Dataset

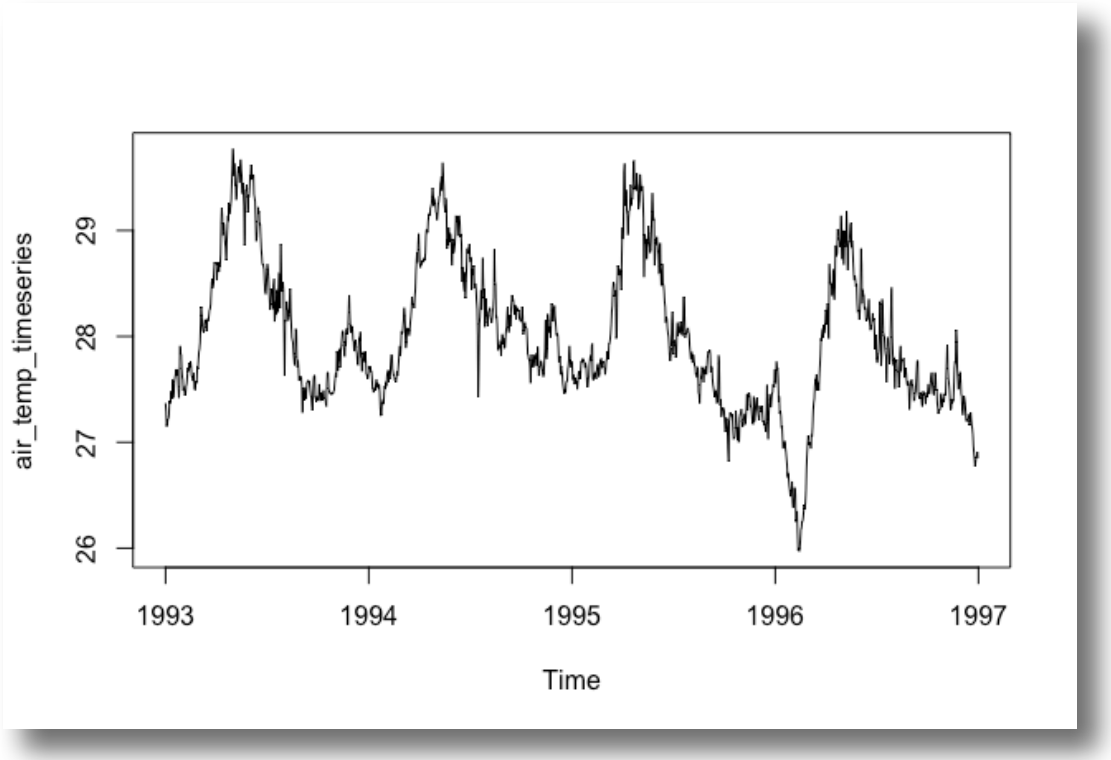
This dataset contains **surface meteorological readings** taken **daily** from a series of buoys positioned throughout the equatorial Pacific. All readings were taken at the same time of the day. This data is critical for understanding and prediction of seasonal-to-inter annual climate variations originating in the tropics.

The data under our looking glass spans a period of 4 years – from **1 January, 1993** to **31 December, 1996.**

There are missing values in the data. Not all buoys were able to measure currents, rainfall, and solar radiation, so these values are missing dependent on the individual buoy. In order to remedy the problem of missing data, new values were inserted at *NULL* by applying '**spline**' operation on data in the vicinity of the missing data. This was achieved by using 'zoo' library in R.

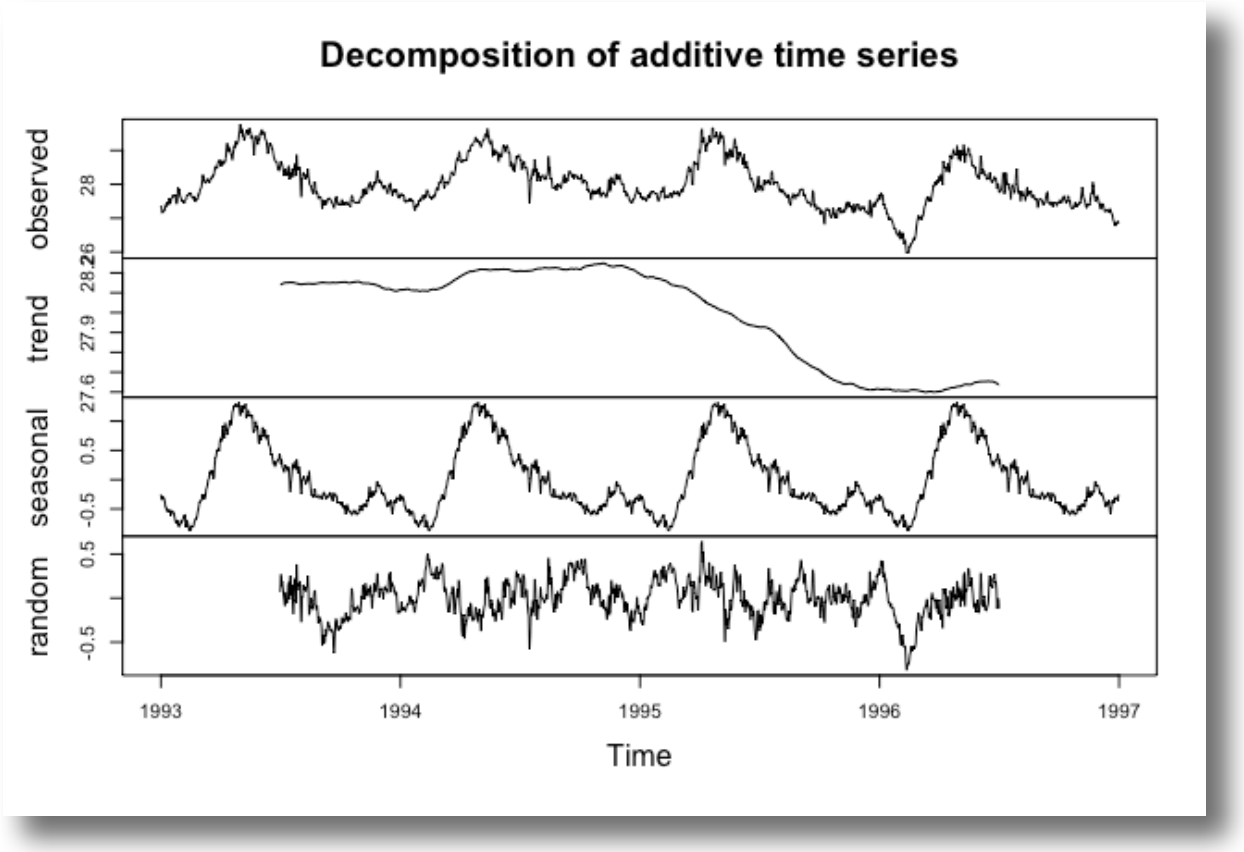
Time Series Plot

We start by plotting our time series data after implementing the aforementioned changes to it.



Decomposing

Next, we decompose this time series data into its various components – trend, seasonality, and randomness.



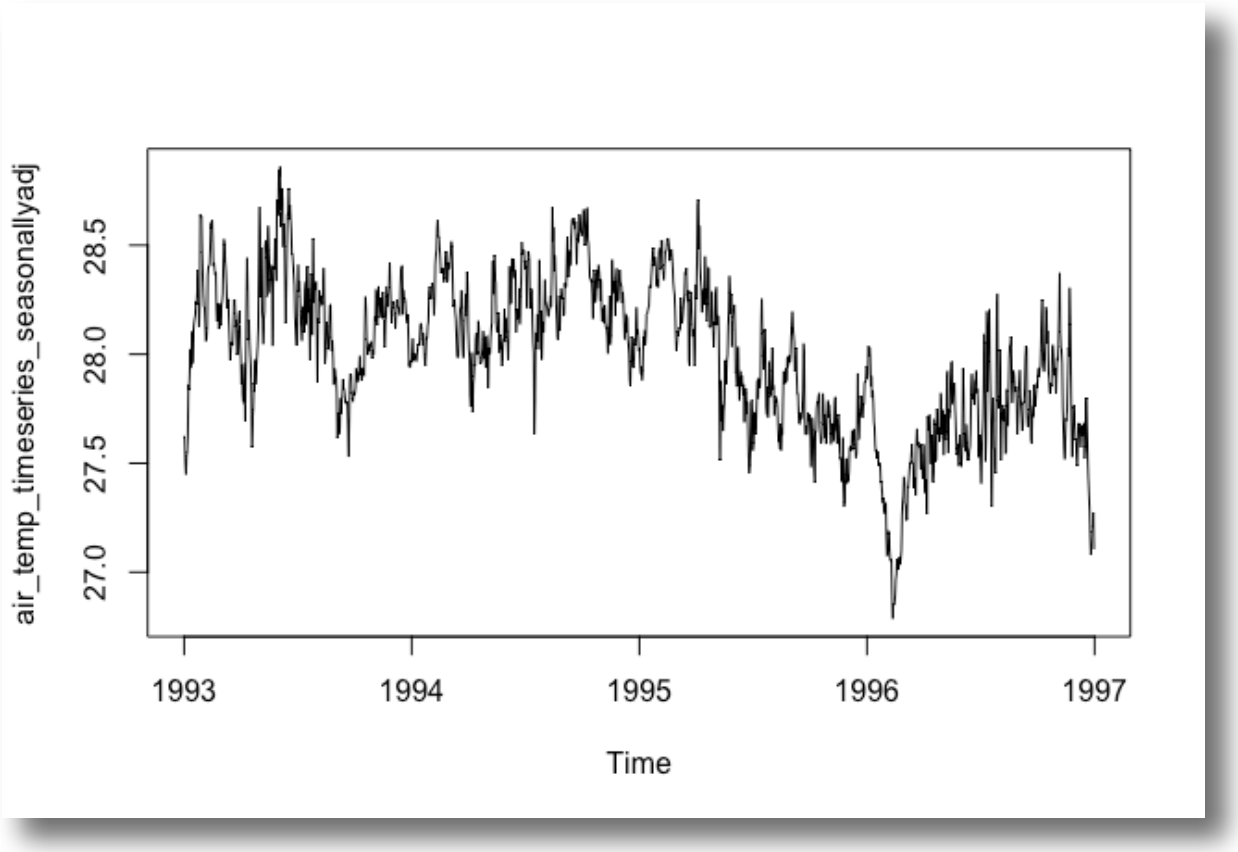
It can be observed that **trend** is **declining** over time and we can observe a prominent seasonal effect too with comparable magnitude.

The trend component suggests that sea surface temperature remained constant for the first couple of years and then there was a remarkable decline in sea surface temperature in the year 1996. This noticeable difference can be attributed to the greater presence of **El Nino** sea currents during that period in tropical pacific.

Seasonal component highlights the variation in sea surface temperature annually. It is found to be the highest around the middle of the year, i.e. in the months of May and June, and the least at the beginning of each year, i.e. in January.

Seasonal Adjustment

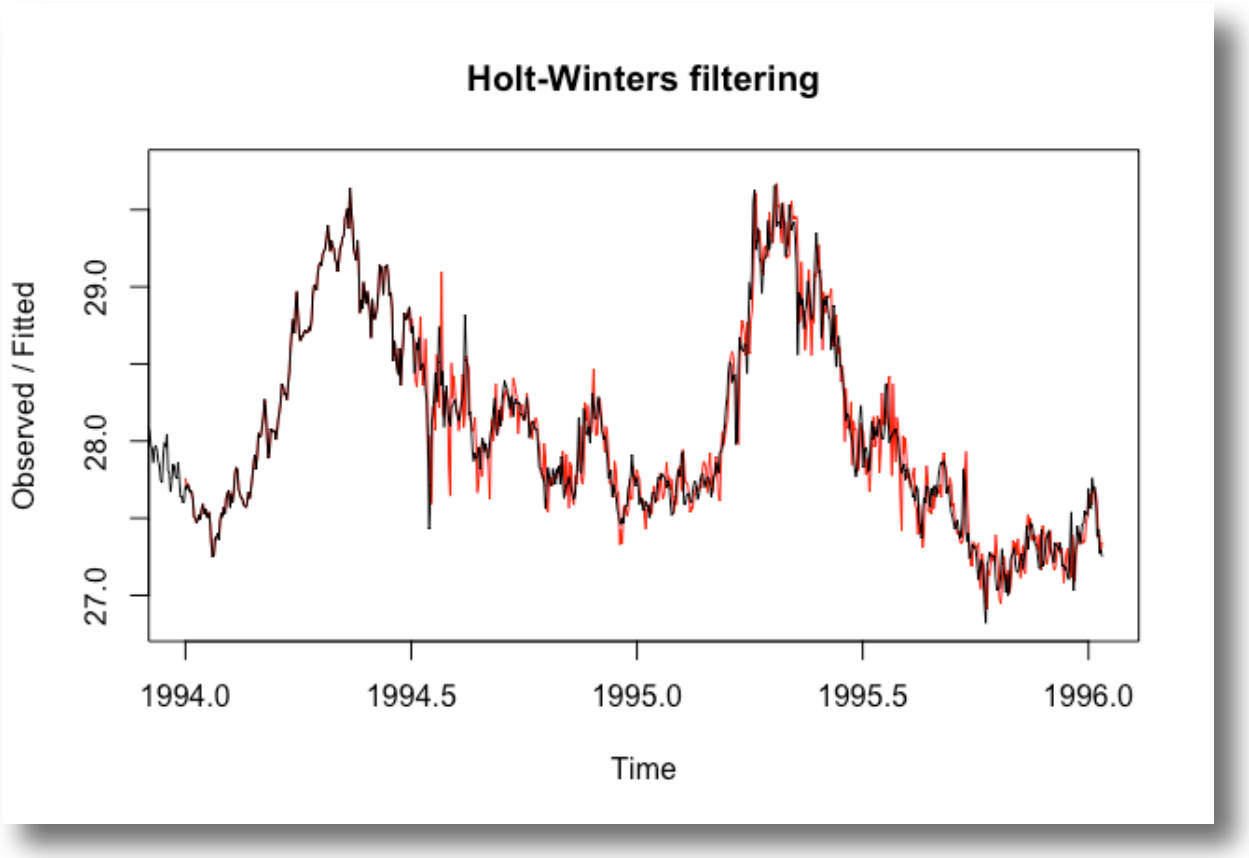
We now attempt to seasonally adjust our time series. We simply subtract the seasonal component from the time series to obtain seasonally adjusted data. This is possible because our model is **additive** in nature – individual components namely trend, seasonality, and randomness are added up together to make up the given time series. We later plot the resulting seasonally adjusted series as shown below:



The series above includes only trend component and random component. The seasonal component is absent here.

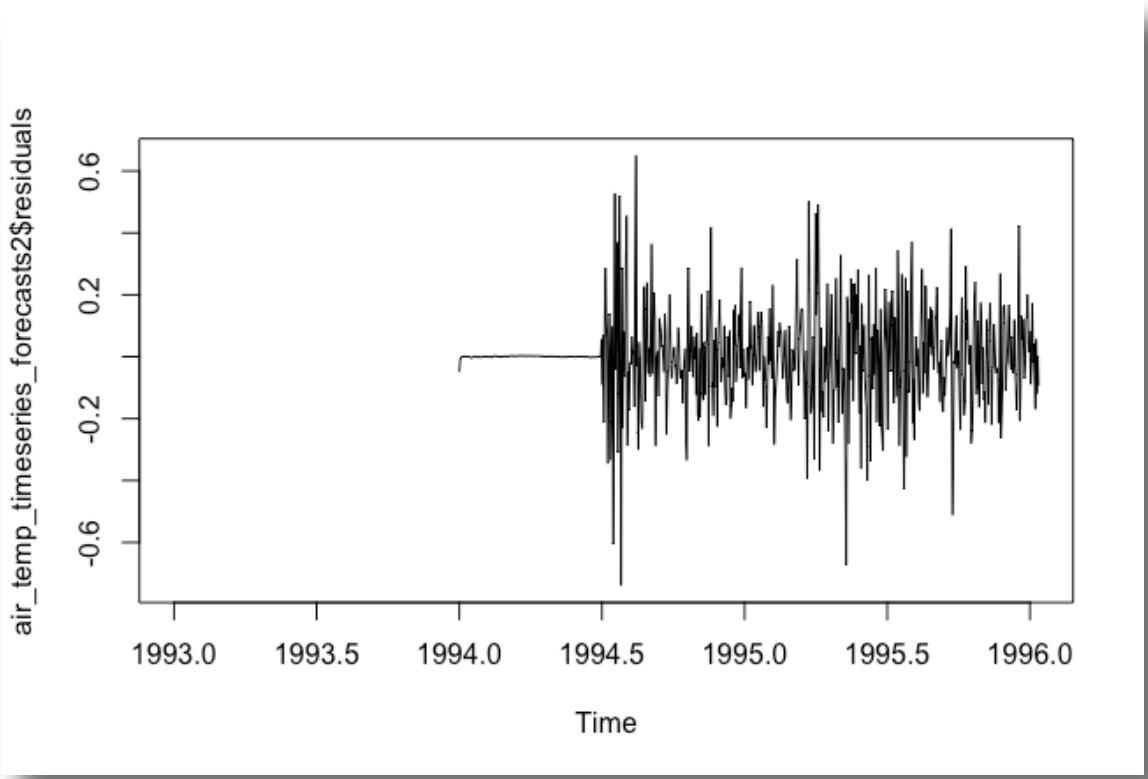
Exponential Smoothing

Since our data exhibits both trend and seasonal components and is an additive model, we implement **Holt-Winters** exponential smoothing.



Here, the `beta` and `gamma` parameters of `HoltWinters()` function are kept TRUE to allow for trend and seasonality while smoothing. In the plot given above, the observed values are plotted in black colour, while the fitted values are in red. We observe that there is no visible difference in the observed and fitted graph, whereas the difference creeps in and grows visibly larger as one moves ahead in time. This can be explained by observing the residuals' plot, which come into being only after a year i.e. from 1994 onwards.

For the forecast to explain the model adequately, the residuals should behave as *white noise*. We notice from the plot below that the residuals have mean equal to zero. To check for serial autocorrelation, we run **Ljung-Box test** on the residuals.

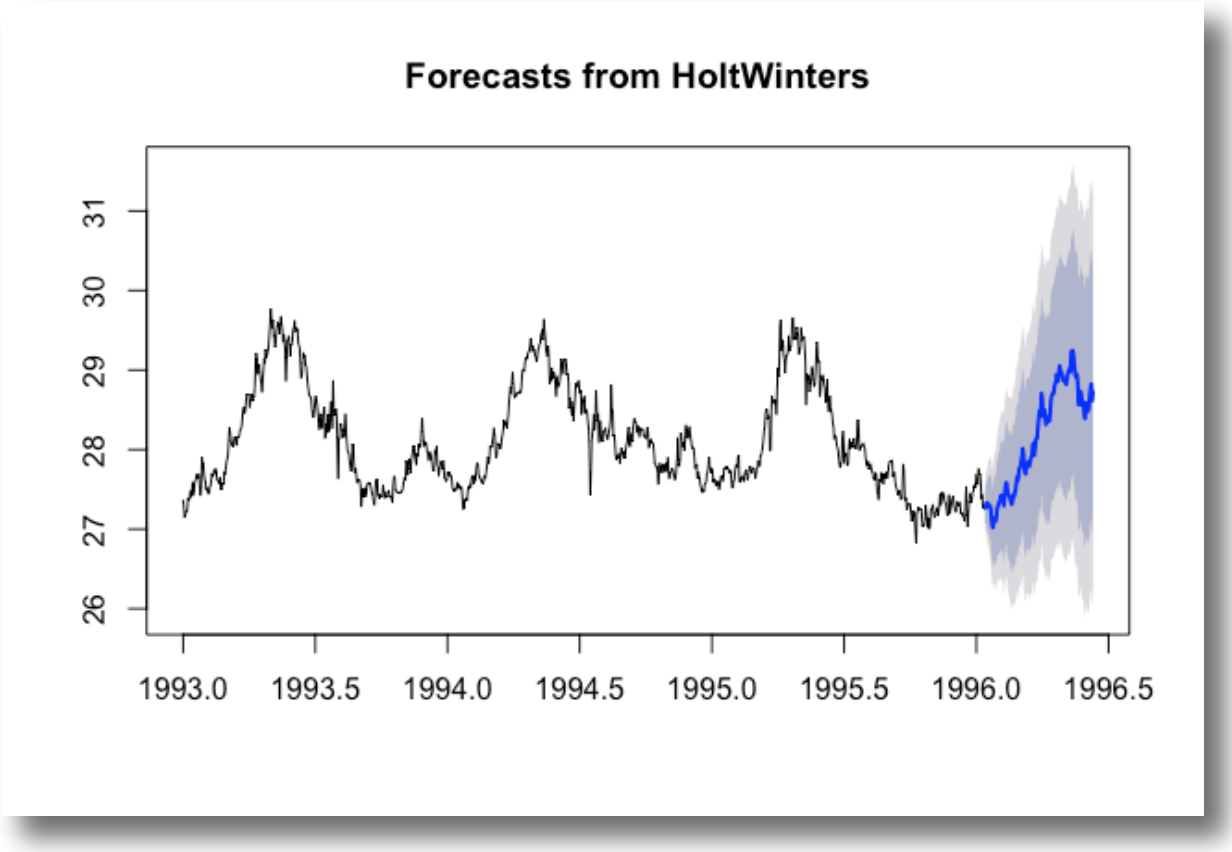


The null hypothesis of the test indicates that the residuals do not suffer from serial autocorrelation. However, the result below **rejects** the null hypothesis. This means there is **serial autocorrelation** of residuals present in our model.

```
> Box.test(air_temp_timeseries_forecasts2$residuals, lag=20, type="Ljung-Box")

        Box-Ljung test

data:  air_temp_timeseries_forecasts2$residuals
X-squared = 79.079, df = 20, p-value = 5.623e-09
```

Forecasts Using Holt-Winters Smoothing

We forecast the sea surface temperature for the next 150 days using Holt-Winters smoothing method. The plot captures the point estimates of the forecast with blue colour, 80% confidence interval estimates with dark grey, and 95% confidence interval forecasts with light grey colour.



Stationarity Tests

We run the **Augmented Dicky-Fuller** test to check for non-stationarity in our time series model. The null hypothesis of the ADF test indicates the presence of a unit root. The alternative hypothesis states the series is stationary. After running the test, we **fail to reject** the null hypothesis and conclude that the time series suffers from **non-stationary**.

```
> adf.test(air_temp_timeseries)

        Augmented Dickey-Fuller Test

data:  air_temp_timeseries
Dickey-Fuller = -2.7, Lag order = 11, p-value = 0.282
alternative hypothesis: stationary
```

The remaining two tests – **Phillips-Perron** unit root test and **KPSS** test – also indicate the presence of **non-stationarity** in our model. PP test and KPSS test have opposite null hypothesis.

```
> pp.test(air_temp_timeseries)

        Phillips-Perron Unit Root Test

data:  air_temp_timeseries
Dickey-Fuller Z(alpha) = -17.945, Truncation lag parameter = 7, p-value = 0.1087
alternative hypothesis: stationary
```

```
> kpss.test(air_temp_timeseries)

        KPSS Test for Level Stationarity

data:  air_temp_timeseries
KPSS Level = 2.4843, Truncation lag parameter = 7, p-value = 0.01
```

To remedy this, we take the **first difference** across our time series. We then check for non-stationarity again against ADF test. As seen below, we now **reject** the null hypothesis and conclude that now our first differenced time series is **stationary**.

```
> air_temp_timeseries_diff1 = diff(air_temp_timeseries, differences=1)
> adf.test(air_temp_timeseries_diff1)

        Augmented Dickey-Fuller Test

data:  air_temp_timeseries_diff1
Dickey-Fuller = -11.236, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```

ARIMA Model

Since our time series model is seasonal in nature, simply plotting ACF and PACF and inferring p and q would not give an accurate ARIMA model. Hence, we run `auto.arima()` command in R and obtain a **Seasonal ARIMA (SARIMA**) model.

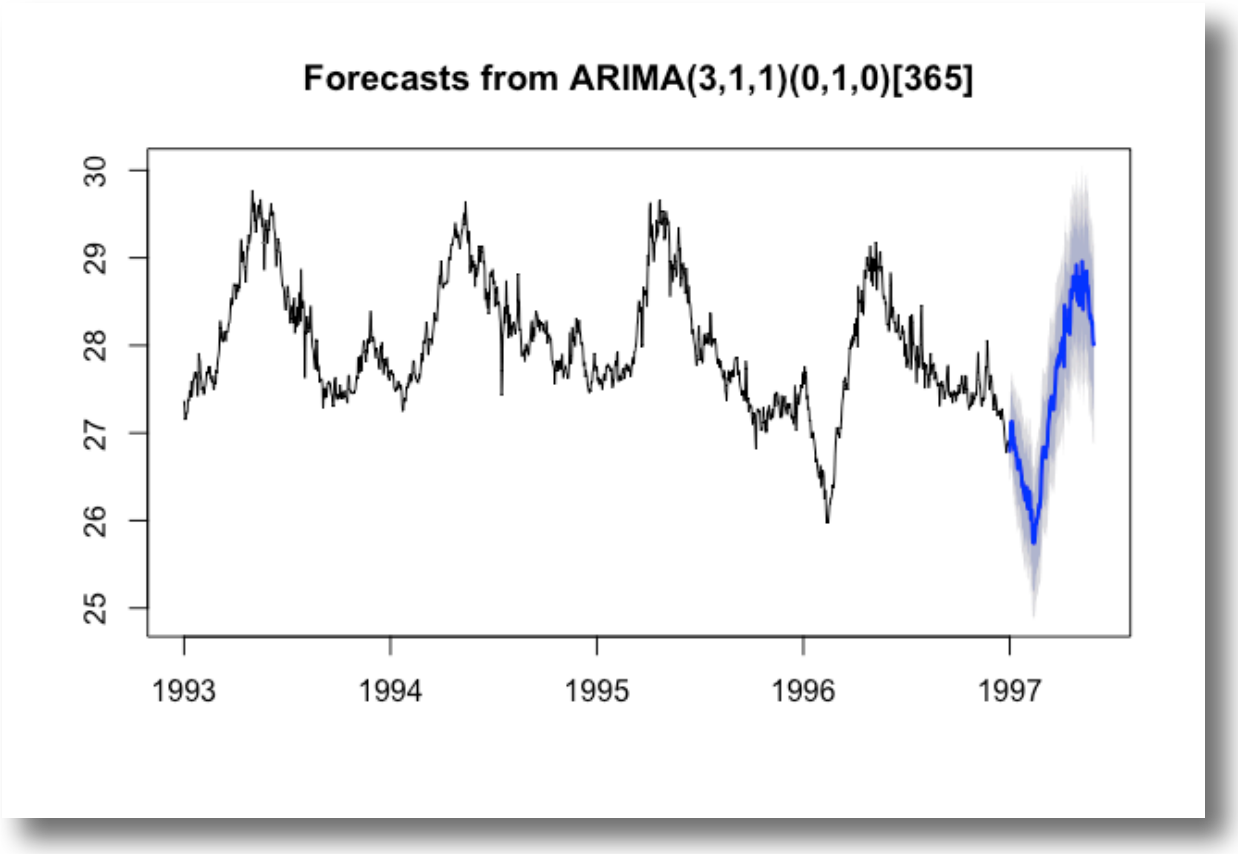
```
> summary(air_temp_timeseries_arima)
Series: air_temp_timeseries
ARIMA(3,1,1)(0,1,0)[365]

Coefficients:
          ar1      ar2     ar3      ma1
      0.8717  -0.1451  0.1599  -0.9747
s.e.  0.0342   0.0398  0.0320   0.0154

sigma^2 estimated as 0.02623:  log likelihood=438.04
AIC=-866.08   AICc=-866.03   BIC=-841.09
```

The model returned above is the best suited model according to AIC values evaluated for other trial SARIMA models internally. **`ARIMA(3,1,1)(0,1,0)[365]`** had the least value for AIC among all the models tested.

`(3,1,1)` gives the `(p,d,q)` of the usual ARIMA parameters. `(0,1,0)` gives `(P,D,Q)` of the seasonal component of the time series. `[365]` indicates the frequency of the daily time series model. Here, we fed undifferenced time series into ARIMA and hence we get '1' as difference here. This has been taken care by ARIMA and its object has been stored in a variable.

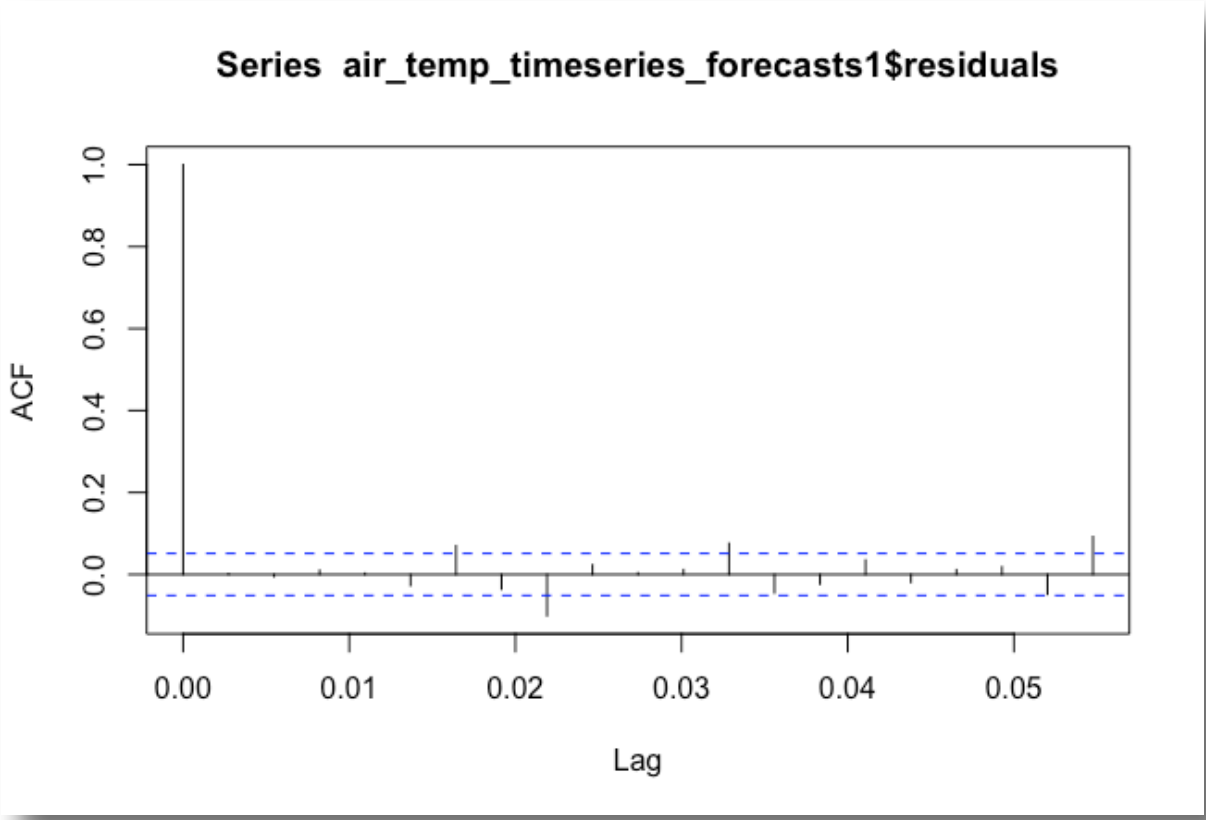
Forecasts Using ARIMA Model

We now use the above model in forecasting future values of the sea surface temperature. We employ the aforementioned variable in `forecast()` to predict values of next 150 days.



The plot of the forecasted values is given above. The plot captures the point estimates of the forecast with blue colour, 80% confidence interval estimates with dark grey, and 95% confidence interval forecasts with light grey colour. We observe that this plot is different to the forecasting plotted using Holt-Winters smoothing result. Further, the confidence intervals are **narrower** in this case, indicating a more **precise forecast** than made previously.

Next, we perform **residual diagnostic** by checking for serial autocorrelation. We plot autocorrelation function (ACF) of the residuals of the ARIMA model. From the ACF plot given below, it is evident that there is **no autocorrelation among the residuals**, since none of the lines (after the first one) in the plot crosses the confidence interval.



To further verify this observation, we perform Ljung-Box test on the residuals of our forecasted ARIMA model. We fail to **reject** the null hypothesis suggesting the residuals in our model are **uncorrelated**. This means that our model has adequately captured the information in the data and the forecasts given are robust.

```
> Box.test(air_temp_timeseries_forecasts1$residuals, type='Ljung-Box')

        Box-Ljung test

data:  air_temp_timeseries_forecasts1$residuals
X-squared = 0.0054201, df = 1, p-value = 0.9413
```