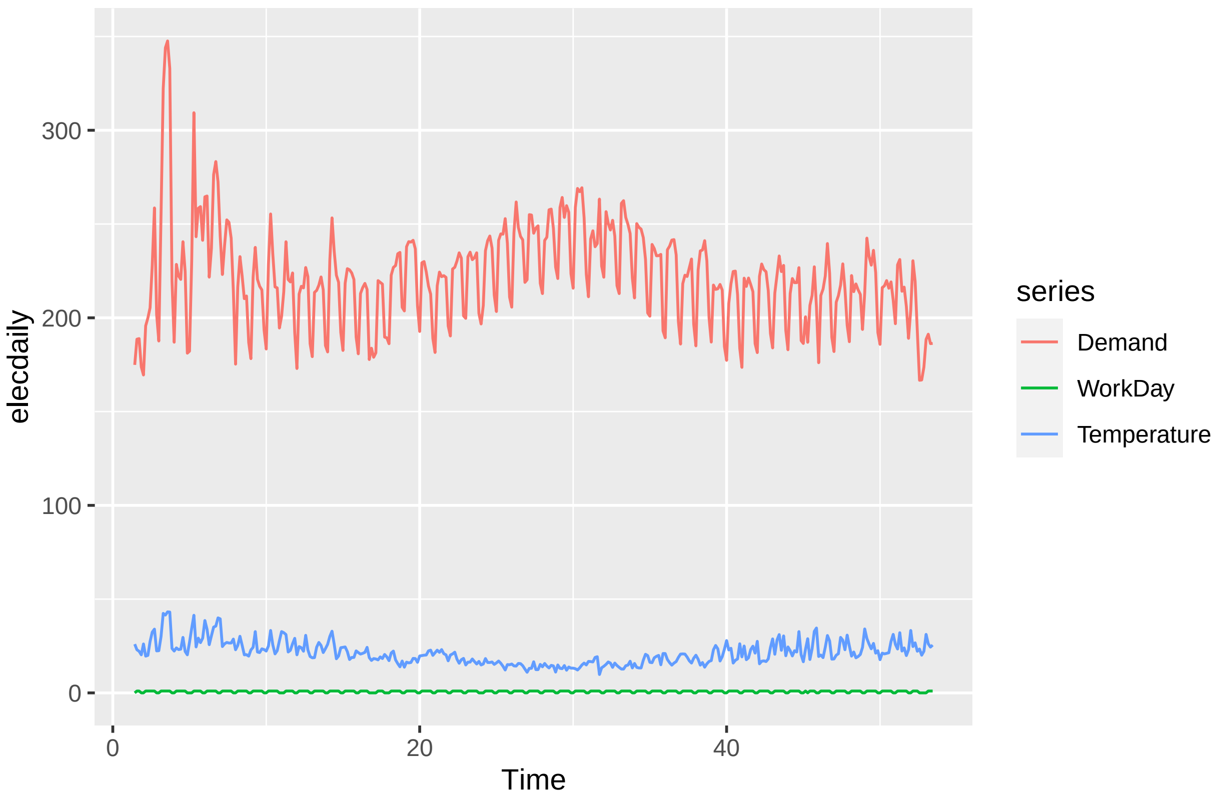
Electricity Demand

Julien JACQUES

2/25/2020

```
library(fpp2)
autoplot(elecdaily)
```



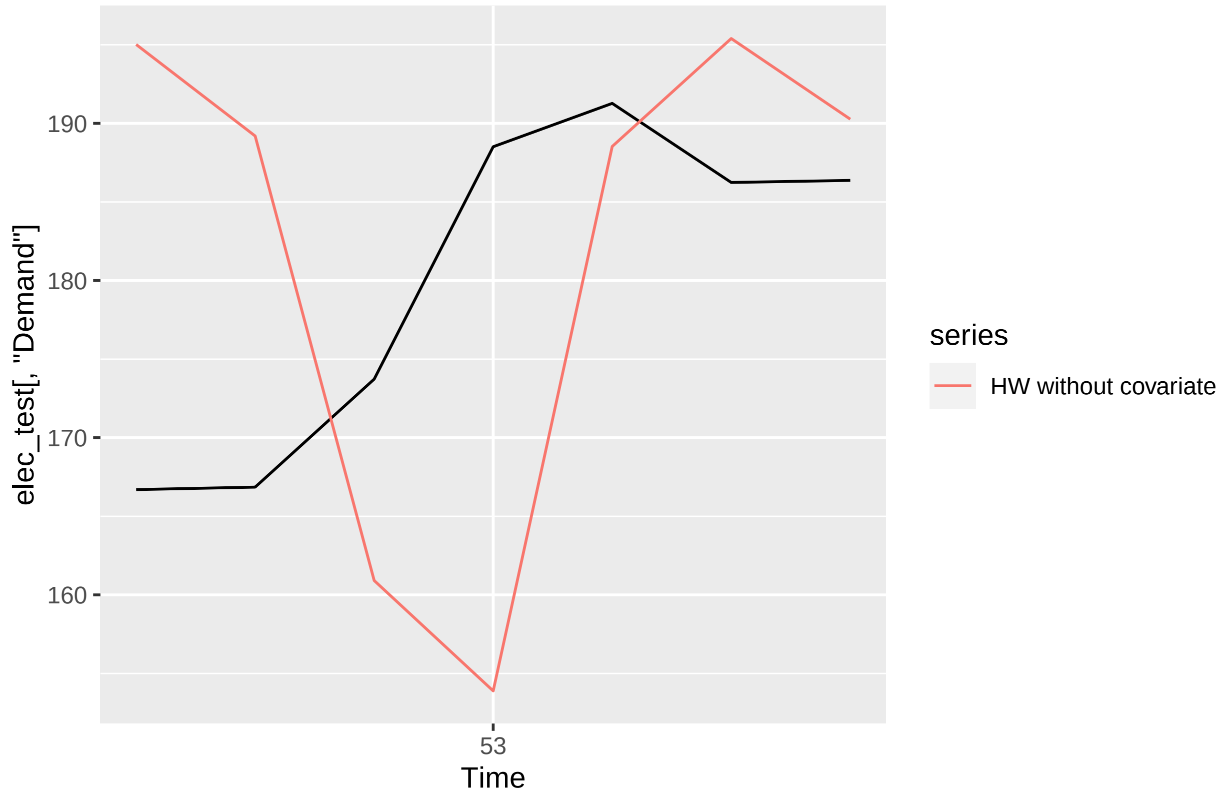
We split into train and test

```
elec_train=window(elecdaily,start=c(1,1),end=c(52,4))
elec_test=window(elecdaily,start=c(52,5),end=c(53,4))
```

Forecasting without covariates

First, we start with an HoltWinters exponential smoothing

```
fit=hw(elec_train[,"Demand"])
prev=forecast(fit,h=7)
autoplot(elec_test[,"Demand"])+autolayer(prev$mean,series="HW without covariate")
```

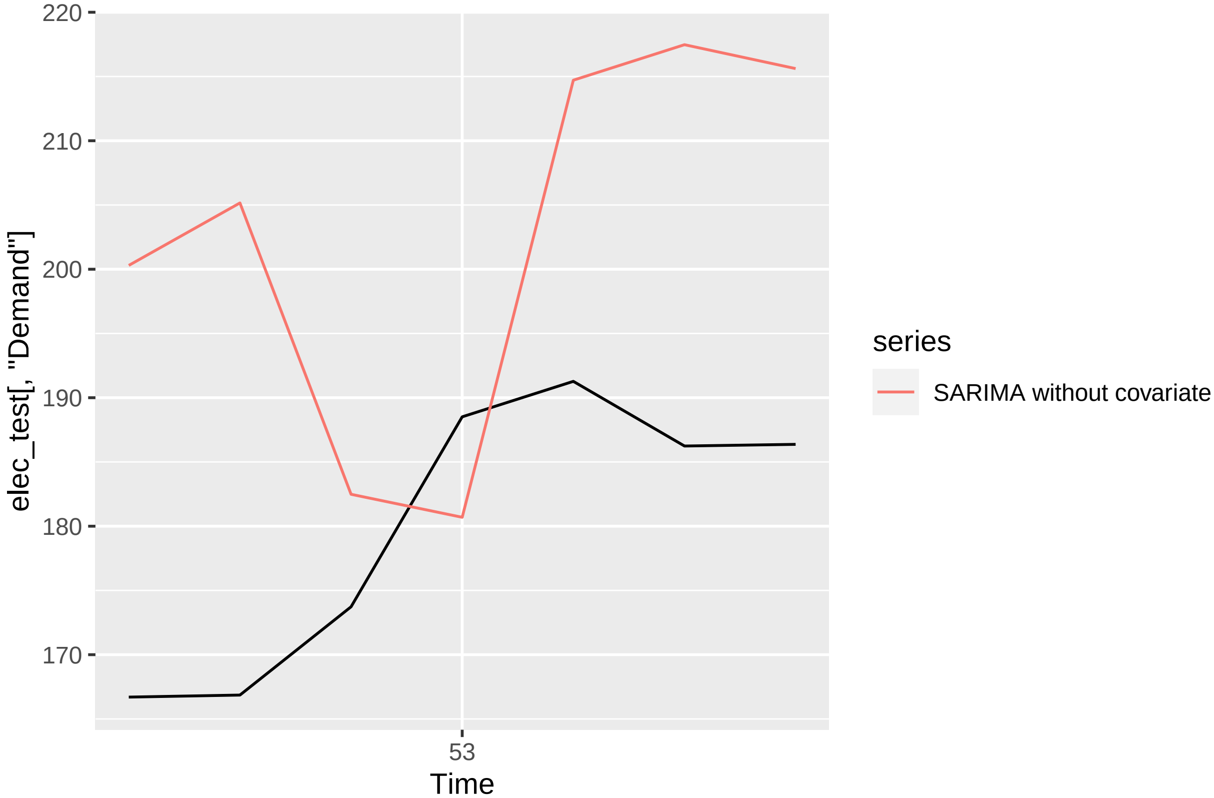



```
print(sqrt(mean((prev$mean-elec_test[,"Demand"])^2)))
```

```
## [1] 19.89454
```

and a SARIMA model:

```
fit=auto.arima(elec_train[,"Demand"])
prev=forecast(fit,h=7)
autoplot(elec_test[,"Demand"])+autolayer(prev$mean,series="SARIMA without covariate")
```



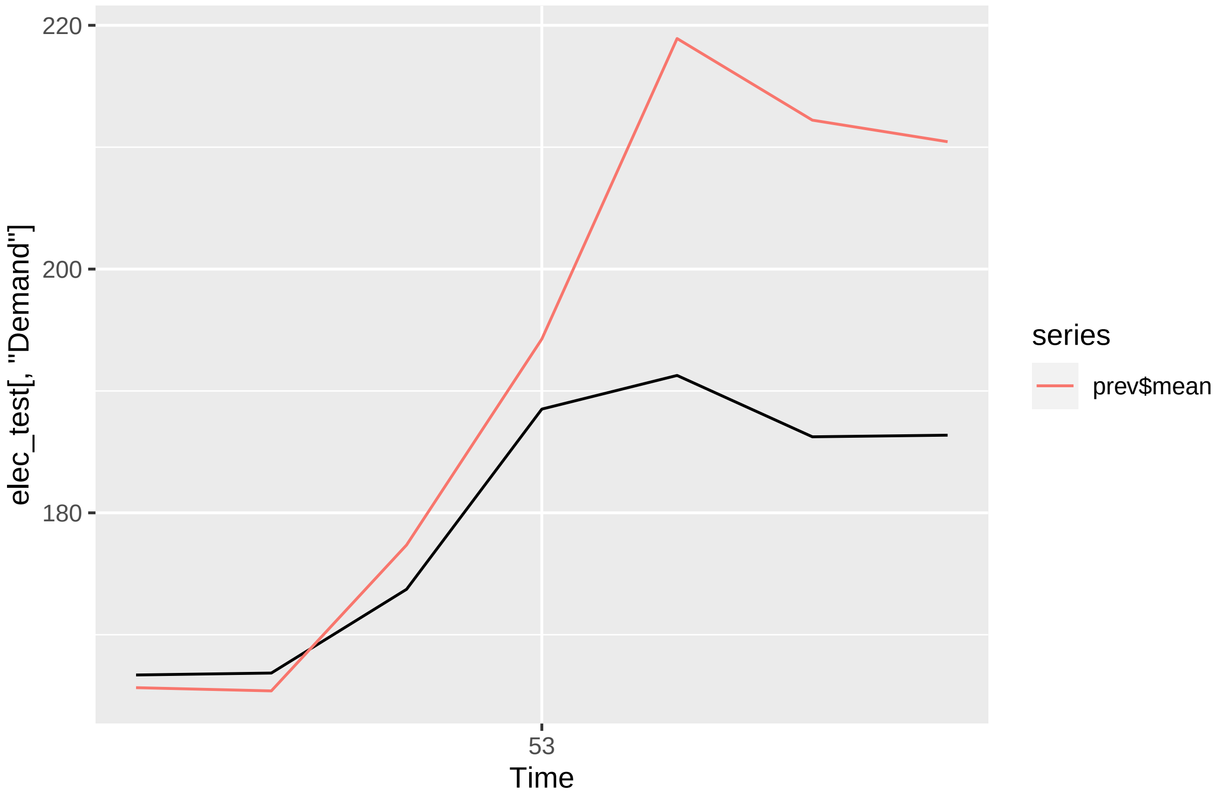

```
print(sqrt(mean((prev$mean-elec_test[,"Demand"])^2)))
```

```
## [1] 27.02949
```

Forecasting with covariates

We will use a dynamic regression model for forecasting electricity demand, using temperature and workday as external covariates. The order of the ARIMA model for the residual part is automaticaly selected

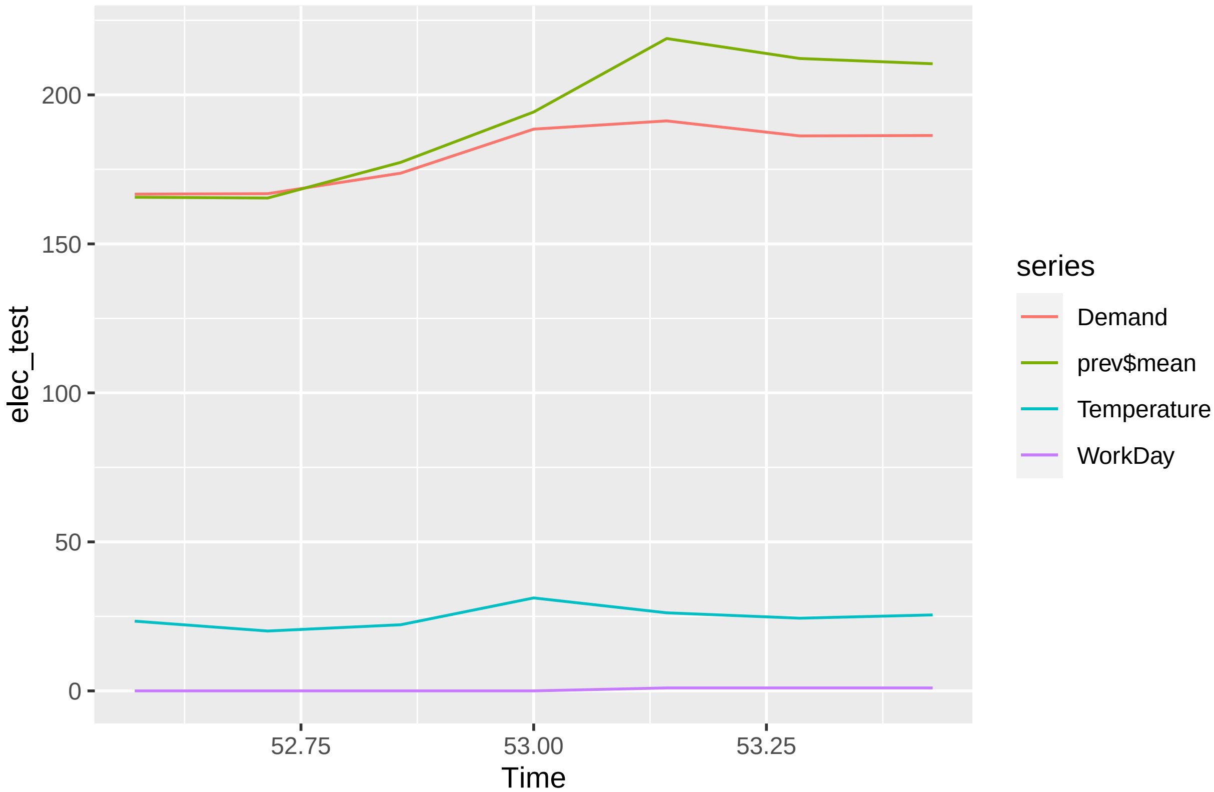
```
fit=auto.arima(elec_train[,"Demand"],xreg=elec_train[,2:3])
prev=forecast(fit,h=7,xreg=elec_test[,2:3])
autoplot(elec_test[,"Demand"])+autolayer(prev$mean)
```




```
print(sqrt(mean((prev$mean-elec_test[,"Demand"])^2)))
```

```
## [1] 17.18817
```

```
autoplot(elec_test)+autolayer(prev$mean)
```



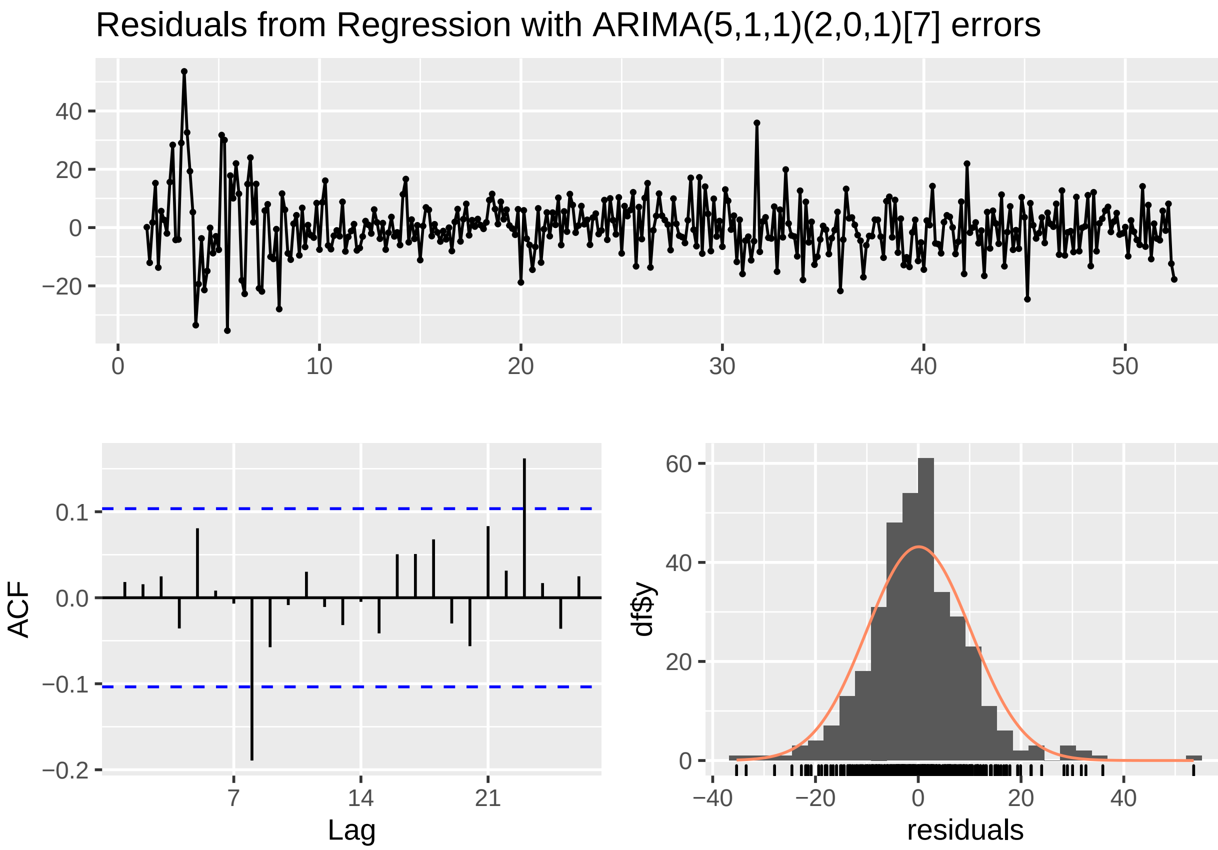
Using covariates allows us to improve the forecasting.

But if we check the residual, there is still some autocorrelations:

```
summary(fit)
```

```
## Series: elec_train[, "Demand"]
## Regression with ARIMA(5,1,1)(2,0,1)[7] errors
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5      ma1     sar1    sar2
##       0.9003  -0.2229  0.0952  -0.1140  0.1567  -0.9625  -0.5076  0.1984
## s.e.  0.0687   0.0791  0.0752   0.0723  0.0597   0.0319   0.2783  0.0599
##         sma1  WorkDay  Temperature
##       0.6309  31.3165       1.4803
## s.e.  0.2813   1.3671       0.1422
##
## sigma^2 estimated as 106.3:  log likelihood=-1334.54
## AIC=2693.08   AICc=2693.99   BIC=2739.62
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.09565789 10.13746 7.353345 -0.1447896 3.283934 0.5146259
##                    ACF1
## Training set 0.0182663
```

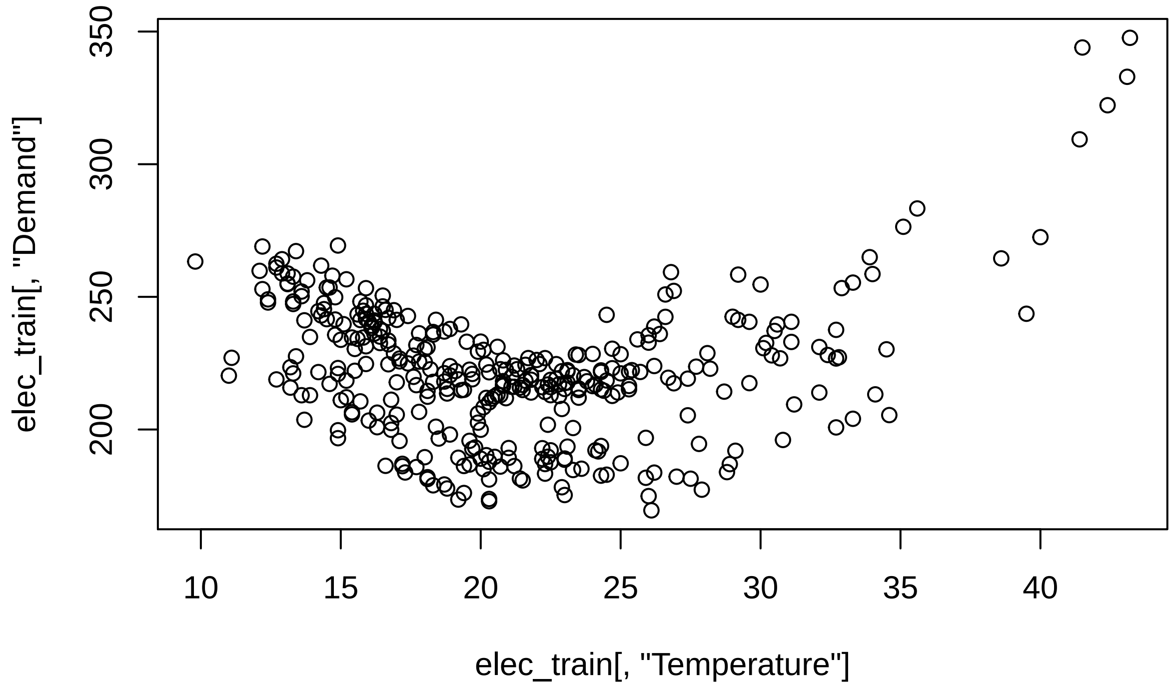
```
checkresiduals(fit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(5,1,1)(2,0,1)[7] errors
## Q* = 18.529, df = 3, p-value = 0.0003422
##
## Model df: 11.   Total lags used: 14
```

We can try to find a better model manually. Let's have a look to the relationship between Demand and Temperature

```
plot(elec_train[,"Temperature"],elec_train[,"Demand"])
```



The link seems to be quadratic. We introduced $Temperature^2$ in the model.

```
elec_train=cbind(Demand=elec_train[,1],WorkDay=elec_train[,2],Temp=elec_train[,3],SquareTemp=elec_train
```

Let's start by removing the effect of covariate.

```
fit2=tslm(Demand~WorkDay+Temp+SquareTemp+trend+season,data=elec_train)
summary(fit2)
```

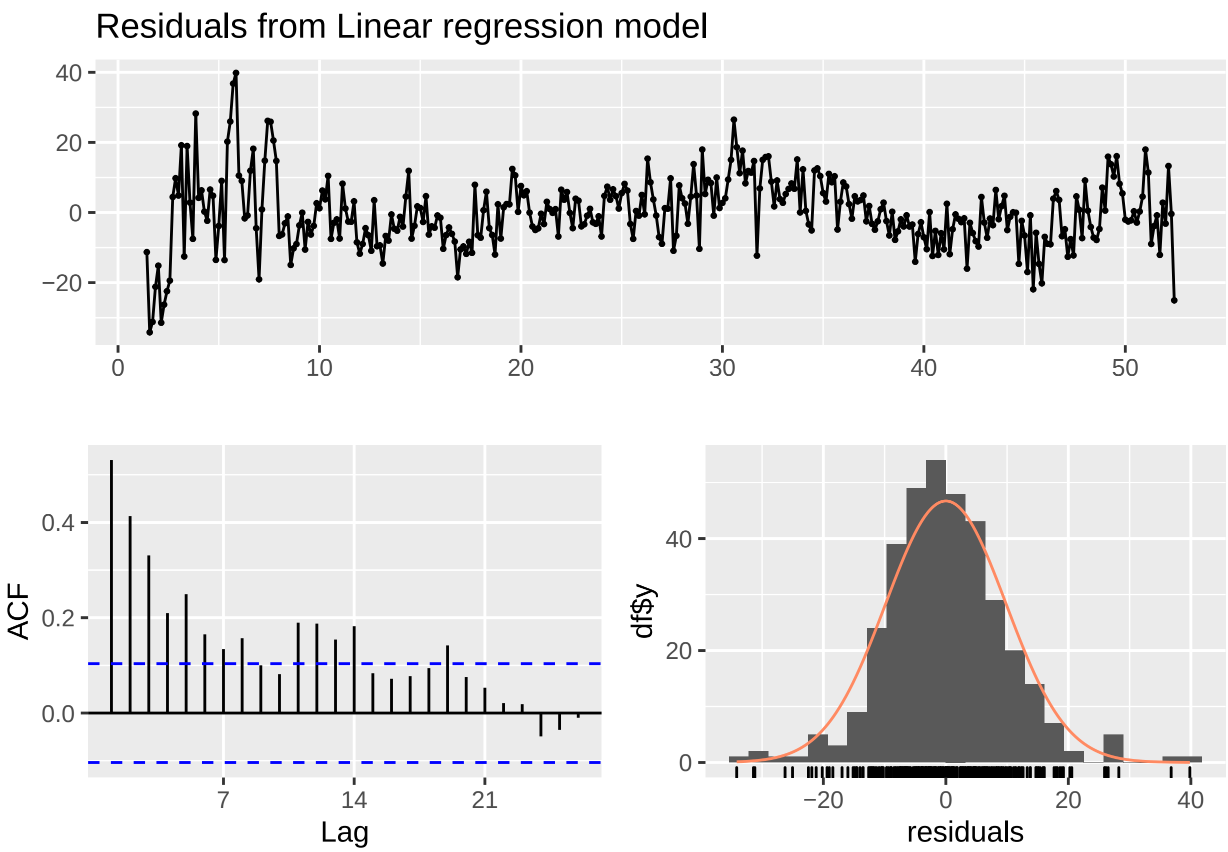
```
##
## Call:
## tslm(formula = Demand ~ WorkDay + Temp + SquareTemp + trend +
##     season, data = elec_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.149  -6.200  -0.524   5.293  39.840
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 357.217153   5.704688  62.618  < 2e-16 ***
## WorkDay      38.891730   3.650921  10.653  < 2e-16 ***
## Temp        -14.849365   0.472118 -31.453  < 2e-16 ***
## SquareTemp    0.315740   0.009600  32.889  < 2e-16 ***
## trend        -0.015952   0.005301  -3.009  0.00281 **
## season2       0.878590   3.902562   0.225  0.82201
## season3       0.850769   4.091306   0.208  0.83539
## season4       1.621769   4.085401   0.397  0.69164
## season5       1.171490   4.151346   0.282  0.77796
## season6      -1.909570   4.025997  -0.474  0.63558
## season7       9.160792   1.976629   4.635 5.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 9.977 on 347 degrees of freedom
## Multiple R-squared:  0.8593, Adjusted R-squared:  0.8552
## F-statistic: 211.9 on 10 and 347 DF,  p-value: < 2.2e-16
```

All the feature seems significant.

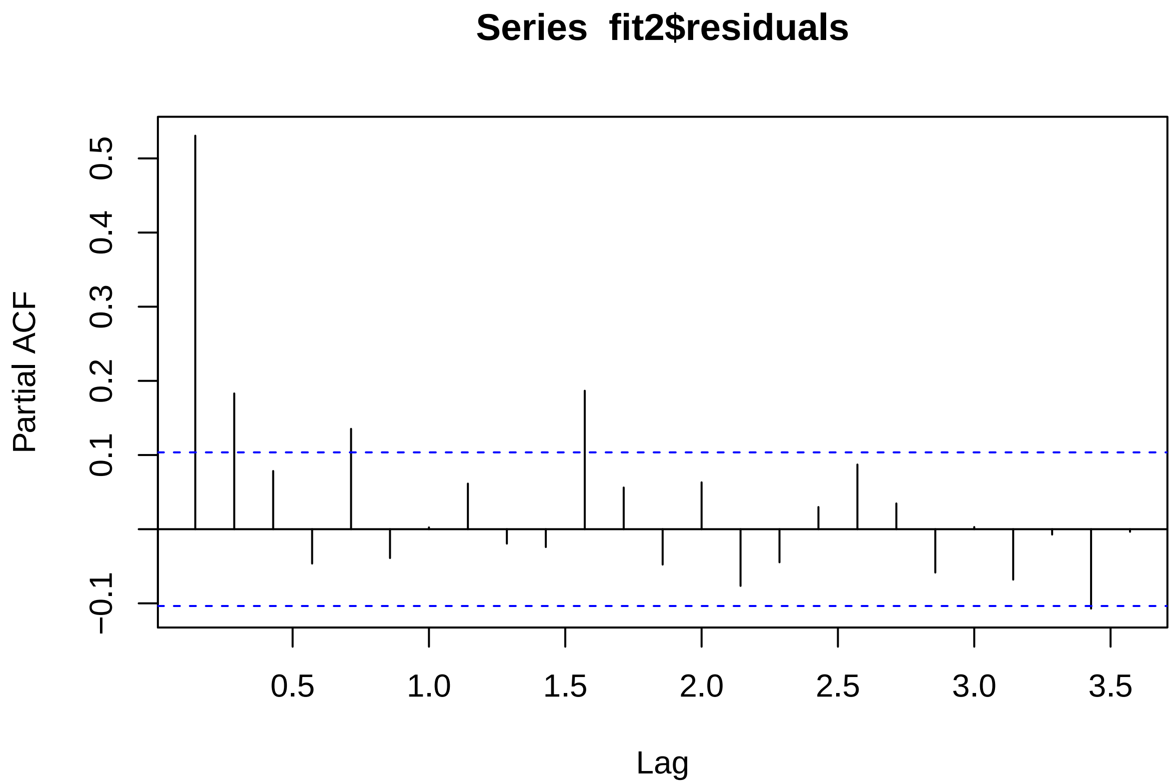
Let's now have a look to the residual

```
checkresiduals(fit2)
```



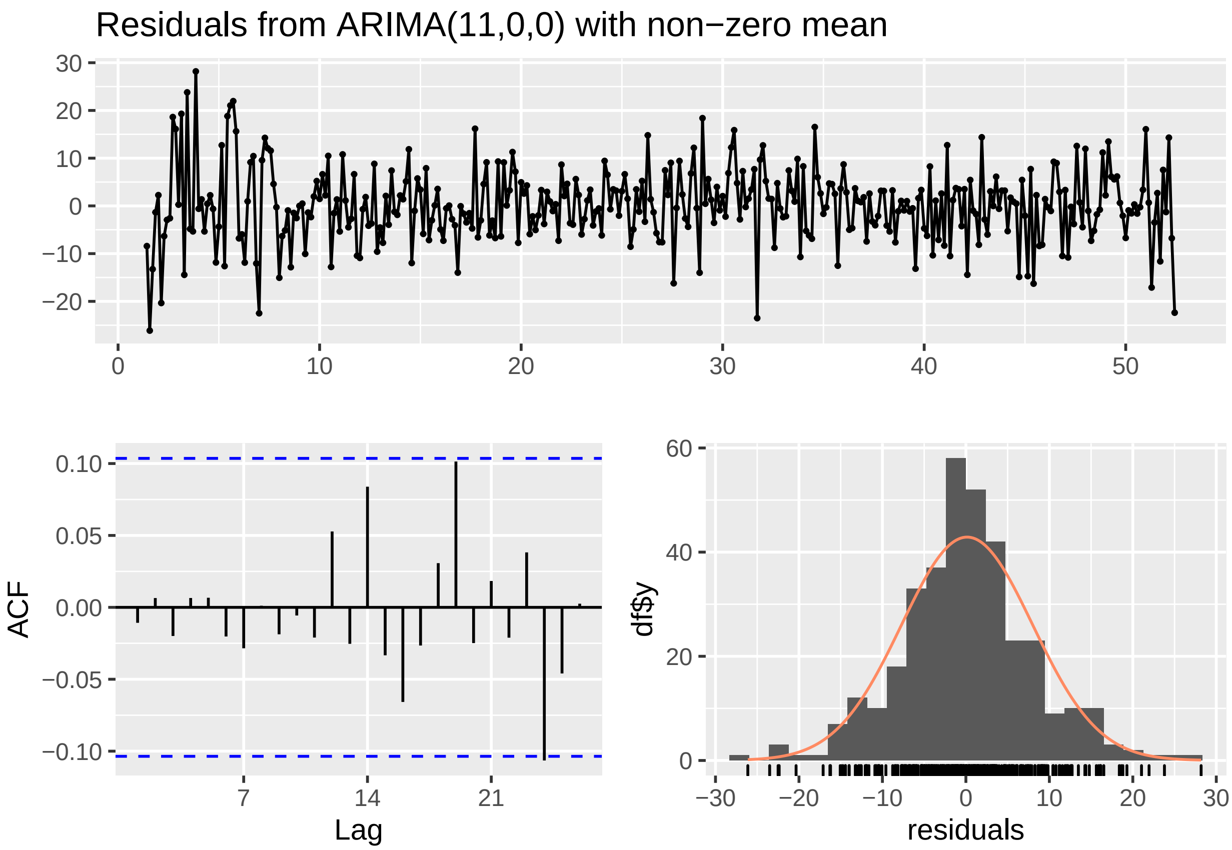

```
##
##  Breusch-Godfrey test for serial correlation of order up to 14
##
## data:  Residuals from Linear regression model
## LM test = 138.73, df = 14, p-value < 2.2e-16
```

```
plot(pacf(fit2$residuals))
```



The ACP and PACF look like those of an AR_{11} model: exponential deacrease of the ACF and significant PCA at lag 11. We can test it:

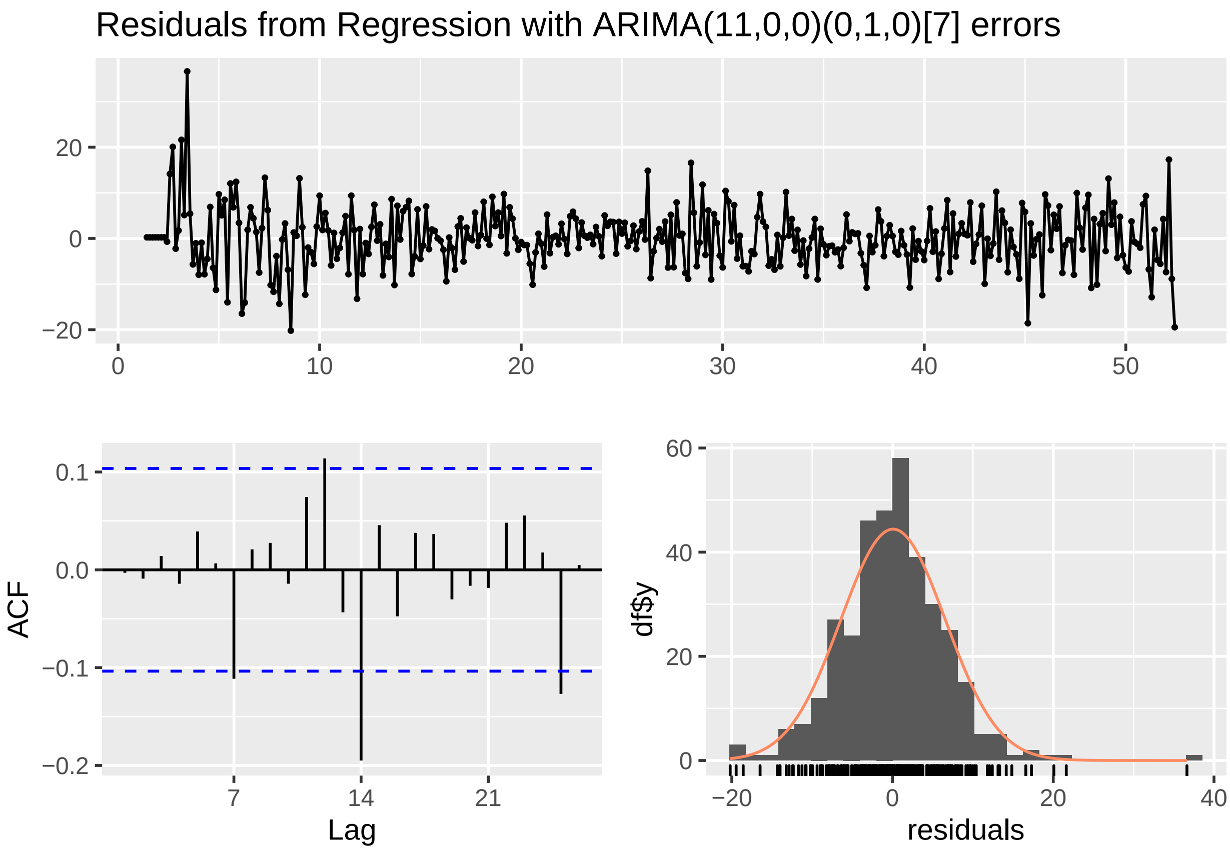
```
tmp=fit2$residuals
fit3=Arima(tmp,order=c(11,0,0))
checkresiduals(fit3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(11,0,0) with non-zero mean
## Q* = 5.3173, df = 3, p-value = 0.15
##
## Model df: 12.   Total lags used: 15
```

The correspondant residuals are uncorrelated. We can now go back to the initial series, and try to propose the following model (where `seasonal=c(0,1,0)` is for taking into account season+trend):

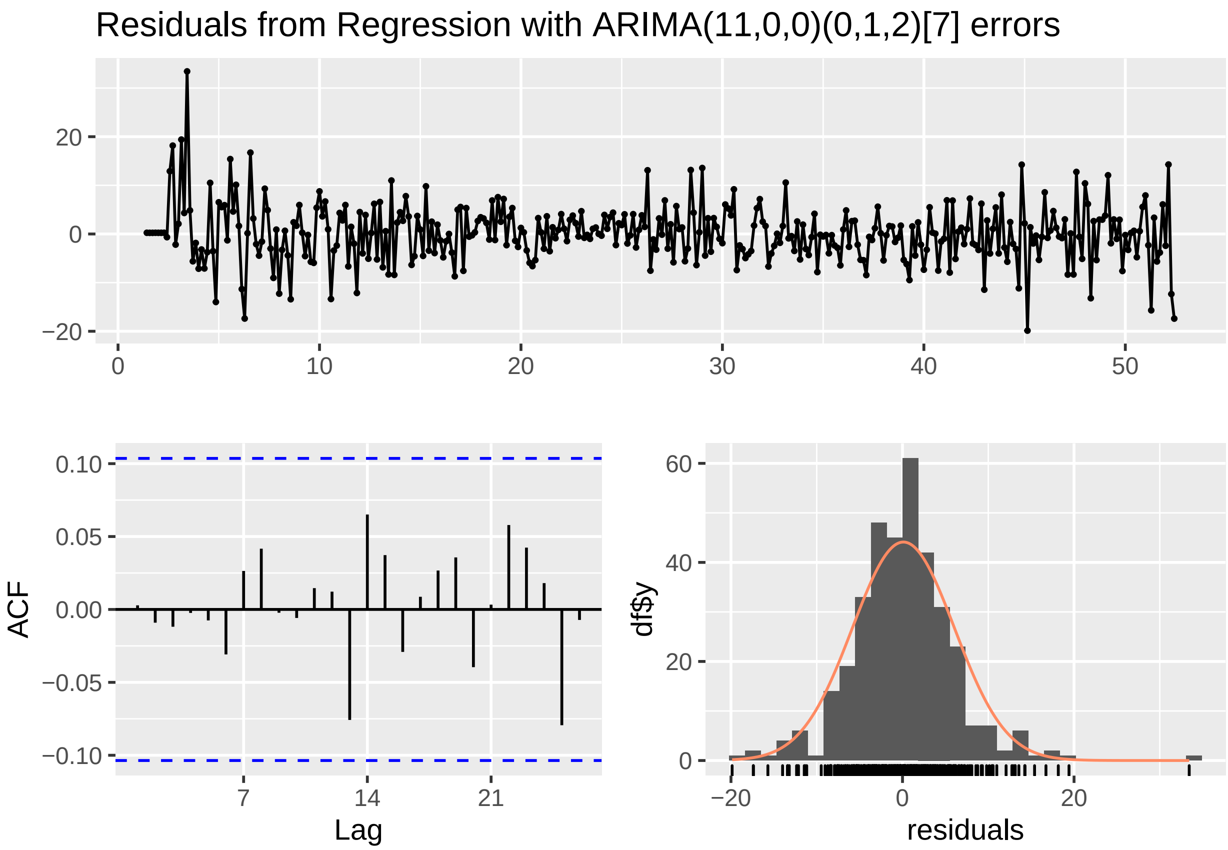
```
fit=Arima(elec_train[,"Demand"],xreg=elec_train[,2:4],order=c(11,0,0),seasonal = c(0,1,0))
checkresiduals(fit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(11,0,0)(0,1,0)[7] errors
## Q* = 29.824, df = 3, p-value = 1.503e-06
##
## Model df: 14.   Total lags used: 17
```

Residual have significant ACF at periodic lag (14). We will add a second order MA in the seasonal pattern:

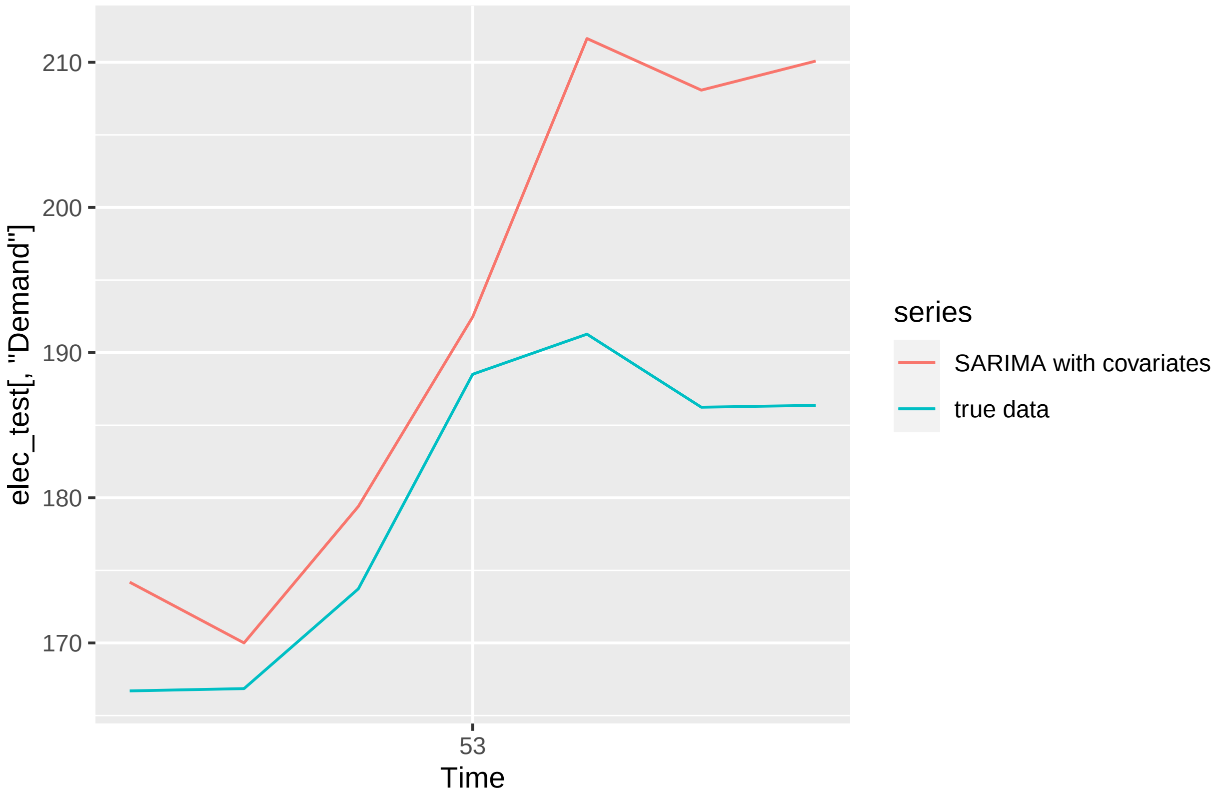
```
fit=Arima(elec_train[,"Demand"],xreg=elec_train[,2:4],order=c(11,0,0),seasonal = c(0,1,2))
checkresiduals(fit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(11,0,0)(0,1,2)[7] errors
## Q* = 6.8553, df = 3, p-value = 0.07666
##
## Model df: 16.   Total lags used: 19
```

Now residuals are uncorrelated. We can then perform forecasting:

```
elec_test=cbind(Demand=elec_test[,1],WorkDay=elec_test[,2],Temp=elec_test[,3],SquareTemp=elec_test[,3]^
prev=forecast(fit,h=7,xreg=elec_test[,2:4])
autoplot(elec_test[,"Demand"], series="true data")+autolayer(prev$mean,series="SARIMA with covariates")
```



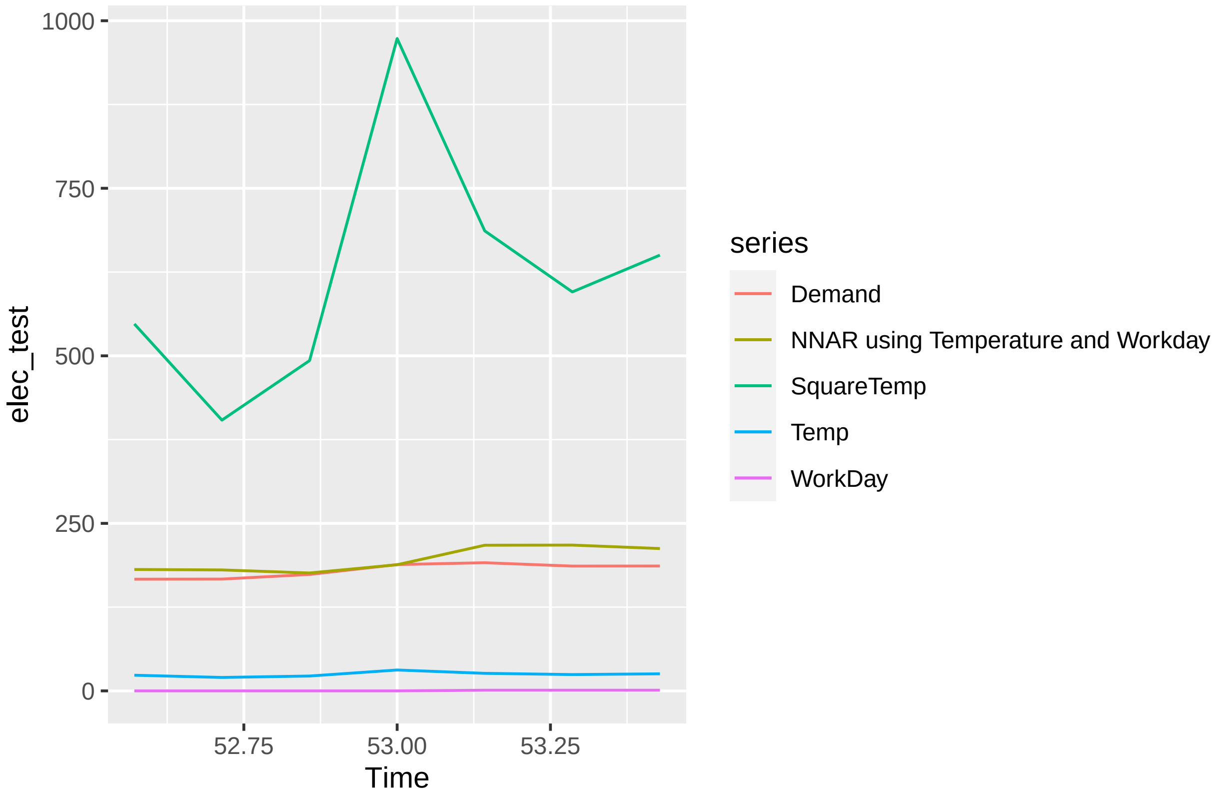

```
print(sqrt(mean((prev$mean-elec_test[,"Demand"])^2)))
```

```
## [1] 14.96637
```

The result are better than those obtained with the auto.arima function.

Finally, we can compare with a NNAR model with covariates, but it does not improve the forecast.

```
fit=nnetar(elec_train[,"Demand"],xreg=elec_train[,2:4])
prev=forecast(fit,h=7,xreg=elec_test[,2:4])
autoplot(elec_test)+autolayer(prev$mean,series="NNAR using Temperature and Workday")
```




```
print(sqrt(mean((prev$mean-elec_test[,"Demand"])^2)))
```

```
## [1] 19.79481
```