CSC 120 (R Section, L0201), Spring 2016 — Assignment #2

Worth 10% of the course grade. Due by the start of class on March 22 (instructions for how to hand it in will be posted on the course web page, probably the same as for assignment 1). This assignment may be handed in late, with a 20% penalty, by start of class on March 25. Assignments will not usually be accepted after that. Contact the instructor as soon as possible if you have a legitimate excuse (eg, documented illness) for handing in the assignment late (without penalty).

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you shouldn't leave a discussion with someone else with any written notes (either paper or electronic).

In this assignment, you will write R functions for forecasting future values of a time series, and apply them to observations on numbers of deaths and maximum temperatures in Houston, Texas. Doing this will provide more practice in basic R programming, and on the use of data frames and of subscripts that are numeric or logical vectors.

The data is derived from that distributed by the U.S. National Morbidity Mortality Air Pollution Study, NMMAPS, with some strangely missing temperature values filled in by reference to `www.wunderground.com`. I have provided data from 1994-01-01 to 2000-12-31 as a file on the course web page, which may be read as a data frame as follows:

```
houston <- read.table("http://www.cs.utoronto.ca/~radford/csc120/houston",head=TRUE)
```

The rows of this data frame have data for successive days, with the row names being the dates. The column names are as follows:

`deaths`	Number of deaths in Houston that day
`tmax`	Maximum temperature that day, in degrees Farenheit
`day_of_year`	Day of the year, starting at 1 for January 1
`day_of_week`	Day of the week, 1 to 7, with 1=Sunday
`month`	Month day is in, 1 to 12, with 1=January

We wish to forecast the number of deaths and the maximum temperature for each day, based only on data before that day, except that when forecasting the number of deaths on a day, we may use the maximum temperature for that day, as well as previous days. (Of course, if we actually wished to forecast the number of deaths beforehand using the same day's maximum temperature, we'd have to substitute a temperature forecast for the actual temperature, but we won't worry about that for this assignment.)

Once you have produced forecasts for each day, you will produce plots of the forecasts, the actual values, and the errors in the forecasts. You will also evaluate how good these forecasts were in terms of the average absolute value of the error.

You should write several functions for forecasting the value of some variable on a single day. All these functions should take as their first arguments a data frame, containing variables that may be used in making the forecast, and as their second argument a series of past values for the variable being forecast (which will have at least one past value, and for some functions will have to have more than one past value). This forecasting function should return a forecast for

the next value in this series. The data frame will have at least as many rows as the length of the series plus one (so there will be values for the day for which the forecast is being made). The data frame may have additional rows, but they should not be looked at when making the forecast for this day.

You should write several forecasting functions of this sort, which make forecasts as follows:

`forecast_previous`	The value for the immediately previous day
`forecast_week_ago`	The value seven days previous
`forecast_mean`	The average of all previous values
`forecast_mean14`	The average of the values on the 14 previous days
`forecast_same_day_of_week`	Average of previous days that are the same day of the week
`forecast_same_month`	Average of previous days that are the same month
`forecast_similar_tmax`	For forecasting deaths only. The average of previous days for which the maximum temperature was no more than 4 degrees different from the maximum temperature this day

Note that the first four functions above (which should be very simple) will not actually look at the data frame that they are given as their first argument.

You should also write a function called `predictions`, which makes forecasts for all days from some start index to the last day for which data is provided. This function will take as its arguments a function to use for forecasting, a data frame with values that may be used for forecasting, the series of values for which forecasts are to be made, and the start point for making forecasts of this series. It should return a vector of forecasts for values in the series from the specified start point to the end.

These functions (plus the `predictions2` function described below) should be defined in a script file that does nothing except define these functions, since these functions are of general use, perhaps for other data sets than the one we are using for this assignment.

In another script file, you should read in the Houston data (with the command shown above), and make forecasts for the numbers of deaths and maximum temperatures in Houston. You should start your forecasts at the beginning of the third year of the data provided, so that all forecasts will have at least two years of data available (though not all the forecasting functions above will use all this data). In other words, you should set the "start" argument for `predictions` to $365 + 365 + 1$ (since 1994 and 1995 are not leap years); this should result in your making predictions for 1827 days.

You should make predictions for the numbers of deaths with each of the seven forecasting functions described above, and for maximum temperatures with the first six. For each of these 13 forecasts, you should produce (and hand in) a set of four plots — put together in one big plot, by using `par(mfrow=c(2,2))` — which show the actual values versus time index, the forecasts of these values vesus time index, the errors in these forecasts versus time index, and the errors in these forecasts versus day of the year. These plots should have appropriate titles that identify what they show. You may find it convenient to write a function that calls `predictions` and makes these plots, which should be defined in this script file, not the one with general funtions definitions, since it is specific to this assignment.

You should also output the average absolute value of the error for each of these forecasts.

Finally, you should try a more elaborate forecasting method, in which you first make forecasts with some method, and then try to use another method to forecast the error in the first method. The idea is that if you can manage to forecast the error well, you can get a better forecast by just adding the forecasted error to the original prediction.

You should write a `predictions2` function that does this. It should take *two* forecasting functions are arguments, along with a data frame of variable to use, a series of values to forecast, and *two* starting points, the second of which is later than the first. It should use the first forecasting function and the first starting point to make a first set of forecasts, then find the errors in these forecasts, and try to forecast these errors with the second forecasting function, starting at the second starting point. It should return forecasts (starting at the second starting point) that are equal to the first forecasts plus the forecast error in these forecasts.

You should try out `predictions2` for predicting deaths, with the first forecasting function being `forecast_same_month` and the second being `forecast_similar_tmax`, and produce plots and output as for the other forecasts.

sd1 ———> sd2 ————> ed

prediction(ff1, df, sd1, ed)

find errors between sd1 and ed

then we use errors from sd1 to sd2 to predict errors from sd2 to ed