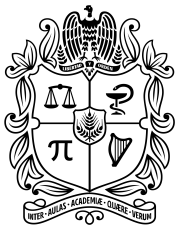
Series de tiempo univariadas - Presentación 16

Mauricio Alejandro Mazo Lopera

Universidad Nacional de Colombia
Facultad de Ciencias
Escuela de Estadística
Medellín



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Modelos SARIMA:

Como vimos en la presentación pasada, el modelo SARIMA$(p,\, d,\, q)\times(P,\, D,\, Q)_s$ se define como:

$$\Phi_P(B^s)\phi(B)(1-B^s)^D(1-B)^d X_t = \alpha+\Theta_Q(B^s)\theta(B)w_t$$

donde w_t es un ruido blanco Gaussiano y α es una constante.

Modelos SARIMA:

Como vimos en la presentación pasada, el modelo SARIMA$(p,\ d,\ q)\times(P,\ D,\ Q)_s$ se define como:

$$\Phi_P(B^s)\phi(B)(1-B^s)^D(1-B)^d X_t = \alpha+\Theta_Q(B^s)\theta(B)w_t$$

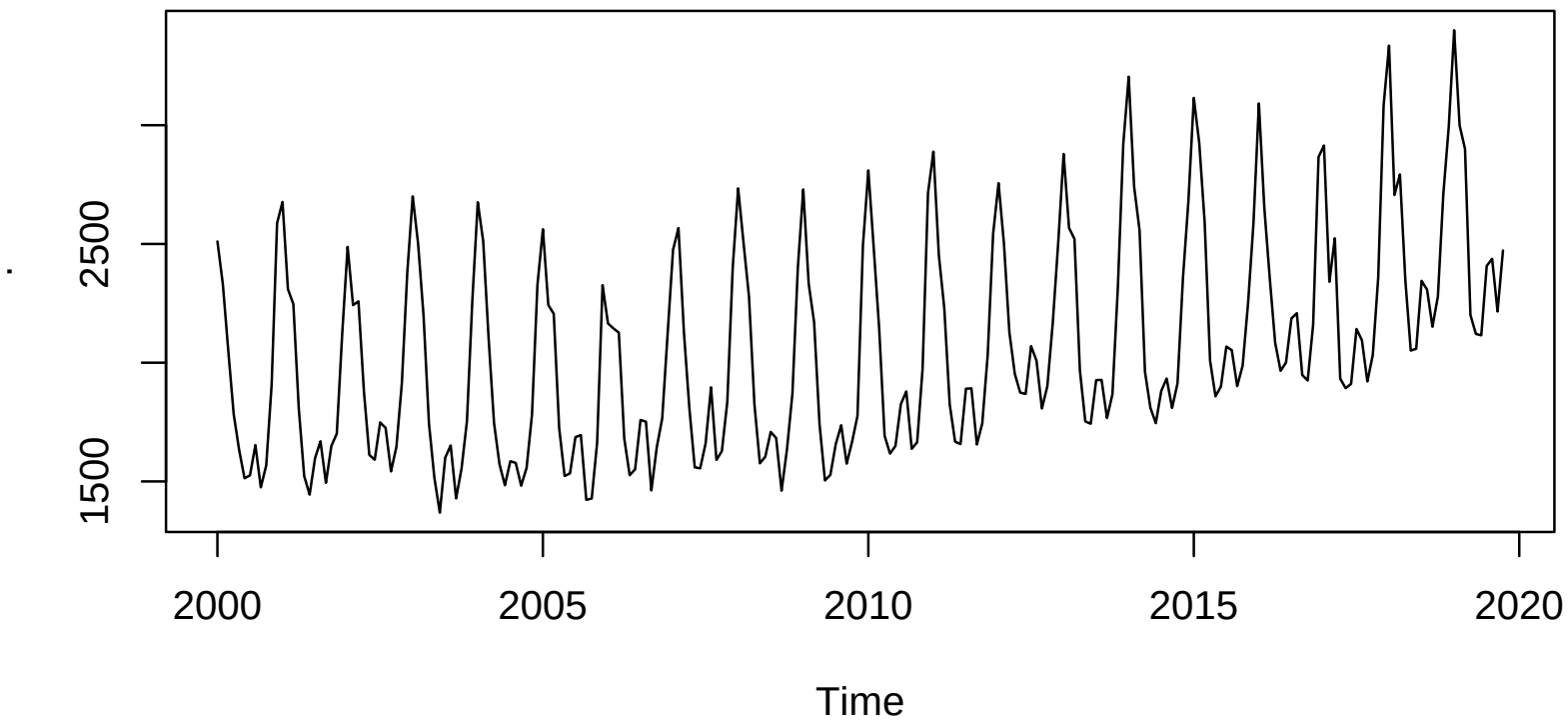
donde w_t es un ruido blanco Gaussiano y α es una constante.

La selección de los órdenes p, d, q, P, D y Q puede hacerse siguiendo varios procesos, entre los cuales vemos el siguiente:

Modelos SARIMA:

Veamos los datos consumo de gas en EEUU:

```
require(magrittr)
require(TSstudio)
USgas %>% plot()
```



Modelos SARIMA:

Aplicamos entonces el test de raíces unitarias de Dickey-Fuller:

```
require(tseries)
adf.test(USgas)
```

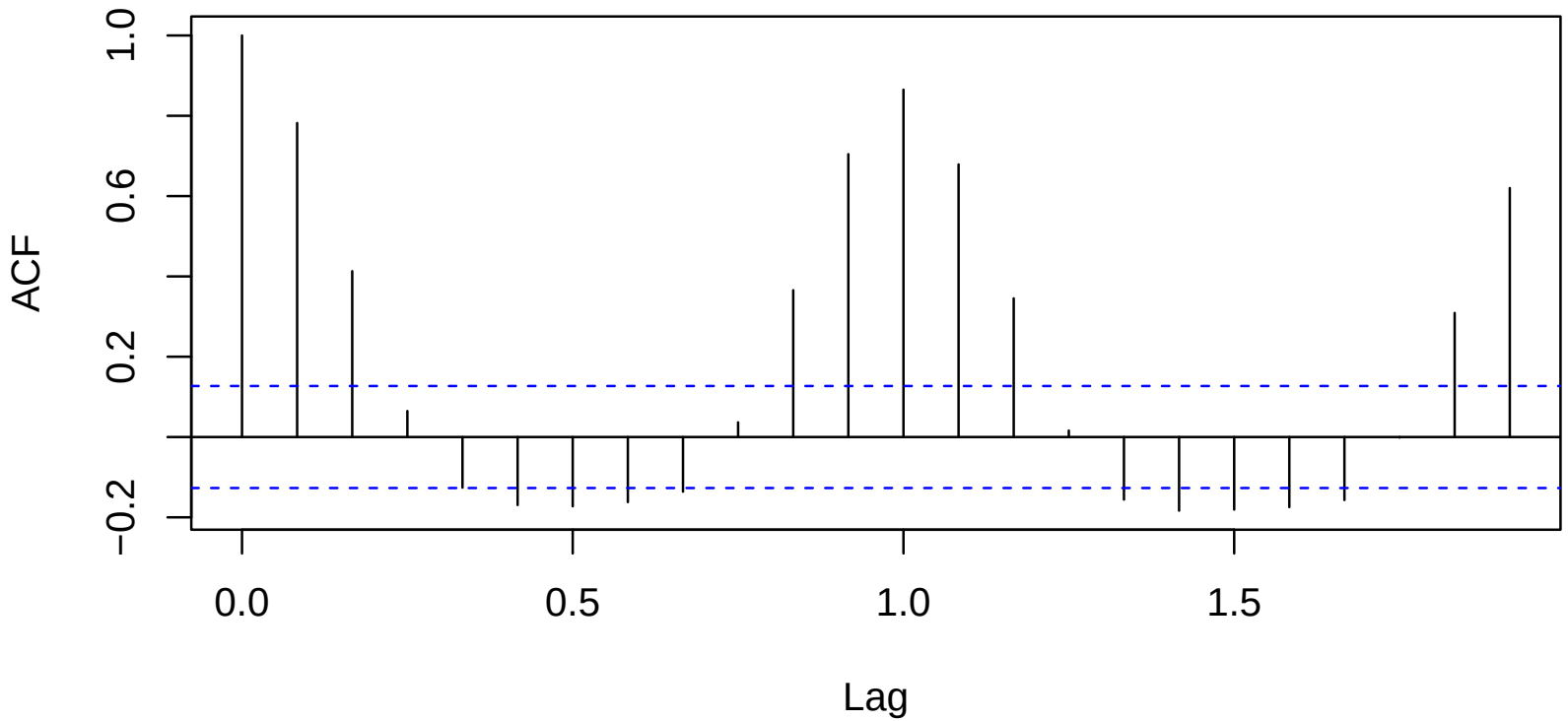
```
##
##  Augmented Dickey-Fuller Test
##
## data:  USgas
## Dickey-Fuller = -10.824, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Aparentemente no es necesario aplicar una direfencia y por tanto, $d = 0$.

Modelos SARIMA:

Vemos la ACF:

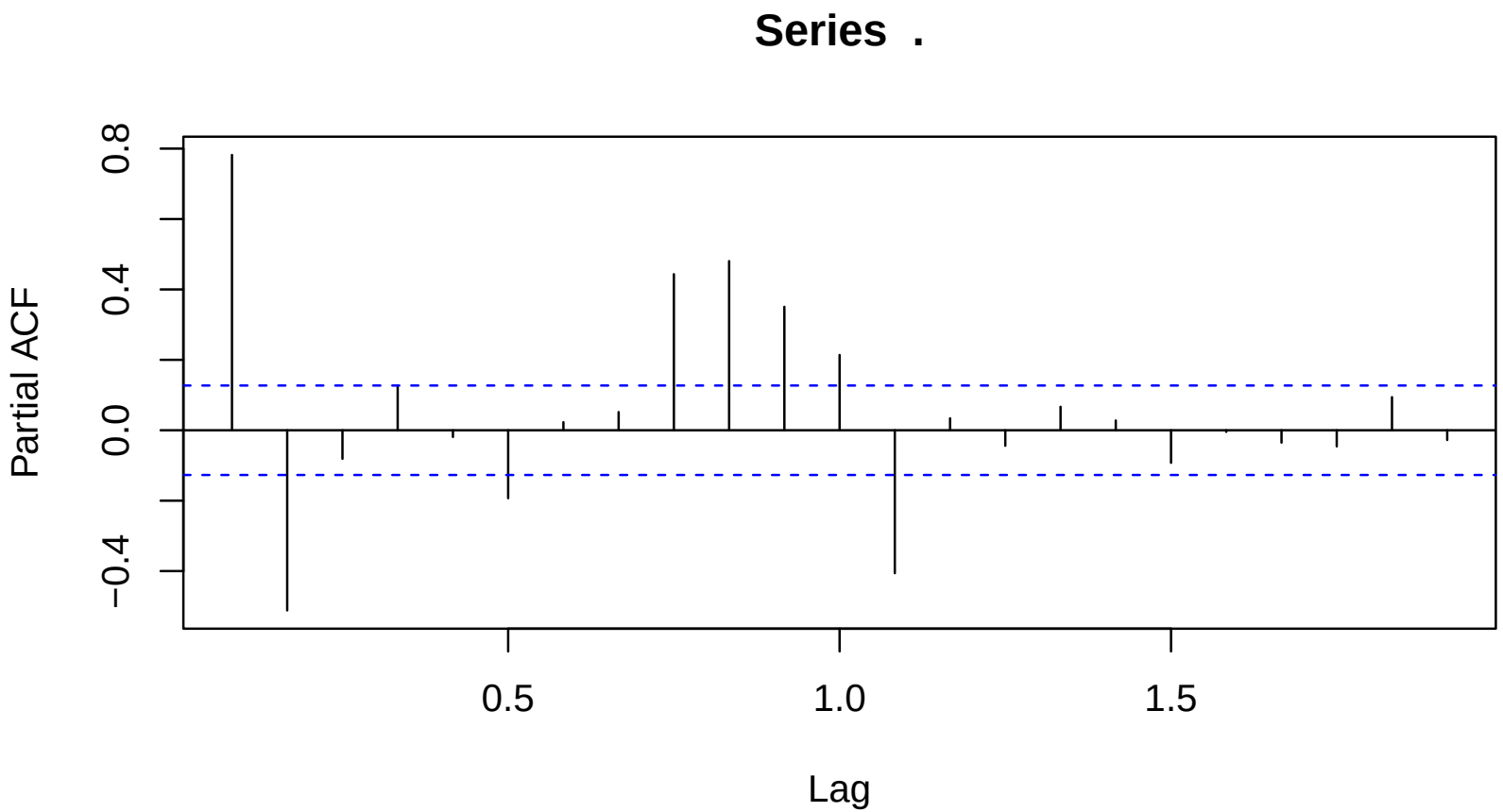
```
USgas %>% acf()
```



Hay un decaimiento estacional muy lento en los lags $12, 24, 36, \ldots$.
Esto indica tomar una diferencia estacional $D = 1$.

Vemos la PACF:

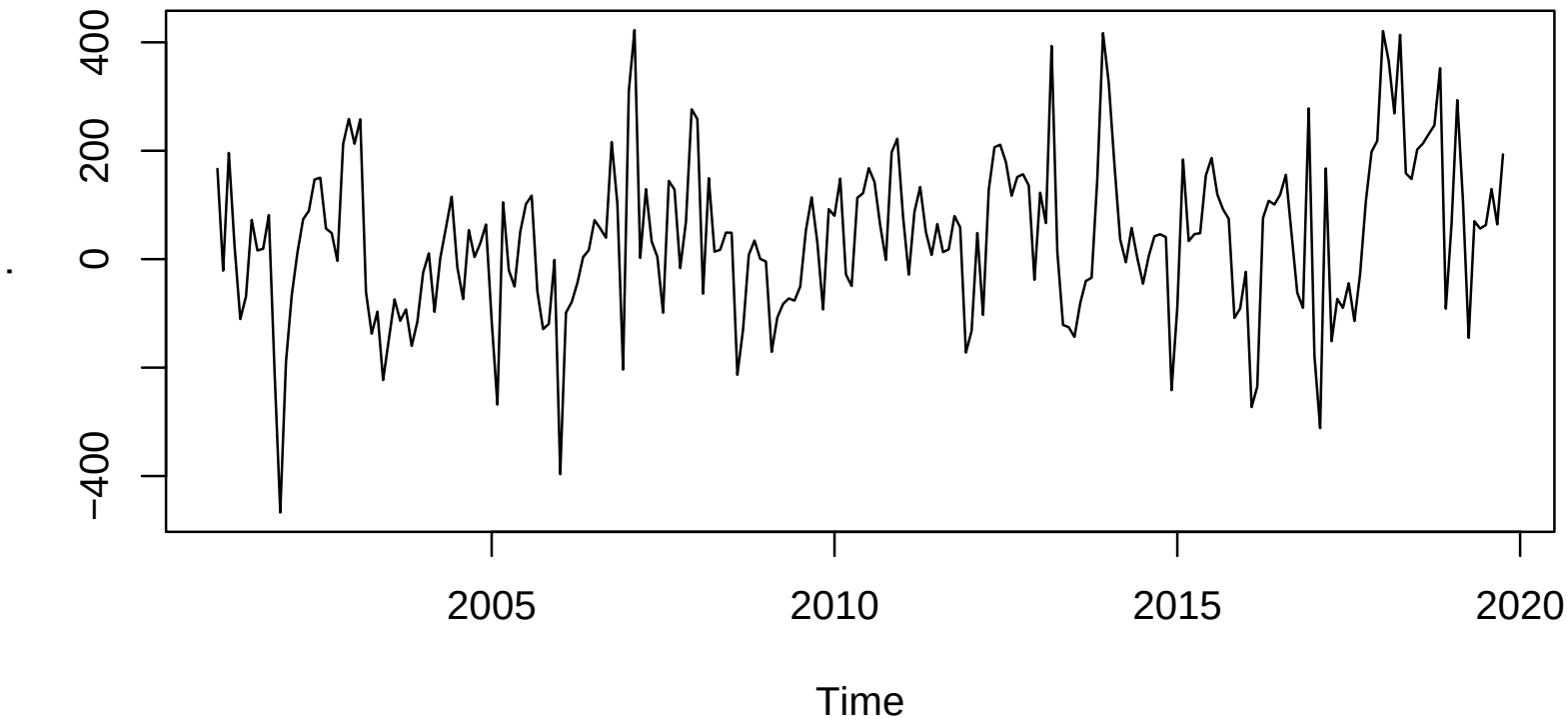
```
USgas %>% pacf()
```



Modelos SARIMA:

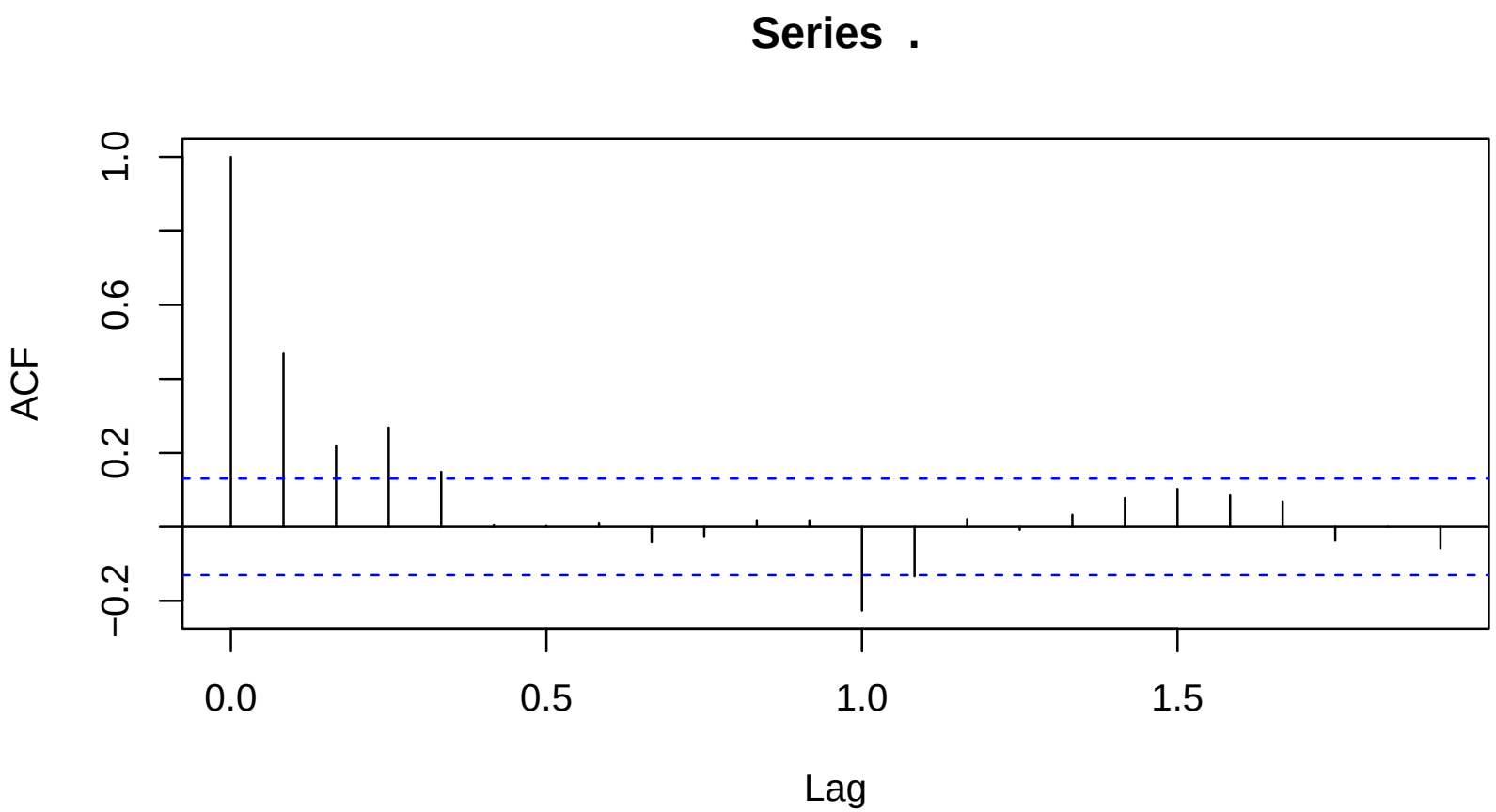
Vemos el gráfico de la serie diferenciada anualmente ($s = 12$):

```
USgas %>% diff(lag=12) %>% plot()
```



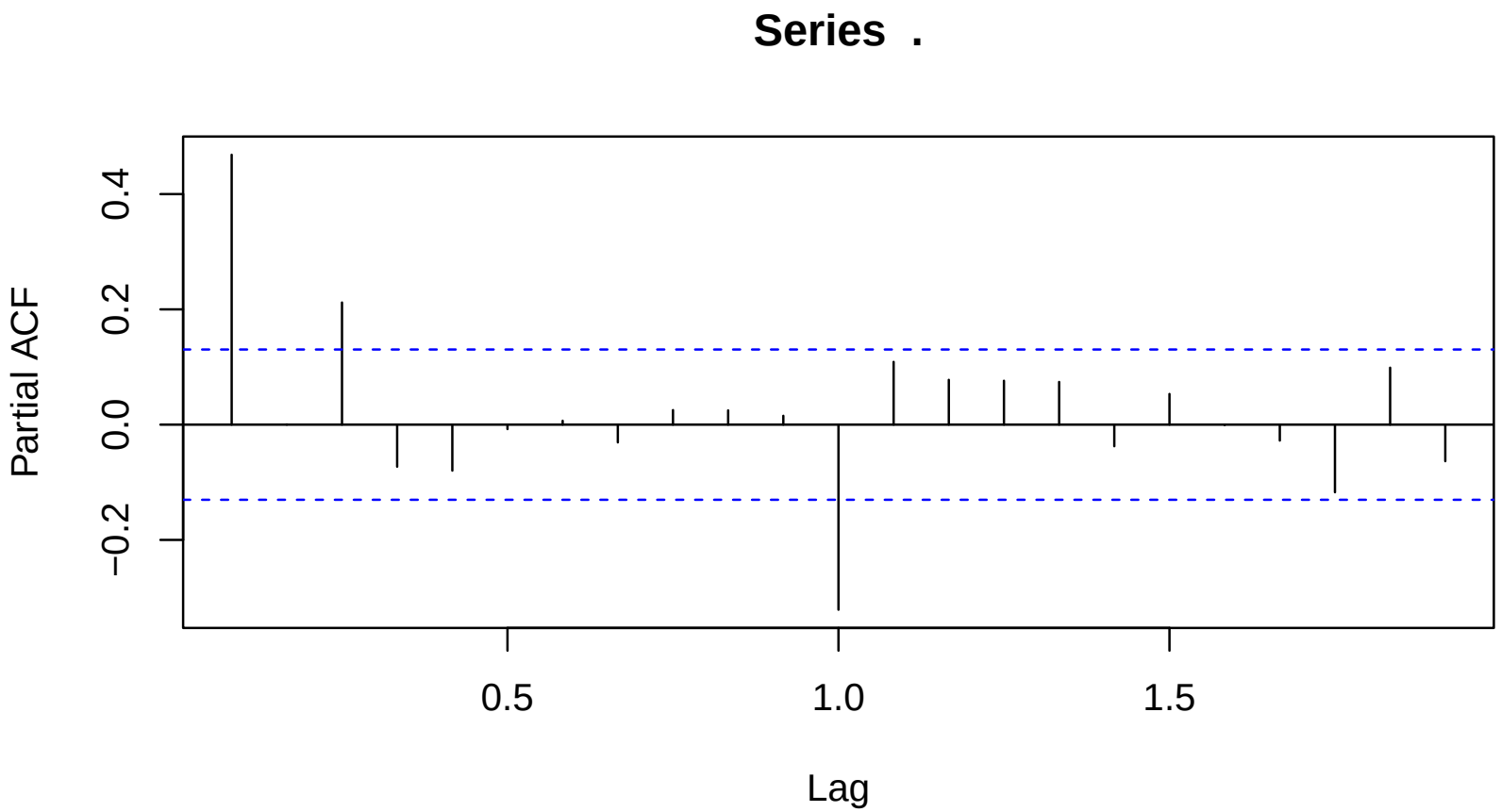
Vemos la ACF de la diferencia 12:

```
USgas %>% diff(lag=12) %>% acf()
```



Vemos la PACF:

```
USgas %>% diff(lag=12)  %>% pacf()
```



Modelos SARIMA:

El comportamiento de las series anteriores aún no evidencia claramente los ordenes p, d, P y Q. Por tanto, podemos plantear una gama de posibles combinaciones de valores que vayan desde 0 hasta 2 (Ver página 365 del **TEXTO 1** - EN R - Rami Krispin - Hands-On Time Series Analysis With R_ Perform Time Series Analysis And Forecasting Using R-Packt Publishing (2019)):

```
p <- q <- P <- Q <- 0:2
d <- 0:1
arima_grid <- expand.grid(p,d,q,P,Q)
names(arima_grid) <- c("p", "d","q", "P", "Q")
arima_grid$D <- 1
```

Modelos SARIMA:

```
arima_grid %>% dim()
```

```
## [1] 162   6
```

```
arima_grid %>% head(5)
```

```
##   p d q P Q D
## 1 0 0 0 0 0 1
## 2 1 0 0 0 0 1
## 3 2 0 0 0 0 1
## 4 0 1 0 0 0 1
## 5 1 1 0 0 0 1
```

Modelos SARIMA:

Con el fin de no sobreparametrizar el modelo, reducimos las 81 posibles combinaciones de órdenes teniendo como criterio, por ejemplo, modelos con máximo 7 parámetros:

```
require(tidyverse)
arima_grid %<>%  filter(rowSums(arima_grid)<=7)
arima_grid %>% dim()
```

```
## [1] 142   6
```

```
arima_grid %>% head(3)
```

```
##   p d q P Q D
## 1 0 0 0 0 0 1
## 2 1 0 0 0 0 1
## 3 2 0 0 0 0 1
```

Modelos SARIMA:

```
arima_grid %>% tail(10)
```

```
##     p d q P Q D
## 133 0 1 2 1 2 1
## 134 0 0 0 2 2 1
## 135 1 0 0 2 2 1
## 136 2 0 0 2 2 1
## 137 0 1 0 2 2 1
## 138 1 1 0 2 2 1
## 139 0 0 1 2 2 1
## 140 1 0 1 2 2 1
## 141 0 1 1 2 2 1
## 142 0 0 2 2 2 1
```

Modelos SARIMA:

Creamos una función que ajuste los modelos con los órdenes plantea-dos:

```
selec_aux <- function(i){
md <- NULL
md <- try(arima(USgas, order = c(arima_grid$p[i], arima_grid$d[i],
                                 arima_grid$q[i]),
    seasonal = list(order = c(arima_grid$P[i], 1, arima_grid$Q[i]),
    period=12)),  silent = TRUE)
if(class(md)=="try-error"){ }
else{
results <- data.frame(p = arima_grid$p[i],
                      d = 1,
                      q = arima_grid$q[i],
                      P = arima_grid$P[i], D
                      = 1, Q = arima_grid$Q[i],
                      AIC = md$aic)
}}
arima_search <- lapply(1:nrow(arima_grid), selec_aux) %>%
  bind_rows() %>% arrange(AIC)
```

Modelos SARIMA:

Examinamos los resultados:

```
arima_search %>% head(10)
```

```
##     p d q P D Q      AIC
## 1   1 1 1 2 1 1 2745.121
## 2   0 1 2 2 1 1 2746.093
## 3   1 1 1 1 1 2 2752.907
## 4   1 1 1 0 1 1 2752.973
## 5   1 1 1 0 1 2 2753.021
## 6   1 1 1 1 1 1 2753.915
## 7   0 1 2 0 1 1 2754.034
## 8   0 1 2 0 1 2 2754.079
## 9   0 1 2 1 1 2 2754.127
## 10 1 1 2 0 1 1 2754.814
```

Modelos SARIMA:

Según los resultados anteriores, el "mejor" modelo entre los propuestos es el $SARIMA(1,1,1) \times (2,1,1)_{12}$:

```
modelo1 <- arima(USgas, order = c(1, 1, 1),
    seasonal = list(order = c(2, 1,1), period=12))
```

Vemos los resultados con el código:

```
require(lmtest)
modelo1 %>% coeftest()
```

Modelos SARIMA:

```
##
## z test of coefficients:
##
##        Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.4056152  0.0758024   5.3510 8.749e-08 ***
## ma1  -0.9103378  0.0365763 -24.8887 < 2.2e-16 ***
## sar1 -0.0019279  0.0857188  -0.0225 0.9820564
## sar2 -0.2686752  0.0767499  -3.5007 0.0004641 ***
## sma1 -0.7231529  0.0703717 -10.2762 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

El modelo arroja los resultados:

```
modelo1
```

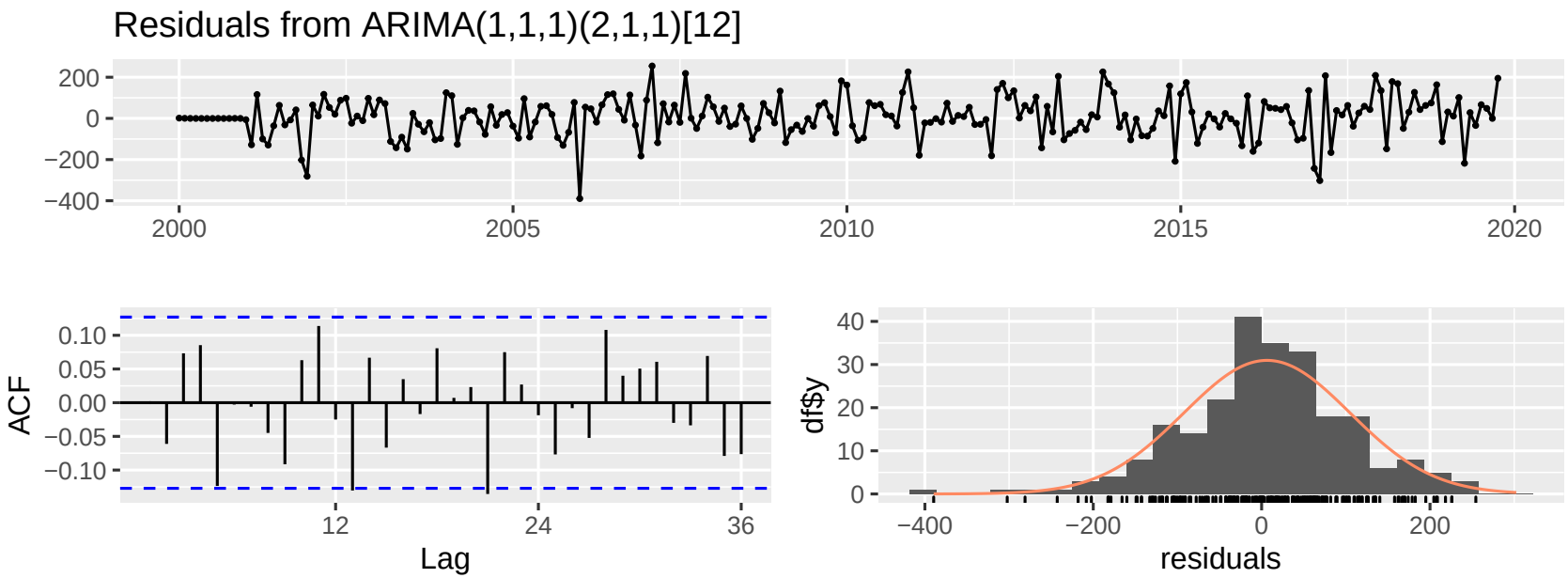
Modelos SARIMA:

```
##
## Call:
## arima(x = USgas, order = c(1, 1, 1), seasonal = list(or
##
## Coefficients:
##          ar1      ma1     sar1     sar2     sma1
##       0.4056  -0.9103  -0.0019  -0.2687  -0.7232
## s.e.  0.0758   0.0366   0.0857   0.0767   0.0704
##
## sigma^2 estimated as 10307:  log likelihood = -1366.56,
```

El diagnóstico del modelo es:

```
require(forecast)
modelo1 %>% checkresiduals(lag=25)
```

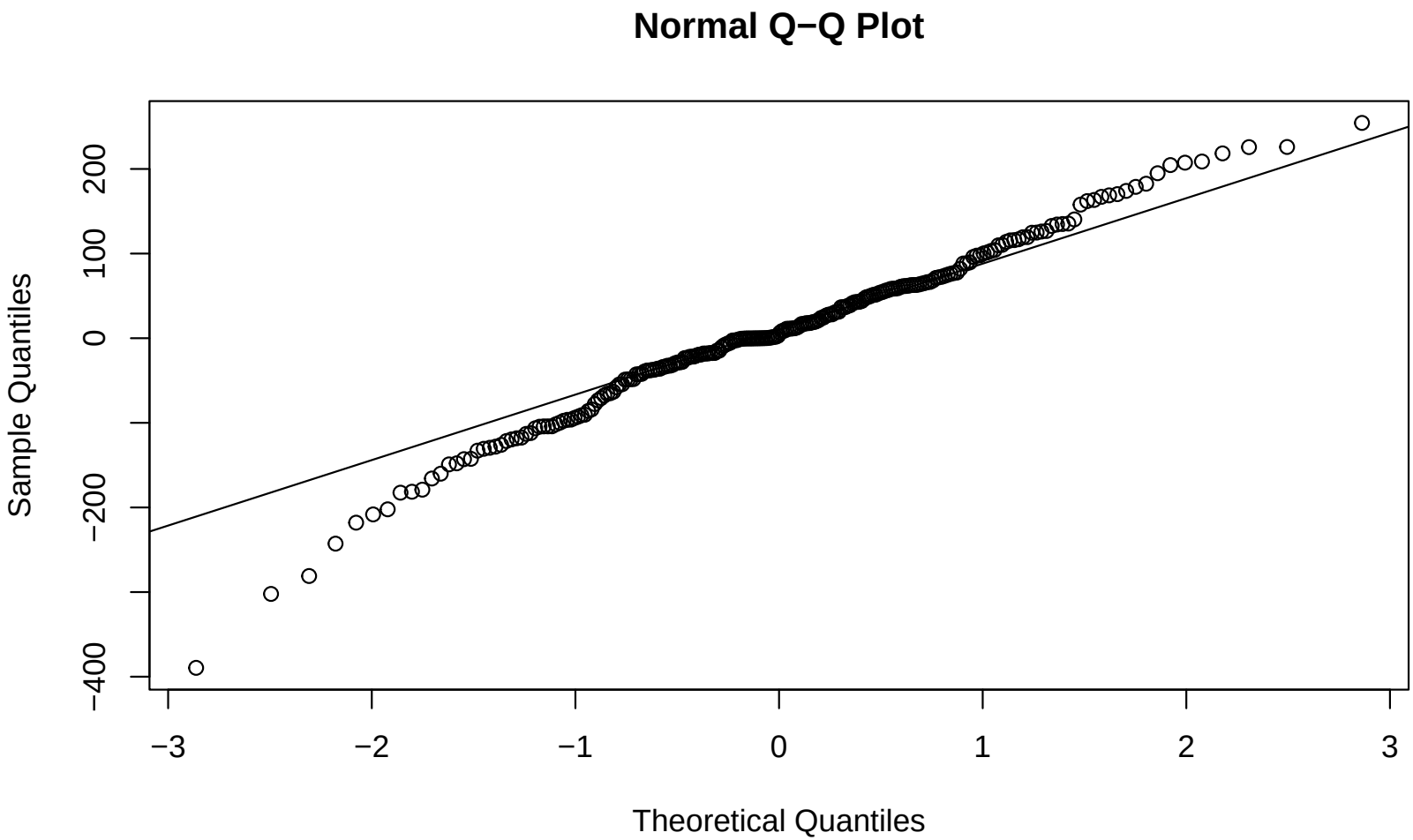
Modelos SARIMA:




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(2,1,1)[12]
## Q* = 31.755, df = 20, p-value = 0.04598
##
## Model df: 5.   Total lags used: 25
```

Modelos SARIMA:

```
modelo1$residuals %>% qqnorm()
modelo1$residuals %>% qqline()
```



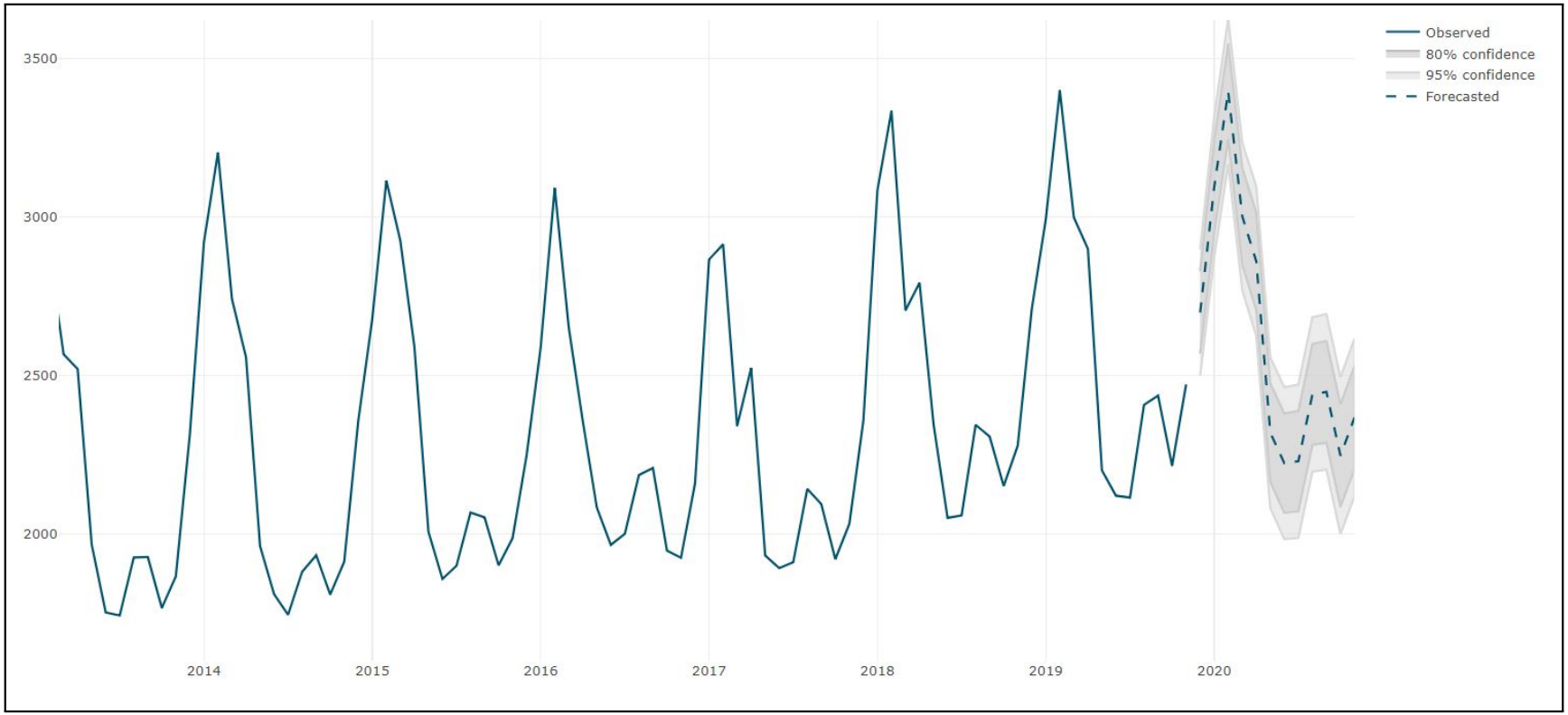
Modelos SARIMA:

```
require(forecast)
USgas_fc1 <- forecast(modelo1, h = 12)
USgas_fc1
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Nov 2019       2699.032 2568.927 2829.137 2500.053 2898.011
## Dec 2019       3094.330 2949.141 3239.518 2872.283 3316.376
## Jan 2020       3396.585 3246.556 3546.614 3167.136 3626.035
## Feb 2020       3002.766 2850.327 3155.205 2769.630 3235.901
## Mar 2020       2860.670 2706.562 3014.777 2624.983 3096.357
## Apr 2020       2319.756 2164.246 2475.266 2081.923 2557.588
## May 2020       2223.667 2066.861 2380.473 1983.853 2463.481
## Jun 2020       2229.836 2071.782 2387.891 1988.113 2471.560
## Jul 2020       2440.475 2281.196 2599.753 2196.879 2684.070
## Aug 2020       2448.721 2288.234 2609.208 2203.277 2694.164
## Sep 2020       2247.101 2085.417 2408.785 1999.826 2494.376
## Oct 2020       2368.025 2205.153 2530.897 2118.934 2617.116
```

Modelos SARIMA:

```
plot_forecast(USgas_fc)
```



Modelos SARIMA:

La función **auto.arima** del paquete **forecast** permite automatizar el proceso manual que vimos antes:

```
modelo_auto1 <- auto.arima(USgas)
modelo_auto1
```

```
## Series: USgas
## ARIMA(2,1,2)(2,1,1)[12]
##
## Coefficients:
##          ar1     ar2      ma1      ma2     sar1     sar2     sma1
##       0.0091  0.1113  -0.4935  -0.3702  -0.0038  -0.2688  -0.7131
## s.e.  0.4090  0.1897   0.4018   0.3685   0.0865   0.0771   0.0729
##
## sigma^2 = 10631:  log likelihood = -1366.25
## AIC=2748.5   AICc=2749.17   BIC=2775.83
```

Recuerde que los criterios AIC y BIC están dados por:

- **AIC**: $-2\ell(\boldsymbol{\theta}) + 2k$
- **BIC**: $-2\ell(\boldsymbol{\theta}) + \ln(n) * k$

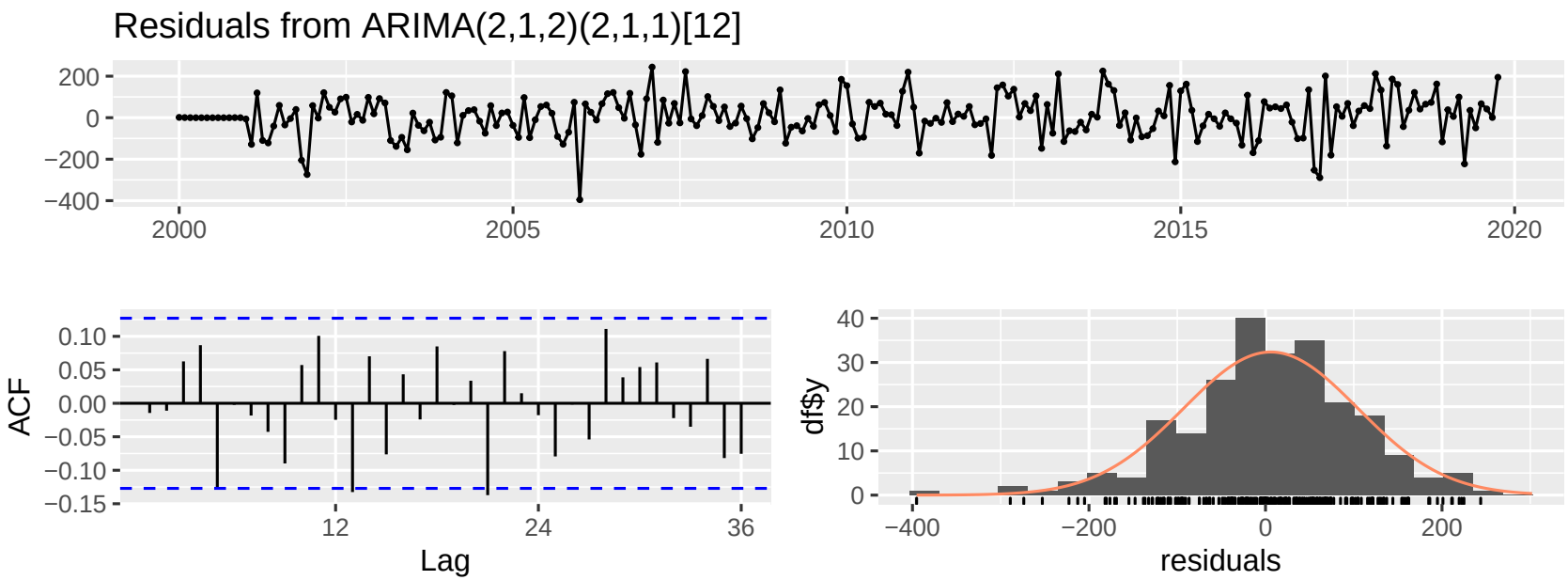
donde $\ell(\boldsymbol{\theta}) = \log[L(\boldsymbol{\theta})]$ es la función de log-verosimilitud, $\boldsymbol{\theta}$ representa el vector de parámetros a ser estimados, k es el número de parámetros que fueron estimados y n es el número de obsevaciones.

Otro criterio que puede ser usado es el AIC corregido:

- **AICc**: $-2\ell(\boldsymbol{\theta}) + \frac{2k^2+2k}{n-k-1}$

Modelos SARIMA:

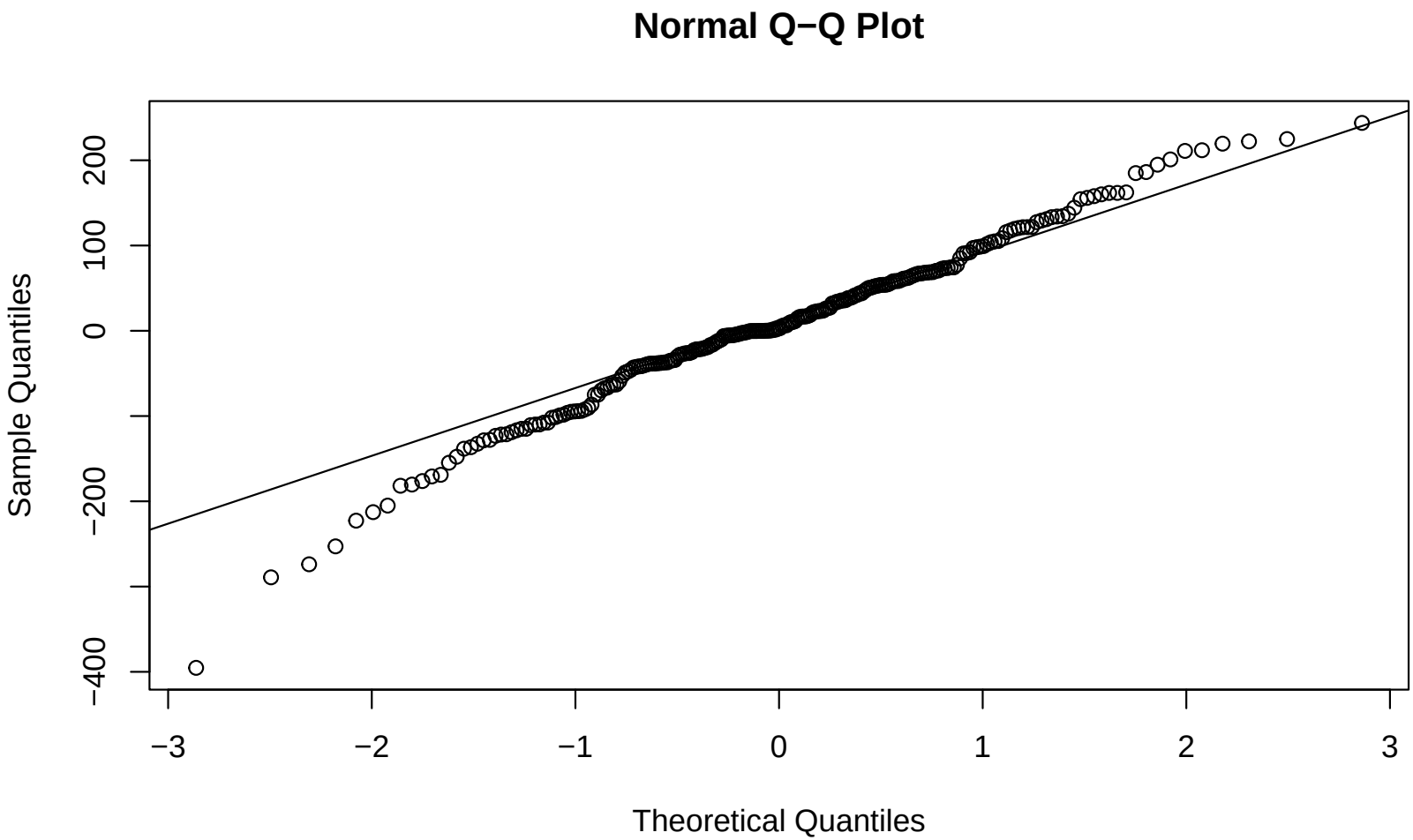
```
modelo_auto1 %>% checkresiduals(lag=25)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2)(2,1,1)[12]
## Q* = 31.256, df = 18, p-value = 0.02689
##
## Model df: 7.   Total lags used: 25
```

Modelos SARIMA:

```
modelo_auto1$residuals %>% qqnorm()
modelo_auto1$residuals %>% qqline()
```



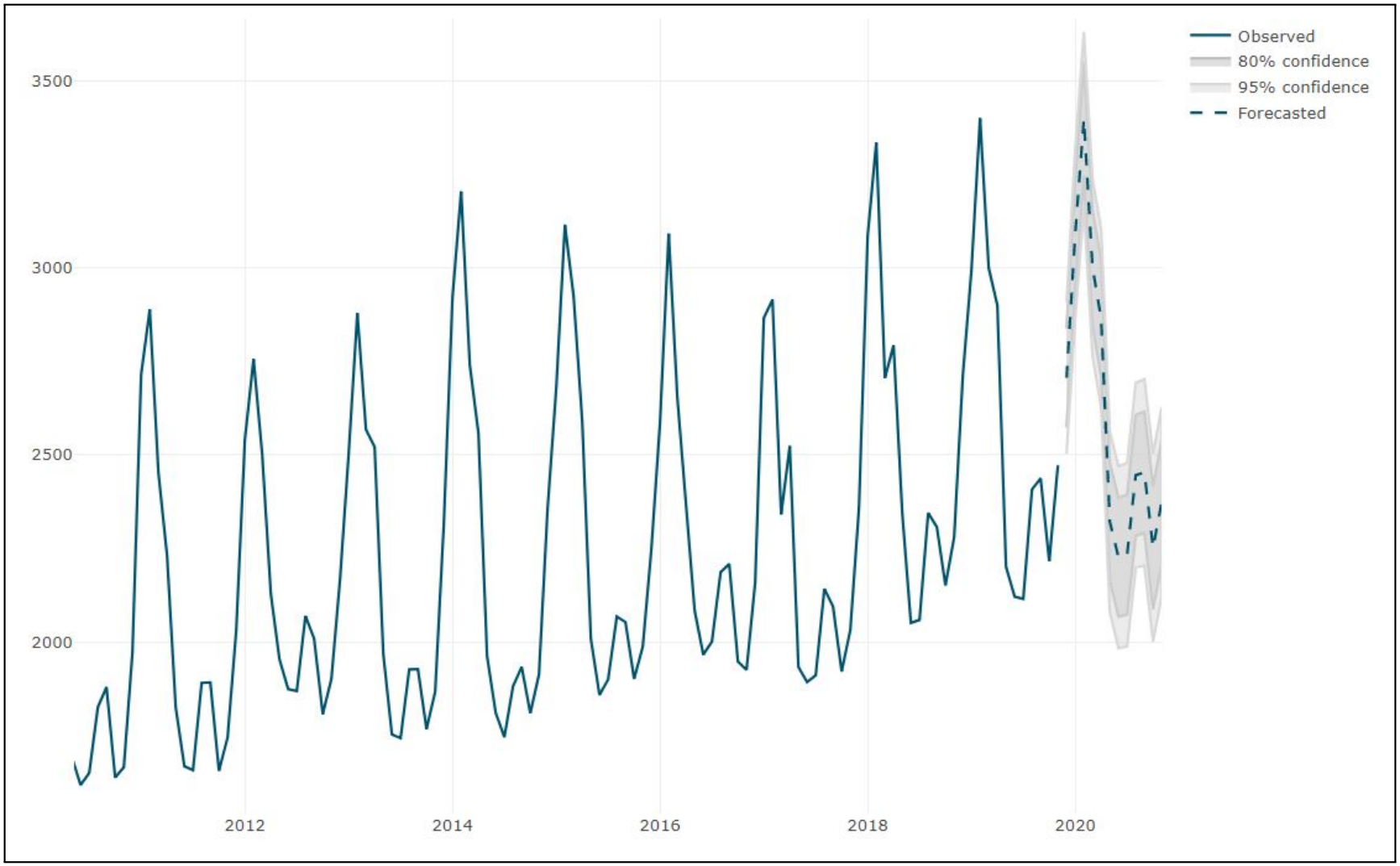
Modelos SARIMA:

```
require(forecast)
USgas_fc_auto1 <- forecast(modelo_auto1, h = 12)
USgas_fc_auto1
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Nov 2019       2705.868 2573.733 2838.002 2503.785 2907.950
## Dec 2019       3089.874 2941.210 3238.538 2862.512 3317.236
## Jan 2020       3397.921 3245.567 3550.274 3164.916 3630.926
## Feb 2020       3001.105 2846.567 3155.642 2764.760 3237.450
## Mar 2020       2864.693 2708.605 3020.782 2625.976 3103.410
## Apr 2020       2321.463 2163.957 2478.970 2080.578 2562.348
## May 2020       2226.434 2067.581 2385.286 1983.489 2469.378
## Jun 2020       2232.971 2072.796 2393.146 1988.005 2477.938
## Jul 2020       2445.581 2284.101 2607.062 2198.619 2692.544
## Aug 2020       2453.295 2290.522 2616.069 2204.355 2702.236
## Sep 2020       2251.787 2087.731 2415.842 2000.886 2502.688
## Oct 2020       2374.213 2208.885 2539.540 2121.366 2627.059
```

Modelos SARIMA:

```
plot_forecast(USgas_fc_auto1)
```



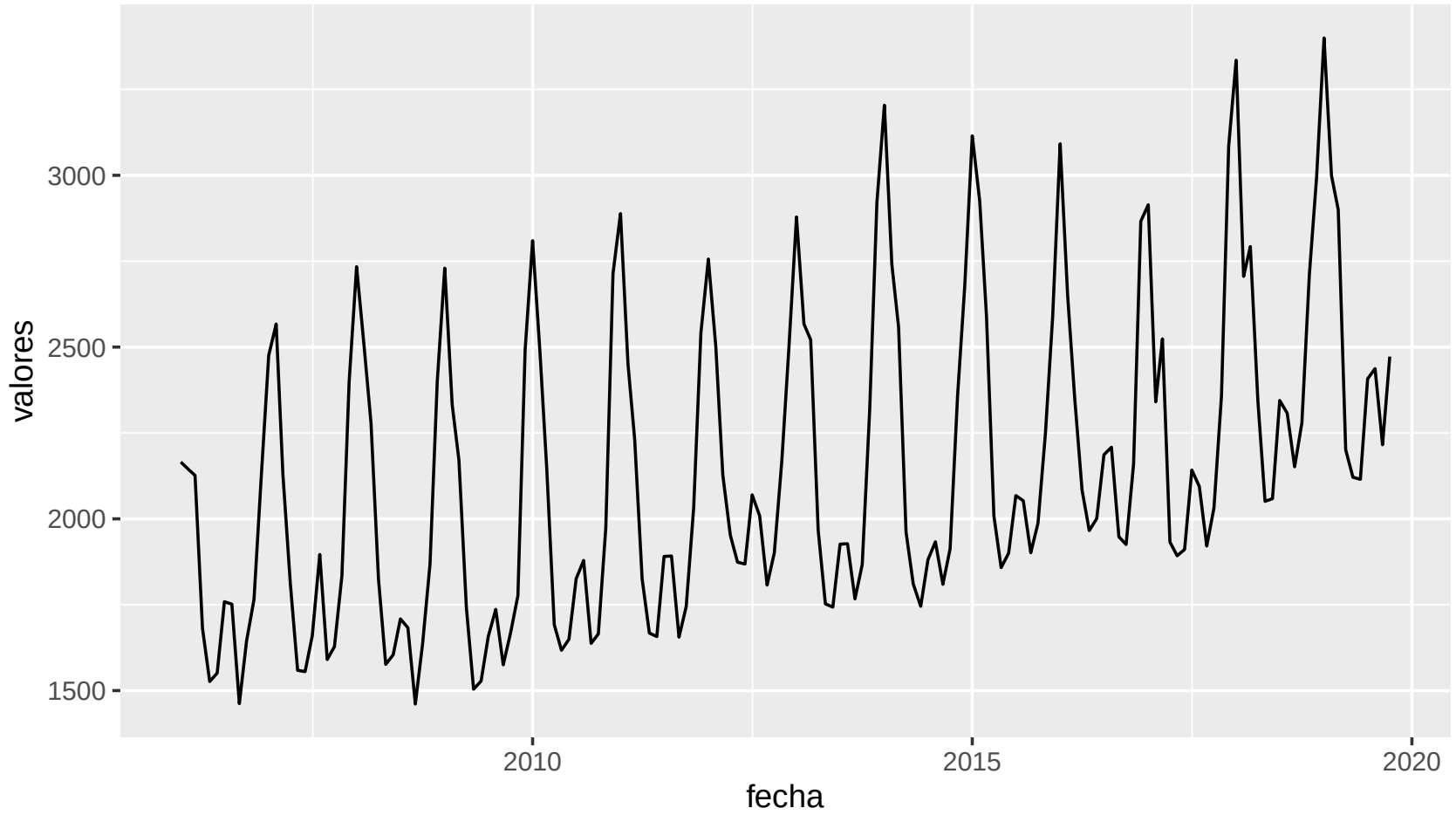
Modelos SARIMA:

```
df1 <- ts_to_prophet(USgas)
colnames(df1) <- c("fecha", "valores")
df1 %>% head(6)
```

```
##        fecha valores
## 1 2000-01-01  2510.5
## 2 2000-02-01  2330.7
## 3 2000-03-01  2050.6
## 4 2000-04-01  1783.3
## 5 2000-05-01  1632.9
## 6 2000-06-01  1513.1
```

Modelos SARIMA:

```
require(lubridate)
df2 <- df1 %>% filter(year(fecha)>2005)
df2 %>% ggplot(aes(x=fecha,y=valores)) + geom_line()
```

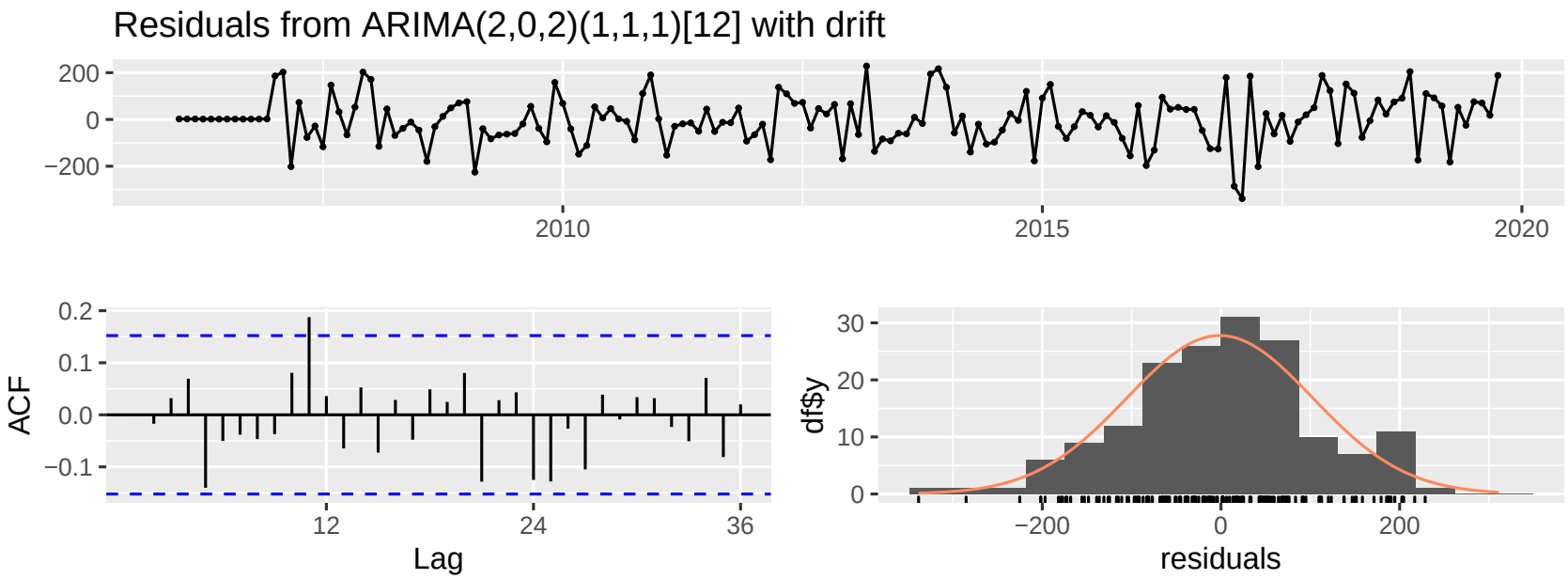


Modelos SARIMA:

```
USgas2 <- ts(df2$valores, start = c(2006,1,1),
             frequency = 12)
modelo_auto2 <- auto.arima(USgas2)
modelo_auto2
```

```
## Series: USgas2
## ARIMA(2,0,2)(1,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2     ma1      ma2    sar1     sma1   drift
##       0.4850  0.1620  0.0051  -0.1818  0.2010  -0.8108  4.6630
## s.e.  0.4465  0.2665  0.4410   0.1329  0.1202   0.0972  0.5984
##
## sigma^2 = 12146:  log likelihood = -943.77
## AIC=1903.55   AICc=1904.54   BIC=1927.84
```

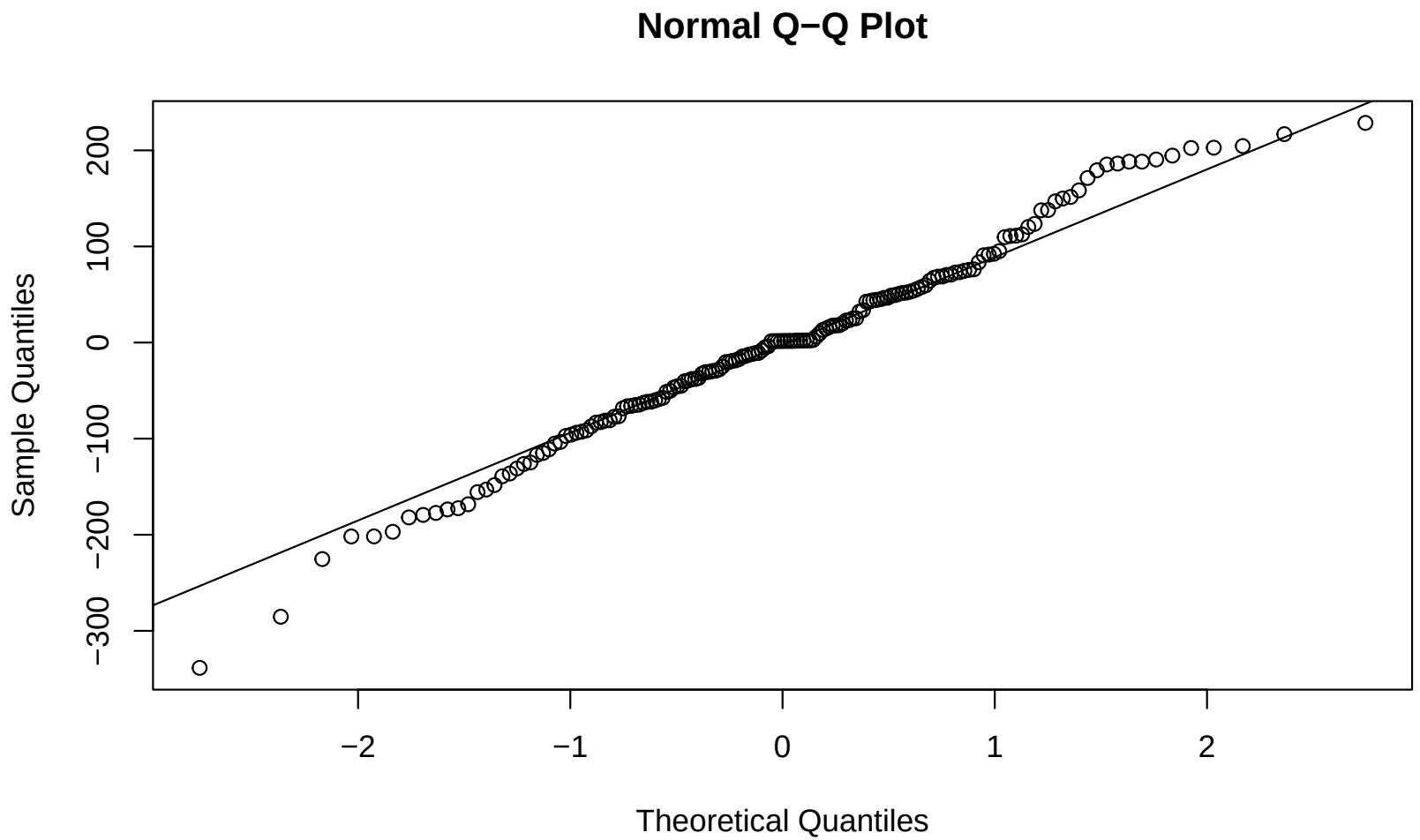
Modelos SARIMA:




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,2)(1,1,1)[12] with drift
## Q* = 28.116, df = 18, p-value = 0.06032
##
## Model df: 7.   Total lags used: 25
```

Modelos SARIMA:

```
modelo_auto2$residuals %>% qqnorm()
modelo_auto2$residuals %>% qqline()
```



Modelos SARIMA:

```
modelo_auto2$residuals %>% shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.98872, p-value = 0.2068
```

```
modelo_auto2$residuals %>% jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  .
## X-squared = 0.90367, df = 2, p-value = 0.6365
```

Modelos SARIMA:

```
require(forecast)
USgas_fc_auto2 <- forecast(modelo_auto2, h = 12)
USgas_fc_auto2
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Nov 2019       2711.599 2570.280 2852.918 2495.470 2927.727
## Dec 2019       3099.208 2941.839 3256.577 2858.534 3339.883
## Jan 2020       3418.874 3258.547 3579.200 3173.676 3664.071
## Feb 2020       2996.080 2833.638 3158.522 2747.646 3244.514
## Mar 2020       2877.206 2713.806 3040.605 2627.308 3127.103
## Apr 2020       2317.002 2153.102 2480.901 2066.339 2567.665
## May 2020       2181.206 2017.056 2345.357 1930.160 2432.252
## Jun 2020       2181.373 2017.096 2345.651 1930.132 2432.614
## Jul 2020       2416.424 2252.082 2580.766 2165.085 2667.764
## Aug 2020       2425.052 2260.677 2589.426 2173.662 2676.441
## Sep 2020       2221.369 2056.978 2385.760 1969.955 2472.784
## Oct 2020       2375.039 2210.640 2539.438 2123.612 2626.465
```

Modelos SARIMA:

```
plot_forecast(USgas_fc_auto2)
```