# Estimación de Storage para un Oracle Database 12c utilizando Oracle R

Por Francisco Riccio

## Introducción

Muchas organizaciones están alineadas a las mejores prácticas para la administración de servicios de TI utilizando el marco de referencia ITIL v3. En el ciclo de vida de ITIL v3, la fase de Diseño incluye el proceso de Management Capacity, el cual nos indica que debemos tomar en cuenta todos los recursos necesarios para llevar a cabo los servicios de TI y proveer las necesidades de la empresa a corto, medio y largo plazo.

Por este motivo, se hace muy importante poder estimar las capacidades de hardware (CPU, Memoria y Storage) que requiere nuestro servidor de base de datos en los próximos n meses.

Este artículo está enfocado a poder ayudar a calcular el espacio que se requerirá en un tiempo determinado en el futuro para una base de datos Oracle específica.

Cabe mencionar que pueden haber diversas soluciones para este requerimiento, incluyendo soluciones cognitivas que permitan ayudarnos en el cálculo. El método que presentaré no es complicado implementarlo y está basado en regresiones, teniendo un fundamento estadístico y de gran valor para la empresa.

Para cumplir dicho fin, nos apoyaremos de: PL/SQL, Oracle R Distribution y nuestros conceptos de Estadística Básica.

Oracle R Distribution es una redistribución del Open Source R y puede ser descargado de manera gratuita.

Más información del producto lo encontramos en el siguiente url:

http://www.oracle.com/technetwork/topics/bigdata/r-offerings-1566363.html

R es un es un entorno y lenguaje de programación gratuito con un enfoque en el análisis estadístico.

Más información del proyecto R lo podemos encontrar en el siguiente url: https://www.r-project.org/

Este artículo no será útil para aquella persona que esté buscando un mecanismo de estimar el tamaño de una base de datos nueva, debido a que la estimación de storage lo realizaremos utilizando el espacio histórico de la base de datos para proyectar.

## Implementación

### I. Setup

A continuación se definirá la implementación para cumplir el objetivo.

1. Creamos un usuario en la base de datos.

```
SQL> create user friccio identified by oracle;

User created.

SQL> grant connect, resource, unlimited tablespace to friccio;

Grant succeeded.

SQL> grant scheduler_admin to friccio;

Grant succeeded.

SQL> grant select any dictionary to friccio;

Grant succeeded.
```

2. Creamos una tabla que almacenará la siguiente información:

```
SQL> desc HISTORIAL_TBS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
NOMBRE                                    NOT NULL VARCHAR2(30)
TAMANO_FREE_GB                            NOT NULL NUMBER
TAMANO_OCUPADO_GB                         NOT NULL NUMBER
TAMANO_RESERVADO_GB                       NOT NULL NUMBER
FECHA                                     NOT NULL CHAR(8)
```

Los campos son:

- NOMBRE = Nombre del tablespace.
- TAMANO_FREE_GB = Tamaño libre del tablespace.
- TAMANO_OCUPA_GB = Tamaño de los datos almacenados en el tablespace.
- TAMANO_RESERVADO_GB = Espacio que ocupa el tablespace en el Sistema Operativo.
- FECHA = Fecha de la captura del dato en formato YYYYMMDD

A pesar que todos los campos no serán requeridos para el cálculo final como por ejemplo: Tamaño Libre o el Espacio Reservado, es recomendable almacenarlos para un futuro análisis como también es

importante llevar este control a nivel de tablespace porque nos permite realizar proyecciones únicamente a un tablespace específico.

La tabla HISTORIAL_TBS debe ser creado con el usuario previamente creado.

3. Creamos un Stored Procedure que se encargará de poblar la información en la tabla previamente creada.

```
SQL> create or replace procedure spu_historial_tbs
  2  --Programado por Francisco Riccio.
  3  is
  4   cursor v_tbs is
  5   select nombre, round(tamano_free_gb,2) as tamano_free_gb,
  6   round(tamano_reservado_gb-tamano_free_gb,2) as tamano_ocupado_gb,
  7   round(tamano_reservado_gb,2) as tamano_reservado_gb
  8   from
  9   (
 10    select nombre, tamano_free_gb,
 11    (
 12     select tamano_reservado_gb
 13     from
 14     (
 15      (select tablespace_name, sum(bytes)/1024/1024/1024 as tamano_reservado_gb
 16      from dba_data_files
 17      group by tablespace_name)
 18     )
 19     where tablespace_name=nombre
 20    ) as tamano_reservado_gb
 21    from (
 22     select tablespace_name nombre, sum(bytes)/1024/1024/1024 as tamano_free_gb
 23     from dba_free_space
 24     group by tablespace_name
 25    )
 26   );
 27  begin
 28   for c in v_tbs loop
 29    insert into HISTORIAL_TBS (nombre,tamano_free_gb,tamano_ocupado_gb,tamano_reservado_gb,fecha)
  values (c.nombre, c.tamano_free_gb, c.tamano_ocupado_gb,
 30   31                 c.tamano_reservado_gb,to_char(sysdate,'YYYYMMDD'));
 32   end loop;
 33   commit;
 34  end;
 35  /

Procedure created.
```

```
create or replace procedure spu_historial_tbs

--Programado por Francisco Riccio.

is

 cursor v_tbs is

 select nombre, round(tamano_free_gb,2) as tamano_free_gb,

 round(tamano_reservado_gb-tamano_free_gb,2) as tamano_ocupado_gb,

 round(tamano_reservado_gb,2) as tamano_reservado_gb

 from

 (
```

```
 select nombre, tamano_free_gb,
 (
  select tamano_reservado_gb
  from
  (
   (select tablespace_name, sum(bytes)/1024/1024/1024 as tamano_reservado_gb
   from dba_data_files
   group by tablespace_name)
  )
  where tablespace_name=nombre
 ) as tamano_reservado_gb
 from (
  select tablespace_name nombre, sum(bytes)/1024/1024/1024 as tamano_free_gb
  from dba_free_space
  group by tablespace_name
 )
);
begin
 for c in v_tbs loop
  insert into HISTORIAL_TBS (nombre,tamano_free_gb,tamano_ocupado_gb,tamano_reservado_gb,fecha)
  values (c.nombre, c.tamano_free_gb, c.tamano_ocupado_gb,
                        c.tamano_reservado_gb,to_char(sysdate,'YYYYMMDD'));
 end loop;
 commit;
end;
/
```

4. Creamos un JOB que ejecutará el Stored Procedure de manera diaria.

a) Creamos un SCHEDULE.

execute
dbms_scheduler.create_schedule(schedule_name=>'**sched_01_hor**',start_date=>trunc(SYSTIMESTAMP,'day'),repeat_interval=>'freq=hourly;bysecond=00;interval=1',end_date=>null,comments=>'Frequencia de 1 hora');

/

b) Creamos un PROGRAM

execute
dbms_scheduler.create_program(program_name=>'**prog_historial_tbs**',program_type=>'stored_procedure',program_action=>'spu_historial_tbs',enabled=>true,comments=>'Registro del tamano de los tablespaces');

/

c) Creamos el JOB

execute
dbms_scheduler.create_job(job_name=>'**job_historial_tbs**',program_name=>'prog_historial_tbs',schedule_name=>'sched_01_hor',enabled=>true,auto_drop=>false,comments=>'Registro del tamano de los tablespaces cada dia');

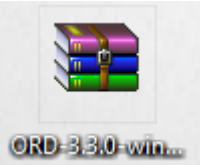
/

5. Procedemos a Instalar Oracle R en nuestra computadora personal.

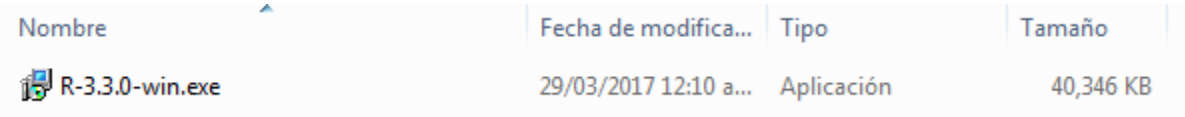
Oracle R lo podemos descargar del siguiente url:

http://www.oracle.com/technetwork/database/database-technologies/r/r-distribution/downloads/index.html

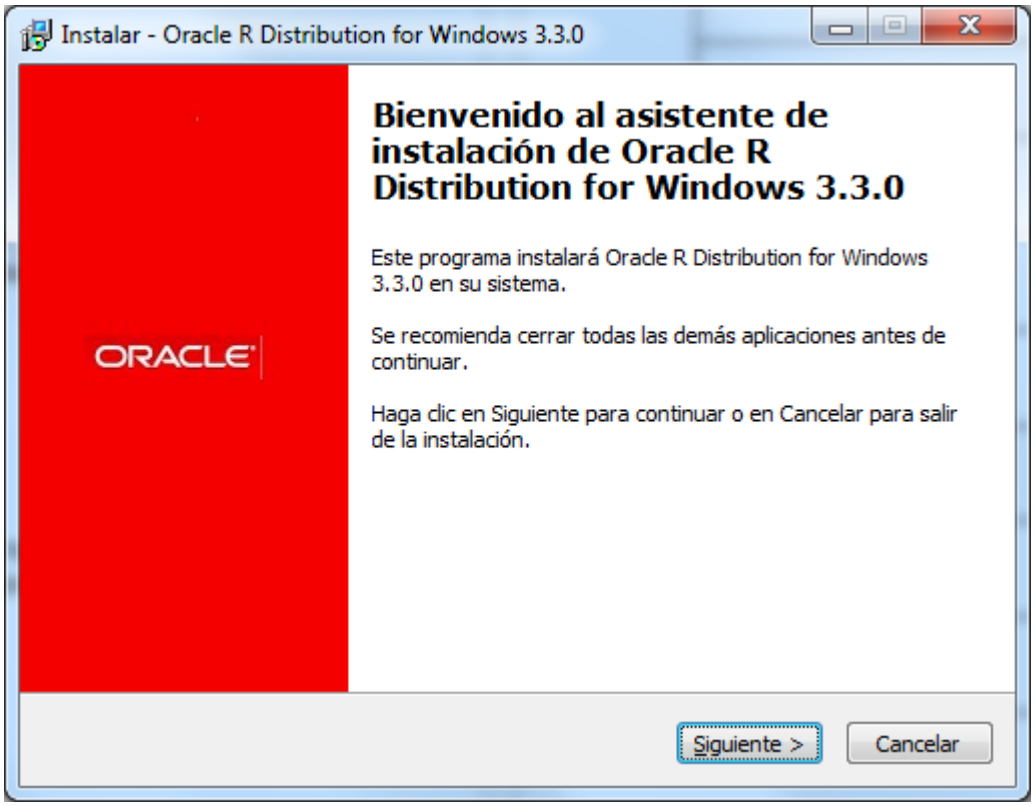
Descomprimimos el archivo descargado:

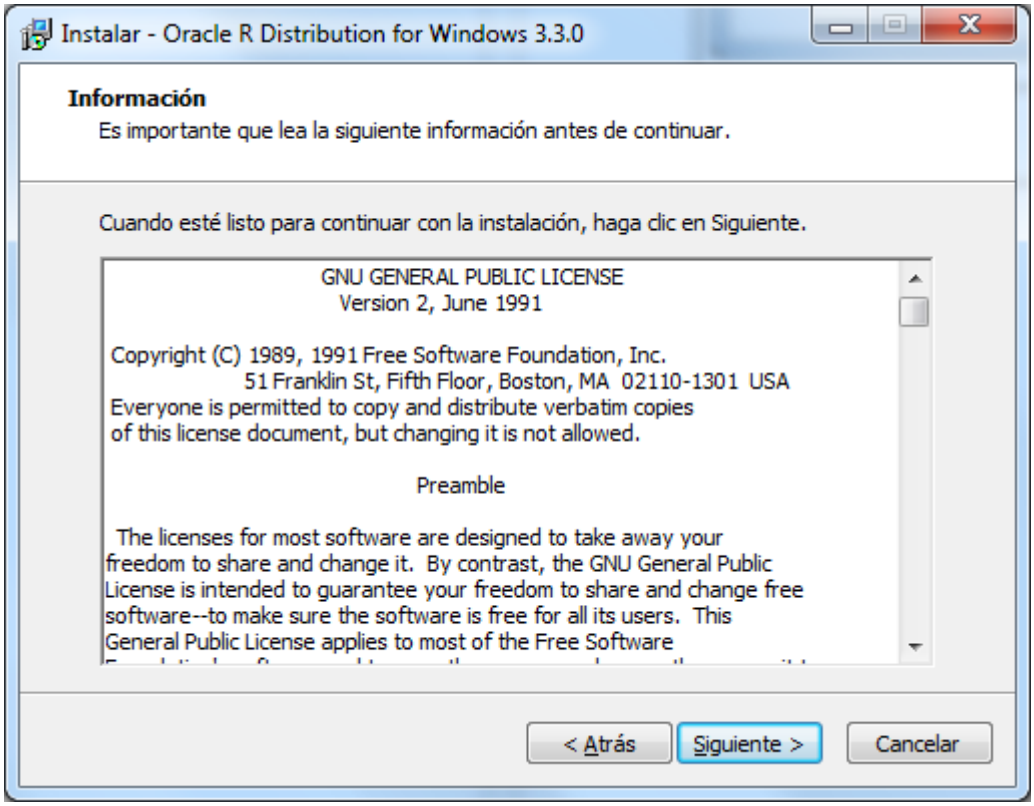


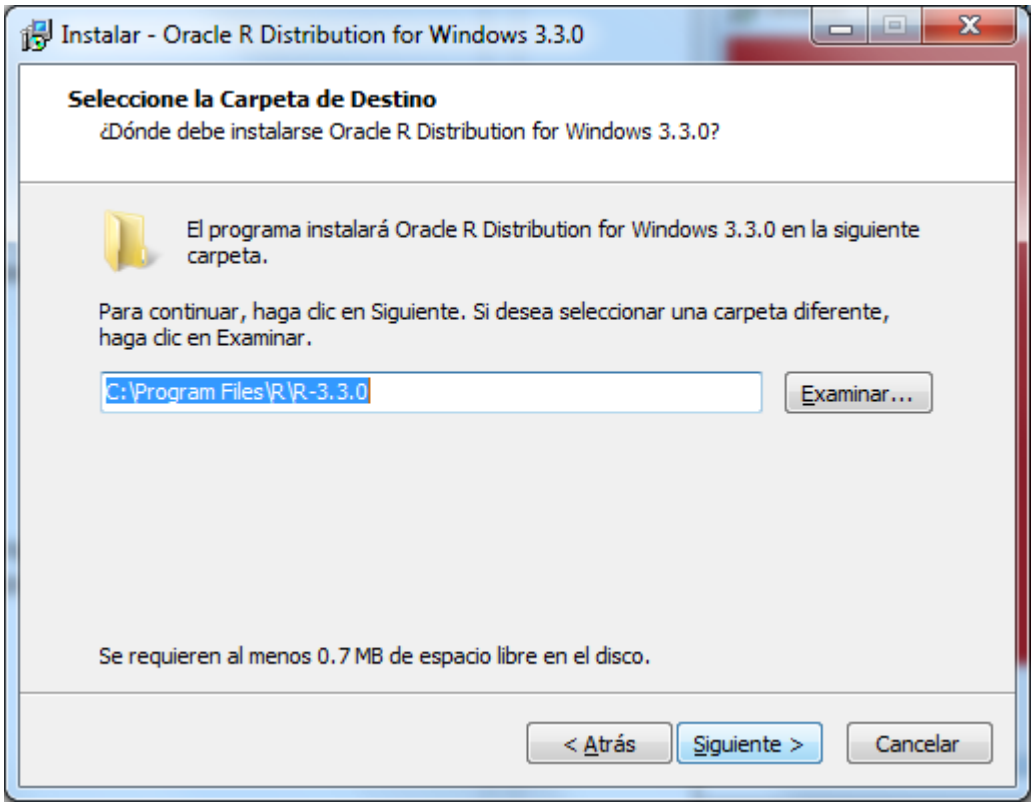

Ejecutamos el instalador:

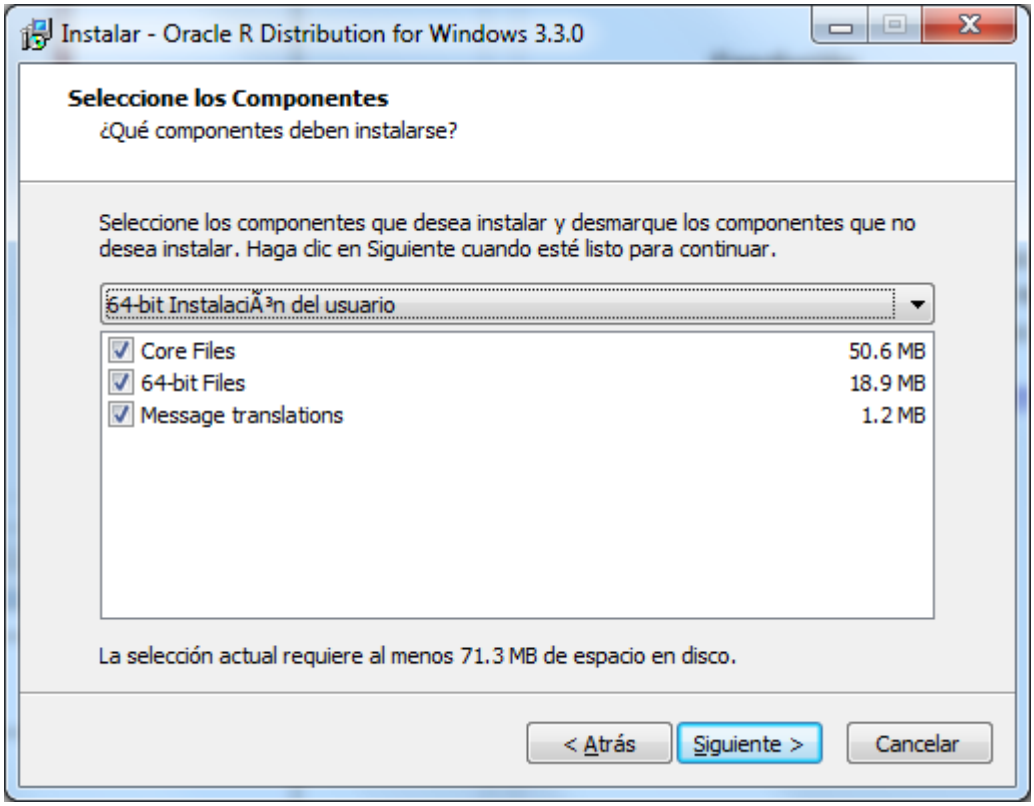


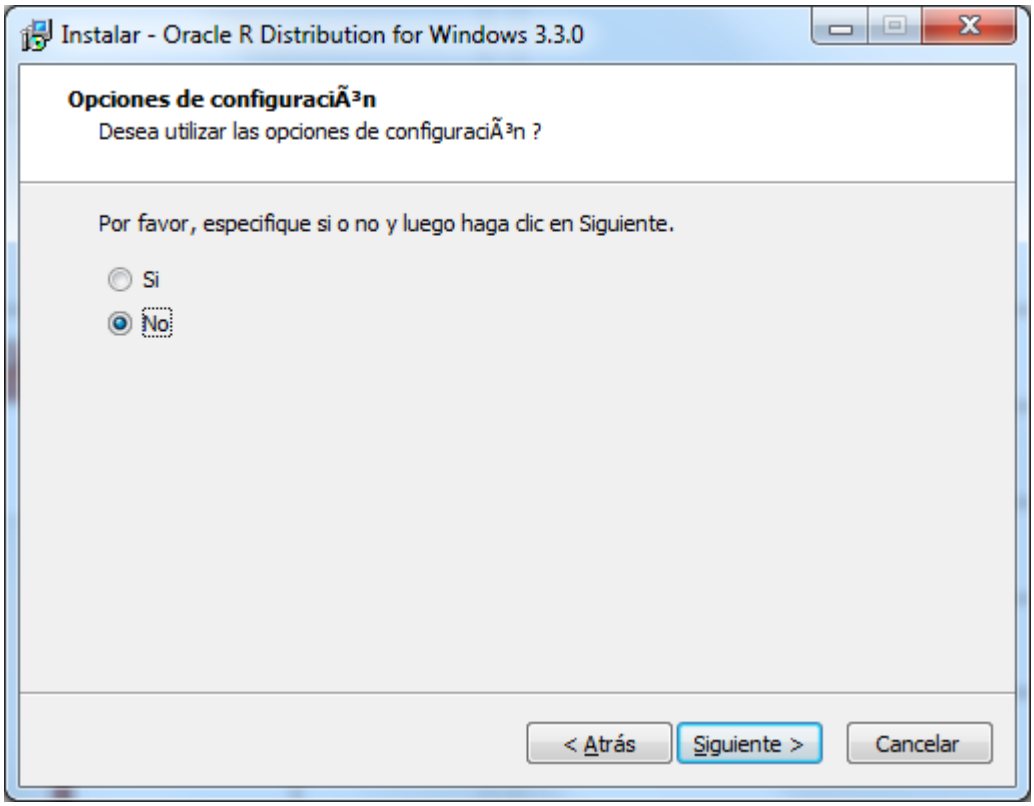

A continuación se adjuntan las pantallas de la instalación:

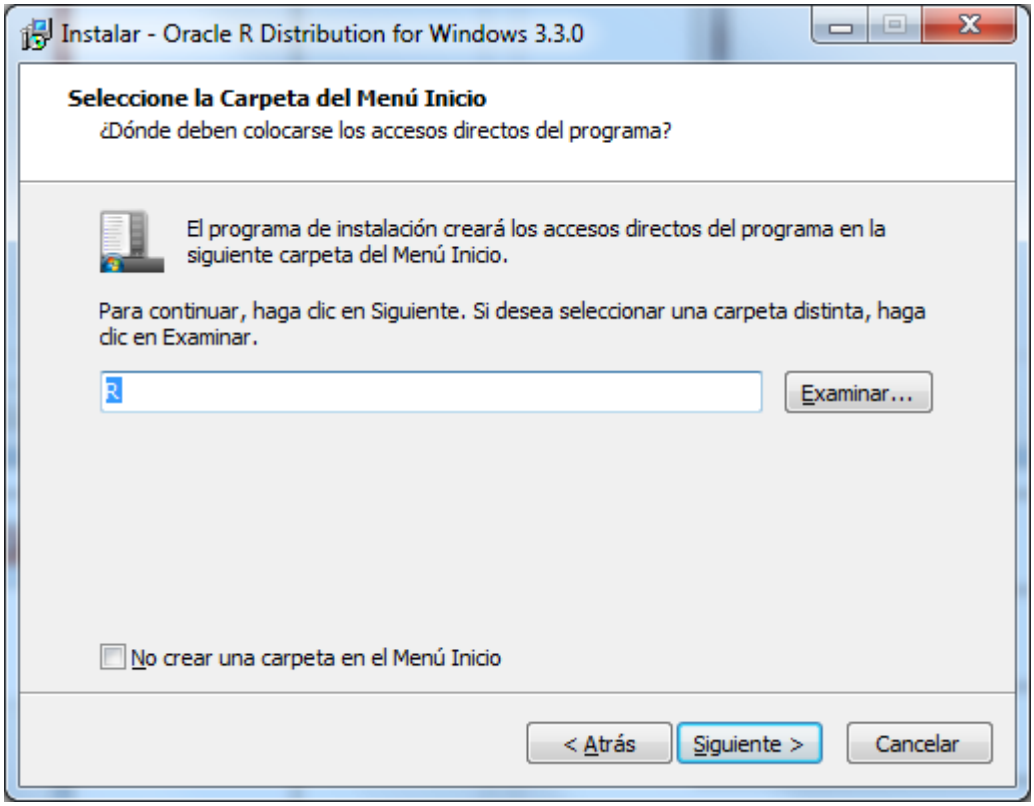

Instalar - Oracle R Distribution for Windows 3.3.0
ORACLE
Bienvenido al asistente de instalación de Oracle R Distribution for Windows 3.3.0
Este programa instalará Oracle R Distribution for Windows 3.3.0 en su sistema.
Se recomienda cerrar todas las demás aplicaciones antes de continuar.
Haga clic en Siguiente para continuar o en Cancelar para salir de la instalación.
Siguiente >
Cancelar

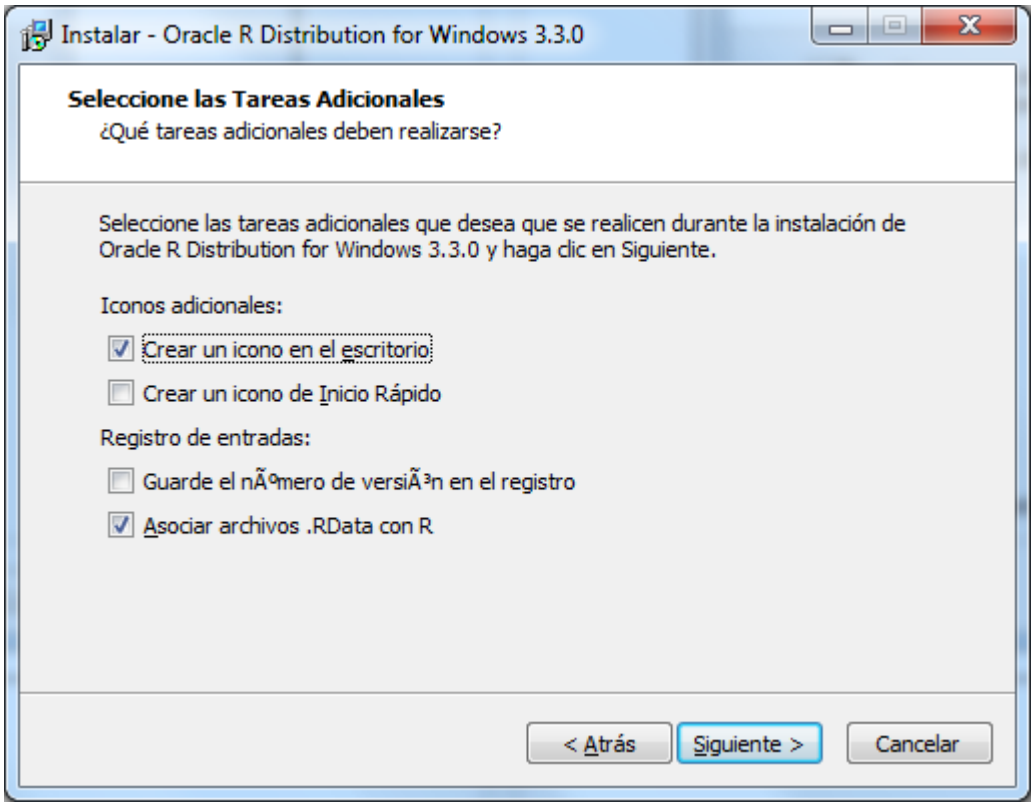

Instalar - Oracle R Distribution for Windows 3.3.0
Información
Es importante que lea la siguiente información antes de continuar.
Cuando esté listo para continuar con la instalación, haga clic en Siguiente.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
< Atrás
Siguiente >
Cancelar

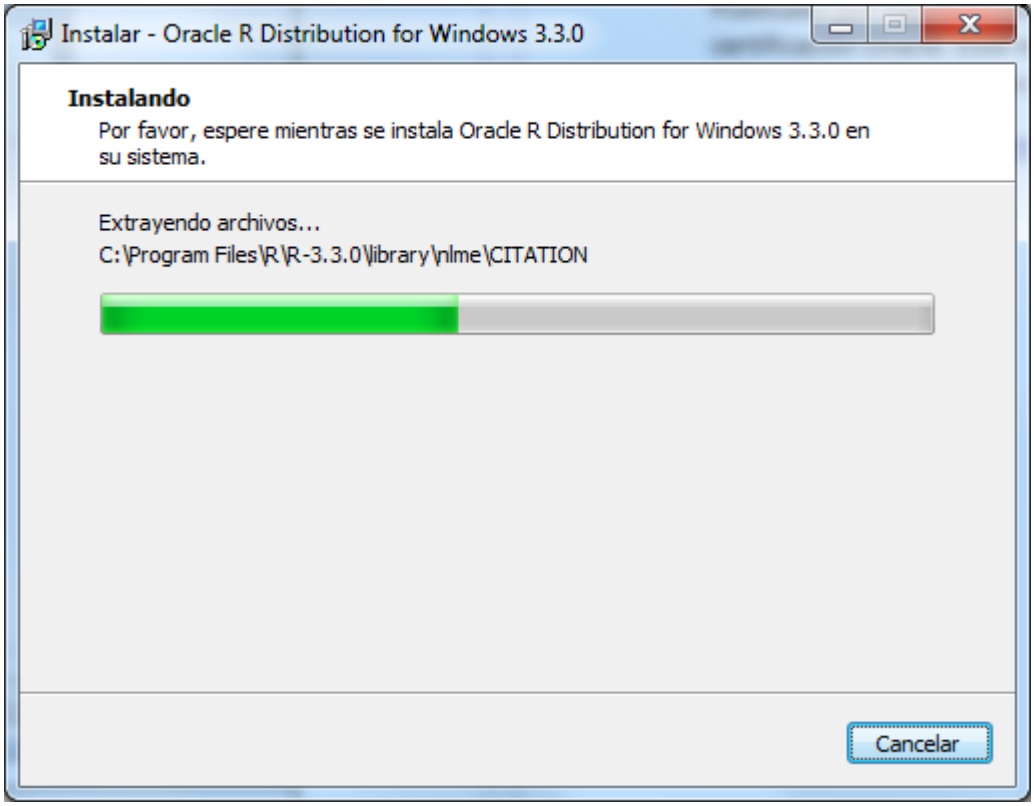

Instalar - Oracle R Distribution for Windows 3.3.0
Seleccione la Carpeta de Destino
¿Dónde debe instalarse Oracle R Distribution for Windows 3.3.0?
El programa instalará Oracle R Distribution for Windows 3.3.0 en la siguiente carpeta.
Para continuar, haga clic en Siguiente. Si desea seleccionar una carpeta diferente, haga clic en Examinar.
C:\Program Files\R\R-3.3.0
Examinar...
Se requieren al menos 0.7 MB de espacio libre en el disco.
< Atrás
Siguiente >
Cancelar

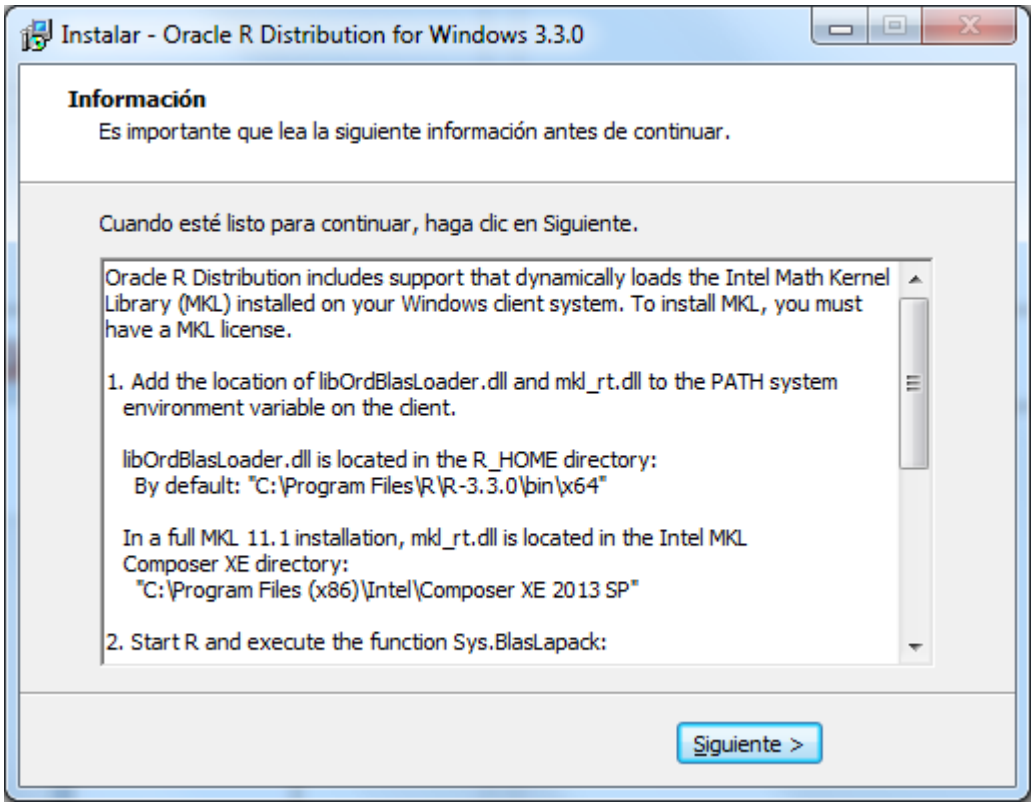

Instalar - Oracle R Distribution for Windows 3.3.0
Seleccione los Componentes
¿Qué componentes deben instalarse?
Seleccione los componentes que desea instalar y desmarque los componentes que no desea instalar. Haga clic en Siguiente cuando esté listo para continuar.
64-bit InstalaciÃ³n del usuario
Core Files 50.6 MB
64-bit Files 18.9 MB
Message translations 1.2 MB
La selección actual requiere al menos 71.3 MB de espacio en disco.
< Atrás
Siguiente >
Cancelar

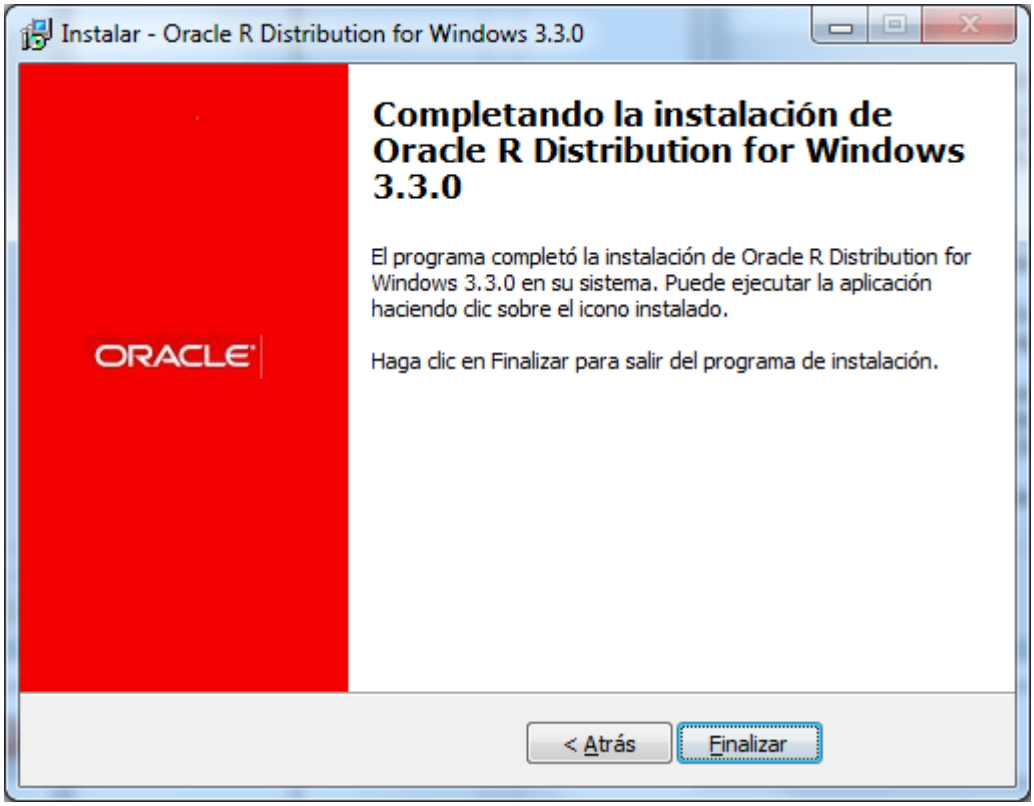

Instalar - Oracle R Distribution for Windows 3.3.0
Opciones de configuraciÃ³n
Desea utilizar las opciones de configuraciÃ³n ?
Por favor, especifique si o no y luego haga clic en Siguiente.
Si
No
< Atrás
Siguiente >
Cancelar


Instalar - Oracle R Distribution for Windows 3.3.0
Seleccione la Carpeta del Menú Inicio
¿Dónde deben colocarse los accesos directos del programa?
El programa de instalación creará los accesos directos del programa en la siguiente carpeta del Menú Inicio.
Para continuar, haga clic en Siguiente. Si desea seleccionar una carpeta distinta, haga clic en Examinar.
Examinar...
No crear una carpeta en el Menú Inicio
< Atrás
Siguiente >
Cancelar


Instalar - Oracle R Distribution for Windows 3.3.0
Seleccione las Tareas Adicionales
¿Qué tareas adicionales deben realizarse?
Seleccione las tareas adicionales que desea que se realicen durante la instalación de Oracle R Distribution for Windows 3.3.0 y haga clic en Siguiente.
Iconos adicionales:
Crear un icono en el escritorio
Crear un icono de Inicio Rápido
Registro de entradas:
Guarde el nÃºmero de versiÃ³n en el registro
Asociar archivos .RData con R
< Atrás
Siguiente >
Cancelar


Instalar - Oracle R Distribution for Windows 3.3.0
Instalando
Por favor, espere mientras se instala Oracle R Distribution for Windows 3.3.0 en su sistema.
Extrayendo archivos...
C:\Program Files\R\R-3.3.0\library\nlme\CITATION
Cancelar


Instalar - Oracle R Distribution for Windows 3.3.0
Información
Es importante que lea la siguiente información antes de continuar.
Cuando esté listo para continuar, haga clic en Siguiente.
Oracle R Distribution includes support that dynamically loads the Intel Math Kernel Library (MKL) installed on your Windows client system. To install MKL, you must have a MKL license.
1. Add the location of libOrdBlasLoader.dll and mkl_rt.dll to the PATH system environment variable on the client.
libOrdBlasLoader.dll is located in the R_HOME directory:
By default: "C:\Program Files\R\R-3.3.0\bin\x64"
In a full MKL 11.1 installation, mkl_rt.dll is located in the Intel MKL Composer XE directory:
"C:\Program Files (x86)\Intel\Composer XE 2013 SP"
2. Start R and execute the function Sys.BlasLapack:
Siguiente >


Instalar - Oracle R Distribution for Windows 3.3.0
ORACLE
Completando la instalación de Oracle R Distribution for Windows 3.3.0
El programa completó la instalación de Oracle R Distribution for Windows 3.3.0 en su sistema. Puede ejecutar la aplicación haciendo clic sobre el icono instalado.
Haga clic en Finalizar para salir del programa de instalación.
< Atrás
Finalizar

6. Descargamos el paquete RORacle

Este paquete nos permitirá conectarnos a la base de datos y extraer los datos que requerimos para el análisis.

El paquete lo descargamos del siguiente url:

http://www.oracle.com/technetwork/database/database-technologies/r/roracle/downloads/index.html

Antes de proceder a instalar los paquetes es importante asegurarnos que tenemos instalado **Oracle Instant Client o un Oracle Database Client** en nuestra computadora personal.

Asegurándonos que tenemos instalado, procedemos a instalar primero el paquetes DBI, el cual define las interfaces hacia las bases de datos.

La instalación lo realizamos a través de la instrucción **install.packages** como a continuación se presenta:

```
Oracle Distribution of R version 3.3.0 (2016-05-03) -- "Supposedly Educational"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

You are using Oracle's distribution of R. Please contact
Oracle Support for any problems you encounter with this
distribution.

> install.packages("DBI")
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/DBI_0.6-1.zip'
Content type 'application/zip' length 743287 bytes (725 KB)
downloaded 725 KB

package 'DBI' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        E:\Temp\Rtmpgh7WXQ\downloaded_packages
```

Terminado la instalación prodemos a instalar el paquete ROracle.

```
> setwd('C:\\Program Files\\R\\R-3.3.0\\packages')
> install.packages('ROracle_1.3-1.zip')
inferring 'repos = NULL' from 'pkgs'
package 'ROracle' successfully unpacked and MD5 sums checked
```

Con estos 2 pasos estamos listos para realizar una prueba de conexión hacia la base de datos Oracle 12c.

```
> library('ROracle')
Loading required package: DBI
Warning message:
package 'DBI' was built under R version 3.3.3
> drv <- dbDriver("Oracle")
> con <- dbConnect(drv,"friccio","oracle",dbname='132.68.1.30:1521/PRD')
> dbReadTable(con, 'DUAL')
  DUMMY
1     X
```

Asimismo veremos en la base de datos una conexión realizada:

```
SQL> select terminal,program from v$session where username='FRICCIO';

TERMINAL                       PROGRAM
------------------------------ ----------------------------------------
PC-01                          Rgui.exe
```

Para conocer todas las funciones disponibles del paquete podemos descargar su documentación en el siguiente url: https://cran.r-project.org/web/packages/ROracle/ROracle.pdf

7. Construcción del Reporte de Espacio

Con la siguiente sentencia SQL podemos determinar el máximo tamaño que ha tenido la base de datos en cada mes desde que comenzó a ejecutarse el JOB.

Dicha sentencia SQL lo encapsularemos en una vista llamada VREPORTE_ESPACIO.

```
SQL> connect / as sysdba
Connected.
SQL> grant create view to friccio;

Grant succeeded.

SQL> connect friccio/oracle
Connected.

SQL> create view VREPORTE_ESPACIO
  2  as
  3   select rownum as num_mes, total from (
  4    select fecha, max(total) as total
  5    from
  6    (
  7     select substr(fecha,1,6) as fecha, total
  8     from
  9     (
 10      select fecha, sum(tamano_ocupado_gb) as total
 11      from historial_tbs
 12      group by fecha
 13      order by 1
 14     )
 15    )
 16   group by fecha
 17   order by 1 );

View created.
```

**II. Análisis de los Datos**

Llegado a este punto, tenemos todo listo para analizar los datos.

**Nota:** Para un mejor análisis, utilizaré los datos de un servidor de base de datos Productivo cuya información ha venido registrándose desde el 13 de Febrero del 2012.

a) Obtendremos los datos de la vista VREPORTE

Ejecutamos el siguiente código en Oracle R Distribution.

```
> library('ROracle')
Loading required package: DBI
Warning message:
package 'DBI' was built under R version 3.3.3
> drv <- dbDriver("Oracle")
> con <- dbConnect(drv,"friccio","oracle",dbname='132.68.1.30:1521/PRD')
> sql <- "select * from VREPORTE_ESPACIO"
> rows <- dbGetQuery(con,sql)
> print(rows)
   NUM_MES  TOTAL
1        1 187.80
2        2 198.71
3        3 189.36
4        4 196.13

...

62      62 701.55
63      63 733.92
> dbDisconnect(con)
[1] TRUE
```

```
library('ROracle')

drv <- dbDriver("Oracle")

con <- dbConnect(drv,"friccio","oracle",dbname='132.68.1.30:1521/PRD')

sql <- "select * from VREPORTE_ESPACIO"

rows <- dbGetQuery(con,sql)

print(rows)

dbDisconnect(con)
```

Toda la información de las filas extraídas ha sido almacenada en una variable llamada rows, que ha sido definida en el código.

Podemos obtener un resumen de la información extraída con la instrucción **summary**.

```
> summary(rows)
    NUM_MES          TOTAL
 Min.   : 1.0   Min.   :187.8
 1st Qu.:16.5   1st Qu.:277.8
 Median :32.0   Median :362.3
 Mean   :32.0   Mean   :386.9
 3rd Qu.:47.5   3rd Qu.:469.2
 Max.   :63.0   Max.   :733.9
```

Aquí podemos ver que el tamaño de la base de datos inició en 187.8 GB y a la fecha ha llegado a un valor de 733.9 GB en 63 meses.

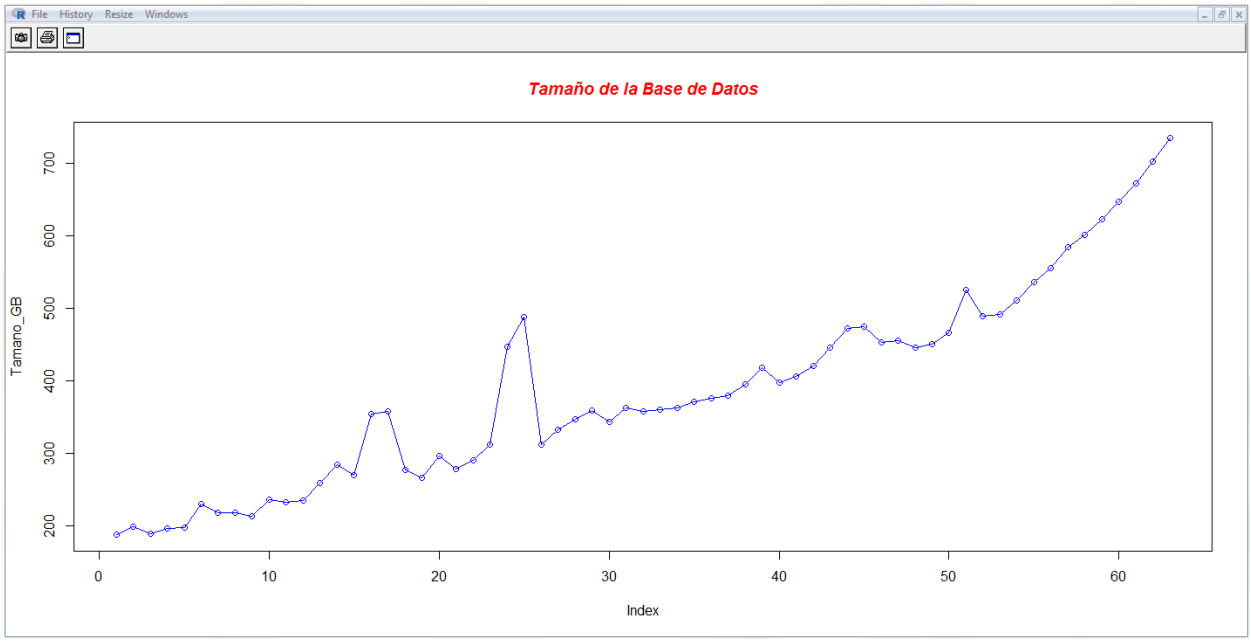
Oracle R Distribution nos permite realizar diferentes gráficos con la finalidad de visualizar los datos extraídos. En este caso realizaré un gráfico de líneas.

El código es el siguiente:

```
> Tamano_GB <- rows[,2]
> plot(Tamano_GB, type="o", col="blue")
> title(main="Tamaño de la Base de Datos", col.main="red", font.main=4)
```

Nota: Hemos creado una variable Tamano_GB que copiará los datos de la columna Total de la variable Rows.

La salida del reporte es la siguiente:



b) Regresión Lineal

El análisis de regresión es un proceso estadístico para estimar las relaciones entre variables, en nuestro caso son el tiempo y el espacio ocupado.

Existen diversos tipos de Regresiones como: Lineal, Lineal Múltiple, Exponencial, Logarítmica, Polinomial y otros.

Por experiencia el tamaño de las bases de datos suelen crecer bajo un modelo de regresión lineal, salvo en casos donde se estén ejecutando pruebas de esfuerzo u operaciones de carga masiva diarias por un período constante, logrando distorsionar los datos.

Como nuestra única variable que tenemos registrada es el tiempo, tendremos que modelar la regresión lineal simple. Si identificamos otras variables que puedan explicar el crecimiento del espacio y las registramos, deberíamos utilizar un modelo de regresión lineal múltiple.

A pesar de ello, igual validaremos el modelo de regresión una vez calculado en Oracle R.

Recordemos que el Modelo Lineal tiene la siguiente ecuación:

**Y (Espacio GB) = Bo + B1 * X (# Mes)**

Además debe cumplir con los siguientes criterios:

- Validación #1

  Prueba de Hipótesis:

  Ho: B1 = 0

  H1: B1 ≠ 0 (RHo)

  Debemos asegurarnos que B1 no tendrá el valor de 0, sino sería una línea constante el modelo.

- Validación #2

  Validamos que la variable # de Mes tenga una significancia en el modelo.

- Validación #3

  Calculamos el coeficiente de determinación $R^2$, el cual expresa el porcentaje de la variabilidad total que es explicada por el modelo.

- Validación #4

  Validar que los errores aleatorios $e_1$, $e_n$, etc; asociados a cualquier par de valores asociados a la variable dependiente Y, son siempre independientes. Esta prueba lo validamos a través de la prueba de Durbin-Watson.

- Validación #5

  Los residuos (los errores de los puntos con respecto al modelo) deben seguir una distribución Normal. Esto lo validamos mediante la prueba de Kolmogorov-Smirnov.

Creamos una variable RegLineal que almacenará el resultado de la Regresión Lineal y posterior realizaremos las pruebas de validación.

Para calcular la regresión lineal utilizamos la instrucción **lm**.

```
> x <- rows[,1]
> y <- rows[,2]
> RegLineal <- lm(formula=y~x)
```

En la instrucción indicamos que **la variable Espacio (Y) es dependiente de la varible # de Mes (X)**.

Procederemos luego a validar el modelo.

c) Validación #1 - Modelo de Regresión Lineal

Instrucción: **summary**

```
> summary(RegLineal)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-55.362 -32.593 -10.782   6.947 149.663

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 161.6862    11.6876   13.83   <2e-16 ***
x             7.0364     0.3175   22.16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 45.83 on 61 degrees of freedom
Multiple R-squared:  0.8895,    Adjusted R-squared:  0.8877
F-statistic:   491 on 1 and 61 DF,  p-value: < 2.2e-16
```

El valor (F-statistics "p-value") es la probabilidad que el coeficiente B1 sea igual a 0. Si la probabilidad es menor a 5% (convención) entonces rechazamos Ho (Acorde a la prueba de Hipótesis).

El resultado dio: $2.2e^{-16}$ por lo tanto es menor a 5%, entonces rechazamos Ho. Por lo tanto, es aceptada esta validación.

d) Validación #2 – Modelo de Regresión Lineal

El valor Pr(>|t|) para la variable X debe ser menor a 5%. En este caso, se acepta la validación.

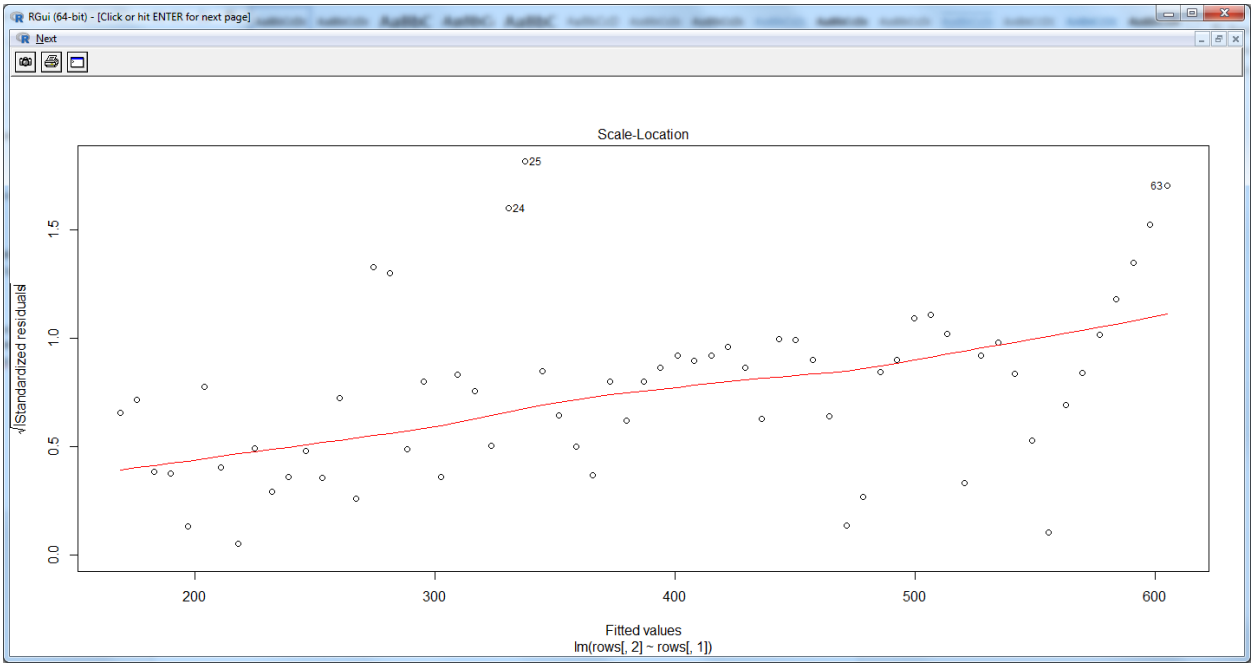
e) Validación #3 – Modelo de Regresión Lineal

El $R^2$ dio 0.8877. Este valor indica que el 88.77% de la dispersión del espacio utilizado por la base de datos es explicado por el modelo. Un valor más cercano al 100% es un mejor modelo.

Si deseamos graficar la regresión lineal ejecutamos la siguiente instrucción:

```
> plot(RegLineal)
```

Obtendremos el siguiente gráfico (Presionar ENTER por cada gráfico):




f) Validación #4 - Modelo de Regresión Lienal

Para realizar la prueba de Durbin y Watson primero debemos instalar el paquete lmtest.

```
> install.packages("lmtest")
--- Please select a CRAN mirror for use in this session ---
also installing the dependency 'zoo'

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/zoo_1.8-0.zip'
Content type 'application/zip' length 902427 bytes (881 KB)
downloaded 881 KB

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/lmtest_0.9-35.zip'
Content type 'application/zip' length 288519 bytes (281 KB)
downloaded 281 KB

package 'zoo' successfully unpacked and MD5 sums checked
package 'lmtest' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        E:\Temp\Rtmp2R1xfb\downloaded packages
```

Luego realizamos el análisis con las siguientes instrucciones:

- Library(lmtest)
- dwtest(variable)

```
> library(lmtest)
> dwtest(RegLineal, alternative="two.sided")

        Durbin-Watson test

data:  RegLineal
DW = 0.63125, p-value = 8.692e-11
alternative hypothesis: true autocorrelation is not 0
```

Al ser el p-value menor a 5%, entonces no podemos afirmar que no existe correlación entre los residuos.

Al fallar esta validación, debemos ser cuidadosos en el resultado que se obtenga del modelo de regresión lineal, debido a que una de sus pruebas ha fallado.

g) Validación #5 - Modelo de Regresión Lineal

Validamos que los residuos deben seguir una distribución Normal.

Para ello debemos validar la siguiente Prueba de Hipótesis:

Ho: Los errores o residuos siguen una distribución normal

H1: Los errores o residuos NO siguen una distribución normal

Al ejecutar la prueba con el instrucción: **shapiro**.test, obtenemos:

```
> residuos <- residuals(RegLineal)
> shapiro.test(residuos)

        Shapiro-Wilk normality test

data:  residuos
W = 0.84524, p-value = 1.378e-06
```

Al ser el p-value menor a 5% se acepta Ho, por lo tanto damos como aceptada que los residuos siguen una distribución normal.

Hasta este punto, 4 de 5 validaciones fueron exitosas y por ende se continuará con el análisis.

h) Obtener la fórmula del Modelo

```
> print(RegLineal)

Call:
lm(formula = rows[, 2] ~ rows[, 1])

Coefficients:
(Intercept)     rows[, 1]
    161.686         7.036
```

Por lo tanto sería: **Y (Espacio GB) = 161.686 + 7.036 * X (# de Mes)**.

**La fórmula es basada en una muestra, pero necesitamos calcular un Intervalo de Confianza para estimar la Población** que será el valor que necesitamos para estimar el espacio requerido a un tiempo futuro.

Esto lo realizamos con la instrucción **predict**.

Por ejemplo: Si queremos estimar el tamaño de la base de datos dentro de 1 año con un nivel de confianza del 99% sería:

El último mes registrado fue el 63, como deseamos a 1 año, entonces sería: 63 + 12 = 75

Por lo tanto:

```
> resultado <- predict(RegLineal, data.frame(x=75), interval="prediction", level=0.99)
> tail(resultado)
       fit      lwr      upr
1 689.4195 561.3409 817.498
```

Con un 99% de nivel de confianza tenemos evidencia que la base de datos ocupará 817.498 GB en los próximos 12 meses si es que no viene a futuro nuevos proyectos que cambien el comportamiento.

Asimismo recomiendo tener presente 2 puntos:

- Siempre considerar el valor del límite superior del intervalo de confianza para estar en el peor escenario.
- Al valor obtenido debemos aumentarle entre 10% a 15%. La razón de este incremento es porque siempre hemos tomado capacidad de utilización y siempre deberíamos mantener la base de datos con un margen libre para que nuestros monitoreos no estén alertados.

Lo interesante de Oracle R Distribution es que podemos almacenar nuestro script y ejecutarlo cuando deseemos sin tener que estar cambiando alguna configuración.

A continuación copio todas las instrucciones de Oracle R que ha sido utilizado durante esta implementación:

```
#Conexión con la Base de Datos.
library('ROracle')
drv <- dbDriver("Oracle")
con <- dbConnect(drv,"friccio","oracle",dbname='132.68.1.30:1521/PRD')
#Obtención de los datos.
sql <- "select * from VREPORTE_ESPACIO"
rows <- dbGetQuery(con,sql)
print(rows)
dbDisconnect(con)

#Gráfico Lineal de los datos obtenidos.
Tamano_GB <- rows[,2]
plot(Tamano_GB, type="o", col="blue")
title(main="Tamaño de la Base de Datos", col.main="red", font.main=4)

#Calculo del Modelo de Regresión Lineal
x <- rows[,1]
y <- rows[,2]
RegLineal <- lm(formula=y~x)
summary(RegLineal)
#Revisar (pvalue de la variable y de la estadística F, ambos deben estar < 5%)
#        (R cuadrado ajustado)

#Prueba de Durbin-Watson, el pvalue debe > 5%
library(lmtest)
dwtest(RegLineal, alternative="two.sided")

#Prueba de Kolmogorov-Smirnov, el pvalue debe < 5%
residuos <- residuals(RegLineal)
shapiro.test(residuos)

#Proyectando el mes #75 con un intervalo de confianza del 99%
resultado <- predict(RegLineal, data.frame(x=75), interval="prediction", level=0.99)
tail(resultado)
```

#Conexión con la Base de Datos.

library('ROracle')

drv <- dbDriver("Oracle")

con <- dbConnect(drv,"friccio","oracle",dbname='132.68.1.30:1521/PRD')

#Obtención de los datos.

sql <- "select * from VREPORTE_ESPACIO"

rows <- dbGetQuery(con,sql)

print(rows)

dbDisconnect(con)

```
#Gráfico Lineal de los datos obtenidos.
Tamano_GB <- rows[,2]
plot(Tamano_GB, type="o", col="blue")
title(main="Tamaño de la Base de Datos", col.main="red", font.main=4)

#Calculo del Modelo de Regresión Lineal
x <- rows[,1]
y <- rows[,2]
RegLineal <- lm(formula=y~x)
summary(RegLineal)
#Revisar (pvalue de la variable y de la estadística F, ambos deben estar < 5%)
#               (R cuadrado ajustado)

#Prueba de Durbin-Watson, el pvalue debe > 5%
library(lmtest)
dwtest(RegLineal, alternative="two.sided")

#Prueba de Kolmogorov-Smirnov, el pvalue debe < 5%
residuos <- residuals(RegLineal)
shapiro.test(residuos)

#Proyectando el mes #75 con un intervalo de confianza del 99%
resultado <- predict(RegLineal, data.frame(x=75), interval="prediction", level=0.99)
tail(resultado)
```

## Conclusión

Hemos podido comprobar que la estimación de Storage que requiere nuestra base de datos a un tiempo futuro lo podemos calcular con herramientas estadísticas como Oracle R Distribution, teniendo un fundamento más sólido en el análisis, ya que en muchas oportunidades realizamos nuestro cálculo vía una regla de tres o simplemente mencionamos un número sin mayor fundamento.

Oracle R provee todas las librerías estadísticas requeridas para cumplir con nuestro objetivo y gracias al paquete ROracle podemos extraer los datos de una base de datos Oracle de manera más simple y confiable. Oracle R también provee un paquete específico para trabajar con MySQL.

Realizando los puntos vistos en este material, nos permitirá ser más asertivos al momento de realizar el proceso de Management Capacity que al principio se mencionó.

Publicado por Ing. Francisco Riccio. Es un Cloud Architect en IBM Perú e instructor de cursos oficiales de certificación Oracle. Está reconocido por Oracle como un Oracle ACE y certificado en productos de Oracle Application & Base de Datos.

e-mail: franciscoriccio@gmail.com

web: www.friccio.com