Business, Economical and Financial Data

Alessandro Manente, 1236286

```r
# All the libraries
library("readxl")
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(splines)
library(gam)
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.20
```

Data Exploration

```r
data <- read_excel("OS.xlsx")
data
```

```
## # A tibble: 32 x 3
##    Quarter Android   iOS
##    <chr>     <dbl> <dbl>
##  1 Q1 '10      9.6  15.4
##  2 Q2 '10     17.2  14.1
##  3 Q3 '10     25.3  16.6
##  4 Q4 '10     30.5  15.8
##  5 Q1 '11     36.4  16.9
##  6 Q2 '11     43.4  18.2
##  7 Q3 '11     52.5  15
##  8 Q4 '11     50.9  23.8
##  9 Q1 '12     56.9  22.5
## 10 Q2 '12     64.2  18.8
## # ... with 22 more rows
```

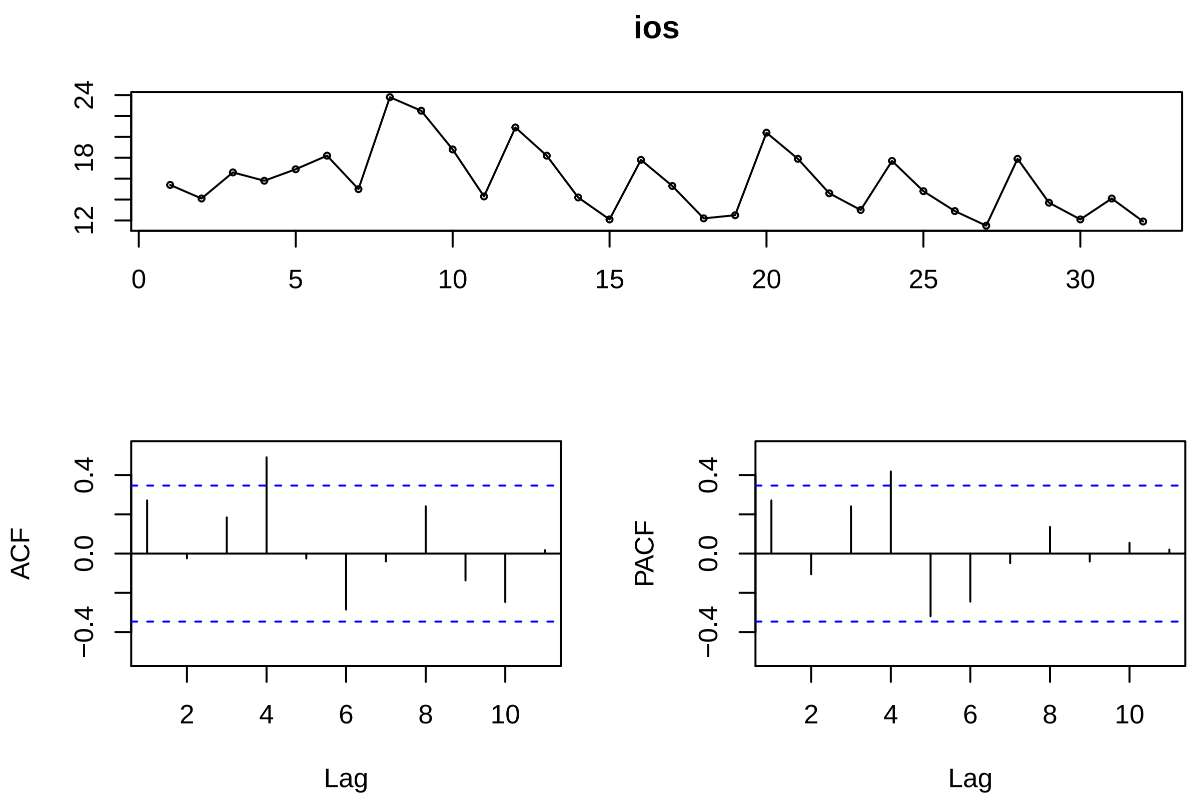
```r
summary(data)
```

```
##    Quarter             Android           iOS
##  Length:32          Min.   : 9.60   Min.   :11.50
```

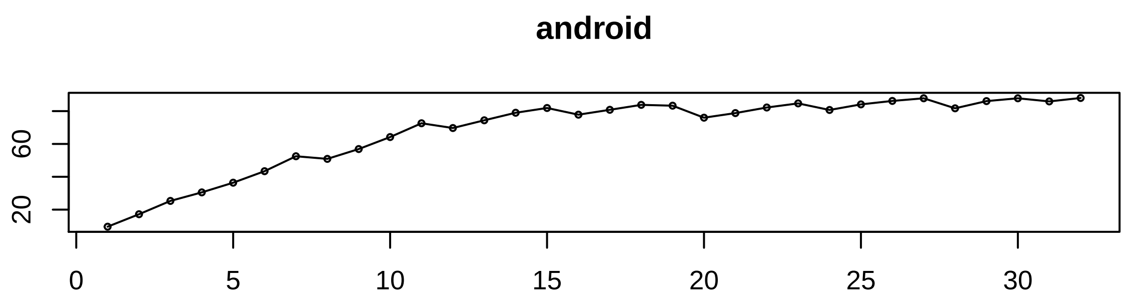
```
##   Class :character   1st Qu.:55.80   1st Qu.:13.53
##   Mode  :character   Median :78.90   Median :15.15
##                      Mean   :68.13   Mean   :15.85
##                      3rd Qu.:83.88   3rd Qu.:17.90
##                      Max.   :88.00   Max.   :23.80
```

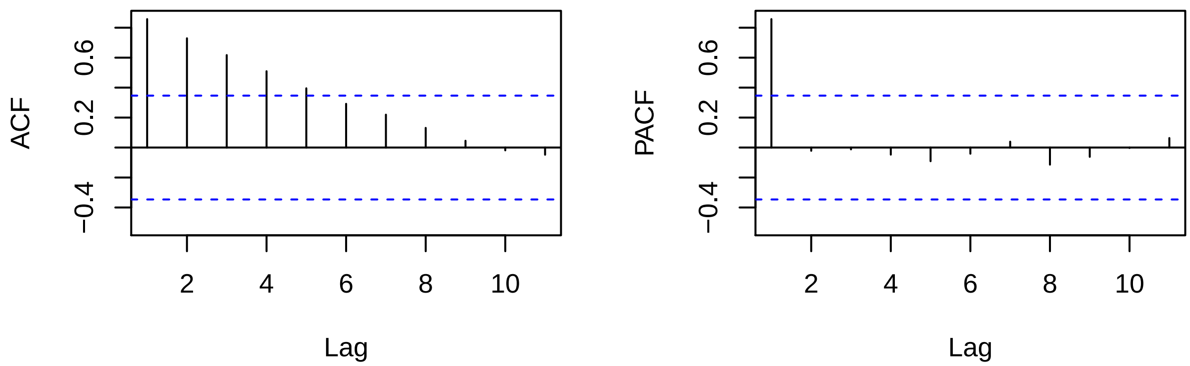
```
ios = data$iOS
ts_ios = ts(ios, frequency=4)
android = data$Android
ts_android = ts(android, frequency=4)
```

```
tsdisplay(ios)
```




```
tsdisplay(android)
```





We can see that iOS data have a small decreasing trend and seasonality, although not perfect; while Android data shows no seasonality eand a general increasing trend.

Exercise 1 and 2

Provide some modelling options for the series "iOS", by possibly accounting also for the information regarding "Android". (8pt) For each of the modelling solutions proposed, discuss the results obtained, highlighting merits and limitations of these. (8pt)

Linear Model

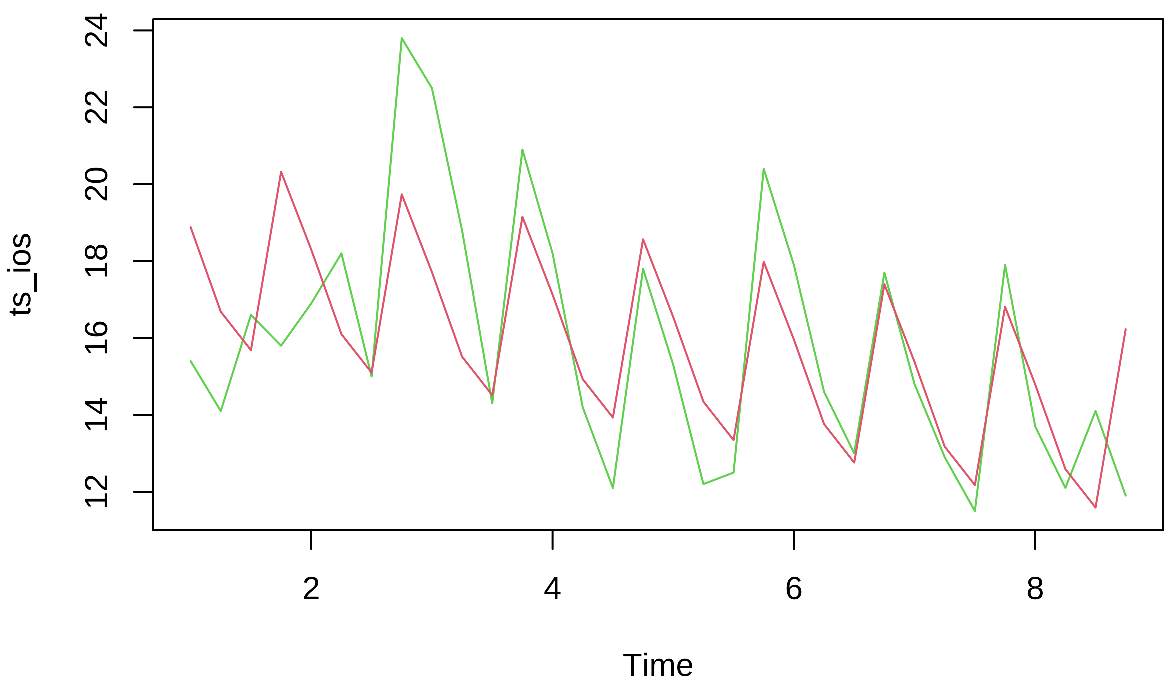
```
ios_tslm = tslm(ts_ios~ trend+season)
summary(ios_tslm)
```

Linear model done without taking into account data for Android sales

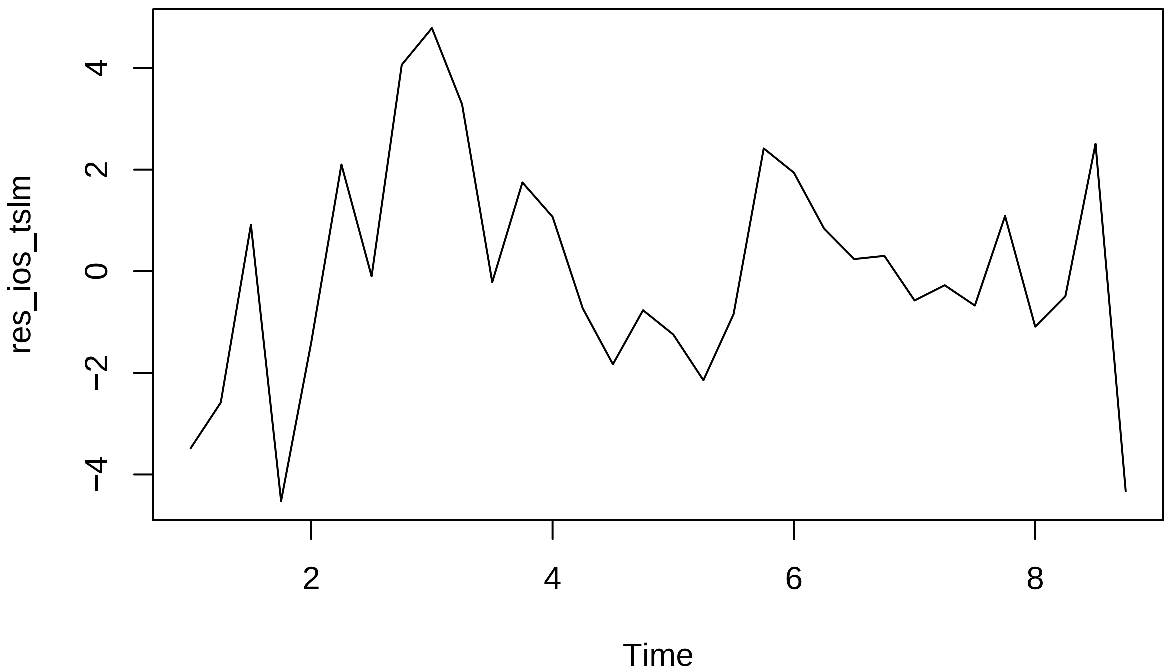
```
##
## Call:
## tslm(formula = ts_ios ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -4.5219 -1.1292 -0.2451  1.2522  4.7853
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.03058    1.08120  17.601 2.53e-16 ***
## trend       -0.14621    0.04565  -3.203  0.00347 **
## season2     -2.05379    1.18422  -1.734  0.09427 .
## season3     -2.90759    1.18686  -2.450  0.02106 *
## season4      1.87612    1.19124   1.575  0.12692
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.367 on 27 degrees of freedom
## Multiple R-squared:  0.5191, Adjusted R-squared:  0.4478
## F-statistic: 7.285 on 4 and 27 DF,  p-value: 0.0004089
```

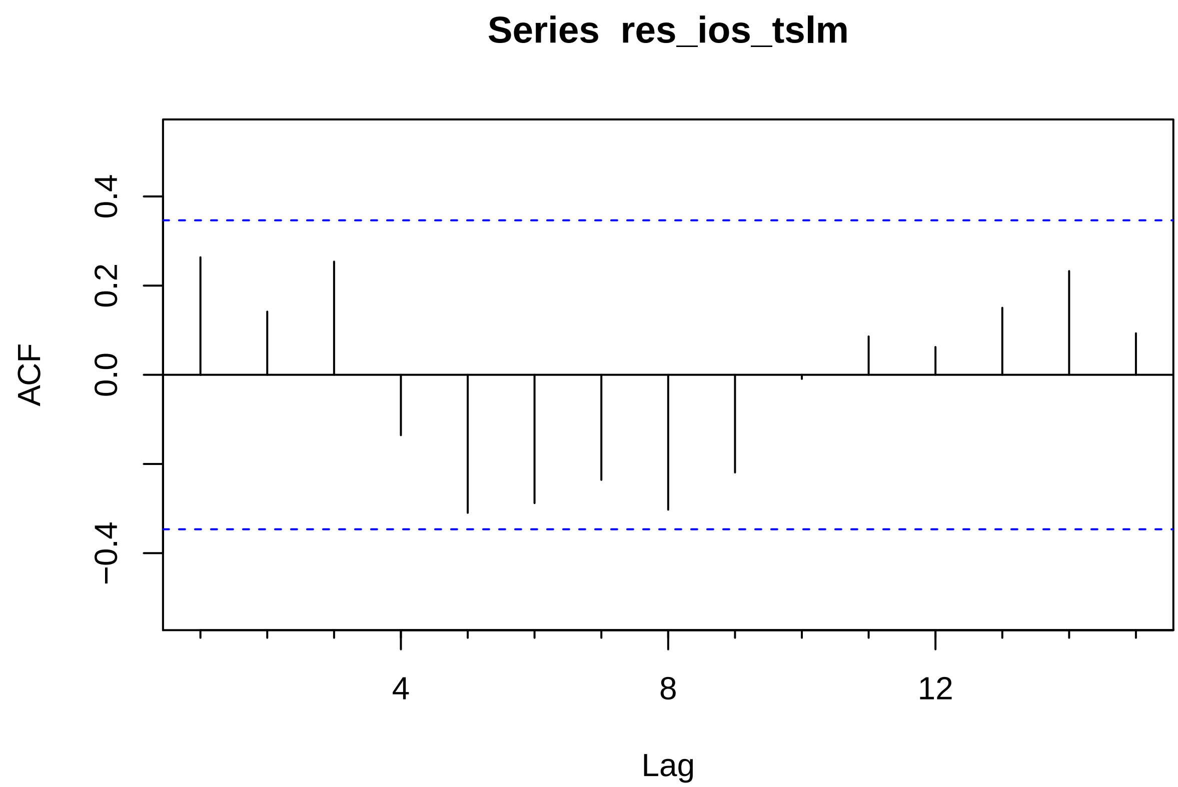
```
fit_ios_tslm<- fitted(ios_tslm)
plot(ts_ios, col=3)
lines(fit_ios_tslm, col=2)
```




```
res_ios_tslm<- residuals(ios_tslm)
plot(res_ios_tslm)
```

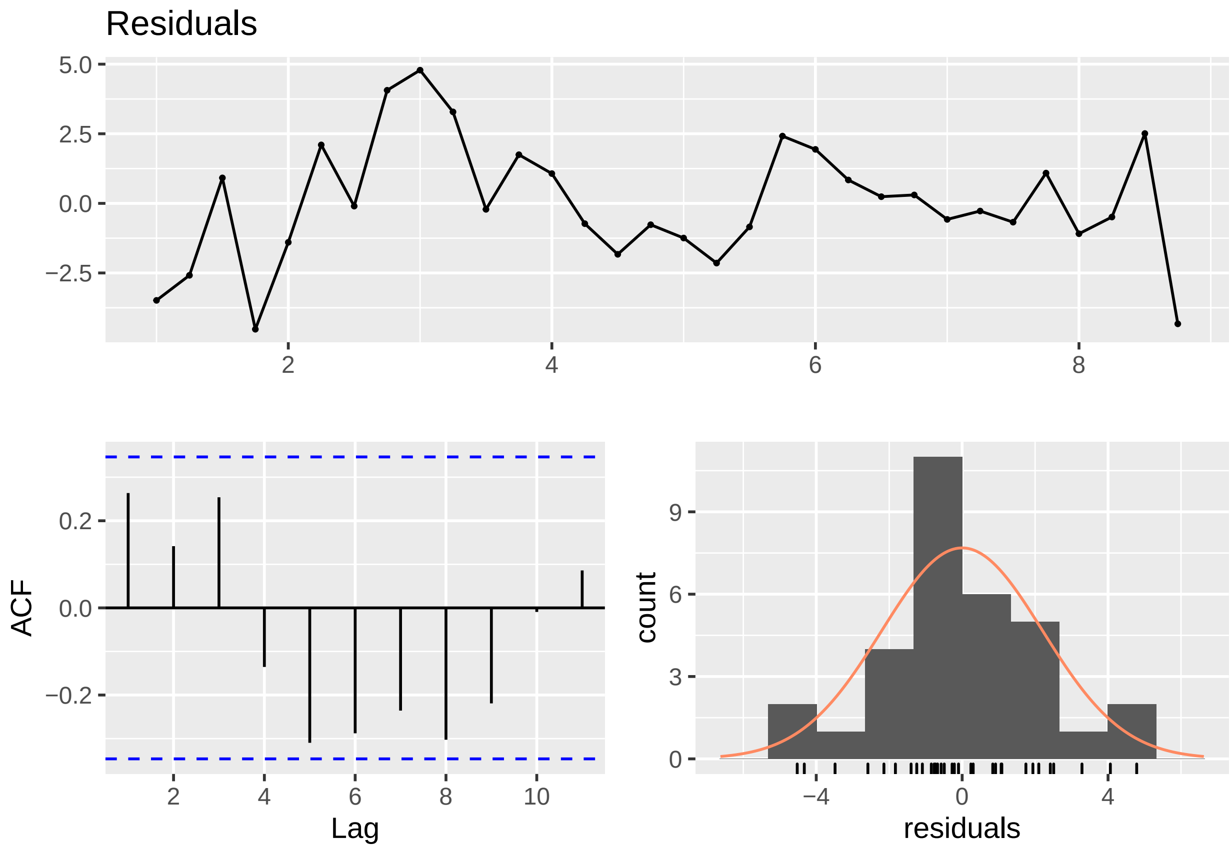



```
Acf(res_ios_tslm)
```




```
checkresiduals(res_ios_tslm)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



The model performs better on the last part of the data and shows no autocorrelation on the residuals, but the time series is clearly non linear, hence the need for more complex models.

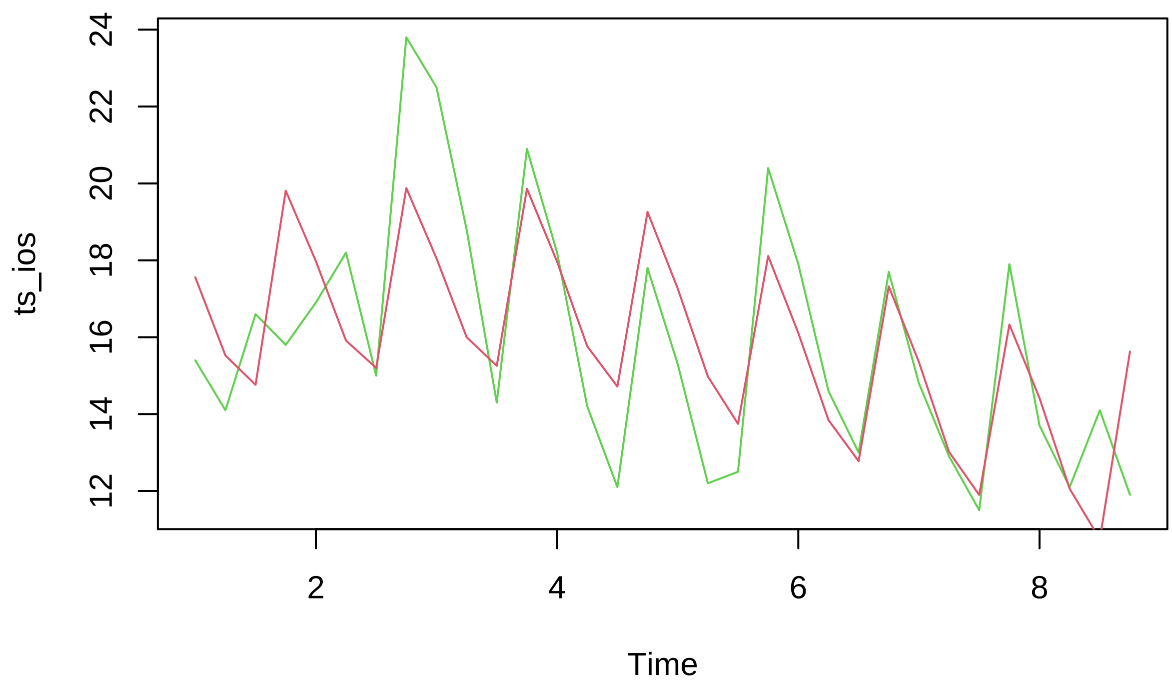
```
ios_tslm_a = tslm(ts_ios~ trend+season+ts_android)
summary(ios_tslm_a)
```

Linear Model done taking into account data for Android sales.

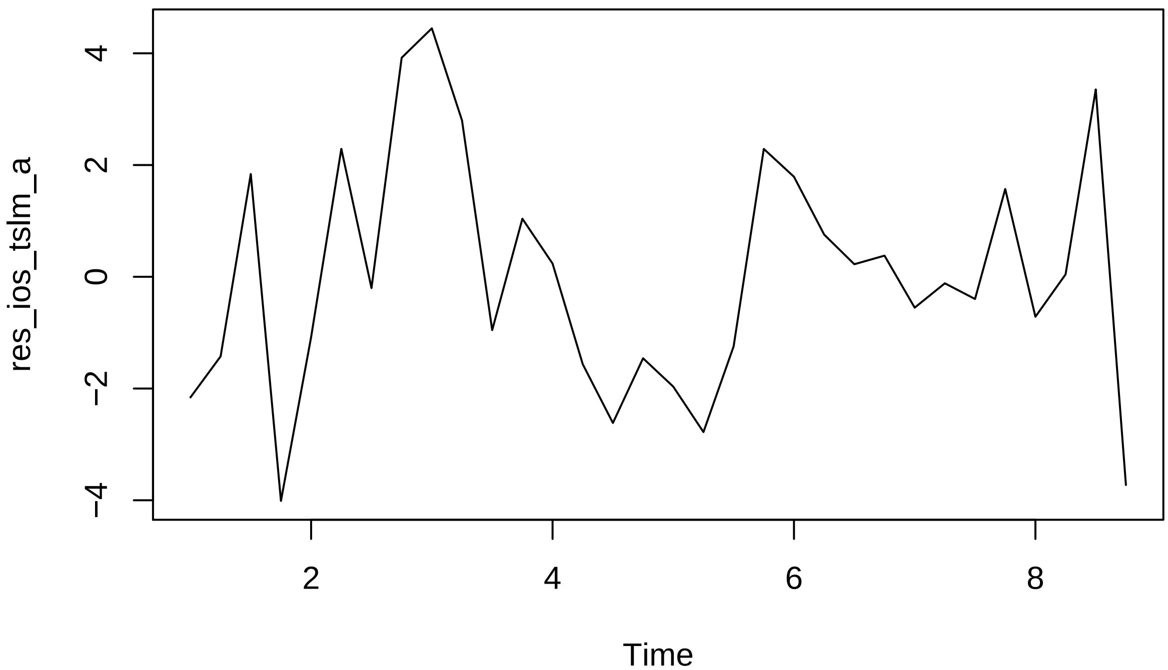
```
##
## Call:
## tslm(formula = ts_ios ~ trend + season + ts_android)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0108 -1.4331 -0.1595  1.6255  4.4468
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.29629    1.60123  10.802 4.15e-11 ***
## trend       -0.26178    0.09164  -2.856  0.00832 **
## season2     -2.18920    1.16483  -1.879  0.07144 .
## season3     -3.13395    1.17413  -2.669  0.01293 *
## season4      1.89300    1.16799   1.621  0.11714
```

```
## ts_android   0.05471    0.03786   1.445  0.16037
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.32 on 26 degrees of freedom
## Multiple R-squared:  0.5548, Adjusted R-squared:  0.4692
## F-statistic: 6.481 on 5 and 26 DF,  p-value: 0.0004906
```

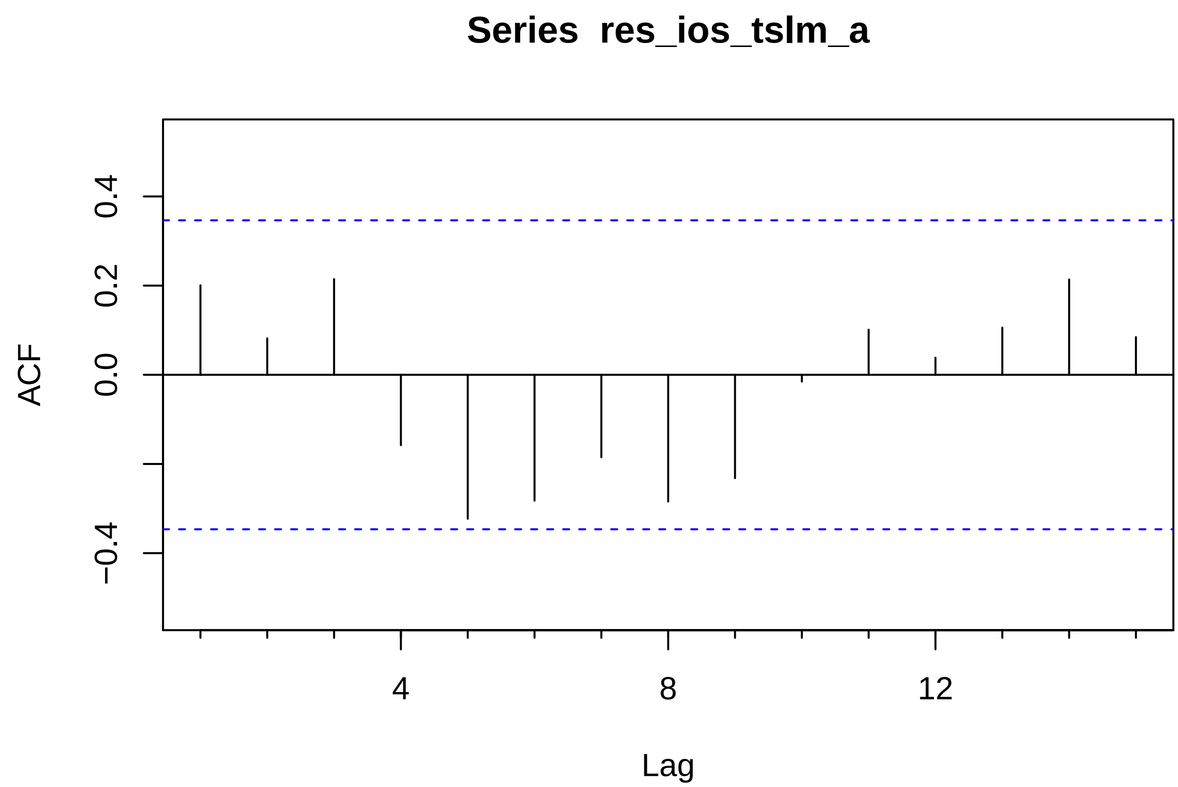
```
fit_ios_tslm_a<- fitted(ios_tslm_a)
plot(ts_ios, col=3)
lines(fit_ios_tslm_a, col=2)
```




```
res_ios_tslm_a<- residuals(ios_tslm_a)
plot(res_ios_tslm_a)
```

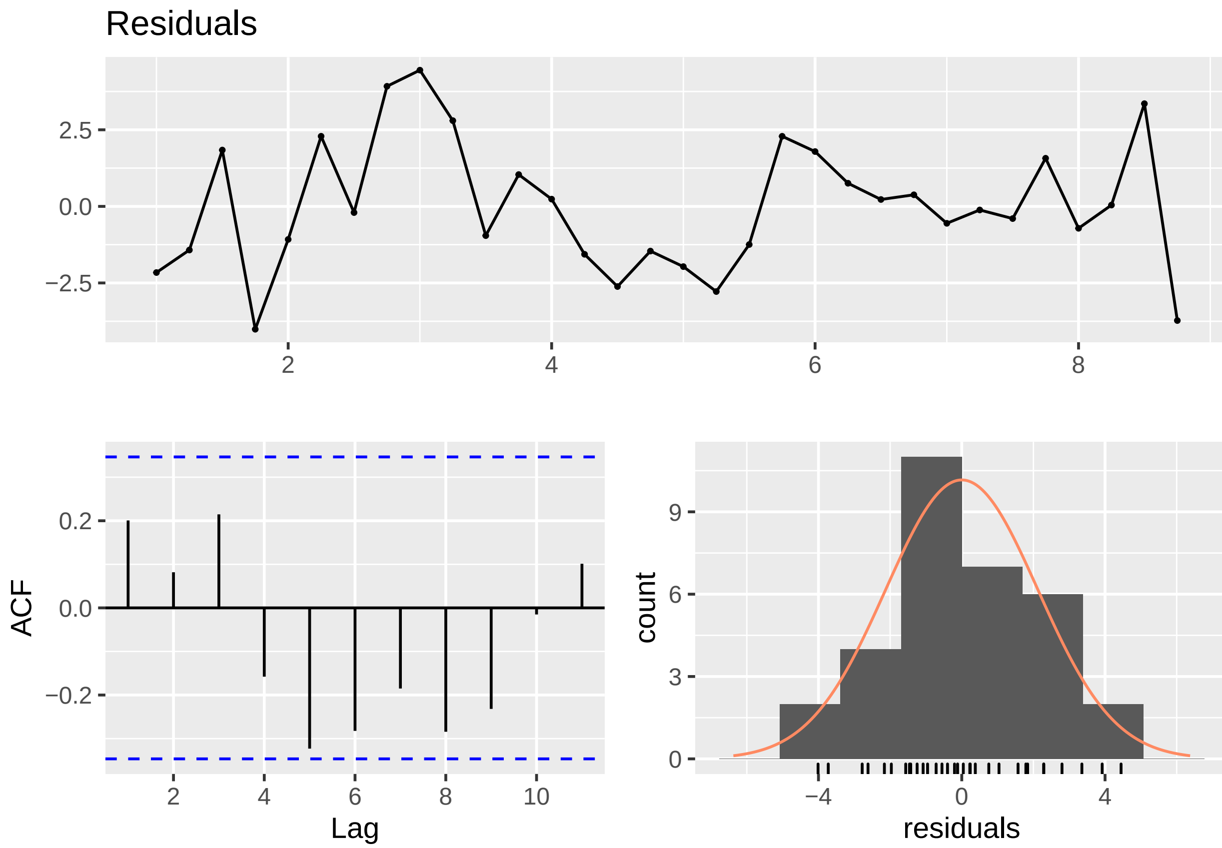



```
Acf(res_ios_tslm_a)
```




```
checkresiduals(res_ios_tslm_a)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



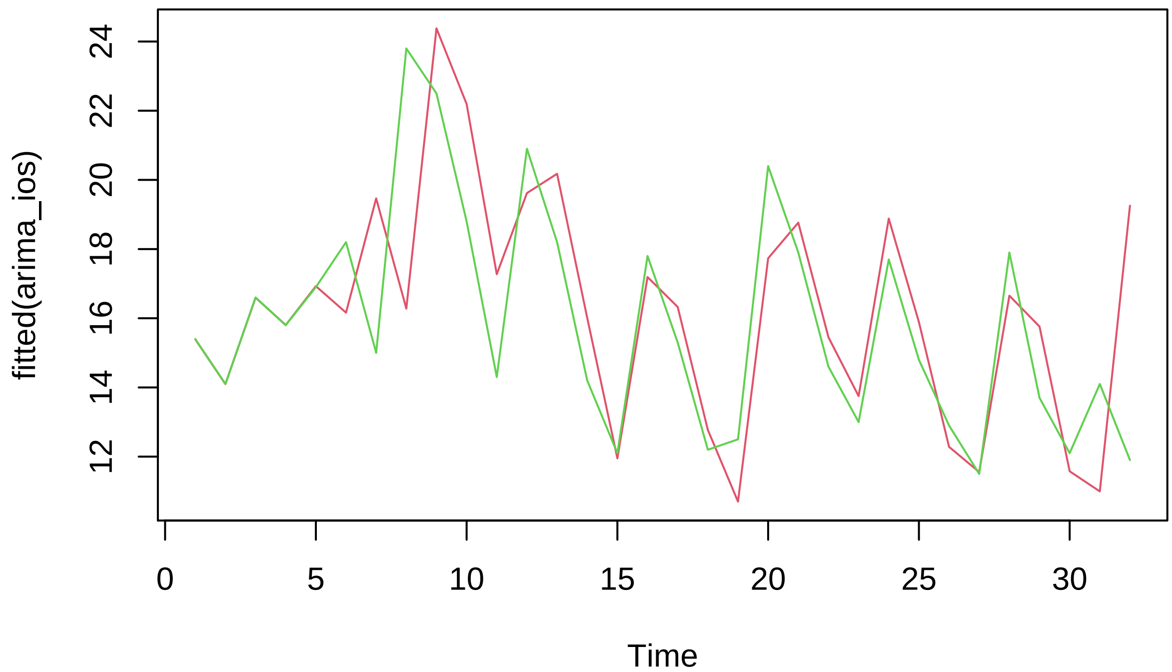
Taking into account the variable Android sales the model performs slightly better, showing again no auto-correlation on the residuals, but still evidencing the need for a more complex model.

ARIMA Model

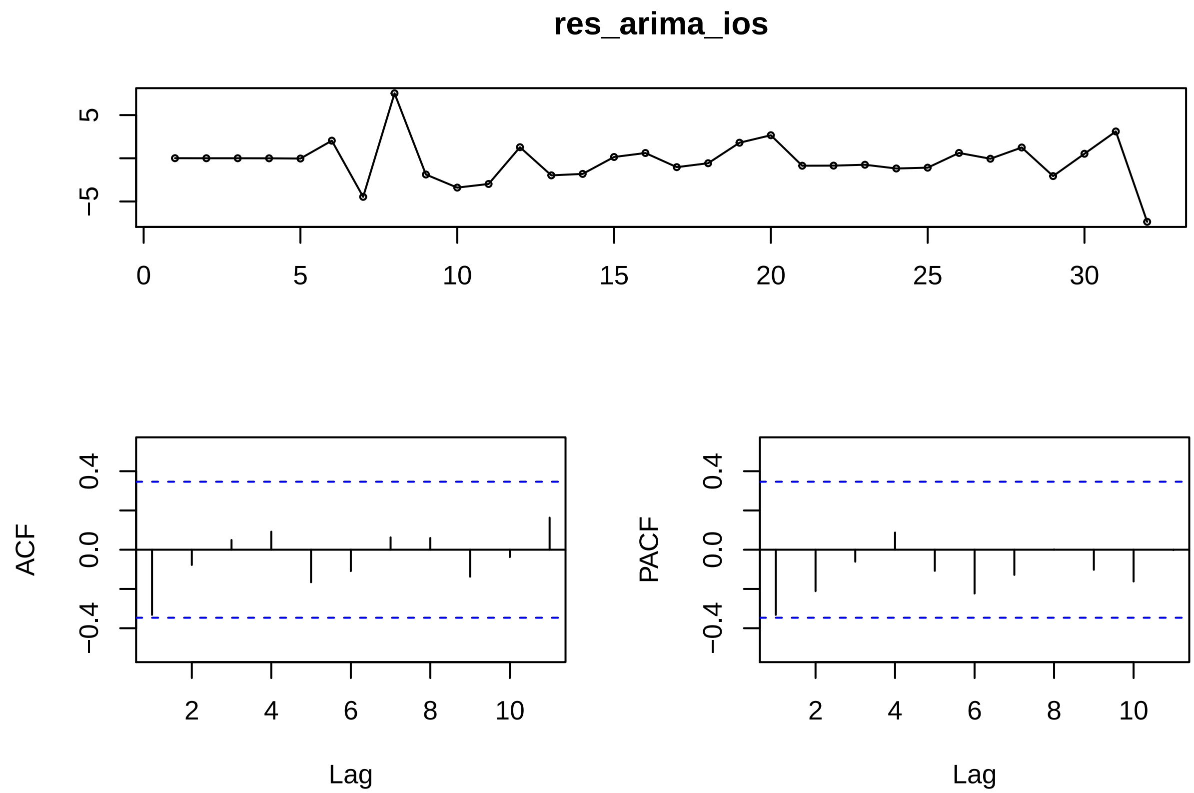
```r
arima_ios<- Arima(ios, order = c(0,1,0), seasonal = list(order=c(1,1,1), period=4))
arima_ios
```

```
## Series: ios
## ARIMA(0,1,0)(1,1,1)[4]
##
## Coefficients:
##         sar1     sma1
##       0.2274  -1.0000
## s.e.  0.3795   0.3765
##
## sigma^2 estimated as 8.012:  log likelihood=-68.68
## AIC=143.36   AICc=144.4   BIC=147.24
```

```r
plot(fitted(arima_ios), col=2)
lines(ios, col=3)
```

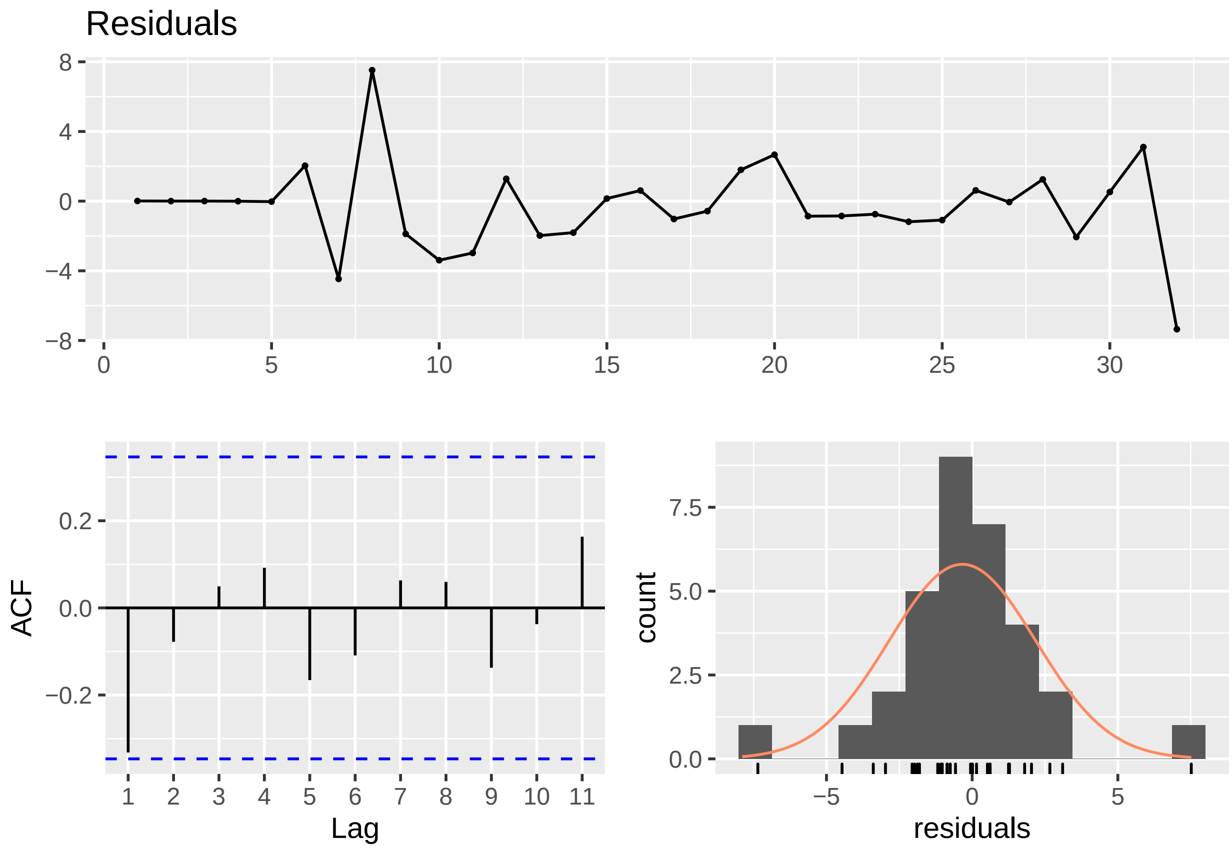



```
res_arima_ios<- residuals(arima_ios)
tsdisplay(res_arima_ios)
```




```
checkresiduals(res_arima_ios)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```

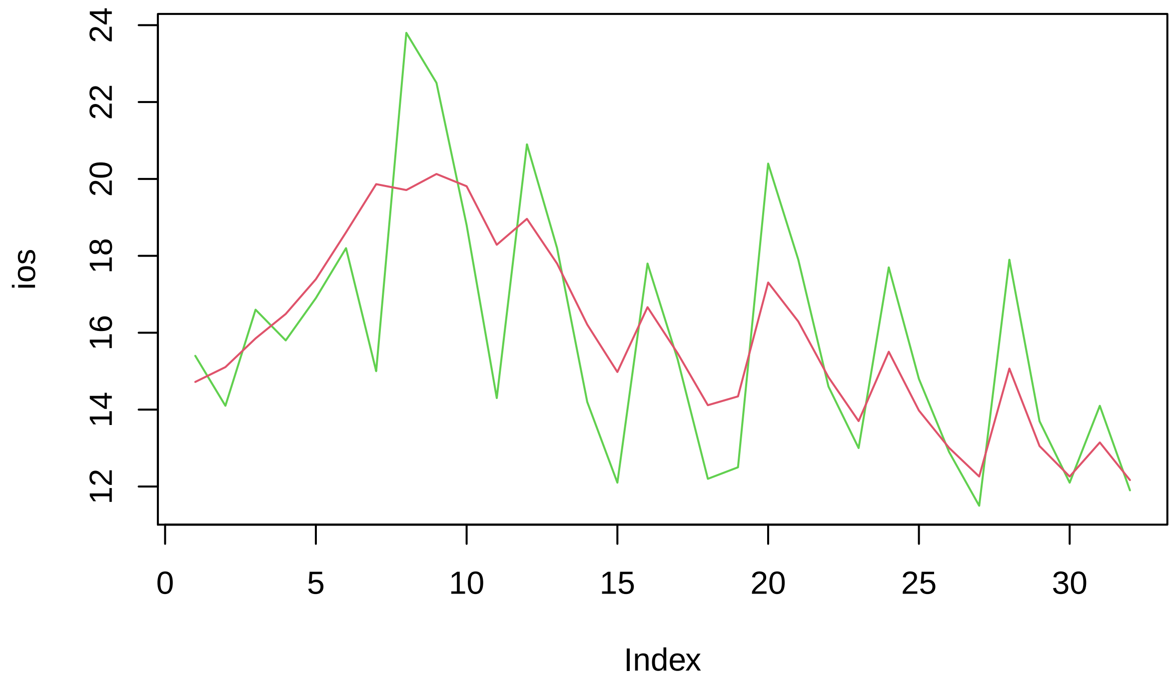


The ARIMA model performs fairly well, giving as a result a good approximation of the data and no apparent autocorrelation on the residuals. Still it is to be noted that it has difficulties in modelling the beginning of the time series.

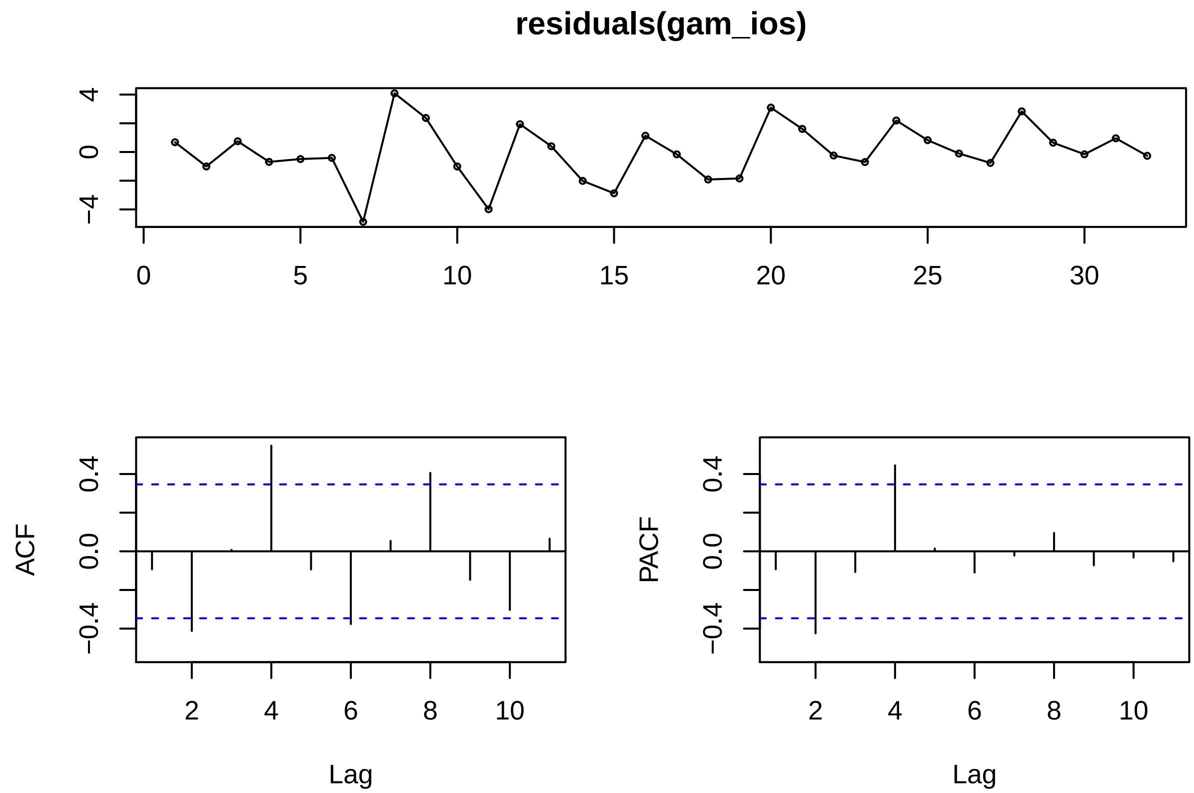
GAM

```
gam_ios <- gam(ios~s(android),arg=c("df=2","df=3","df=4"))

plot(ios, col=3, type="l")
lines(fitted(gam_ios), col=2)
```




```
tsdisplay(residuals(gam_ios))
```



The GAM model captures the general trend fairly well, while not fitting really well the dataset. The residuals shows some seasonality that could be the cause behind the performance of the model.

Conclusion

Given the models tested and the results obtained we have that the best model found is the $ARIMA(0,1,0)(1,1,1)_4$. The others have problems such as not being the right kind of model for the problem (non-linear problem vs linear model) or not being enough close to the dataset itself.