

RLM SERVICIOS MENSUALES

Miguel Angel Villegas

2025-03-11

Introducción

Este modelo abarca todas las ventas de los servicios mensuales, se utilizan las variables: Totales, Valor Unitario y Cantidad. Se realizan las correlaciones entre las tres variables y se determina el modelo lineal múltiple. El conjunto de entrenamiento y prueba esta dividido en una proporción de 80/20, sin embargo, la división es aleatoria, lo hace que el resultado sea mas confiable. Se siembra una semilla para permitir que los valores de la muestra sean los mismos.

```
ruta <- "/cloud/project/Ventas_Suministros_Totales.xlsx"
excel_sheets(ruta)

## [1] "Ventas Totales Original"      "Servicios Totales Original"
# "Ventas Totales Original"      "Servicios Totales Original"

Servicios_Totales <- as.data.frame(read_xlsx(ruta,
                                             sheet = "Servicios Totales Original"))
Servicios_Totales$Semana <- format(Servicios_Totales$Fecha, format = "%Y-%U")
Servicios_Totales$mes <- format(Servicios_Totales$Fecha, format = "%Y-%m")

Servicios_Totales <- Servicios_Totales %>%
  group_by(Fecha = as.character(mes)) %>%
  summarize(Ventas_Totales = sum(Total),
            Ventas_Unitario = sum(ValorUnitario),
            Ventas_Cantidad = sum(Cantidad),
            .groups = "keep")
head(Servicios_Totales)

## # A tibble: 6 x 4
## # Groups:   Fecha [6]
##   Fecha   Ventas_Totales Ventas_Unitario Ventas_Cantidad
##   <chr>         <dbl>         <dbl>         <dbl>
## 1 2019-07         78827.         24048          78
## 2 2019-08         32980.         18771          33
## 3 2019-09         31320          8100          45
## 4 2019-10         24708         10500          29
## 5 2019-11         41522.         24095.          33
## 6 2019-12         73834         42370          53

Servicios_Totales <- as.data.frame(Servicios_Totales)
head(Servicios_Totales)

##   Fecha Ventas_Totales Ventas_Unitario Ventas_Cantidad
## 1 2019-07      78826.64      24048.00          78
## 2 2019-08      32979.96      18771.00          33
```

## 3	2019-09	31320.00	8100.00	45
## 4	2019-10	24708.00	10500.00	29
## 5	2019-11	41521.75	24094.61	33
## 6	2019-12	73834.00	42370.00	53

Se calcula el valor del parámetro lambda para la transformación Boxcox.

```
VS_lambda_mes <- boxcox(Servicios_Totales$Ventas_Totales,
                        objective.name = "Log-Likelihood", optimize = T)
# 0.07283614
VUs_lambda_mes <- boxcox(Servicios_Totales$Ventas_Unitario,
                        objective.name = "Log-Likelihood", optimize = T)
# 0.1786999
VCs_lambda_mes <- boxcox(Servicios_Totales$Ventas_Cantidad,
                        objective.name = "Log-Likelihood", optimize = T)
# -0.1807043
```

Se obtiene la transformación boxcox para los servicios, valor unitario de los servicios y cantidad de venta de los servicios, todos por semana.

```
RLM_serv_mes <- Servicios_Totales %>%
  mutate(
    Ventas_Totales = boxcoxTransform(Servicios_Totales$Ventas_Totales, lambda = 0.07283614),
    Ventas_Unitario = boxcoxTransform(Servicios_Totales$Ventas_Unitario, lambda = 0.1786999),
    Ventas_Cantidad = boxcoxTransform(Servicios_Totales$Ventas_Cantidad, lambda = -0.1807043)
  )
head(RLM_serv_mes)
```

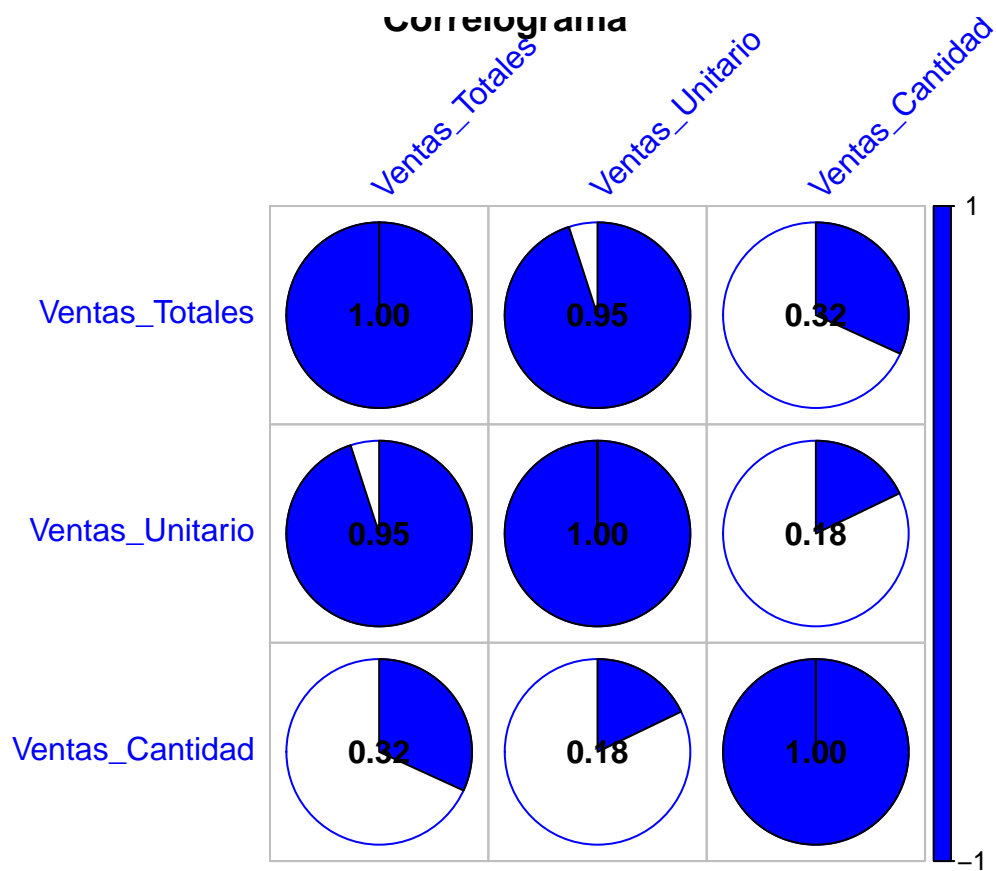
##	Fecha	Ventas_Totales	Ventas_Unitario	Ventas_Cantidad
## 1	2019-07	17.48156	28.34889	3.015519
## 2	2019-08	15.56228	26.87890	2.591992
## 3	2019-09	15.45231	22.35015	2.752340
## 4	2019-10	14.95262	23.67666	2.522492
## 5	2019-11	16.05780	28.36064	2.591992
## 6	2019-12	17.33317	31.96446	2.833382

Se crea una matriz para el cálculo y visualización de las correlaciones, además se siembra la semilla para garantizar que los valores sean los mismos.

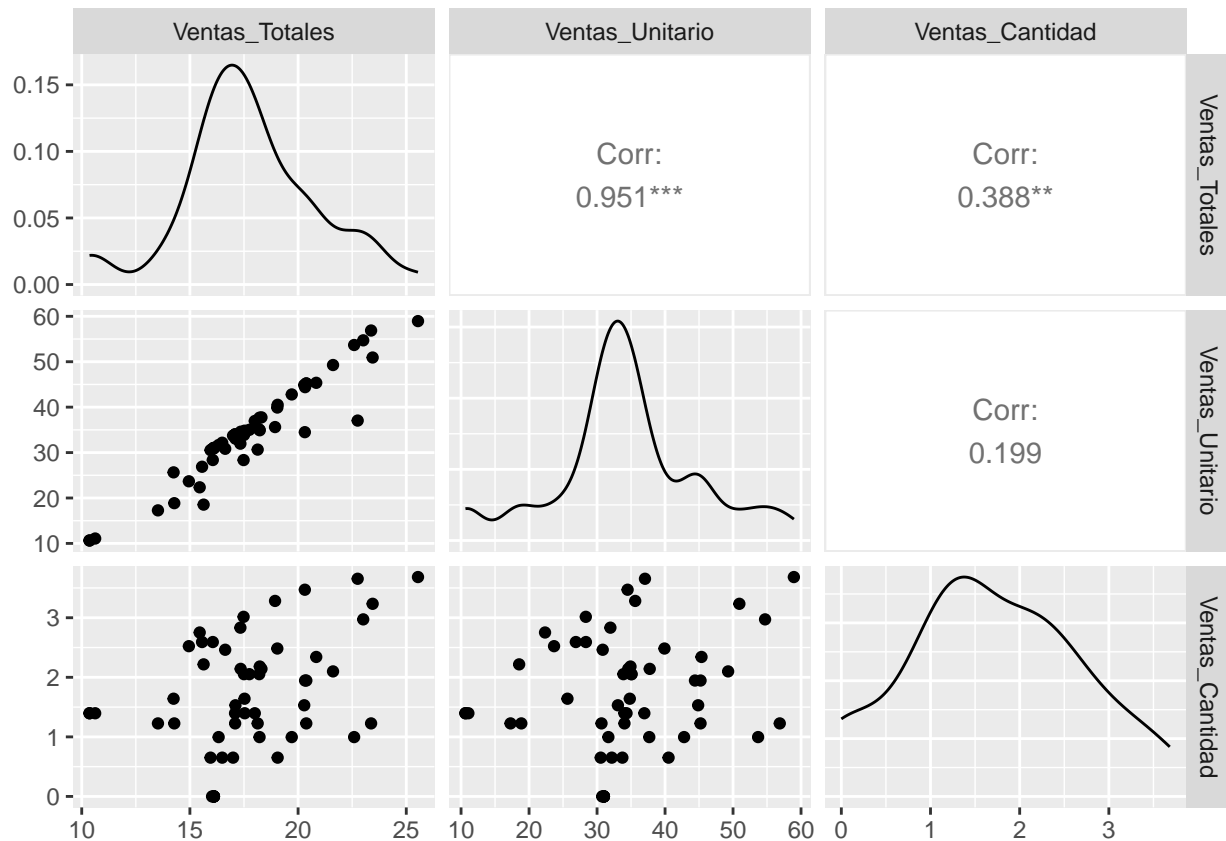
```
set.seed(12345)
RLM_serv_mes_mtx <- cbind(RLM_serv_mes$Ventas_Totales,
                          RLM_serv_mes$Ventas_Unitario,
                          RLM_serv_mes$Ventas_Cantidad)
colnames(RLM_serv_mes_mtx) <- c("Ventas_Totales", "Ventas_Unitario", "Ventas_Cantidad")
head(RLM_serv_mes_mtx)
```

##	Ventas_Totales	Ventas_Unitario	Ventas_Cantidad
## [1,]	17.48156	28.34889	3.015519
## [2,]	15.56228	26.87890	2.591992
## [3,]	15.45231	22.35015	2.752340
## [4,]	14.95262	23.67666	2.522492
## [5,]	16.05780	28.36064	2.591992
## [6,]	17.33317	31.96446	2.833382

Gráficas de correlación



```
RLM_serv_mes <- as.data.frame(RLM_serv_mes_mtx)
RLM_serv_mes %>% GGally::ggpairs(cardinality_threshold = 10)
```



Se hace la división del conjunto de datos en una proporción de 80-20.

```
ms_lm_mes <- sample.split(RLM_serv_mes$Ventas_Totales, SplitRatio = 0.80)
es_lm_mes <- subset(RLM_serv_mes, ms_lm_mes == T)
ps_lm_mes <- subset(RLM_serv_mes, ms_lm_mes == F)
```

Modelo

Se determina el modelo lineal.

```
Moslm_mes <- lm(Ventas_Totales ~ ., data = es_lm_mes)
```

Resumen del modelo obtenido.

```
print(summary(Moslm_mes))
```

```
##
## Call:
## lm(formula = Ventas_Totales ~ ., data = es_lm_mes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04129 -0.43496 -0.07642  0.32320  2.82749
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.57408    0.40160   18.86 < 2e-16 ***
## Ventas_Unitario 0.26168    0.01075   24.35 < 2e-16 ***
## Ventas_Cantidad 0.72726    0.12244    5.94 5.27e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7815 on 41 degrees of freedom
## Multiple R-squared:  0.9465, Adjusted R-squared:  0.9439
## F-statistic: 362.7 on 2 and 41 DF,  p-value: < 2.2e-16
# Multiple R-squared:  0.9465, Adjusted R-squared:  0.9439
```

Pronóstico

```
pronostico_lms_mes <- predict(Moslm_mes, ps_lm_mes)
```

Se crea una data frame con los resultados y los valores actuales.

```
resul_s_mes <- cbind(pronostico_lms_mes, ps_lm_mes$Ventas_Totales)
resul_s_mes <- as.data.frame(resul_s_mes)
colnames(resul_s_mes) <- c("prediccion", "actual")
head(resul_s_mes)
```

```
##      prediccion  actual
## 2      16.49267 15.56228
## 4      15.60418 14.95262
## 10     15.47848 14.24546
## 20     19.82525 19.04060
## 23     20.60867 20.32337
## 31     16.87308 16.99450
```

Si es hay valores menores que cero se substituyen por cero.

Función.

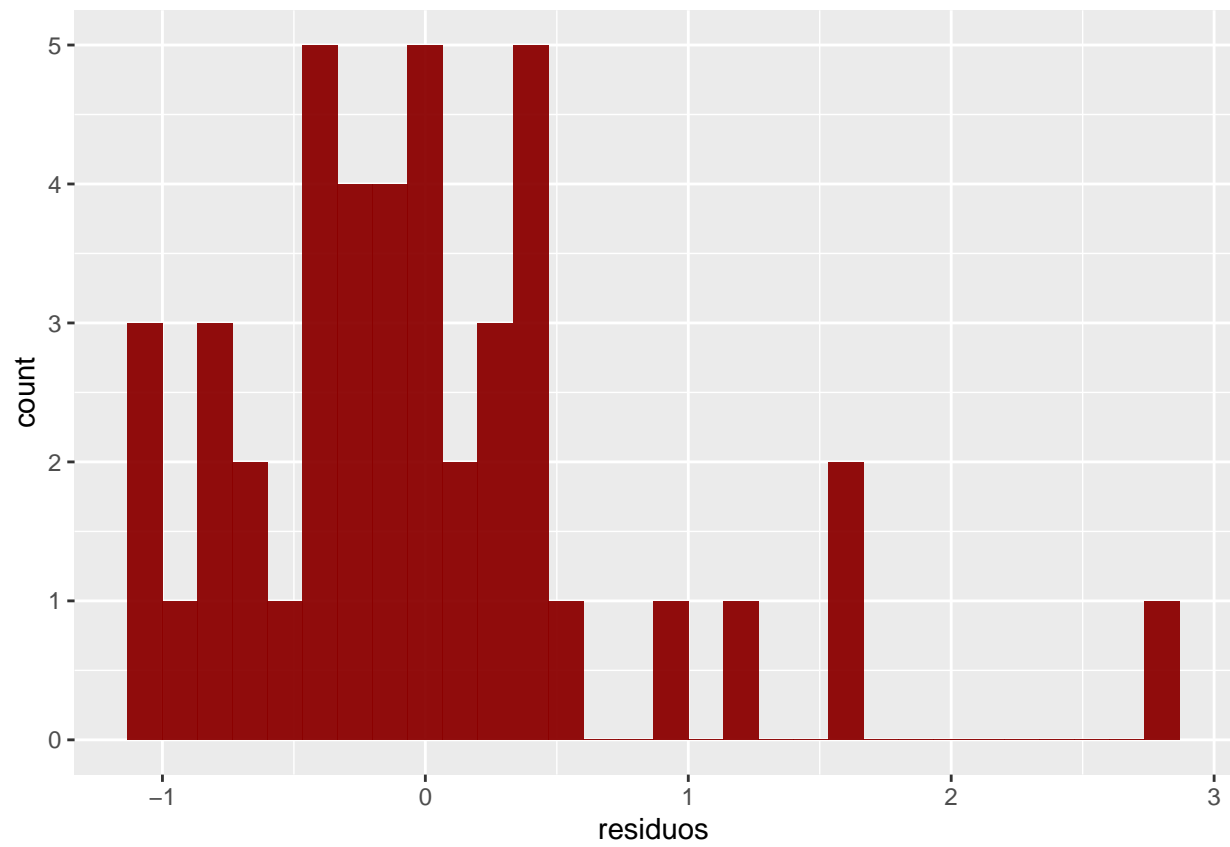
Exactitud del modelo.

```
summary(Moslm_mes)$r.squared
```

```
## [1] 0.9465006
```

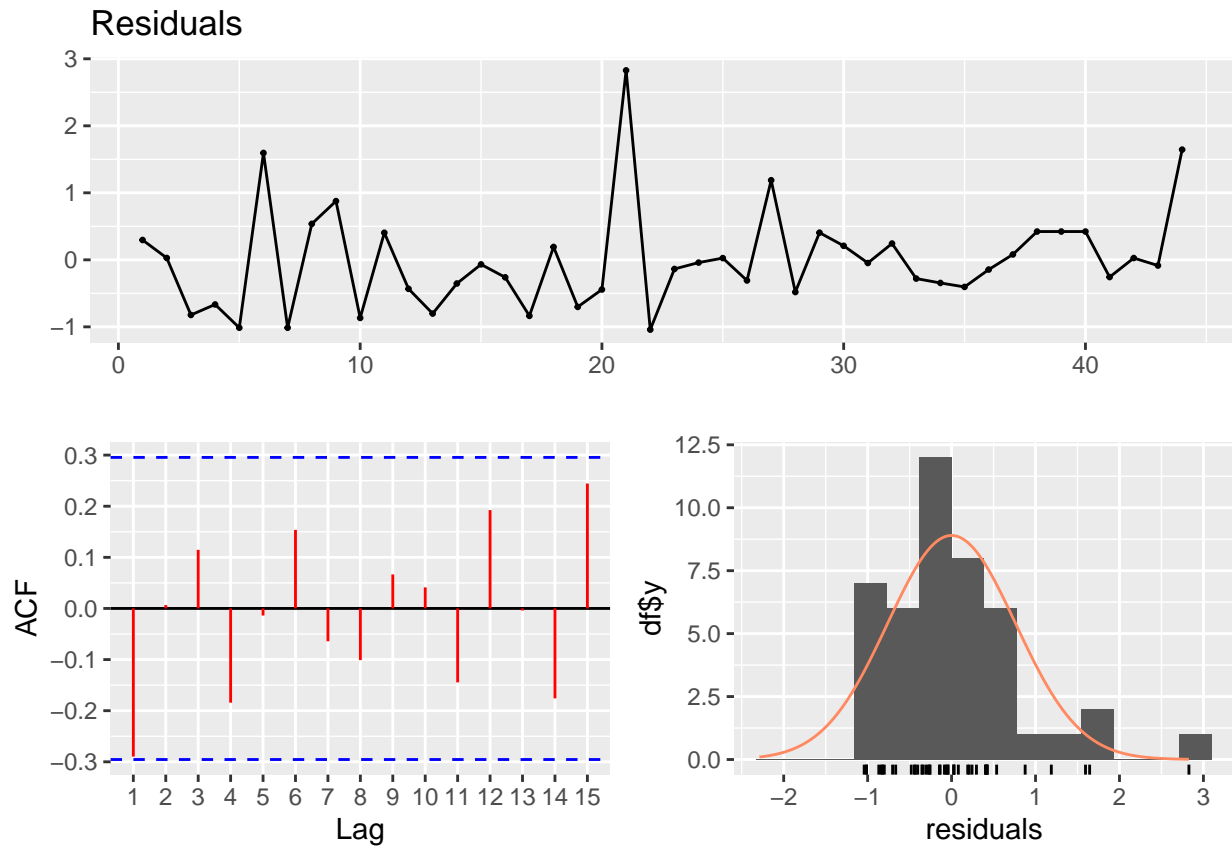
```
# [1] 0.9465006
```

Inspección de los residuales.



Residuales.

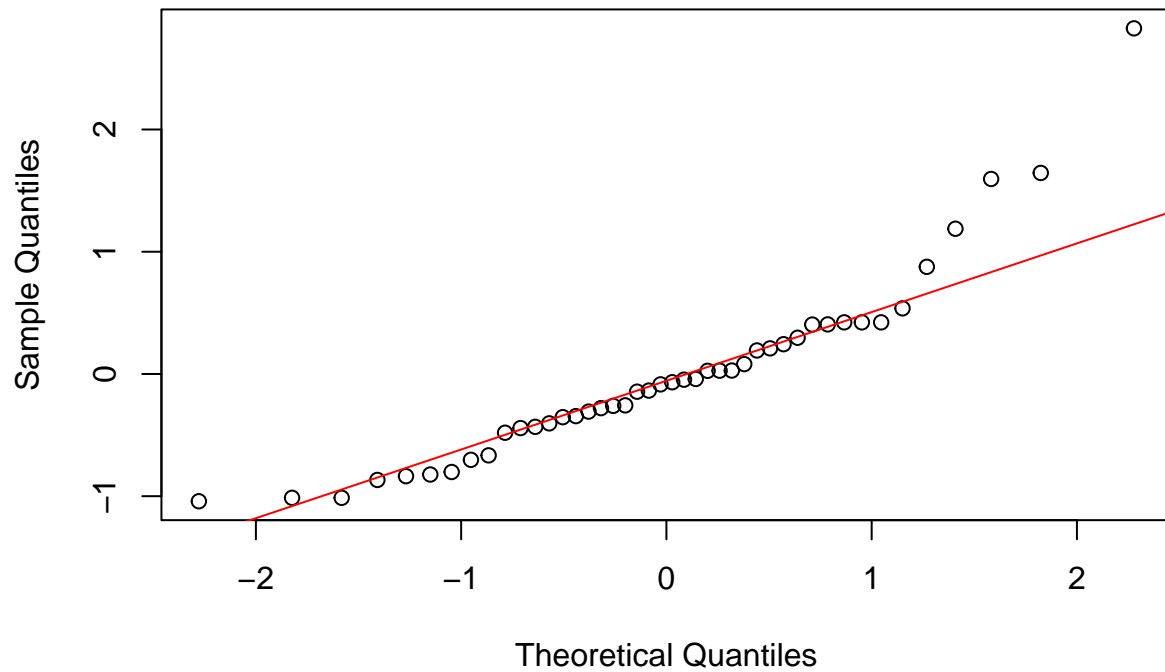
```
checkresiduals(Moslm_mes, col = "red")
```



```
##
## Breusch-Godfrey test for serial correlation of order up to 9
##
## data: Residuals
## LM test = 9.1138, df = 9, p-value = 0.4268
# p-value = 0.4268
```

Inspeccionando si existe normalidad en los residuales.

Normal Q-Q Plot

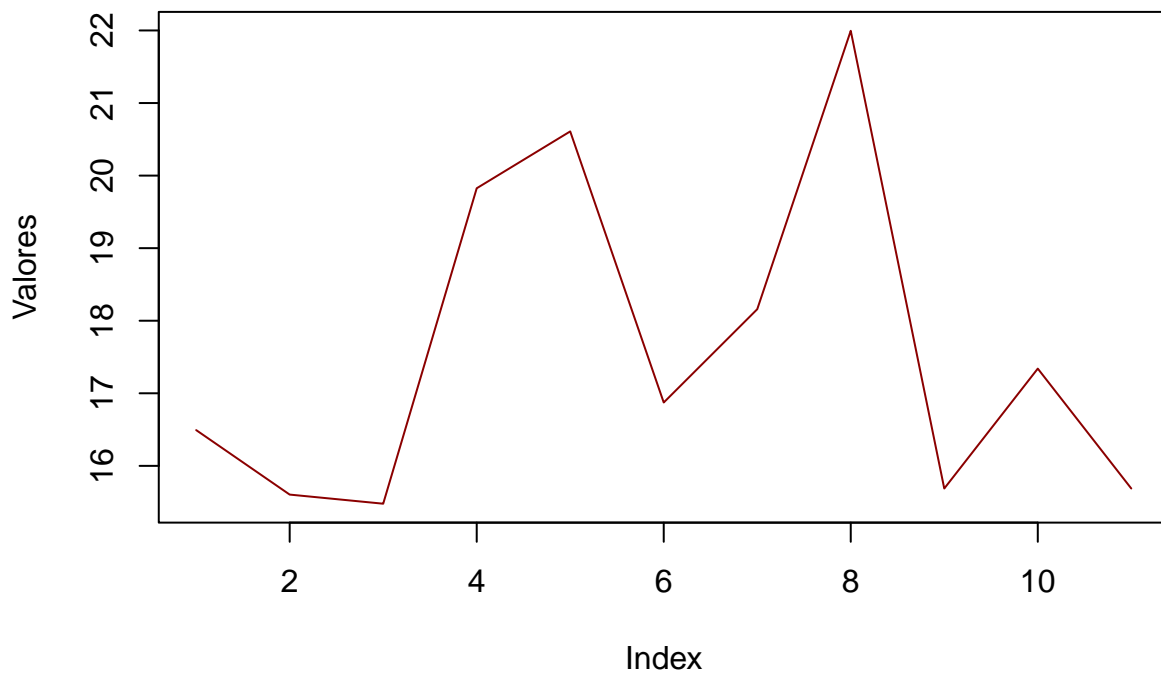


Se grafican los pronósticos.

```
pronostico_serv_lm_mes <- predict(Moslm_mes, ps_lm_mes)
```

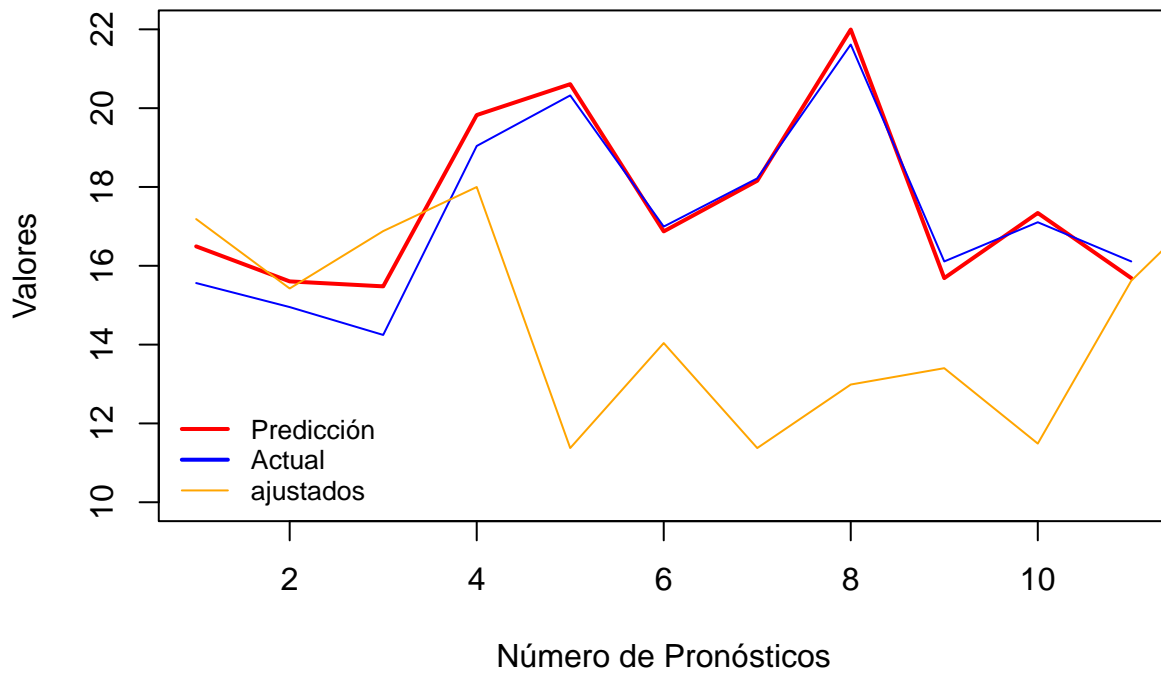
```
plot(pronostico_serv_lm_mes, type = "lines", col = "darkred", ylab = "Valores")
```

```
## Warning in plot.xy(xy, type, ...): plot type 'lines' will be truncated to first  
## character
```

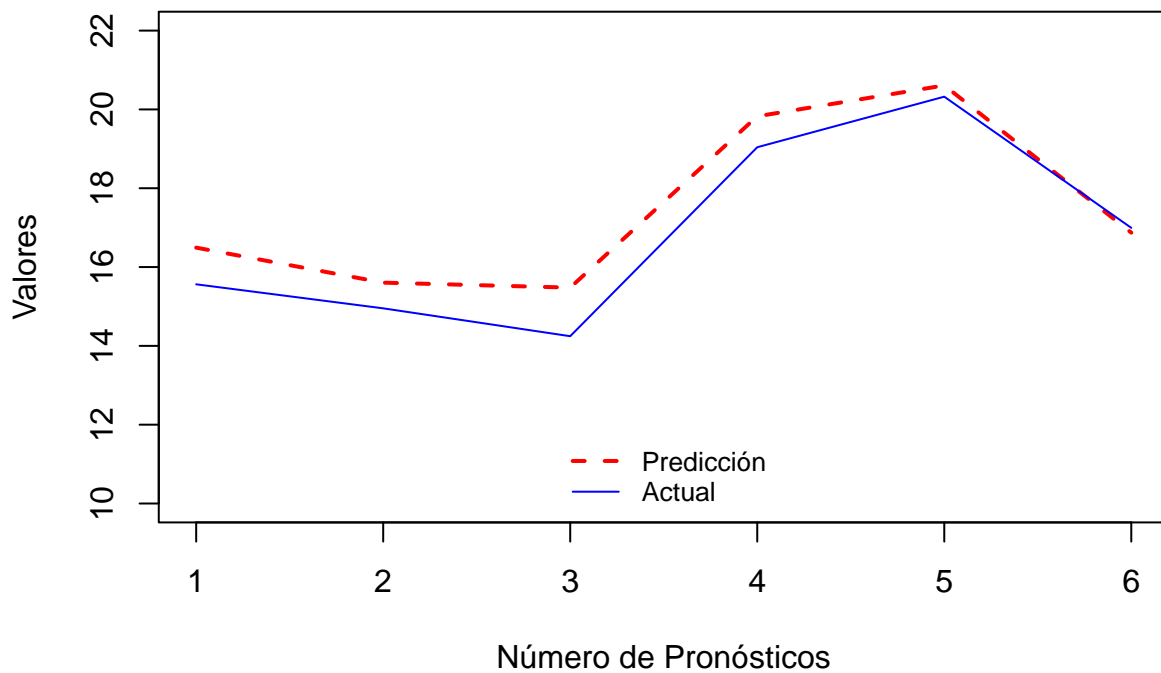


Gráfica de los pronósticos junto con los valores reales.

Predicción vs Actual por Regresión Multi-Lineal



Predicción vs Actual por Regresión Multi-Lineal



Se determina la exactitud del modelo.

```
accuracy(resul_s_mes$prediccion, ps_lm_mes$Ventas_Totales)
```

##	ME	RMSE	MAE	MPE	MAPE
## Test set	-0.3158728	0.6092055	0.5023596	-1.942315	3.084989

#	<i>ME</i>	<i>RMSE</i>	<i>MAE</i>	<i>MPE</i>	<i>MAPE</i>
# Test set	-0.6272476	0.7661916	0.6677216	-3.966952	4.205111

Conclusiones

El modelo captura 94.65% la dinámica de la serie, los residuales están muy por encima del valor ideal de $p > 0.05$, lo que indica que no existe una correlación entre los residuos.