

Time Series Forecasting with Excel, JMP, Minitab, SPSS, CrunchIt!, R, and TI-83/-84 Calculators

Time series models can vary widely depending on the structure of the data and the type of model desired, from a simple regression model to extremely complex ones. The basic data assumption is that all observations were taken at equally spaced intervals.


Time Series Graphs and Simple Trend Models



Excel

There are two ways in Excel to create a time series graph and fit a trend line.

1. You can create a time series graph using Insert → Scatterplot with Straight


Lines . The time index (1, 2, ...) or date (X) variable must be the left column in the two adjoining columns of data in the spreadsheet (use copy and paste to make the columns adjacent to one another, if needed).

Simple trend models can be fit using linear regression from the Data → Data Analysis menu.

2. Highlight the column of the time series, then Insert → Line. Add the trend line by right-clicking on a data point and selecting “Add Trendline.”



There are three options to create a time series graph.

1. Use Graph → Graph Builder and drag a variable with the time index (or date) to the x axis; drag the variable of interest to the y axis. Click the  icon, right-clicking on any point, and select “Add trendline” from the pop-up to connect the points.
2. Use Analyze → Time Series. Click to enter the series of interest into the “Y, Time Series” box. If you have a data or index variable, that can be entered into the “X, Time ID” box.
3. Graph → Overlay Plot. Select the time series data column into Y, and click “OK.” Connect the points by clicking the red triangle and selecting “Connect Thru Missing.”

Simple trend models can be fit using the time (or date) as X and the variable of interest as Y with Analyze → Fit Y by X.

**Minitab**

There are three options to create a time series graph.

1. Use Graph → Time Series Plot. Use the default “Simple” and click “OK.”
 - a. Click to select and enter the variable of interest into the main box.
 - b. Minitab will default to using an index (1, 2, ...) for the x axis. If you want a different axis labeling, click “Time/Scale” and select the radio button for the appropriate option. To start graphing yearly data, for example, click “Calendar” and use the drop-down to select “Year.” Enter a starting year in the box. If the data are annual (increment 1), simply click “OK” to return to the main dialog.
2. With a time variable (or an index variable), use Graph → Scatterplot.
 - a. Select “With Connect Line” and click “OK.”
 - b. Click to enter the variable of interest as Y and the time (or index) as X . Click “OK.”
3. Use Stat → Time Series → Trend Analysis. Click to enter the column containing the series. For the time variable, you can use the default (indices 1, 2, ..., n) or click “Time” and change to years, months, etc. Click “OK” for the plot with the trend line.

Simple trend models can be fit using Stat → Regression → Regression → Fit Regression Model with the variable of interest as the response and the time (or index) variable as a continuous predictor.


SPSS

Assuming you have the correct version of the software (there are several from Base to Standard to Professional), there are two ways to create a time series plot.

1. Use Graphs → Legacy Dialogs → Line.
 - a. In the next dialog, Simple should be the default; move the radio button in “Values in Chart are:” to “Values of individual cases.” Click “Define.”
 - b. Click to enter the variable of interest into the box labeled “Line Represents.” If you have a time variable, move the radio button from “Case Number” to “Variable” and enter that variable name. Click “OK.”
2. Use Analyze → Forecasting → Sequence Charts (only if you have this module).
 - a. Click to enter the variable of interest into the box labeled “Variables.” You can leave “Time Axis Labels” blank if you simply want a time index used on the X axis.
 - b. Click “OK.”




Simple trend models can be fit using Analyze → Regression → Linear if you have a time variable (or index) in the data set. Click to enter the variable of interest as the dependent variable and the time variable as the independent variable.



CRUNCH IT!

Use Graphics → Scatterplot. Define the plot with the time variable as the x variable and the statistic as y . Set the “Display” option to either “Line” or “Both.”

Simple trend models can be fit using Statistics → Regression → Simple Linear. Use the time variable as the independent variable and the variable of interest as the dependent variable.



You can create the time series plot using the connected Scatterplot () option from   (STAT PLOTS). Define the plot with the time variable as the x variable and the variable of interest as y .

Simple trend models can be fit using  \rightarrow Calc \rightarrow LinReg($a+bX$).



The simplest way to graph a time series is with the time series plot command

```
> ts.plot(var)
```

You can also use the regular plot command. The `type="l"` part tells R to connect the points. Note that no time variable is needed here.

```
> plot(var, type="l", xlab="time", ylab="Variable Name")
```

The simple trend model can be fit using `lm` (linear model) command as detailed in Chapters 2 and 10.

Trend with Seasonality

These models allow cyclical behavior in the variable of interest. You need one fewer indicator variable than there are periods in the year (3 for quarterly data, for example). They are fit as a multiple regression.



Excel

Data files may have dates entered as, for example, 01/2010. We need to separate the month from the date to add the indicator variable while preserving the original for the trend part of the model.

1. Place the cursor in a blank column on the same row as the first date. Enter the command `=text(a2,"MM")` if the first date is in cell A2. Copy the formula down the column to extract the month from the rest of the dates.
2. Create the indicator variables using the "If logical" function. Supposing the months were placed in column D, this would look like `=if(d2="01",1,0)`. Then copy/paste to fill that column.
3. Repeat step 2 until you have created all the indicator variables needed.
4. Copy/paste columns as needed until you have all the predictor variables in a contiguous range.
5. Use Data \rightarrow Data Analysis \rightarrow Regression to compute the multiple regression using the original date variable and the indicators as predictors.



Data files may have dates entered as, for example, 01/2010. We need to separate the month from the date to add the indicator variable while preserving the original for the trend part of the model. This requires an add-in that can be found at <https://community.jmp.com/docs/DOC-7537>. You need only install this add-in once. (This procedure also works for quarterly data.)

1. Right-click in the column header (the variable name). Select New Formula Column → Date Time → Month. This creates a new column with [Month] as part of the variable name.
2. Right-click in the new column header and select “Column Info” to create a new name without the square brackets.
3. Click Add-Ins → Make Dummy Variables. Select the variable just created and click “OK.” One dummy variable will be created for each level of the original.
4. Use Analyze → Fit Model with the original month/year variable and one fewer indicator than there were months (or quarters) to fit the regression model.



Minitab

Data files may have dates entered as, for example, 01/2010. We need to separate the month from the date to add the indicator variable while preserving the original for the trend part of the model.

1. Click Data → Date/Time → Extract to Numeric. If you want the month, enter the column with the date to convert in the “Extract from ...” box and a new column to receive the month. Check the box labeled “Month” and click “OK.”
2. Create the indicator variables using Calc → Make Indicator Variables. Enter the column just created as “Indicator variables for.” Minitab will populate a table of variables it intends to create, along with the proposed column names. Click “OK.”
3. Use Stat → Regression → Regression → Fit Regression Model to compute the multiple regression using the original date variable (as continuous) and the indicators as categorical predictors. Be sure to *not* use the last indicator.



Data files may have dates entered as, for example, 01/2010. We need to separate the month from the date to add the indicator variable while preserving the original for the trend part of the model (**requires Python Essentials**).

1. Click Transform → Compute Variable. Enter a new variable name in the box at upper left. In the “Function Group” box, select “Date Extraction.” In the “Functions” box, select “Xdate.Month.” Double-click to send that shell to the formula box. Click on the original variable to replace the ? in the function with the variable name. Click “OK.”
2. Now, create the indicator variables. Click Transform → Create Dummy Variables. Click to select and enter the name of the variable just created; give a stub name for the new indicator variables. Click “OK.”
3. Use Analyze → Regression → Linear to fit the model using one fewer indicator than was just created.



Data files may have dates entered as, for example, 01/2010. We need to separate the month from the date to add the indicator variable while preserving the original for the trend part of the model. The first step here ensures R understands the date as a date.

```
> date<-as.Date(Month, format="%m/%d/%Y")
```

Next, extract the month from the date

```
> month2<-format(date, "%m")
```

Create the indicator variables

```
> month.f=factor(month2)
> dummies=model.matrix(~month.f)
```

Perform the regression

```
> res = lm(Retail.Sales ~ date + dummies)
```

Lag Regression Models

These models exploit the relationship between an observation and its previous value. The degree of relationship is measured by the autocorrelation function.



Excel

To find the autocorrelations in a time series, copy and paste the column of data (with the variable of interest) so it is offset by 1 observation (paste so the first observation is now paired with the second). Use Data → Data Analysis → Correlation to find the correlation of (Y_t, Y_{t-1}) .

Use Data → Data Analysis → Regression to fit the model using the offset column as the predictor (X).



To find the autocorrelation (and partial autocorrelation) in a time series, use Analyze → Time Series (in the student version) or Analyze → Modeling → Time Series (in the full version). Click to enter the variable of interest into the “Y, Time Series” block and (if desired) the dates into the “X, Time ID” block. You can set the number of autocorrelation lags to something like 2 from the default 25. Click “OK.”

To fit the autoregressive model, there are two options.

1. Lag the series the desired number of lags by selecting the column containing the series and using Columns → Row → Lag. Fill the command shell with the desired number of lags. Use Analyze → Fit Y by X to fit the regression.
2. In time series mode, click the red triangle at the top of the output and select “ARIMA.” Set the “Autoregressive” order to 1 (or the desired number of lags) and click “Estimate.”



Minitab

Find the autocorrelation (partial autocorrelation) using Stat → Time Series → Autocorrelation (Partial Autocorrelation). Click to select and enter the series variable name. You can also move the radio button to only ask for a reasonable (2 or 3) autocorrelation lags.

To fit the model, you have two options:

1. Use Stat → Time Series → ARIMA and set the Autoregressive order to 1 (or the desired number of lags),
or
2. Use Stat → Time Series → Lag and lag the time series by 1 or 2 or whatever lag is of interest. Click to enter the time series name, a column for the new variable and enter the desired lag. Then use Stat → Regression → Regression → Fit regression model. Use the original time series as the response and the lagged series as a continuous predictor.

SPSS

Use Analyze → Forecasting → (Partial) Autocorrelations to find the (partial) autocorrelation. Click to select and enter the time series as the variable. Uncheck the box for Partial Autocorrelations. To limit the number of autocorrelations shown, click “Options,” then enter a smaller number (like 2 or 3). Click “Continue” to return to the main dialog and click “OK.”

To fit the model, first lag the series. Use Transform → Shift Values. Click to select and enter the series name. Enter a name for the result variable in the box labeled “Name.” Click “Change.” The default is a 1 period lag; if a different lag is desired, enter that. Click “OK.” Use Analyze → Regression → Linear to fit the model using the lagged variable as the predictor.



TEXAS
INSTRUMENTS
TI-83/-84

Create a lagged time series using the following steps:

1. Place the cursor so it highlights the name of an empty list. Use $\boxed{2nd}$ to enter the list name you wish to duplicate. The command at the bottom of the screen should look like $L2 = L1$, for example.
2. The cursor will be on the first value in the new list. Press $\boxed{2nd} \boxed{DEL} = \boxed{INS}$ (Insert).
3. Delete the entries in the first row of both lists (to eliminate pairing the first observation with the inserted 0), and the last entry in the new list.
4. If more lags are desired, repeat the insert until you have offset by the desired amount. Be sure to delete entries as needed until the sizes of the lists match.

For the autocorrelation and regression, use \boxed{STAT} , Calc, 8:LinReg $a+bx$ using the original list as Y and the new list as X . For example, LinReg ($a+bx$) $L2,L1$.



For the autocorrelation (or partial autocorrelation), the command shown below uses “series” as the name of the variable; nn is a reasonable number (usually 2 or 3) for the maximum lag to consider, and plot=F tells R to display the numeric coefficients instead of simply plotting them. For partial autocorrelations, the command is pacf with the same parameters.

```
> acf(series, lag.max=nn, plot=F)
```

To create a lagged series by one time step, use the command like

```
> lagseries <- lag(series, k=-1)
```

where “series” is the name of the original time series. For the regression, use a command like

```
> res = lm(Retail.Sales~lagseries)
```

Moving Average and Smoothing Models

Moving average (MA) models smooth a time series by sliding a window of width k and computing the average of those k previous observations to replace observation y_t . Exponential smoothing models use a weighted average of the previous observation and its forecast, which progressively lowers the weight (impact) of observations far distant on the estimate of a current value.



For a moving average, use Data → Data Analysis → Moving Average.

1. Drag to select the column with the time series.
2. Enter the value of k in the box labeled “Interval.”
3. Specify a cell in an empty column for the “Output Range.”
4. Check the box for “Chart Output,” if desired. Examining the standard errors is beyond the scope of this text, but if you look at them, you’ll notice these are higher on the ends of the series and lower in the center.

For exponential smoothing, use Data → Data Analysis → Exponential Smoothing.

1. Drag to select the column with the time series.
2. Enter $1-w$ in the box labeled “Damping Factor.” This means w will be the weight used on observation y_{t-1} .
3. Specify a cell in an empty column for the “Output Range.”
4. Check the box for “Chart Output,” if desired. Examining the standard errors is beyond the scope of this text, but if you look at them, you’ll notice these are higher on the ends of the series and lower in the center.



Use Analyze → Time Series as discussed earlier.

For moving average smoothing, click the red triangle at the top of the output and select Smoothing Model → Simple Moving Average.

1. Enter the value of k ,
2. Click “OK” to select “No centering” (centering replaces observation $y_{t-(k-1)/2}$ instead of observation y_t as used in your text).

For exponential smoothing as described in your text, click the red triangle at the top of the output and select Smoothing Model → Simple Exponential Smoothing.

1. To specify the weight, select “Custom” from the Constraints drop-down.
2. Use the drop-down for Level to select “Fixed.”
3. Enter the weight.
4. Click “Estimate.”



Minitab

For a moving average smooth, use Stat → Time Series → Moving Average.

1. Click to select and enter the time series variable. Enter k in the box labeled “MA length.”
2. Click “OK.” Minitab will superimpose the smooth on the original time series plot.

For exponential smoothing, use Stat → Time Series → Single Exp Smoothing.

1. Click to select and enter the time series variable.
2. For specified weights, move the radio button from “Optimal” to “Use” and enter the specified weight.
3. Click “OK.” Minitab will superimpose the smooth on the original time series plot.



SPSS does not have a built-in function to do moving average smoothing.

SPSS can do exponential smoothing, but not with specified weights. The system will determine an “optimal” weight or you if you use Analyze → Forecasting → Create Traditional Models and move the drop-down to “Exponential smoothing.”



Using these methods requires we actually declare the variable to be a time series. Use the `ts` command as illustrated below. The terms “frequency” and “start” are optional, but can be used to specify the time labels. For example, “frequency=12, start=c(2000,1)” means that the series is monthly and starts in January, 2000.

```
> Vartimeseries <-ts(var, frequency=nn, start=c(t1,inc))
```

With this definition, you can also use `plot.ts` to plot the time series as

```
> plot.ts(vartimeseries)
```

The smoothing commands shown below require you to install and load the package TTR. (Click Packages → Set CRAN Mirror and choose a site; Packages → Install Package and locate TTR; Packages → Load Package → TTR.)

A moving average smooth of length k is created with the `SMA` command. Use `plot.ts` to see the smoothed time series.

```
> SMA(vartimeseries,n=k)
```

The exponential smooth is created with the `HoltWinters` function. with `alpha` being the specified weight. Plotting this afterward shows both the smooth and the original series.

```
> ExpSM <-HoltWinters(vartimeseries,Gamma=F, Beta=F,
alpha = wt)
```