PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE TALLER DE INVESTIGACIÓN APLICADO DIPLOMADO DE MÉTODOS ESTADÍSTICOS 2018

PREDICCIÓN DE LA TASA DE PROCESAMIENTO DE UN MOLINO SEMI AUTÓGENO SAG EN PLATAFORMA R STUDIO

Enero del 2019

PROCESOS PRODUCTIVOS DE LA GRAN MINERÍA

Dentro de la cadena productiva de la gran minería del cobre existen varios procesos cuyo propósito es concentrar y purificar el mineral que se extrae desde el yacimiento. Principalmente podemos mencionar los siguientes:

1. Mina

Se incluyen todos los procesos necesarios para la extracción de la roca mineral desde los yacimientos ya sean de tipo rajo abierto o subterráneos.

2. Conminución

Da cuenta de múltiples procesos en serie tendientes a reducir el tamaño de los minerales extraídos en la mina a fin de prepararlos para los procesos de separación físico químicos.

3. Concentración

Recuperar y concentrar las especies de interés, esto es, los minerales de Cobre (calcopirita, bornita, etc.) y Molibdeno (molibdenita), así como también desechar aquellos que son considerados residuos (sílice, pirita, etc.).

4. Refinación

Diferentes etapas de purificación tendientes a lograr un producto final (cátodo de Cobre) que cumpla con los estándares de pureza necesarios para su correcta comercialización.

PROCESOS DE CONMINUCIÓN

El proceso de conminución es un término genérico que se utiliza para designar a los diferentes procesos de reducción de tamaño. Dichos

procesos se caracterizan por un alto consumo de energía y acero.

La necesidad de estos procesos radica en la liberación de el (o los) minerales de interés desde la roca matriz, a fin de acelerar la velocidad de separación de estos en los procesos de concentración.

Con respecto a las etapas del proceso tradicional de conminución, podemos mencionar principalmente el chancado y la molienda. Estos procesos se realizan en etapas consecutivas, en donde cada uno procesa diferentes tamaños de roca según se muestra en la Figura 1, que se muestra a continuación:

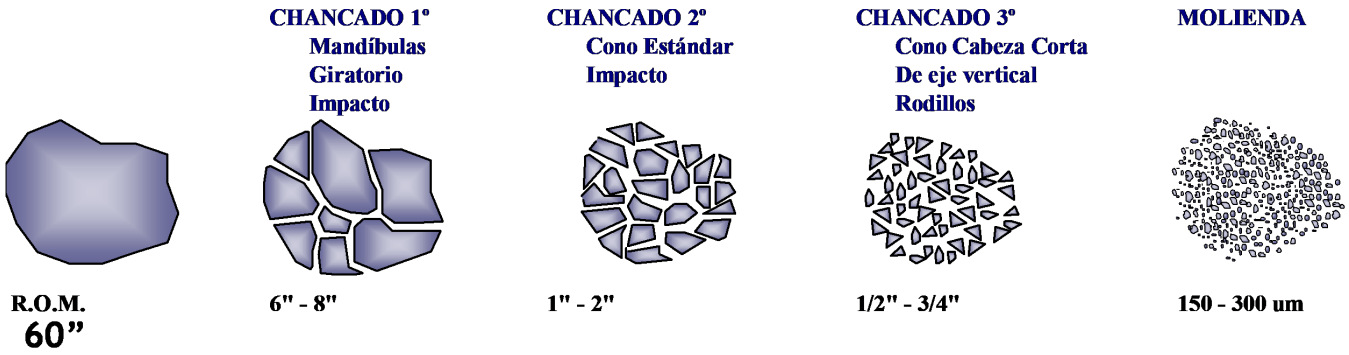


Figura 1: Diferentes Etapas del Proceso de Conminución

PROCESOS DE MOLIENDA

La etapa de molienda es donde se lleva a cabo la última etapa de reducción de tamaño, el propósito final de esta es generar un tamaño óptimo para el proceso de concentración vía flotación de minerales.

La molienda se considera como una de las etapas más importantes en una operación minera, ya que de esta depende la capacidad de tratamiento total (producción) de la faena, asi como el grado de recuperación y concentración de las especies de interés (minerales de cobre) en los procesos subsiguientes de flotación.

En la industria existen varias configuraciones del proceso de molienda, entre los cuales se pueden resaltar las siguientes:

Molienda Convencional

En este proceso los principales equipos son molinos de barras (RM) y bolas (BM) mas equipos clasificadores conocidos como baterías de hidrociclones (BHC), un ezquema habitual de este tipo de configuración se muestra a continuación en la Figura 2:

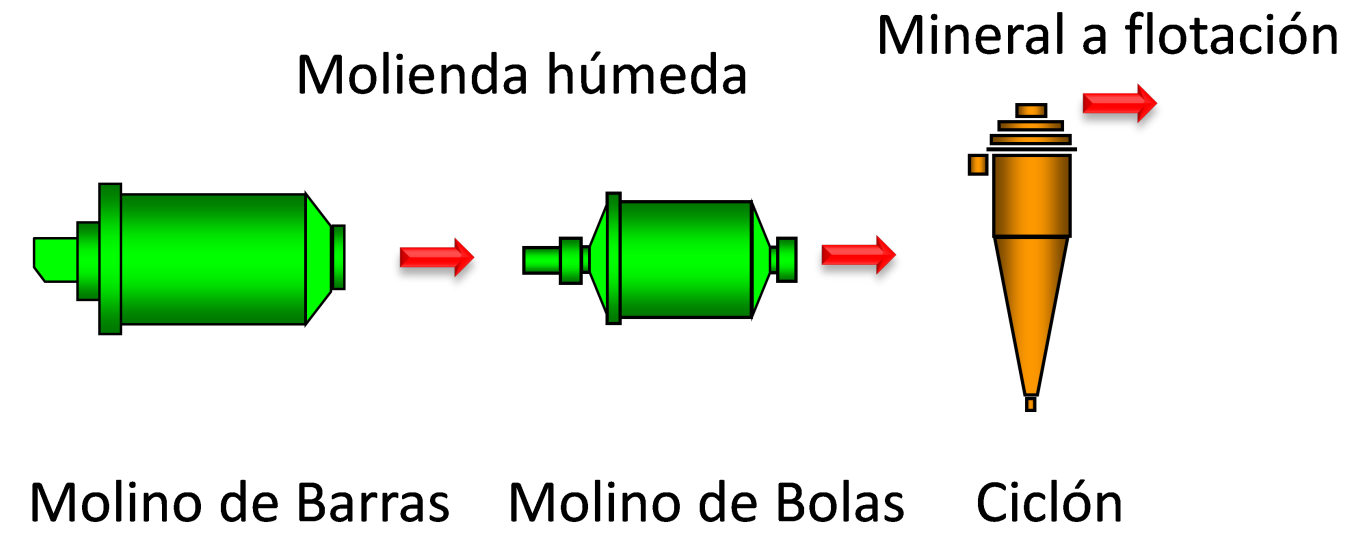


Figura 2: Ezquema de un Circuito de Molienda Convencional

Procesos de Molienda Semi Autógena (SAG)

En este proceso los principales equipos son molinos semi autógenos (SAG), a su vez cada molino SAG debe ir acompañado por una serie de equipos de apoyo, siendo lo mas importantes los molinos de bolas (generalmente 2 por cada molino SAG) y baterías de hidrociclones como clasificadores (generalmente 1 batería por cada molino de bolas), según se muestra en la Figura 3:

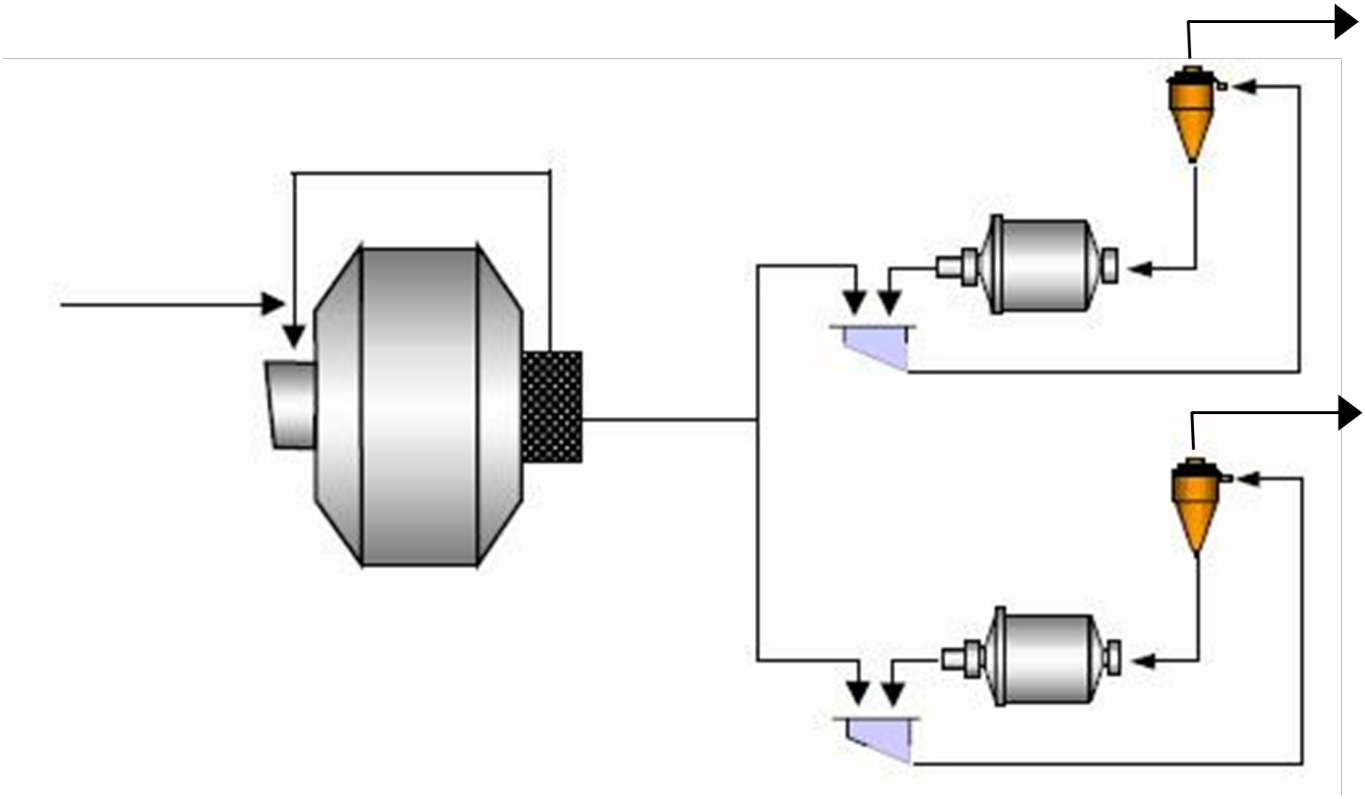


Figura 3: Ezquema de un Circuito de Molienda SAG

Cabe señalar a este respecto que los circuitos SAG son comunes a la mayoría de las grandes faenas de procesamiento de minerales, lo anterior debido principalmente a su alta capacidad de tratamiento.

La eficiencia de los procesos de molienda SAG depende en gran manera de diversos parámetros de operación como:

- Distribución de tamaño del mineral que alimenta al molino.
- Velocidad del molino (rpm).
- Propiedades de dureza de mineral.
- Potencia consumida por el molino.
- Volumen de llenado, tanto de bolas como mineral.
- Adición de agua al molino.
- Distribución de tamaño del mineral producto del molino.

Planta de Molienda SAG División Chuquicamta (DCHU).

La sección de molienda SAG (también conocida como circuito A-2) de División Chuquicamata (DCHU) esta compuesta por dos molinos SAG (n°16 y 17) operando en paralelo, cada uno con dos molinos de bolas (BM) operando en circuito inverso. Los Pebbles (termino operacional que hace alusión a un tamaño de partícula característico que se acumula dentro de los molinos SAG's) generados por ambos molinos son enviados a una plata de chancado dedicada, posteriomente este mineral ya chancado se reporta en el molino de bolas unitario de la seccion n°19 (conocido también como quinto molino). Adicionalmente este molino unitario es alimentado con pulpa proveniente de los cajones que alimentan a las baterías de hidrociclones (BHC) de ambas líneas SAG. El diagrama de flujo de la sección de molienda A-2 se puede ver a continuación, en la Figura 4:

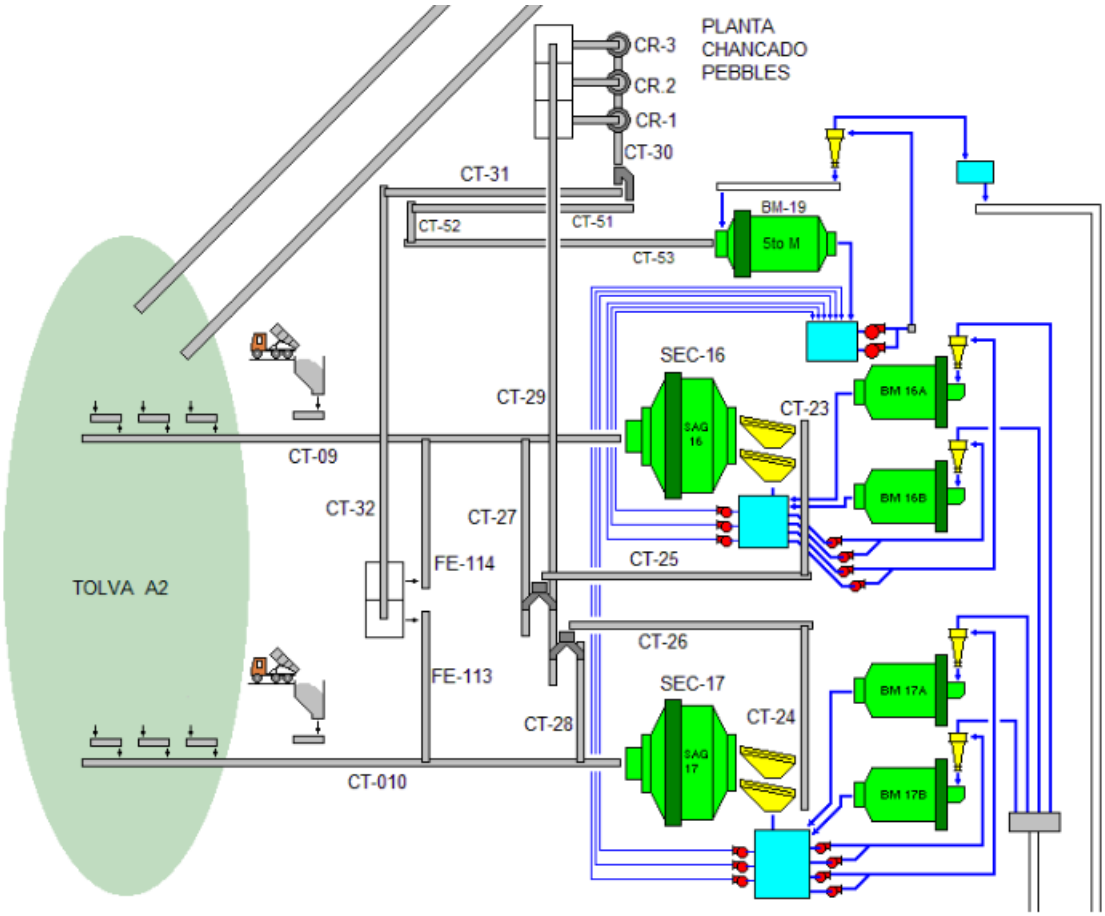


Figura 4: Ezquema del Circuito de Molienda SAG A-2 de DCHU

Los principales equipos que componen la sección de molienda A-2 son los siguientes:

- Molinos semi autogenos SAG n° 16 Y 17.
- Molinos de bolas BM n°16a, 16b y 17a, 17b.
- Baterías de hidrociclones BHC n°16a, 16b y 17a. 17b (equipos clasificadores).
- Cajones de alimentación a las baterías de hidrociclones, con sus respectivas bombas de impulsión.
- Planta de Chancado de Pebbles, compuesta por 3 chancadores de cono.
- Quinto Molino BM19.

OBJETIVO GENERAL

El principal objetivo de este trabajo es desarrollar un modelo de rendimiento para un Molino SAG vía métodos de “Machine Learning”, el cual permita maximizar el poder predictivo de la tasa de procesamiento fresca en función de variables operacionales medidas en línea.

OBJETIVOS ESPECÍFICOS

- Recopilación de Datos
- Realizar resúmenes estadísticos descriptivos de los datos.
- Limpieza y Preprocesamiento de datos.
- Modelamiento de datos.
- Evaluación de Modelos.
- Comparación de Modelos.
- Ensamble final.
- Predecir la tasa de procesamiento horaria de un molino semi-autógeno (SAG).

METODOLOGIA

La planificación del presente trabajo se dividió en las siguientes etapas:

RECOPILACIÓN DE DATOS

Con el propósito de entender la variabilidad operacional se extrajo data de operación histórica en un período que comprendió desde el 01-01-2017 hasta el 16-10-2018 (22 meses de operación) en una frecuencia de 5 minutos. La data se extrajo sin ningun tipo de pre procesamiento.

La data recopilada consideró ambos molinos SAG y sus respectivos circuitos, sin embargo y dado que los molinos oepran en paralelo y no hay diferencias en la cantidad de datos válidos para ambos circuitos, solo se trabajara con los datos del molino SAG n°16, sin embargo el análisis realizado puede ser fácilmente extendido al molino sag n°17,

Algunas abreviaturas utilizadas se listan a continuación:

- tph: toneladas por hora con recirculación.
- tph_f: alimentación por hora fresca.
- bajo/sobre_Xplg: medida del tamaño del mineral que alimenta al molino SAG. Porcentaje menor o mayor (segun corresponda) a "X" pulgada.
- hoy: en referencia al día actual.
- cp: porcentaje de sólidos.
- mw: mega watts.
- kw: kilo watts.
- h2o: agua.
- fe104: alimentador n°104.
- rpm: revoluciones por minuto.
- kgcm2: presión en los descansos del molino SAG.
- bhc: batería de hidrociclones.
- m3h: metros cúbicos por hora.
- psi: unidad de presión.
- bm: molino de bolas.
- bm18: molino de bolas sección n°18.
- 65ty: porcentaje del flujo mineral que esta sobre el tamaño de 65 malla tyler (aprox. 212 micrones)

EXPLORACIÓN DESCRIPTIVA

A continuación se exploran las principales características de los datos:

```
# Dimensiones
dim(data)
```

```
[1] 188221     24
```

Se puede apreciar que existen 24 variables con aproximadamente 189 mil datos por cada variable, correspondiente a 2 años de operación con observaciones cada 5 minutos.

```
# Estructura
str(data)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':   188221 obs. of  24 variables:
 $ fecha               : POSIXct, format: "2017-01-01 00:00:00" "2017-01-01 00:05:00" ...
 $ tph_f_sag16         : num  1797 1380 1439 1398 1402 ...
 $ bajo_1plg_sag16     : num  57 71.4 79.5 78.5 77.4 ...
 $ una_cuatro_plg_sag16: num  28.8 21.3 18.7 17.6 17.8 ...
 $ sobre_4plg_sag16    : num  14.14 7.27 1.83 3.93 4.79 ...
 $ bajo_1/2plg_sag16   : num  36.9 36.2 36.8 37.7 38.6 ...
 $ h2o_m3h_sag16       : num  742 742 742 742 742 ...
 $ cp_sag16            : num  74.8 74.8 74.8 75.5 75.5 ...
 $ rpm_sag16           : num  9.39 9.34 9.23 9.1 9.08 ...
 $ mw_sag16            : num  6.61 6.54 6.43 6.34 6.26 ...
 $ kgcm2_sag16         : num  57.1 57 56.9 56.9 56.9 ...
 $ peso_tph_fe104      : num  152 152 152 152 152 152 152 152 152 152 ...
 $ h2o_m3h_bhc_sag16   : num  963 558 524 559 596 ...
 $ nivel_cajon_sag16   : num  70.2 70.2 70.2 76.2 76.2 ...
 $ kw_bm_16a           : num  3078 3222 3201 3224 3212 ...
 $ kw_bm_16b           : num  3111 3111 3111 3111 3111 ...
 $ cp_bhc_16a          : num  69.5 69.5 68.1 68.1 68.1 ...
 $ cp_bhc_16b          : num  70.1 70.1 70.1 70.1 70.1 ...
 $ m3h_bhc_16a         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ m3h_bhc_16b         : num  7583 7583 7583 7583 7583 ...
 $ psi_bhc_16a         : num  12.3 13.2 13.2 12.8 13.3 ...
 $ psi_bhc_16b         : num  11.5 11.5 11.5 11.5 11.5 ...
 $ 65ty_bhc_16a        : num  28.3 28.3 28.3 28.3 28.3 ...
 $ 65ty_bhc_16b        : num  28.1 28.1 28.1 28.1 28.1 ...
```

Todas las variables son numéricas a escepción de la variable "fecha" que tiene unidades de fecha.

```
# Resumen
summary(data)
```

```
     fecha                       tph_f_sag16     bajo_1plg_sag16
 Min.   :2017-01-01 00:00:00   Min.   :-154.4   Min.   : 10.86
 1st Qu.:2017-06-13 08:15:00   1st Qu.:1794.4   1st Qu.: 34.69
 Median :2017-11-23 18:30:00   Median :1973.3   Median : 51.23
 Mean   :2017-11-23 18:13:17   Mean   :1806.8   Mean   : 52.56
 3rd Qu.:2018-05-06 03:45:00   3rd Qu.:2163.8   3rd Qu.: 68.77
 Max.   :2018-10-16 13:00:00   Max.   :3150.0   Max.   :100.00
                               NA's   :2451     NA's   :1996
 una_cuatro_plg_sag16 sobre_4plg_sag16 bajo_1/2plg_sag16 h2o_m3h_sag16
 Min.   : 0.00        Min.   : 0.000   Min.   : 4.322    Min.   :   0.0
 1st Qu.:24.33        1st Qu.: 6.363   1st Qu.:14.710    1st Qu.: 433.0
 Median :33.69        Median :15.253   Median :22.317    Median : 550.0
 Mean   :30.17        Mean   :16.970   Mean   :25.887    Mean   : 494.7
 3rd Qu.:37.86        3rd Qu.:26.006   3rd Qu.:33.064    3rd Qu.: 618.1
 Max.   :58.10        Max.   :71.057   Max.   :84.222    Max.   :1200.0
 NA's   :1996         NA's   :1996     NA's   :2450      NA's   :2056
    cp_sag16       rpm_sag16         mw_sag16      kgcm2_sag16
 Min.   : 0.00   Min.   : 0.000   Min.   :0.000   Min.   :  0.00
 1st Qu.:75.00   1st Qu.: 8.255   1st Qu.:5.524   1st Qu.: 59.66
 Median :77.00   Median : 9.223   Median :6.444   Median : 61.66
 Mean   :70.28   Mean   : 8.375   Mean   :5.895   Mean   : 58.90
 3rd Qu.:78.80   3rd Qu.: 9.857   3rd Qu.:7.170   3rd Qu.: 63.73
 Max.   :98.00   Max.   :14.981   Max.   :8.754   Max.   :139.65
 NA's   :2057    NA's   :2038     NA's   :2037    NA's   :2043
 peso_tph_fe104  h2o_m3h_bhc_sag16 nivel_cajon_sag16   kw_bm_16a
 Min.   :  0.0   Min.   :   0.0    Min.   :  0.0     Min.   :   0
 1st Qu.:  0.0   1st Qu.: 497.5    1st Qu.: 59.6     1st Qu.:2988
 Median :  0.0   Median : 725.5    Median : 66.0     Median :3127
 Mean   :144.7   Mean   : 654.7    Mean   : 64.3     Mean   :2975
 3rd Qu.:300.0   3rd Qu.: 836.0    3rd Qu.: 71.0     3rd Qu.:3235
 Max.   :936.0   Max.   :1351.8    Max.   :100.0     Max.   :3992
 NA's   :2350    NA's   :2052      NA's   :2095      NA's   :10331
   kw_bm_16b      cp_bhc_16a       cp_bhc_16b      m3h_bhc_16a
 Min.   :   0   Min.   :  0.00   Min.   :  0.00   Min.   :   0.0
 1st Qu.:2972   1st Qu.: 68.00   1st Qu.: 59.00   1st Qu.:   0.0
 Median :3105   Median : 71.00   Median : 64.00   Median :   0.0
 Mean   :2968   Mean   : 68.12   Mean   : 63.14   Mean   : 650.8
 3rd Qu.:3227   3rd Qu.: 74.97   3rd Qu.: 70.68   3rd Qu.: 117.0
 Max.   :4406   Max.   :100.00   Max.   :100.00   Max.   :6000.0
 NA's   :9000   NA's   :3801     NA's   :15492    NA's   :2713
  m3h_bhc_16b    psi_bhc_16a      psi_bhc_16b    65ty_bhc_16a
 Min.   :   0   Min.   : 0.000   Min.   : 0.0   Min.   : 0.000
 1st Qu.:7541   1st Qu.: 9.621   1st Qu.:10.0   1st Qu.: 0.081
 Median :7568   Median :10.971   Median :11.0   Median :11.880
 Mean   :7572   Mean   : 9.991   Mean   :10.2   Mean   :13.699
 3rd Qu.:7601   3rd Qu.:12.000   3rd Qu.:12.0   3rd Qu.:26.480
 Max.   :8533   Max.   :21.672   Max.   :21.0   Max.   :40.000
 NA's   :2487   NA's   :4704     NA's   :4692   NA's   :5699
  65ty_bhc_16b
 Min.   : 0.000
 1st Qu.: 0.000
 Median : 0.000
 Mean   : 9.688
 3rd Qu.:22.540
 Max.   :39.990
 NA's   :16848
```

Se pueden apreciar que en todas las variables existen observaciones no disponibles o NA's (not available) asi como gran cantidad de ceros (por ejemplo: 65ty_bhc_16b). Lo anterior nos indica que es necesario un proceso de limpieza preliminar de la base de datos. En total, existe un % de datos no disponibles, en donde el desgloce por variable se puede ver a continuación:

```
# NA's por variable
data %>%
miss_var_summary(order = FALSE) %>%
  datatable(colnames = c('Variable', 'total n° NA', 'total % NA')) %>%
  formatRound(columns=c('pct_miss'), digits=2)
```

Se puede apreciar que algunas variables poseen gran cantidad de NA's. Dado lo anterior se define que aquellas variables que tienen un porcentaje de NA´s > 8% se eliminarán del conjunto de datos. Estas variables serían las siguientes:

```
#Variables con %NA > 8%
data %>%
  miss_var_summary(order = FALSE) %>%
  filter(pct_miss>=8) %>%
  datatable(colnames = c('Variable','total n° NA', 'total % NA')) %>%
  formatRound(columns=c('pct_miss'), digits=2)
```

Como complemento a los resúmenes antes vistos a continuación se muestra la desviación estándar por variable:

```
# Desviación estándard
data.frame(Desv_Estandar=sapply(data[,2:24], sd, na.rm=TRUE)) %>%
  datatable() %>%
  formatRound(columns=c('Desv_Estandar'), digits=2)
```

También se muestra la asimetría por variable:

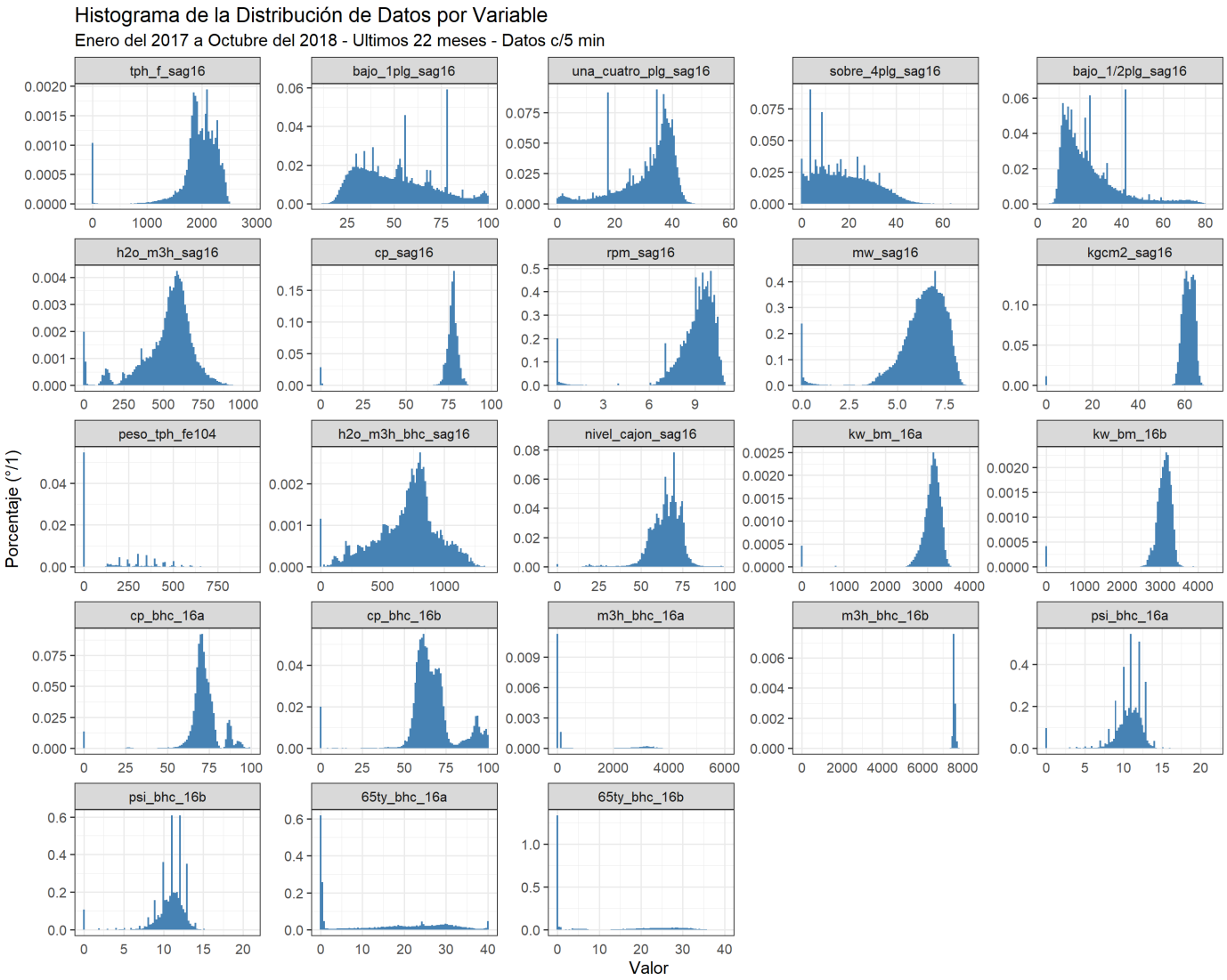
```
# Asimetria
data.frame(Asimetria=sapply(data[,2:24], skewness, na.rm=TRUE)) %>%
  datatable() %>%
   formatRound(columns=c('Asimetria'), digits=3)
```

En algunos casos se aprecian estadísticos con valores muy extremos, lo cual queda de manifiesto (en parte) con los valores de asimetría negativa (cola izquierda) reportados. Estos razgos se podrán apreciar/confirmar en la exploración gráfica.

EXPLORACION GRÁFICA

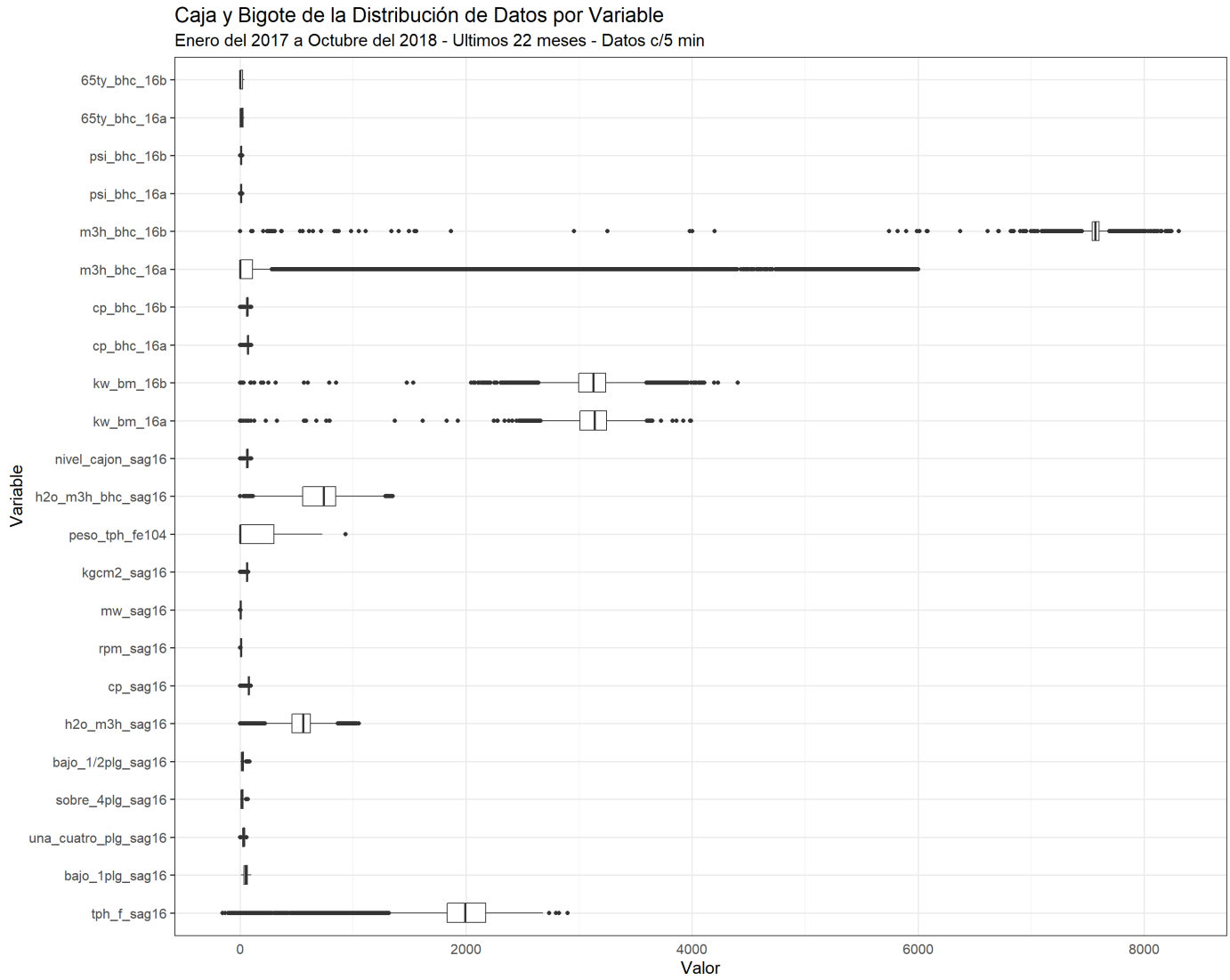
A continuación se aprecian una serie de gráficas que nos entregaran mas información respecto a las diferentes distribuciones de los datos:

```
# Histogramas
data %>%
  dplyr::select(-fecha) %>%
  na.omit() %>%
  stack() %>%
  ggplot(aes(x=values, y=stat(density), fill=values))+
  geom_histogram(bins = 100, fill='steelblue')+
  facet_wrap(~ind, scales = "free")+
  labs(title = "Histograma de la Distribución de Datos por Variable",
       subtitle = "Enero del 2017 a Octubre del 2018 - Ultimos 22 meses - Datos c/5 min",
       y = "Porcentaje (°/1)",
       x = "Valor")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



Los histogramas nos confirman lo expuesto en el análisis descriptivo, esto es, que muchas distribuciones son asimétricas con colas negativas y otras estan muy concentradas en itervalos de valores acotados. Se aprecia que en casi todas las variables existe cierta concentración en torno a cero, lo cual tentativamente daría cuenta de los tiempos de detención de los equipos.

```
#BoxPlots
data %>%
  dplyr::select(-fecha) %>%
  na.omit() %>%
  stack() %>%
  ggplot(aes(x=ind, y=values))+
  geom_boxplot(width=0.5)+
  coord_flip()+
  labs(title = "Caja y Bigote de la Distribución de Datos por Variable",
       subtitle = "Enero del 2017 a Octubre del 2018 - Ultimos 22 meses - Datos c/5 min",
       y = "Valor",
       x = "Variable")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



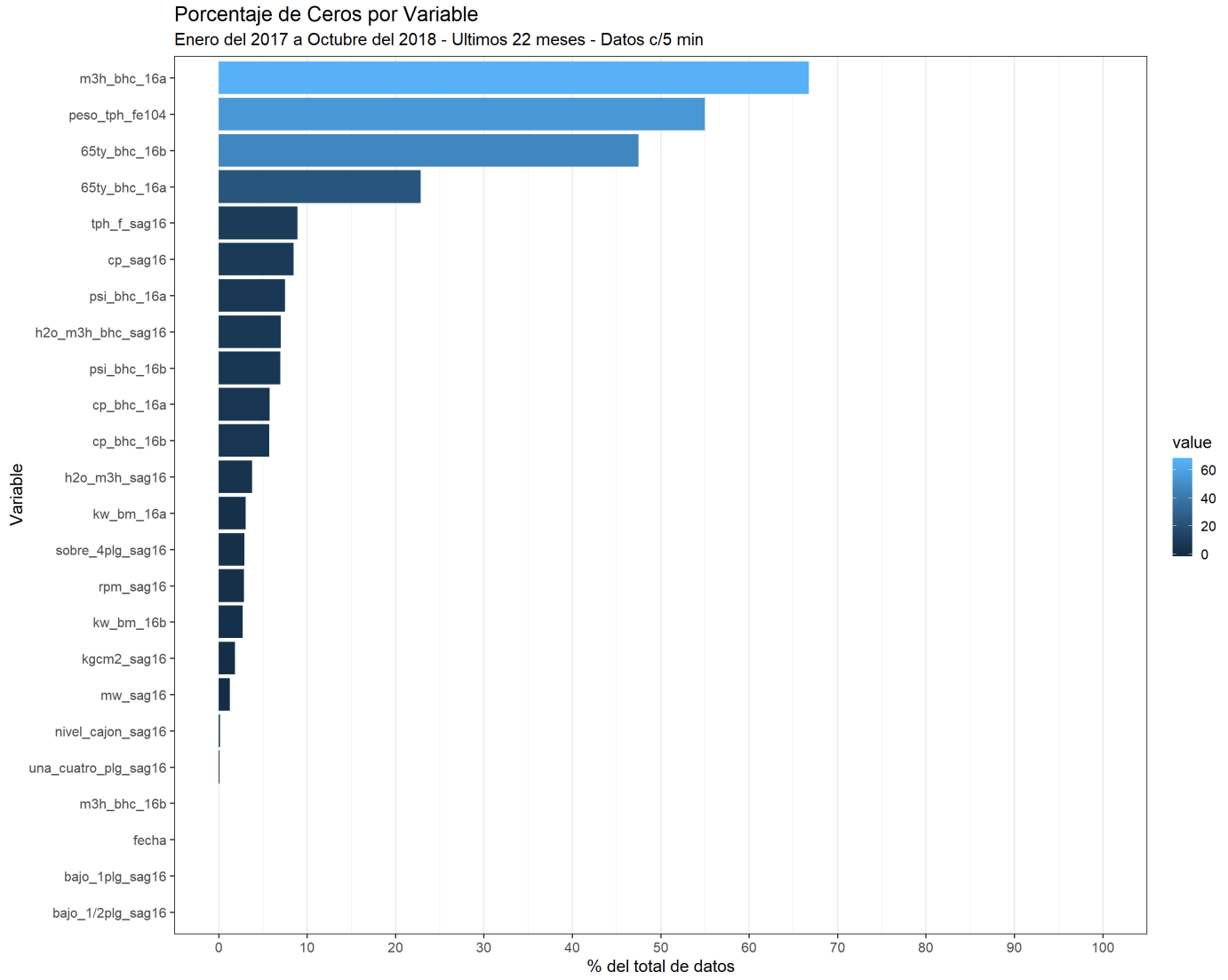
De los gráficos de caja y bigote, se puede apreciar una gran cantidad de datos atípicos en todas las variables, asi como también muchos “ceros” y una diferencia significativa en las escalas de estas.

La principal característica observada radica en la existencia de una fuerte asimetría negativa (cola izquierda) de los datos, lo cual nos indica la existencia de períodos de operación cuyas mediciones se reportan hacia el intervalo izquierdo de los indicadores de tendencia central (media y mediana). Este tipo de variabilidad en la operación puede tener diferentes causas, por mencionar algunas: motivos de índole operacional que impliquen un procesamiento menor al habitual, paradas programadas, paradas no programadas, entre otras.

También se observan concentraciones no despreciables de observaciones reportadas como “ceros” y/o cercanas a “cero” en todas las variables, incluso se aprecian observaciones negativas en el caso de la alimentación a los molinos SAG’s (tph_f_sag16). Estas lecturas (“ceros”), pueden tener su origen ya sea en mediciones efectivas, lo que implicaría un indicativo del tiempo total que el equipo NO estubo disponible para operar (disponibilidad), o que estubo disponible para operar pero que por diversas razones no operó (utilización), o a fallas en los instrumentos de medición en el caso de los valores negativos (en cuyo caso estas observaciones deben considerarse como NA’s).

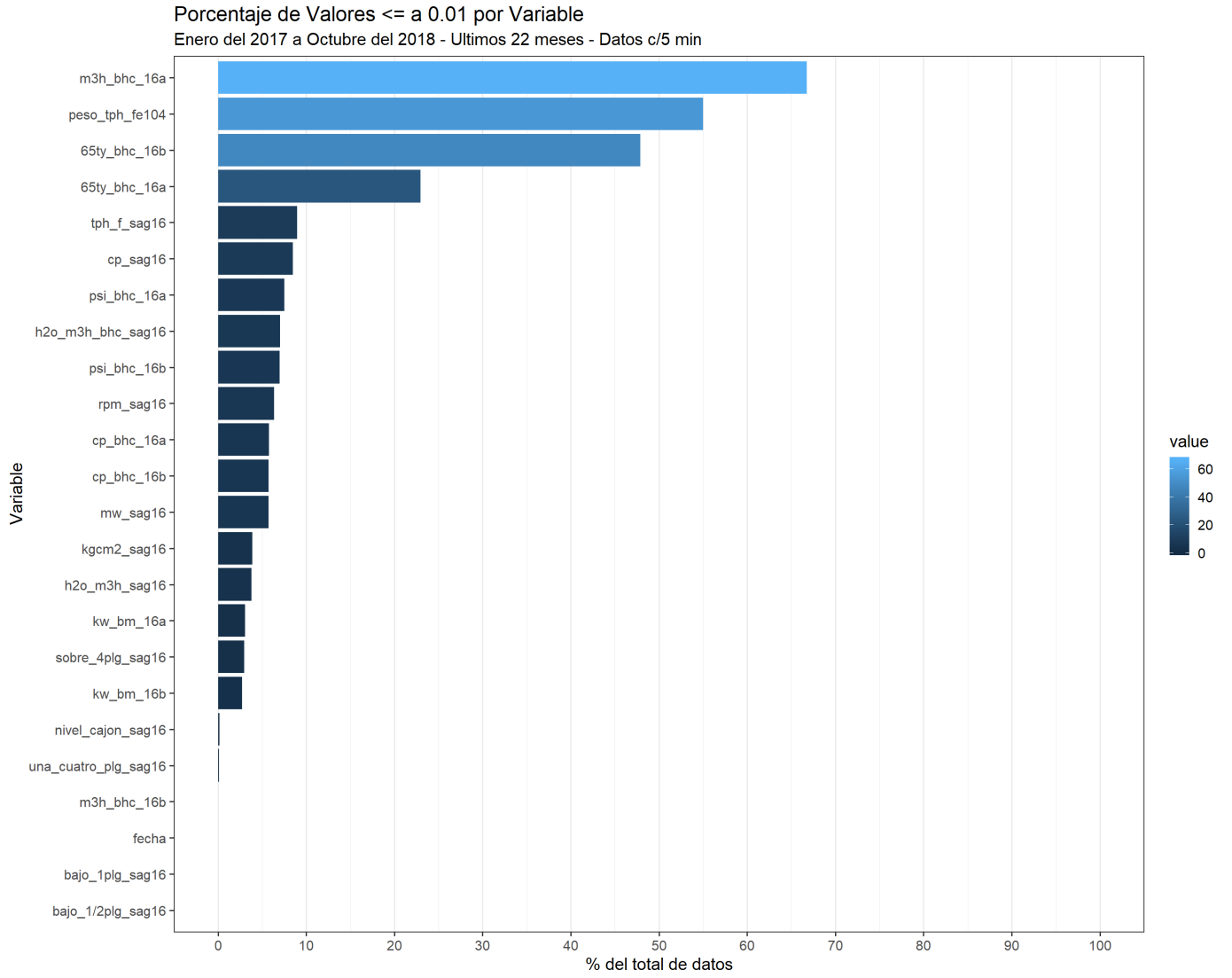
En el gráfico que se muestra a continuación se puede apreciar de forma mas clara el porcentaje del tiempo total en que las diferentes variables reportaron “ceros”:

```
# Porcentaje de ceros por Variable
data %>%
  lapply(function(x){ length(which(x==0))/length(x)*100}) %>%
  melt() %>%
  ggplot(aes(x=reorder(L1,value), y=value, fill=value))+
  geom_col()+
  coord_flip()+
  scale_y_continuous(breaks = c(0, 10, 20, 30, 40, 50, 60, 70, 80, 90,100),  limits=c(0, 100))+
  theme(panel.grid.major.y = element_blank())+
  labs(title = "Porcentaje de Ceros por Variable",
       subtitle = "Enero del 2017 a Octubre del 2018 - Ultimos 22 meses - Datos c/5 min",
       y = "% del total de datos",
       x = "Variable")+
  theme_bw(base_size = 15)+
  theme(panel.grid.major.y = element_blank())
```



Cabe señalar que adicionalmente se reporta una cantidad no menor de observaciones cuyos valores están cerca de cero y/o son negativos, los cuales también influyen en el grado de asimetría negativa observada en las gráficas antes mostradas. Por ende a continuación se muestra el porcentaje del tiempo total en que las diferentes variables reportaron valores menores e iguales a “0.01” (valor elegido arbitrariamente):

```
# Porcentaje de valores cercanos a cero por variable
data %>%
  lapply(function(x){ length(which(x<=0.01))/length(x)*100}) %>%
  melt() %>%
  ggplot(aes(x=reorder(L1,value), y=value, fill=value))+
  geom_col()+
  coord_flip()+
  scale_y_continuous(breaks = c(0, 10, 20, 30, 40, 50, 60, 70, 80, 90,100),  limits=c(0, 100))+
  theme(panel.grid.major.y = element_blank())+
  labs(title = "Porcentaje de Valores <= a 0.01 por Variable",
       subtitle = "Enero del 2017 a Octubre del 2018 - Ultimos 22 meses - Datos c/5 min",
       y = "% del total de datos",
       x = "Variable")+
  theme_bw(base_size = 15)+
  theme(panel.grid.major.y = element_blank())
```



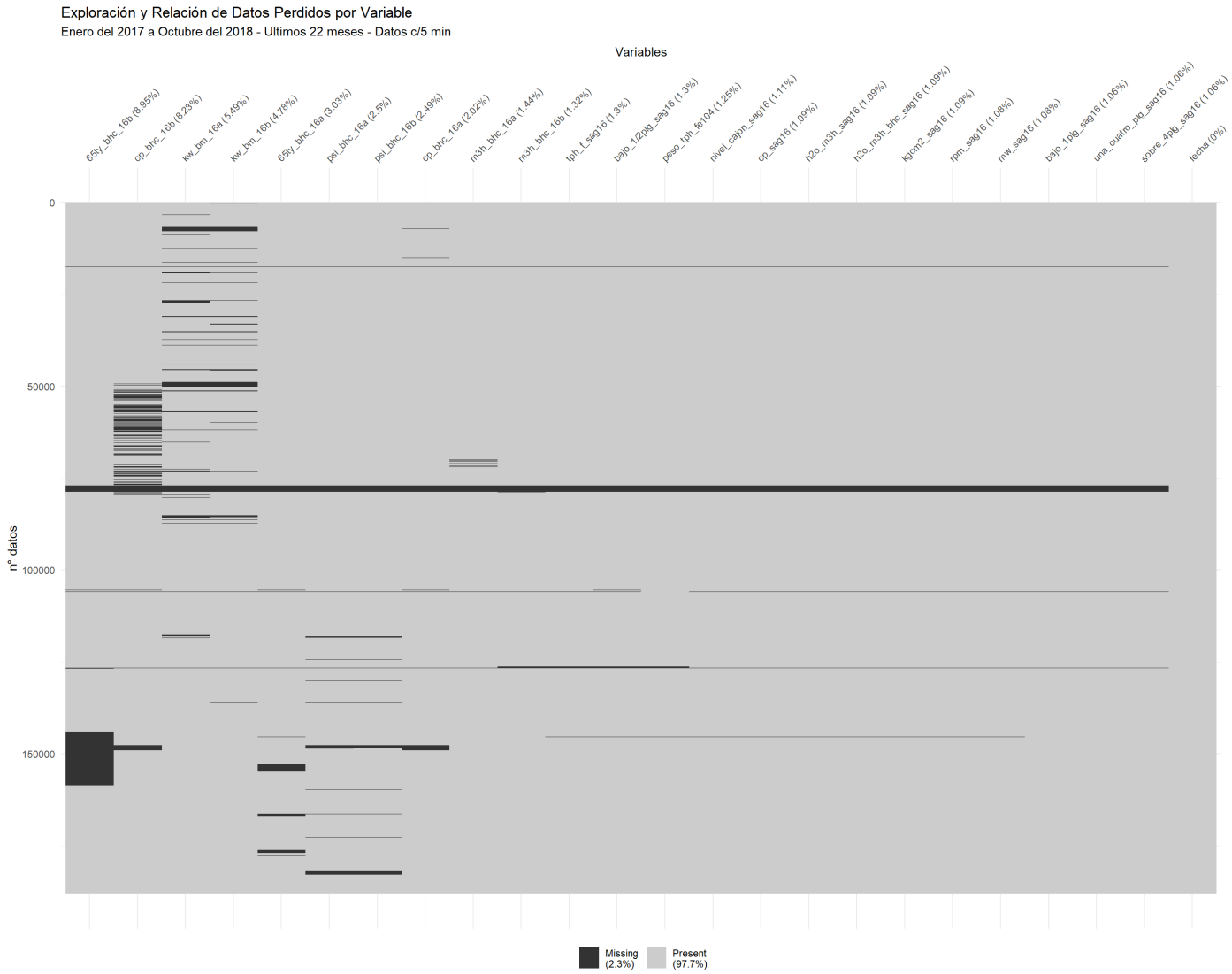
Se aprecia que el porcentaje de datos en este intervalo de valores aumenta, pero el orden no se altera. Se observa que son 4 las variables que poseen mayor cantidad de valores menores a 0.01, a decir: **m3h_bhc_16a**, **peso_tph_fe104**, **65ty_bhc_16a** y **65ty_bhc_16b**, cuyos porcentajes superan el 20% de los datos. Para el resto de las variables los porcentajes no superan el 10% del total.

Otro punto importante a considerar tiene razón con los datos perdidos (NA's), los cuales corresponden a observaciones que debieron ser registrados pero que por diversas razones (usualmente fallas en los instrumentos de medición) no fueron leídos.

En general estos datos perdidos o NA's (en función de la cantidad que exista) se pueden reemplazar o descartar, aunque es de vital importancia considerar que los métodos de eliminación y/o reemplazo que se utilizen, pueden tener efectos determinantes sobre las conclusiones extraídas desde los análisis posteriores.

Un resumen gráfico de este tipo de observaciones se muestra a continuación:

```
# Grafico de NA's
data %>%
  vis_miss(sort_miss=TRUE,show_perc = TRUE, warn_large_data = FALSE)+
  labs(title = "Exploración y Relación de Datos Perdidos por Variable",
       subtitle = "Enero del 2017 a Octubre del 2018 - Ultimos 22 meses - Datos c/5 min",
       y = "n° datos",
       x = "Variables")
```



Se puede apreciar que para el conjunto completo de datos el porcentaje de datos perdidos se mueve en torno al 2%, además se observa que existe un período común de datos perdidos (líneas negras horizontales) a todas las variables.

Sin embargo en este análisis deben considerarse además aquellas observaciones que reportaron “ceros” y valores numéricos muy distantes al resto (outliers), lo cual se aborda en el capitulo siguiente.

Conclusiones Exploración Descriptiva y Gráfica

1. Se observa una fuerte asimetría negativa en la mayoría de las variables.
2. Se observa gran cantidad de datos atípicos, para la mayoría de las variables.
3. Las variables; **m3h_bhc_16a**, **peso_tph_fe104**, **65ty_bhc_16a** y **65ty_bhc_16b** contienen un porcentaje de datos cercanos a cero mayor al 20%. Por lo tanto estas variables son candidatas a ser eliminadas dada la poca información que aportan. Para el resto de las varaibles los porcentajes dedatos cercanos a cero no superan el 10% del total.
4. El porcentaje total de NA’s es de 2.3% del total de la base de datos. Con dos variables que superan el 8% de NA´s;**65ty_bhc_16b** y **cp_bhc_16b**, y que al igual que en el punto anterior son candidatas a ser eliminadas debido a la mala calidad en la información que aportan.
5. En la etapa de pre-procesamiento se podrian abordar las siguientes estrategias:

- Normalizar las variables para eliminar el efecto de diferentes escalas.
- Estandarizar para eliminar el efecto de diferentes distribuciones.
- Aplicar una transformación a los datos de las variables predictoras (por ejemplo BoxCox) para evaluar su efecto en las

distribuciones con alta asimetría.

LIMPIEZA DE DATOS

Esta etapa da cuenta de diferentes acciones que apuntan a "limpiar" el conjunto de datos de aquellas observaciones perdidas y/o anómalas, las cuales no son de interés y además podrían interferir en el análisis estadístico a realizar. Este tipo de información da cuenta de:

- Variables con alto % de "ceros" y NA´s
- Data perdida o NA (Not Available)
- Outliers o valores atípicos

Eliminación de variables con alto porcentaje de ceros

Las variables; **m3h_bhc_16a**, **peso_tph_fe104**, **65ty_bhc_16a** y **65ty_bhc_16b** contienen un % mayor al 20% de datos menores a 0.01 por ende se eliminarán del conjunto de datos al igual que la variable **cp_bhc_16b** dado que supera el 8% de NA´s. Tambien se elimina la variable **fecha** ya que no se ocupara de acá en adelante.

```
# Eliminacion de columnas con % "ceros">20%
data<-data %>%
  dplyr::select(-m3h_bhc_16a, -peso_tph_fe104, -'65ty_bhc_16a', -'65ty_bhc_16b', -cp_bhc_16b, -fecha)
summary (data)
```

```
  tph_f_sag16     bajo_1plg_sag16  una_cuatro_plg_sag16 sobre_4plg_sag16
 Min.   :-154.4   Min.   : 10.86   Min.   : 0.00        Min.   : 0.000
 1st Qu.:1794.4   1st Qu.: 34.69   1st Qu.:24.33        1st Qu.: 6.363
 Median :1973.3   Median : 51.23   Median :33.69        Median :15.253
 Mean   :1806.8   Mean   : 52.56   Mean   :30.17        Mean   :16.970
 3rd Qu.:2163.8   3rd Qu.: 68.77   3rd Qu.:37.86        3rd Qu.:26.006
 Max.   :3150.0   Max.   :100.00   Max.   :58.10        Max.   :71.057
 NA's   :2451     NA's   :1996     NA's   :1996         NA's   :1996
 bajo_1/2plg_sag16 h2o_m3h_sag16       cp_sag16       rpm_sag16
 Min.   : 4.322    Min.   :   0.0   Min.   : 0.00   Min.   : 0.000
 1st Qu.:14.710    1st Qu.: 433.0   1st Qu.:75.00   1st Qu.: 8.255
 Median :22.317    Median : 550.0   Median :77.00   Median : 9.223
 Mean   :25.887    Mean   : 494.7   Mean   :70.28   Mean   : 8.375
 3rd Qu.:33.064    3rd Qu.: 618.1   3rd Qu.:78.80   3rd Qu.: 9.857
 Max.   :84.222    Max.   :1200.0   Max.   :98.00   Max.   :14.981
 NA's   :2450      NA's   :2056     NA's   :2057    NA's   :2038
    mw_sag16       kgcm2_sag16     h2o_m3h_bhc_sag16 nivel_cajon_sag16
 Min.   :0.000    Min.   :  0.00   Min.   :   0.0    Min.   :  0.0
 1st Qu.:5.524    1st Qu.: 59.66   1st Qu.: 497.5    1st Qu.: 59.6
 Median :6.444    Median : 61.66   Median : 725.5    Median : 66.0
 Mean   :5.895    Mean   : 58.90   Mean   : 654.7    Mean   : 64.3
 3rd Qu.:7.170    3rd Qu.: 63.73   3rd Qu.: 836.0    3rd Qu.: 71.0
 Max.   :8.754    Max.   :139.65   Max.   :1351.8    Max.   :100.0
 NA's   :2037     NA's   :2043     NA's   :2052      NA's   :2095
   kw_bm_16a        kw_bm_16b       cp_bhc_16a       m3h_bhc_16b
 Min.   :   0     Min.   :   0   Min.   :  0.00   Min.   :   0
 1st Qu.:2988     1st Qu.:2972   1st Qu.: 68.00   1st Qu.:7541
 Median :3127     Median :3105   Median : 71.00   Median :7568
 Mean   :2975     Mean   :2968   Mean   : 68.12   Mean   :7572
 3rd Qu.:3235     3rd Qu.:3227   3rd Qu.: 74.97   3rd Qu.:7601
 Max.   :3992     Max.   :4406   Max.   :100.00   Max.   :8533
 NA's   :10331    NA's   :9000   NA's   :3801     NA's   :2487
  psi_bhc_16a      psi_bhc_16b
 Min.   : 0.000   Min.   : 0.0
 1st Qu.: 9.621   1st Qu.:10.0
 Median :10.971   Median :11.0
 Mean   : 9.991   Mean   :10.2
 3rd Qu.:12.000   3rd Qu.:12.0
 Max.   :21.672   Max.   :21.0
 NA's   :4704     NA's   :4692
```

Imputación de NA

Para afrontar este problema existen 2 posibilidades:

- Eliminar todas las observaciones (filas) que contengan algun NA.
- Reemplazar los NA mediante técnicas de imputación.

En nuestro caso nos guiaremos por el segundo criterio, en donde se realizará una imputación mediante reemplazo por mediana. Se eligió este indicador estadístico dado que en general no se ve afectado por los datos extremos (como si lo es el promedio).

```
#Imputacion por mediana
data_imp <- data %>%
    bind_shadow() %>%
    impute_median_all()

data_imp_median <- shadow_long(data_imp, tph_f_sag16, bajo_1plg_sag16, una_cuatro_plg_sag16,  sobre_4plg_sag
16, `bajo_1/2plg_sag16`,
                                    h2o_m3h_sag16, cp_sag16, rpm_sag16, mw_sag16, kgcm2_sag16, h2o_m3h_bhc
_sag16, nivel_cajon_sag16, kw_bm_16a, kw_bm_16b,
                                    cp_bhc_16a, m3h_bhc_16b, psi_bhc_16a, psi_bhc_16b)
```

```
summary(data_imp[,1:18])
```

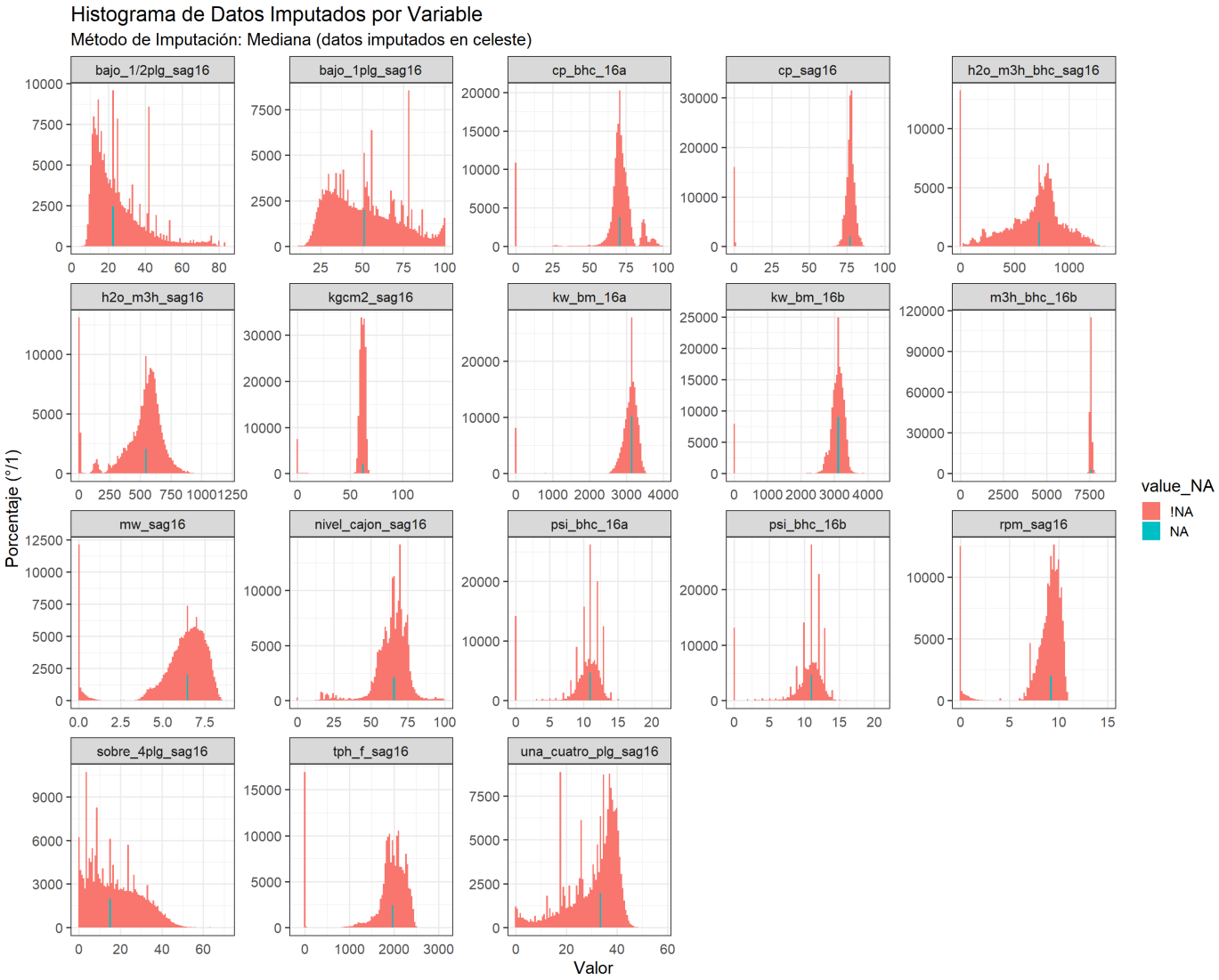
```
  tph_f_sag16     bajo_1plg_sag16  una_cuatro_plg_sag16 sobre_4plg_sag16
 Min.   :-154.4   Min.   : 10.86   Min.   : 0.00        Min.   : 0.000
 1st Qu.:1797.4   1st Qu.: 34.84   1st Qu.:24.56        1st Qu.: 6.455
 Median :1973.3   Median : 51.23   Median :33.69        Median :15.253
 Mean   :1809.0   Mean   : 52.55   Mean   :30.20        Mean   :16.952
 3rd Qu.:2160.4   3rd Qu.: 68.47   3rd Qu.:37.82        3rd Qu.:25.864
 Max.   :3150.0   Max.   :100.00   Max.   :58.10        Max.   :71.057
 bajo_1/2plg_sag16 h2o_m3h_sag16       cp_sag16       rpm_sag16
 Min.   : 4.322    Min.   :   0.0   Min.   : 0.00   Min.   : 0.000
 1st Qu.:14.788    1st Qu.: 435.4   1st Qu.:75.00   1st Qu.: 8.276
 Median :22.316    Median : 550.0   Median :77.00   Median : 9.223
 Mean   :25.841    Mean   : 495.3   Mean   :70.36   Mean   : 8.384
 3rd Qu.:32.765    3rd Qu.: 617.5   3rd Qu.:78.70   3rd Qu.: 9.849
 Max.   :84.222    Max.   :1200.0   Max.   :98.00   Max.   :14.981
    mw_sag16      kgcm2_sag16     h2o_m3h_bhc_sag16 nivel_cajon_sag16
 Min.   :0.000   Min.   :  0.00   Min.   :   0.0    Min.   :  0.00
 1st Qu.:5.539   1st Qu.: 59.69   1st Qu.: 499.8    1st Qu.: 59.70
 Median :6.444   Median : 61.66   Median : 725.5    Median : 66.00
 Mean   :5.901   Mean   : 58.93   Mean   : 655.5    Mean   : 64.32
 3rd Qu.:7.161   3rd Qu.: 63.71   3rd Qu.: 835.0    3rd Qu.: 71.00
 Max.   :8.754   Max.   :139.65   Max.   :1351.8    Max.   :100.00
   kw_bm_16a      kw_bm_16b      cp_bhc_16a      m3h_bhc_16b
 Min.   :   0   Min.   :   0   Min.   :  0.00   Min.   :   0
 1st Qu.:2999   1st Qu.:2980   1st Qu.: 68.00   1st Qu.:7542
 Median :3127   Median :3105   Median : 71.00   Median :7568
 Mean   :2983   Mean   :2974   Mean   : 68.18   Mean   :7572
 3rd Qu.:3228   3rd Qu.:3221   3rd Qu.: 74.84   3rd Qu.:7601
 Max.   :3992   Max.   :4406   Max.   :100.00   Max.   :8533
  psi_bhc_16a     psi_bhc_16b
 Min.   : 0.00   Min.   : 0.00
 1st Qu.: 9.68   1st Qu.:10.00
 Median :10.97   Median :11.00
 Mean   :10.02   Mean   :10.22
 3rd Qu.:12.00   3rd Qu.:12.00
 Max.   :21.67   Max.   :21.00
```

Desde ambos resumenes se puede apreciar que no hay diferencias relevantes para los promedios de las variables, lo anterior se corrobora mediante el cómputo de las desviaciones estándar por variable, antes y después del proceso de imputación, en donde las diferencias observadas no son relevantes.

```
# Desviaciones estandar para variables originales e imputadas
Desv_Estandard_pre_imp<-sapply(data, sd, na.rm=TRUE)
Desv_Estandard_post_imp<-sapply(data_imp[,1:18], sd)
data.frame(Desv_Estandard_pre_imp, Desv_Estandard_post_imp) %>%
  datatable() %>%
  formatRound(columns=c('Desv_Estandard_pre_imp','Desv_Estandard_post_imp'), digits=2)
```

En complemento al análisis anterior, a continuación se pueden apreciar la posición de los datos imputados (barras en color celeste):

```
#Grafica de Imputación
ggplot(data_imp_median, aes(x=value, fill=value_NA))+
  geom_histogram(bins = 100)+
  facet_wrap(~variable, scales = "free")+
  labs(title = "Histograma de Datos Imputados por Variable",
       subtitle = "Método de Imputación: Mediana (datos imputados en celeste)",
       y = "Porcentaje (°/1)",
       x = "Valor")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



Finalmente corroboramos que nuestro conjunto de datos post-limpieza no posee NA's:

```
# % de NA's
prop_miss(data_imp[1:18])*100
```

```
[1] 0
```

Datos Atípicos (outliers)

Los valores atípicos dan cuenta principalmente de todos aquellos que definen la marcada asimetría negativa (cola izquierda que se extiende hasta cero) de las distribuciones de las variables en estudio. Se asume por tanto que estos datos dan cuenta, en la mayoría de los casos, de condiciones operacionales atípicas que estan fuera del marco de operación estable del proceso (estado estacionario).

Dado lo anterior, para cada variable, se define un rango o límite bajo y sobre el cual cualquier dato será considerado anómalo y descartado del conjunto de datos:

- Rango inferior: 1er cuartil - 3*IQR
- Rango superior: 3er cuartil + 3*IQR

En donde:

- IQR es el rango intercuartílico de cada variable y se obtiene de la diferencia entre el tercer y primer cuartil.

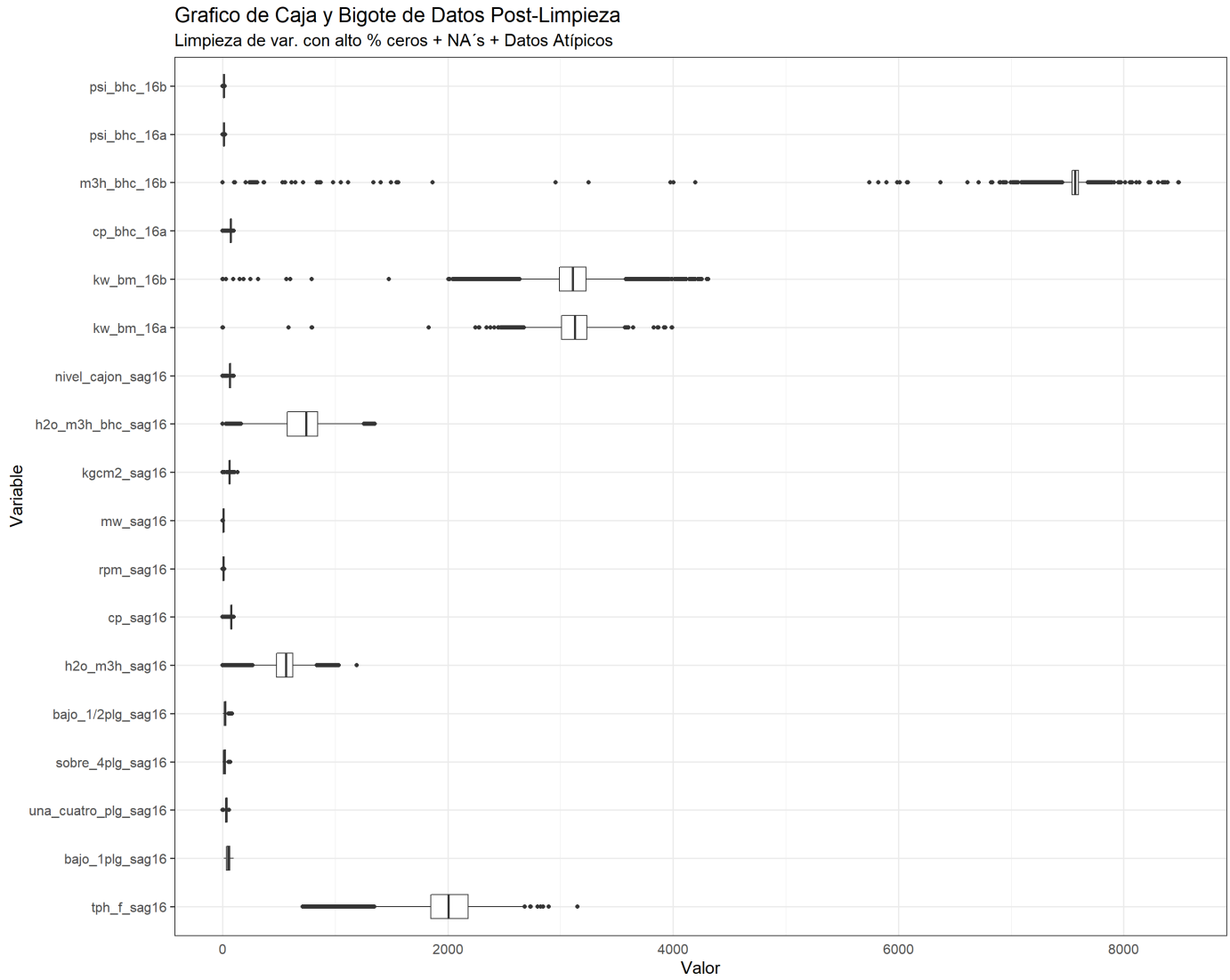
Lo anterior se aplicará solo a la variable de respuesta**tph_f_sag16**. eliminándose todas las observaciones (filas) para las cuales la variable respuesta este fuera de los rangos previamente definidos.

```
# Limite: q1-3*IQR & q3+3*IQR
fct<-3
out <- c((quantile(data_imp$tph_f_sag16, 0.25)-fct*IQR(data_imp$tph_f_sag16)), (quantile(data_imp$tph_f_sag1
6, 0.75)+fct*IQR(data_imp$tph_f_sag16)))

# Filtado de filas que cumplen condicion de variable respuesta
dataclean<-data_imp[,1:18]%>%
  filter (tph_f_sag16>=out[1]&tph_f_sag16<=out[2])



summary(dataclean)
```

```
  tph_f_sag16   bajo_1plg_sag16  una_cuatro_plg_sag16 sobre_4plg_sag16
 Min.   : 712   Min.   : 10.86   Min.   : 0.00        Min.   : 0.000
 1st Qu.:1849   1st Qu.: 33.83   1st Qu.:26.00        1st Qu.: 7.971
 Median :2006   Median : 49.43   Median :34.45        Median :15.859
 Mean   :1993   Mean   : 50.87   Mean   :31.13        Mean   :17.803
 3rd Qu.:2182   3rd Qu.: 65.72   3rd Qu.:38.08        3rd Qu.:26.465
 Max.   :3150   Max.   :100.00   Max.   :56.46        Max.   :69.689
 bajo_1/2plg_sag16 h2o_m3h_sag16       cp_sag16       rpm_sag16
 Min.   : 4.322    Min.   :   0.0   Min.   : 0.00   Min.   : 0.000
 1st Qu.:14.571    1st Qu.: 481.0   1st Qu.:75.90   1st Qu.: 8.616
 Median :21.553    Median : 562.2   Median :77.40   Median : 9.326
 Mean   :24.609    Mean   : 543.3   Mean   :77.17   Mean   : 9.181
 3rd Qu.:30.803    3rd Qu.: 623.7   3rd Qu.:79.00   3rd Qu.: 9.913
 Max.   :84.222    Max.   :1193.2   Max.   :98.00   Max.   :14.414
    mw_sag16      kgcm2_sag16     h2o_m3h_bhc_sag16 nivel_cajon_sag16
 Min.   :0.000   Min.   :  0.00   Min.   :   0.0    Min.   :  2.301
 1st Qu.:5.843   1st Qu.: 60.03   1st Qu.: 573.8    1st Qu.: 60.500
 Median :6.569   Median : 61.84   Median : 744.0    Median : 65.999
 Mean   :6.471   Mean   : 61.86   Mean   : 713.2    Mean   : 65.721
 3rd Qu.:7.229   3rd Qu.: 63.81   3rd Qu.: 845.8    3rd Qu.: 71.001
 Max.   :8.754   Max.   :134.21   Max.   :1350.0    Max.   :100.000
   kw_bm_16a      kw_bm_16b      cp_bhc_16a     m3h_bhc_16b
 Min.   :   0   Min.   :   0   Min.   : 0.00   Min.   :   0
 1st Qu.:3011   1st Qu.:2992   1st Qu.:68.76   1st Qu.:7541
 Median :3129   Median :3110   Median :71.09   Median :7567
 Mean   :3097   Mean   :3095   Mean   :71.60   Mean   :7570
 3rd Qu.:3235   3rd Qu.:3228   3rd Qu.:75.00   3rd Qu.:7598
 Max.   :3991   Max.   :4311   Max.   :99.99   Max.   :8489
  psi_bhc_16a     psi_bhc_16b
 Min.   : 0.00   Min.   : 0.00
 1st Qu.:10.00   1st Qu.:10.12
 Median :11.00   Median :11.00
 Mean   :10.75   Mean   :10.96
 3rd Qu.:12.00   3rd Qu.:12.00
 Max.   :21.67   Max.   :21.00
```

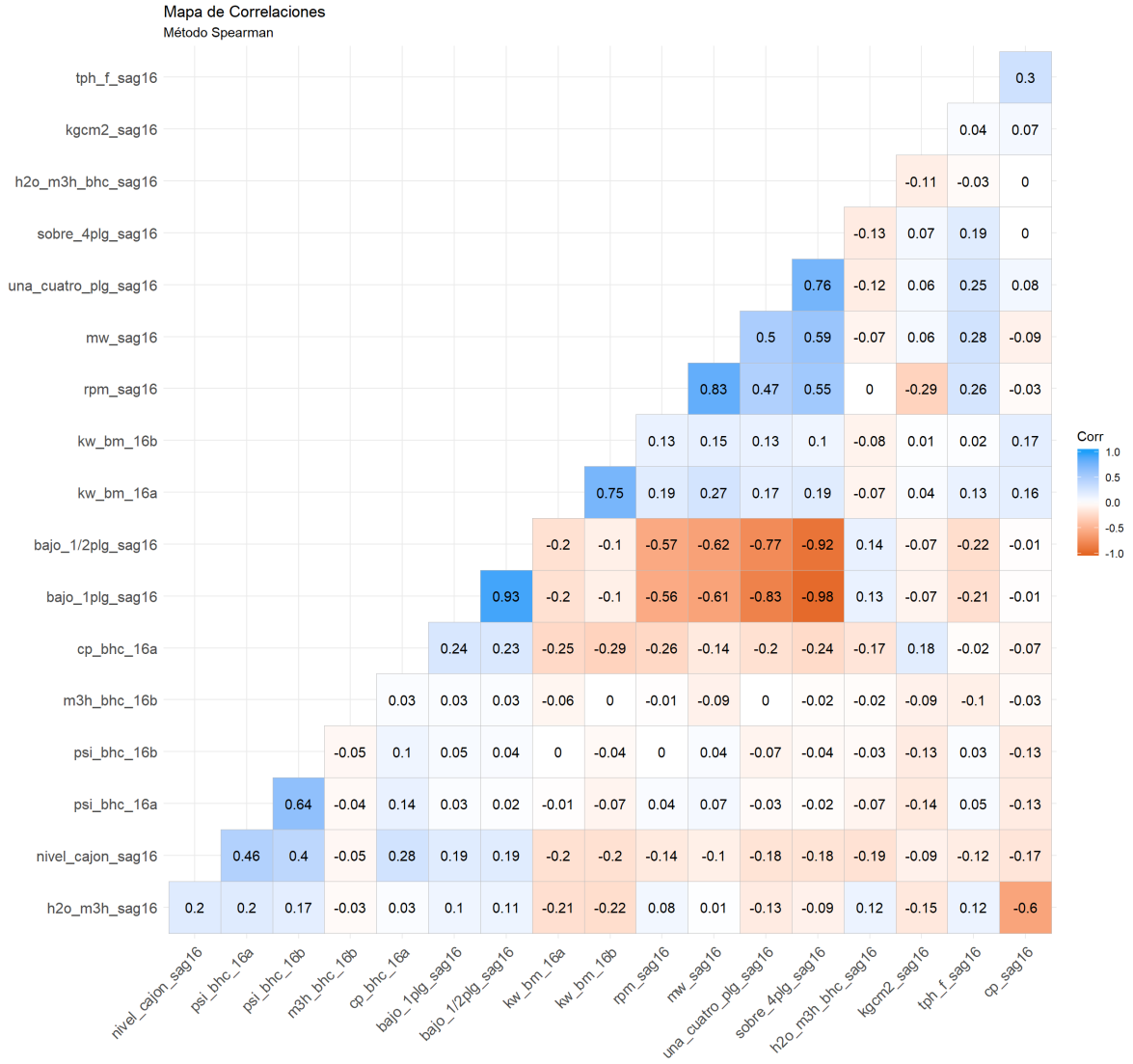
Las gráficas descriptivas del conjunto de datos post-limpieza, se muestran a continuación:

```
# BoxPlot
dataclean %>%
  stack() %>%
  ggplot(aes(x=ind, y=values))+
  geom_boxplot(width=0.5)+
  coord_flip()+
  labs(title = "Grafico de Caja y Bigote de Datos Post-Limpieza",
       subtitle = "Limpieza de var. con alto % ceros + NA´s + Datos Atípicos",
       y = "Valor",
       x = "Variable")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



Finalmente se muestra la gráfica de correlaciones entre las diferentes variables del conjunto de datos post-limpieza, estos es, sin NA's ni datos atípicos:

```
ggcorrplot(cor(dataclean, method = "spearman"),hc.order = TRUE, colors = c("#E46726", "white", "#1D9FF9"), t
ype = "lower", lab = TRUE, lab_size = 4)+
  labs(title = "Mapa de Correlaciones",
       subtitle = "Método Spearman")
```



De la gráfica se pueden ver algunas variables predictoras con un alto nivel de correlación entre ellas (>70%), lo cual podría generar problemas de colinealidad en modelos lineales, esto nos sugiere que algunas de estas variables deberían eliminarse para asi evitar problemas de inestabilidad de los modelos (lineales) y también para disminuir los tiempos de entrenamiento. La variable respuesta **tph_f_sag16** (alimentacion fresca) no muestra correlaciones importantes.

MODELOS DE LÍNEA BASE

Esta sección tiene el objetivo de evaluar el comportamiento base de los datos, bajo ciertos algoritmos, con el fin de visualizar de forma mas clara potenciales estrategias de mejora ya sea en pre-tratamiento de los datos y/o tipos de modelos a evaluar. En esta etapa no se realizará ningún tipo de transformación o pre- procesamiento de los datos.

Primeramente los datos originales se dividen en un grupo para construir/entrenar los algoritmos, conocido como conjunto de **entrenamiento** (80% de los datos), y otro grupo de datos, conocido como conjunto de**prueba** (20% de los datos) cuyos datos no se utilizan en la construccion de los modelos y cuyo propósito radica en validar los resultados de los modelos construidos con el conjunto de entrenamiento.

```
# Grupos de entrenamiento y prueba
set.seed(107)
inTrain <- createDataPartition(dataclean$tph_f_sag16, p = 0.8, list = FALSE)

#Conjuntos de Datos
entrenamiento <- dataclean[inTrain,]
prueba <- dataclean[-inTrain,]
```

También se define un ezquema de entrenamiento para los diferentes algoritmos mediante la técnica de validación cruzada (CV) con K igual a 10 grupos. Esta técnica permite dividir el conjunto de entrenamiento en 10 sub grupos (K=10) cada uno con un sub conjunto de prueba y entrenamiento, los modelos se construyen en el sub conjunto de entrenamiento y se evalúan en el sub conjunto de prueba, de esta forma se evita sobre ajustar los parámetros de los algoritmos y se obtienen métricas que son mas certeras.

Finalmente se define como métrica de evaluación de los modelos el **RMSE**, esto es, la raíz del promedio cuadratico del error.

```
# Se Define Esquema de Entrenamiento
trainControl <- trainControl(method="cv", number=10, savePredictions=TRUE)

# Se define Metrica de Evaluacion
metric <- "RMSE"
```

Los modelos evaluados fueron los siguientes:

1. Regresión por mínimos cuadrados (OLS)
2. Regresión lineal generalizada (GLM)
3. Regresión paso a paso (steplm)
4. Regresión Elastic Net (GlmNet)
5. Regresión Regularizada Ridge (lmrid)
6. Regresión Regularizada Lasso (lmlas)

```
# Modelos de Linea Base
# LM
set.seed(7)
fitBase.lm <- train(tph_f_sag16~., data=entrenamiento, method="lm", metric=metric, trControl=trainControl)
# GLM
set.seed(7)
fitBase.glm <- train(tph_f_sag16~., data=entrenamiento, method="glm", metric=metric, trControl=trainControl)
# LMSTEP
set.seed(7)
fitBase.steplm <- train(tph_f_sag16~., data=entrenamiento, method="leapSeq", metric=metric, trControl=trainC
ontrol)
# GLMNET
set.seed(7)
fitBase.glmnet <- train(tph_f_sag16~., data=entrenamiento, method="glmnet", metric=metric, trControl=trainCo
ntrol)
# RIDGE LM
set.seed(7)
fitBase.lmrid <- train(tph_f_sag16~., data=entrenamiento, method="ridge", metric=metric, trControl=trainCont
rol)
# LASSO LM
set.seed(7)
fitBase.lmlas <- train(tph_f_sag16~., data=entrenamiento, method="lasso", metric=metric, trControl=trainCont
rol)
```

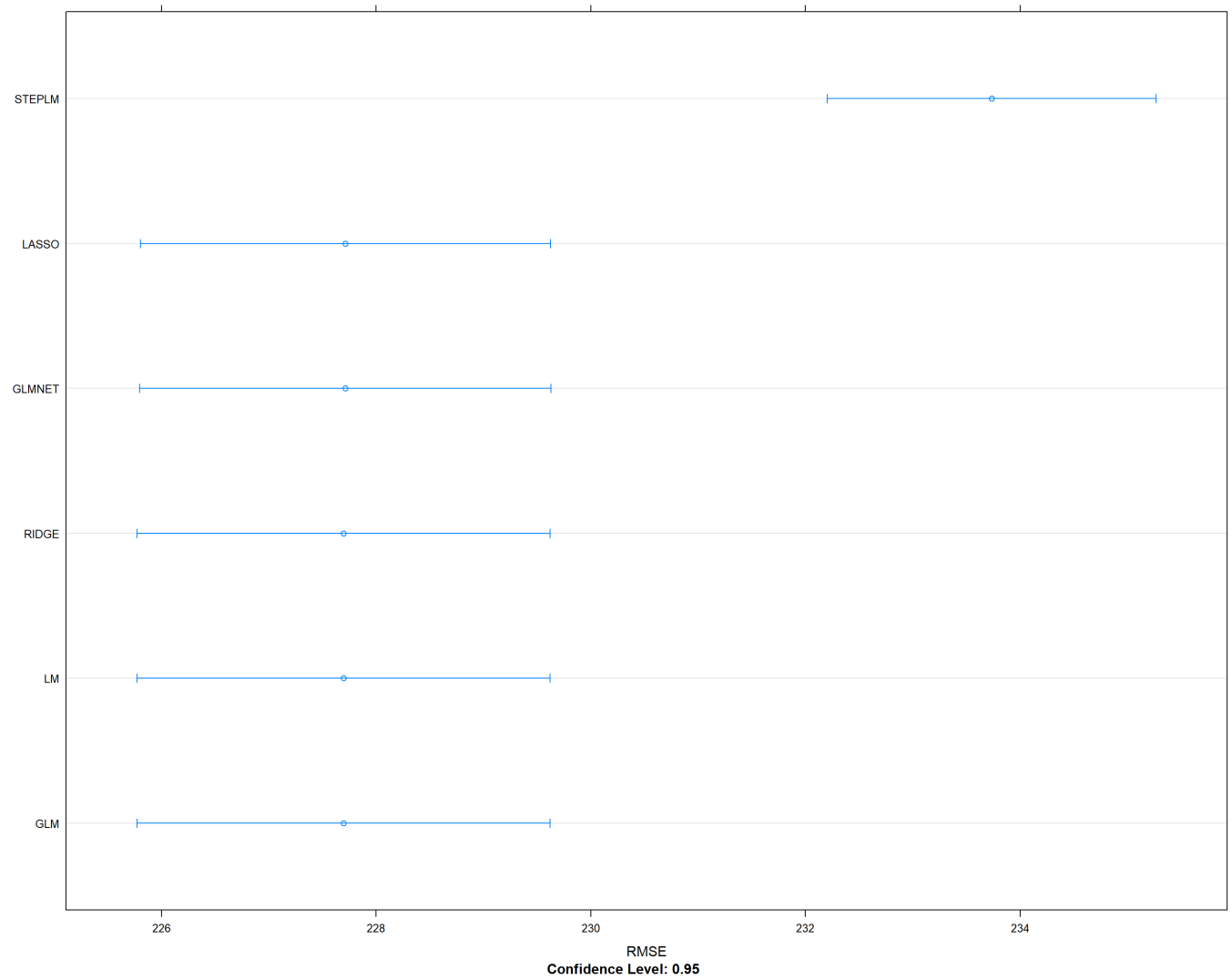
Los resultados de los diferentes modelos de línea base se muestran a continuación:

```
# RESULTADOS
resultsBase <- resamples(list(LM=fitBase.lm, GLM=fitBase.glm, STEPLM=fitBase.steplm, GLMNET=fitBase.glmnet,
RIDGE=fitBase.lmrid, LASSO=fitBase.lmlas))
summary(resultsBase, metric=c("RMSE", "Rsquared"))
```

```

Call:
summary.resamples(object = resultsBase, metric = c("RMSE", "Rsquared"))

Models: LM, GLM, STEPLM, GLMNET, RIDGE, LASSO
Number of resamples: 10

RMSE
           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
LM     224.2391 225.7489 226.6021 227.6974 230.4423 231.4104    0
GLM    224.2391 225.7489 226.6021 227.6974 230.4423 231.4104    0
STEPLM 231.3789 231.9129 233.4744 233.7361 235.4498 236.8780    0
GLMNET 224.2466 225.7639 226.6383 227.7120 230.4517 231.4129    0
RIDGE  224.2391 225.7489 226.6021 227.6974 230.4423 231.4104    0
LASSO  224.2598 225.7668 226.6462 227.7130 230.4538 231.3892    0

Rsquared
            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
LM     0.2332633 0.2410200 0.2552331 0.2546055 0.2688230 0.2740469    0
GLM    0.2332633 0.2410200 0.2552331 0.2546055 0.2688230 0.2740469    0
STEPLM 0.1997398 0.2066828 0.2154449 0.2143035 0.2218154 0.2292467    0
GLMNET 0.2331273 0.2410654 0.2551557 0.2545152 0.2686829 0.2737705    0
RIDGE  0.2332633 0.2410200 0.2552331 0.2546055 0.2688230 0.2740469    0
LASSO  0.2331868 0.2410508 0.2551156 0.2545063 0.2686901 0.2737244    0
```

En los resultados se puede apreciar el resumen de las métricas RMSE y R2, considerando la evaluación realizada en cada grupo (K=10) de los sub conjuntos de prueba, los cuales forman parte del conjunto de entrenamiento. La representación gráfica los resultados obtenidos se muestra a continuación:

```
dotplot(resultsBase, metric=c("RMSE"))
```



Inicialmente se probaron tanto modelos lineales como no lineales, pero el tiempo de cómputo de los modelos no lineales, específicamente Máquina Soporte Vector (SVM), Gradiente Estocástico Potenciado (GBM), Gradiente Extremo Potenciado (XGBoost) y Bosque Aleatorio (RF) fue exesivamente alto lo cual hizo imposible ejecutarlos de forma eficiente. Por su parte el modelo K Vecinos Cercanos (KNN) no fue posible ejecutarlo debido a un error probablemente debido a la alta dimensionalidad de los datos. Otros modelos como Arboles de regresion (CART) y Modelos Aditivos Generalizados (GAM) presentaron errores que no fue posible depurar.

Los modelos lineales de linea base muestran resultados deficientes en todos los casos, con ajustes pobres, que no sueran un R2 del 30%.

Finalmente se computo el RMSE y R2 en el conjunto de prueba original, con el propósito de validar los resultados obtenidos en los sub conjuntos de prueba del conjunto de entrenamiento, considerando el modelo base de Regresión Ridge el cual fue uno de los que reporto mejor RMSE. Como se mencionó antes, esta acción tiene como propósito evaluar el poder predictivo del modelo en un conjunto de datos que el modelo no a procesado.

A continuación se computan tanto el RMSE como el R2 con el modelo Ridge considerando el conjunto de prueba:

```
#Prediccion en el conjunto de prueba
set.seed(7)
prediccionesBase<-predict(fitBase.lmrid, prueba[,2:18])
(rmse_base <- RMSE(prediccionesBase, prueba[["tph_f_sag16"]]))
```

```
[1] 227.0151
```

```
(r2_base <- R2(prediccionesBase, prueba[["tph_f_sag16"]]))
```

```
[1] 0.255386
```

Se puede observar que el modelo de regresión ridge muestra un comportamiento similar, en terminos de RMSE y R2, tanto en el conjunto de entrenamiento (sub conjuntos de prueba) como en el de prueba original, lo anterior nos demuestra que el modelo no sobre-ajusta los datos y que por ende no se aprecian signos de una alta varianza y por lo tanto generaliza bien.

Sin embargo uno de los objetivos de las siguientes secciones será poder realizar los ajustes necesarios para que el modelo posea un menor sesgo o Bías, es decir, que tenga un mayor poder predictivo traducido en un menor RMSE (o mayor R2).

PRE PROCESAMIENTO

Sobre la base de lo anterior se decidió realizar una etapa de pre-procesamiento de los datos en donde principalmente se evaluaron las siguentes estrategias:

- Estandarizar los datos para eliminar el efecto de diferentes distribuciones.
- Eliminar Variables con alto nivel de correlación (>70%), lo cual puede generar problemas de multicolinealidad.
- Buscar y eliminar variables que aportan nula o poca variabilidad.
- Selección de variables predictoras.

Todas estas acciones, en mayor o menor medidda, tienen el propósito de:

- Disminuir el tiempo de procesamiento de los modelos (al reducir la dimensionalidad del conjunto de datos).
- Reducir la complejidad del modelo (mayor parsimonia) eliminando variables que aportan poca o la misma información que otras variables.
- Reducir el sobreajuste del modelo.

Centrado y Escalado de Variables Predictoras

También conocido como estandarizado, las variables numéricas tendrán un promedio de 0 y desviación estándar igual a 1. Esta transformación principalmente es útil para cuando las variables predictoras tienen magnitudes numéricas diferentes unas de otras, la cual genera que aquellas con una mayor magnitud tengan un efecto dominante en los modelos.

```
# Centrado y Escalado (Entrenamiento)
summary(entrenamiento)
```

```
  tph_f_sag16   bajo_1plg_sag16  una_cuatro_plg_sag16 sobre_4plg_sag16
 Min.   : 712   Min.   : 10.86   Min.   : 0.00        Min.   : 0.000
 1st Qu.:1849   1st Qu.: 33.84   1st Qu.:26.00        1st Qu.: 7.984
 Median :2006   Median : 49.35   Median :34.48        Median :15.913
 Mean   :1993   Mean   : 50.85   Mean   :31.13        Mean   :17.816
 3rd Qu.:2182   3rd Qu.: 65.68   3rd Qu.:38.07        3rd Qu.:26.481
 Max.   :3150   Max.   :100.00   Max.   :56.46        Max.   :69.118
 bajo_1/2plg_sag16 h2o_m3h_sag16       cp_sag16       rpm_sag16
 Min.   : 4.322    Min.   :   0.0   Min.   : 0.00   Min.   : 0.000
 1st Qu.:14.569    1st Qu.: 481.0   1st Qu.:75.90   1st Qu.: 8.617
 Median :21.504    Median : 562.0   Median :77.40   Median : 9.325
 Mean   :24.598    Mean   : 543.0   Mean   :77.17   Mean   : 9.182
 3rd Qu.:30.758    3rd Qu.: 623.4   3rd Qu.:79.00   3rd Qu.: 9.914
 Max.   :84.222    Max.   :1193.2   Max.   :98.00   Max.   :14.414
    mw_sag16      kgcm2_sag16     h2o_m3h_bhc_sag16 nivel_cajon_sag16
 Min.   :0.000   Min.   :  0.00   Min.   :   0.0    Min.   :  2.301
 1st Qu.:5.845   1st Qu.: 60.03   1st Qu.: 573.0    1st Qu.: 60.601
 Median :6.571   Median : 61.84   Median : 743.9    Median : 65.999
 Mean   :6.472   Mean   : 61.86   Mean   : 713.1    Mean   : 65.729
 3rd Qu.:7.229   3rd Qu.: 63.82   3rd Qu.: 846.0    3rd Qu.: 71.001
 Max.   :8.754   Max.   :134.21   Max.   :1350.0    Max.   :100.000
   kw_bm_16a      kw_bm_16b      cp_bhc_16a     m3h_bhc_16b
 Min.   :   0   Min.   :   0   Min.   : 0.00   Min.   :   0
 1st Qu.:3011   1st Qu.:2992   1st Qu.:68.77   1st Qu.:7541
 Median :3129   Median :3110   Median :71.11   Median :7567
 Mean   :3097   Mean   :3096   Mean   :71.60   Mean   :7570
 3rd Qu.:3235   3rd Qu.:3228   3rd Qu.:75.00   3rd Qu.:7598
 Max.   :3991   Max.   :4311   Max.   :99.99   Max.   :8489
  psi_bhc_16a     psi_bhc_16b
 Min.   : 0.00   Min.   : 0.00
 1st Qu.:10.00   1st Qu.:10.12
 Median :11.00   Median :11.00
 Mean   :10.75   Mean   :10.97
 3rd Qu.:12.00   3rd Qu.:12.00
 Max.   :21.67   Max.   :21.00
```

```r
preproModel <- preProcess(entrenamiento[,2:18], method=c("center", "scale")) # Se Entrena Modelo con data de entrenamiento
print(preproModel)
```

```
Created from 136641 samples and 17 variables

Pre-processing:
  - centered (17)
  - ignored (0)
  - scaled (17)
```

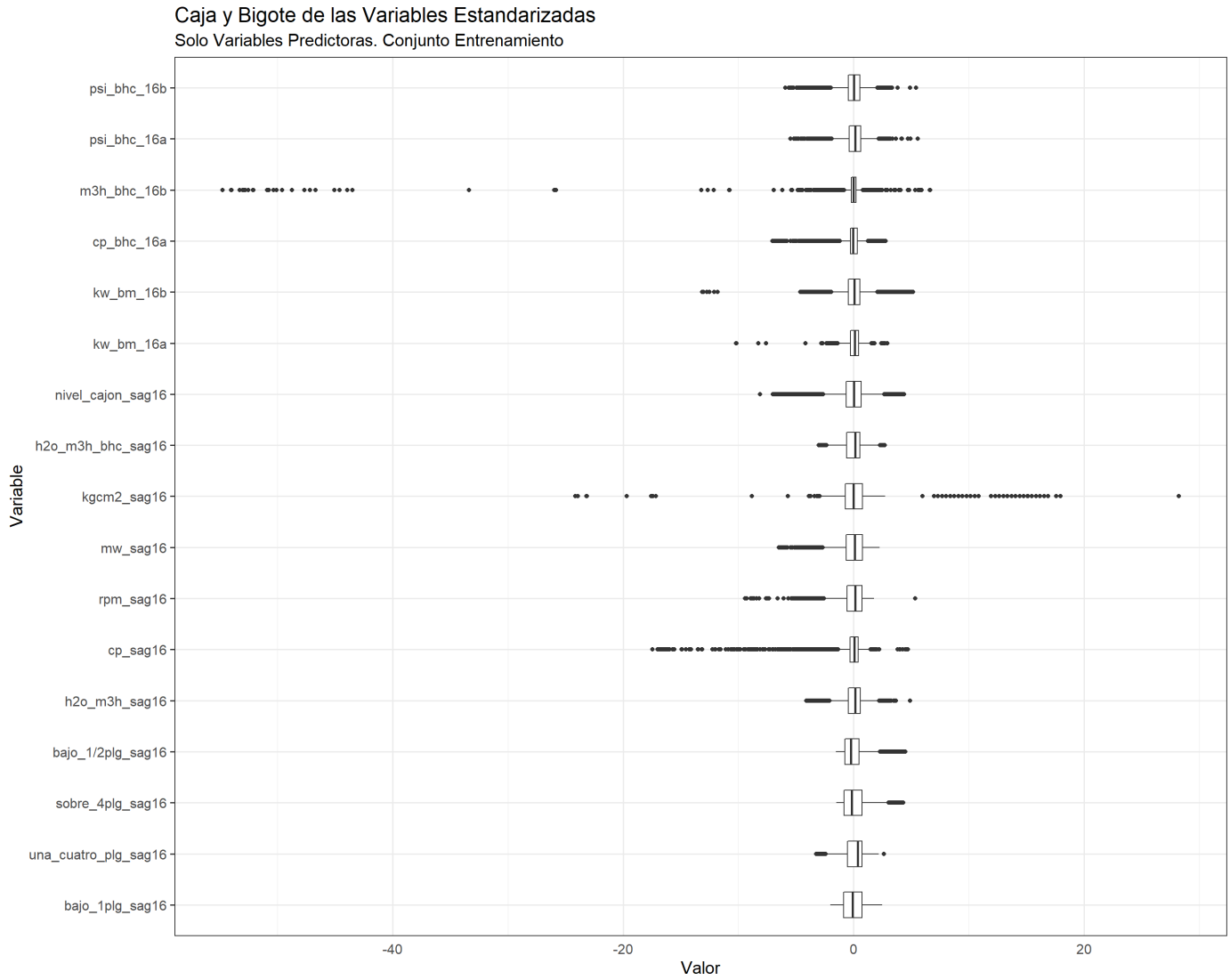
```r
preproEntrenamiento <- predict(preproModel, entrenamiento[,2:18]) # Se aplica modelo a data entrenamiento (escepto var. rpta)
preproEntrenamiento$tph_f_sag16 <- entrenamiento$tph_f_sag16      # Se reingresa var. rpta original en conj. entrenamiento con predictores centrados y escalados
summary(preproEntrenamiento)
```

```
 bajo_1plg_sag16    una_cuatro_plg_sag16 sobre_4plg_sag16
 Min.   :-2.01250   Min.   :-3.2497      Min.   :-1.4968
 1st Qu.:-0.85615   1st Qu.:-0.5353      1st Qu.:-0.8260
 Median :-0.07544   Median : 0.3495      Median :-0.1599
 Mean   : 0.00000   Mean   : 0.0000      Mean   : 0.0000
 3rd Qu.: 0.74642   3rd Qu.: 0.7251      3rd Qu.: 0.7279
 Max.   : 2.47325   Max.   : 2.6450      Max.   : 4.3101
 bajo_1/2plg_sag16 h2o_m3h_sag16        cp_sag16           rpm_sag16
 Min.   :-1.5292   Min.   :-4.0835   Min.   :-17.46432   Min.   :-9.4021
 1st Qu.:-0.7563   1st Qu.:-0.4666   1st Qu.: -0.28797   1st Qu.:-0.5783
 Median :-0.2333   Median : 0.1426   Median :  0.05182   Median : 0.1468
 Mean   : 0.0000   Mean   : 0.0000   Mean   :  0.00000   Mean   : 0.0000
 3rd Qu.: 0.4646   3rd Qu.: 0.6041   3rd Qu.:  0.41372   3rd Qu.: 0.7499
 Max.   : 4.4968   Max.   : 4.8888   Max.   :  4.71368   Max.   : 5.3574
    mw_sag16          kgcm2_sag16         h2o_m3h_bhc_sag16
 Min.   :-6.46833   Min.   :-24.136620   Min.   :-3.0331
 1st Qu.:-0.62695   1st Qu.: -0.711841   1st Qu.:-0.5959
 Median : 0.09928   Median : -0.005842   Median : 0.1310
 Mean   : 0.00000   Mean   :  0.000000   Mean   : 0.0000
 3rd Qu.: 0.75669   3rd Qu.:  0.764498   3rd Qu.: 0.5653
 Max.   : 2.28107   Max.   : 28.230718   Max.   : 2.7092
 nivel_cajon_sag16    kw_bm_16a           kw_bm_16b
 Min.   :-8.11078   Min.   :-10.1955   Min.   :-13.14890
 1st Qu.:-0.65582   1st Qu.: -0.2825   1st Qu.: -0.44145
 Median : 0.03453   Median :  0.1068   Median :  0.06086
 Mean   : 0.00000   Mean   :  0.0000   Mean   :  0.00000
 3rd Qu.: 0.67415   3rd Qu.:  0.4545   3rd Qu.:  0.56317
 Max.   : 4.38231   Max.   :  2.9427   Max.   :  5.16173
   cp_bhc_16a         m3h_bhc_16b          psi_bhc_16a
 Min.   :-7.02121   Min.   :-54.77389   Min.   :-5.4791
 1st Qu.:-0.27768   1st Qu.: -0.20906   1st Qu.:-0.3836
 Median :-0.04814   Median : -0.02093   Median : 0.1263
 Mean   : 0.00000   Mean   :  0.00000   Mean   : 0.0000
 3rd Qu.: 0.33311   3rd Qu.:  0.20338   3rd Qu.: 0.6358
 Max.   : 2.78376   Max.   :  6.65044   Max.   : 5.5643
  psi_bhc_16b         tph_f_sag16
 Min.   :-5.93527   Min.   : 712
 1st Qu.:-0.45710   1st Qu.:1849
 Median : 0.01915   Median :2006
 Mean   : 0.00000   Mean   :1993
 3rd Qu.: 0.56033   3rd Qu.:2182
 Max.   : 5.43191   Max.   :3150
```

```
# Centrado y Escalado (Prueba)
preproPrueba <- predict(preproModel, prueba[,2:18])
preproPrueba$tph_f_sag16 <- prueba$tph_f_sag16
```

El efecto del proceso de centrado y escalado se visualiza a continuación:

```
# Grafica de Variables Estandarizadas
preproEntrenamiento %>%
  dplyr::select(-tph_f_sag16) %>%
  stack() %>%
  ggplot(aes(x=ind, y=values))+
  geom_boxplot(width=0.5)+
  coord_flip()+
  labs(title = "Caja y Bigote de las Variables Estandarizadas",
       subtitle = "Solo Variables Predictoras. Conjunto Entrenamiento",
       y = "Valor",
       x = "Variable")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



Eliminación de Variables Predictoras con Alta Correlación.

Se define una alta correlación toda vez que esta es mayor o igual a 70%. Se computa la correlación vía método de spearman.

```
# Predictores correlacionados >|70%| (se excluye variable respuesta tph_f_sag16)
set.seed(7)
colinealidad <- findCorrelation(cor(preproEntrenamiento[,1:17], method = "spearman"), cutoff=0.7, names=TRUE
, verbose = TRUE)
```

```
Compare row 1  and column  4 with corr  0.933
  Means:  0.315 vs 0.18 so flagging column 1
Compare row 4  and column  3 with corr  0.922
  Means:  0.267 vs 0.165 so flagging column 4
Compare row 3  and column  2 with corr  0.755
  Means:  0.211 vs 0.154 so flagging column 3
Compare row 8  and column  7 with corr  0.833
  Means:  0.186 vs 0.146 so flagging column 8
Compare row 12  and column  13 with corr  0.746
  Means:  0.175 vs 0.14 so flagging column 12
All correlations <= 0.7
```

```
colinealidad
```

```
[1] "bajo_1plg_sag16"   "bajo_1/2plg_sag16" "sobre_4plg_sag16"
[4] "mw_sag16"          "kw_bm_16a"
```

Las primeras 3 comparativas involucran las 4 variables relacionadas con el tamaño de partícula que alimenta al molino SAG. Finalmente se eliminan 3 de las 4, dado que entregan la misma información.

La comparativa entre las variables 7 y 8 da cuenta de las rpm del molino versus la potencia (mw) respectivamente, siendo

operacionalmente una dependiente de la otra (a mayor rpm del molino mayor potencia consumida) era esperable esta relación, sin embargo, dado que la variable potencia se enmarca dentro de los KPI del proceso, esta última se conservará en desmedro de la variable rpm.

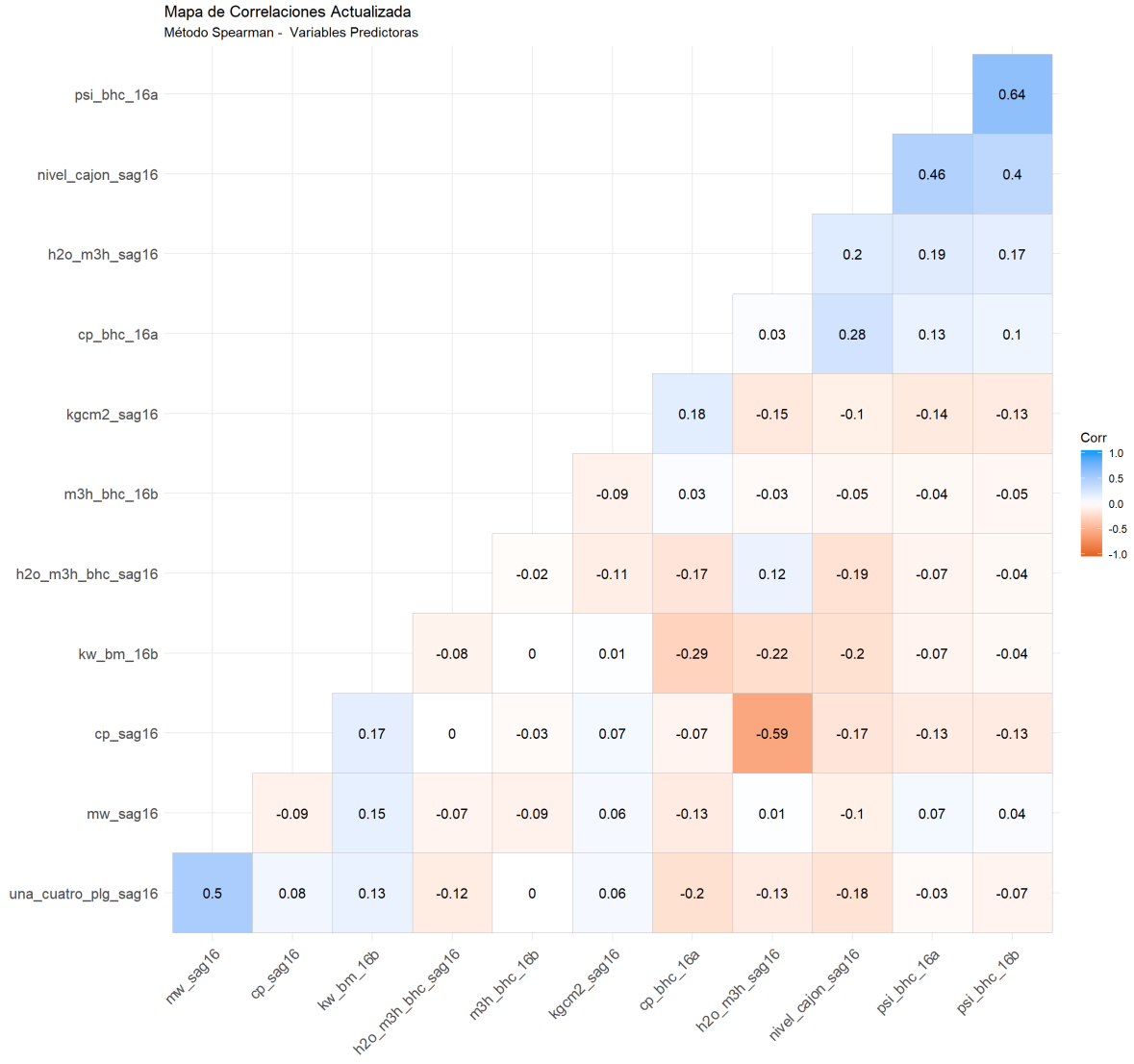
Finalmente la última comparativa da cuenta de las potencias de los 2 molinos de bolas que operan en paralelo después del molino SAG, por lo cual era esperable esta relación y se elimina la variable que nos sugiere el algoritmo.

Hecho el análisis, se procede a retirar las variables previamente definidas desde los conjunto de entrenamiento y prueba.

```
# Eliminacion de Predictores Correlacionados (Entrenamiento)
preproCorEntrenamiento<-preproEntrenamiento %>%
  dplyr::select(-bajo_1plg_sag16, -'bajo_1/2plg_sag16', -sobre_4plg_sag16, -rpm_sag16, -kw_bm_16a)

# Eliminacion de Predictores Correlacionados (Prueba)
preproCorPrueba<-preproPrueba%>%
 dplyr::select(-bajo_1plg_sag16, -'bajo_1/2plg_sag16', -sobre_4plg_sag16, -rpm_sag16, -kw_bm_16a)
```

La gráfica de correlaciones actualizadas se aprecia a continuación:

```
preproCorEntrenamiento %>%
  dplyr::select(-tph_f_sag16) %>%
  cor(method = "spearman") %>%
  ggcorrplot(hc.order = TRUE, colors = c("#E46726", "white", "#1D9FF9"), type = "lower", lab = TRUE, lab_siz
e = 4)+
  labs(title = "Mapa de Correlaciones Actualizada",
       subtitle = "Método Spearman -  Variables Predictoras")
```



Selección de Variables que Aportan Poca o Nula Variabilidad

```
# Verificacion de variables con variaza cercana a cero
preproCorEntrenamiento %>%
  dplyr::select(-tph_f_sag16) %>%
  nearZeroVar()
```

```
integer(0)
```

Se puede apreciar que no hay variables predictoras que tengan valores constantes o que aporten poca variabilidad.

Selección de Variables Predictoras

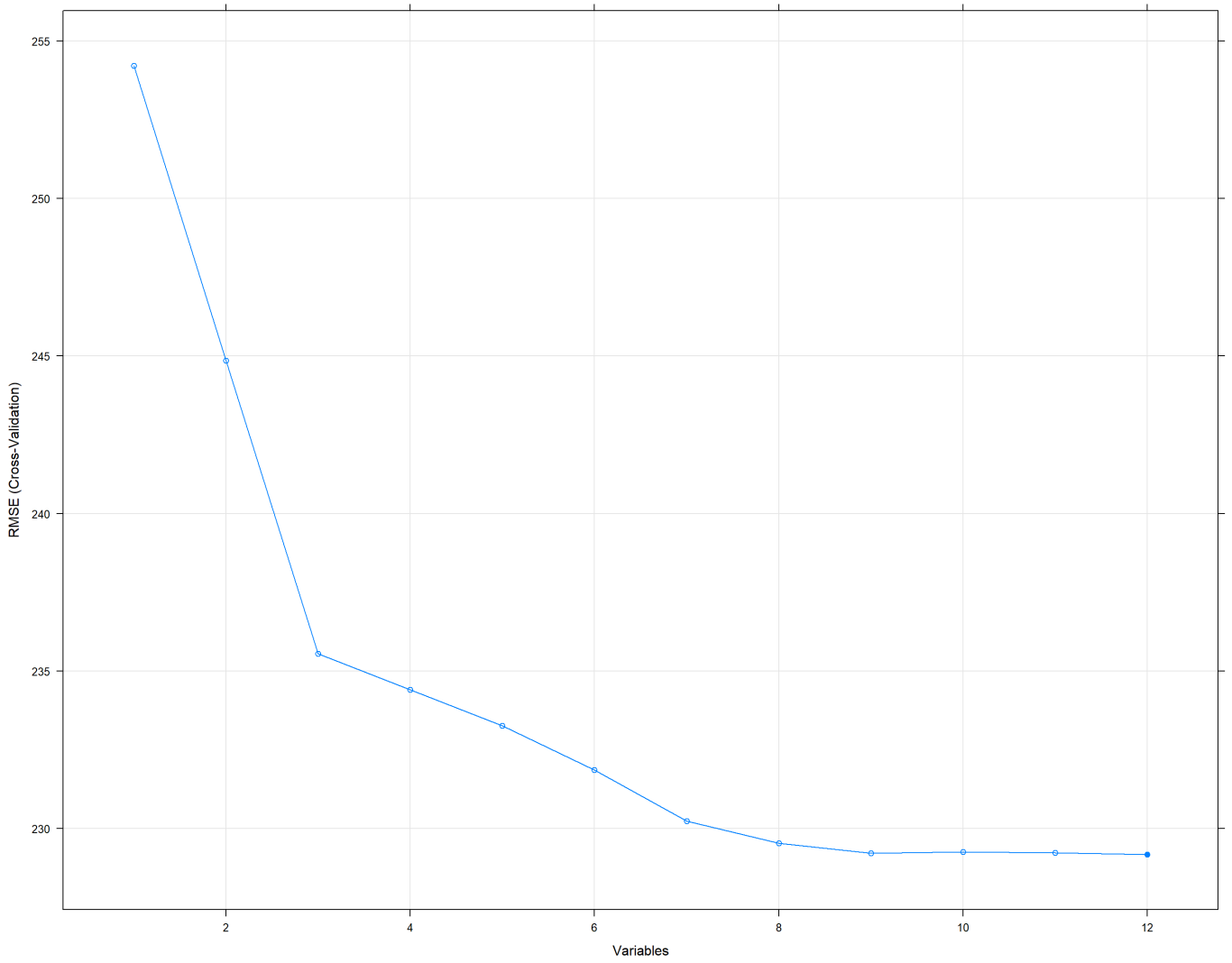
Esta etapa da cuenta del proceso de selección de las variables que mas importancia tienen en la predicción de la respuesta**tph_f_sag_16**. En este caso se utilizará la técnica de la Eliminación Recursiva de Variables o RFE.

En esta técnica es de vital importancia elegir el algoritmo con el cual se realizará la selección de variables, ya que el procedimiento entrena un modelo base que ocupa para medir la importancia de estas.

Es siempre una buena practica ocupar varios tipos de modelos en la selección de variables, ya que cada modelo entrega información diferente, lo cual nos permite elegir las variables sin un sesgo debido al tipo de algoritmo.

A continuación se presentan los resultados al entrenar un modelo lineal por mínimos cuadrados:

```
#Seleccion de Variables

## se define rutina de control
set.seed(7)
rfeControl <- rfeControl(functions = lmFuncs, method = "cv", number = 10, verbose = FALSE)

## rfe
set.seed(7)
varSelect <- rfe(preproCorEntrenamiento[,1:12],preproCorEntrenamiento[[13]],metric = "RMSE", sizes = c(1:12)
,rfeControl = rfeControl)
plot(varSelect, type=c("g", "o")) # grafica de resultados
```



En la gráfica se puede apreciar como a medida que se ingresan variables al modelo el RMSE disminuye hasta un punto en que se mantiene casi constante, en este caso el algoritmo nos sugiere que consideremos todas las variables en el modelo (designado con un círculo compacto). Ahora bien, dependiendo del criterio experto a utilizar, también se podrían incluir menos variables, como por ejemplo las 7 primeras o donde se estime que la mejoría en términos de disminución de RMSE deja de ser relevante. Finalmente, el modelo lineal nos sugiere que se deben considerar la totalidad (12) variables, estas serían:

```
# Listado de Predictores seleccionados
predictors(varSelect)
```

```
 [1] "cp_sag16"             "h2o_m3h_sag16"        "mw_sag16"
 [4] "una_cuatro_plg_sag16" "nivel_cajon_sag16"    "psi_bhc_16b"
 [7] "psi_bhc_16a"          "cp_bhc_16a"           "h2o_m3h_bhc_sag16"
[10] "kw_bm_16b"            "kgcm2_sag16"          "m3h_bhc_16b"
```

Sobre la base de lo anterior no se estaria excluyendo ninguna variable:

```
#Variables a excluir segun Seleccion de Variables

preproCorEntrenamiento[,1:12] %>%
  dplyr::select(-one_of(predictors(varSelect))) %>%
  names()
```

```
character(0)
```

Otra forma de efectuar una pre- selección de variables predictoras es a traves de la regresion Elastic Net, la cual es una mezcla de los algoritmos de regresión regularizada Lasso y Ridge. La “regularización”" contrae los parámetros de los diferentes coeficientes de las variables predictoras hacia cero. Exsten 2 tipos de regularizaciones:

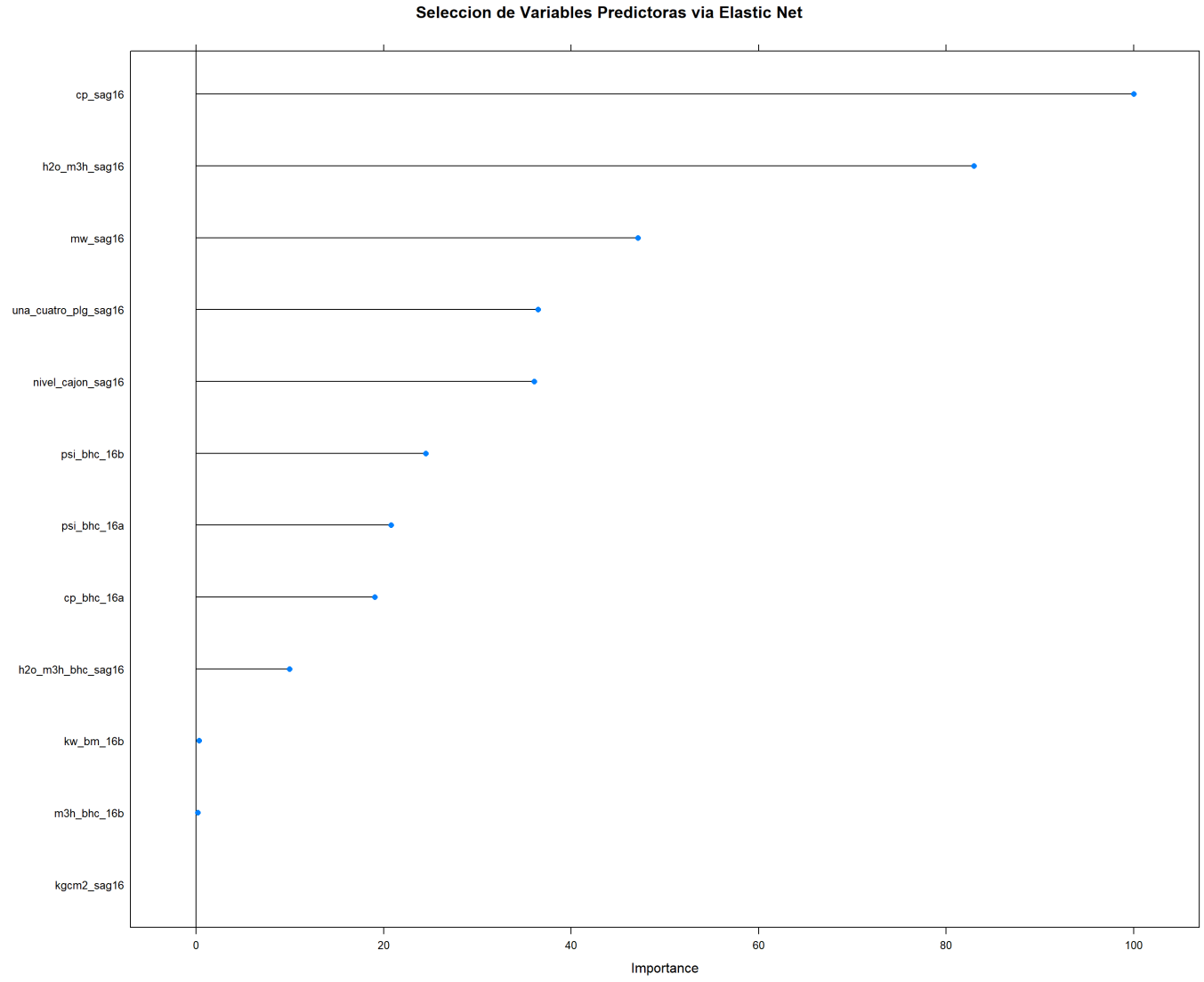
- Regularización tipo L1 o lasso, la cual contrae algunos coeficientes haciéndolos igual a cero.
- Regularización tipo L2 o ridge, la cual contrae algunos coeficientes haciéndolos cercanos a cero (pero no iguales a cero).

En este contexto usualmante se utiliza la regresión lasso como un algoritmo de preselección de variables. Sin embargo la desventaja radica en que, en ciertas ocasiones, elimina variables que pueden ser útiles en la predicción de la variable respuesta en razón de una mayor interpretabilidad del modelo, siendo esta la principal diferencia con la regresión ridge la cual igualmente penaliza los coeficientes pero deja todos los términos (variables) en el modelo con el fin de maximizar el poder predictivo.

En este contexto la regresión elastic net genera un compromiso entre ambas técnicas intentando maximizar la interpretabilidad y el poder predictivo vía un componenete de penalización que se mueve entre ridge y lasso.

El desarrollo de la regresión elastic net se presenta a continuación:

```
# Seleccion de variable via Elastic Net
set.seed(7)
fit.varSelectglmnet <- train(tph_f_sag16~., data=preproCorEntrenamiento, method="glmnet", metric=metric, trC
ontrol=trainControl)

##Coeficientes
#fit.varSelectglmnet$bestTune$alpha # alpha=1 -> 100% lasso.
#fit.varSelectglmnet$bestTune$lambda


plot(varImp(object=fit.varSelectglmnet),main="Seleccion de Variables Predictoras via Elastic Net")
```



La gráfica en parte nos corrobora lo que el método RFE vía regresión lineal simple nos indicó, esto es que todas las variables deben ingresarse al modelo. Aunque es importante señalar que la regresión elastic net contrajo cerca de cero 3 variables con valores absolutos alrededor de 3, lo cual nos indica que dependiendo del criterio se podrian eliminar. El detalle numérico de los coeficientes se muestra a continuación:

```
# Listado de Predictores seleccionados
coef(fit.varSelectglmnet$finalModel, fit.varSelectglmnet$bestTune$lambda)
```

```
13 x 1 sparse Matrix of class "dgCMatrix"
                                1
(Intercept)          1992.725014
una_cuatro_plg_sag16   35.361310
h2o_m3h_sag16          76.131570
cp_sag16               91.077448
mw_sag16               44.687099
kgcm2_sag16             3.385963
h2o_m3h_bhc_sag16     -12.109459
nivel_cajon_sag16     -34.998948
kw_bm_16b              -3.659017
cp_bhc_16a             20.080999
m3h_bhc_16b            -3.554516
psi_bhc_16a            21.643303
psi_bhc_16b            24.874590
```

Finalmente se decide mantener todas las variables, lo cual será examinado en función de los resultados que se obtengan en la etapa de modelamiento.

EVALUACIÓN DE ALGORITMOS

Se evaluarán los siguientes modelos:

1. Regresión Lasso (Lasso)
2. Regresión Ridge (Ridge)
3. Regresión Elastic Net (GlmNet)
4. Modelo Generalizado Aditivo Potenciado (GamBoost)
5. Modelo Cúbico (Cubist)
6. Gradiente Extremo Potenciado (XGboost)
7. Bosque Aleatorio (RF)
8. Máquina Gradiente Potenciado (GBM)

Se ocupa la misma métrica y ezquema de entrenamiento que en los modelos base.

```
#Data
datafit <- preproCorEntrenamiento
```

```
#Evaluacion de Modelos

# GLMNET
set.seed(7)
fit.glmnet <- train(tph_f_sag16~., data=datafit, method="glmnet", metric=metric, trControl=trainControl)
# RIDGE LM
set.seed(7)
fit.ridge <- train(tph_f_sag16~., data=datafit, method="ridge", metric=metric, trControl=trainControl)
# LASSO LM
set.seed(7)
fit.lasso <- train(tph_f_sag16~., data=datafit, method="lasso", metric=metric, trControl=trainControl)
# GAMBOOST
set.seed(7)
fit.gamBoost <- train(tph_f_sag16~.,data=datafit, method="gamboost", metric=metric, trControl=trainControl)
# CUBIST
set.seed(7)
fit.cubist <- train(tph_f_sag16~.,data=datafit, method="cubist", metric=metric,trControl=trainControl)
# XGBOOST
set.seed(7)
fit.xgBoost <- train(tph_f_sag16~., data=datafit, method="xgbLinear", metric=metric,trControl=trainControl)
# RF
set.seed(7)
fit.rf <- train(tph_f_sag16~.,data=datafit, method="ranger", metric=metric, trControl=trainControl)
# GBM
set.seed(7)
fit.gbm <- train(tph_f_sag16~., data=datafit, method="gbm", metric=metric, trControl=trainControl)
```

Los resultados, RMSE y R2, de los diferentes modelos se muestran a continuación:

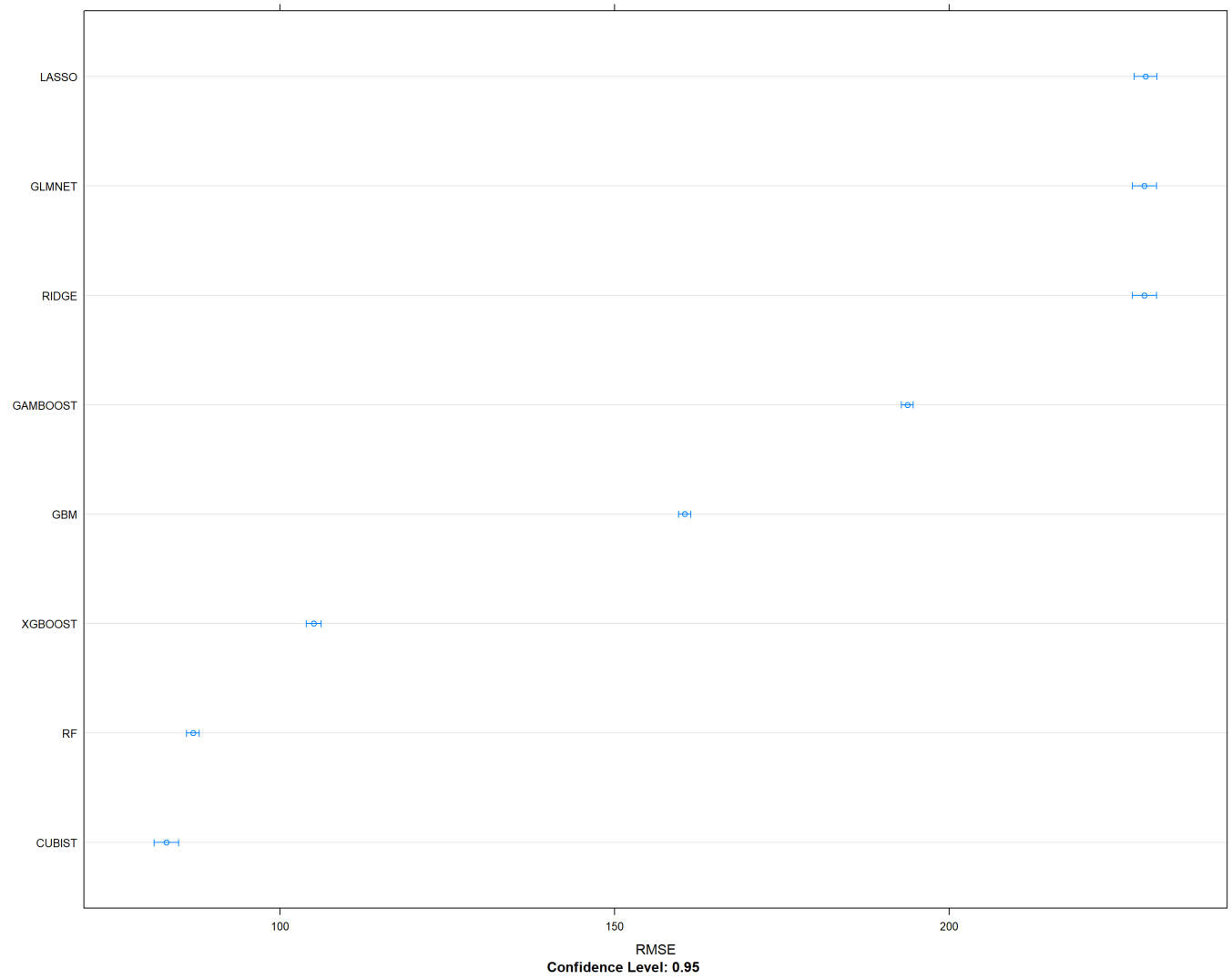
```
# RESULTADOS
results <- resamples(list(GLMNET=fit.glmnet, RIDGE=fit.ridge, LASSO=fit.lasso, GAMBOOST=fit.gamBoost, CUBIST
=fit.cubist, XGBOOST=fit.xgBoost, RF=fit.rf, GBM=fit.gbm))
summary(results, metric=c("RMSE", "Rsquared"))
```

```
Call:
summary.resamples(object = results, metric = c("RMSE", "Rsquared"))

Models: GLMNET, RIDGE, LASSO, GAMBOOST, CUBIST, XGBOOST, RF, GBM
Number of resamples: 10

RMSE
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
GLMNET   226.13308 227.11727 228.49981 229.16193 231.77132 232.39784    0
RIDGE    226.09821 227.12030 228.47201 229.16138 231.81853 232.40264    0
LASSO    226.56868 227.28599 228.91151 229.35310 231.57849 232.56569    0
GAMBOOST 192.65614 192.83764 193.06996 193.72556 194.08514 196.03900    0
CUBIST    78.71450  81.73351  83.57314  82.98431  84.29139  86.34315    0
XGBOOST  102.68999 103.66031 105.33594 105.02555 106.28225 107.06632    0
RF        85.08938  85.80473  87.06355  86.96410  87.83089  89.10210    0
GBM      158.48359 159.78546 160.20472 160.44053 161.01423 163.10903    0

Rsquared
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
GLMNET   0.2269694 0.2311726 0.2466574 0.2448622 0.2567884 0.2621741    0
RIDGE    0.2269582 0.2311799 0.2466196 0.2448636 0.2567919 0.2622011    0
LASSO    0.2263685 0.2308868 0.2464796 0.2442798 0.2563202 0.2614213    0
GAMBOOST 0.4682748 0.4711978 0.4725185 0.4780152 0.4829600 0.5007447    0
CUBIST   0.8938247 0.8978190 0.8997635 0.9011315 0.9032285 0.9119576    0
XGBOOST  0.8360443 0.8383715 0.8406795 0.8412993 0.8442732 0.8492658    0
RF       0.8890500 0.8917404 0.8932963 0.8936988 0.8947655 0.9006596    0
GBM      0.6283796 0.6309674 0.6341742 0.6357744 0.6365334 0.6558453    0
```

Al igual que en los modelos de línea base los resultados muestran las métricas para los sub grupos de prueba presentes en cada K, con K igual a 10, de acuerdo al ezquema de entrenamiento mediante validación cruzada.

La interpretación gráfica de los resultados se puede ver a continuación:

```
# Resumen estadistico de modelos
dotplot(results, metric=c("RMSE"))
```



Se puede apreciar desde la gráfica que los mejores modelos fueron:**Cubico**, **Random Forest (RF)** y **Gradiente Extremo Poenciado (XGBOOST)**, los cuales reportaron RMSE en torno a 100. Por su parte los modelos lineales fueron los que reportaron los resultados mas bajos, sin embargo el Modelo Aditivo Generalizado Potenciado (GAMBOOST) se comporto mejor que los modelos lineales Lasso, Ridge y Elastic Net. Lo anterior nos confirma que los datos no se describen mediante una relación lineal.

OPTIMIZACIÓN DE MODELOS

A fin de maximizar aun más el poder predictivo de los modelos, existen 2 metodos que se pueden ocupar:

Optimización de Hiperparámetros de los Modelos

Cada modelo tiene diferentes hiperparámetros, los cuales deben seleccionarse arbitrariamente (o según conocimiento experto) antes de ejecutar el modelo. En general el valor final de estos se selecciona vía minimización de la métrica elegida dentro de un ezquema de iteración vía validación cruzada.

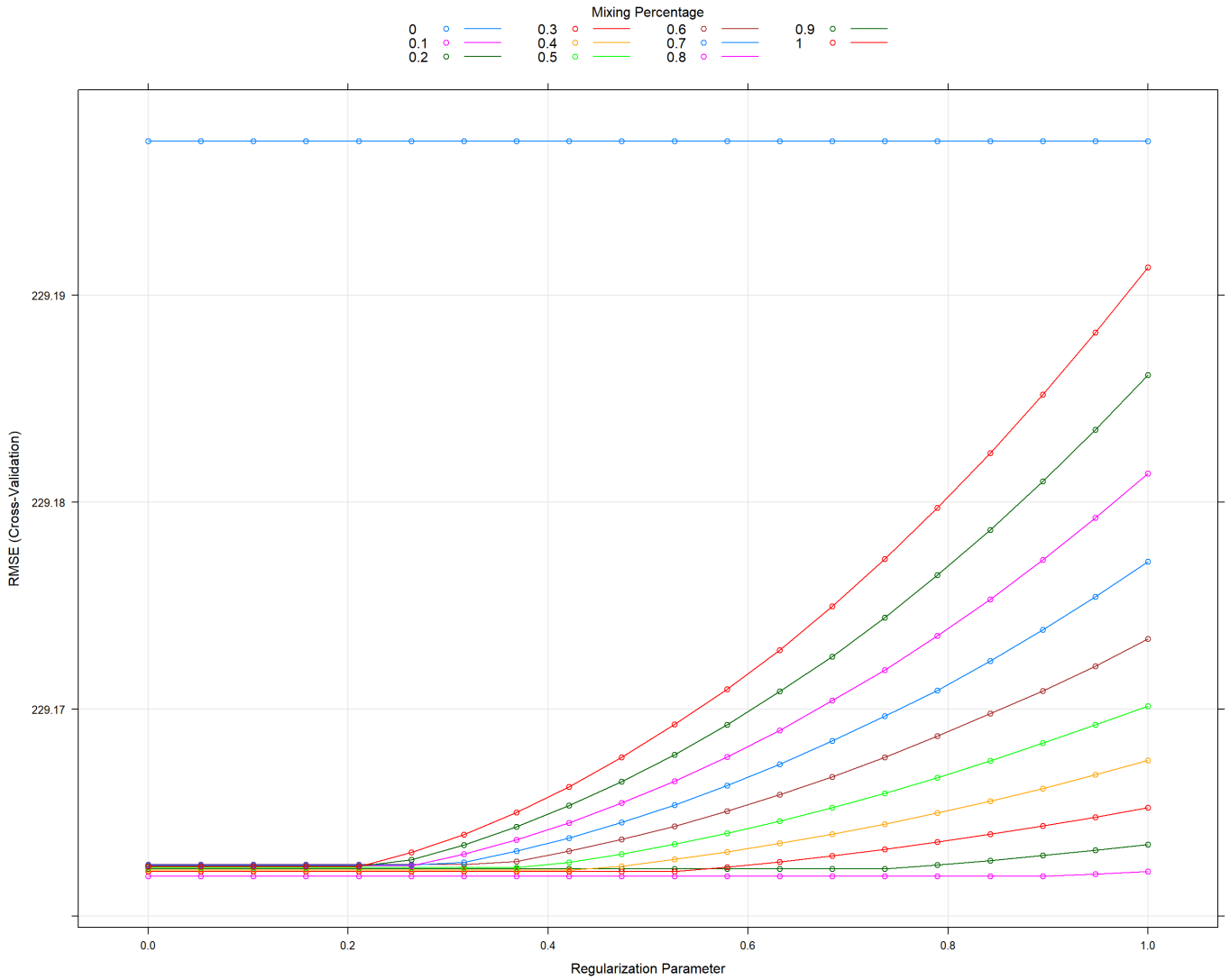
Cabe señalar que el valor final de estos hiperparámetros tiene un gran impacto en el rendimiento de los algoritmos. Por defecto la librería Caret realiza 3 iteraciones para encontrar el mejor valor de cada hiperparámetro, finalmente conserva el valor que minimiza la métrica objetivo (en nuestro caso RMSE).

Con el propósito de optimizar estos valores, y de esta forma el poder predictivo de los modelos, el usuario puede indicarle al algortimo que itere mas de 3 veces (las que se deseen) y/o entregarle un rango de valores en los cuales interpolar. Lo anterior tiene como propósito encontrar el mejor valor de hiperparámetro que maximize el poder predictivo del modelo.

A modo de ejemplo se buscarán los mejores hiperparámetros para la regresión Elastic Net, debido al alto tiempo de entrenamiento que esto significaría lo anterior no se realizará para el resto de los algortimos.

```
# GLMNET
set.seed(7)
fit.glmnetTunning <- train(tph_f_sag16~., data=datafit, tuneGrid=expand.grid(alpha=seq(0,1,0.1), lambda=seq(
0.0001, 1, length=20)), method="glmnet", metric=metric, trControl=trainControl)
```

```
plot(fit.glmnetTunning)
```



En la gráfica el eje X da cuenta de los diferentes valores del hiperparámetro lambda (parámetro de regularización), mientras que las difernetes rectas dan cuenta del hiperparametro alfa (% de mezcla entre ridge y lasso, desde 0 [100% ridge] hasta 1 [100% lasso]), el eje Y muestra el RMSE cuyo valor, en este caso, se busca minimizar. Se puede observar que el menor RMSE se encuentra en la recta de color rosa que da cuenta de una mezcla o alfa igual a 0.1, esto es, 90% ridge y 10% lasso, siendo esta mezcla conocida como elastic net.

El valor de los mejores hiperparámetros son los siguientes:

```
# Tunning Hiperparametros Elastic Net
fit.glmnetTunning$bestTune %>%
  datatable() %>%
   formatRound(columns=c('lambda'), digits=4)
```

Estos valores son los que minimizan el RMSE y son los parámetros finales del modelo Elastic Net.

Ensamble de Modelos

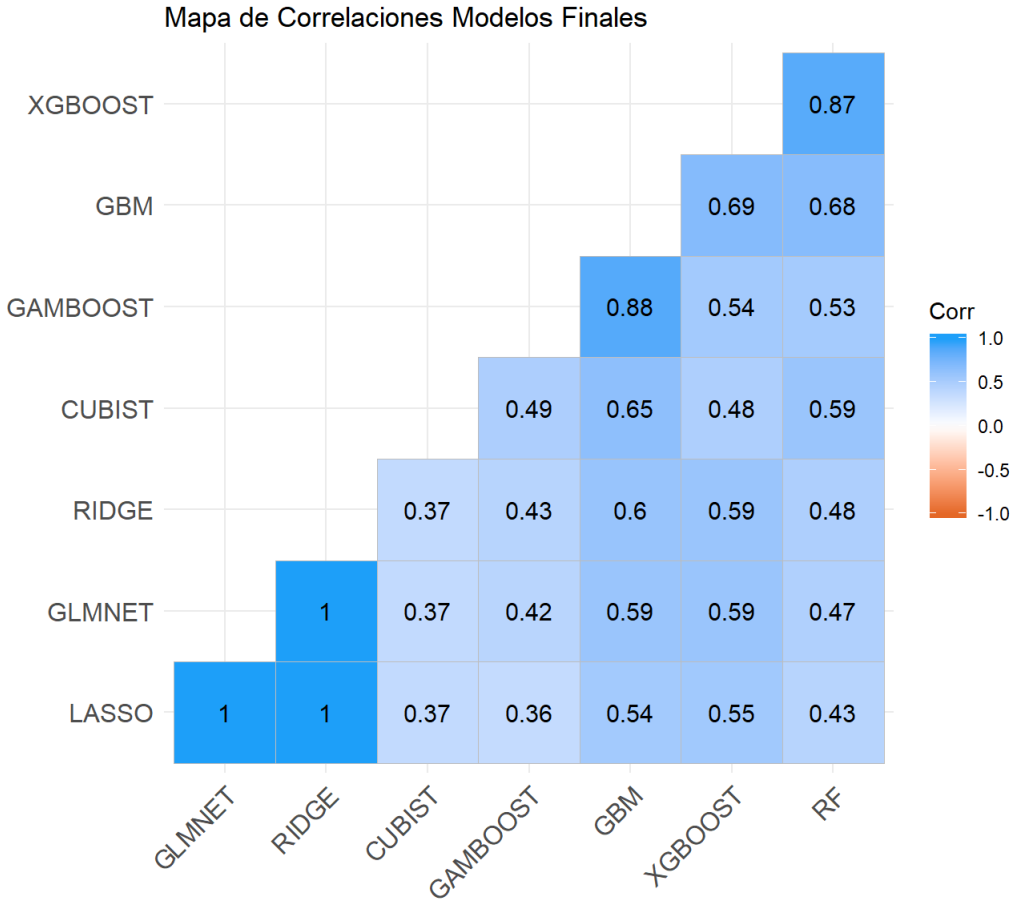
El ensamble da cuenta de la fusión de diferentes modelos, mas específicamente sus predicciones, con el propósito que juntas puedan maximizar el poder predictivo. Sin embargo, a fin de que la fusión de las predicciones sea superior a las originales, es necesario que estas cumplan con las siguientes condiciones:

- Las predicciones de los modelos deben tener un RMSE parecido.
- No deben tener una correlacion >75%: caso contrario las predicciones aportarian información similar y no existiría beneficio al unirlas.

Acorde a los resultados obtenidos podemos indicar que los modelos: glmnet, lasso, ridge, gamboost y gbm, presentan R2 muy por debajo de los valores óptimos esperados (>85%) y del resto de los modelos, por ende no serán considerados como candidatos para ensamble.

Respecto a las correlaciones entre las predicciones de los diferentes modelos, estas se pueden ver a continuación:

```
# Correlacion Mejores Modelos
modelCor(results) %>%
  ggcorrplot(hc.order = TRUE, colors = c("#E46726", "white", "#1D9FF9"), type = "lower", lab = TRUE, lab_siz
e = 4)+
  labs(title = "Mapa de Correlaciones Modelos Finales")
```



Se puede ver claramente que los modelos RF y XGBOOST tienen una correlación que excede lo antes definido, por ende y dado que las métricas reportadas por el modelo RF fueron ligeramente superiores a las del modelo XGBOOST, el ensable se realizará entre los modelos **CUBIC** y **RF**.

A continuación se muestra el ensamblaje de predicciones de los modelos Cubist y RF vía metodo de regresión Elastic Net.

```
#Modelos Ensamble 1
list1 <- c(fit.cubist, fit.rf)

# Ezquema de Entrenamiento
stackControl <- trainControl(method="cv", number=10, savePredictions = TRUE)

# Ensamble 1
set.seed(7)
ensamble.glmnet <- caretStack(list1, method="glmnet", metric="RMSE", trControl=stackControl)
print(ensamble.glmnet)
```

```
A glmnet ensemble of 2 base models: cubist, ranger

Ensemble results:
glmnet

136641 samples
     2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 122976, 122977, 122977, 122976, 122977, 122977, ...
Resampling results across tuning parameters:

  alpha  lambda      RMSE      Rsquared   MAE
  0.10    0.5003908  80.31206  0.9070653  46.65803
  0.10    5.0039080  80.36656  0.9070231  46.66077
  0.10   50.0390805  84.07564  0.9068387  50.98915
  0.55    0.5003908  80.30575  0.9070882  46.65844
  0.55    5.0039080  80.39551  0.9070672  46.69153
  0.55   50.0390805  88.44459  0.9069060  56.11571
  1.00    0.5003908  80.30436  0.9070943  46.67372
  1.00    5.0039080  80.44389  0.9070914  46.78098
  1.00   50.0390805  94.73616  0.9070444  63.06330

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 1 and lambda = 0.5003908.
```

El ensamble via Elastic Net muestra un RMSE de 80.30 (para lambda=0.5 y alfa=1) versus 82.98 reportado por el modelo Cubist el cual fue el mejor de la sección anterior. Por lo tanto se logró aumentar aún más, aunque no de forma relevante, el poder predictivo.

CIERRE DE MODELO

A continuación se presentan los pasos finales a realizar en la ejecución del modelo:

1. Predicción del Modelo en el Conjunto de Prueba

Las predicciones se realizarán en el conjunto de prueba original con el propósito de validar los resultados obtenidos en los sub conjuntos de prueba del conjunto de entrenamiento, además, solo se realizaron para los modelos individuales dado que la funcion “predict” arojo un error constante en el modelo de ensamble **ensamble1.glmnet**, posiblemente aducible a un error en las rutinas internas del paquete caretEnsamble. Lo anterior podría abordarse en otra etapa del presente trabajo.

A continuación se muestran los resultados de las métricas RMSE y R2 calculadas en el Conjunto de Prueba original para los mejores modelos:

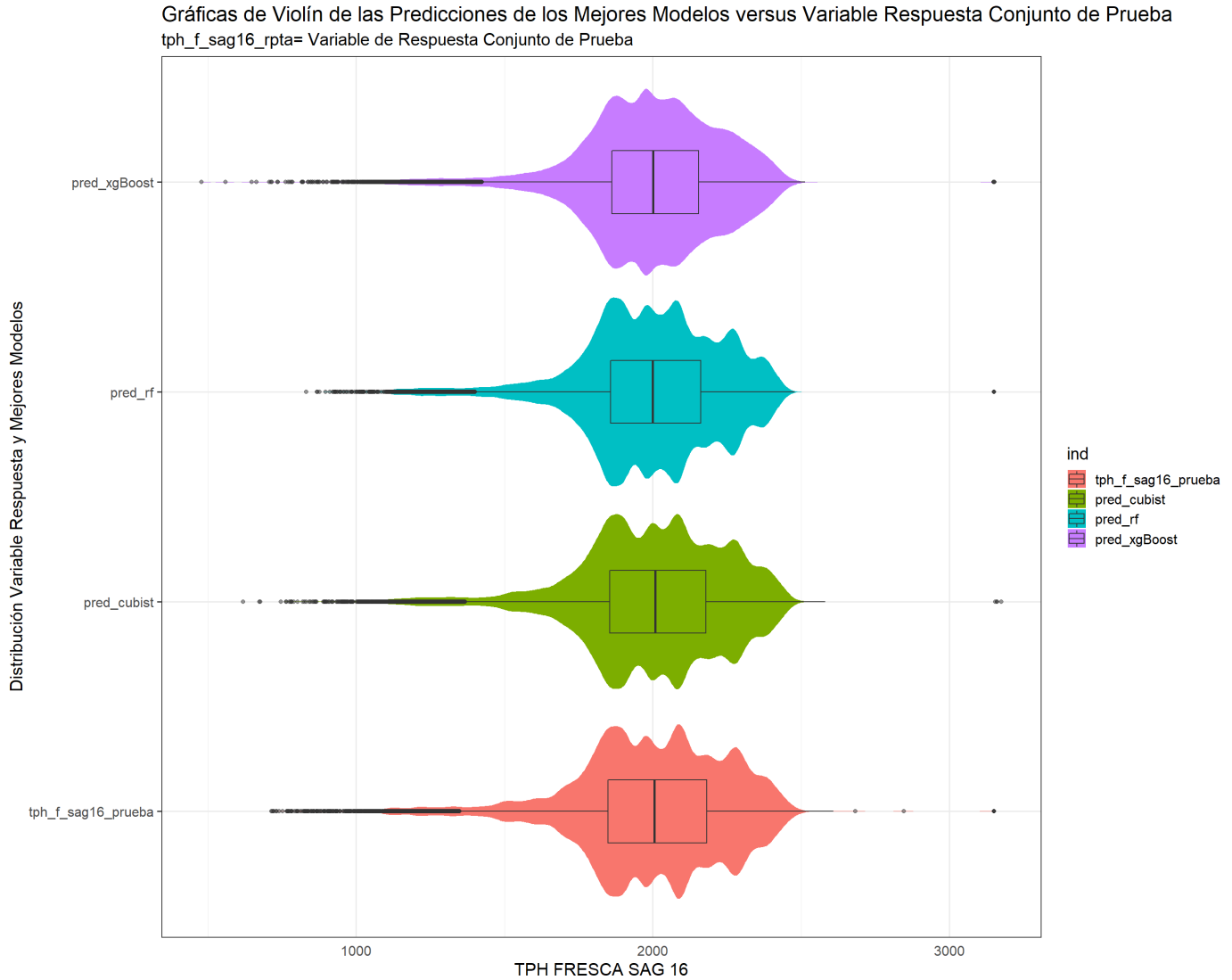
```
#Prediccion Principales Modelos en el Conjunto de Prueba
set.seed(7)
pred_cubist <- predict(fit.cubist, newdata = preproCorPrueba[,1:12])
pred_rf <- predict(fit.rf, newdata = preproCorPrueba[,1:12])
pred_xgBoost <- predict(fit.xgBoost, newdata = preproCorPrueba[,1:12])

#RMSE PRUEBA
pred_RMSE <- data.frame(CUBIST= RMSE(pred_cubist, preproCorPrueba[["tph_f_sag16"]]),
                        RF= RMSE(pred_rf, preproCorPrueba[["tph_f_sag16"]]),
                        XGBOOST= RMSE(pred_xgBoost, preproCorPrueba[["tph_f_sag16"]]))
datatable(pred_RMSE)%>%
  formatRound(columns=c('CUBIST', 'RF', 'XGBOOST'), digits=2)
```

```
#R2 PRUEBA
pred_R2 <- data.frame(CUBIST= R2(pred_cubist, preproCorPrueba[["tph_f_sag16"]]),
                      RF= R2(pred_rf, preproCorPrueba[["tph_f_sag16"]]),
                      XGBOOST= R2(pred_xgBoost, preproCorPrueba[["tph_f_sag16"]]))
datatable(pred_R2) %>%
  formatRound(columns=c('CUBIST', 'RF', 'XGBOOST'), digits=4)
```

Se puede ver que los resultados reportados son casi idénticos a los obtenidos en los sub conjuntos de prueba del conjunto de entrenamiento, con lo cual se validan los resultados antes obtenidos y también el comportamiento de los modelos en datos que no han

procesado previamente (nuevos datos). Lo anterior se puede apreciar a continuación, en donde se muestran las distribuciones de las predicciones de los mejores modelos **pred_xgBoost**, **pred_rf** y **pred_cubist** contrastadas con la distribución de la variable respuesta **"tph_f_sag16"** del Conjunto de Prueba (indicada como tph_f_sag16_prueba):

```
#Mejores Modelos
graf<-data.frame(Fecha=seq.POSIXt(ISOdate(2018,1,1), by="5 min", length.out=34158), tph_f_sag16_prueba=prepr
oCorPrueba$tph_f_sag16, pred_cubist, pred_rf, pred_xgBoost)

graf %>%
  dplyr::select(-Fecha) %>%
  stack() %>%
  ggplot(aes(x=ind, y=values, fill=ind))+
  geom_violin(width=0.9, bw=20, col=NA)+
  geom_boxplot(alpha=0.5, width=0.3)+
  coord_flip()+
  labs(title = "Gráficas de Violín de las Predicciones de los Mejores Modelos versus Variable Respuesta Conj
unto de Prueba",
       subtitle = "tph_f_sag16_rpta= Variable de Respuesta Conjunto de Prueba",
       y = "TPH FRESCA SAG 16",
       x = "Distribución Variable Respuesta y Mejores Modelos")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```

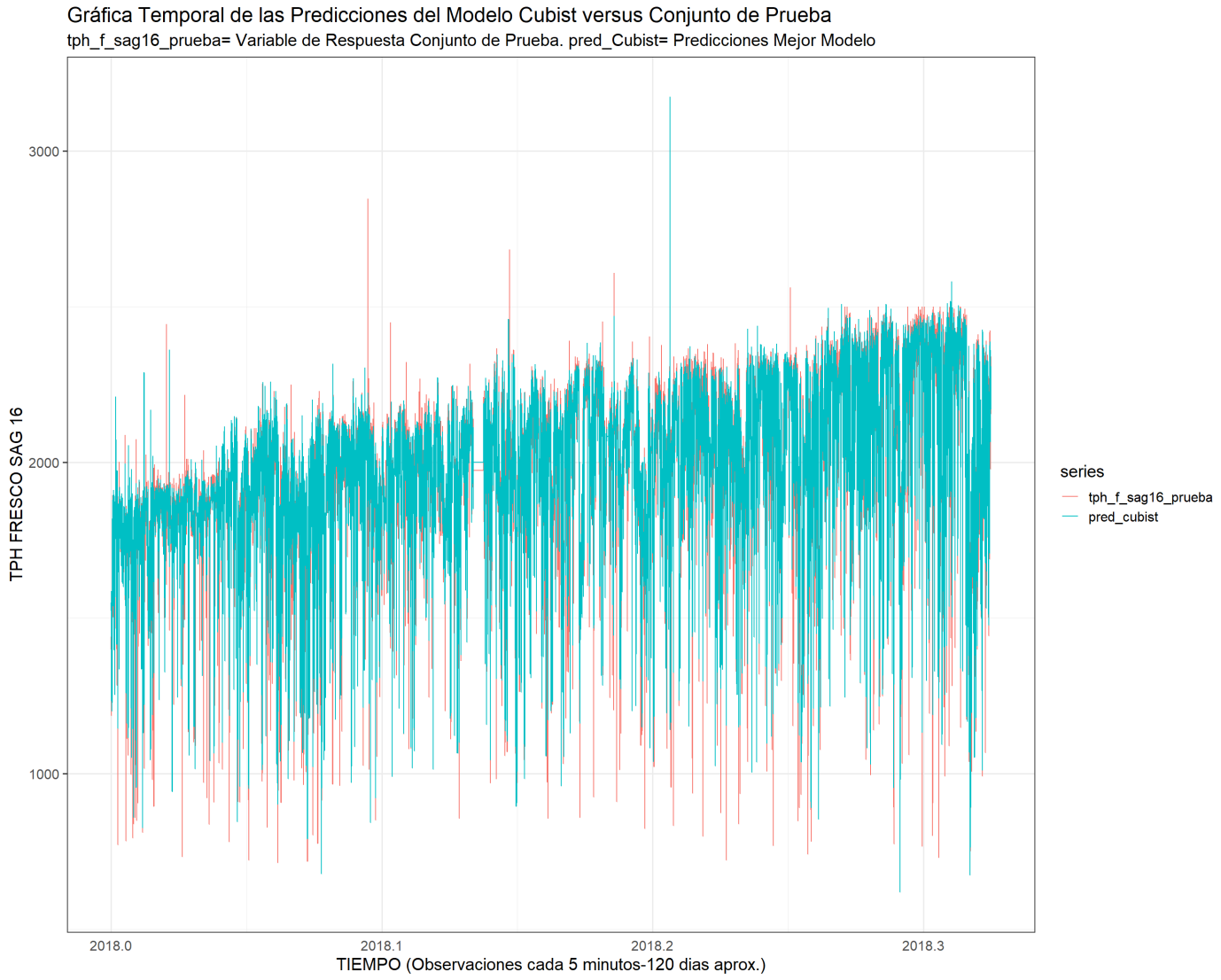


Se puede apreciar la gráfica de violín de los mejores modelos y de los datos de la variable de respuesta **tph_f_sag16_prueba** del conjunto de prueba original, la cual no se ocupó en la construcción de los modelos y por tanto son valores legítimos contra los cuales las predicciones de los modelos se pueden comparar.

Se puede apreciar que los modelos **cubist** y **random forest** son los que mejor simulan la agrupación y distribución de los datos, esto último confirma los valores de RMSE y R2 obtenidos por estos modelos, los cuales fueron los más altos.

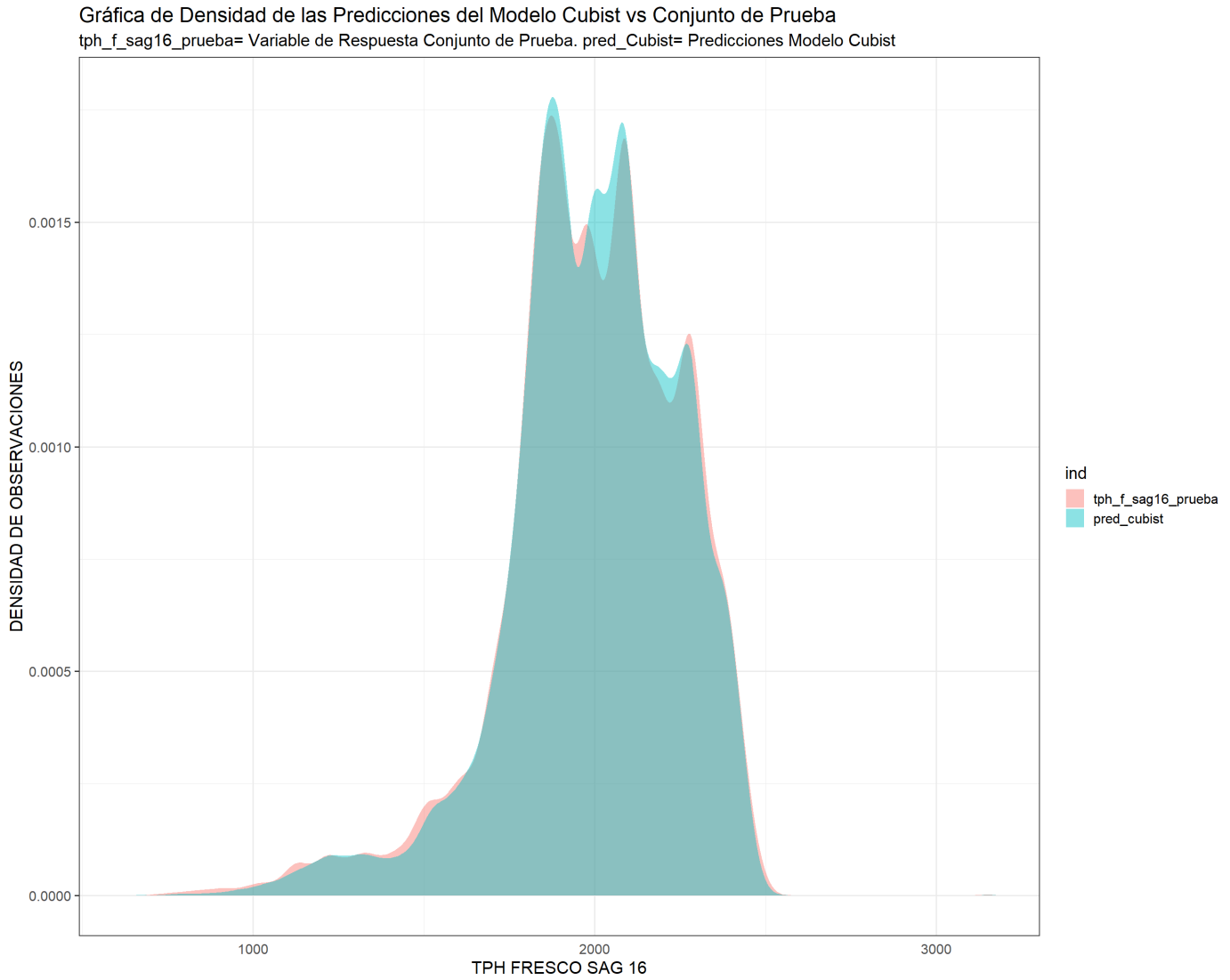
A continuación se muestran diferentes gráficas de las predicciones del modelo **Cubist**, el cual reportó los mejores resultados, versus la variable respuesta del conjunto de prueba:

```
# Mejor Modelo
graf %>%
 dplyr::select(-Fecha, -pred_rf, -pred_xgBoost) %>%
 ts(start = c(2018,1), frequency = 105120) %>%
 forecast:: autoplot()+
 labs(title = "Gráfica Temporal de las Predicciones del Modelo Cubist versus Conjunto de Prueba",
       subtitle = "tph_f_sag16_prueba= Variable de Respuesta Conjunto de Prueba. pred_Cubist= Predicciones M
ejor Modelo",
       y = "TPH FRESCO SAG 16",
       x = "TIEMPO (Observaciones cada 5 minutos-120 dias aprox.)")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



Se puede apreciar que en general las predicciones del modelo**Cubist** (en verde) siguen la tendencia de los valores de la variable respuesta (rosa). La gráfica se ve congestionada debido al alto número de observaciones, aproximadamente 35 mil observaciones correspondientes al conjunto de Prueba original (20% de los datos originales).

```
graf %>%
  dplyr::select(-Fecha, -pred_rf, -pred_xgBoost) %>%
  stack() %>%
  ggplot(aes(x=values, fill=ind))+
  geom_density(alpha=0.45, col=NA)+
  labs(title = "Gráfica de Densidad de las Predicciones del Modelo Cubist vs Conjunto de Prueba",
       subtitle = "tph_f_sag16_prueba= Variable de Respuesta Conjunto de Prueba. pred_Cubist= Predicciones M
odelo Cubist",
       y = "DENSIDAD DE OBSERVACIONES",
       x = "TPH FRESCO SAG 16")+
  theme_bw(base_size = 15)+
  theme(axis.ticks.x = element_blank())
```



En la gráfica se pueden ver las distribuciones de las predicciones del modelo**Cubist** en verde claro y los valores de la variable respuesta **tph_f_sag16_prueba** en rosa. El color verde oscuro da cuenta de como ambas distribuciones se sobreponen, idealmente este fenómeno se debería reportar en la totalidad de los datos lo cual sería un indicador de que la distribución de las predicciones toman el mismo valor de la distribución de la variable respuesta.

En nuestro caso las predicciones se comportan de forma muy cercana a los valores de la variable respuesta salvo por 2 intervalos a la altura de las 2000 y 2300 tph. Lo anterior nos demuestra, al igual que las gráficas anteriores, el alto poder predictivo del modelo Cubist.

CONCLUSIONES

1. Se ha desarrollado un procedimiento para la ejecución de modelos supervisados de regresión vía métodos de "Machine Learning" con diferentes herramientas y librerías en plataforma R Studio, lo anterior, tendiente a generar algoritmos que sean capaces de maximizar el poder predictivo de la variable de interés.
2. El modelo **Cubist** fue el que reportó los mejores resultados en el conjunto de Prueba, alcanzando un RMSE de 82.81 y un R2 de 90.12%. Seguido por los modelos **Random Forest** y **xgBoost**.
3. No fue posible realizar un ajuste de los Hiperparámetros de los modelos indicados en el punto anterior dado el alto tiempo de entrenamiento de estos. Esta importante etapa, que podría mejorar aun más el poder predictivo de los modelos, se abordará en una etapa posterior.
4. El modelo de ensamble se ejecutó con éxito mostrando una mejoría marginal. Lamentablemente no se pudo validar, vía predicción con el conjunto de prueba original, lo anterior debido a un error que no fue posible depurar. Esto último se podría investigar en una etapa futura.
5. Es posible desarrollar un modelo con un alto poder predictivo de la tasa de procesamiento horaria de un molino SAG a partir de variables operacionales que se registran en línea, sin embargo, se sugiere incorporar predictores adicionales que puedan mejorar aún más el poder predictivo de los modelos, como por ejemplo, dureza de los minerales alimentados.