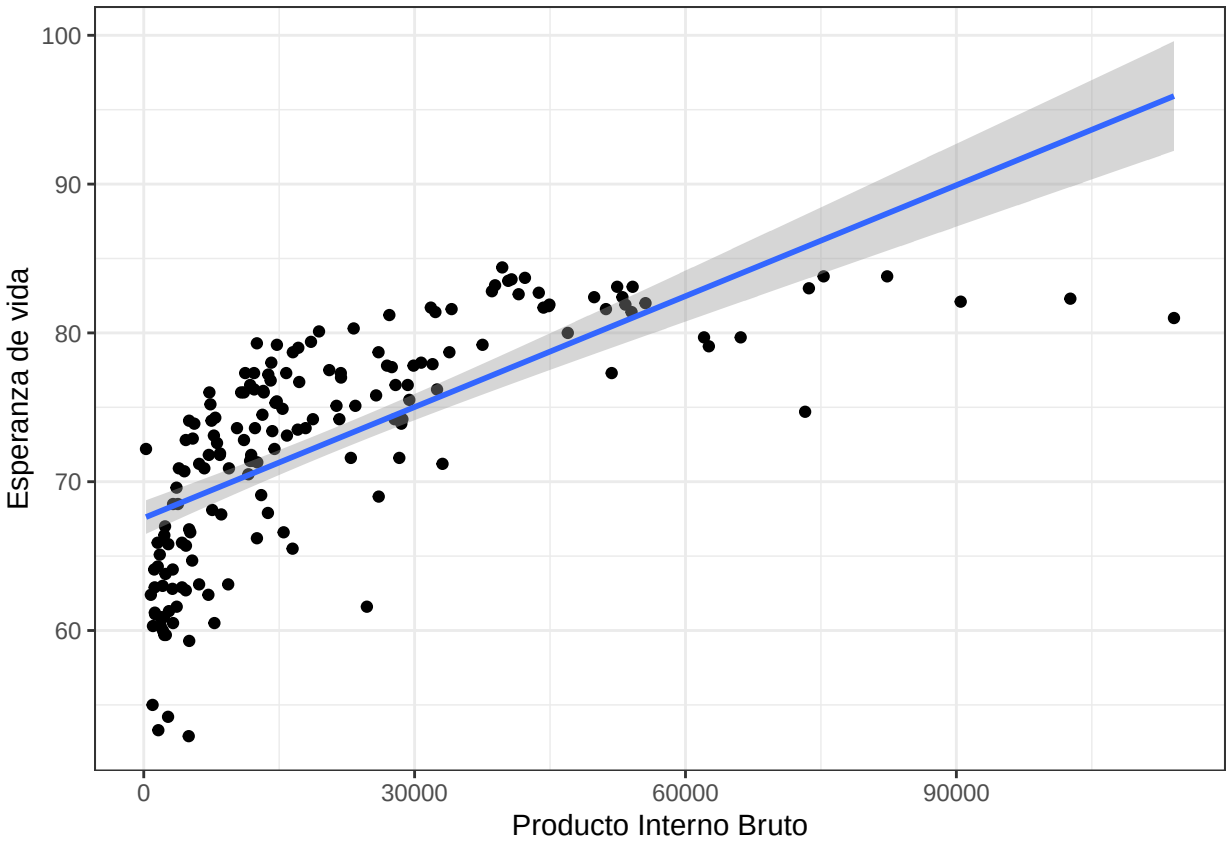
```
tidy(gdp_vs_life)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic   p.value
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) 67.6      0.580         117.  2.30e-163
## 2 GDP          0.000248 0.0000196      12.7 2.83e- 26
```

Además de los estadísticos de confianza, estas funciones generan los coeficientes que permiten generar la función que explica el modelo. Proporcionando la intercepción con el eje de las "y", y la relación de entre variables.

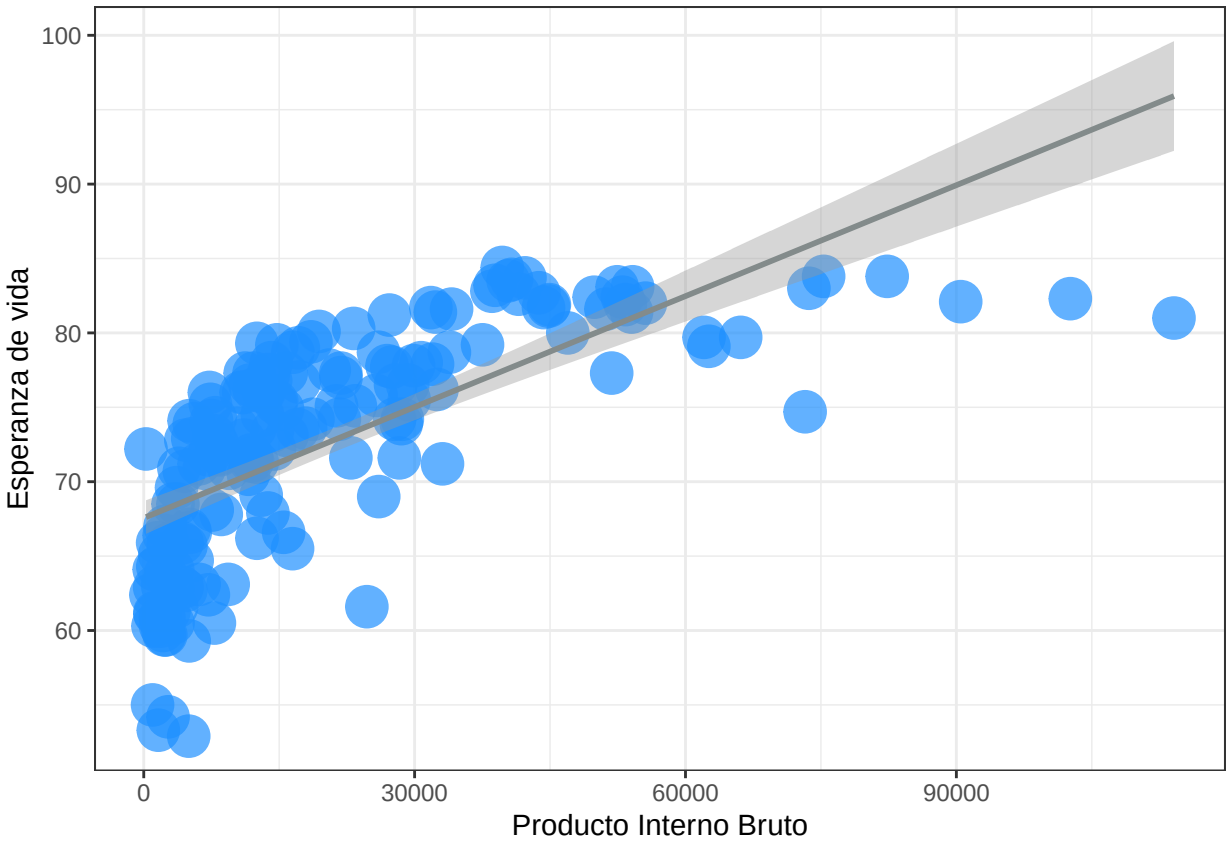
De esta manera obtenemos la siguiente fórmula $y = (0.000248)x + 67.6$. Posteriormente, podemos graficar el modelo. Para esto generaremos un gráfico de dispersión en la paquetería *ggplot2*, agregando la función *geom_smooth(method = 'glm')*, este argumento crea un modelo lineal generalizado.

```
ggplot(vacunas_df, aes(x = GDP, y = Life_expectancy)) +
  geom_point()+
  geom_smooth(method = 'glm')+
  xlab('Producto Interno Bruto')+
  ylab('Esperanza de vida')+
  theme_bw()
```



Para una mejor visualización se cambiará el color y tamaño de los puntos. Así como el color de la línea del modelo.

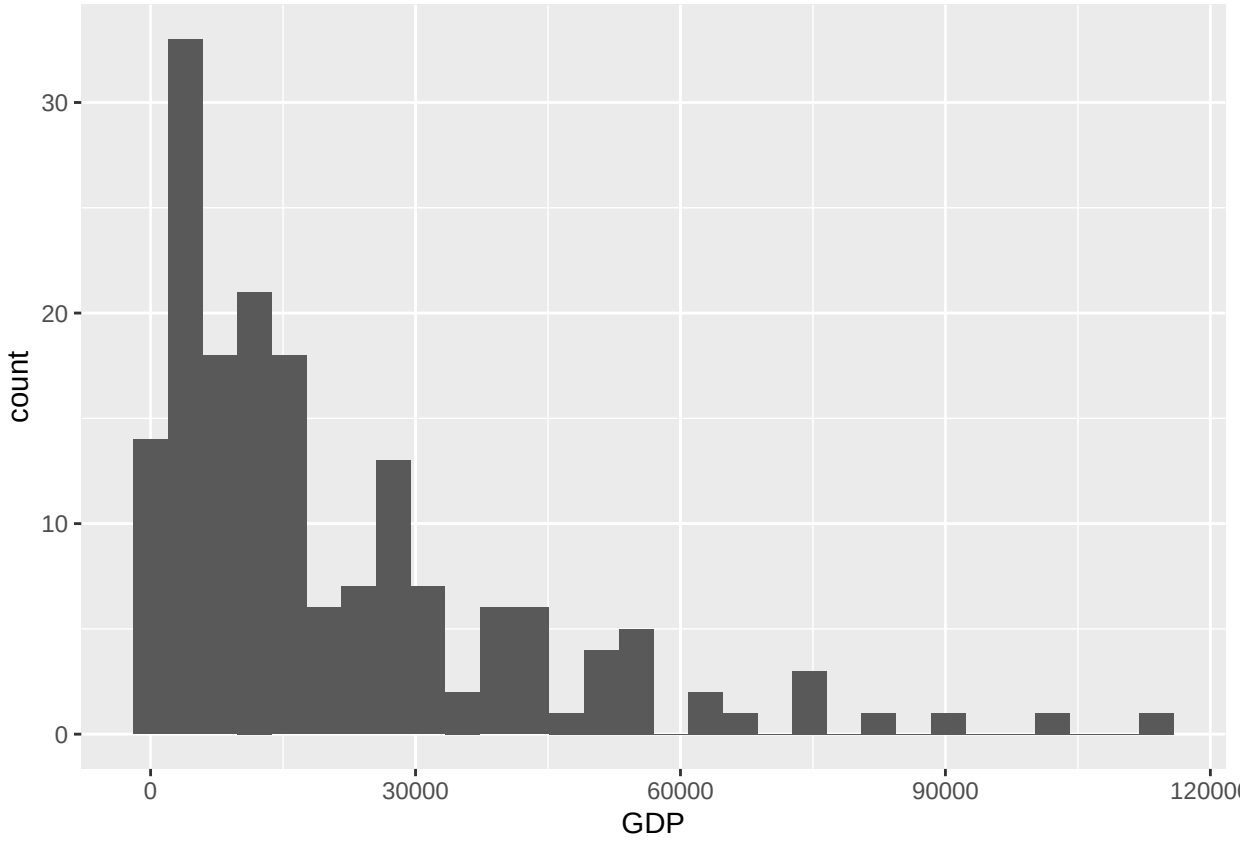
```
ggplot(vacunas_df, aes(x = GDP, y = Life_expectancy)) +
  geom_point(size = 7, color = 'dodgerblue', alpha = 0.7)+
  geom_smooth(method = 'glm', color = 'azure4')+
  xlab('Producto Interno Bruto')+
  ylab('Esperanza de vida')+
  theme_bw()
```



Uno de los puntos a resaltar es que la mayoría de los datos del producto interno bruto se encuentran desplazados a la izquierda. Para una mejor visualización se graficará la distribución de los datos.

```
ggplot(vacunas_df, aes(GDP)) +
  geom_histogram()
```

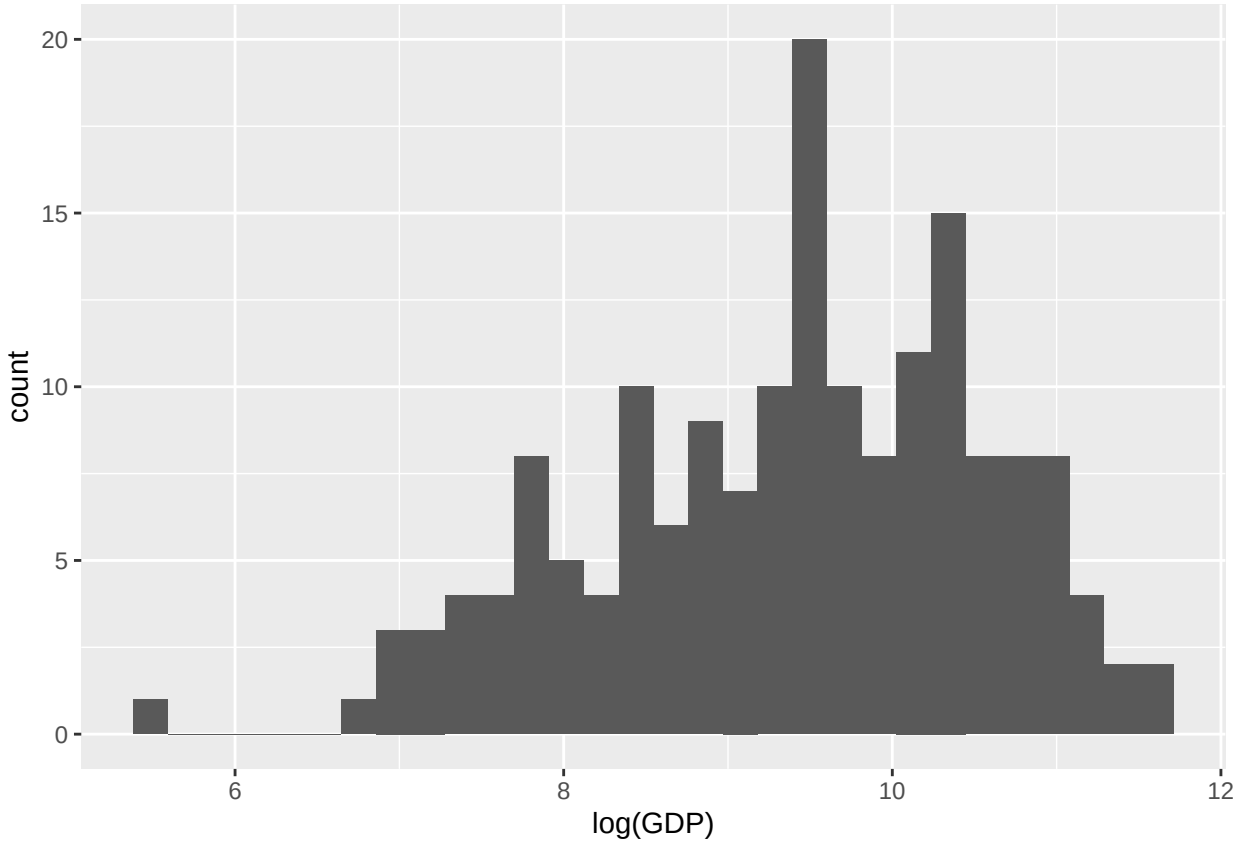
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Al observar la distribución se confirma el sesgo que presentan los datos. Cuando esto ocurre es válido realizar una transformación de los datos. Para esto se utiliza la transformación logarítmica o cuadrática. En este caso se realizará una transformación logarítmica de los datos.

```
ggplot(vacunas_df, aes(log(GDP)))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Los datos no siguen una distribución simétrica. No obstante, se ha reducido el sesgo de distribución. A continuación, se repetirá el modelo lineal, no obstante, los datos de PIB presentarán una transformación logarítmica.

```
gdp_vs_life_log <- lm(formula = Life_expectancy~log(GDP), data = vacunas_df)
```

De la misma forma, se extraerán los estadísticos e información del modelo. Donde la nueva función se explica de la siguiente manera $y = (5.24)x + 23.6$

```
summary(gdp_vs_life_log)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ log(GDP), data = vacunas_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2579  -2.0333   0.7346   3.0888  19.6932
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.5638     2.6597    8.86 1.08e-15 ***
## log(GDP)      5.2378     0.2812   18.63  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.303 on 169 degrees of freedom
## Multiple R-squared:  0.6725, Adjusted R-squared:  0.6706
```

```
## F-statistic:   347 on 1 and 169 DF,  p-value: < 2.2e-16
```

```
confint(gdp_vs_life_log)
```

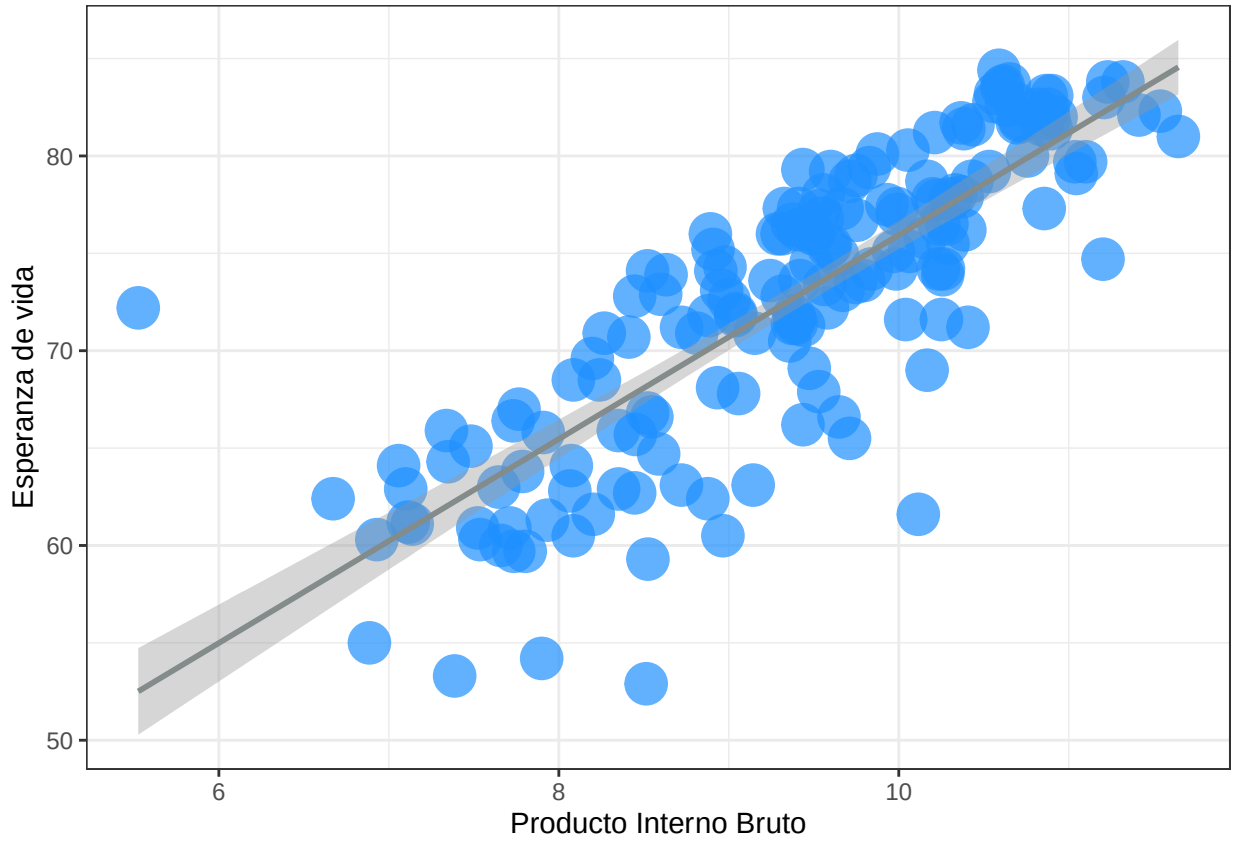
```
##                 2.5 %    97.5 %
## (Intercept) 18.313344 28.814272
## log(GDP)     4.682732  5.792827
```

```
tidy(gdp_vs_life_log)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    23.6      2.66       8.86 1.08e-15
## 2 log(GDP)        5.24     0.281     18.6  8.08e-43
```

En este punto, graficaremos el nuevo modelo.

```
ggplot(vacunas_df, aes(x = log(GDP), y = Life_expectancy)) +
  geom_point(size = 7, color = 'dodgerblue', alpha = 0.7)+
  geom_smooth(method = 'glm', color = 'azure4')+
  xlab('Producto Interno Bruto')+
  ylab('Esperanza de vida')+
  theme_bw()
```



Al observar el gráfico se detalla que los datos ya no se encuentran agrupados en un solo sitio. Por último compararemos ambos modelos.

```
tidy(gdp_vs_life)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic   p.value
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) 67.6      0.580         117.  2.30e-163
## 2 GDP          0.000248 0.0000196      12.7 2.83e- 26
```

```
tidy(gdp_vs_life_log)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    23.6      2.66       8.86 1.08e-15
## 2 log(GDP)        5.24     0.281     18.6  8.08e-43
```

El modelo con transformación logarítmica explica de manera más clara la relación entre las variables.

Regresión múltiple

En ocasiones dos o más variables explican el efecto presente sobre la variable respuesta. Cada efecto es independiente y puede ser positivo o negativo. Para realizar un regresión múltiple en R se utiliza la función *lm()*. Sin embargo, se concatenan las variables explicativas mediante el signo + en el argumento de formula. En el siguiente ejemplo, se muestra la relación de la esperanza de vida con las variables PIB, porcentaje de vacunación para las inmunizaciones BCG y poliomielitis.

```
life_vs_bcg_gdp_pol <- lm(formula = Life_expectancy~GDP+BCG+Pol3,
                          data = vacunas_df)
```

La extracción de los coeficientes y estadísticos se realiza de la misma manera que en el modelo lineal simple.

```
summary(life_vs_bcg_gdp_pol)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ GDP + BCG + Pol3, data = vacunas_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2765  -3.0328   0.7161   2.9667  10.6545
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 49.0696635  2.7701208  17.714  < 2e-16 ***
## GDP          0.0001540  0.0000212   7.263 1.37e-11 ***
## BCG         -0.0404308  0.0108487  -3.727 0.000265 ***
## Pol3         0.2629111  0.0322866   8.143 8.54e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.533 on 167 degrees of freedom
## Multiple R-squared:  0.6408, Adjusted R-squared:  0.6344
## F-statistic: 99.31 on 3 and 167 DF,  p-value: < 2.2e-16
```

```
confint(life_vs_bcg_gdp_pol)
```

```
##                    2.5 %       97.5 %
```

```
## (Intercept) 43.6006943794 54.5386325495
## GDP          0.0001121284  0.0001958369
## BCG         -0.0618490677 -0.0190124989
## Pol3         0.1991685300  0.3266536244
```

```
tidy(life_vs_bcg_gdp_pol)
```

```
## # A tibble: 4 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 49.1      2.77          17.7  3.43e-40
## 2 GDP          0.000154 0.0000212      7.26 1.37e-11
## 3 BCG         -0.0404   0.0108        -3.73 2.65e- 4
## 4 Pol3         0.263    0.0323         8.14 8.54e-14
```

De esta manera, es posible conocer una relación de un conjunto de variables explicativas con la variables respuesta.

Nota: al igual que en el modelo lineal simple, se recomienda conocer la distribución de los datos previo a la aplicación de cualquier transformación o desarrollo de modelos

Cuando se desea comparar una variable con el resto, existe un atajo para no concatenar todas las variables dentro del argumento fórmula. En este punto se describe lo siguiente: *formula* = *y*~. . De esta manera, se indica que se comparará la variable "y" con el resto de las variables en el dataset. En el siguiente ejemplo, se comparará la esperanza de vida con el resto de las variables, eliminando el código y nombre de cada región.

```
data_vacunas_2 <- vacunas_df %>%
  select(-Entity, -Code)

modelo_multivariable <- lm(formula=Life_expectancy~.,
                           data = data_vacunas_2)
```

Extracción de información del modelo.

```
summary(modelo_multivariable)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ ., data = data_vacunas_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5791  -2.1237   0.4144   2.2626   9.8391
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.370e+01  2.829e+00  18.985  < 2e-16 ***
## GDP          1.261e-04  1.836e-05   6.869 1.40e-10 ***
## BCG         -3.622e-02  9.386e-03  -3.859 0.000165 ***
## HepB3       -4.843e-03  1.900e-02  -0.255 0.799102
## Hib3        -1.427e-02  2.313e-02  -0.617 0.538229
## IPV1         2.303e-01  5.179e-02   4.447 1.63e-05 ***
## MCV1         4.744e-02  4.954e-02   0.958 0.339693
## PCV3        -6.007e-03  9.220e-03  -0.651 0.515683
## Pol3         6.602e-02  1.169e-01   0.565 0.573058
## RCV1         5.363e-02  1.446e-02   3.708 0.000288 ***
## RotaC       -2.643e-02  8.274e-03  -3.194 0.001692 **
## YFV         -2.206e-02  1.180e-02  -1.869 0.063439 .
```

```
## DTP3        -1.400e-01  1.213e-01  -1.154 0.250251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.797 on 158 degrees of freedom
## Multiple R-squared:  0.7616, Adjusted R-squared:  0.7435
## F-statistic: 42.06 on 12 and 158 DF,  p-value: < 2.2e-16
```

```
confint(modelo_multivariable)
```

```
##                     2.5 %       97.5 %
## (Intercept)  4.811700e+01 59.291335559
## GDP          8.986814e-05  0.000162400
## BCG         -5.475840e-02 -0.017683811
## HepB3       -4.235973e-02  0.032674549
## Hib3        -5.995438e-02  0.031418011
## IPV1         1.280374e-01  0.332630375
## MCV1        -5.040191e-02  0.145285238
## PCV3        -2.421724e-02  0.012203834
## Pol3        -1.648782e-01  0.296917754
## RCV1         2.506426e-02  0.082187520
## RotaC       -4.277031e-02 -0.010087245
## YFV         -4.536046e-02  0.001248865
## DTP3        -3.796813e-01  0.099633601
```

```
tidy(modelo_multivariable)
```

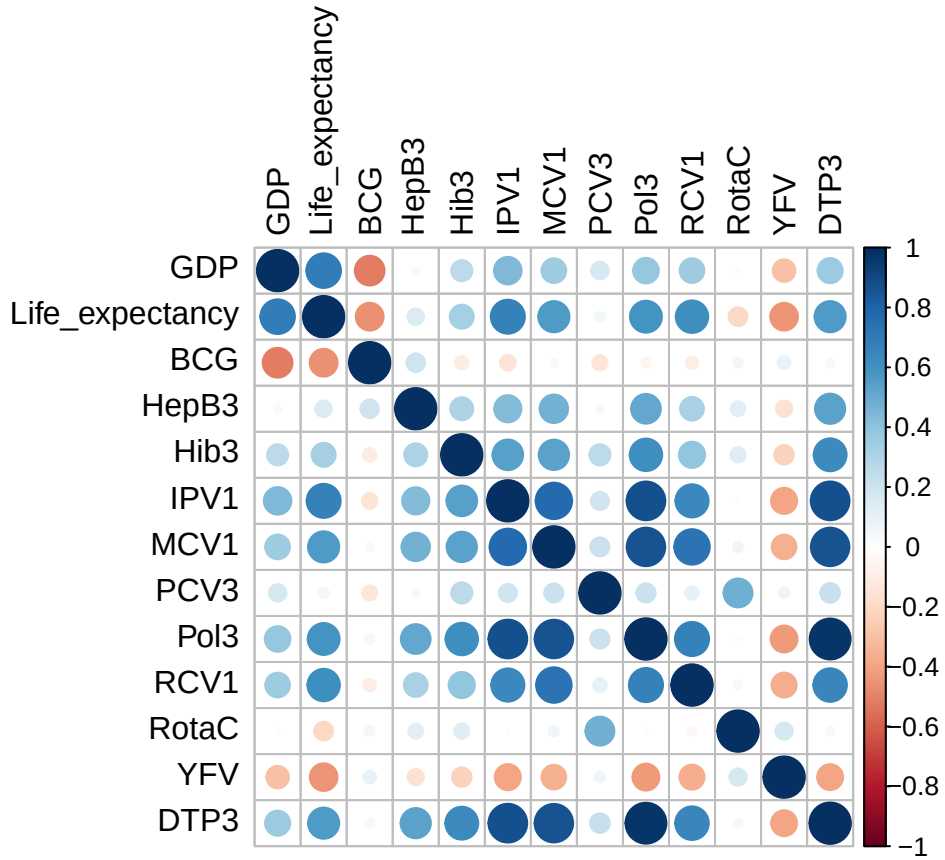
```
## # A tibble: 13 x 5
##    term         estimate std.error statistic  p.value
##    <chr>           <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept) 53.7      2.83         19.0   1.30e-42
##  2 GDP          0.000126 0.0000184     6.87  1.40e-10
##  3 BCG         -0.0362   0.00939      -3.86  1.65e- 4
##  4 HepB3       -0.00484  0.0190       -0.255 7.99e- 1
##  5 Hib3        -0.0143   0.0231       -0.617 5.38e- 1
##  6 IPV1         0.230    0.0518        4.45  1.63e- 5
##  7 MCV1         0.0474   0.0495        0.958 3.40e- 1
##  8 PCV3        -0.00601  0.00922      -0.651 5.16e- 1
##  9 Pol3         0.0660   0.117         0.565 5.73e- 1
## 10 RCV1         0.0536   0.0145        3.71  2.88e- 4
## 11 RotaC       -0.0264   0.00827      -3.19  1.69e- 3
## 12 YFV         -0.0221   0.0118       -1.87  6.34e- 2
## 13 DTP3        -0.140    0.121        -1.15  2.50e- 1
```

Correlación

La correlación define la dependencia de una o más variables de manera lineal. A partir de la función *cor()* de la paquetería *stats()* es posible construir una matriz de correlación entre dos o más variables. Por defecto esta función genera la matriz de correlación mediante el método de Pearson. No obstante, mediante el argumento *method* = , es posible modificar el método estadístico, teniendo como posibilidades las variantes de Kendall y Spearman. Esta información puede ser más atarctiva de manera visual al ser graficada. Esto se realiza mediante la función *corrplot()* de la paquetería *corrplot*. El siguiente ejemplo tomará la información del dataset de vacunas usado en el ejercicio previo, en el cual se eliminó la información de la entidad.

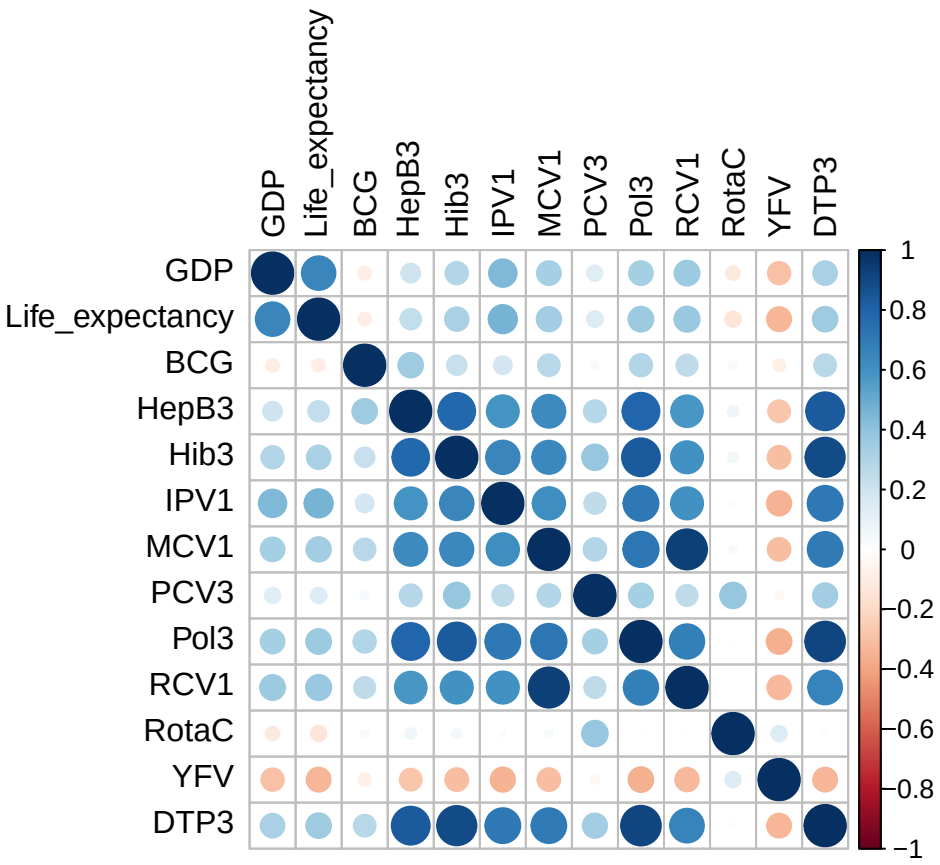
```
correlaciones <- cor(data_vacunas_2)
```

```
corrplot(correlaciones, method = 'circle', tl.col = 'black')
```



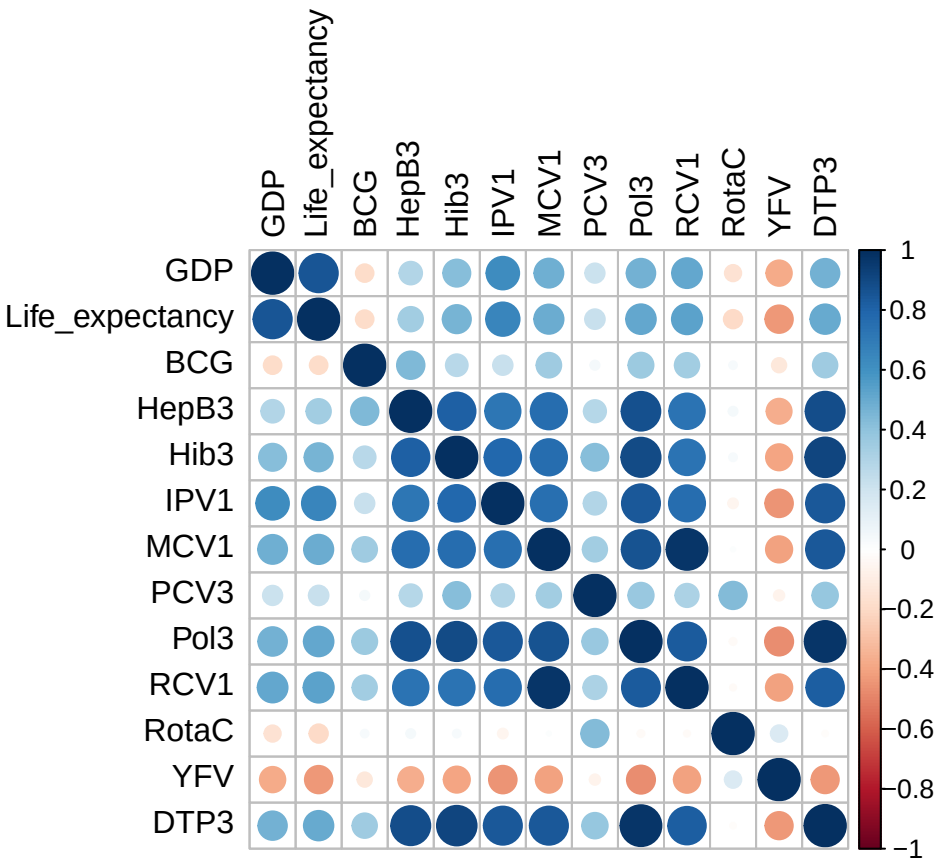
Como se ha mencionado, si se desea cambiar el método estadístico, se utiliza el argumento homónimo dentro de la función *cor()*. Posteriormente, es posible graficar esta información utilizando la función *corrplot()*.

```
correlaciones_kendall <- cor(data_vacunas_2, method = 'kendall')

corrplot(correlaciones_kendall, method = 'circle', tl.col = 'black')
```



En este ejemplo, se grafica la matriz de correlación utilizando el método de Spearman.

```
correlaciones_spearman <- cor(data_vacunas_2, method = 'spearman')

corrplot(correlaciones_spearman, method = 'circle', tl.col = 'black')
```



Regresión logisitica

La regresión logistíca permite generar modelos donde la información está representada por una serie de 1s y 0s. Indicando acierto o error e incluso presencia o ausencia de alguna característica; dependiendo cada caso. En el siguiente ejemplo utilizaremos información del dataset de cáncer cervicouterino, proveniente del *repositorio de aprendizaje de máquina de la universidad de Michigan*. En este conjunto de datos se muestra la relación de presentar cáncer cervicouterino a partir de la relación con factores de riesgo descritos en la literatura. El primer paso consta en llamar el dataset al ambiente de trabajo. Al igual que en el dataset previo, se utilizará la función *attach()* para trabajar con cada variable de manera independiente.

```
cancer_df <- readRDS('../data/cancer_data.RDS')

summary(cancer_df)
```

```
##       age         number_of_sexual_partners  smokes_years
##  Min.   :13.00   Min.   : 1.000            Min.   : 0.000
##  1st Qu.:21.00   1st Qu.: 2.000            1st Qu.: 0.000
##  Median :26.00   Median : 2.000            Median : 0.000
##  Mean   :27.28   Mean   : 2.521            Mean   : 1.228
##  3rd Qu.:33.00   3rd Qu.: 3.000            3rd Qu.: 0.000
##  Max.   :84.00   Max.   :28.000            Max.   :37.000
##  hormonal_anticonceptives stds_hepatitis_b   stds_condylomatosis
##  Min.   :0.0000           Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.0000           1st Qu.:0.000000   1st Qu.:0.00000
##  Median :1.0000           Median :0.000000   Median :0.00000
##  Mean   :0.6421           Mean   :0.001393   Mean   :0.05571
##  3rd Qu.:1.0000           3rd Qu.:0.000000   3rd Qu.:0.00000
```

```
##  Max.   :1.0000           Max.   :1.000000   Max.   :1.00000
##      dx_hpv        stds_genital_herpes stds_syphilis        stds_hiv
##  Min.   :0.00000   Min.   :0.000000    Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.000000    1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.000000    Median :0.00000   Median :0.0000
##  Mean   :0.02228   Mean   :0.001393    Mean   :0.02089   Mean   :0.0195
##  3rd Qu.:0.00000   3rd Qu.:0.000000    3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.000000    Max.   :1.00000   Max.   :1.0000
##  stds_mulluscum         dx_cin          dx_cancer
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.00000
##  Mean   :0.001393   Mean   :0.01114   Mean   :0.02368
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.00000
```

```
attach(cancer_df)
```

Los modelos de regresión logistica se generán con la función *glm()* de la paquetería *stats*. Los argumentos son similares a los utilizados en la función *lm()*, describiendo la fórmula y el dataset a utilizar. Sin embargo, se agrega el argumento *family = 'binomial'* para indicar el tipo de modelo a generar.

```
cancer_model <- glm(formula = dx_cancer~stds_condylomatosis, data = cancer_df,
    family = 'binomial')
```

La extracción de coeficientes se genera de la misma manera. Utilizando las funciones *print()*, *summary()* y *tidy()*.

```
print(cancer_model)
```

```
##
## Call:  glm(formula = dx_cancer ~ stds_condylomatosis, family = "binomial",
##     data = cancer_df)
##
## Coefficients:
##         (Intercept)  stds_condylomatosis
##              -3.661              -14.906
##
## Degrees of Freedom: 717 Total (i.e. Null);  716 Residual
## Null Deviance:       160.9
## Residual Deviance: 158.9     AIC: 162.9
```

```
summary(cancer_model)
```

```
##
## Call:
## glm(formula = dx_cancer ~ stds_condylomatosis, family = "binomial",
##     data = cancer_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2254  -0.2254  -0.2254  -0.2254   2.7151
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.6605     0.2456 -14.902   <2e-16 ***
## stds_condylomatosis  -14.9055  1031.3197  -0.014    0.988
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 160.86  on 717  degrees of freedom
## Residual deviance: 158.89  on 716  degrees of freedom
## AIC: 162.89
##
## Number of Fisher Scoring iterations: 17
```

```
tidy(cancer_model)
```

```
## # A tibble: 2 x 5
##   term                estimate std.error statistic  p.value
##   <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)            -3.66     0.246  -14.9    3.18e-50
## 2 stds_condylomatosis   -14.9   1031.      -0.0145 9.88e- 1
```

Asimismo, para definir la relación con dos o más variables, se genera una cadena de vectores utilizando el signo $+$.

```
cancer_model_2variables <- glm(formula = dx_cancer~stds_hepatitis_b+dx_hpv,
                               data = cancer_df, family = 'binomial')
```

Extracción de coeficientes

```
print(cancer_model_2variables)
```

```
##
## Call:  glm(formula = dx_cancer ~ stds_hepatitis_b + dx_hpv, family = "binomial",
##     data = cancer_df)
##
## Coefficients:
##      (Intercept)  stds_hepatitis_b            dx_hpv
##           -5.857           -10.710             8.565
##
## Degrees of Freedom: 717 Total (i.e. Null);  715 Residual
## Null Deviance:       160.9
## Residual Deviance: 34.91     AIC: 40.91
```

```
summary(cancer_model_2variables)
```

```
##
## Call:
## glm(formula = dx_cancer ~ stds_hepatitis_b + dx_hpv, family = "binomial",
##     data = cancer_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3548  -0.0756  -0.0756  -0.0756   3.4233
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -5.8565     0.7081  -8.271  < 2e-16 ***
## stds_hepatitis_b  -10.7096  2399.5448  -0.004    0.996
## dx_hpv              8.5646     1.2522   6.839 7.95e-12 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 160.865  on 717  degrees of freedom
## Residual deviance:  34.913  on 715  degrees of freedom
## AIC: 40.913
##
## Number of Fisher Scoring iterations: 15
```

```
tidy(cancer_model_2variables)
```

```
## # A tibble: 3 x 5
##   term             estimate std.error statistic  p.value
##   <chr>               <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)         -5.86     0.708  -8.27    1.33e-16
## 2 stds_hepatitis_b   -10.7   2400.     -0.00446 9.96e- 1
## 3 dx_hpv               8.56     1.25    6.84    7.95e-12
```

De la misma manera, utilizando el argumento ~. como variable respuesta podemos seleccionar el total de las variables y definir su relación con la variable dependiente. En el siguiente ejemplo se descartarán las columnas con atributos no binarios. A continuación se generará el modelo describiendo la relación del resto de los factores de riesgo con el diagnóstico de cáncer.

```
cancer_df_binomial <- cancer_df %>%
  select(-age, -number_of_sexual_partners, -smokes_years)

cancer_multiva <- glm(formula = dx_cancer~., data = cancer_df_binomial,
                      family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Extracción de coeficientes

```
print(cancer_multiva)
```

```
##
## Call:  glm(formula = dx_cancer ~ ., family = "binomial", data = cancer_df_binomial)
##
## Coefficients:
##              (Intercept)  hormonal_anticonceptives          stds_hepatitis_b
##                  -5.2853                   -0.8544                   -0.7982
##      stds_condylomatosis                    dx_hpv       stds_genital_herpes
##                 -14.6633                    8.6873                  -14.4263
##            stds_syphilis                  stds_hiv            stds_mulluscum
##                 -14.6283                  -14.4826                  -15.2807
##                   dx_cin
##                 -14.5577
##
## Degrees of Freedom: 717 Total (i.e. Null);  708 Residual
## Null Deviance:       160.9
## Residual Deviance: 34.02     AIC: 54.02
```

```
summary(cancer_multiva)
```

```
##
## Call:
```

```
## glm(formula = dx_cancer ~ ., family = "binomial", data = cancer_df_binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2904  -0.1005  -0.0656  -0.0656   3.5048
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -5.2853     0.9056  -5.836 5.34e-09 ***
## hormonal_anticonceptives    -0.8544     1.2931  -0.661    0.509
## stds_hepatitis_b            -0.7982 18316.1324   0.000    1.000
## stds_condylomatosis        -14.6633  2727.7193  -0.005    0.996
## dx_hpv                       8.6873     1.3723   6.330 2.44e-10 ***
## stds_genital_herpes        -14.4263 17730.3699  -0.001    0.999
## stds_syphilis              -14.6283  4413.7156  -0.003    0.997
## stds_hiv                   -14.4826  4595.0722  -0.003    0.997
## stds_mulluscum             -15.2807 17730.3699  -0.001    0.999
## dx_cin                     -14.5577  6079.3657  -0.002    0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 160.865  on 717  degrees of freedom
## Residual deviance:  34.024  on 708  degrees of freedom
## AIC: 54.024
##
## Number of Fisher Scoring iterations: 19
```

```
tidy(cancer_multiva)
```

```
## # A tibble: 10 x 5
##    term                      estimate std.error  statistic  p.value
##    <chr>                        <dbl>     <dbl>      <dbl>    <dbl>
##  1 (Intercept)                -5.29      0.906 -5.84      5.34e- 9
##  2 hormonal_anticonceptives   -0.854     1.29  -0.661     5.09e- 1
##  3 stds_hepatitis_b           -0.798 18316.    -0.0000436 1.00e+ 0
##  4 stds_condylomatosis       -14.7    2728.    -0.00538   9.96e- 1
##  5 dx_hpv                      8.69      1.37   6.33      2.44e-10
##  6 stds_genital_herpes       -14.4   17730.    -0.000814  9.99e- 1
##  7 stds_syphilis             -14.6    4414.    -0.00331   9.97e- 1
##  8 stds_hiv                  -14.5    4595.    -0.00315   9.97e- 1
##  9 stds_mulluscum            -15.3   17730.    -0.000862  9.99e- 1
## 10 dx_cin                    -14.6    6079.    -0.00239   9.98e- 1
```

Modelos de predicción

Derivado de la información generada de los modelos es posible realizar predicciones. Es decir, cuál será el efecto de una variable hipotética en el conjunto de datos. Para esto podemos generar una función a partir de la extracción de los coeficientes del modelo. Antes de comenzar, vamos a recordar el modelo sin tranformación de la relación del PIB y la esperanza de vida.

```
gdp_vs_life <- lm(Life_expectancy~GDP, data = vacunas_df)
tidy(gdp_vs_life)
```

```
## # A tibble: 2 x 5
```

```
##   term         estimate std.error statistic   p.value
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) 67.6      0.580         117.  2.30e-163
## 2 GDP          0.000248 0.0000196      12.7 2.83e- 26
```

Posteriormente, desarrolaremos una función con los valores de intercepción y estimación del modelo. Por último, colocaremos un nuevo valor para predecir la esperanza de vida teórica.

```
funcion_esperanza <- function(x){
  gdp = 0.000248
  intercept = 67.6
  resultado = intercept + gdp*x
  return(resultado)
}

funcion_esperanza(12531)
```

```
## [1] 70.70769
```

Esta función es eficaz. Sin embargo, la función *predict()* de la paquetería *stats* toma la información de un modelo y plantea valores teóricos para datos nuevos. En el siguiente ejemplo se usará la función *predict()* para calcular el valor de una variable nueva.

```
nueva_esperanza <- data.frame(esperanza = c(12531))

head(predict(gdp_vs_life, nueva_esperanza, type = 'response'),1)
```

```
## Warning: 'newdata' had 1 row but variables found have 171 rows
```

```
##        1
## 70.67657
```

Esta función tiene como ventaja el realizar más de una predicción a la vez.

```
nueva_esperanza_2 <- data.frame(esperanza = c(12531, 11787, 7159))

head(predict(gdp_vs_life, nueva_esperanza_2, type = 'response'),3)
```

```
## Warning: 'newdata' had 3 rows but variables found have 171 rows
```

```
##        1        2        3
## 70.67657 70.49162 69.34161
```