

Sunrise-Sunset Calendar

William Kreuter

University of Washington

This program uses proc calendar to create a display of the current month. Each day's cell shows the times of sunrise and sunset for the location which is named in the TITLE statement and whose coordinates appear in the CARDS data. The three numbers in the CARDS statement are, respectively, latitude in degrees (north is positive), longitude in degrees (east is positive), and time zone in hours (east is positive; do not correct it for daylight savings time; the US Pacific time zone is thus "-8" during all months).

The program computes whether daylight savings time applies on each day according to current US law. However, locations using year-round standard time are unknown to the program and will have times displayed with daylight savings applied during the usual months.

Other months than the current one, or multiple months, can be computed by making minor modifications to the do-loop code.

The algorithm comes from a publication of the United States Naval Observatory. For the duration of the 20th and 21st centuries, it should be accurate to within three minutes for latitudes less extreme than 60°. This three-minute accuracy is with reference to the result of more precise algorithms, rather than the time you'd actually see the sun rise or set. The actual time is affected by a number of conditions, including hills, obstructions of the view at the horizon, weather and certain other atmospheric conditions. Some of these factors are estimated by this and the more accurate algorithms, but most cannot be conveniently modeled.

My program follows. It's an adaptation of a similar program presented at the 1996 PNWSUG, here rewritten with the nice new timeampm format available in SAS 6.12. I can supply this in machine-readable form; inquire at billyk@u.washington.edu.

```

options ps=59 ls=80;
data;
  input latitude longitud timezone;
  array sun{0:1} sunset sunrise;
  pi=arccos(-1);
  radian=pi/180;
  phi=latitude*radian;
  lambda=longitud/15;
  do date=intnx('month',today(),0) to intnx('month',today(),1)-1;
    n=date-intnx('year',date,0)+1;
    select (month(date));
      when (4) dst=date>=intnx('month',date,0)
        -weekday(intnx('month',date,0))+8;
      when (10) dst=date<intnx('month',date,1)
        -weekday(intnx('month',date,1))+1;
      otherwise dst=5<=month(date)<=9;
    end;
    do s=0 to 1;
      t=n+(18-12*s-lambda)/24;
      m=(.9856*t-3.289)*radian;
      l=m+(1.916*sin(m)+.02*sin(2*m)+282.634)*radian;
      delta=.39782*sin(l);
      sun{s}=3600*(mod(24*((1-2*s)
        *arccos((-0.01454-sin(delta)*sin(phi))
          /(cos(delta)*cos(phi)))
          /(2*pi)+s)+12*(atan(.91746*tan(l))/pi
            +(mod(floor(l*2/pi)+4,4)
              in (1,2)))-.06571*t-6.622
            -lambda+timezone+dst+48,24));
    end;
  output;
end;
format sunrise sunset timeampm8.;
keep date sunrise sunset;
cards;
47.612 -122.308 -8
;
title 'Seattle';

proc calendar legend;
  start date;
  var sunrise sunset;
run;

```