Actividad 1. Modelamiento de Demanda 2

nclivio

2014

Introducción

A continuación se detalla la comparación de modelos para pronosticar la demanda empleando modelos de curvas de crecimiento, modelo Bass y el modelo Fisher-Pry.

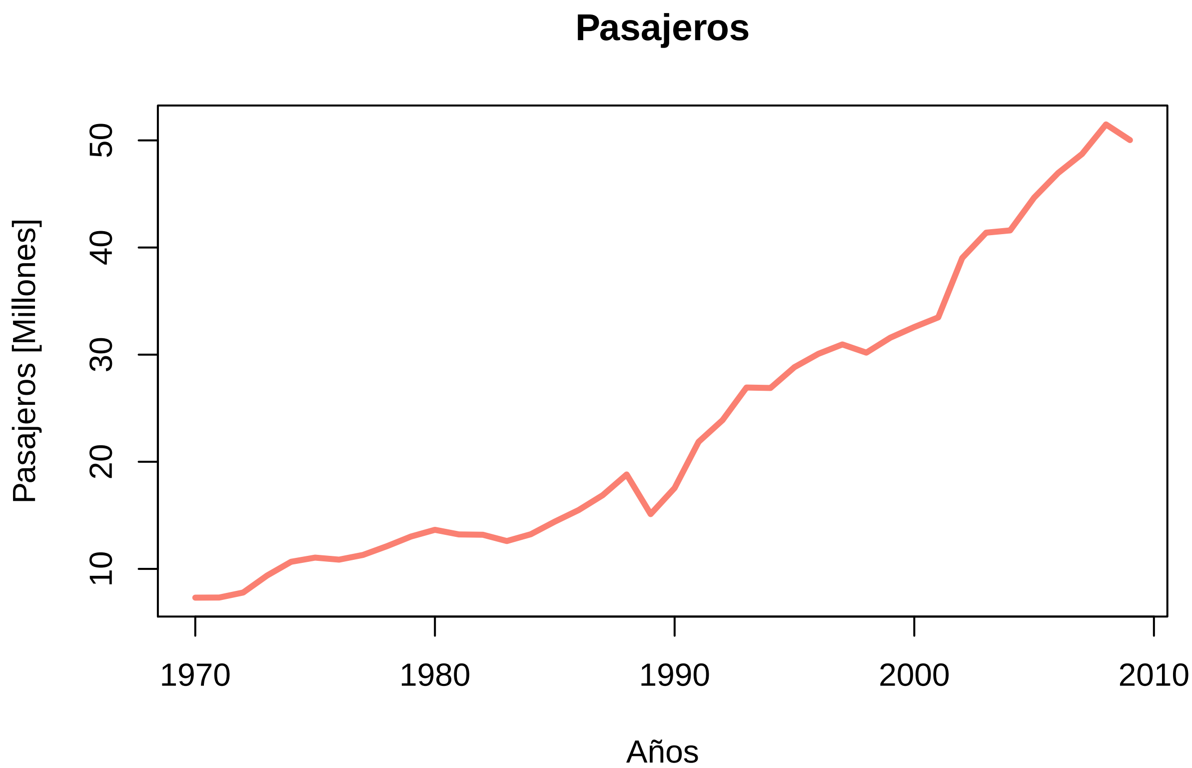
Descripción de los Datos

Se eligieron datos provenientes del paquete **fpp**, del libro Forecasting: Principles and Practice por Rob J Hyndman and George Athanasopoulos. Se trabajo con un dataset con un formato de series de tiempo anual ya que coincide con el formato de la demanda de usuarios requerido para el desarrollo de la tesis.

Preparación de Datos

Es necesario cargar el paquete **fpp**, se trabajará con el data set **ausair** Air Transport Passengers Australia, que contiene el total de pasajeros los años comprendidos entre 1970 y 2009.

```
library(fpp)
data(ausair)

plot(ausair, xlab="Años",ylab="Pasajeros [Millones]",col="salmon",lwd=3,
     main="Pasajeros")
```

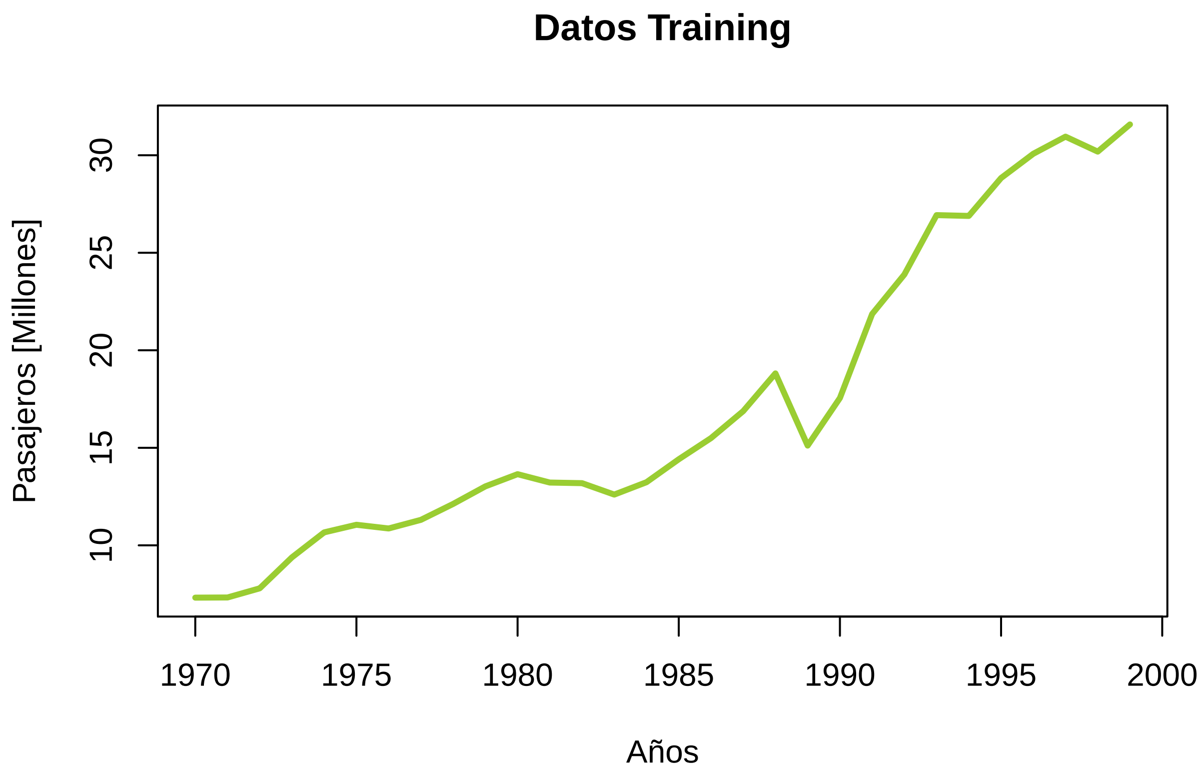


Se submuestrean los datos en dos grupos *training* para aplicar el modelo y *testing* para hacer las predicciones.

```
library(caret)
library(ggplot2)

ts<-ausair

tstrain<-window(ts,start=1970,end=1999)
tstest<-window(ts,start=1999,end=(2009))

years<-c(1970:1999)
datatrain<-data.frame(Años=years,Pasajeros=tstrain)

years2<-c(1999:2009)
datatest<-data.frame(Años=years2,Pasajeros=tstest)

#inTrain<-createDataPartition(y=data$Años, p=0.75, list=FALSE)
#training<-data[inTrain,]
#testing<-data[-inTrain,]
```

El conjunto de datos *training*, se muestra a continuación:

```
plot(tstrain, xlab="Años",ylab="Pasajeros [Millones]",col="yellowgreen",lwd=3,main="Datos Training")
```



Modelos y Predicciones

```
library(grofit)
library (forecast)
```

1.Modelos Lineales

A continuación se aplica, el modelo de regresión lineal, donde la variable independiente es el tiempo (Años) y la dependiente la cantidad de pasajeros.

```
fitlm<-tslm(tstrain~trend)

fitlm
```

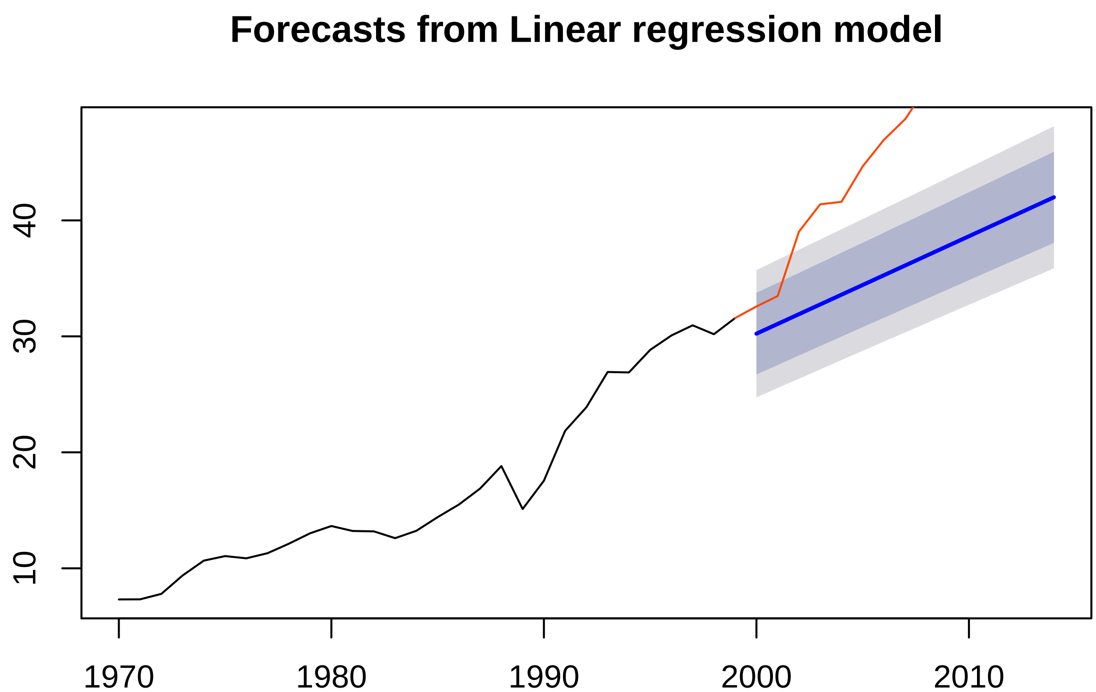
```
##
## Call:
## lm(formula = formula, data = "tstrain", na.action = na.exclude)
##
## Coefficients:
## (Intercept)        trend
##        4.18         0.84
```

Se hacen las predicciones

```r
predlm<-forecast(fitlm, h=15)
```

Graficamente la predicción del modelo lineal, se ve asi:

```r
plot(predlm)
lines(tstest,lwd=1,col="orangered")
```



2.Modelos Parabólico

Aplicando un modelo con tendencia no lineal, de forma polinómica de segundo grado de la forma:

y = b0 + b1 x + b2 x^2

Donde: b0 : Intersección estimada con el eje y b1 : efecto lineal estimado sobre y b2 : efecto curvilíneo estimado sobre y

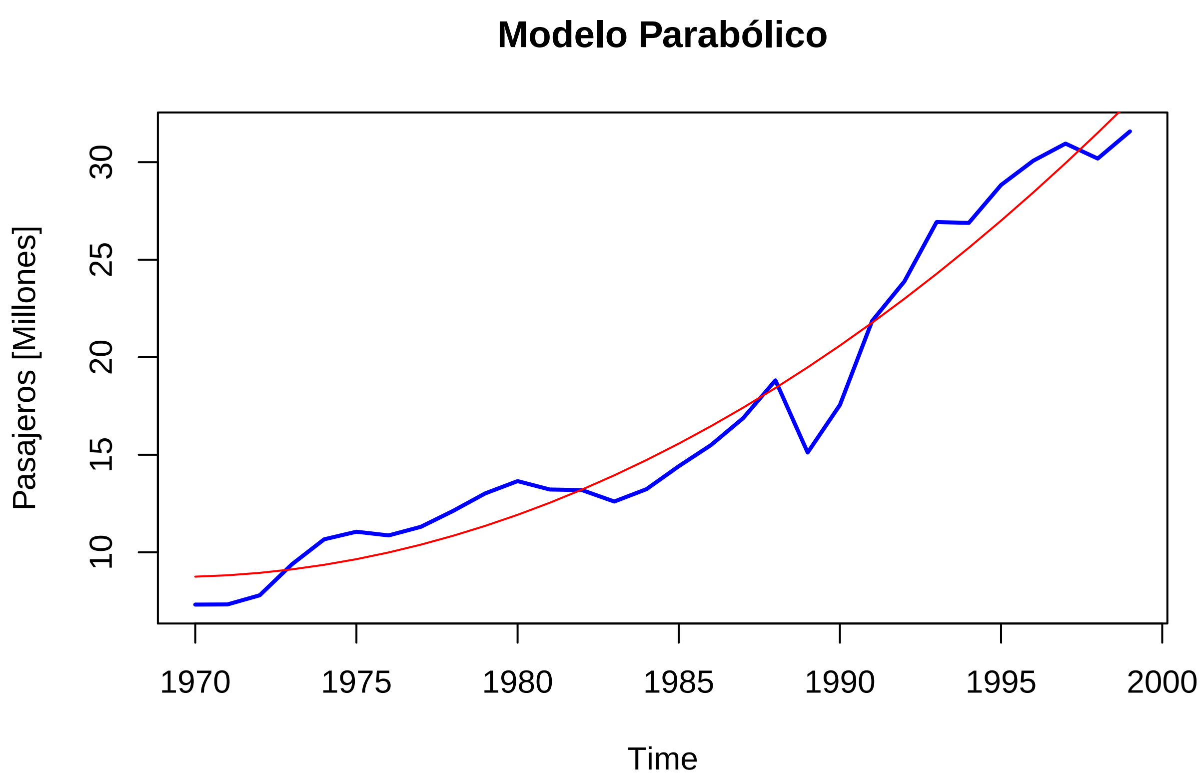
Se obtiene:

```r
time<-1:30
regpol = lm(tstrain ~ time + I(time^2))

summary(regpol)
```

```
##
## Call:
```

```
## lm(formula = tstrain ~ time + I(time^2))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.368 -1.280  0.328  1.277  2.653
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.73707    0.97411    8.97  1.4e-09 ***
## time        -0.01353    0.14485   -0.09     0.93
## I(time^2)    0.02754    0.00453    6.07  1.7e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.66 on 27 degrees of freedom
## Multiple R-squared:  0.958,  Adjusted R-squared:  0.955
## F-statistic:  306 on 2 and 27 DF,  p-value: <2e-16
```

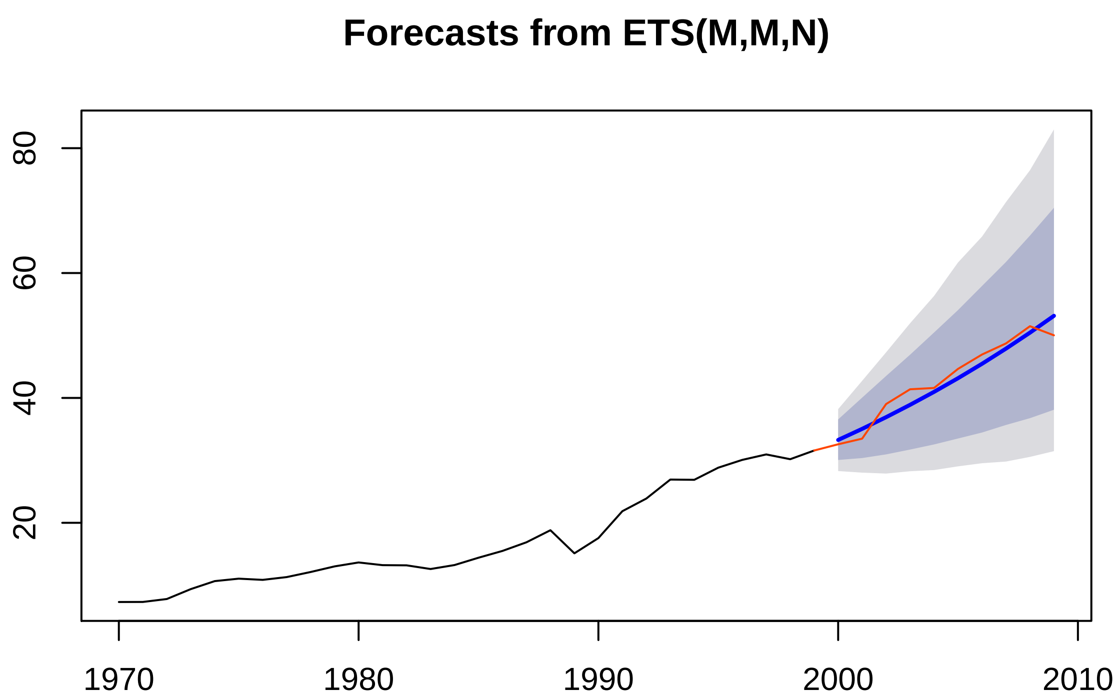
```r
predpol<-predict(regpol)

plot(tstrain,lwd=2,main="Modelo Parabólico",ylab="Pasajeros [Millones]",col="blue")
lines(years,predpol, col="red")
```



3.Modelos Exponencial

La función **ets** del paquete forecast, sirve para ajustar modelos de tipo exponencial. *Fuente:* http://www.statmethods.net/advstats/timeseries.html

```
#Exponential smoothing state space model

fitexp <- ets(tstrain)
predexp<-forecast(fitexp)

plot(predexp)
lines(tstest,lwd=1,col="orangered")
```




```
predexp
```

```
##      Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 2000          33.27 30.06 36.56 28.29 38.21
## 2001          35.05 30.37 40.04 28.04 42.74
## 2002          36.92 30.96 43.52 27.89 47.30
## 2003          38.89 31.73 46.92 28.26 51.96
## 2004          40.97 32.55 50.48 28.45 56.34
## 2005          43.16 33.51 54.07 29.05 61.70
## 2006          45.47 34.46 57.93 29.54 65.83
## 2007          47.90 35.67 61.77 29.81 71.37
## 2008          50.46 36.76 66.00 30.56 76.46
## 2009          53.15 38.11 70.45 31.48 83.00
```

```r
summary(fitexp)
```

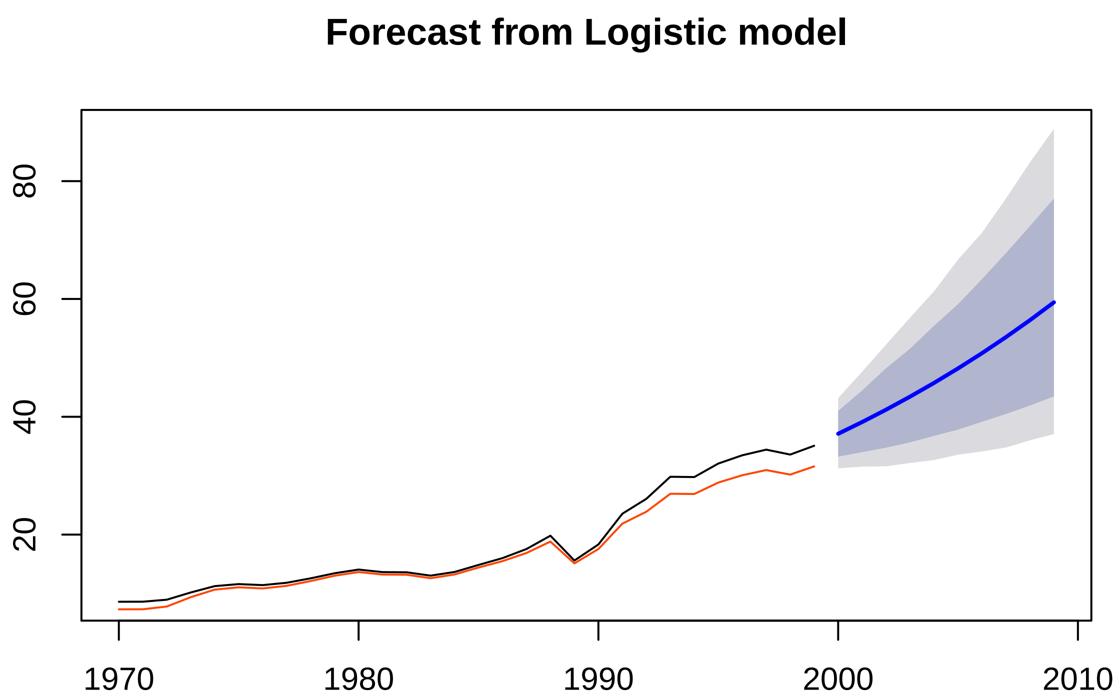
```
## ETS(M,M,N)
##
## Call:
##  ets(y = tstrain)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##
##   Initial states:
##     l = 6.9854
##     b = 1.0534
##
##   sigma:  0.0763
##
##   AIC  AICc   BIC
## 120.7 122.3 126.3
##
## Training set error measures:
##                    ME  RMSE    MAE     MPE  MAPE  MASE    ACF1
## Training set -0.05609 1.381 0.9224 -0.4938 5.503 0.748 0.00348
```

4.Modelos Logístico

Usando el paquete **growthmodels** se usa la función *logistic* para calcular la curva de crecimiento logistica.

Usage logistic(t, alpha, beta, k)

Arguments t time x size alpha upper asymptote beta growth range k growth rate




```
##
## Forecast method: ETS(M,M,N)
##
## Model Information:
## ETS(M,M,N)
##
## Call:
##  ets(y = object, lambda = lambda)
##
##   Smoothing parameters:
##     alpha = 0.8301
##     beta  = 1e-04
##
##   Initial states:
##     l = 7.5671
##     b = 1.0537
##
##   sigma:  0.0811
##
##   AIC  AICc   BIC
## 128.5 130.1 134.1
##
## Error measures:
##                    ME  RMSE   MAE     MPE  MAPE   MASE   ACF1
## Training set -0.03008 1.618 1.094 -0.5357 5.992 0.8119 0.1662
##
```

```
## Forecasts:
##      Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 2000          37.11 33.24 40.99 31.25 43.17
## 2001          39.11 34.00 44.47 31.54 47.65
## 2002          41.21 34.76 48.25 31.60 52.27
## 2003          43.42 35.67 51.57 32.15 56.83
## 2004          45.75 36.76 55.46 32.66 61.31
## 2005          48.21 37.82 59.12 33.56 66.69
## 2006          50.80 39.16 63.39 34.12 71.25
## 2007          53.53 40.47 67.78 34.79 77.06
## 2008          56.40 41.91 72.37 36.00 83.19
## 2009          59.43 43.46 77.05 37.06 88.87
```

5.Modelos Gompertz

Usando el paquete **growthmodels** se usa la función *gompertz* para calcular la curva de crecimiento.

Usage gompertz(t, alpha, beta, k)

gompertz.inverse(x, alpha, beta, k)

Arguments t time

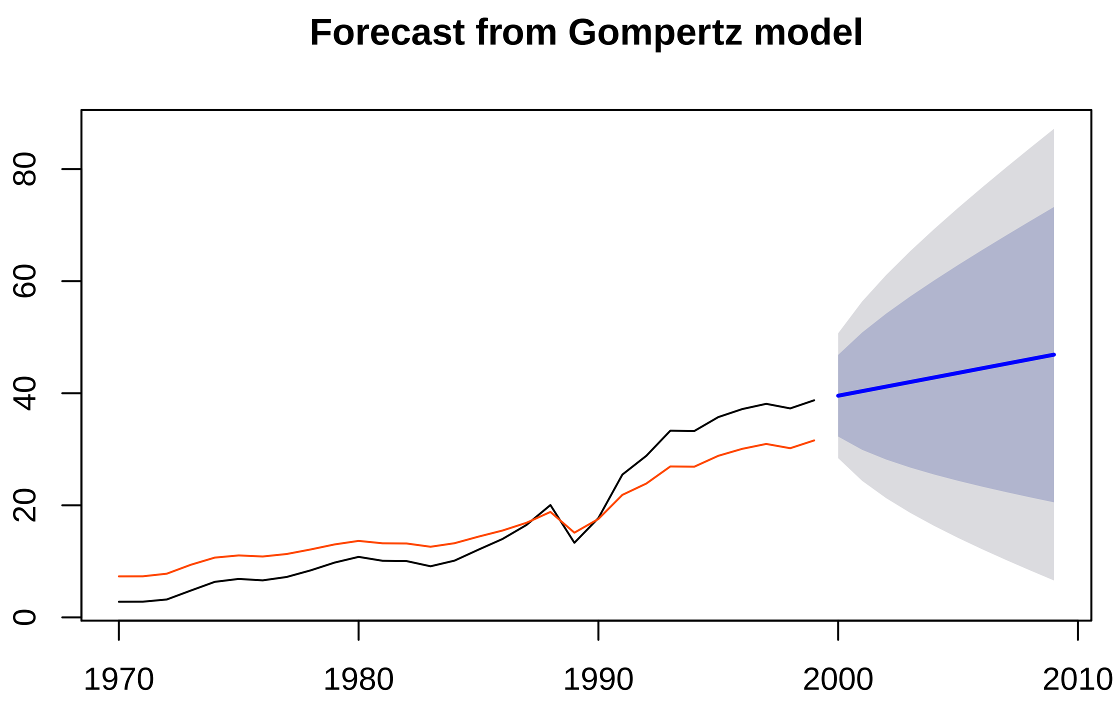
x size

alpha upper asymptote

beta growth displacement

k growth rate

```
alphag<-50 #alpha upper asymptote
betag<-6  #beta growth displacement
kg<-0.1   #growth rate

fitgom<-gompertz(tstrain, alphag, betag, kg)

predgom<-forecast(fitgom)

plot(predgom,lwd=1,main="Forecast from Gompertz model")
lines(tstrain,lwd=1,col="orangered")
```




```
summary(predgom)
```

```
##
## Forecast method: ETS(M,A,N)
##
## Model Information:
## ETS(M,A,N)
##
## Call:
##  ets(y = object, lambda = lambda)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##
##   Initial states:
##     l = 1.9151
##     b = 0.813
##
##   sigma:  0.1437
##
##   AIC  AICc   BIC
## 146.1 147.7 151.7
##
## Error measures:
```

```
##                 ME  RMSE   MAE     MPE MAPE   MASE    ACF1
## Training set 0.4145 2.365 1.545 -0.3483 11.1 0.8142 0.03371
##
## Forecasts:
##      Point Forecast Lo 80 Hi 80  Lo 95 Hi 95
## 2000          39.56 32.27 46.84 28.414 50.70
## 2001          40.37 29.91 50.84 24.368 56.37
## 2002          41.19 28.17 54.20 21.284 61.09
## 2003          42.00 26.74 57.26 18.669 65.33
## 2004          42.81 25.50 60.13 16.335 69.29
## 2005          43.63 24.38 62.88 14.186 73.07
## 2006          44.44 23.34 65.55 12.168 76.72
## 2007          45.26 22.36 68.15 10.243 80.27
## 2008          46.07 21.43 70.71  8.389 83.75
## 2009          46.89 20.53 73.24  6.586 87.18
```

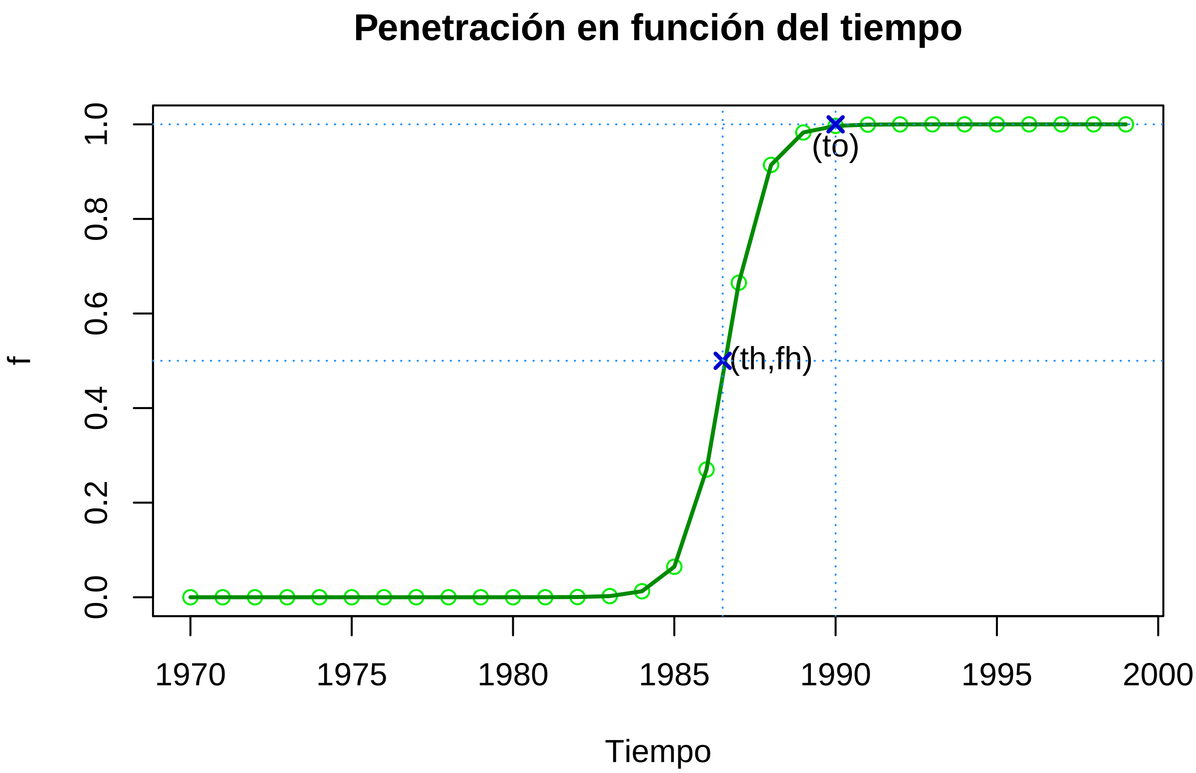
5.Modelos Fisher Pry

Modelo matemático utilizado para predecir la adopción de tecnología cuando la sustitución es impulsado por una tecnología superior en el que el nuevo producto o servicio presenta alguna ventaja tecnológica sobre el anterior.

```
time<-(years-mean(years))*2

parmlm<-as.list(fitlm$coeff)

a<-parmlm$"(Intercept)"
b<-parmlm$trend

fitpry<-1/(1+exp(-b*(time-a)))
```

Curva **S** representa la adopción del servicio, es decir la penetración del mercado.

```
plot(years,fitpry,lwd=1,col="green2", main="Penetración en función del tiempo", xlab="Tiempo",ylab="f")
lines(years,fitpry,lwd=2,col="green4")
points(1986.5,0.5,lwd=2,col="blue3",pch=4)
text(1988, 0.5, "(th,fh)")
points(1990,1,lwd=2,col="blue3",pch=4)
text(1990, 0.95, "(to)")
abline(h=1,v=1990,lty=3,col="dodgerblue")
abline(h=0.5,v=1986.5,lty=3,col="dodgerblue")
```

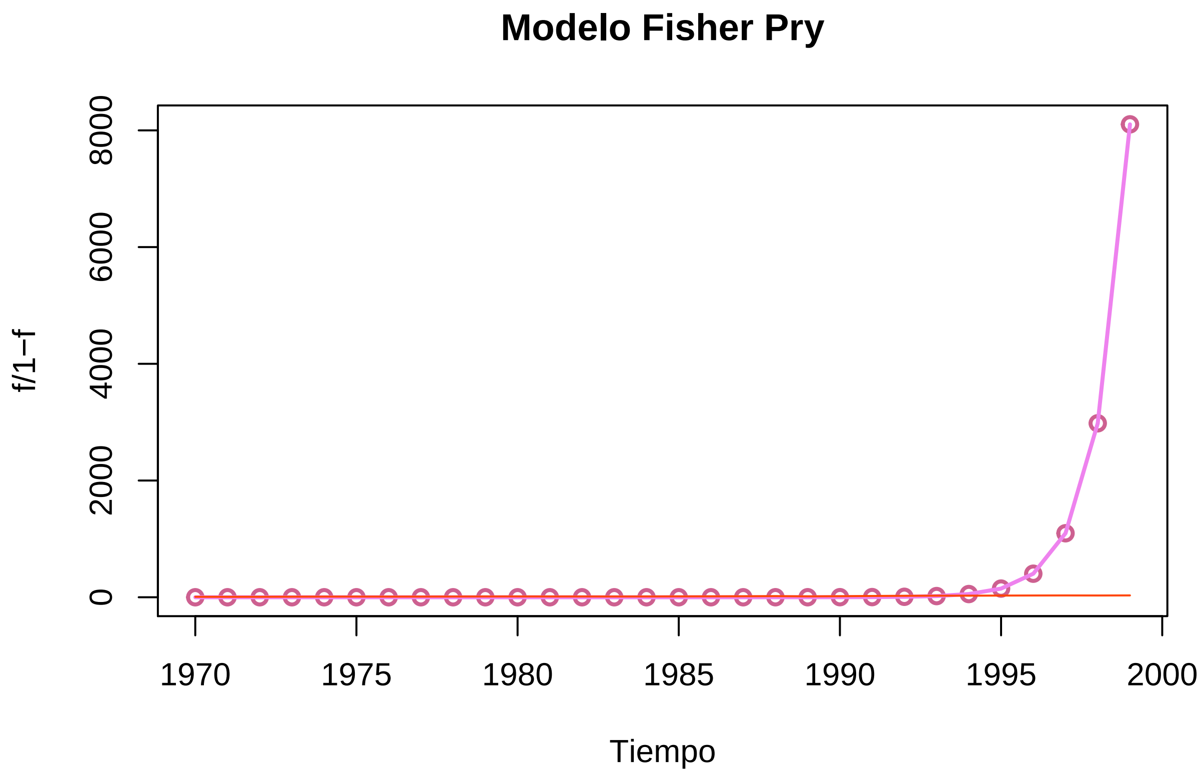


Se ha demostrado que la relación de Fisher Pry se cumple en bastantes sustituciones tecnológicas. Cuando se combina con la idea de que la curva de sustitución sigue también la curva S.

f/(1-f) = exp.2alpha(t-t0)

donde: f= fracción de tecnología antigua sustituida por la nueva. alpha= 1/2 del crecimiento porcentual anual en los primeros años. to= tiempo cuando f= 1/2

```
alpha<-2*0.5
to<-1990
tdelta<-to-years

modpry<-exp(alpha*(years-to))

plot(years,modpry,lwd=2,col="hotpink3", main="Modelo Fisher Pry", xlab="Tiempo",ylab="f/1-f")
lines(years,modpry,lwd=2,col="violet")
lines(tstrain,lwd=1,col="orangered")
```



6.Modelos Bass Model

Modelo propuesto por Frank M Bass en 1969. Este modelo propone que la difusión es motivada por la innovación que son los que adoptan un producto y los que los siguen son los imitadores.

M Numero total de personas que eventualmente comprará el producto **P** Coeficiente de Innovación **Q** Coeficiente de Imitación

Para resolver el modelo, se hace una regresión lineal para determinar los coeficientes del modelo

```
time<-(years-mean(years))*2

regpol= lm(tstrain ~ time + I(time^2))

parm<-as.list(regpol$coeff)

a<-parm$"(Intercept)"
b<-parm$time*(1)
c<-parm$"I(time^2)"


#Coeficientes del Modelo Bass

P<-c
Q<-b+P
M<-a/P
```

```
#M<-(-b-sqrt((b^2)-(4*a*c)))/(2*c)
#P<-a/M
#Q<--c*M


Coeff<-data.frame(P=P,Q=Q,M=M)
Coeff
```

```
##          P     Q    M
## 1 0.006885 0.427 2199
```

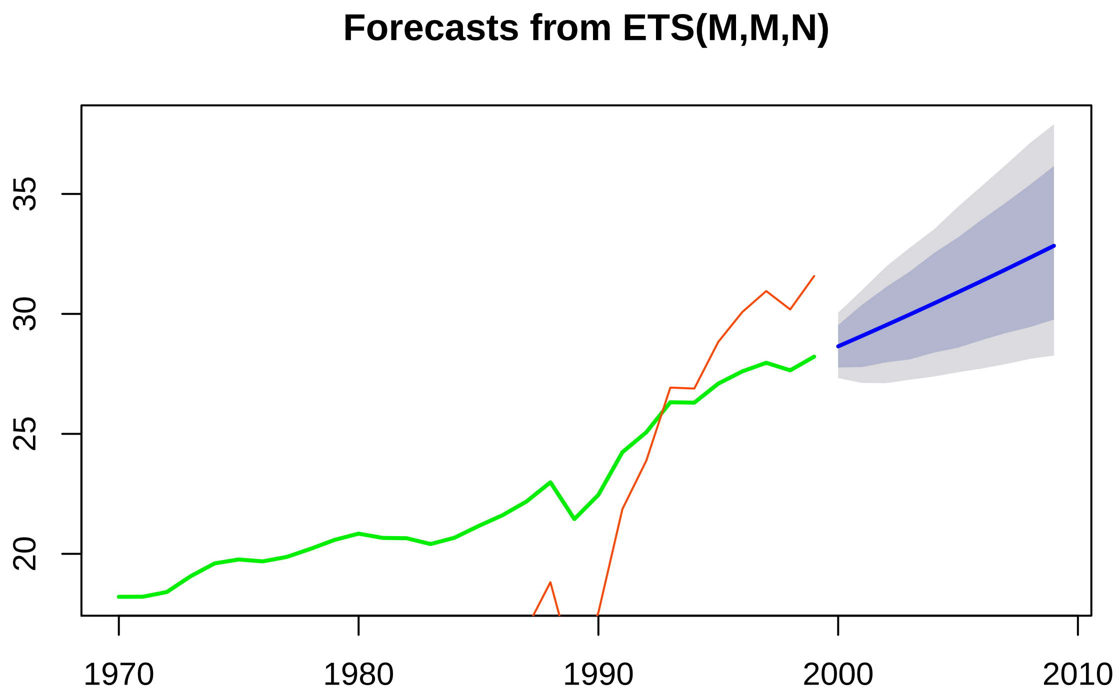
Una vez se obtienen los parámetros, se reemplazan en la fórmula del modelo:

```
ter1<-P*M
ter2<-(Q-P)*tstrain
ter3<-(Q/M)*tstrain^2

fitbas<-ter1+ter2-ter3
predbas<-forecast(fitbas)
predbas
```

```
##      Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 2000          28.65 27.77 29.54 27.32 30.05
## 2001          29.09 27.78 30.38 27.12 31.00
## 2002          29.53 27.98 31.12 27.11 31.97
## 2003          29.98 28.11 31.78 27.26 32.77
## 2004          30.44 28.39 32.54 27.39 33.53
## 2005          30.91 28.59 33.19 27.57 34.47
## 2006          31.38 28.91 33.94 27.73 35.34
## 2007          31.86 29.20 34.64 27.91 36.22
## 2008          32.34 29.45 35.38 28.13 37.12
## 2009          32.84 29.76 36.16 28.26 37.90
```

Graficamente el modelo se comporta:

```
plot(predbas,lwd=2,col="green2")

lines(tstrain,lwd=1,col="orangered")
```



Comparación de Modelos

```
#Modelo Lineal
acc_a<-accuracy(predlm)
#Modelo Parabólico
acc_b<-accuracy(predpol,tstrain)
#Modelo Exponencial
acc_c<-accuracy(predexp)
#Modelo Logístico
acc_d<-accuracy(predlog)
#Modelo Gompertz
acc_e<-accuracy(predgom)
#Modelo Fisher-Pry
acc_f<-accuracy(fitpry,tstrain) #Duda
#Modelo Bass
acc_g<-accuracy(predbas)        #Duda

acc_all<-rbind(acc_a,acc_b,acc_c,acc_d,acc_e,acc_f,acc_g)
Modelos<-c("Lineal","Parabólico", "Exponencial"
          ,"Logístico","Gompertz","Fisher-Pry","Bass")

perform<-data.frame(Modelos,acc_all)

p1<-as.list(fitlm$coeff)
```

```r
p2<-as.list(regpol$coeff)

a1<-round(p1$"(Intercept)",3)
b1<-round(p1$trend,3)

a2<-round(p2$"(Intercept)",3)
b2<-round(p2$time,3)
c2<-round(p2$"I(time^2)",3)

p<-round(P,3)
q<-round(Q,3)
m<-round(M,3)

a_all<-rbind(a1,a2,"NA","NA","NA",a1,a2)
b_all<-rbind(b1,b2,"NA","NA","NA",b1,b2)
c_all<-rbind("NA",c2,"NA","NA","NA","NA",c2)
p_all<-rbind("NA","NA","NA","NA","NA","NA",p)
q_all<-rbind("NA","NA","NA","NA","NA","NA",q)
m_all<-rbind("NA","NA","NA","NA","NA","NA",m)


parametr<-data.frame(Modelos,a=a_all,b=b_all,c=c_all,p=p_all,q=q_all,m=m_all)

perform
```

```
##       Modelos         ME    RMSE     MAE      MPE   MAPE   MASE     ACF1
## 1      Lineal -3.256e-16  2.4245  2.0510 -0.10573 13.607 1.6633  0.81359
## 2  Parabólico  8.082e-15  1.5760  1.3252 -0.96587  9.023 0.6051  1.08605
## 3 Exponencial -5.609e-02  1.3805  0.9224 -0.49381  5.503 0.7480  0.00348
## 4   Logístico -3.008e-02  1.6178  1.0944 -0.53568  5.992 0.8119  0.16622
## 5    Gompertz  4.145e-01  2.3651  1.5454 -0.34828 11.098 0.8142  0.03371
## 6  Fisher-Pry  1.678e+01 18.2789 16.7770 98.13394 98.134 0.8879 10.53795
## 7        Bass  5.825e-03  0.5597  0.3812 -0.04365  1.685 0.7490  0.01107
```

```r
paramet
```

```
##       Modelos      a    b     c     p     q        m
## 1      Lineal  4.184 0.84    NA    NA    NA       NA
## 2  Parabólico 15.144 0.42 0.007    NA    NA       NA
## 3 Exponencial     NA   NA    NA    NA    NA       NA
## 4   Logístico     NA   NA    NA    NA    NA       NA
## 5    Gompertz     NA   NA    NA    NA    NA       NA
## 6  Fisher-Pry  4.184 0.84    NA    NA    NA       NA
## 7        Bass 15.144 0.42 0.007 0.007 0.427 2199.498
```

OTRAS PRUEBAS

Modelo Bass con los datos de Cablevisón

```r
dat<-ts(c(0,0,0,6.76237,9.00095,9.7213,11.30214,13.56488,15.19591,17.01668),start=2004,end=2013,frequen
```

```r
y<-2004:2013
tdat<-(y-mean(y))*2

regdat= lm(dat ~ tdat + I(tdat^2))
a<-8.573783
b<-1.035411
c<--0.009617

M<-(-b-sqrt((b^2)-(4*a*c)))/(2*c)
P<-a/M
Q<--c*M

ter1<-P*M
ter2<-(Q-P)*dat
ter3<-(Q/M)*dat^2

fitbas<-ter1+ter2-ter3
plot(y,dat,col="blue2",pch=20,ylim=c(0,25),xlab="Años",ylab="Usuarios",main="Bass Model")
lines(y,dat,col="blue")
lines(y,fitbas,col="red")
legend("topleft",legend=c("Datos CV","Bass Model"),col=c("blue","red"),lty=1,lwd=2)
```