

Práctica 2: Limpieza y validación de los datos

1. Descripción del dataset

El conjunto de datos a analizar se ha obtenido a través de UCI (<https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008>). Los datos son relativos a pacientes de diabetes que tras haber sido tratados, necesitan de una readmisión o no.

El dataset se compone de un único csv que agrupa datos clínicos durante 10 años (1999-2008) de 130 hospitales de los Estados Unidos.

Cada fila representa un ingreso hospitalario de un paciente con diabetes, donde el motivo del ingreso y el diagnóstico fue algún tipo de diabetes. La duración del ingreso es de entre 1 y 14 días y se llevaron a cabo test de laboratorio, además de suministrar medicinas a los pacientes.

Hay un total de 101.767 registros, donde cada fila contiene un total de 50 atributos:

- encounter_id: id de ingreso
- patient_nbr: número de paciente
- race: etnia del paciente. Valores: Caucasian, Asian, African American, Hispanic, y otros.
- gender: sexo del paciente. Valores: male, female, and unknown/invalid
- age: edad del paciente. Agrupado en intervalos de 10 años.
- weight: peso del paciente en libras.
- admission_type_id: tipo de admisión con 9 valores posibles, por ejemplo, emergencia, urgente, neonato y no disponible.
- discharge_disposition_id: tipo de alta del paciente, con 29 valores posibles.
- admission_source_id: tipo de admisión, por ejemplo derivación de especialista, emergencia o transferido de hospital.
- time_in_hospital: tiempo en días que permanece el paciente en el hospital desde que se ingresa hasta que se le da el alta.
- payer_code: identificador del medio de pago.
- medical_specialty: especialidad desde donde se derivó al paciente

- num_lab_procedures: número de test de laboratorio que se realizaron desde el ingreso del paciente.
- num_procedures: número de procedimientos, aparte de los test del laboratorio, realizados.
- num_medications: número de las distintas medicinas que se le proporciona al enfermo.
- number_outpatient: número de visitas como paciente externo en el año previo al ingreso.
- number_emergency: número de emergencias sufridas por el paciente en el año previo al ingreso
- number_inpatient: número de visitas como paciente interno
- diag_1: diagnóstico primario
- diag_2: diagnostico secundario
- diag_3: diagnostico secundario adicional
- number_diagnoses: número de diagnósticos introducidos en el sistema
- max_glu_serum: resultado del test de glucosa
- A1Cresult: resultado del test A1c. Indica su valor o si no se ha tomado. Values: ">8" if the result was greater than 8%, ">7" if the result was greater than 7% but less than 8%, "normal" if the result was less than 7%, and "none" if not measured.

Los siguiente 24 atributos indican características de los medicamentos. Sus valores son **up** si se aumentó la dosis, **down** si se redujo, **steady** si se mantuvo y **no** si el medicamento no fue prescrito.

- metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-pioglitazone,
- change: indica si ha habido un cambio en la medicación, con valores yes y no
- diabetesMed: indica si hubo alguna medicación prescrita. Toma los valores yes y no.
- readmitted: Indica si el paciente tuvo que ser readmitido. Toma valores **<30** si tuvo que ser readmitido en menos de 30 días, **>30** si fue readmitido en más de 30 días y **No** sino necesitó ser readmitido.

Objetivo de análisis

El objetivo de análisis es determinar qué variables influyen en la readmisión o no de un paciente ya tratado. De esta forma, será posible establecer modelos de regresión que permitan predecir en base a ciertas características si un paciente necesitará volver a ser tratado o no.

Este tipo de análisis adquiere relevancia en centros médicos, permitiendo optimizar los recursos y personalizar los tratamientos, ya que cuando un paciente necesita de una readmisión, el coste económico es elevado.

2. Integración y selección de los datos de interés a analizar.

Este dataset ya ha sido tratado, ya que se ha usado en diversos estudios científicos. Es por eso que todos los atributos resultan relevantes a primera vista. Aun así, si se realiza un primer análisis sobre los datos para determinar los datos faltantes por columnas, se observa que los atributos de weight (97%), payer code (52%) y medical specialty (53%) tienen un alto porcentaje de datos perdidos. Para el resto de los atributos este valor no supera el 2%.

Existiendo tanto valores desconocidos, asumimos que son atributos que no tendrán gran importancia sobre el resultado. Por tanto, para evitar la dispersión de los datos y disminuir la complejidad del dataset se eliminarán estas columnas a la hora de realizar el tratamiento de los datos y su posterior análisis.

```
> diabetic_data <- diabetic_data[, -6]
> diabetic_data <- diabetic_data[, -10]
> diabetic_data <- diabetic_data[, -10]
```

Esto reduce el número de columnas a 47. Sigue siendo un número considerable, así que para reducir el alcance de la práctica se eliminarán las columnas de medicamentos y variables que no estén directamente ligadas a la fisionomía del paciente, para poder ligar un reingreso a las condiciones físicas del paciente. También eliminaremos la columna **patient_nbr** y **encounter_id**, ya que todos los valores son únicos e identifican a un sólo registro.

De esta forma, el dataset final contará con 17 columnas, siendo estas:

race, gender, age, discharge_disposition_id, time_in_hospital, num_lab_procedures, num_procedures, diag_1, diag_2, diag_3, number_diagnoses, max_glu_serum, A1Cresult, change, diabetesMed, readmitted.

Para seleccionarlas se usará el método subset

```
diabetic_data <- subset(diabetic_data,select=c(3,4,5,8,10,13, 14, 19,  
20, 21, 22, 23, 24, 48, 49, 50))
```

Al echar un primer vistazo al dataset se observa que los valores nulos se representan con un carácter ?. Por ello, se realizará una sustitución de estos caracteres por una cadena vacía de forma que R los trate como valores NA (not available).

3. Limpieza de los datos

Antes de empezar a limpiar los datos, realizamos una lectura del fichero en formato csv en el que se encuentran mediante la función read.csv. Este proceso se realizará en el lenguaje de programación R y con el IDE RStudio.

El resultado obtenido tras la lectura será un objeto data.frame. Al leer el csv indicamos a su que se traten las cadenas vacías como datos de tipo NA y que las strings no se traten como factores, sino como cadenas de caracteres.

```
directory <- getwd()  
diabetic_data <- read.csv(file.path(directory,"diabetic_data.csv"),  
header=TRUE, na.strings = c("", "NA"), stringsAsFactors=FALSE)  
str(diabetic_data)  
  
'data.frame': 101766 obs. of 17 variables:  
 $ encounter_id : int 2278392 149190 64410 500364 16680 35754 55842 63768 12522 15738 ...  
 $ race : chr "Caucasian" "Caucasian" "AfricanAmerican" "Caucasian" ...  
 $ gender : chr "Female" "Female" "Female" "Male" ...  
 $ age : chr "[0-10]" "[10-20]" "[20-30]" "[30-40]" ...  
 $ discharge_disposition_id: int 25 1 1 1 1 1 1 1 3 ...  
 $ time_in_hospital : int 1 3 2 2 1 3 4 5 13 12 ...  
 $ num_lab_procedures : int 41 59 11 44 51 31 70 73 68 33 ...  
 $ num_procedures : int 0 0 5 1 0 6 1 0 2 3 ...  
 $ diag_1 : chr "250.83" "276" "648" "8" ...  
 $ diag_2 : chr "" "250.01" "250" "250.43" ...  
 $ diag_3 : chr "" "255" "v27" "403" ...  
 $ number_diagnoses : int 1 9 6 7 5 9 7 8 8 8 ...  
 $ max_glu_serum : chr "None" "None" "None" "None" ...  
 $ A1Cresult : chr "None" "None" "None" "None" ...  
 $ change : chr "No" "Ch" "No" "Ch" ...  
 $ diabetesMed : chr "No" "Yes" "Yes" "Yes" ...  
 $ readmitted : chr "NO" ">30" "NO" "NO" ...
```

Comprobamos el tipo de los atributos del objeto obtenido.

```
sapply(diabetic_data, function(x) class(x))
```

```

encounter_id      race      gender      age
"integer"         "character" "character" "character"
discharge_disposition_id  time_in_hospital  num_lab_procedures  num_procedures
"integer"         "integer"         "integer"         "integer"
diag_1            diag_2    diag_3    number_diagnoses
"character"       "character" "character" "integer"
max_glu_serum     A1Cresult  change    diabetesMed
"character"       "character" "character" "character"
readmitted
"character"

```

Tenemos tanto tipo de datos char como integer. Para num_lab_procedures y num_procedures comprobamos si existen valores menores que 0, situación que se trataría como un error en los datos.

```
> apply(diabetic_data, function(x) sum(x < 0))
```

Para ambos atributos obtenemos que no existen valores de este estilo.

Finalmente, la variable de output es una variable categórica que puede tomar tres valores, **<30** si el paciente tuvo que ser readmitido en menos de 30 días, **>30** si fue readmitido en más de 30 días y **No** sino necesitó ser readmitido. Para simplificar el análisis, se realizará una transformación de forma que la variable de output refleje si el paciente necesitó ser readmitido (1) o no (0) directamente sobre el csv de datos antes de cargarlo.

3.1. Ceros y elementos vacíos

A continuación comprobamos si alguna de las variables contiene valores nulos.

```

apply(diabetic_data, function(x) sum(is.na(x)))
encounter_id      race      gender      age
0                2273      0          0
discharge_disposition_id  time_in_hospital  num_lab_procedures  num_procedures
0                0          0          0
diag_1            diag_2    diag_3    number_diagnoses
21               358      1423      0
max_glu_serum     A1Cresult  change    diabetesMed
0                0          0          0
readmitted
0

```

Vemos que los atributos **race**, **diag_1**, **diag_2** y **diag_3** contienen valores nulos. Es necesario tratar con los registros que contienen valores desconocidos en algún campo. Por otra parte, la columna de **gender** presenta campos con valor "Unknown/Invalid" que también deberíamos tratar como nulos.

```

> unique(diabetic_data$gender, incomparables = FALSE)
[1] Female      Male      Unknown/Invalid
Levels: Female Male Unknown/Invalid

```

Una opción sería eliminar estos registros, pero como se incurriría en pérdida de información, se plantea rellenar los valores faltantes mediante el imputación basada en k vecinos más próximos.

Se elige esta opción basándose en que los registros guardan cierta relación entre sí y puede predecirse el valor faltante basándose en el resto de variables. Trabajar con datos aproximados permitirá tener un menor margen de error que trabajando con datos vacíos. Este método se aplicará a los atributos **race** y **diag_1**, ya que todos los pacientes tienen una etnia, aunque esta no se haya especificado, y han recibido un diagnóstico, ya que son registros que implican que existe un alta del paciente.

```
#Instalacion del paquete VIM
>install.packages("VIM")
# Imputación de valores mediante la función kNN() del paquete VIM
suppressWarnings(suppressMessages(library(VIM)))
```

```
diabetic_data$race <- kNN(diabetic_data)$race
diabetic_data$diag_1 <- kNN(diabetic_data)$diag_1
diabetic_data$gender <- kNN(diabetic_data)$gender
```

Tras realizar la aproximación, comprobamos de nuevo la existencia de datos vacíos

```
sapply(diabetic_data, function(x) sum(is.na(x)))
```

encounter_id	patient_nbr	race	gender
0	0	0	0
age	admission_type_id	discharge_disposition_id	admission_source_id
0	0	0	0
time_in_hospital	num_lab_procedures	num_procedures	num_medications
0	0	0	0
number_outpatient	number_emergency	number_inpatient	diag_1
0	0	0	0
diag_2	diag_3	number_diagnoses	max_glu_serum
358	1423	0	0
A1Cresult	metformin	repaglinide	nateglinide
0	0	0	0
chlorpropamide	glimepiride	acetohexamide	glipizide
0	0	0	0
glyburide	tolbutamide	pioglitazone	rosiglitazone
0	0	0	0
acarbose	miglitol	troglitazone	tolazamide
0	0	0	0
examide	citoglipton	insulin	glyburide.metformin
0	0	0	0
glipizide.metformin	glimepiride.pioglitazone	metformin.rosiglitazone	metformin.pioglitazone
0	0	0	0
change	diabetesMed	readmitted	
0	0	0	

En el caso de **diag_2** y **diag_3** son diagnósticos que no necesariamente deben existir, ya que se trata de diagnósticos que complementan al diagnóstico principal. Por ello, si no aparecen asumiremos que no se emitió ningún diagnóstico secundario, así que sustituimos estos valores desconocidos por “No diagnosis”

```
diabetic_data$diag_2[diabetic_data$diag_2 == ""] <- "No diagnosis"
```

3.2. Identificación y tratamiento de valores extremos.

El siguiente paso sería comprobar si existen valores extremos en alguno de los parámetros. Se considera valor atípico o extremo aquel valor que está numéricamente distante del resto de los datos, en concreto, a 3 desviaciones estándar alejados de la media.

En este punto, analizaremos los valores numéricos. Calcularemos el umbral a partir del cual consideraríamos que se trata de un valor atípico.

A continuación calculamos el máximo y el histograma de cada uno de los atributos para localizar estos valores

Time_in_hospital

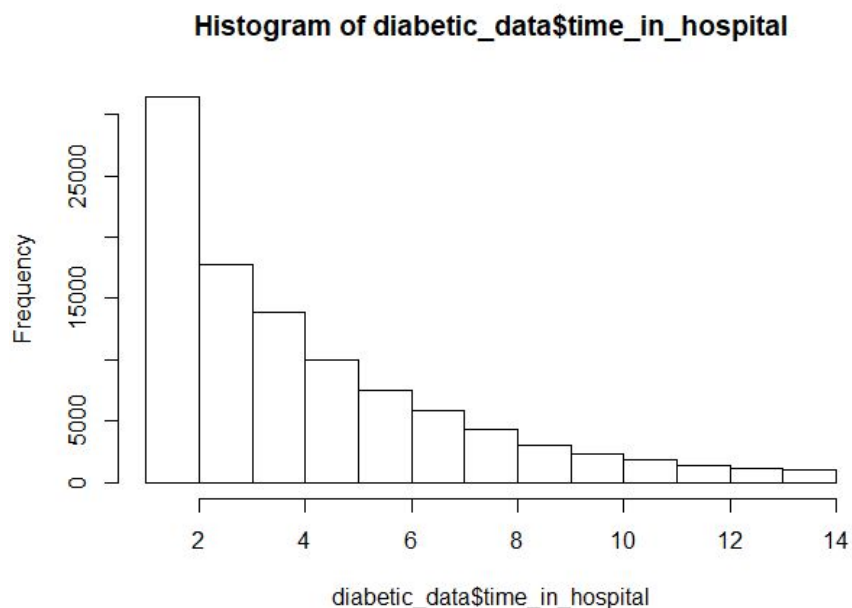
```
>mean(diabetic_data$time_in_hospital,na.rm=T)+3*sd(diabetic_data$time_in_hospital,na.rm=T)
```

```
[1] 13.35131
```

```
>max(diabetic_data$time_in_hospital, na.rm=T)
```

```
[1] 14
```

```
> hist(diabetic_data$time_in_hospital)
```



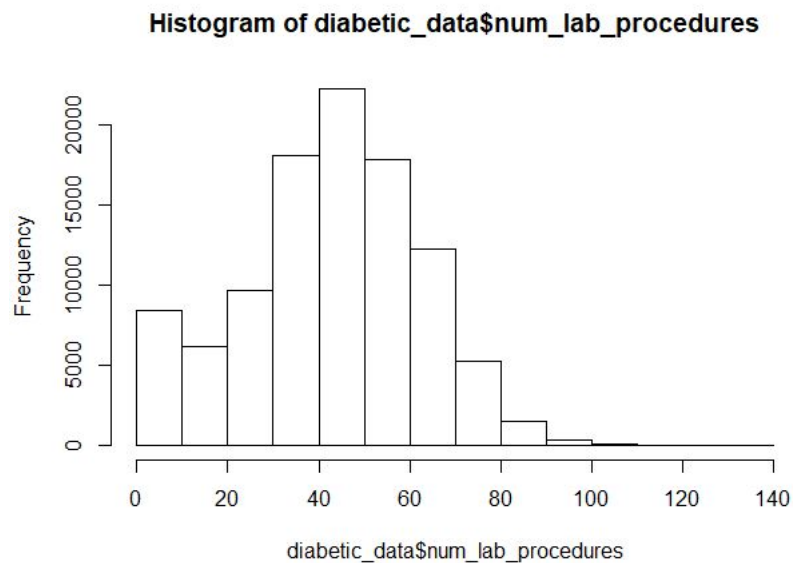
num_lab_procedures

```
>mean(diabetic_data$num_lab_procedures,na.rm=T)+3*sd(diabetic_data$num_lab_procedures,na.rm=T)
```

```
[1] 48.21306
```

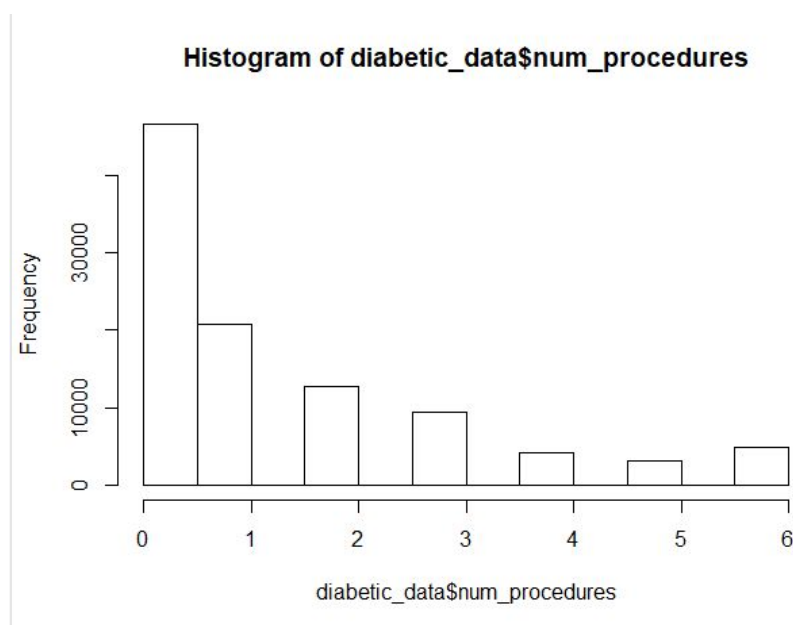
```
> max(diabetic_data$num_lab_procedures, na.rm=T)
```

```
[1] 132
```



num_procedures

```
> mean(diabetic_data$num_procedures, na.rm=T) + 3 * sd(diabetic_data$num_procedures, na.rm=T)
[1] 6.457151
> max(diabetic_data$num_procedures, na.rm=T)
[1] 6
```



number_diagnoses

Podemos observar que tanto `time_in_hospital` como `num_lab_procedures` contienen datos atípicos. En el caso de `time_in_hospital` estos datos podrían considerarse fringelier más que como outliers, debido a la cercanía con el límite. El tercer parámetro en cambio mantiene sus valores dentro de rangos esperados.

Llegados a este punto, habría que valorar si estos valores extremos se deben a errores o son valores reales.

Ya que estos datos han sido ya tratados con anterioridad y es un dataset lo suficientemente grande, se asume que estamos tratando con valores reales, por lo que se mantendrán.

Otra estrategia puede ser analizar los datos con o sin estos valores extremos y decidir el resultado correcto en función de las diferencias y tendencias obtenidas en cada uno de los casos.

El resto de atributos son atributos categóricos, para los cuales no tiene sentido aplicar este análisis. Por otra parte, tampoco se realizará análisis sobre el parámetro de salida readmitted.

Con esto el dataset quedaría listo para realizar el análisis. Para facilitar este proceso, se exportará el dataset obtenido a un nuevo fichero

```
write.csv(diabetic_data, "diabetic_data_clean.csv")
```

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

A continuación se seleccionan los grupos dentro del conjunto de datos que pueden resultar interesantes para analizar y/o comparar.

Seleccionaremos las variables categóricas de sexo, edad, etnia y primer diagnóstico, y como variables numéricas número de diagnósticos, días pasados en el hospital, número de test de laboratorio y número de procedimientos.

Las variables categóricas se convertirán en factores, puesto que actualmente son de tipo carácter.

```
#sexo
> diabetic_data$gender <- as.factor(diabetic_data$gender)
> unique(diabetic_data$gender, incomparables = FALSE)
[1] Female Male
Levels: Female Male
#edad
> diabetic_data$age <- as.factor(diabetic_data$age)
> unique(diabetic_data$age, incomparables = FALSE)
```

```

[1] [0-10) [10-20) [20-30) [30-40) [40-50) [50-60) [60-70)
[70-80)
[9] [80-90) [90-100)
10 Levels: [0-10) [10-20) [20-30) [30-40) [40-50) [50-60) [60-70) ...
[90-100)
#etnia
> diabetic_data$race <- as.factor(diabetic_data$race)
> unique(diabetic_data$race, incomparables = FALSE)
[1] Caucasian AfricanAmerican Other Asian
Hispanic
Levels: AfricanAmerican Asian Caucasian Hispanic Other
#diagnostico
> diabetic_data$diag_1 <- as.factor(diabetic_data$diag_1)
> unique(diabetic_data$diag_1)
[1] 250.83 276 648 8 197 414 428 398 434 250.7 157 518 999 410 682 402
[17] 737 572 v57 189 786 427 996 277 584 462 473 411 174 486 998 511
[33] 432 626 295 196 250.6 618 182 845 423 808 250.4 722 403 250.11 784 707
[49] 440 151 715 997 198 564 812 38 590 556 578 250.32 433 v58 569 185
[65] 536 255 250.13 599 558 574 491 560 244 250.03 577 730 188 824 250.8 332
[81] 562 291 296 510 401 263 438 70 250.02 493 642 625 571 738 593 250.42
[97] 807 456 446 575 250.41 820 515 780 250.22 995 235 250.82 721 787 162 724
[113] 282 514 v55 281 250.33 530 466 435 250.12 v53 789 566 822 191 557 733
[129] 455 711 482 202 280 553 225 154 441 250.81 349 962 592 507 386 156
[145] 200 728 348 459 426 388 607 337 82 531 596 288 656 573 492 220
[161] 516 210 922 286 885 958 661 969 250.93 227 112 404 823 532 416 346
[177] 535 453 250 595 211 303 250.01 852 218 782 540 457 285 431 340 550
[193] 54 351 601 723 555 153 443 380 204 424 241 358 694 331 345 681
[209] 447 290 158 579 436 335 309 654 805 799 292 183 78 851 458 586
[225] 311 892 305 293 415 591 794 803 79 655 429 278 658 598 729 585
[241] 444 604 727 214 552 284 680 708 41 644 481 821 413 437 968 756
[257] 632 359 275 512 781 420 368 522 294 825 135 304 320 250.31 669 868
[273] 496 250.43 826 567 3 203 53 251 565 161 495 49 250.1 297 663 576
[289] 355 850 287 250.2 611 840 350 726 537 620 180 366 783 11 751 716
[305] 250.3 199 464 580 836 664 283 813 966 289 965 184 480 608 333 972
[321] 212 117 788 924 959 621 238 785 714 942 250.23 710 47 933 508 478
[337] 844 7 736 233 42 250.5 397 395 201 421 253 250.92 600 494 977 39
[353] 659 312 614 647 652 646 274 861 425 527 451 485 217 250.53 442 970
[369] 193 160 322 581 475 623 374 582 568 465 801 237 376 150 461 913
[385] 226 617 987 641 298 790 336 362 228 513 383 746 353 911 506 873
[401] 155 860 534 802 141 v45 396 310 341 242 719 239 533 616 519 301
[417] v66 5 989 230 385 300 853 871 570 848 463 9 934 250.21 236 361
[433] 594 501 810 643 430 528 205 791 983 992 490 172 171 622 306 863
[449] 864 474 660 759 356 634 967 551 695 187 732 747 323 308 370 252
[465] 152 846 164 365 718 48 266 720 94 344 797 170 878 904 v56 882
[481] 843 709 973 454 686 939 487 229 991 483 357 692 796 693 935 936
[497] 800 920 v26 261 307 262 250.9 831 145 223 v71 839 685 v54 35 34
[513] 179 964 136 324 389 815 334 143 526 588 192 v67 394 917 88 219
[529] 325 792 717 994 990 793 207 637 195 373 847 827 31 891 814 v60
[545] 703 865 352 627 378 342 886 369 745 705 816 541 986 610 633 640
[561] 753 173 835 379 445 272 382 945 619 881 250.52 866 405 916 215 893
[577] 75 671 928 906 897 725 867 115 890 734 521 674 470 834 146 696
[593] 524 980 691 384 142 879 250.51 246 208 448 955 653 149 245 735 883
[609] 854 952 838 194 v43 163 216 147 354 27 477 318 880 921 377 471
[625] 683 175 602 250.91 982 706 375 417 131 347 870 148 862 61 817 914
[641] 360 684 314 v63 36 57 240 915 971 795 988 452 963 327 731 842
[657] v25 645 665 110 944 603 923 412 363 957 976 698 299 700 273 974
[673] 97 529 66 98 605 941 52 806 84 271 837 657 895 338 523 542
[689] 114 543 372 v70 E909 583 v07 422 615 279 500 903 919 875 381 804
[705] 704 23 58 649 832 133 975 833 391 690 10 v51
716 Levels: 10 11 110 112 114 115 117 131 133 135 136 141 142 143 145 146 147 148 149 150 151 152 153 154 155 156 ... v71

```

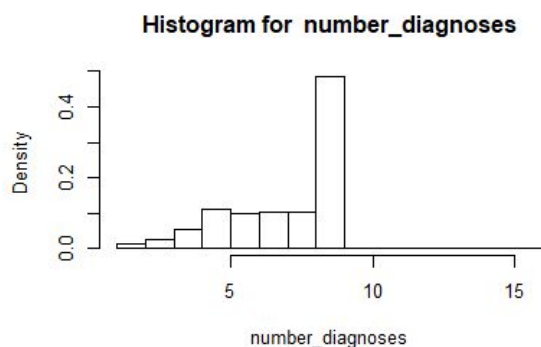
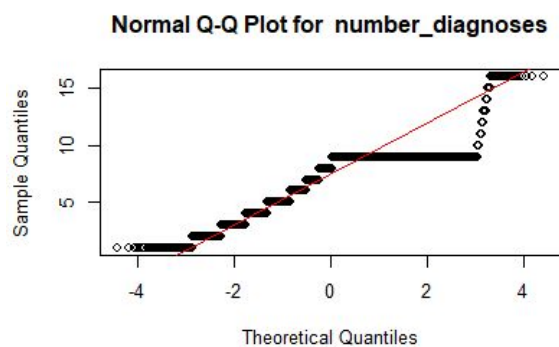
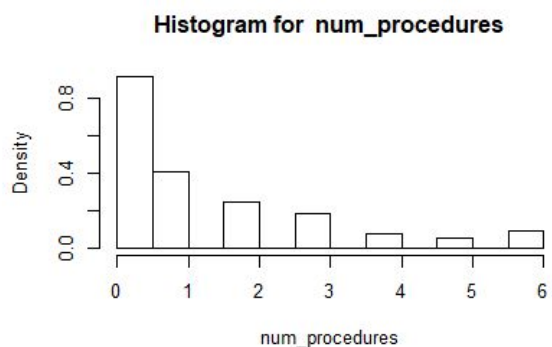
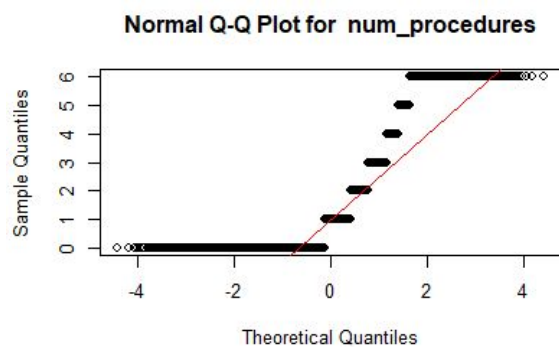
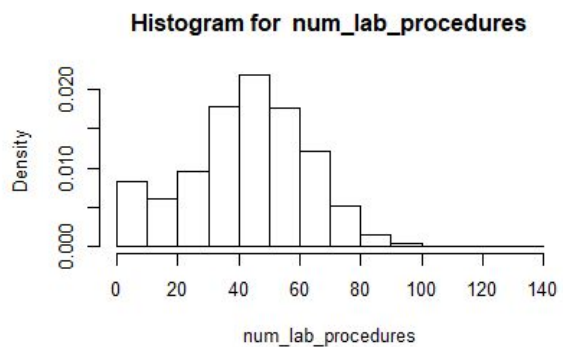
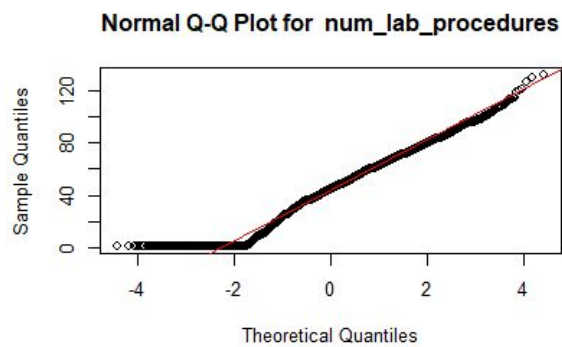
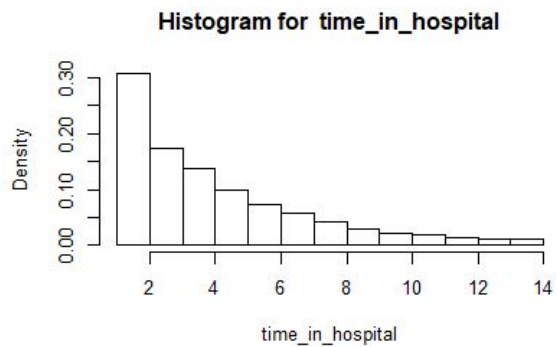
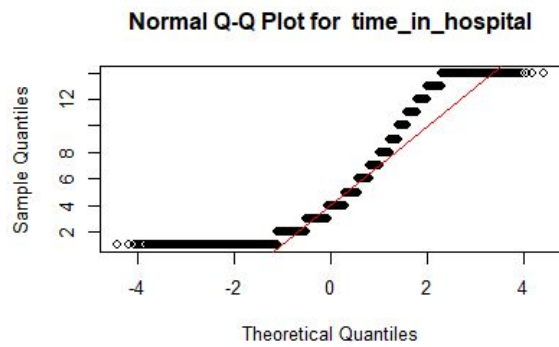
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

En este punto se comprobará si las variables cuantitativas provienen de una distribución normal.

Para comprobar la normalidad de las variables cuantitativas se utilizarán las gráficas de QQplots, ya que permiten observar la similitud entre la distribución del conjunto de datos a analizar y una distribución normal ideal.

El siguiente código mostrará el histograma y qq-plot de todas las variables numéricas

```
par(mfrow=c(2,2))
for(i in 1:ncol(diabetic_data)) {
  if (is.numeric(diabetic_data[,i])){
    qqnorm(diabetic_data[,i],main = paste("Normal Q-Q Plot for
",colnames(diabetic_data)[i]))
    qqline(diabetic_data[,i],col="red")
    hist(diabetic_data[,i],
        main=paste("Histogram for ", colnames(diabetic_data)[i]),
        xlab=colnames(diabetic_data)[i], freq = FALSE)
  }
}
```



Los resultados nos indican que la variable num_lab_procedures podría ser candidata a la normalización, pero observamos que hay diversas muestras fuera

de la recta de regresión. Para comprobar si estas variables están normalizadas, utilizaremos la prueba de Anderson-Darling. En este caso no se podría aplicar el test de Shapiro Wilk debido a que hay más de 5000 registros y queremos hacer el análisis sobre la totalidad de los datos.

Aplicaremos el test a todas las variables, comprobando que se obtiene un p-valor superior al nivel de significación prefijado $\alpha = 0,05$. Si se cumple esta condición se considera que la variable en cuestión sigue una distribución normal.

```
> library(nortest)
> alpha = 0.05
> col.names = colnames(diabetic_data)
>
> for (i in 1:ncol(diabetic_data)) {
+   if (i == 1) cat("Variables que no siguen una distribución
normal:\n")
+   if (is.integer(diabetic_data[,i]) | is.numeric(diabetic_data[,i]))
+   {
+     p_val = ad.test(diabetic_data[,i])$p.value
+     if (p_val < alpha) {
+       cat(col.names[i])
+       # Format output
+       if (i < ncol(diabetic_data) - 1) cat(", ")
+       if (i %% 3 == 0) cat("\n")
+     }
+   }
+ }
Variables que no siguen una distribución normal:
encounter_id, discharge_disposition_id, time_in_hospital,
num_lab_procedures, num_procedures, number_diagnoses,
```

Se asume que los datos no pueden normalizarse y se analizarán con pruebas no paramétricas que no presuponen esta característica. Del mismo modo, al no ser datos normales no sería necesario analizar la homogeneidad en la varianza.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Se plantean las siguientes preguntas e hipótesis a comprobar y que se irán respondiendo a lo largo de este apartado:

- ¿Cuáles son las variables que más afectan al reingreso de un paciente?
- ¿Influyen los niveles de glucosa máximos? ¿Un nivel alto aumenta la probabilidad de un reingreso?
- ¿La edad, raza y género influyen más que el número de procedimientos realizados y el resultado obtenido en los tests para un reingreso?

En primer lugar se realizará un análisis de correlación entre las distintas variables para determinar cuáles ejercen mayor influencia sobre un posible reingreso. Para ello se usará el coeficiente de correlación de Spearman, ya que estos datos no siguen una distribución normal.

```
>cor.test(diabetic_data$readmitted,diabetic_data$time_in_hospital,method="spearman")
```

Spearman's rank correlation rho

```
data: diabetic_data$readmitted and diabetic_data$time_in_hospital
S = 1.6493e+14, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.06102197
```

Warning message:

```
In cor.test.default(diabetic_data$readmitted,
diabetic_data$time_in_hospital, :
Cannot compute exact p-value with ties
```

Parece que existen muestras con los mismos valores o ties. Por ellos, se utilizará el coeficiente tau-b de Kendall, ya que está adaptado para trabajar con ties. Consideraremos la hipótesis nula como que dos variables no están relacionadas entre sí cuando su índice de correlación es 0.

```
>cor(diabetic_data$readmitted,diabetic_data$number_diagnoses,method="kendall", use="pairwise")
[1] 0.09857357
```

Para comprobar que no están correlacionadas, se comprueba la hipótesis nula con la función `cor.test`.

```
>cor.test(diabetic_data$readmitted,diabetic_data$number_diagnoses,method="kendall", use="pairwise")
```

Kendall's rank correlation tau

```
data: diabetic_data$readmitted and diabetic_data$number_diagnoses
z = 34.733, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.09857357
```

Debido a que p-value es menor que 0,05, se rechaza la hipótesis nula y se afirma que existe correlación entre ambas variables. Realizamos el mismo proceso con el resto de variables:

```
>cor(diabetic_data$readmitted,diabetic_data$time_in_hospital,method="kendall", use="pairwise")
[1] 0.05264863
```

```
>cor.test(diabetic_data$readmitted,diabetic_data$time_in_hospital,method="kendall", use="pairwise")
```

Kendall's rank correlation tau

```
data: diabetic_data$readmitted and diabetic_data$time_in_hospital
z = 19.466, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.05264863
```

```
>cor(diabetic_data$readmitted,diabetic_data$num_lab_procedures,method="kendall", use="pairwise")
[1] 0.03369054
>cor.test(diabetic_data$readmitted,diabetic_data$num_lab_procedures,method="kendall", use="pairwise")
```

Kendall's rank correlation tau

```
data: diabetic_data$readmitted and diabetic_data$num_lab_procedures
```

```

z = 13.061, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.03369054
>cor(diabetic_data$readmitted,diabetic_data$num_procedures,method="kendall", use="pairwise")
[1] -0.04217783
>cor.test(diabetic_data$readmitted,diabetic_data$num_procedures,method="kendall", use="pairwise")

```

Kendall's rank correlation tau

```

data: diabetic_data$readmitted and diabetic_data$num_procedures
z = -14.794, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
-0.04217783
.

```

Como p-value es inferior a 0,05 en todos los casos, podemos afirmar que existe correlación entre estas variables y la variable de output. Aun así, este factor no es muy elevado. Esto puede ser debido al gran número de variables que teníamos en el dataset, lo que hace que el factor de correlación se reparta entre ellas, haciendo que no exista una única variable muy significativa

Para la variable gender, change y diabetesMed, al ser datos dicotómicos, se aplica el test de Wilcoxon.

```
> wilcox.test(diabetic_data$readmitted~diabetic_data$gender)
```

Wilcoxon rank sum test with continuity correction

```

data: diabetic_data$readmitted by diabetic_data$gender
W = 1310400000, p-value = 8.757e-09
alternative hypothesis: true location shift is not equal to 0

```

```
> wilcox.test(diabetic_data$readmitted~diabetic_data$change)
```

Wilcoxon rank sum test with continuity correction

```

data: diabetic_data$readmitted by diabetic_data$change
W = 1346200000, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

```



```
> wilcox.test(diabetic_data$readmitted~diabetic_data$diabetesMed)
```

```
Wilcoxon rank sum test with continuity correction
```

```
data: diabetic_data$readmitted by diabetic_data$diabetesMed
W = 850160000, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Estas variables muestran que hay diferencias estadísticamente significativas entre los distintos grupos, ya que p-value es menor de 0,05. Esto implica que existen diferencias entre hombres y mujeres, si existe o no un cambio en la medicación y si se realizó una prescripción de medicación.

En el caso de que existan tres o más grupos de datos, como ocurre en el atributo de etnia, edad, max_glu_serum o A1Cresult, es posible aplicar el test de Kruskal-Wallis.

```
> kruskal.test(diabetic_data$readmitted~diabetic_data$race)
```

```
Kruskal-Wallis rank sum test
```

```
data: diabetic_data$readmitted by diabetic_data$race
Kruskal-Wallis chi-squared = 81.943, df = 4, p-value < 2.2e-16
```

```
> diabetic_data$max_glu_serum <- as.factor(diabetic_data$max_glu_serum)
> kruskal.test(diabetic_data$readmitted~diabetic_data$max_glu_serum)
```

```
Kruskal-Wallis rank sum test
```

```
data: diabetic_data$readmitted by diabetic_data$max_glu_serum
Kruskal-Wallis chi-squared = 49.147, df = 3, p-value = 1.214e-10
```

```
> diabetic_data$A1Cresult <- as.factor(diabetic_data$A1Cresult)
> kruskal.test(diabetic_data$readmitted~diabetic_data$A1Cresult)
```

```
Kruskal-Wallis rank sum test
```

```
data: diabetic_data$readmitted by diabetic_data$A1Cresult
Kruskal-Wallis chi-squared = 53.504, df = 3, p-value = 1.432e-11
```

En este caso se observa que los grupos son diferentes estadísticamente entre ellos al obtener p-values menores de 0,05.

Por último, vamos a intentar crear un modelo predictivo. En este caso, nos encontramos ante un problema de regresión lógica, ya que la variable de output puede tomar dos valores, reingreso (1) o no (0).

Primero, retiramos la columna de `enconter_id`, ya que no aporta información relevante. Tras varios intentos, se valora eliminar también las columnas de diagnósticos, ya que al ser una variable categórica con demasiadas variables estaba aportando demasiado ruido a la estimación y no se conseguían resultados claros.

De esta forma se mantienen las variables de etnia, género, edad, tiempo de hospitalización, test de laboratorio, número de procedimientos, número de diagnósticos, máximo nivel de glucosa, resultado del test A1C, cambios en la medicación y receta de medicación para diabetes

```
train <- diabetic_data[1:80000,]
test  <- diabetic_data[80001:88900,]

model <- glm(readmitted ~.,family=binomial(link='logit'),data=train)

summary(model)
```

Call:

```
glm(formula = readmitted ~ ., family = binomial(link = "logit"),
    data = train)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.5040	-1.1359	-0.8611	1.1652	1.9558

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-2.2362480	0.2241741	-9.975	< 2e-16	***
raceAsian	-0.1652015	0.1039637	-1.589	0.112053	
raceCaucasian	0.0172183	0.0182227	0.945	0.344719	
raceHispanic	0.0384147	0.0550101	0.698	0.484977	
raceOther	-0.2402845	0.0669462	-3.589	0.000332	***
genderMale	-0.0534123	0.0145782	-3.664	0.000248	***
age[10-20)	0.9047973	0.2269889	3.986	6.72e-05	***
age[20-30)	0.9745714	0.2182156	4.466	7.97e-06	***
age[30-40)	0.8224237	0.2144308	3.835	0.000125	***
age[40-50)	0.8473229	0.2126427	3.985	6.76e-05	***
age[50-60)	0.7996060	0.2122068	3.768	0.000165	***
age[60-70)	0.8625417	0.2121521	4.066	4.79e-05	***
age[70-80)	0.8871180	0.2121133	4.182	2.89e-05	***

```

age[80-90)          0.8275724  0.2124530   3.895 9.81e-05 ***
age[90-100)         0.4664176  0.2164881   2.154 0.031203 *
time_in_hospital    0.0133437  0.0026091   5.114 3.15e-07 ***
num_lab_procedures  0.0016907  0.0004151   4.073 4.65e-05 ***
num_procedures      -0.0749116  0.0044839 -16.707 < 2e-16 ***
number_diagnoses     0.1360320  0.0039753  34.219 < 2e-16 ***
max_glu_serum>300    0.2019684  0.0804309   2.511 0.012036 *
max_glu_serumNone   -0.0143638  0.0547166  -0.263 0.792926
max_glu_serumNorm   -0.0419310  0.0673873  -0.622 0.533784
A1Cresult>8         0.0997309  0.0460380   2.166 0.030290 *
A1CresultNone       0.1632674  0.0392772   4.157 3.23e-05 ***
A1CresultNorm       -0.0899389  0.0507774  -1.771 0.076521 .
changeNo            -0.0476874  0.0168136  -2.836 0.004565 **
diabetesMedYes       0.2567239  0.0196697  13.052 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

(Dispersion parameter for binomial family taken to be 1)

```

Null deviance: 110737 on 79999 degrees of freedom
Residual deviance: 108459 on 79973 degrees of freedom
AIC: 108513

```

Number of Fisher Scoring iterations: 4

Ahora podemos analizar los resultados obtenidos del modelo. Podemos afirmar que la variable de glucosa no es estadísticamente relevante en comparación al resto de variables.

Con respecto a las variables estadísticamente relevantes vemos que el número de procedimientos, número de diagnósticos y el cambio en la medicación tienen los p-valores más bajos, con lo que podemos asociar una fuerte correlación entre estas variables y el reingreso del paciente. Curiosamente, los datos de edad, genero o etnia tienen menos influencia.

Los valores positivos indican que influyen de forma positiva en un ingreso, mientras que los valores negativos indican que, siendo el resto de variables iguales, el número de procedimientos reduce en un 0,07 la posibilidad de un reingreso.

Vamos a ver el modelo de nuevo pero seleccionando algunas de estas variables más significativas.

```

> model2 <- glm(readmitted ~
(age+num_procedures+number_diagnoses+diabetesMed),family=binomial(link='
logit'),data=train)

```

```
> summary(model2)
```

Call:

```
glm(formula = readmitted ~ (age + num_procedures + number_diagnoses +  
  diabetesMed), family = binomial(link = "logit"), data = train)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.3789	-1.1385	-0.8718	1.1656	1.9333

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-2.133714	0.211157	-10.105	< 2e-16	***
age[10-20)	0.920297	0.226756	4.059	4.94e-05	***
age[20-30)	1.022388	0.217784	4.695	2.67e-06	***
age[30-40)	0.857276	0.213966	4.007	6.16e-05	***
age[40-50)	0.876296	0.212150	4.131	3.62e-05	***
age[50-60)	0.827343	0.211712	3.908	9.31e-05	***
age[60-70)	0.895913	0.211630	4.233	2.30e-05	***
age[70-80)	0.929164	0.211571	4.392	1.12e-05	***
age[80-90)	0.878016	0.211905	4.143	3.42e-05	***
age[90-100)	0.525925	0.215917	2.436	0.0149	*
num_procedures	-0.069376	0.004344	-15.971	< 2e-16	***
number_diagnoses	0.142612	0.003858	36.968	< 2e-16	***
diabetesMedYes	0.289934	0.016992	17.063	< 2e-16	***

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 110737 on 79999 degrees of freedom
Residual deviance: 108632 on 79987 degrees of freedom
AIC: 108658

Number of Fisher Scoring iterations: 4

Para comprobar la validez del modelo, se procede a realizar predicciones con los datos de test.

```
> fitted <- predict(model2,newdata=test,type='response')  
> fitted <- ifelse(fitted > 0.5,1,0)  
> Error <- mean(fitted != test$readmitted)  
> print(paste('Accuracy',1-Error))  
[1] "Accuracy 0.541573033707865"
```

Comprobamos que la precisión del modelo no es muy alta, a pesar de escoger las variables más influyentes. Al realizar las estimaciones con todos los parámetros se obtiene una precisión mayor

```
[1] "Accuracy 0.552921348314607"
```

Con lo cual es posible que aumentando el número de variables en juego sea posible aumentar la precisión de este modelo, ya que hemos reducido las variables a tratar de 47 a únicamente 4. Debido al alcance de la práctica, este supuesto se plantea como una mejora a realizar.

5. Representación de los resultados a partir de tablas y gráficas.

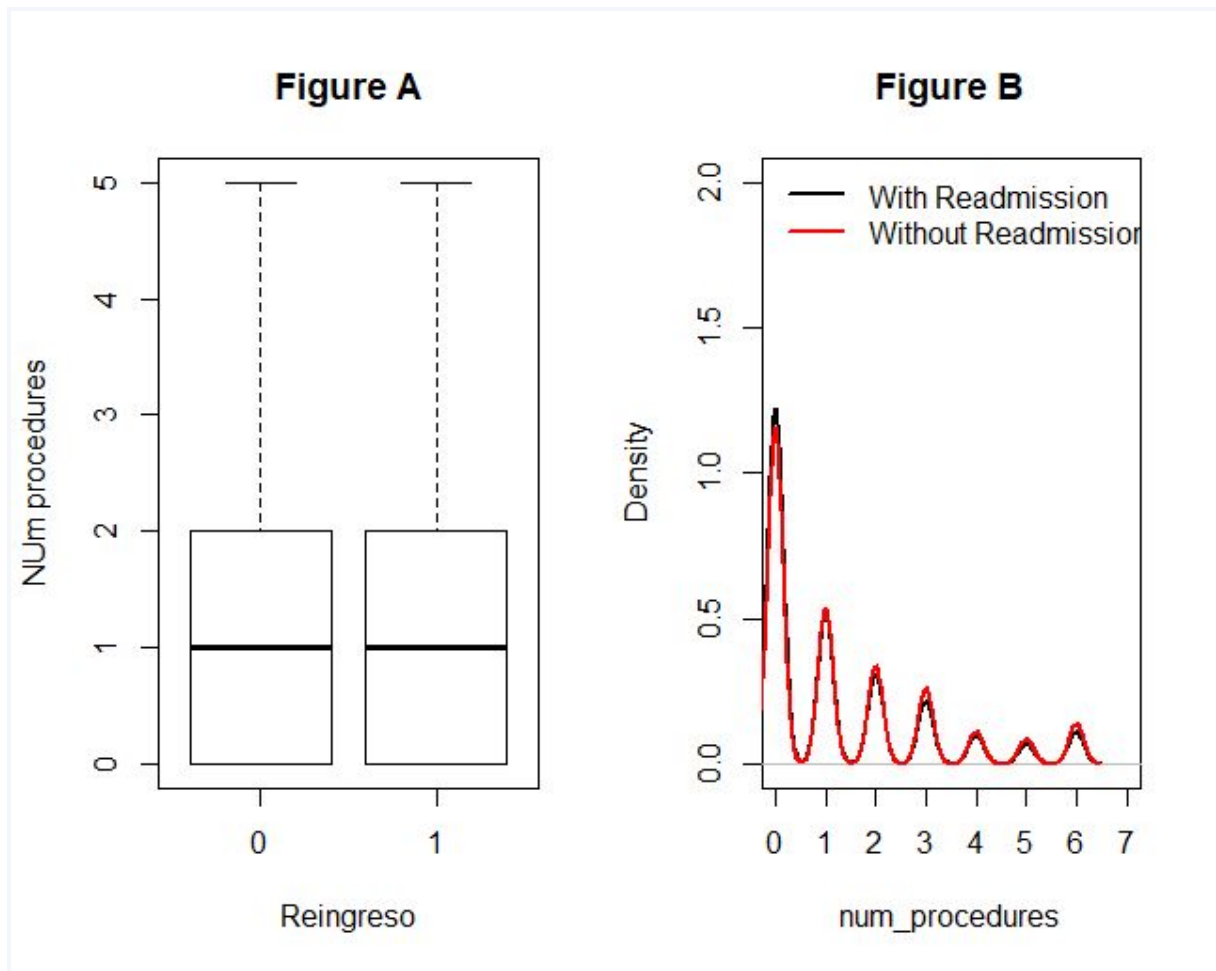
Vamos a analizar gráficamente los resultados de correlación obtenidos. Una forma de hacerlo es observar cómo se ven afectadas las variables teniendo en cuenta que la persona haya sido reingresada o no. Analizaremos el número de procedimientos y de diagnósticos, ya que son dos de las variables más correlacionadas en base a los resultados obtenidos.

```
par(mfrow = c(1, 2))

with(diabetic_data, boxplot(num_procedures ~ readmitted,
                             ylab = "Num procedures",
                             xlab = "Reingreso",
                             main = "Figure A",
                             outline = FALSE))

with <- diabetic_data[diabetic_data$readmitted == 1, ]
without <- diabetic_data[diabetic_data$readmitted == 0, ]

plot(density(with$num_procedures),
     xlim = c(0, 7),
     ylim = c(0, 2),
     xlab = "num_procedures",
     main = "Figure B",
     lwd = 2)
lines(density(without$num_procedures),
     col = "red",
     lwd = 2)
legend("topleft",
     col = c("black", "red"),
     legend = c("With Readmission", "Without Readmission"),
     lwd = 2,
     bty = "n")
```



En este caso, vemos que al readmisiones son ligeramente superiores cuando no se realiza ningún procedimiento. En cambio, a partir de un procedimiento realizado siempre es mayor la densidad de no readmisiones que de readmisiones, siendo más significativas las diferencias con 3 o 6 procedimientos. Esta situación podría explicarse afirmando que cuantas más pruebas se realicen sobre el paciente, con más certeza puede emitirse un diagnóstico adecuado y evitar una readmisión.

```
par(mfrow = c(1, 2))
```

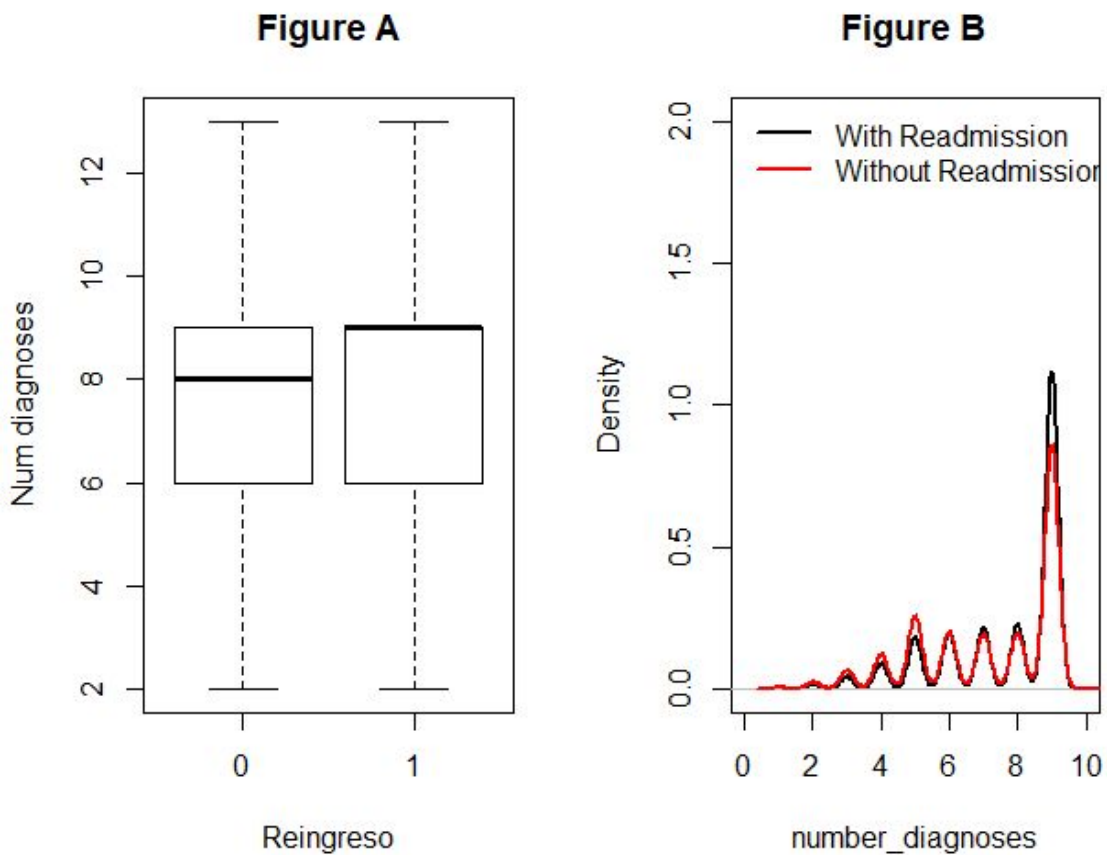
```
with(diabetic_data, boxplot(number_diagnoses ~ readmitted,
                             ylab = "Num diagnoses",
                             xlab = "Reingreso",
                             main = "Figure A",
                             outline = FALSE))
```

```
with <- diabetic_data[diabetic_data$readmitted == 1, ]
without <- diabetic_data[diabetic_data$readmitted == 0, ]
```

```

plot(density(with$number_diagnoses),
     xlim = c(0, 10),
     ylim = c(0, 2),
     xlab = "number_diagnoses",
     main = "Figure B",
     lwd = 2)
lines(density(without$number_diagnoses),
      col = "red",
      lwd = 2)
legend("topleft",
      col = c("black", "red"),
      legend = c("With Readmission", "Without Readmission"),
      lwd = 2,
      bty = "n")

```



En este caso, vemos que el número de readmisiones es menor que el de no admisiones entre 1 y 6 diagnósticos. Más de 6 diagnósticos derivan en un mayor número de readmisiones, siendo especialmente significativo para 9 diagnósticos. Esto puede explicarse si asumimos que aquellos casos con mayor número de diagnósticos son casos complejos, que requieren de múltiples opiniones y resulta complicado llegar a un diagnóstico consensuado.

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

De los resultados observamos que las características físicas, véase género, edad o etnia, no influyen tanto en una futura readmisión como los procedimientos y test que puedan realizarse sobre el paciente durante su ingreso para obtener el diagnóstico correcto. Igualmente también se ha observado que tampoco depende de sus valores de glucosa, a pesar de aparentemente ser un factor de riesgo.

Aun así, se ha observado al realizar la construcción del modelo que no ha sido posible conseguir una buena precisión al reducir las variables. Esto nos indica que determinar una readmisión es un proceso complejo, donde haría falta un análisis más extenso y detallado con el fin de localizar qué otras variables del conjunto de las descartadas tienen una fuerte correlación con la variable de output e incluirlas para mejorar esta precisión.

7. Código

```
directory <- getwd()
diabetic_data <- read.csv(file.path(directory,"diabetic_data.csv"),
header=TRUE, na.strings = c("", "NA"), stringsAsFactors=FALSE)
diabetic_data <- subset(diabetic_data,select=c(3,4,5,8,10,13, 14, 19,
20, 21, 22, 23, 24, 48, 49, 50))
str(diabetic_data)
sapply(diabetic_data, function(x) class(x))

sapply(diabetic_data, function(x) sum(is.na(x)))

#Instalacion del paquete VIM
>install.packages("VIM")
# Imputación de valores mediante la función kNN() del paquete VIM

suppressWarnings(suppressMessages(library(VIM)))

diabetic_data$race <- kNN(diabetic_data)$race
diabetic_data$diag_1 <- kNN(diabetic_data)$diag_1
diabetic_data$gender <- kNN(diabetic_data)$gender
```



```

sapply(diabetic_data, function(x) sum(is.na(x)))

mean(diabetic_data$time_in_hospital, na.rm=T)+3*sd(diabetic_data$time_in_hospital, na.rm=T)
max(diabetic_data$time_in_hospital, na.rm=T)
hist(diabetic_data$time_in_hospital)

mean(diabetic_data$num_lab_procedures, na.rm=T)+3*sd(diabetic_data$num_procedures, na.rm=T)
max(diabetic_data$num_lab_procedures, na.rm=T)
hist(diabetic_data$num_lab_procedures)

mean(diabetic_data$num_procedures, na.rm=T)+3*sd(diabetic_data$num_procedures, na.rm=T)
max(diabetic_data$num_procedures, na.rm=T)
hist(diabetic_data$num_procedures)

write.csv(diabetic_data, "diabetic_data_clean.csv")

#sexo
diabetic_data$gender <- as.factor(diabetic_data$gender)
#edad
diabetic_data$age <- as.factor(diabetic_data$age)
#etnia
diabetic_data$race <- as.factor(diabetic_data$race)
#diagnostico
diabetic_data$diag_1 <- as.factor(diabetic_data$diag_1)

#histograma y qqplot
par(mfrow=c(2,2))
for(i in 1:ncol(diabetic_data)) {
  if (is.numeric(diabetic_data[,i])){
    qqnorm(diabetic_data[,i], main = paste("Normal Q-Q Plot for ", colnames(diabetic_data)[i]))
    qqline(diabetic_data[,i], col="red")
    hist(diabetic_data[,i],
         main=paste("Histogram for ", colnames(diabetic_data)[i]),
         xlab=colnames(diabetic_data)[i], freq = FALSE)
  }
}

#normalidad en las variables

```

```

library(nortest)
alpha = 0.05
col.names = colnames(diabetic_data)

for (i in 1:ncol(diabetic_data)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(diabetic_data[,i]) | is.numeric(diabetic_data[,i])) {
    p_val = ad.test(diabetic_data[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if (i < ncol(diabetic_data) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}

#correlacion con test de spearman
cor.test(diabetic_data$readmitted,diabetic_data$time_in_hospital,method=
"spearman")

#coeficiente tau-b
cor(diabetic_data$readmitted,diabetic_data$number_diagnoses,method="kend
all", use="pairwise")
cor.test(diabetic_data$readmitted,diabetic_data$number_diagnoses,method=
"kendall", use="pairwise")

cor(diabetic_data$readmitted,diabetic_data$time_in_hospital,method="kend
all", use="pairwise")
cor.test(diabetic_data$readmitted,diabetic_data$time_in_hospital,method=
"kendall", use="pairwise")

cor(diabetic_data$readmitted,diabetic_data$num_lab_procedures,method="ke
ndall", use="pairwise")
cor.test(diabetic_data$readmitted,diabetic_data$num_lab_procedures,metho
d="kendall", use="pairwise")

cor(diabetic_data$readmitted,diabetic_data$num_procedures,method="kendal
l", use="pairwise")
cor.test(diabetic_data$readmitted,diabetic_data$num_procedures,method="k
endall", use="pairwise")

#wilcoxon para datos dicotomicos
wilcox.test(diabetic_data$readmitted~diabetic_data$gender)

```

```

wilcox.test(diabetic_data$readmitted~diabetic_data$change)
wilcox.test(diabetic_data$readmitted~diabetic_data$diabetesMed)

#kruskal-wallis para datos categoricos
kruskal.test(diabetic_data$readmitted~diabetic_data$race)
diabetic_data$max_glu_serum <- as.factor(diabetic_data$max_glu_serum)
kruskal.test(diabetic_data$readmitted~diabetic_data$max_glu_serum)
diabetic_data$A1Cresult <- as.factor(diabetic_data$A1Cresult)
kruskal.test(diabetic_data$readmitted~diabetic_data$A1Cresult)

#modelo de regresion lógica
train <- diabetic_data[1:80000,]
test <- diabetic_data[80001:88900,]

model <- glm(readmitted ~.,family=binomial(link='logit'),data=train)
summary(model)

model2<- glm(readmitted ~
(age+num_procedures+number_diagnoses+diabetesMed),family=binomial(link='
logit'),data=train)
summary(model2)

#precision del modelo
fitted <- predict(model2,newdata=test,type='response')
fitted <- ifelse(fitted > 0.5,1,0)
Error <- mean(fitted != test$readmitted)
print(paste('Accuracy',1-Error))

#representacion grafica
par(mfrow = c(1, 2))

with(diabetic_data, boxplot(num_procedures ~ readmitted,
                           ylab = "Num procedures",
                           xlab = "Reingreso",
                           main = "Figure A",
                           outline = FALSE))

with <- diabetic_data[diabetic_data$readmitted == 1, ]
without <- diabetic_data[diabetic_data$readmitted == 0, ]

plot(density(with$num_procedures),
     xlim = c(0, 7),
     ylim = c(0, 2),
     xlab = "num_procedures",
     main = "Figure B",

```

```

      lwd = 2)
lines(density(without$num_procedures),
      col = "red",
      lwd = 2)
legend("topleft",
      col = c("black", "red"),
      legend = c("With Readmission", "Without Readmission"),
      lwd = 2,
      bty = "n")

par(mfrow = c(1, 2))

with(diabetic_data, boxplot(number_diagnoses ~ readmitted,
                           ylab = "Num diagnoses",
                           xlab = "Reingreso",
                           main = "Figure A",
                           outline = FALSE))

with <- diabetic_data[diabetic_data$readmitted == 1, ]
without <- diabetic_data[diabetic_data$readmitted == 0, ]

plot(density(with$number_diagnoses),
     xlim = c(0, 10),
     ylim = c(0, 2),
     xlab = "number_diagnoses",
     main = "Figure B",
     lwd = 2)
lines(density(without$number_diagnoses),
     col = "red",
     lwd = 2)
legend("topleft",
     col = c("black", "red"),
     legend = c("With Readmission", "Without Readmission"),
     lwd = 2,
     bty = "n")

```

Contribuciones	Firma
Investigación previa	IFM
Redacción de las respuestas	IFM
Desarrollo del código	IFM