

# RLM SERVICIOS SEMANALES

Miguel Angel Villegas

2025-03-11

## Introducción

Este modelo abarca todas las ventas de los servicios semanales, 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(Semana)) %>%
  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 2019-26         31552             5000             48
## 2 2019-27          2494             2150              2
## 3 2019-28          8839             7620              1
## 4 2019-29         11484              450             22
## 5 2019-30        24457             8828              5
## 6 2019-31        7750             5121              7

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

##   Fecha Ventas_Totales Ventas_Unitario Ventas_Cantidad
## 1 2019-26      31552.00          5000             48
## 2 2019-27       2494.00          2150              2
```

## 3	2019-28	8839.20	7620	1
## 4	2019-29	11484.00	450	22
## 5	2019-30	24457.44	8828	5
## 6	2019-31	7749.96	5121	7

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

```
VS_lambda_sem <- boxcox(Servicios_Totales$Ventas_Totales,
                        objective.name = "Log-Likelihood", optimize = T)
# 0.02906853
VUs_lambda_sem <- boxcox(Servicios_Totales$Ventas_Unitario,
                        objective.name = "Log-Likelihood", optimize = T)
# 0.1123158
VCs_lambda_sem <- boxcox(Servicios_Totales$Ventas_Cantidad,
                        objective.name = "Log-Likelihood", optimize = T)
# -0.4499683
```

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_sem <- Servicios_Totales %>%
  mutate(
    Ventas_Totales = boxcoxTransform(Servicios_Totales$Ventas_Totales, lambda = 0.02906853),
    Ventas_Unitario = boxcoxTransform(Servicios_Totales$Ventas_Unitario, lambda = 0.1123158),
    Ventas_Cantidad = boxcoxTransform(Servicios_Totales$Ventas_Cantidad, lambda = -0.4499683)
  )
head(RLM_serv_sem)
```

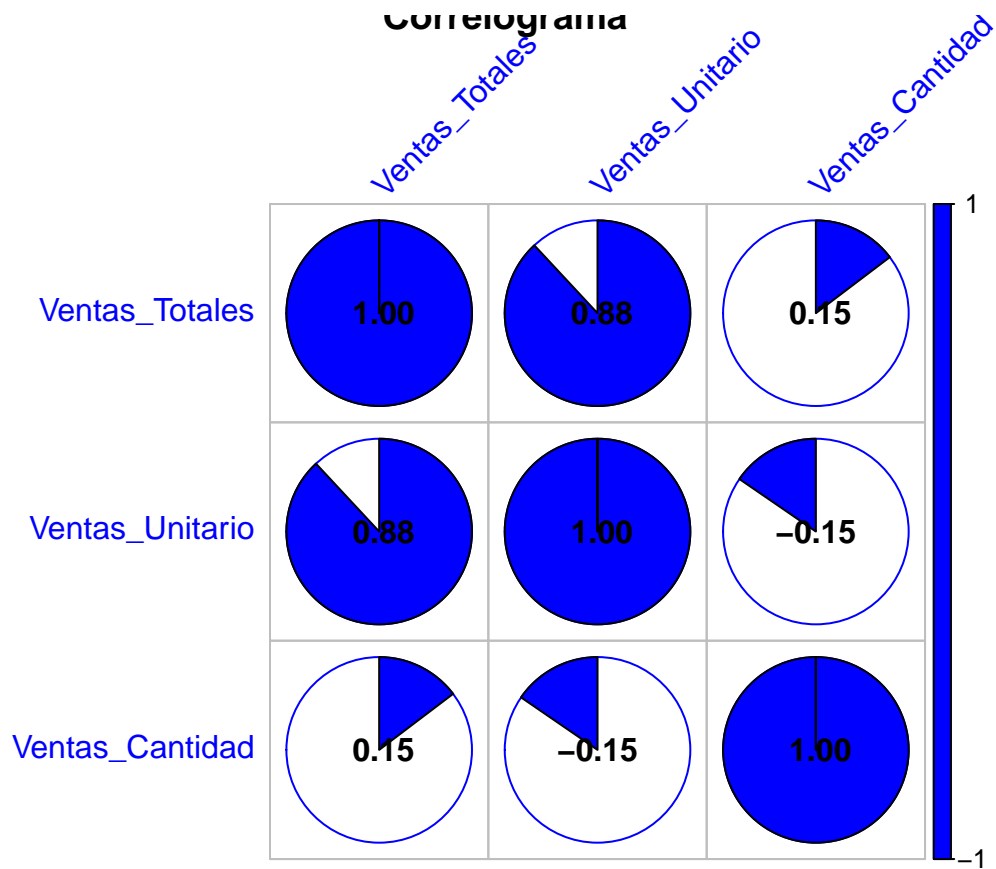
##	Fecha	Ventas_Totales	Ventas_Unitario	Ventas_Cantidad
## 1	2019-26	12.088266	14.271116	1.8330533
## 2	2019-27	8.782222	12.175273	0.5954665
## 3	2019-28	10.400118	15.394169	0.0000000
## 4	2019-29	10.742310	8.779569	1.6693211
## 5	2019-30	11.745335	15.799086	1.1451604
## 6	2019-31	10.229179	14.333439	1.2965072

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.

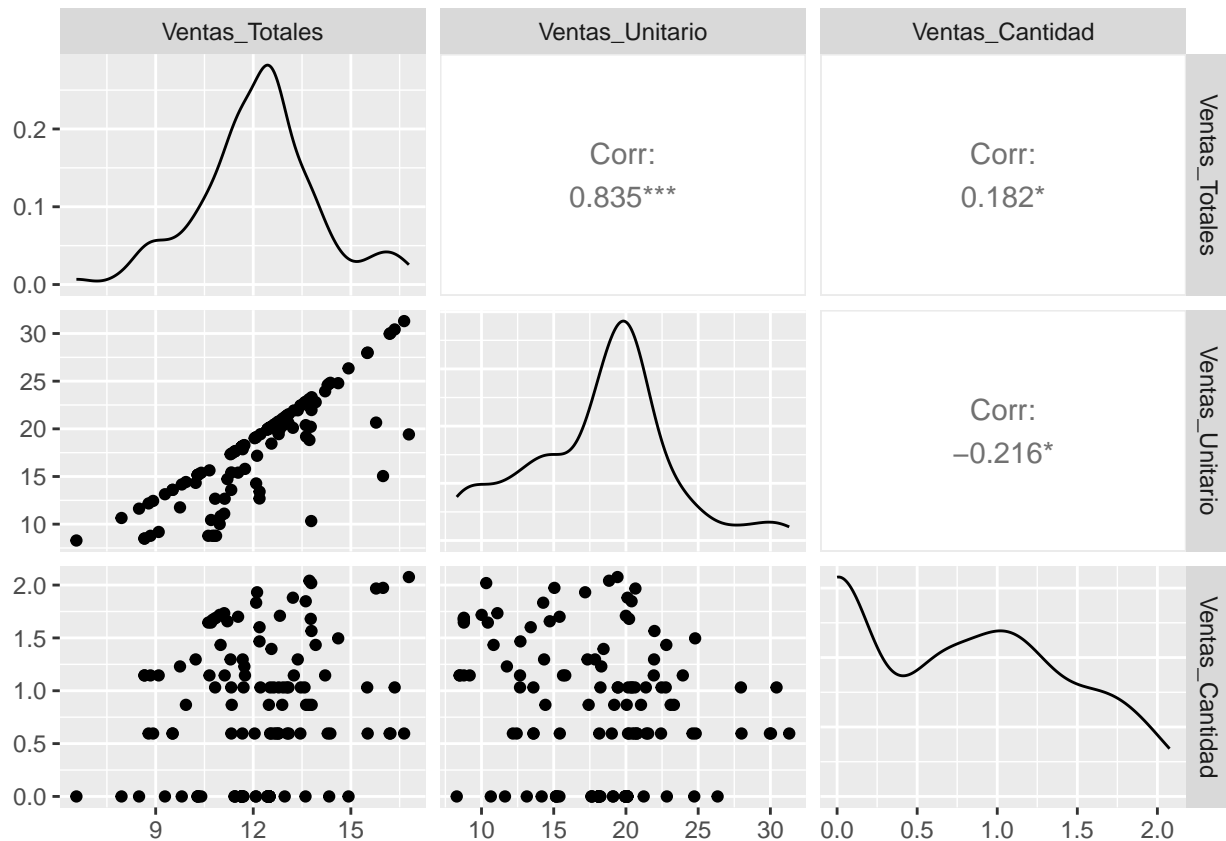
```
set.seed(1234)
RLM_serv_sem_mtx <- cbind(RLM_serv_sem$Ventas_Totales,
                        RLM_serv_sem$Ventas_Unitario,
                        RLM_serv_sem$Ventas_Cantidad)
colnames(RLM_serv_sem_mtx) <- c("Ventas_Totales", "Ventas_Unitario", "Ventas_Cantidad")
head(RLM_serv_sem_mtx)
```

##	Ventas_Totales	Ventas_Unitario	Ventas_Cantidad
## [1,]	12.088266	14.271116	1.8330533
## [2,]	8.782222	12.175273	0.5954665
## [3,]	10.400118	15.394169	0.0000000
## [4,]	10.742310	8.779569	1.6693211
## [5,]	11.745335	15.799086	1.1451604
## [6,]	10.229179	14.333439	1.2965072

## Gráficas de correlación



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



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

```
ms_lm_sem <- sample.split(RLM_serv_sem$Ventas_Totales, SplitRatio = 0.80)
es_lm_sem <- subset(RLM_serv_sem, ms_lm_sem == T)
ps_lm_sem <- subset(RLM_serv_sem, ms_lm_sem == F)
```

## Modelo

Se determina el modelo lineal.

```
Moslm_sem <- lm(Ventas_Totales ~ ., data = es_lm_sem)
```

Resumen del modelo obtenido.

```
print(summary(Moslm_sem))
```

```
##
## Call:
## lm(formula = Ventas_Totales ~ ., data = es_lm_sem)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3989 -0.5883 -0.0301  0.4679  3.5712
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.11285    0.33620  15.208 < 2e-16 ***
## Ventas_Unitario 0.34373    0.01591  21.609 < 2e-16 ***
## Ventas_Cantidad 1.08099    0.12223   8.844 3.13e-14 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8167 on 101 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8234
## F-statistic: 241.1 on 2 and 101 DF,  p-value: < 2.2e-16
# Multiple R-squared:  0.8496, Adjusted R-squared:  0.8466
```

## Pronóstico

```
pronostico_lms_sem <- predict(Moslm_sem, ps_lm_sem)
```

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

```
resul_s_sem <- cbind(pronostico_lms_sem, ps_lm_sem$Ventas_Totales)
resul_s_sem <- as.data.frame(resul_s_sem)
colnames(resul_s_sem) <- c("prediccion", "actual")
head(resul_s_sem)
```

```
##      prediccion      actual
## 5    11.781353  11.745335
## 14     9.109233   8.488435
## 16    10.803264  11.106287
## 26    10.900455  11.322394
## 28    11.972233  12.440167
## 29    11.972233  12.440167
```

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

Función.

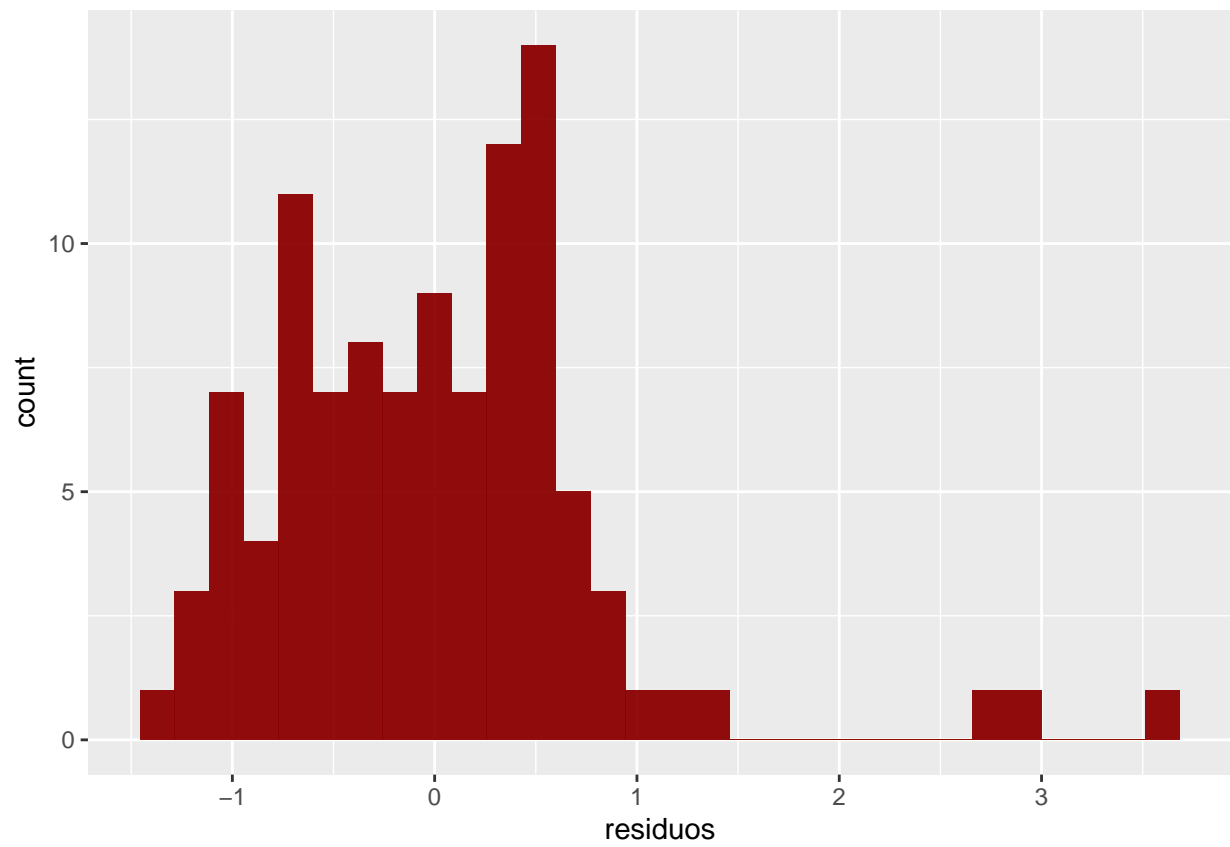
Exactitud del modelo.

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

```
## [1] 0.8268127
```

```
# [1] 0.8268127
```

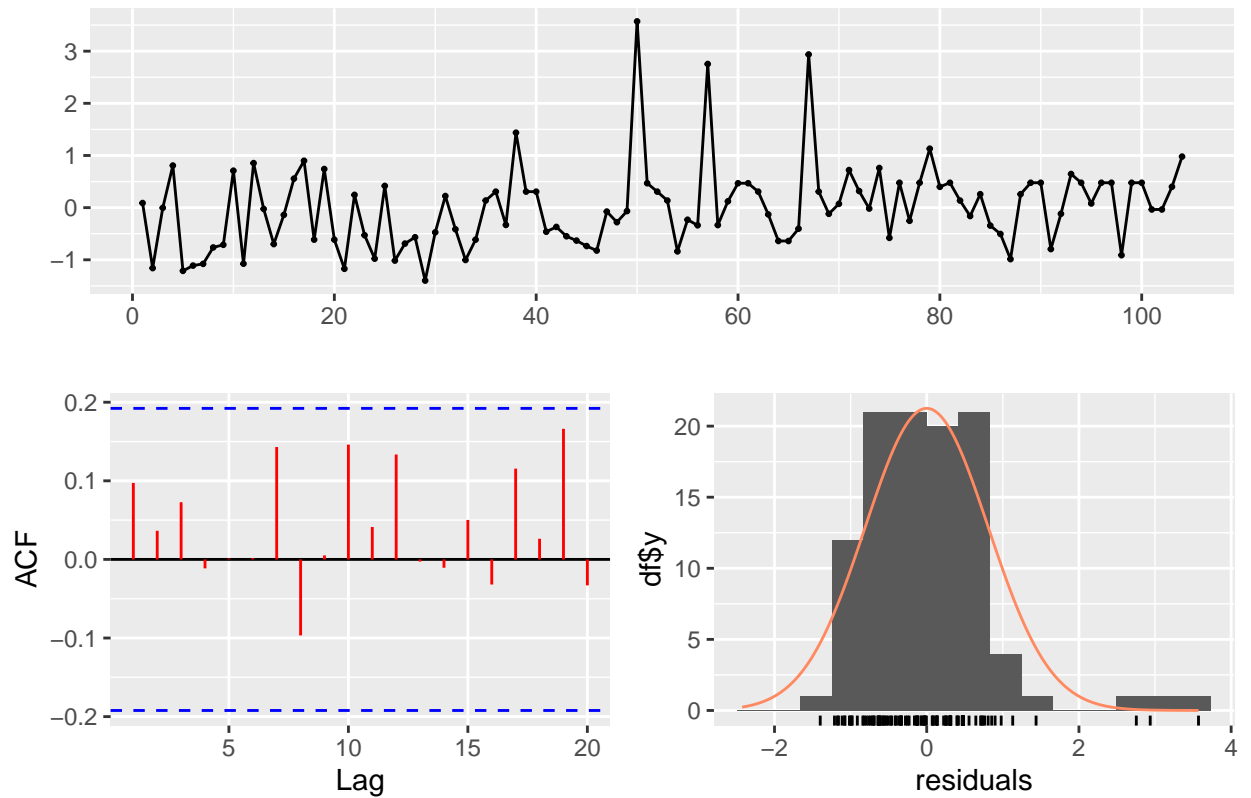
Inspección de los residuales.



Residuales.

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

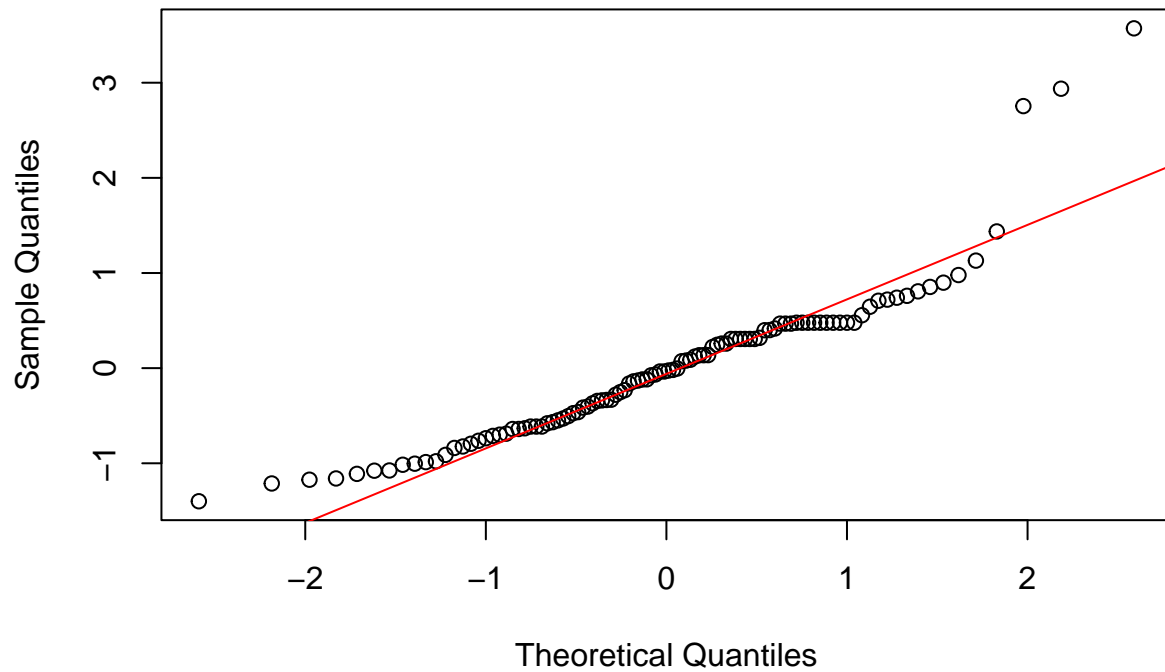
## Residuals



```
##
## Breusch-Godfrey test for serial correlation of order up to 10
##
## data: Residuals
## LM test = 8.1254, df = 10, p-value = 0.6166
# p-value = 0.6166
```

Inspeccionando si existe normalidad en los residuales.

## Normal Q-Q Plot

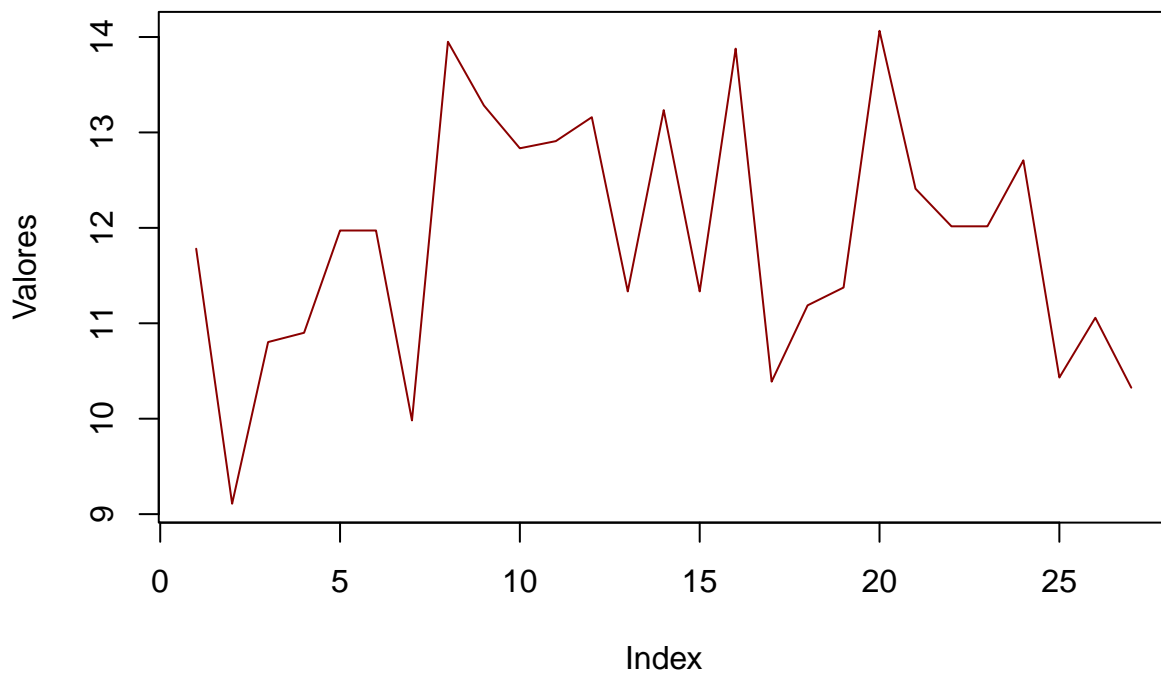


Se grafican los pronósticos.

```
pronostico_serv_lm_sem <- predict(Moslm_sem, ps_lm_sem)
```

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

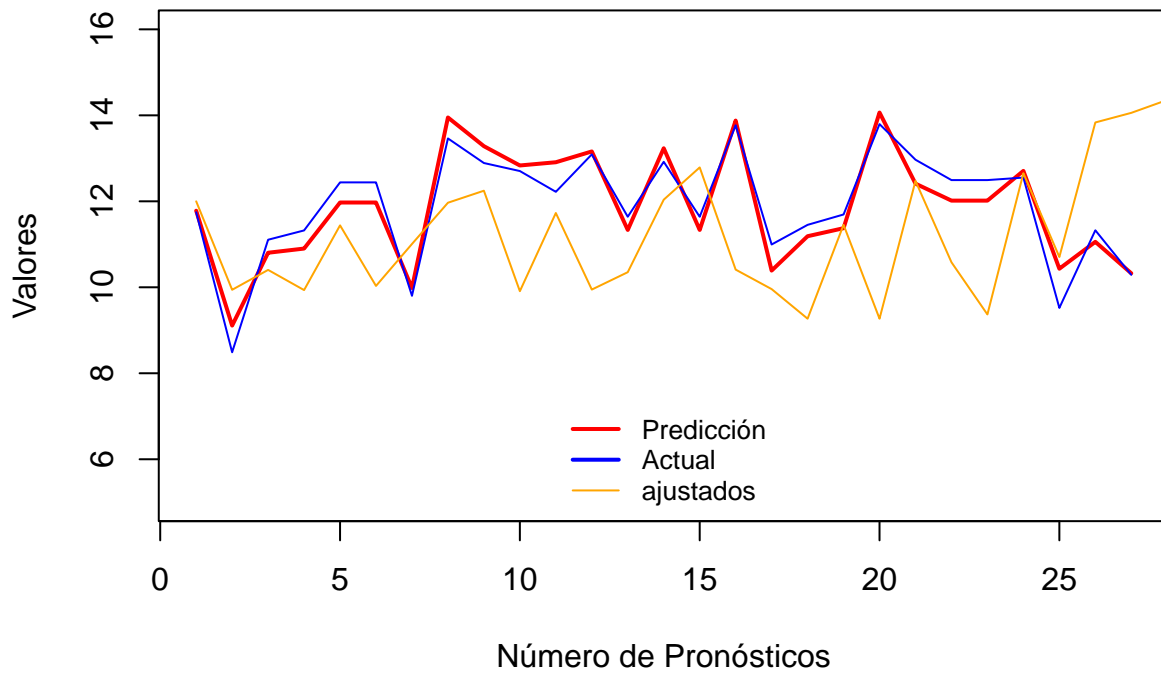
```
## Warning in plot.xy(xy, type, ...): plot type 'lines' will be truncated to first  
## character
```



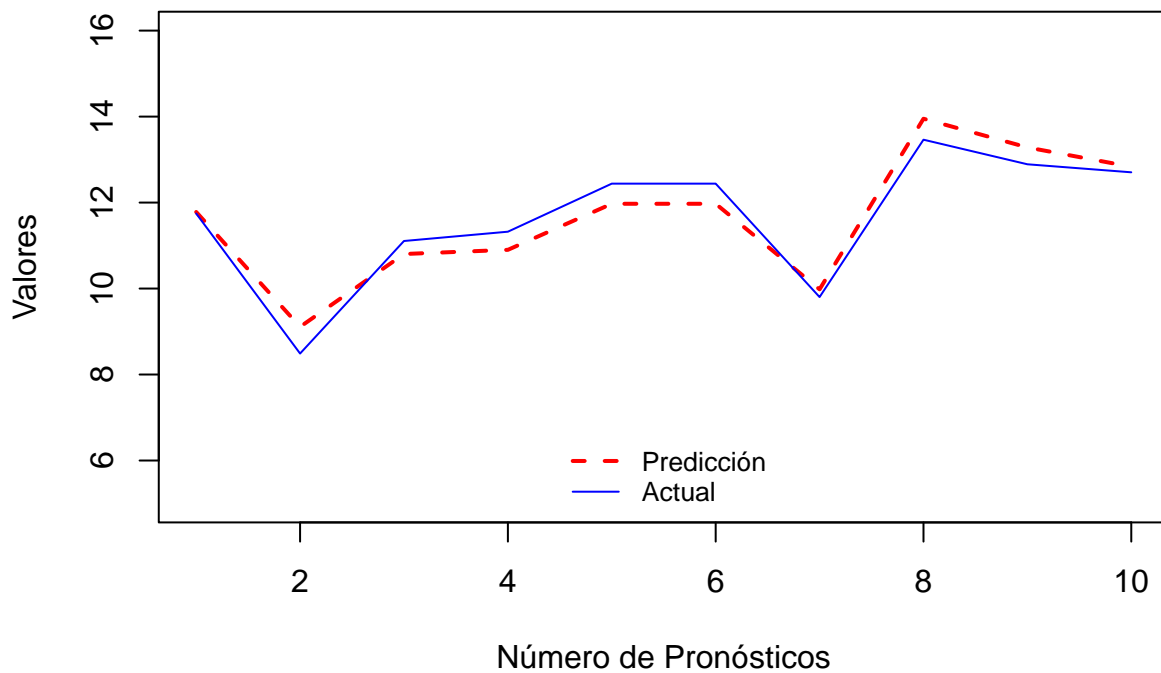


Gráfica de los pronósticos junto con los valores reales.

### Predicción vs Actual por Regresión Multi-Lineal



### Predicción vs Actual por Regresión Multi-Lineal



Se determina la exactitud del modelo.

```
accuracy(resul_s_sem$prediccion, ps_lm_sem$Ventas_Totales)
```

##	ME	RMSE	MAE	MPE	MAPE
## Test set	0.03127016	0.4136471	0.3571761	0.1652995	3.101769

#	ME	RMSE	MAE	MPE	MAPE
# Test set	0.03127016	0.4136471	0.3571761	0.1652995	3.101769

## Conclusiones

El modelo captura 84.96% 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.