|  |  |
| --- | --- |
| Introducción a la ciencia de datos | Trabajo Final Teórico/Práctico  Francisco Pérez Hernández |

Tabla de contenido

[1 Introducción 3](#_Toc471759884)

[2 Descripción del tipo de datos de entrada 3](#_Toc471759885)

[2.1 Baseball 3](#_Toc471759886)

[2.1.1 Cálculo de media, desviación estándar, etc. 4](#_Toc471759887)

[2.1.2 Gráficos que permitan visualizar los datos adecuadamente. 6](#_Toc471759888)

[2.1.3 Descripción del conjunto de datos a partir de los puntos anteriores. 8](#_Toc471759889)

[2.2 Australian 8](#_Toc471759890)

[2.2.1 Cálculo de media, desviación estándar, etc. 8](#_Toc471759891)

[2.2.2 Gráficos que permitan visualizar los datos adecuadamente. 9](#_Toc471759892)

[2.2.3 Descripción del conjunto de datos a partir de los puntos anteriores. 11](#_Toc471759893)

[3 Regresión 12](#_Toc471759894)

[3.1 Regresión simple 12](#_Toc471759895)

[3.2 Regresión múltiple 13](#_Toc471759896)

[3.3 k-NN para regresión 14](#_Toc471759897)

[3.4 Comparación de algoritmos 16](#_Toc471759898)

[4 Clasificación 18](#_Toc471759899)

[4.1 k-NN 18](#_Toc471759900)

[4.2 LDA 19](#_Toc471759901)

[4.3 QDA 19](#_Toc471759902)

[4.4 Comparación de algoritmos 20](#_Toc471759903)

# 1 Introducción

Las bases de datos con las que he trabajado han sido “baseball” para regresión y “australian” para clasificación:

**Dataset Regresión: Baseball**

**Dataset Clasificación: Australian**

Voy a desarrollar lo explicado en clase sobre estos dataset y a analizar los resultados obtenidos. En la primera sección se analizarán las bases de datos asignadas donde se conocerán las estructuras y cuál es el objetivo de cada una de ellas. Esta información se ha extraído de la web Keel, junto a sus archivos asociados, los cuales los he conseguido de los siguientes enlaces:

**Baseball:** [**http://sci2s.ugr.es/keel/dataset.php?cod=76#sub1**](http://sci2s.ugr.es/keel/dataset.php?cod=76#sub1)

**Australian:** [**http://sci2s.ugr.es/keel/dataset.php?cod=53#sub1**](http://sci2s.ugr.es/keel/dataset.php?cod=53#sub1)

De forma que se ha podido extender la información facilitada en los documentos subidos por los profesores.

En la segunda sección se realiza un estudio de regresión sobre el dataset “baseball”, donde buscaré modelos que expliquen el problema. Compararé estos modelos y sus ajustes para encontrar el mejor. Finalmente realizaré el test de Wilcoxon y Friedman para los distintos modelos.

En la tercera sección realizaré un estudio similar a regresión, pero esta vez será sobre el dataset “australian” y es un problema de clasificación.

# 2 Descripción del tipo de datos de entrada

## 2.1 Baseball

El dataset Baseball contiene 17 variables/columnas de las que 16 son entradas y 1 salida. Las entradas son: "Batting\_average", "On-base\_percentage", "Runs", "Hits", "Doubles", "Triples", "HomeRuns", "Runs\_batted\_in", "Walks", "Strike-Outs", "Stolen\_bases", "Errors", "Free\_agency\_eligibility", "Free\_agent", "Arbitration\_eligibility", “Arbitration”. La salida es “Salary”.

Tenemos en total 337 instancias/observaciones y en ninguna hay valores perdidos. Son datos reales, y los tipos de cada variable son 2 reales y 14 enteros. Las dos variables reales son “Batting\_average” y “On-base\_percentage”.

Como podemos ver en la web, tenemos que, de las 14 variables de números enteros, tenemos 4 variables que solo contiene valores de 0 o 1: “Free\_agency\_eligibility”, “Free\_agent”, “Arbitration\_elegibility” y “Arbitration”.

Como información adicional tenemos que: “El dataset contiene los salarios de 1992 de un conjunto de jugadores de la “Major League Baseball” que jugaron al menos un partido en la temporada 1991 y 1992, excluyendo a los pitchers (lanzadores). Para cada jugador, se proporcionan algunas medidas de rendimiento junto con cuatro variables categóricas que indican cuán libre era cada uno para formar parte de otros equipos.”

### 2.1.1 Cálculo de media, desviación estándar, etc.

Para obtener la información la distribución de los datos, su media, valores mínimos, máximos, cuartiles… se ha hecho la función *summary():*

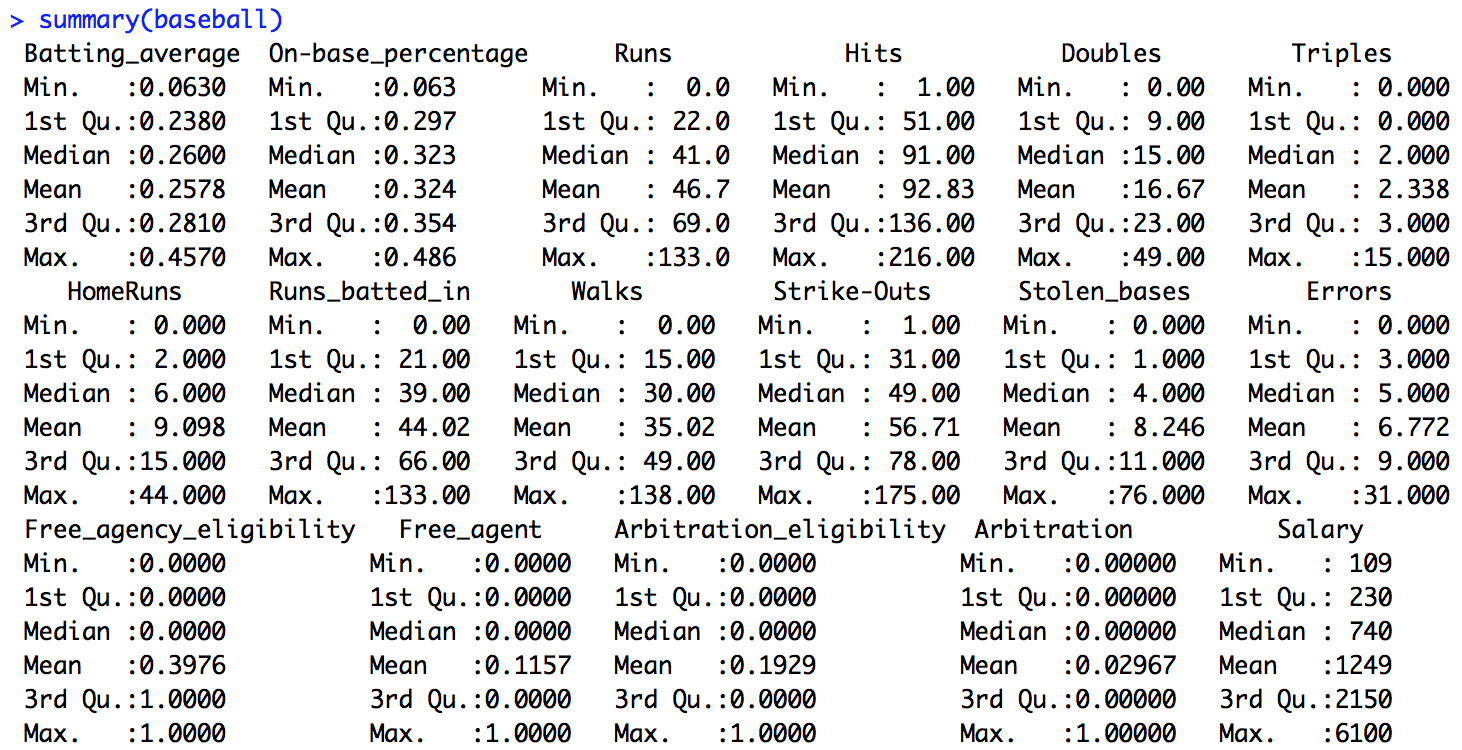


Figura 1: Resumen de distribución de los datos en cada variable para el dataset “baseball”

Si queremos la desviación estándar, con *apply()* podemos obtener la información:

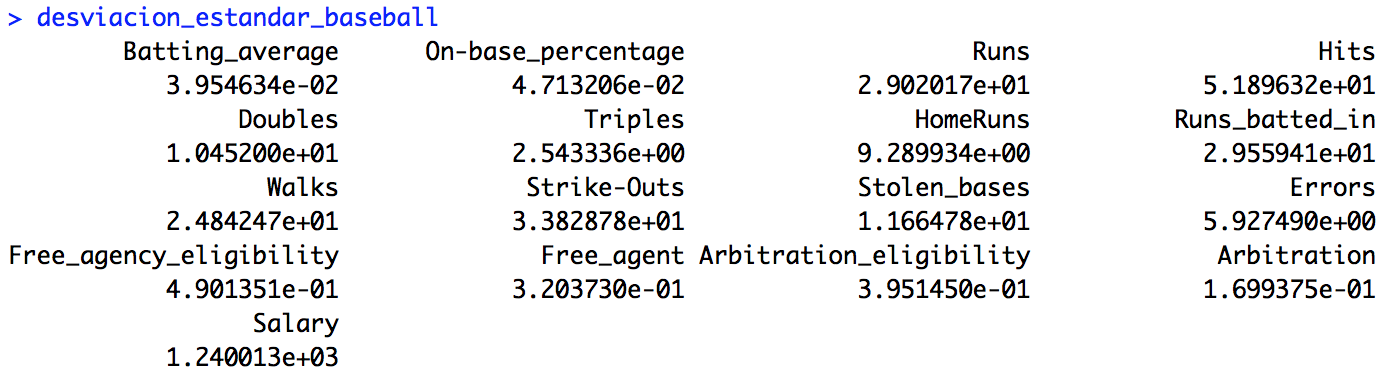


Figura 2: Desviación estándar para el dataset “baseball”

Para la varianza:



Figura 3: Varianza para el dataset “baseball”

Para la desviación absoluta de la mediana:

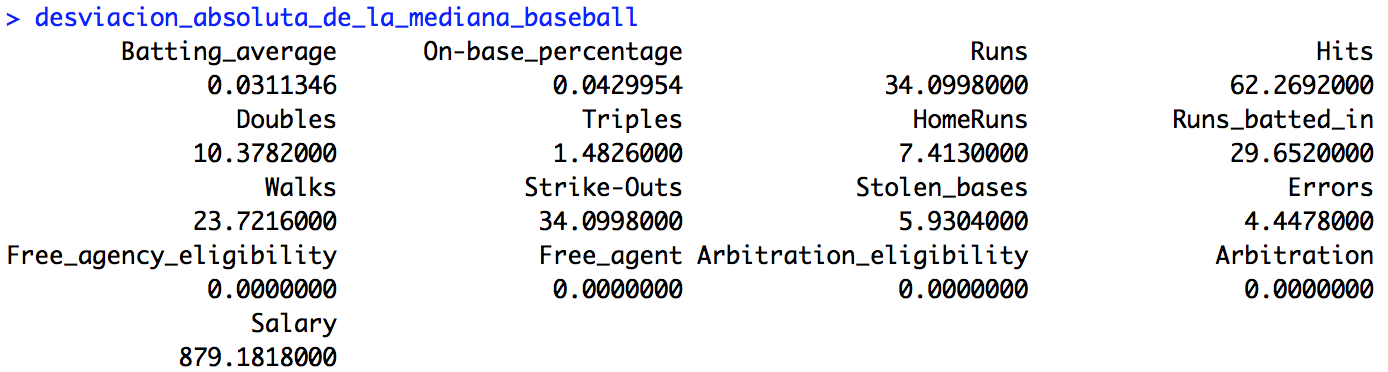


Figura 4: Desviación absoluta de la mediana para el dataset “baseball”

Para el rango intercuartil:

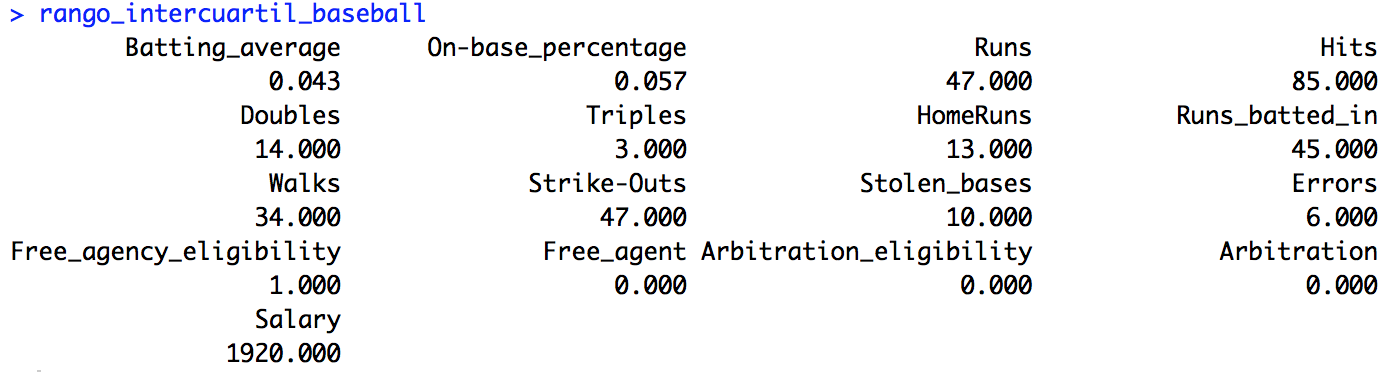


Figura 5: Rango intercuartil para el dataset “baseball”

### 2.1.2 Gráficos que permitan visualizar los datos adecuadamente.

Si queremos ver las gráficas de todas las variables con respecto a todas, podemos hacer uso de la función *plot()*



Figura 6: Plot de todas las variables del dataset “baseball”

Pero si lo que queremos es ver todas las variables con respecto de la variable de salida, es decir *Salary,* entonces tendremos:

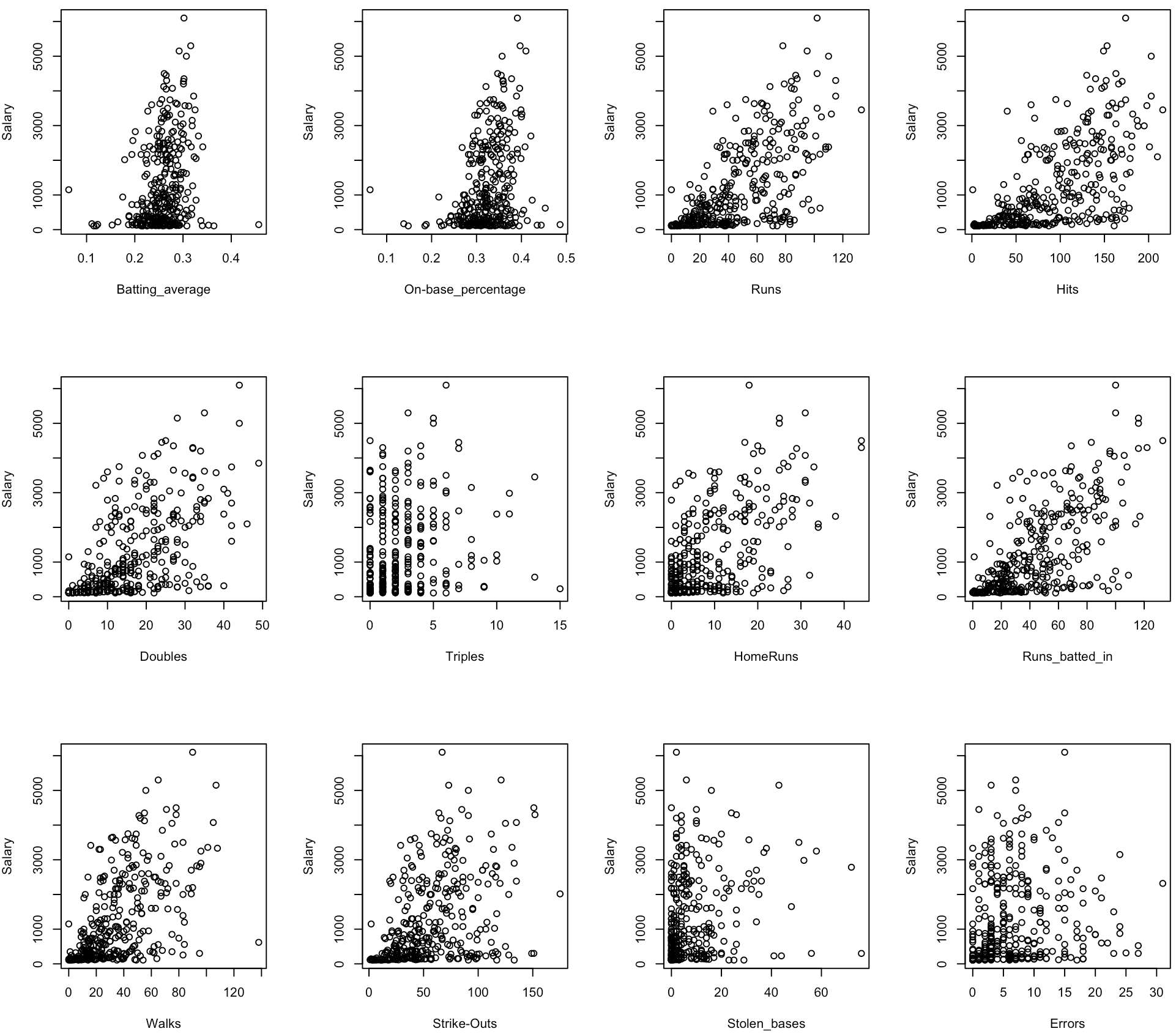


Figura 7: Plot de las primeras 12 variables

Además, tendremos:

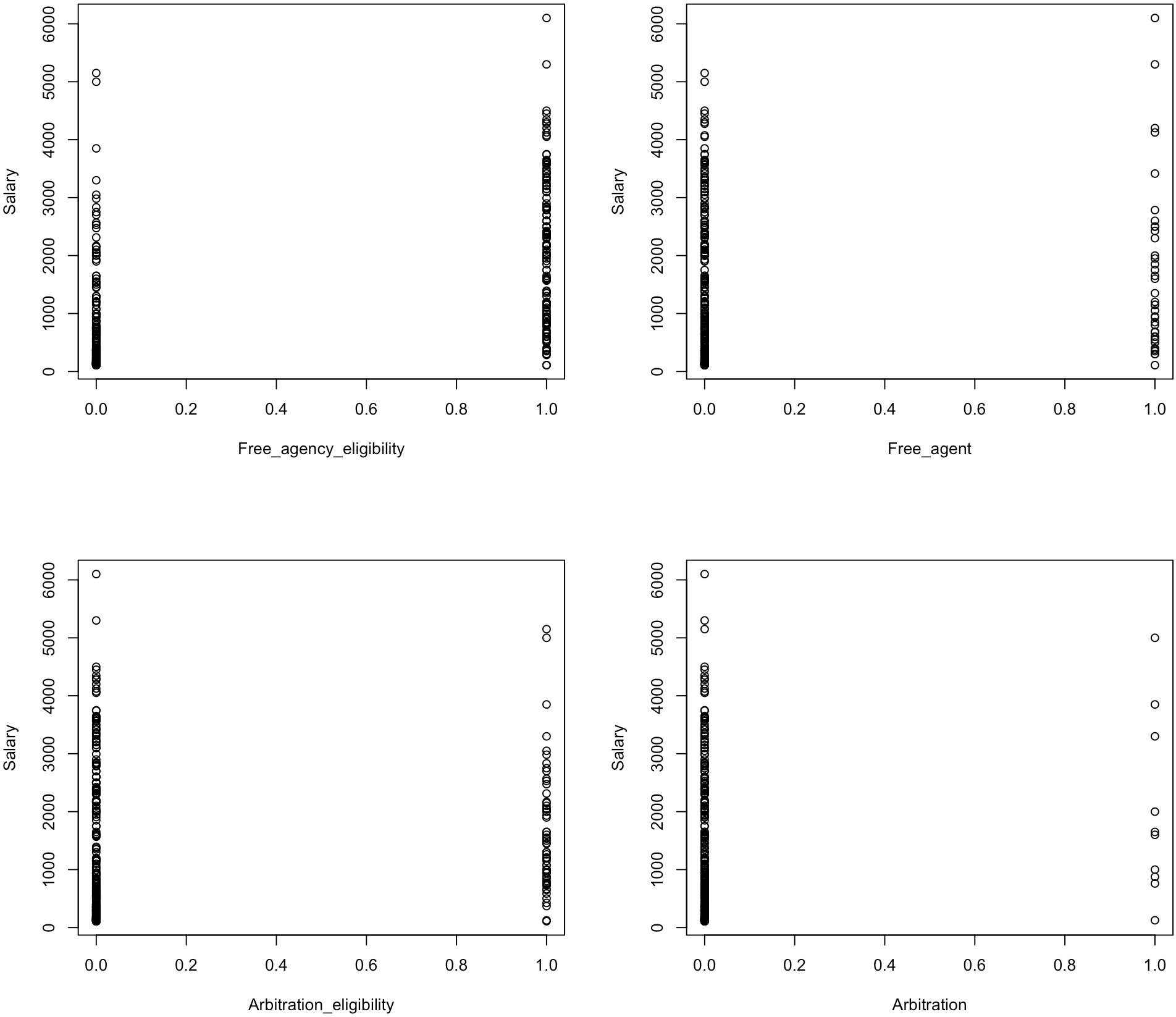


Figura 8: Plot de las variables con valores de 0 o 1.

### 2.1.3 Descripción del conjunto de datos a partir de los puntos anteriores.

En resumen, tenemos un problema de regresión en el que se pretende realizar un modelo para obtener como variable respuesta el salario de los jugadores de un equipo de béisbol. Para ello contamos con 16 variables, de las cuales 4 se mueven entre valores de 0 o 1 y el resto son enteros o valores reales. Hemos podido observar la distribución de estas variables y sus gráficas con respecto a la variable de salida. A simple vista no se podría decir mucho sobre que variables son las más importantes, pero puede decirse que algunas de las más influyentes pueden ser: “Hits”, “Runs”, “HomeRuns” … por lo cual miraremos la matriz de correlación.

Como la correlación simple se da con valores cercanos a 1 o -1, indicando alta correlación entre cada par de variables, y valores cercanos a 0 indican el caso contrario, se tiene que las variables correladas son:

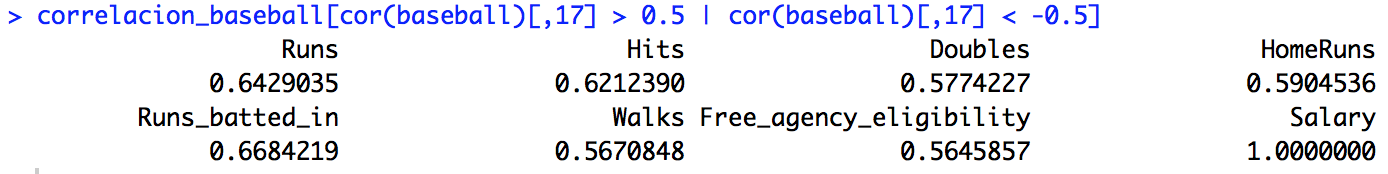


Figura 9: Correlación simple entre cada variable con la variable respuesta

Donde tenemos “Runs”, “Hits”, “Doubles”, “HomeRuns”, “Runs\_batted\_in”, “Walks” y “Free\_agency\_elegibility”. Es decir, estas son las variables que tendremos más en cuenta para nuestros modelos.

## 2.2 Australian

El dataset Australian contiene 15 variables/columnas de las que 14 son entradas y 1 salida. Las variables de este dataset no tienen ningún nombre por lo que no podemos describirlas. Este es un problema de clasificación en el que tenemos 2 clases, la 0 y la 1.

Tenemos en total 690 instancias/observaciones y en ninguna hay valores perdidos. Son datos reales, y los tipos de cada variable son 3 reales, 5 enteros y 6 nominales.

Como podemos ver en la web, tenemos que, de las 14 variables de números enteros, tenemos 4 que se mueven entre 0 o 1. Estas son las variables 1, 8, 9 y 11, además de la de salida, la 15.

Como información adicional tenemos que: “El archivo concierne a la aplicación de tarjetas de crédito. Todos los nombres de los atributos y los valores han sido cambiados para proteger la confidencialidad de los datos. El dataset es interesante porque hay una buena mezcla de atributos: continuos, nominales con pequeños valores de números, y nominales con grandes valores numéricos.”

### 2.2.1 Cálculo de media, desviación estándar, etc.

Para obtener la información la distribución de los datos, su media, valores mínimos, máximos, cuartiles… se ha hecho la función *summary():*

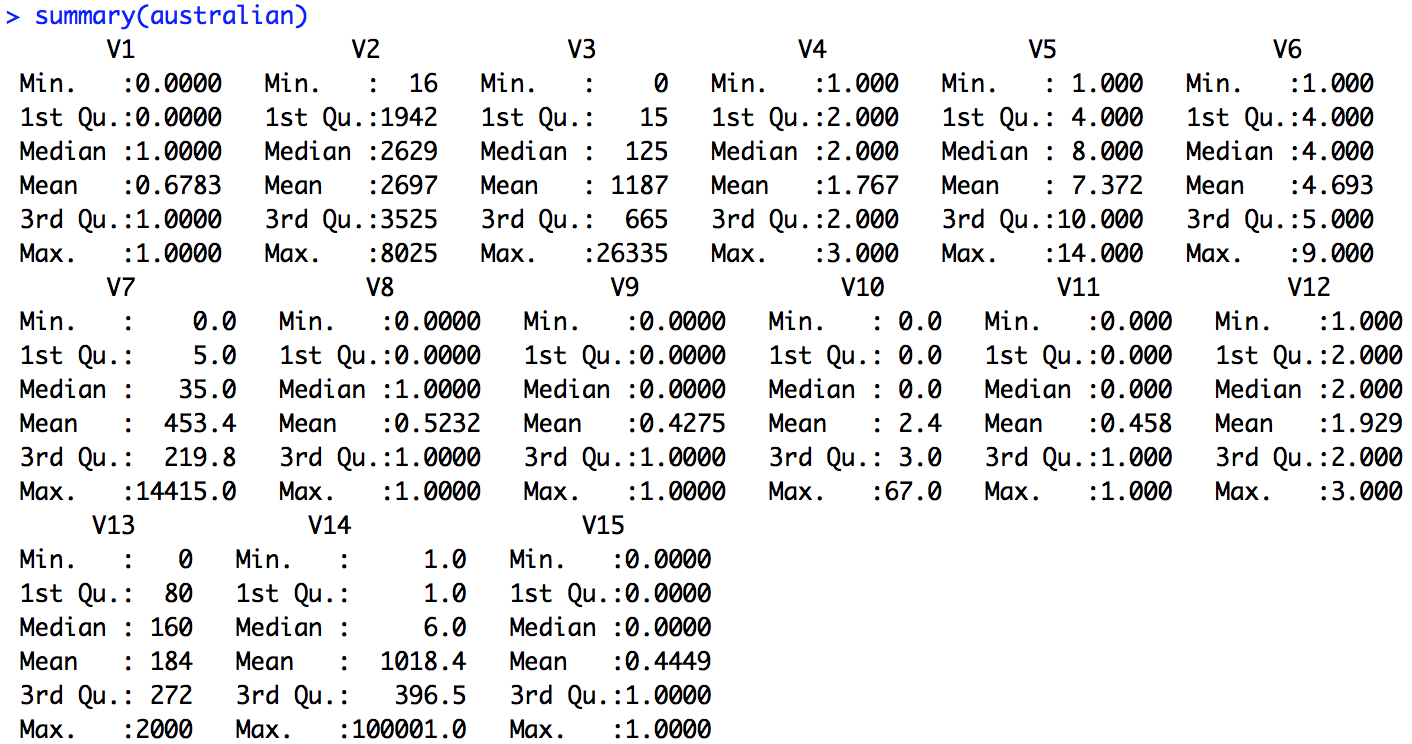


Figura 10: Resumen de distribución de los datos en cada variable para el dataset “australian”

Si queremos la desviación estándar, con *apply()* podemos obtener la información:

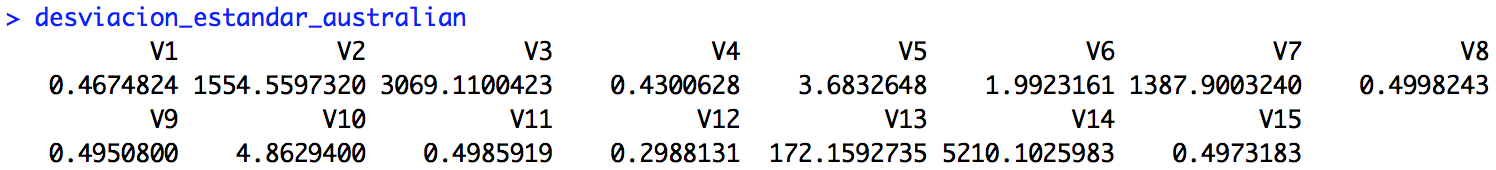


Figura 11: Desviación estándar para el dataset “australian”

Para la varianza:

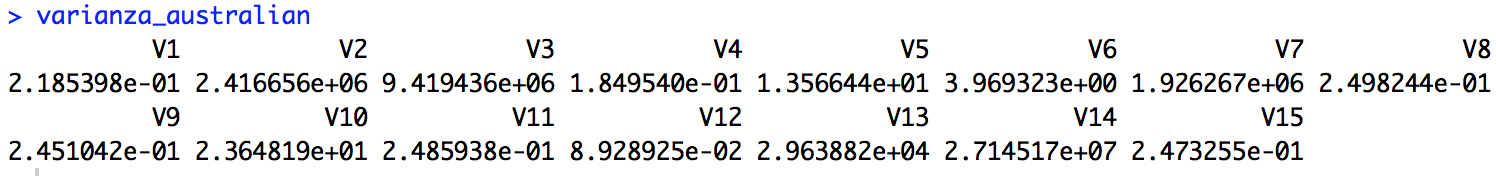


Figura 12: Varianza para el dataset “australian”

Para la desviación absoluta de la mediana:

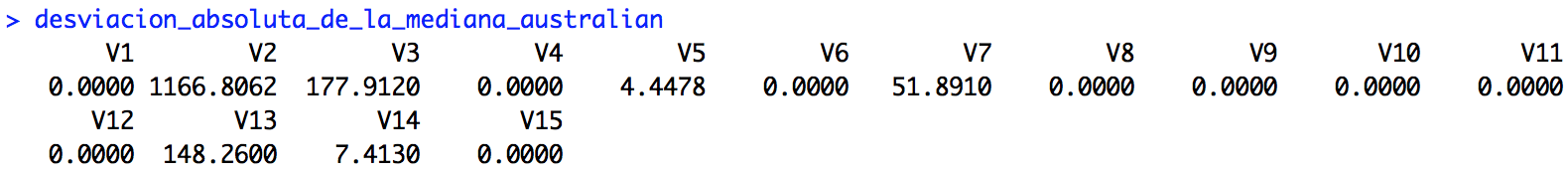


Figura 13: Desviación absoluta de la mediana para el dataset “australian”

Para el rango intercuartil:

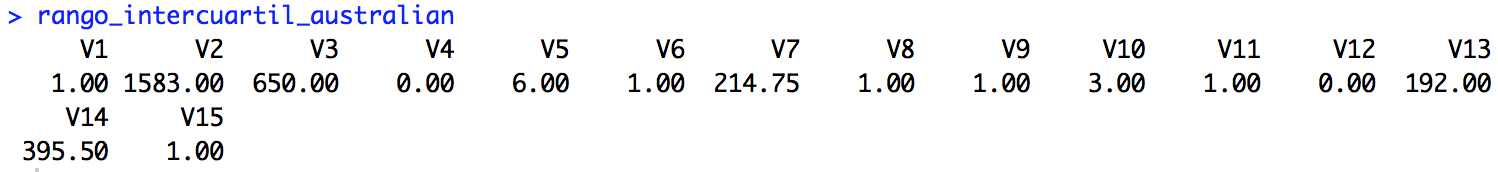


Figura 14: Rango intercuartil para el dataset “australian”

### 2.2.2 Gráficos que permitan visualizar los datos adecuadamente.

Si queremos ver las gráficas de todas las variables con respecto a todas, podemos hacer uso de la función *plot()*

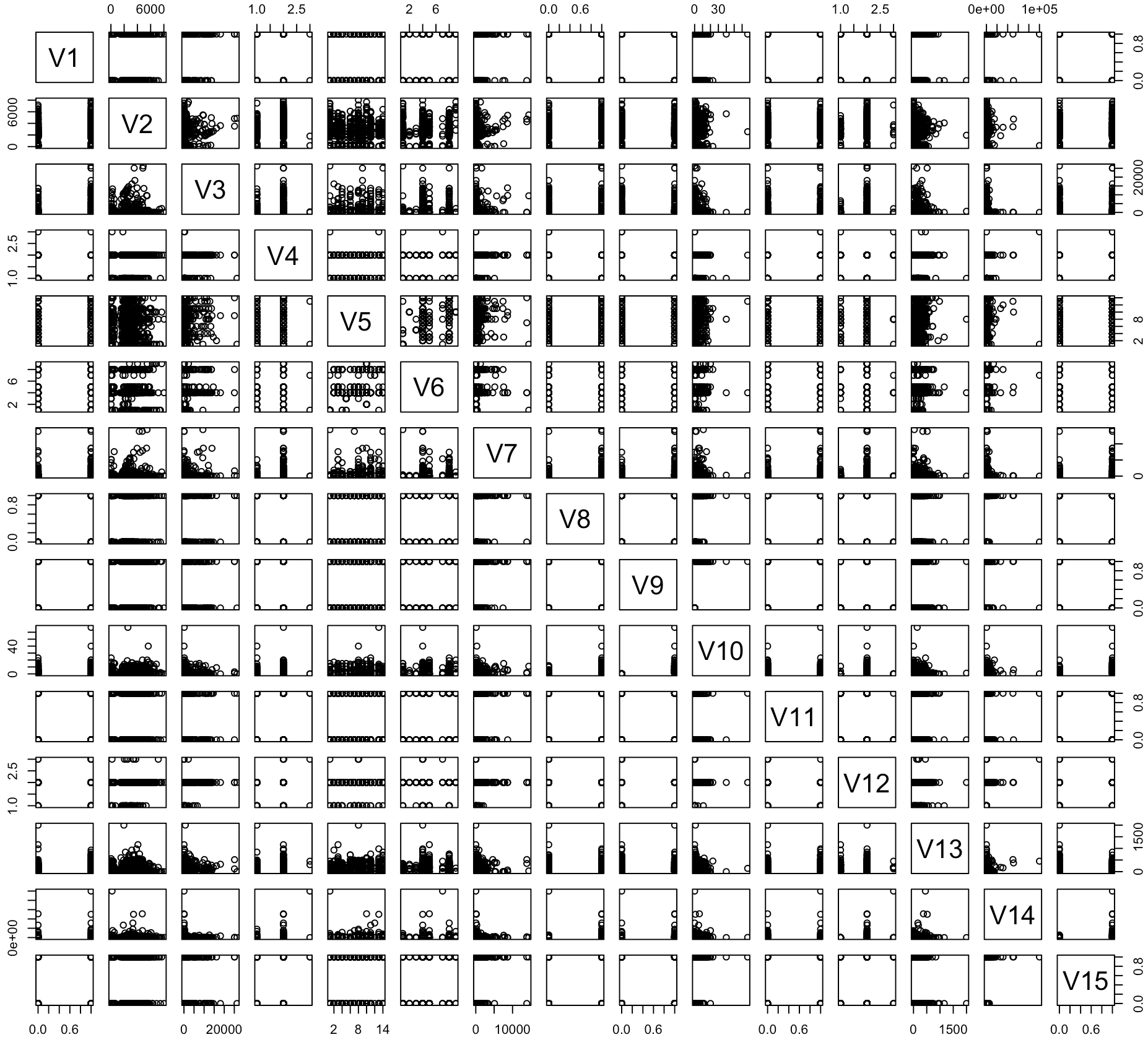


Figura 15: Plot de todas las variables del dataset “australian”

Pero si lo que queremos es ver todas las variables con respecto de la variable de salida, es decir la variable 15, entonces tendremos:

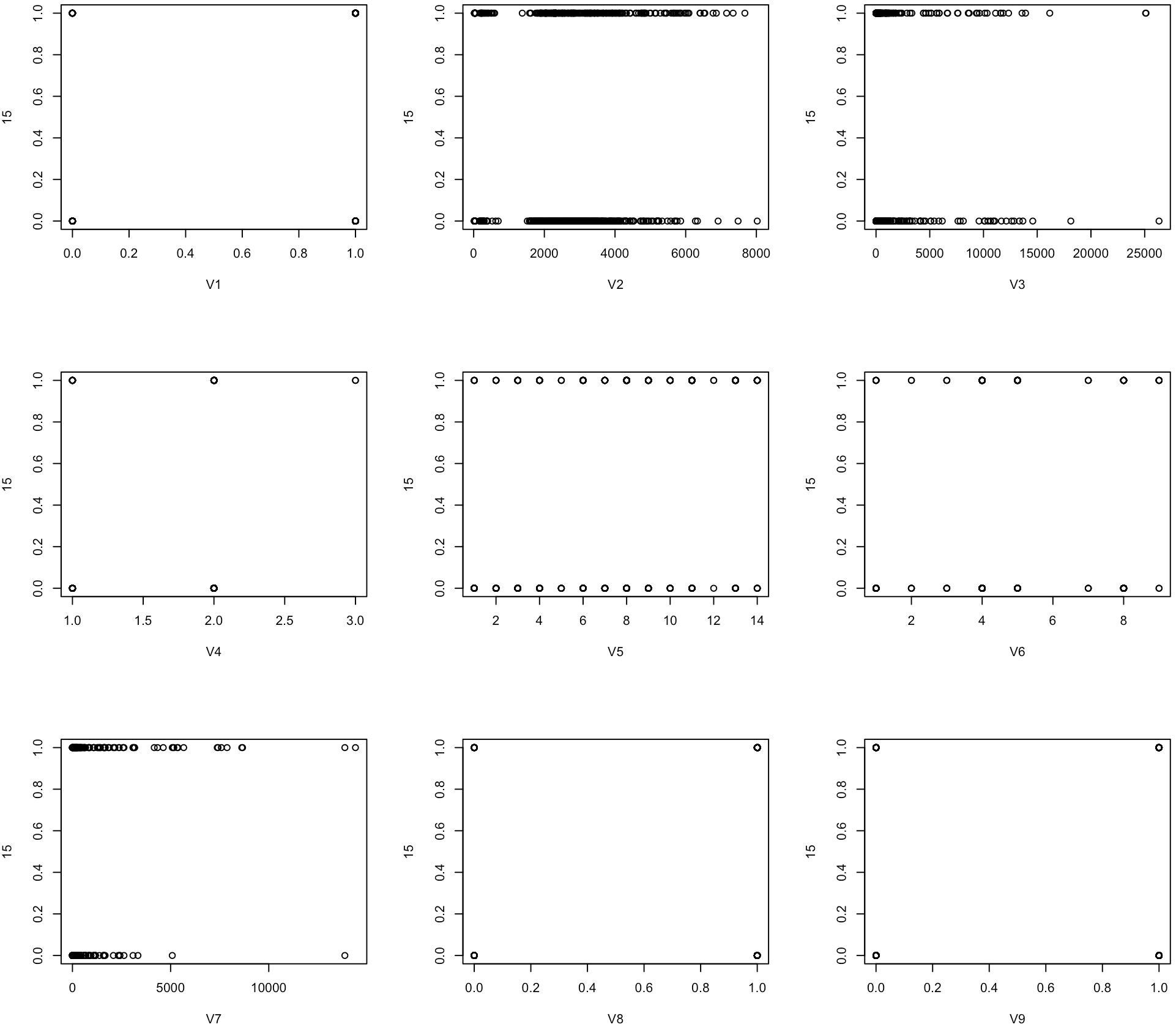


Figura 16: Plot de las primeras 9 variables

Además, tendremos:

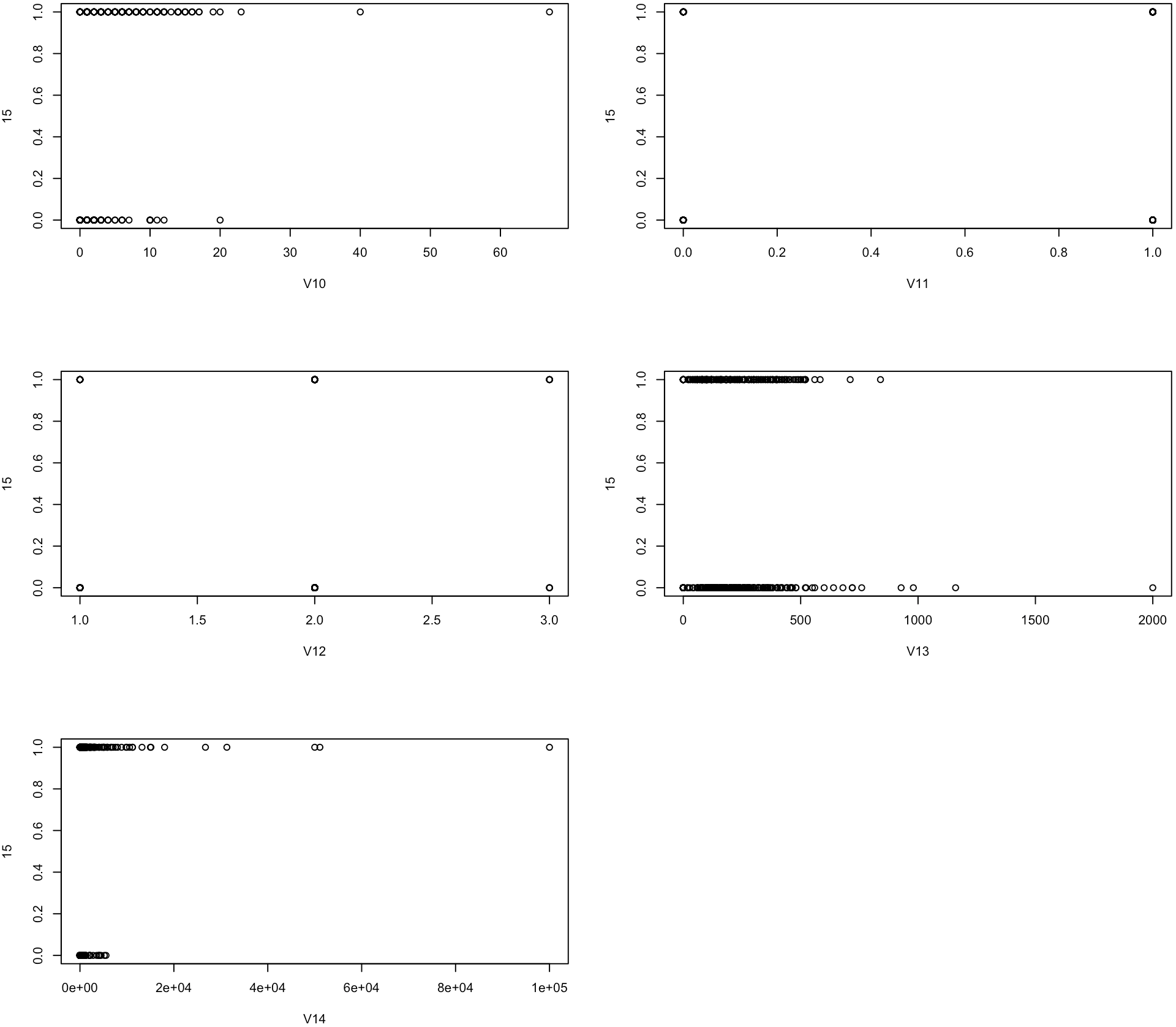


Figura 17: Plot del resto de variables.

### 2.2.3 Descripción del conjunto de datos a partir de los puntos anteriores.

En resumen, tenemos un problema de clasificación en el que se pretende realizar un modelo para obtener como variable respuesta una de las dos clases. Para ello contamos con 14 variables, de las cuales 4 se mueven entre valores de 0 o 1 y el resto son enteros, valores reales o nominales. Hemos podido observar la distribución de estas variables y sus gráficas con respecto a la variable de salida.

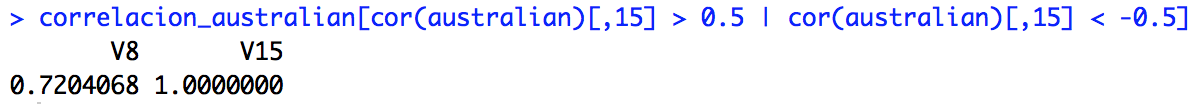


Figura 18: Correlación simple entre cada variable con la variable respuesta.

A simple vista no se podría decir mucho sobre que variables son las más importantes, pero con ayuda de la matriz de correlación, vemos como la Variable 8 está fuertemente correlada con la variable predictora, por lo que la tendremos muy en cuenta cuando realicemos nuestros modelos.

# 3 Regresión

Para la base de datos “baseball” voy a aplicar regresión, de forma que vamos a obtener diferentes modelos, tanto para regresión simple, regresión múltiple y para knn, de forma que finalmente los compararemos para ver cuál se adapta mejor al problema.

## 3.1 Regresión simple

Para el estudio de la regresión simple, voy a usar las variables que tienen más correlación ya que nuestro dataset tiene más de 5 variables. Estas variables nos habían salido:

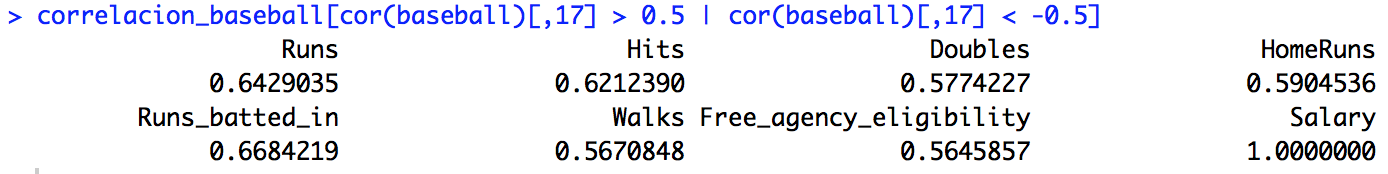


Figura 19: Correlación simple entre cada variable con la variable respuesta

Donde tenemos “Runs”, “Hits”, “Doubles”, “HomeRuns”, “Runs\_batted\_in”, “Walks” y “Free\_agency\_elegibility”. Por lo tanto, yo voy a elegir como las más importantes a: “Runs”, “Hits”, Runs\_batted\_in”, “HomeRuns” y “Doubles”. Veamos que tal salen estos modelos de regresión simple. La visualización de estos modelos nos queda:

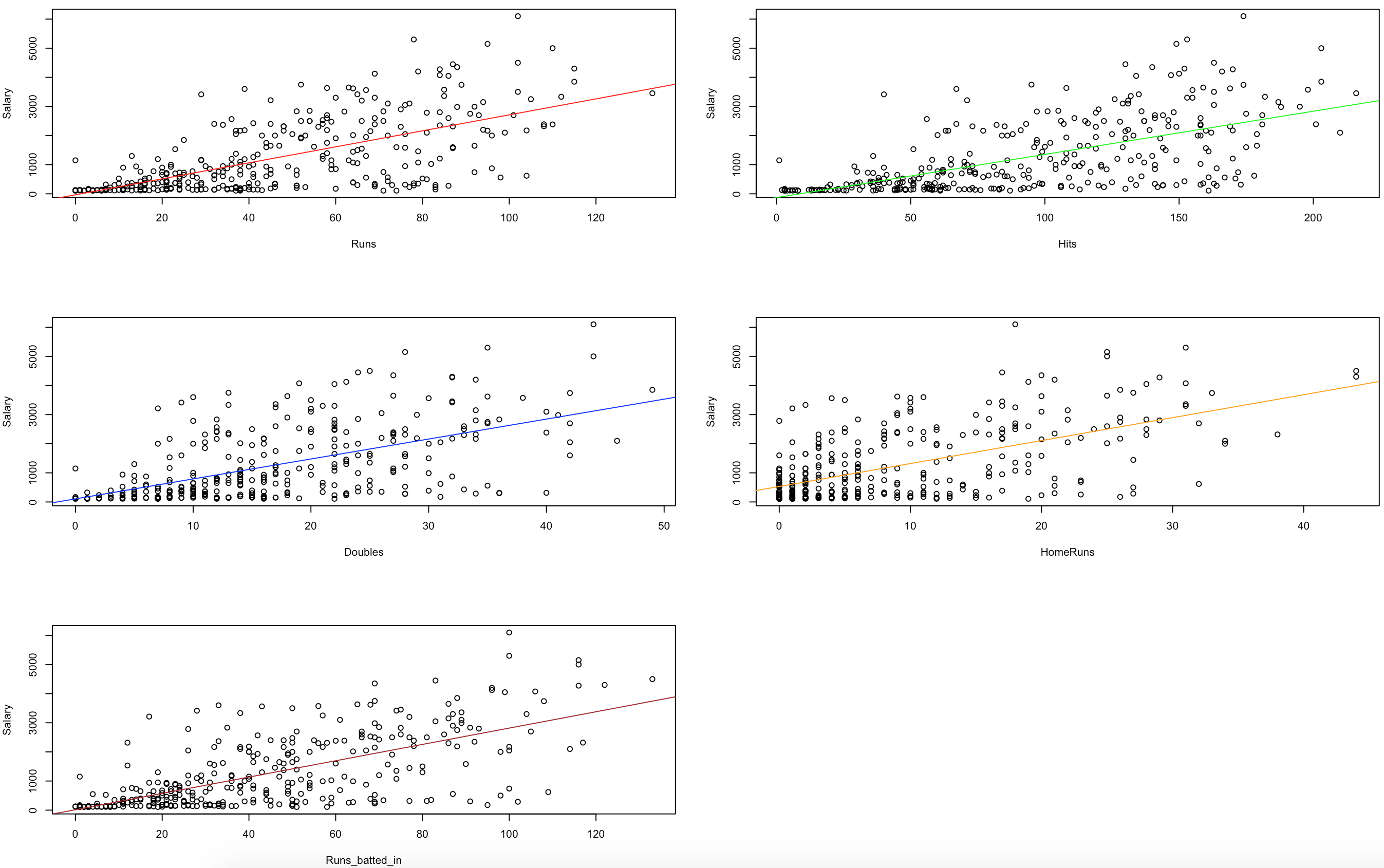


Figura 20: Visualización de los modelos

Además, si queremos ver los datos más importantes:

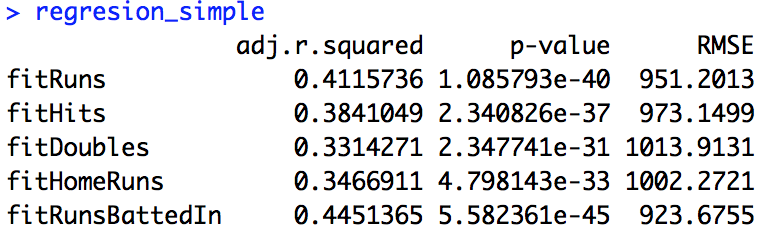


Figura 21: Datos más importantes de los modelos

De donde vemos como los modelos tiene un bajo valor de R^2 por lo que debemos seguir mejorando los modelos.

Como el siguiente paso a realizar es regresión múltiple, y en este curso hemos estudiado regresión mediante cross-validation, he realizado las funciones que se dieron en la sesión de laboratorio, pero modificadas para obtener los datos más interesantes. De esta forma, tras aplicarlo sobre los modelos arriba vistos, de regresión simple, comparando además un modelo de regresión múltiple con todas las variables, obtenemos lo siguiente:

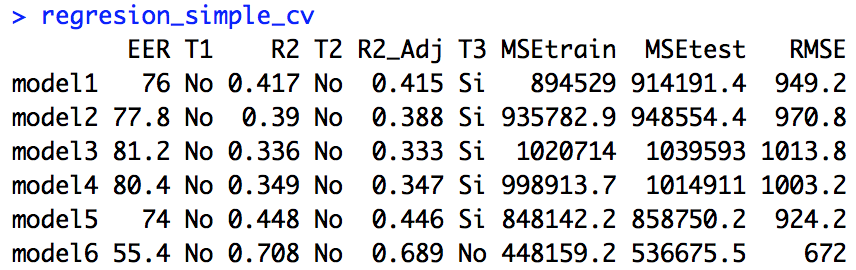
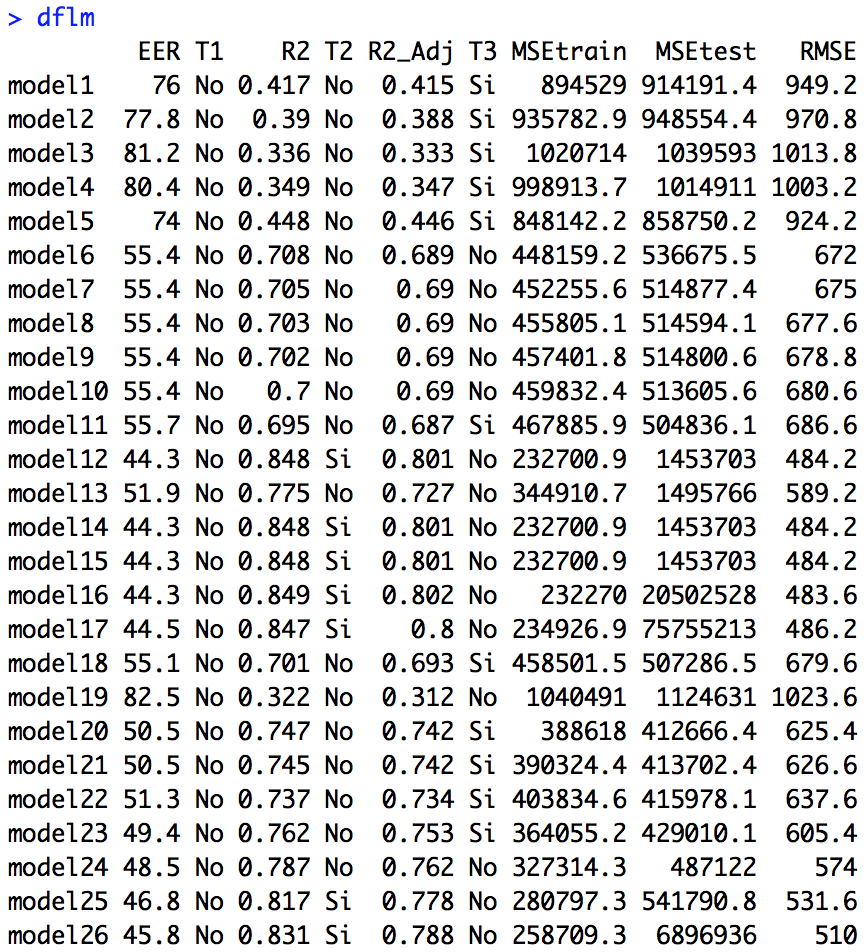


Figura 22: Datos más importantes de los modelos

Por lo que vemos que el modelo con todas las variables (modelo6) obtiene mejores valores que el resto de modelos de regresión simple, con un 68,9% de R2 ajustado.

## 3.2 Regresión múltiple

Como se ha dicho antes, para realizar regresión múltiple se han usado las funciones vistas en el curso para hacer cross-validation. Además, en el código entregado, yo he realizado los modelos sin cross-validation, para observar que se obtienen unos valores casi iguales.

Lo primero que se ha realizado han sido modelos donde se han usado todas las variables y a partir de este se han ido eliminando las que parecían menos importantes o que no ayudaban a explicar el modelo.

Seguidamente se han intentado hacer interacciones y aplicar no linealidad a los modelos, de forma que hemos ido progresando en los resultados.

Finalmente hemos realizado 21 modelos más los 5 modelos simples.

Los resultados de estos modelos son los que se muestran en la figura 23:

Figura 23: Datos más importantes de los modelos

El modelo que ha obtenido un menor MSE en test ha sido el modelo 20 con un R2 y R2adj del 0.742, explicando un 74% del problema. El modelo que mejor MSE en train obtiene es el modelo 16 con muy buen R2. El modelo con mayor MSE en train es el 19 y el que mayor MSE en test obtiene es el 17, indicando sobre-ajuste del modelo. El modelo 16 es el que obtiene un mayor valor de R2 y R2adj.

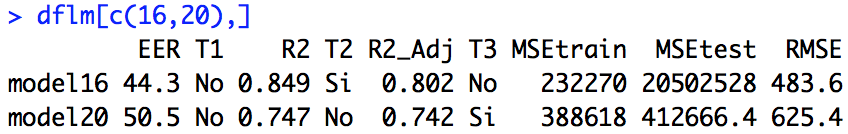


Figura 24: Datos más importantes de los modelos

Por lo tanto, de los mejores modelos obtenidos tenemos al modelo 16 y el modelo 20. Finalmente me decantaré a elegir como el mejor al modelo 20 al tener un menor valor de MSE en test.

Este modelo es el siguiente:

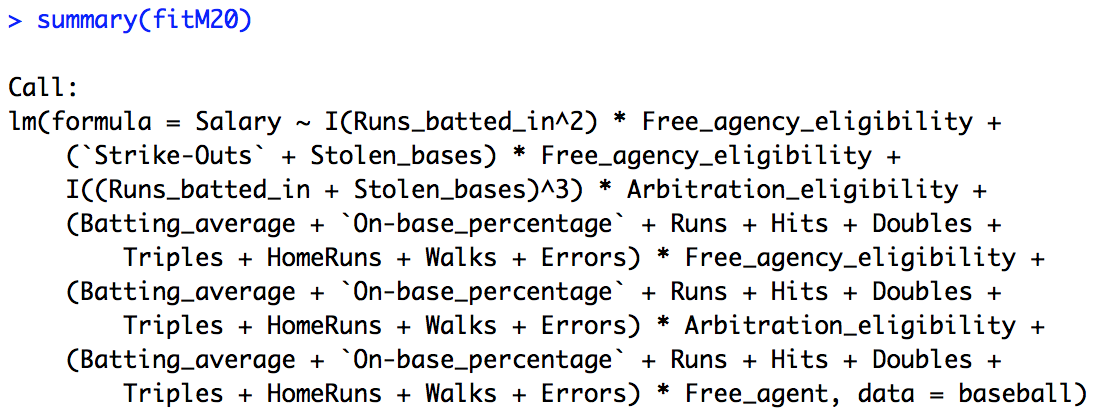


Figura 25: modelo20

O también:

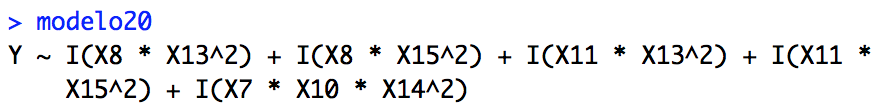


Figura 26: modelo20

El modelo se ve que es bastante difícil de explicar, pero nos explica un 74% del problema.

## 3.3 k-NN para regresión

A continuación, pasamos a knn para regresión. Lo primero que he realizado ha sido como se mostraba en las trasparencias del laboratorio, un modelo knn con todos los datos del dataset para modelo simples. Estos modelos visualizados nos dan el siguiente resultado:

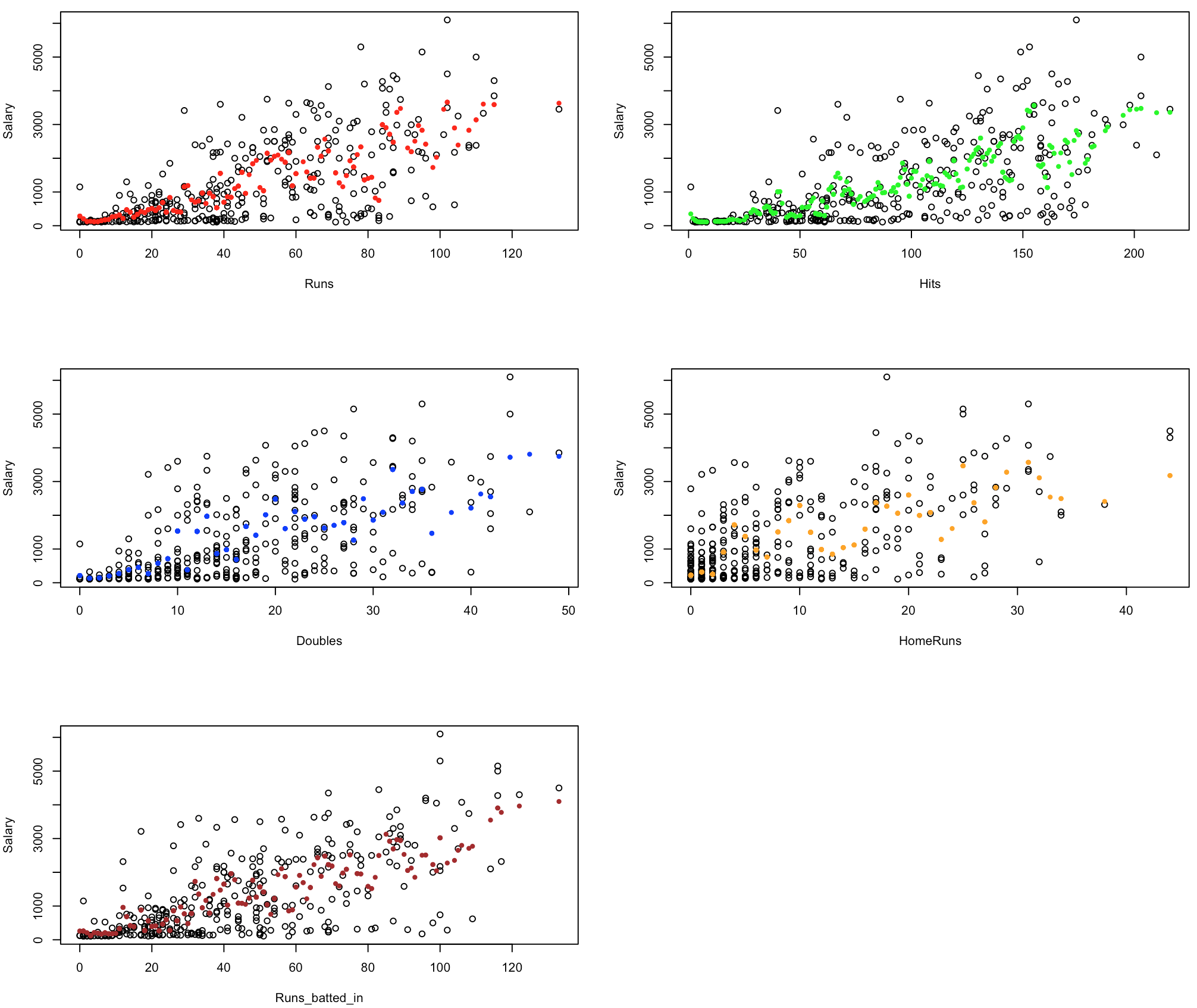


Figura 27: visualización de los modelos

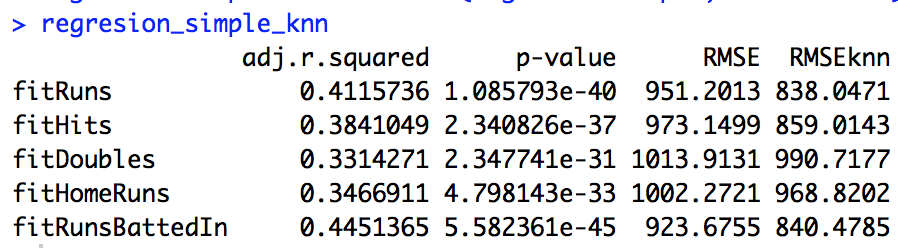


Figura 28: comparación de RMSE para lm y knn

Si observamos la comparación de los valores de RMSE para lm y RMSE para knn vemos como para knn obtenemos valores de RMSE más bajos que en lm.

A continuación, vamos a pasar a estudiar knn pero con cross-validation con las funciones facilitadas (modificadas) en las transparencias de este curso.

Como resultado de todos los modelos para knn tenemos:

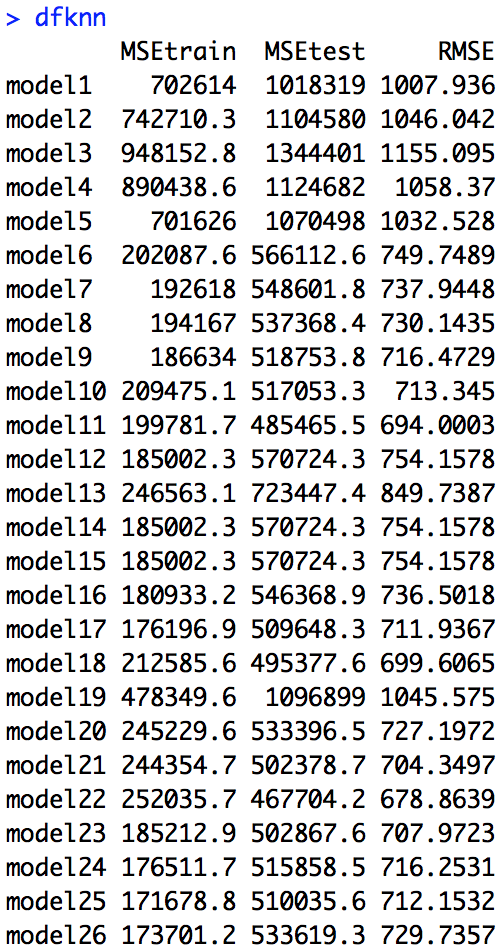


Figura 29: comparación de los modelos para knn

Vemos los resultados obtenidos para todos los modelos knn. Estos modelos son los mismos que en lm, ya que para una correcta comparación tenemos que tener los mismos modelos.

Como podemos observar el que menor MSE en test obtiene es el modelo 22. El que menor MSE en train obtiene es el 25. El modelo que mayor MSE en test y en train presenta es el modelo 3. Por lo tanto, vamos a comparar los modelos 22 y 25 para ver cuál sería el mejor modelo de knn:

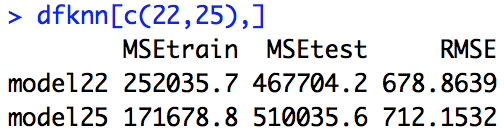


Figura 30: comparación de los modelos para knn

Por lo tanto, para knn, elegiría como mejor modelo el modelo 22 al obtener un valor más pequeño de MSE en test. Este modelo sería:



Figura 31: modelo22

## 3.4 Comparación de algoritmos

Para la comparación de algoritmos debo elegir entre el mejor modelo de lm o de knn. Estos modelos han sido el modelo 20 en lm y el 22 en knn:

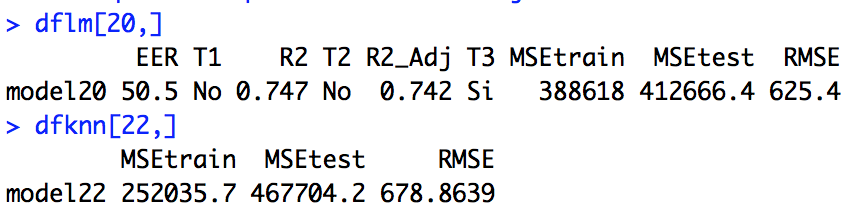


Figura 32: comparación de los modelos de lm y knn

De entre estos dos modelos elegiré como final en las comparaciones el modelo 20 de lm, ya que obtiene el valor de MSE en test más bajo de los dos.

Primeramente, para la comparación de los algoritmos se cargarán los datos de los archivos “regr\_train\_alumnos” y “regr\_test\_alumnos” para sustituir nuestros valores obtenidos de MSE del modelo 20.

El test de Wilcoxon nos proporciona los siguientes valores para test:

|  |  |  |
| --- | --- | --- |
| R+ | R- | P-value |
| 81 | 90 | 0.8650436 |

Donde obtenemos un p-value mayor a 0.05 no existiendo diferencias significativas entre ambos. Solo tenemos un 13,49% de confianza de que existan diferencias. Por lo tanto, no somos capaces a rechazar la hipótesis nula y concluiremos que no hay suficiente evidencia en los datos. Por lo tanto, aceptamos la hipótesis nula: las dos distribuciones son iguales.

El test de Wilcoxon nos proporciona los siguientes valores para train:

|  |  |  |
| --- | --- | --- |
| R+ | R- | P-value |
| 10 | 161 | 0.000328064 |

Los datos obtenidos en el train en el test de Wilcoxon muestra un p-value menor que el nivel de significación 0.05, por lo que el p-value es significativo. Entonces rechazaremos la hipótesis nula y decimos que las distribuciones son distintas. Esta interpretación tiene menor peso que la realizada anteriormente para test, ya que en el training los resultados son con datos ya conocidos por lo que se puede decir que hay sobre-ajuste para cada resultado de las bases de datos y que sean distintas las distribuciones entre los ajustes lm y knn.

Si realizamos el test de Friedman para test obtenemos un p-value del 0.0302 lo que quiere decir que existen diferencias significativas en al menos entre un par de algoritmos.

Como resultado del post-Hoc de Holm obtenemos:

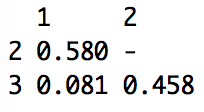


Figura 33: resultado de Post-Hoc de Holm

Por lo que existen diferencias significativas a favor de M5’, mientras que lm vs knn pueden ser considerados equivalentes.

Si realizamos el test de Friedman para train obtenemos un p-value del 8.365e-05 lo que quiere decir que existen diferencias significativas en al menos entre un par de algoritmos.

Como resultado del post-Hoc de Holm obtenemos:

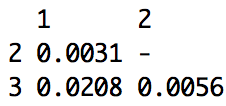


Figura 34: resultado de Post-Hoc de Holm

Por lo que existen diferencias significativas a favor de los 3 algoritmos comparados. Lo que sugiere un sobre-ajuste.

# 4 Clasificación

Para la base datos “australian” voy a aplicar clasificación, de forma que vamos a obtener diferentes modelos como son, kNN (k nearest neighbor), LDA (linear discriminant analysis) y QDA (quadratic discriminant analysis), donde finalmente realizaremos una comparativa para ver cual se adapta mejor al problema.

## 4.1 k-NN

Tras realizar un breve análisis, se ha podido observar que para las diferentes clases hay 383 datos de la clase 0 y 307 de la clase 1. Es decir, un 55,5% de clase 0 y un 44,5% de clase 1.

Si normalizamos las variables y realizamos su visualización obtenemos:

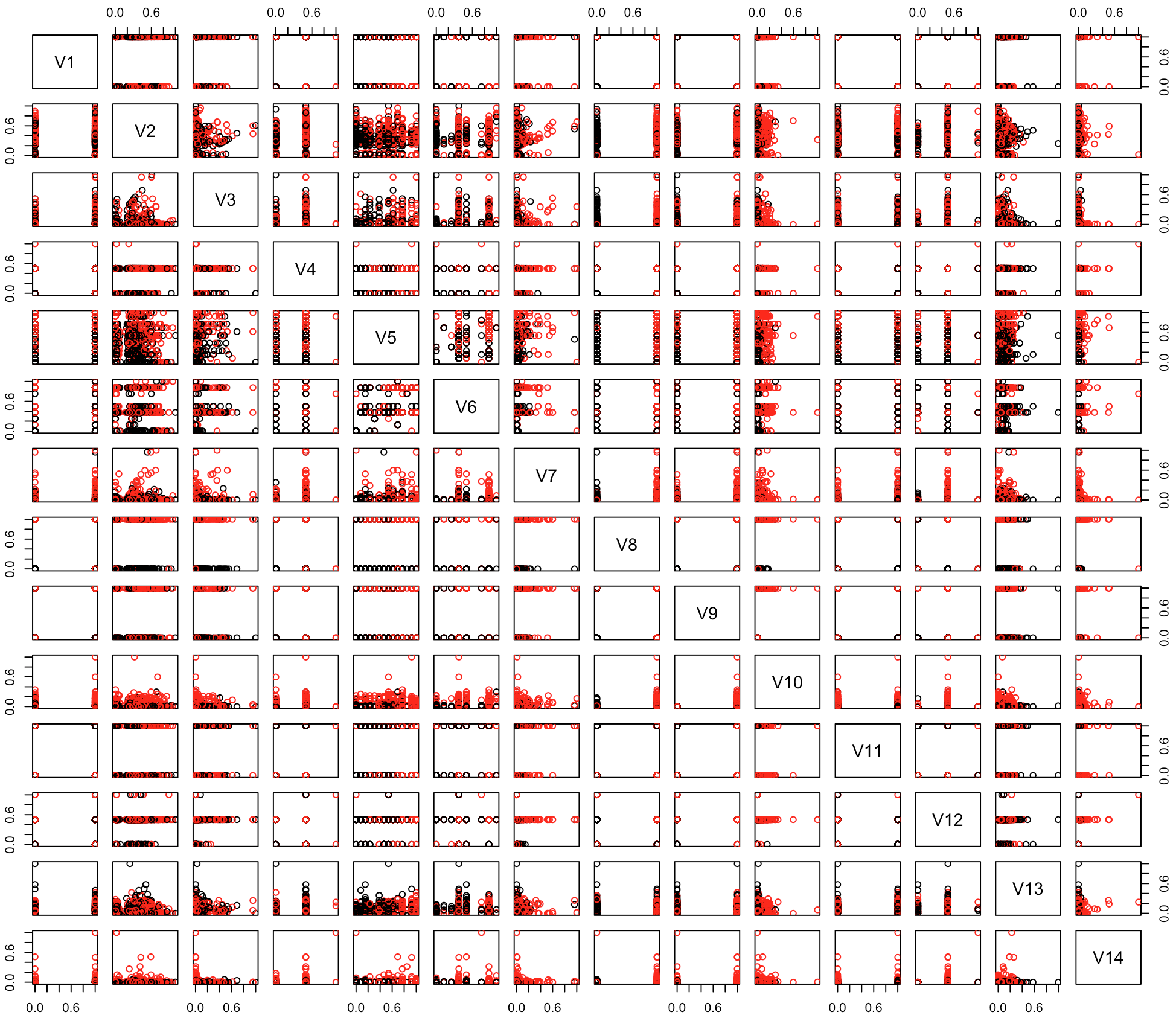


Figura 35: visualización de las variables

Si observamos la correlación entre las variables, podemos observar como la V9 y V10 son las que mayor correlación tiene, obteniendo un 57%. Tendremos en cuenta esto para un posterior modelo.

Comenzando los modelos, he hecho, primeramente, un proceso semejante al de las transparencias, por lo que he divido el conjunto en train y test con un 80-20%. He realizado un knn con el conjunto de train, y un k =21, obteniendo que, a la hora de clasificar el test, se obtiene un modelo con un 85,5% de explicación del problema. Si cambio el k a 7, el modelo obtiene un 84,7% de explicación.

Si hacemos uso de caret, obtenemos que el mejor modelo es con k=9 y que su explicación sería del 84%.

Esto ha sido sobre los datos sin normalizar, por lo que si se hace normalizando los datos obtendríamos, con caret, un modelo que con k = 15, es el mejor favor obteniendo una explicación del 82,6%.

Una vez realizada esta prueba, he creado las funciones para realizar, de forma automática, el estudio de los distintos modelos. Estas funciones están inspiradas en las vistas en las transparencias de clase, usando los conjuntos de train y test que han sido dados junto con los dataset y usando caret en todos los modelos para obtener buenos valores medios.

En los apartados siguientes, concretamente en QDA, explicaré que modelos he usado y los resultados proporcionados por los diferentes métodos para no repetir la misma información en todos los apartados.

## 4.2 LDA

Antes de realizar LDA, he realizado, ya que se vio en clase, el modelo GLM obtenido, primeramente, con la división realizada por mí, un modelo que explica el 87,68% del problema. En el siguiente apartado realizaré la comparativa también con este modelo.

Para LDA, si realizo lo mismo que para kNN y GLM, obtengo un modelo que explica el 87,68% del problema. Y como en los casos anteriores, en el siguiente apartado mostraré la comparativa con el resto de modelo.

## 4.3 QDA

Por último, para QDA he realizado los mismos pasos que para el resto de modelos, obteniendo sobre mi conjunto de train y test un 78,26% de explicación del problema.

Una vez realizados todos los modelos, he pasado a compararlos, obteniendo la siguiente comparativa:

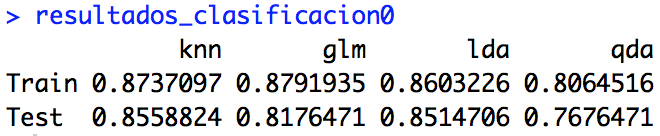


Figura 36: comparativa de distintos modelos

En esta comparativa de la Figura 36, se pueden ver los distintos valores que se han obtenido para los modelos kNN, GLM, LDA y QDA.

Viendo estos resultados, he pasado a realizar distintos modelos de nuestro problema, haciendo uniones de variables o eliminando algunas de ellas que podrían ser innecesarias. Uno de los problemas que he tenido, ha sido la poca información disponible del dataset, ya que las variables no tienen ninguna información asociada y no se puede hacer ninguna suposición sobre ellas.

Por lo tanto, los distintos modelos que he realizado han sido:

* Modelo 0: todas las variables
* Modelo 1: Unión de las variables V9 y V10 ya que se vio que guardaban correlación entre ellas.
* Modelo 2: Al unir V9 con V10, se ha visto que la siguiente correlación más fuerte han sido las variables V5 y V6, por lo que se han unido también.
* Modelo 3: He realizado un GLM con todas las variables, de este nuevo data set, observando los p-value. Con este p-value voy a decidir eliminar las variables que tienen este valor mayor que 0.5. Estas variables son V1 y V3.
* Modelo 4: Para este modelo, he realizado otro GLM con las variables restantes y he decidido eliminar V2, V11, V12 y V7.
* Modelo 5: Con las correlaciones del dataset actual, voy a unir las variables V8 y V9, donde V9 era la unión de V9 y V10.

Una vez probados todos los modelos, kNN, GLM, LDA y QDA, he obtenido los siguientes resultados:

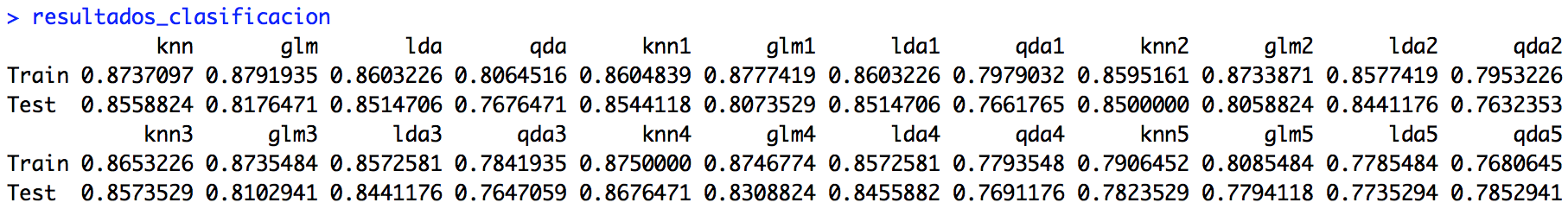


Figura 37: comparativa de distintos modelos

De donde podemos ver como knn4 es el modelo que tiene un mayor valor en test. El mejor modelo para LDA es el modelo 0. Para GLM es el modelo 4. Para QDA el mejor modelo es el 5.

A continuación, pasaremos a realizar la comparación de los algoritmos.

## 4.4 Comparación de algoritmos

Para la comparación de los algoritmos he elegido los mejores de cada modelo, es decir, el mejor de kNN, LDA y QDA.

Primeramente, para la comparación de los algoritmos se cargarán los datos de los archivos “clasif\_test\_alumnos.csv” y “clasif\_train\_alumnos.csv” para sustituir nuestros valores obtenidos.

El test de Wilcoxon nos proporciona los siguientes valores para test:

El test de Wilcoxon nos proporciona los siguientes valores para test:

|  |  |  |
| --- | --- | --- |
| R+ | R- | P-value |
| 113 | 97 | 0.7841263 |

Donde obtenemos un p-value mayor a 0.05 no existiendo diferencias significativas entre ambos. Solo tenemos un 21,58 % de confianza de que existan diferencias. Por lo tanto, no somos capaces a rechazar la hipótesis nula y concluiremos que no hay suficiente evidencia en los datos. Por lo tanto, aceptamos la hipótesis nula: las dos distribuciones son iguales.

El test de Wilcoxon nos proporciona los siguientes valores para train:

|  |  |  |
| --- | --- | --- |
| R+ | R- | P-value |
| 144 | 66 | 0.1536465 |

Al igual que en test, para el train, no existen diferencias significativas al ser p-value mayor que 0.05.

Si realizamos el test de Friedman para test obtenemos un p-value del 0.9512 lo que quiere decir que no existen diferencias significativas en los algoritmos.

Como resultado del post-Hoc de Holm obtenemos:

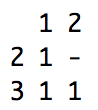


Figura 38: resultado de Post-Hoc de Holm

Si realizamos el test de Friedman para train obtenemos un p-value del 0.522 lo que quiere decir que no existen diferencias significativas los algoritmos.

Como resultado del post-Hoc de Holm obtenemos:

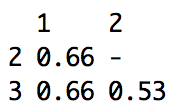


Figura 39: resultado de Post-Hoc de Holm