

Appendix

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, the Runs Test, Autocorrelation and Prediction



To create a time series graph, use **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).

Excel cannot calculate the runs test for you.

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



There are two 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 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 date or index variable, that can be entered into the “X, Time ID” box.
Note: This option also gives the autocorrelations.

JMP cannot calculate the runs test for you. Also, it will not give predictions without a model.



There are two 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 and **OK** to obtain the graph.
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**.

For the runs test, click **Stat → Nonparametrics → Runs Test**. Click to enter the variable of interest and click **OK**.

Find the autocorrelation using **Stat → Time Series → 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. Click **OK** for results.



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. If you have a time (or index) variable in the data set, you can use **Graphs → Chart Builder**.
 - a. Select the chart type of a connected **Line**, .
 - b. Drag the variable of interest into the box labeled “Y-axis?” and the time variable to the “X-axis?” box. Add titles and other axis labels if desired. 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**.

To do the runs test:

1. Click **Analyze → Nonparametric Tests → One Sample**.
2. Move the radio button to “Test sequence for randomness.”
3. Click **Run**.



Find the autocorrelations.

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

CRUNCH IT!

You must have a time (or index) variable. Use **Graphics** → **Scatterplot**. Define the plot with the time variable as the x variable and the variable of interest as y . Set the “Display” option to either “Line” or “Both.” Click **Calculate**.
CrunchIt! cannot help with autocorrelations or the runs test.



TI-83/-84

You can create the time series plot using the connected Scatterplot ($\text{L}\wedge\text{L}$) option from 2nd Y= (STAT PLOTS). Define the plot with the time variable as the x variable and the variable of interest as y .

TI calculators cannot calculate the runs test for you.

Create a lagged time series using the following steps:

1. Place the cursor so it highlights the name of an empty list. Use 2nd n to enter the list name you wish to duplicate. The command at the bottom of the screen should look like $\text{L2} = \text{L1}$, for example.
2. The cursor will be on the first value in the new list. Press 2nd DEL =(Insert). This inserts a place holding 0 as the first entry.
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 STAT ▶ (Calc), 8:LinReg $a+bx$ using the original list as Y and the new list as X , for example, LinReg ($a+bx$) $\text{L2}, \text{L1}$.

TI calculators will not really help you with this chapter more than this, so will not be discussed further here.



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="1"` part tells R to connect the points. Note that no time variable is needed here.

```
> plot(var, type="1", xlab="time", ylab="Variable Name")
```

The runs test requires the `randtests` package. With that package installed and loaded, the simplest form of the command is

```
> Runs.test(series)
```

The command shown below calculates the autocorrelations, where “series” is 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.

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

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. If you select “Chart Output” in addition to the calculated values, you will get a graph showing the original series and the smoothed series.
5. Click **OK**.

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_{t21} .
3. Specify a cell in an empty column for the “Output Range.”
4. If you select “Chart Output” in addition to the calculated values, you will get a graph showing the original series and the smoothed series.
5. Click **OK**.

Excel does not give any calculated model, nor does it calculate the MAD, MAPE, or MSE as discussed in your text.



Use **Analyze → Time Series**. Select the time series variable and click **OK**. You may want to change the number of autocorrelations and partial autocorrelations calculated to a smaller number.

For simple 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 $y_{t2(k21)/2}$ instead of observation y_t as used in your text). This shows the smoothed series along with the original.

For simple 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, or select Unconstrained in the drop-down.
4. Click **Estimate**. The model summary with the diagnostic measures MAPE, MAE, AIC, and R^2 are added to the bottom of the output window.



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.

Minitab calculates all of MAD, MAPE, and MSE.



SPSS does not have a time series analysis in the base Statistics package. You need an additional time series license, which is not inexpensive.



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 & Data → Package Installer** and locate TTR. Once the package has been installed on your computer, click **Packages & Data → Package Manager** then locate TTR in the list and check the box labeled “Load” beside the package name. Note: You will have to load the package each time you use R when you want to do this type of analysis.) 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)
```

Regression-Based Forecasting Models

Models with seasonality 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.



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).

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

Simple linear (or quadratic) trends can be modeled with **Data → Data Analysis → Regression** using a time index (or the square of that index) as the X variable.

When considering seasonality, data files may have dates entered as, for example, 01/2019. 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 → Data Analysis → Regression** to compute the multiple regression using the original date variable and the indicators as predictors.



There are three options to create a time series graph and linear model. 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**.

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

1. Click **Cols → New Columns**. Click **OK**.
2. Right-click in the header of the new column. Give that column a name. Click **OK**.
3. With the name of the new column highlighted, click **Cols → Formula → Character → Substr**. The parameters are the variable name with the year and month, the starting position, and the number of characters to “strip” into the new column.
4. Click **Cols → New Columns**. In the box at the bottom, enter the number of new columns to add. Click **OK**.
5. With the name of one of the new columns highlighted, click **Cols → Formula → Conditional → If**. The parameters are the variable name with the month, a value of the month you are working with, 1 if that is true, and 0 otherwise. Click **OK**.
6. 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.



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.

Data files may have dates entered as, for example, 01/2010. We need to separate the month (or quarter) 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.



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.

Data files may have dates entered as, for example, 01/2010. We need to separate the month (or quarter) 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.



Simple trend models can be fit using **Statistics → Regression → Simple Linear** (or **Multiple Linear** to fit a quadratic model). Use the time variable as the independent variable and the variable of interest as the dependent variable.



The simple trend model can be fit using `lm` (linear model) command as detailed in Chapters 2, 12, and 13.

Data files may have dates entered as, for example, 01/2010. We need to separate the month (or quarter) 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)
```