Homework 7: 'Tis the season to be unemployed

36-350, Fall 2011

Due at 11:59 pm on Tuesday, 1 November 2011

INSTRUCTIONS: Refer to the previous homework for details on what is acceptable for submission.
Direct objective: Applying the split-apply-combine pattern with plyr.
Indirect objectives: Simple decomposition of seasonal effects and trends in time series.

Description The file `fredurn.rda` contains data on unemployment rate for the 50 United States provided by the U.S. Department of Labor: Bureau of Labor Statistics. The rates are provided monthly as a percent, and they go back from September, 2011 to October, 1981. We will examine the more recent past from 1991 to present. Review the slides from lectures 12 and 14, and install the required packages before beginning. Remember to consult the R Cookbook when stumped.

Required packages: `plyr`, `ggplot2`

Preamble Enter the following commands in R.

```
library(plyr)
library(ggplot2)
load('fredurn.rda')
urn <- subset(urn, year >= 1991)
```

This will load the required packages and a data frame named `urn`.

Data Each row of `urn` corresponds to a measurement of the unemployment rate for a particular state during a particular month and year. The columns of `urn` are variables that indicate the `date`, `state`, `month`, and `year` of the measurement `unemployment`.

Problems

1. Look at the data using the commands (1 at a time):

   ```
   qplot(x = date, y = unemployment, data = urn, geom = 'line',
         group = state, color = state, alpha = 2/3)
   qplot(x = date, y = unemployment, data = urn, geom = 'line',
         group = state, facets = ~ state)
   ```

   Describe these two plots. Hint: Make sure to load the `ggplot2` package. (5)

2. Extract the subset of `urn` corresponding to Texas (TX) into a data frame named `df` and make a plot of the unemployment rate series using the command:

   ```
   qplot(x = date, y = unemployment, data = df, geom = 'line')
   ```

   You should see 3 large bumps and many smaller bumps in this plot. What do the small bumps correspond to, i.e. what timescale are they on? (5)

3. Look at the monthly variation of unemployment rate for each of the 30 years with the command:

```
qplot(x = month, y = unemployment, data = df,
      geom = 'line', group = year, color = year)
```

Comparing monthly variation of unemployment with the yearly variation, you should see that the yearly variation is much larger, i.e. the change in unemployment rate across years is larger than the change in unemployment rate across months. This makes it hard to see the monthly variation in the plot that you created. We can try to remove the "effect" of the year by subtracting out the average unemployment rate for each year. Add a column to `df` named `unemployment.deyeared` that is the unemployment rate relative to the average for the year. You will need to compute the average for each year, and then for each row in `df` subtract the corresponding average from `unemployment`. Hint: Use `transform`. (15)

4. Plot the adjusted unemployment rates.

```
qplot(x = month, y = unemployment.deyeared, data = df,
      geom = 'line', group = year, color = year)
```

Explain what you see. Why is there such a large spread around January and December? (5)

5. A slightly more flexible approach to removing the effect of the year on the series is to allow, in addition to a different mean for each year, a linear trend within each year. We can accomplish this by fitting a separate (slope + intercept) linear model for each year. This is a model of the form

$$unemployment_{year,month} = \beta_{0,year} + \beta_{1,year} \times month + error$$

where *month* is the number of the month (1–12), β's are parameters of the model, and *error* is what is not accounted for by the model. Create a column `unemployment.res` in `df` with the residuals (actual - predicted) from the fit of this model using the `lm` function, and add another column `unemployment.fit` containing the fitted values (predictions from the model). Hint: If `df0` is a subset of `df` containing only a single year, then

```
resid(lm(unemployment ~ month, data = df0))
fitted(lm(unemployment ~ month, data = df0))
```

give the residuals and fitted values from a fit of a single year. Plot `unemployment.fit` as in Problem 4 and comment. (20)

6. Plot the newly adjusted unemployment rates in `unemployment.res` and compare with what you saw previously in Problem 4. Do you notice a seasonal pattern? Is it more clear than before? Compute the average for each month (call it `unemployment.monthlyeffect`) and plot it. Hint: Use `summarize`. (15)

7. Congratulations! You just detrended a single time series and estimated a seasonal effect. Now write a function so that your analysis (with a final result as in Problem 6) can be generalized to the other 49 states. This function should take a data frame (corresponding to a state) as input, and return a data frame with two columns: `month` (1–12) and `unemployment.monthlyeffect`. (15)

8. Test the function you wrote for Problem 7 on Wyoming (WY), plot the result, and comment. (5)

9. Apply the function you wrote in Problem 7 to all 50 states. Plot `unemployment.monthlyeffect` for all 50 states (as in Problem 1) and comment. Are different states different, i.e. is there variation among the states in terms of the seasonality of unemployment? (15)