

Section 6: Time series plots

Review of important functions covered in this section:

Functions

time	Generates a vector at which time series was sampled
cycle	Extracts position in the cycle of each observation
start	Extracts the time at which the first observation was sampled
end	Extracts the time at which the last observation was sampled
frequency	Extracts number of samples per unit
tsp	Extracts start, end, frequency
ts	Create time series object
ts.plot	Plots ts objects
window	Extracts time window
acf	Extracts auto-correlation function
pacf	Extracts partial auto-correlation function
zoo	Creates ordered observations (zoo package)
ets	Exponential smoothing state space model (forecast package)
tsdiag	Shows diagnostic plots for time series fits
forecast	Forecasts time series (forecast package)
auto.arima	Fits ARIMA model (forecast package)
coredata	Extracts core data of objects (zoo package)
index	Extracts index of objects (zoo package)
stl	Forecast of stl objects or seasonal data decomposition (forecast and stats package)
stlf	Forecasts stl objects (forecast package)
monthplot	Seasonal plot
seasonplot	Seasonal plot (forecast package)

Exercise

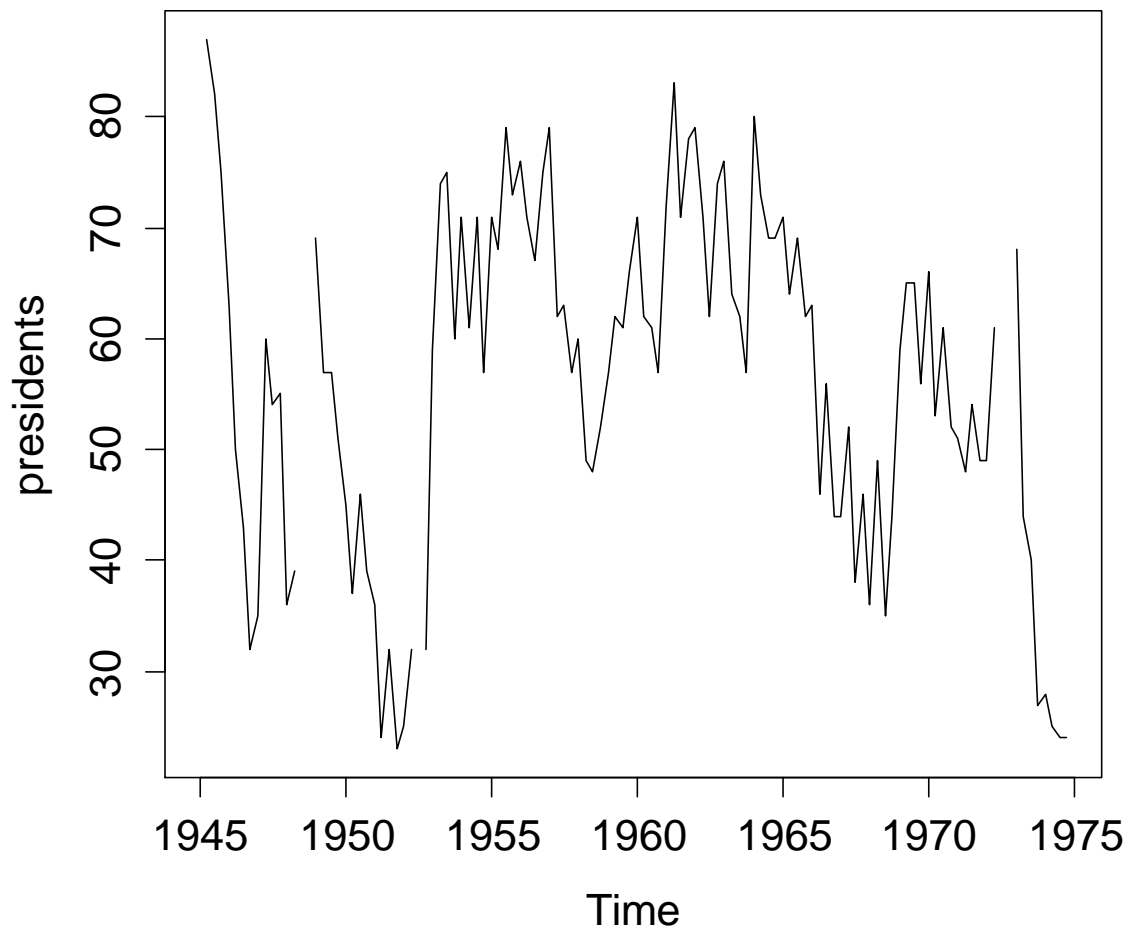
We'll use `presidents` data.

- Check the class of `presidents` data.

```
class(presidents)
```

b. Draw a simple plot of this time series data.

```
plot(presidents)  
ts.plot(presidents)
```



c. Extract the position of each observations of the dataset in the cycle.

```
cycle(presidents)
```

d. Extract the position of the first observation of the dataset in the cycle.

```
start(presidents)
```

e. Extract the position of the last observation of the dataset in the cycle.

```
end(presidents)
```

f. Extract the frequency of the observations of the dataset in the cycle.

```
frequency(presidents)
```

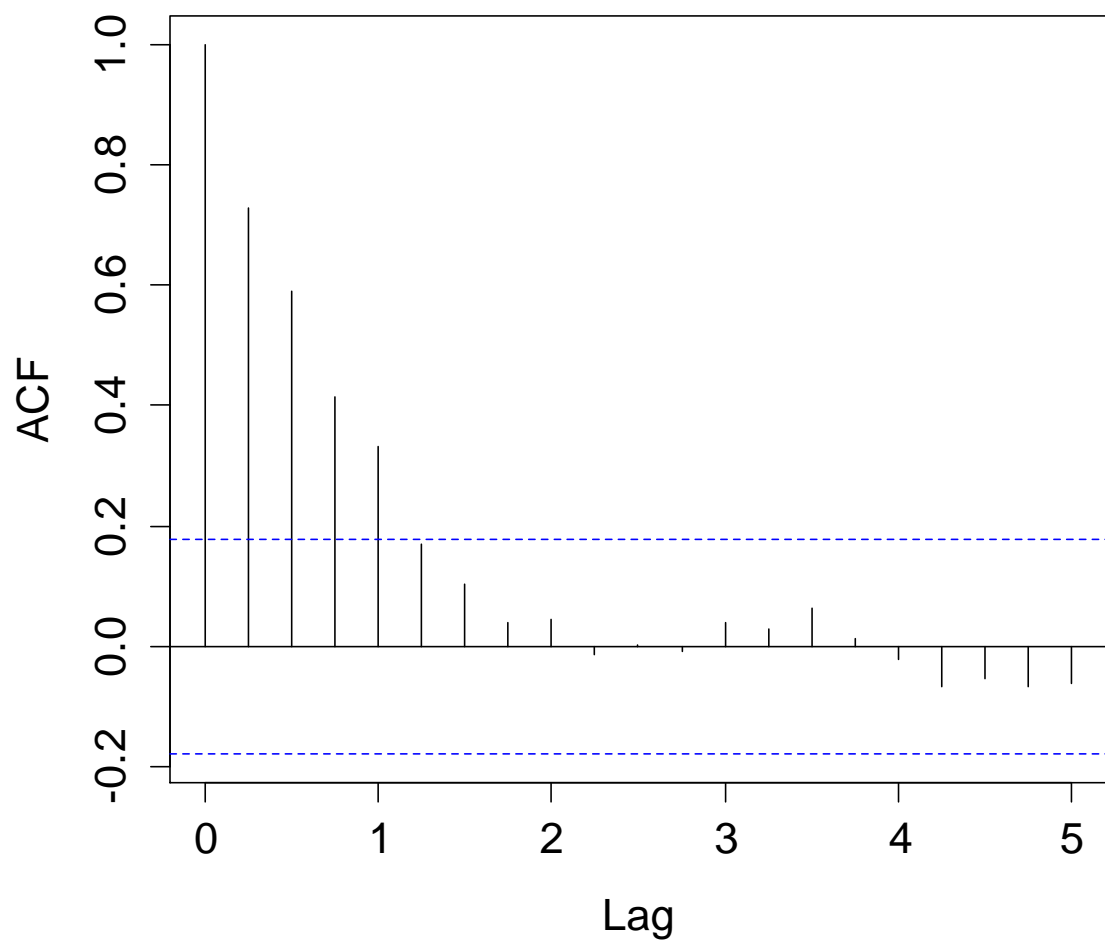
- g. Extract the summary of the observations of the dataset in the cycle (start, end, frequency).

```
tsp(presidents)
```

- h. In the dataset, replace all the missing values by the number 70. Now plot an auto-correlation function.

```
acf(presidents)
```

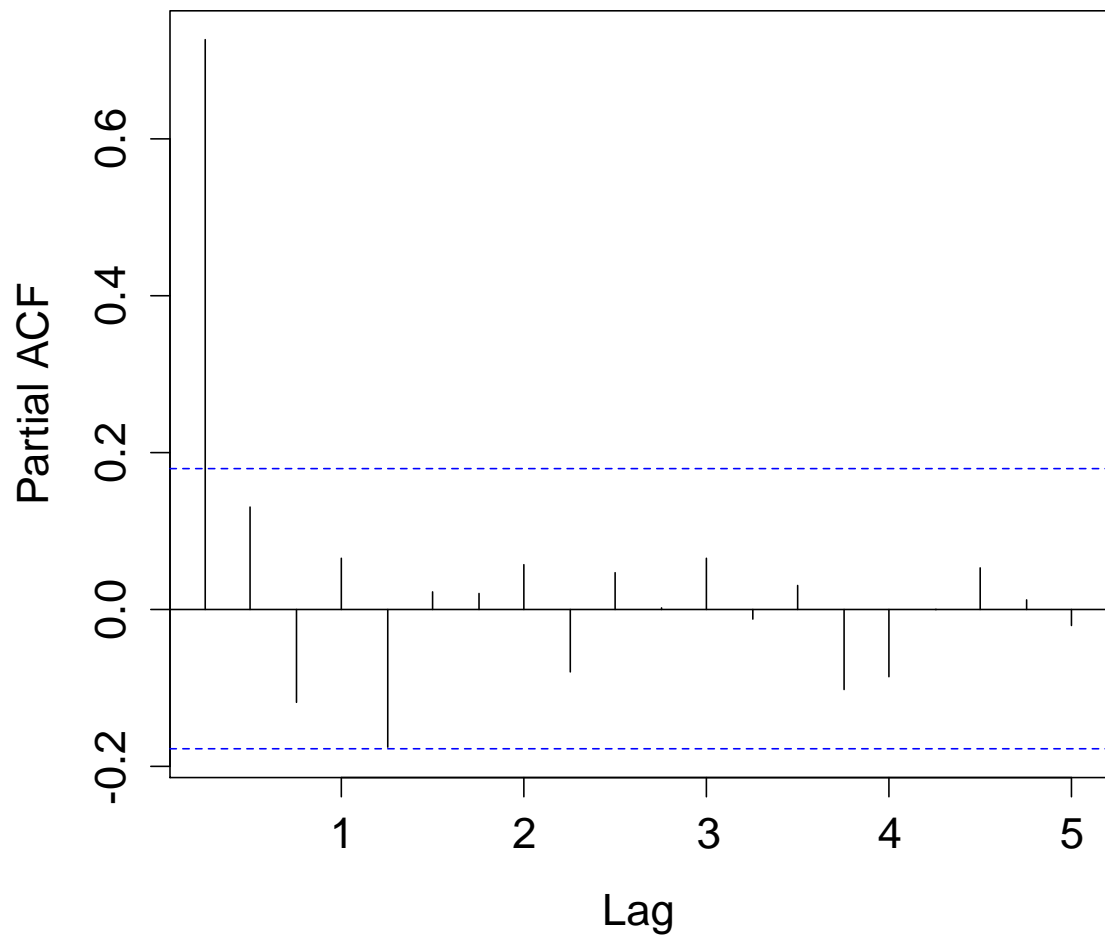
Series presidents



- i. Now plot a partial auto-correlation function.

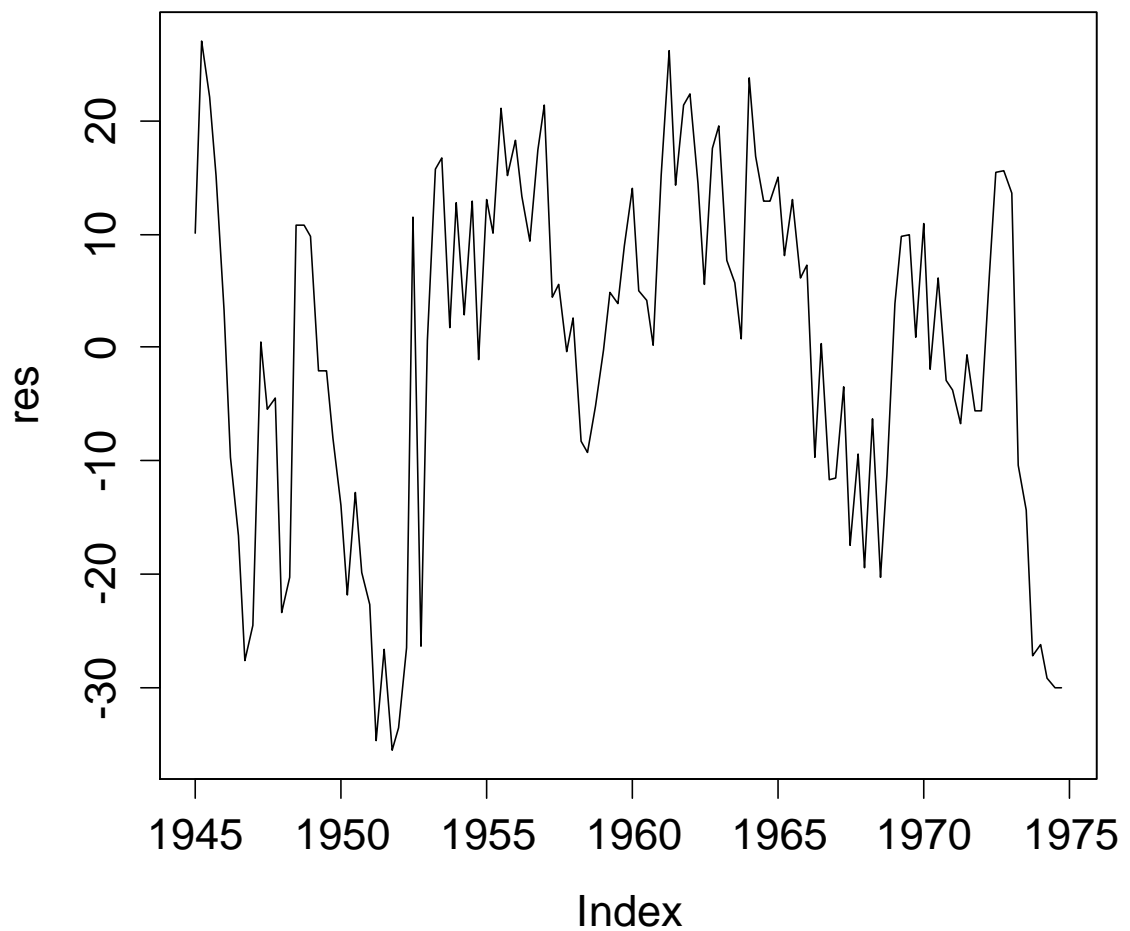
```
pacf(presidents)
```

Series presidents



- j. Using zoo package, fit a linear model of core data and index data and plot the ordered residuals.

```
install.packages("zoo")
require(zoo)
fit = lm(coredata(presidents) ~ index(presidents))
res = zoo(resid(fit), index(presidents))
plot(res)
```

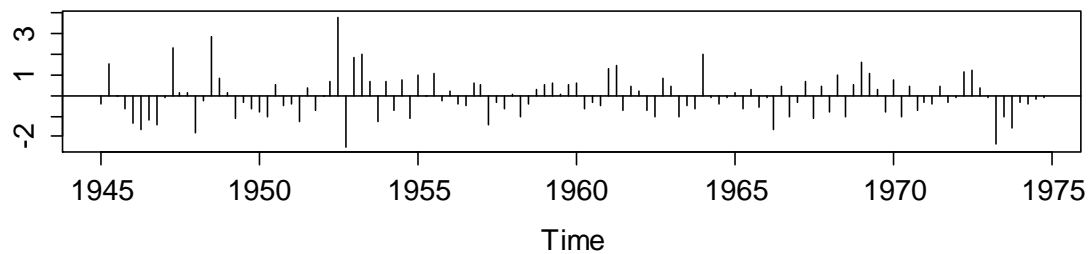


- k. Using forecast package, fit an exponential smoothing state space model.

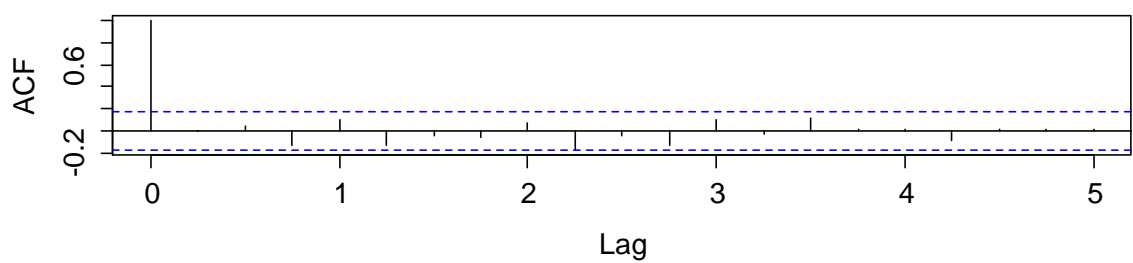
```
install.packages("forecast")
require(forecast)
fite = ets(presidents)
```

I. Check diagnostic of the above model.
`tsdiag(fite)`

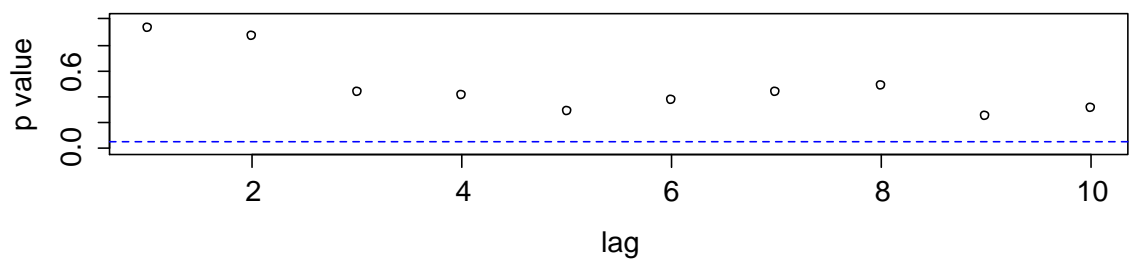
Standardized Residuals



ACF of Residuals

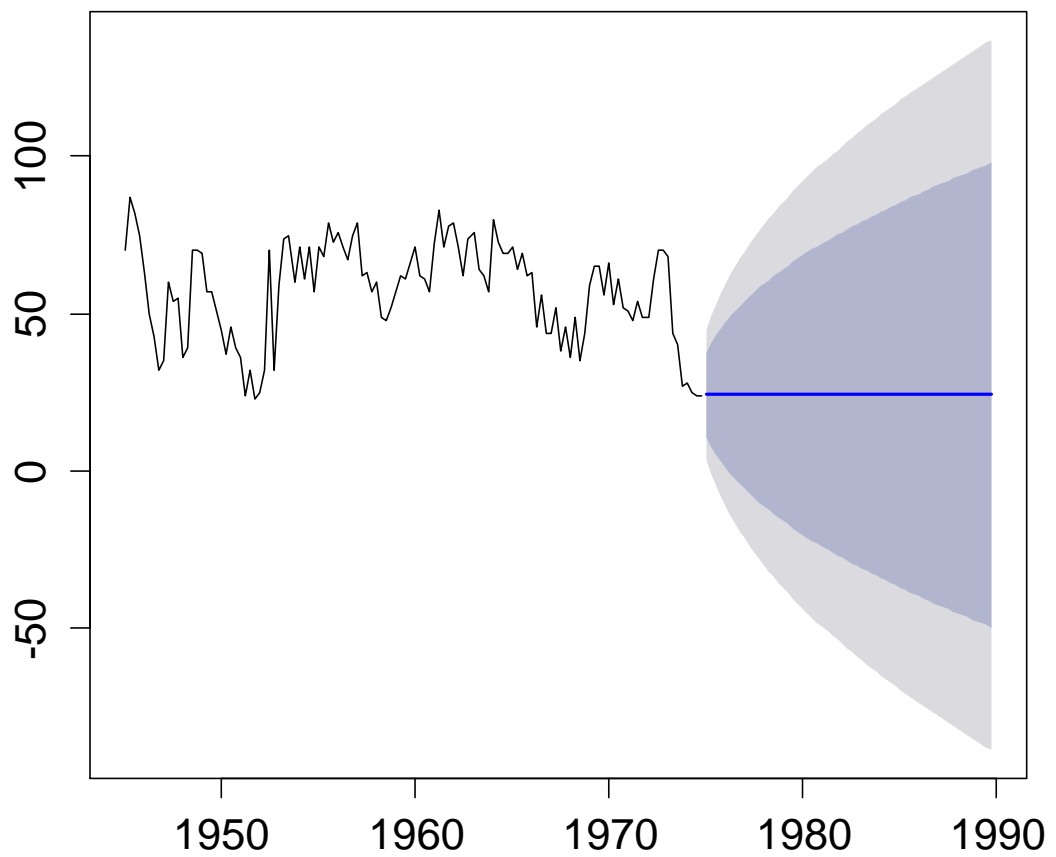


p values for Ljung-Box statistic



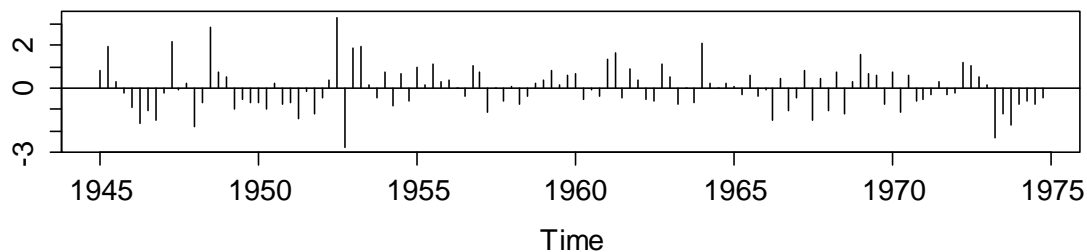
m. Using this model, forecast for next 5 years.
`plot(forecast(fite, 60))`

Forecasts from ETS(A,N,N)

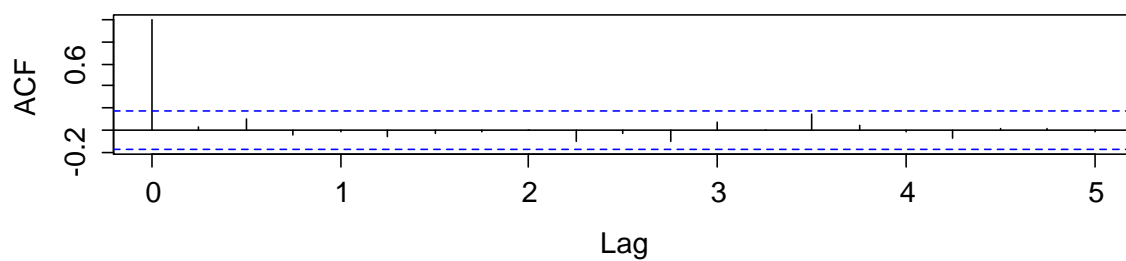


n. Using the data, fir ARIMA model, check diagnostic and forecast.
`fita = auto.arima(presidents)`
`tsdiag(fita)`

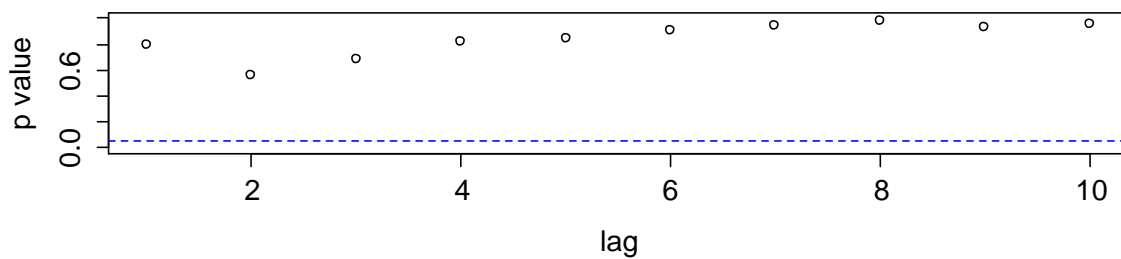
Standardized Residuals



ACF of Residuals

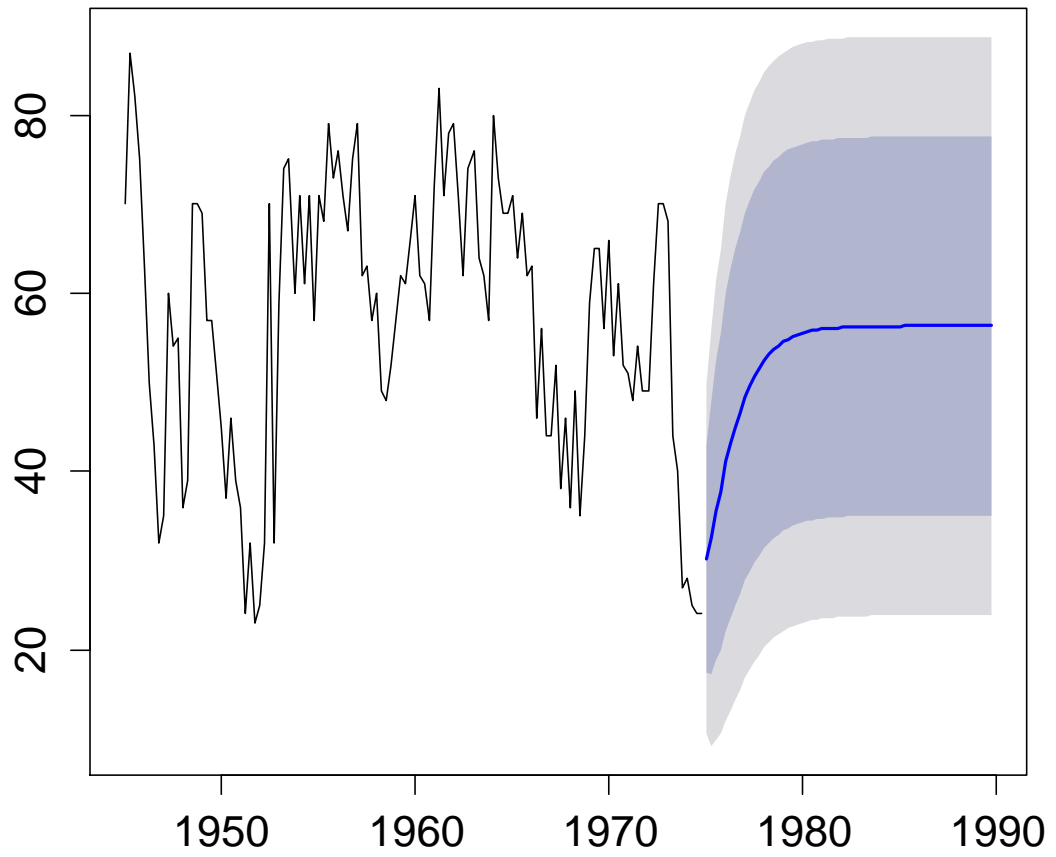


p values for Ljung-Box statistic



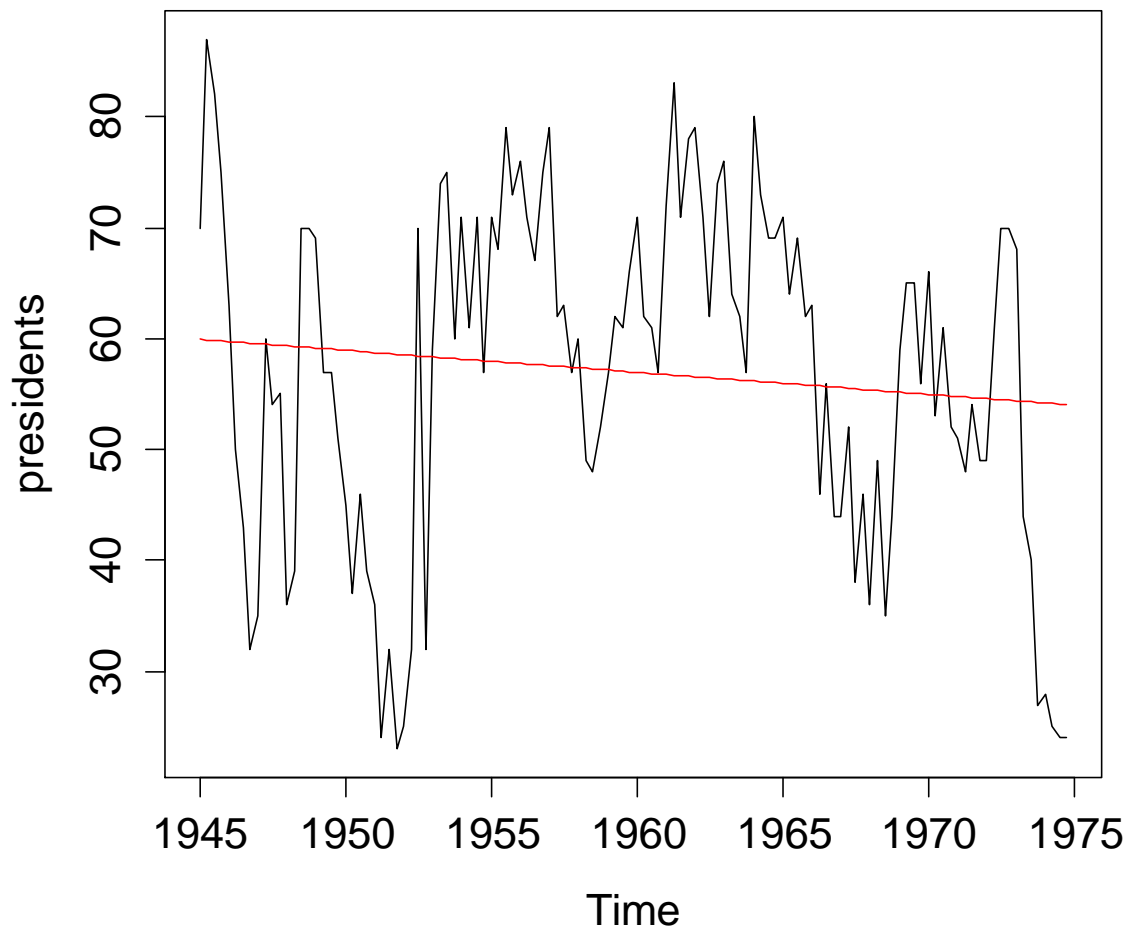
```
plot(forecast(fita, 60))
```

ecasts from ARIMA(2,0,0)(2,0,0)[4] with non-zero



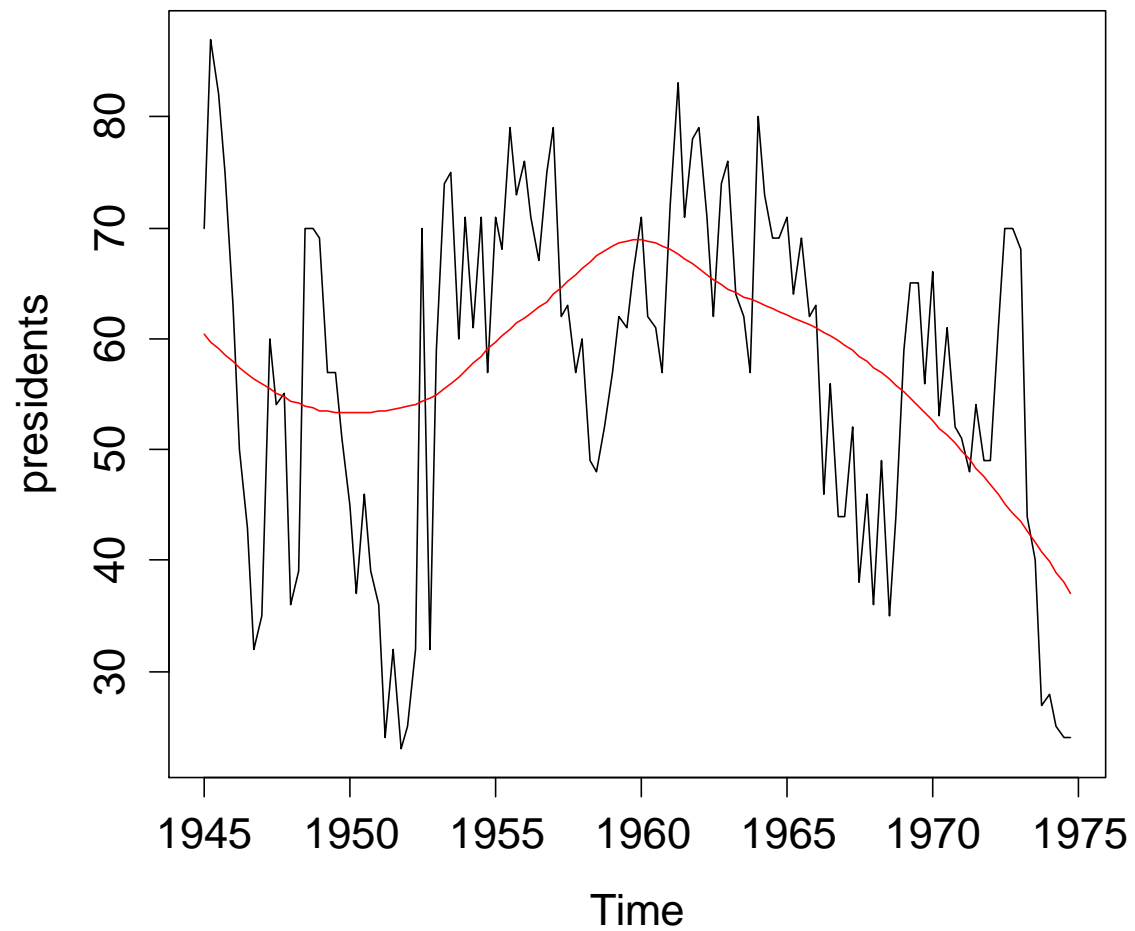
o. Using the data, fit a linear and put a fitted line in the plot.

```
lfit = lm(coredata(presidents) ~ index(presidents))  
plot(presidents)  
lines(predict(lfit) ~ index(presidents), col = "red")
```



p. Using the data, fit a smooth line and put a fitted line in the plot.

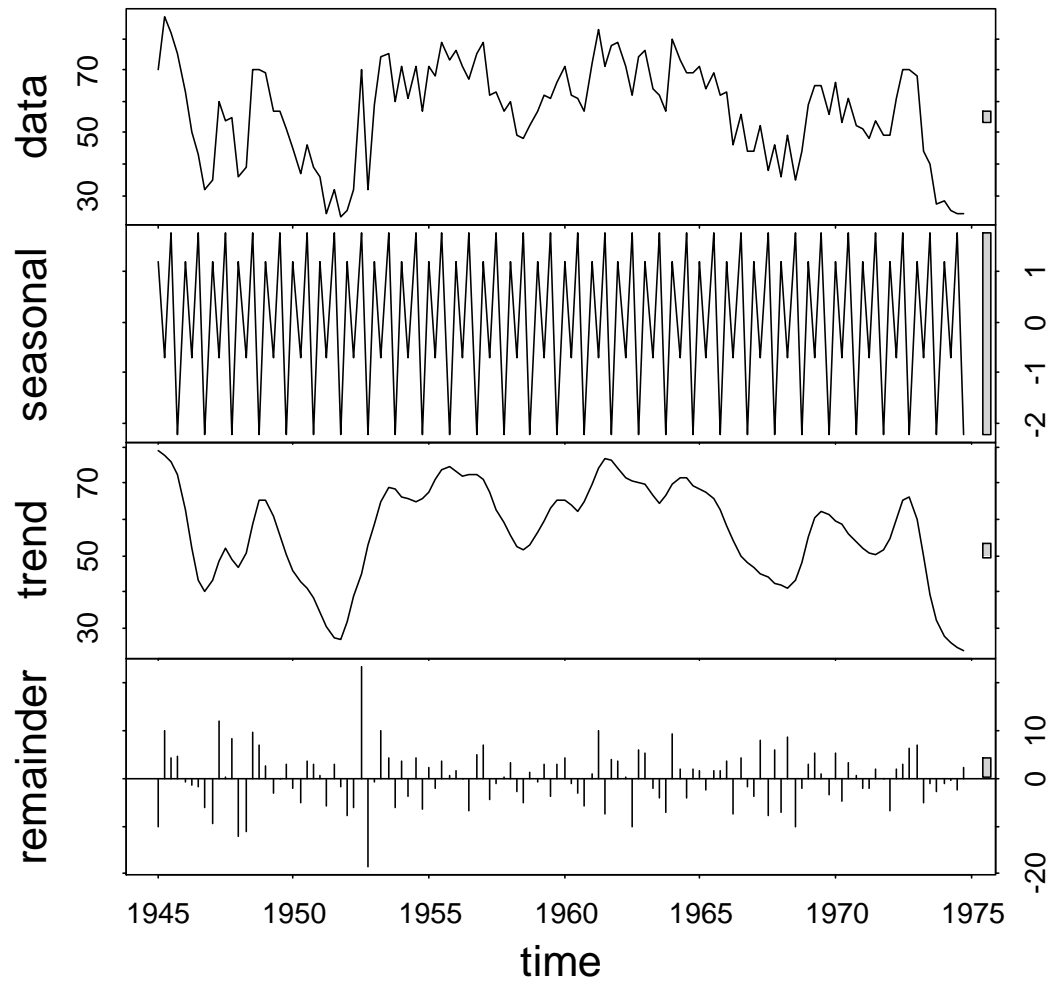
```
lfit = loess(coredata(presidents) ~ index(presidents))  
plot(presidents)  
lines(predict(lfit) ~ index(presidents), col = "red")
```



q. Using the data, plot the seasonal decomposition of the series.

```
dfit1 = stl(presidents, s.window = "per")
```

```
plot(dfit1)
```



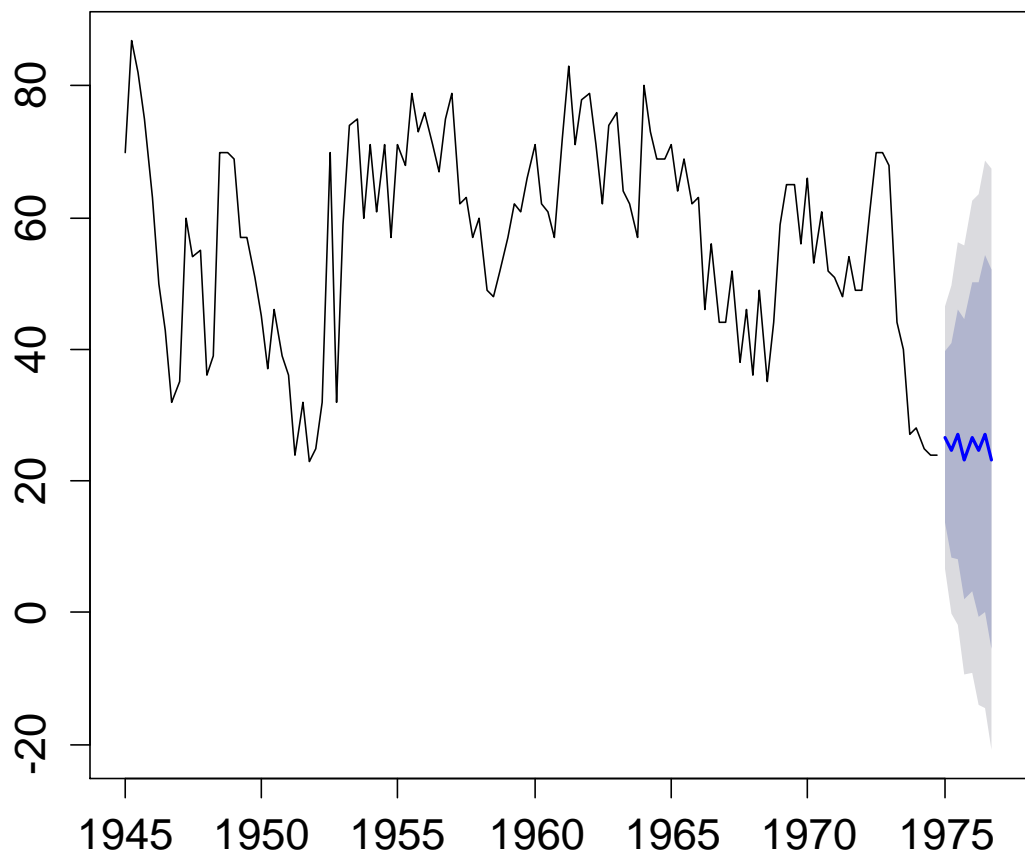
- r. Using the data, create a stl object (forecast package) and forecast from the stl object.

```
require(forecast)
```

```
dfit2 = stlf(presidents,s.window="per")
```

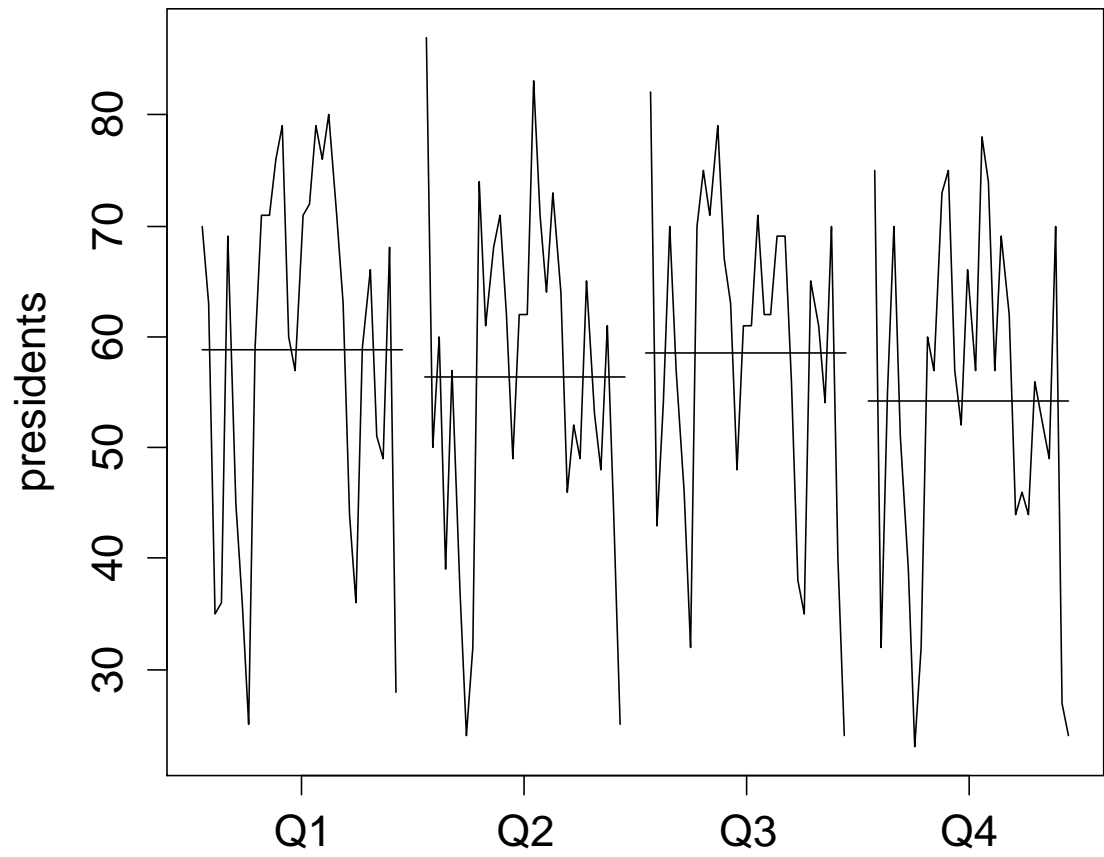
```
plot(dfit2)
```

Forecasts from STL + ETS(A,N,N)



s. Draw season plot.

```
monthplot(presidents)
```



```
require(zoo)
```

```
seasonplot(presidents)
```

Seasonal plot: presidents

