Online Web Based Supplemental Material of

Daylight time and energy: Evidence from an Australian experiment

Ryan Kellogg and *Hendrik Wolff*

Journal of Environmental Economics and Management 56 (2008) 207–220

This Online Appendix consists of the following subsections:

Appendix A: Data processing

Electricity data are missing for occasional half-hours, so we estimate the missing observations via interpolation using adjacent half hours. Weather data are also missing for some occasional hours as well for four entire days (none of which fall in within the treatment dates in any year, except for the air pressure variable). While we estimate weather for isolated missing hours via interpolation, we estimate weather for unobserved days via a regression analysis using information from the daily-level weather dataset. In the two primary states of interest, Victoria and South Australia, the missing values we impute do not comprise more than 0.6% of the data for any variable. Details and code for this procedure are available from the authors upon request.

Schedules for most school vacations, state holidays, and federal holidays were obtained from the Australian Federal Department of Employment and Workplace Relations, The Department of Education and Children's Services (SA), and The Department of Education and Training (VIC). For years in which information was not available from the above institutions, the dates were obtained by internet search. Employment data were obtained from the Australian Bureau of Statistics' Labor Force Spreadsheets, Table 12. Sunrise, sunset, and twilight data were sourced from the U.S. Naval Observatory, and the days and times of switches to and from DST were obtained from the Time and Date AS Company.

While our data are provided in Standard Time, we conduct our analysis in clock time. We therefore convert our data to clock time, which, for most affected observations, requires a simple one-hour shift. However, at the start of a DST period, the 02:00-03:00 interval (in clock time) is missing. To avoid a gap in our data, we duplicate the 01:30-02:00 information into the missing 02:00-02:30 half hour, and likewise equate the missing 02:30-03:00 period to our 03:00-03:30 observation. Further, when a DST period terminates, the 02:00-03:00 period (in clock time) is observed twice. Because our model is designed for only one observation in each half-hour, we average these observations.

Appendix B: Justification of using 12:00 to 14:30 as the control period

Our identification strategy uses the assumption that electricity demand in the mid-day is not affected by DST. The purpose of this appendix is to offer regression results to justify this assumption and to explain our choice of 12:00 to 14:30 as the mid-day control period in our analysis.

In VIC and SA, we observe "typical" switches from standard time to DST in late October of 1999 and 2001-2005. These observations allow us to examine DST's effect on mid-day electricity consumption by performing a regression discontinuity analysis of demand near the date of each switch. Specifically, we form a regression sample consisting of half-hourly demand observations during the week before and week after the switch to DST in each year. We then regress, separately for each half-hour, the logarithm of demand on an indicator variable for when DST is in effect, state-specific within-year time trends, fixed effects for the interaction of state and year, fixed effects for day of week, fixed effects for holidays and vacations, and weather variables.

Before discussing the estimated effect of DST in the mid-day, we first note that this specification produces estimates which show that DST increases demand in the morning and decreases demand in the evening. For example, during 07:00-07:30, we estimate that demand increases by 5.9% following the switch to DST, with a standard error of 1.0% (we report standard errors clustered on year, though these are not appreciably different from OLS standard errors). During 18:30-19:00 we estimate a decrease in demand of 4.9%, with a standard error of 1.7%. The signs and statistical significance of these results are consistent with intuition, and indicate that this specification has sufficient statistical power to resolve non-zero effects where they exist.

In the mid-day, however, the effect of switching to DST is statistically insignificant. Table B displays estimates of the percentage effect of switching to DST, along with standard errors, for several half-hour intervals. These results are robust to the addition of another week of data before and after each switch to DST, and to the addition of quadratic state-specific within-year time trends. The point estimates are smallest in magnitude from 12:30-14:00, and increase both before and particularly after this time period.

Selection of this 12:30-14:00 interval as the base period for estimation of the main specification, equation (1), ultimately yields an estimate of θ of +0.3% (which is statistically indistinguishable from zero). Expanding the base period symmetrically around 12:30-14:00 includes within it hours of the day in which our regression discontinuity estimates in table B indicate that DST is likely to increase demand. Therefore, as we expand the base period, the estimates of θ decrease. This is demonstrated by the fact that the reference case estimate of θ, which uses the longer 12:00-14:30 interval as the base period, is +0.1%. We choose this interval to be our base period, rather than 12:30-14:00, to be conservative in our final estimate.

Appendix C: Half-hourly estimation results

Table C displays the estimated percentage impact of the DST extension on electricity demand in each half hour: these are the point estimates given by $\exp(\hat{\beta}_h) - 1$ and correspond to figure 5. Note that the large effects in the late-night hours are caused by centralized off-peak water heaters in Melbourne [19]. These are triggered by timers set on Standard Time—groups of heaters are activated at 23:30 and 01:30. Each turns off on its own once its heating is complete. During the DST extension, each heater turns on one hour "late" (according to clock time). This drives the negative, then positive, overnight treatment effects. Regressing equation (1) in Standard Time, rather than clock time, eliminates these overnight effects, and produces a point estimate that the extension increased overall electricity consumption by 0.51%.

Appendix D: On Tourism in Australia

Figure D1 displays tourism data for VIC and SA, demonstrating that the 2000 Olympics did not significantly impact tourism in the third and fourth quarters of 2000. Tourism data for Sydney in NSW (Figure D2), however, shows that tourism increased in September 2000, and that there was no such increase in 1998 or 1999 (Australian Bureau of Statistics, 2001a, 2001b). Moreover, anecdotal evidence from Melbourne newspapers shows that Melbourne (the most frequently touristed location in VIC) did not experience any change in tourism before, during, or after the Olympic Games in 2000. Further details on tourism may be found in the Australian Bureau of Statistics' special report on Tourism related to the Olympics (2001b).

Appendix E: Stata Code for the estimation and the simulation of the CEC model

```
* CEC Model
clear
cd "C:\My Documents\Berkeley\ARE\Diss\Ideas\DST\AustraliaData\Analysis\CEC-Refurbished"
log using "C:\My Documents\Berkeley\ARE\Diss\Ideas\DST\AustraliaData\Analysis\CEC-
Refurbished\CEClogCEC.smcl", replace

set mem 800m
set more off
set matsize 2000
set maxvar 7000
use "C:\My Documents\Berkeley\ARE\Diss\Ideas\DST\AustraliaData\Analysis\DSTFullDataset.dta"


***************************
* Fill up data holes, such that VIC1 is (and partially SA1) complete data sets
************************************

sort Province Year Month Date Day Hour
drop if (Province =="SA1"| Province =="QLD1" | Province =="NSW1")

***************************
* Generate Variables that are needed for the CEC - SUR
************************************

* Re-define treatment variables to exclude Olympic period

gen TreatPerI=0
gen TreatPerII=0

replace TreatPerI=1 if (Province=="VIC1" | Province=="NSW1") & Year==2000 & Month ==8 & Day ==27
& Hour >=4
replace TreatPerI=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(28Aug2000) &
Date<=d(8Sep2000)
replace TreatPerI=1 if Province=="VIC1" & Date>=d(28Aug2000) & Date<=d(14Sep2000)
replace TreatPerII=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(7Oct2000) &
Date<=d(28Oct2000)
```

```
replace TreatPerII=1 if (Province=="VIC1" | Province=="NSW1") & Year==2000 & Month ==10 & Day
<=29 & Hour <=3
gen Treat = TreatPerI + TreatPerII

gen TreatPer2001I=0
gen TreatPer2001II=0
replace TreatPer2001I=1 if (Province=="VIC1" | Province=="NSW1") & Year==2001 & Month ==8 & Day
==26 & Hour >=4
replace TreatPer2001I=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(27Aug2001) &
Date<=d(8Sep2001)
replace TreatPer2001I=1 if Province=="VIC1" & Date>=d(27Aug2001) & Date<=d(14Sep2001)
replace TreatPer2001II=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(3Oct2001) &
Date<=d(27Oct2001)
replace TreatPer2001II=1 if (Province=="VIC1" | Province=="NSW1") & Year==2001 & Month ==10 & Day
<=28 & Hour <=3

gen TreatPer1999I=0
gen TreatPer1999II=0
replace TreatPer1999I=1 if (Province=="VIC1" | Province=="NSW1") & Year==1999 & Month ==8 & Day
==30 & Hour >=4
replace TreatPer1999I=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(31Aug1999) &
Date<=d(11Sep1999)
replace TreatPer1999I=1 if Province=="VIC1" & Date>=d(31Aug1999) & Date<=d(18Sep1999)
replace TreatPer1999II=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(3Oct1999) &
Date<=d(27Oct1999)
replace TreatPer1999II=1 if (Province=="VIC1" | Province=="NSW1") & Year==1999 & Month ==10 & Day
<=28 & Hour <=3


gen CECRegression =0
replace CECRegression =1 if (Province=="VIC1" | Province=="NSW1") & (Date<=d(26Aug2000) |
(Date>=d(30Oct2000) & Date<=d(31Dec2001)))



gen Weekday = dow(Date)
tab Weekday, gen(WeekdayD)
/* WeekdayD1=Sunday, ..., WeekdayD7=Saturday*/

sort Province Year Month Date Day Hour
gen TempShort = 0.45*Temp + 0.45*Temp[_n-1] + 0.10*Temp[_n-2]
gen TempShortQuadr  = TempShort*TempShort
gen TempShortCubic = TempShort*TempShort*TempShort

egen AvTemp49HouresT = ma(Temp), t(49)
gen TempMADay = AvTemp49HouresT[_n-24]
gen TempLong = 0.60*(TempMADay) + 0.30*(TempMADay[_n-48]) + 0.10*(TempMADay[_n-96])
gen TempLongH = 0
gen TempLongW = 0
gen TempLongC = 0
replace TempLongH = TempLong if Temp > 21.1111
replace TempLongW = TempLong if (Temp >= 10 & Temp <= 21.1111)
replace TempLongC = TempLong if Temp < 10
destring SunshineHrs, replace
gen SunTemp = SunshineHrs * Temp      /* sunshine reduces elec on cold days, increases it on hot
days */
drop AvTemp49HouresT
drop TempMADay TempLong

* Keep only VIC, and reshape wide by Hour (to enable SUR)
keep if inlist(Province, "VIC1")
sort Province Year Month Date Day Hour
/*
browse if  (Demand ==. |Price==. | Date ==. |Hour ==. |Month==. | Year ==. |Day ==. |DST ==.|Temp
==. |Wind ==. |Pressure==. | /*
*/ Precip ==. |SunshineHrs ==. |PctTwilight ==. |PctDaylight==. | Employment ==. |Olympic ==.
|OffDay ==. |TransDayOff ==. | /*
*/ TreatPerI ==. |TreatPerII ==. |Treat ==. |Weekday==. | WeekdayD1 ==. |WeekdayD2 ==. |WeekdayD3
==. |WeekdayD4==. | WeekdayD5==. | WeekdayD6==. | WeekdayD7 ==. |TempShort ==. |TempShortQuadr
==. |TempShortCubic ==. |TempLongH==. | TempLongW ==. |TempLongC ==. |SunTemp==.)  & (Province
=="VIC1")
```

```
*/
*Check to see missing data
* egen a=rmiss(Province-SunTemp)
* tab a
* browse if a>=1 & Province=="VIC1"
* drop a

*********************************************
/* This is for the SUR - Regression, such that observations are ligned up by clock-time and not
by Standard Time !!  */
*********************************************

*Generate a Clock Time Variable from 0-47
gen DST_Hour=Hour[_n+2] if DST ==1
replace DST_Hour=Hour if DST ==0

*At the end of DST period, clock time 5 and 6 is observed twice.
* Solution: Take the average of the half hours
foreach var of varlist Demand Price Temp Humidity Wind Pressure Precip PctTwilight PctDaylight
TreatPerI TreatPerII /*
                   */ TreatPer2001I TreatPer2001II TreatPer1999I TreatPer1999II  Treat
TempShort TempShortQuadr TempShortCubic TempLongH TempLongW TempLongC SunTemp {

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(28Mar1999) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(28Mar1999) & DST_Hour ==6 &
Province =="VIC1"

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(26Mar2000) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(26Mar2000) & DST_Hour ==6 &
Province =="VIC1"

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(25Mar2001) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(25Mar2001) & DST_Hour ==6 &
Province =="VIC1"

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(31Mar2002) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(31Mar2002) & DST_Hour ==6 &
Province =="VIC1"

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(30Mar2003) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(30Mar2003) & DST_Hour ==6 &
Province =="VIC1"

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(28Mar2004) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(28Mar2004) & DST_Hour ==6 &
Province =="VIC1"

replace `var'= (`var' + `var'[_n+1])/2 if DST == 1 & Date ==d(27Mar2005) & DST_Hour ==5 &
Province =="VIC1"
replace `var'= (`var' + `var'[_n-1])/2 if DST == 0 & Date ==d(27Mar2005) & DST_Hour ==6 &
Province =="VIC1"
}
* Finally we need to drop the Half Hours 5 and 6 for the end of DST that appear twice.
drop if DST == 1 & Date ==d(28Mar1999) & DST_Hour ==6
drop if DST == 0 & Date ==d(28Mar1999) & DST_Hour ==5

drop if DST == 1 & Date ==d(26Mar2000) & DST_Hour ==6
drop if DST == 0 & Date ==d(26Mar2000) & DST_Hour ==5

drop if DST == 1 & Date ==d(25Mar2001) & DST_Hour ==6
drop if DST == 0 & Date ==d(25Mar2001) & DST_Hour ==5

drop if DST == 1 & Date ==d(31Mar2002) & DST_Hour ==6
drop if DST == 0 & Date ==d(31Mar2002) & DST_Hour ==5
drop if DST == 1 & Date ==d(30Mar2003) & DST_Hour ==6
drop if DST == 0 & Date ==d(30Mar2003) & DST_Hour ==5

drop if DST == 1 & Date ==d(28Mar2004) & DST_Hour ==6
drop if DST == 0 & Date ==d(28Mar2004) & DST_Hour ==5

drop if DST == 1 & Date ==d(27Mar2005) & DST_Hour ==6
drop if DST == 0 & Date ==d(27Mar2005) & DST_Hour ==5


*At the beginning of DST period, clock time 4 and 5 are missing
* Solution: Use data of clocktime 3 and dublicate to clocktime 4 & use data of clocktime 6 to
dublicate to clocktime 5.

expand 2 if Date==d(31Oct1999) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"
expand 2 if Date==d(27Aug2000) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"
expand 2 if Date==d(28Oct2001) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"
expand 2 if Date==d(27Oct2002) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"
expand 2 if Date==d(26Oct2003) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"
expand 2 if Date==d(31Oct2004) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"
expand 2 if Date==d(30Oct2005) & (DST_Hour==6 |DST_Hour==3) & Province =="VIC1"

sort Province Year Month Day Hour DST_Hour /*Attention, Don't Sort by DST_Hour first, but by Hour
!!! (otherwise Last hour of the day becomes first hour */

replace DST_Hour = DST_Hour+1 if (Date ==d(31Oct1999) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(31Oct1999) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = DST_Hour+1 if (Date ==d(27Aug2000) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(27Aug2000) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = DST_Hour+1 if (Date ==d(28Oct2001) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(28Oct2001) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = DST_Hour+1 if (Date ==d(27Oct2002) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(27Oct2002) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = DST_Hour+1 if (Date ==d(26Oct2003) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(26Oct2003) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = DST_Hour+1 if (Date ==d(31Oct2004) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(31Oct2004) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = DST_Hour+1 if (Date ==d(30Oct2005) & DST_Hour ==3) & DST_Hour[_n-2] == 2 &
Province =="VIC1"
replace DST_Hour = DST_Hour-1 if (Date ==d(30Oct2005) & DST_Hour ==6) & DST_Hour[_n+2] == 7 &
Province =="VIC1"

replace DST_Hour = 0 if Hour == 46 & Date ==d(31Dec2005) & Province=="VIC1"
replace DST_Hour = 1 if Hour ==47 & Date ==d(31Dec2005) & Province=="VIC1"


/* In order that the reshape works, generate now a DST consistent Date Year Month and Day etc. */
replace Date = Date+1 if DST == 1 & (DST_Hour==0 | DST_Hour ==1)
format Date %dD_m_Y
drop Year Month Day
gen Year = year(Date)
gen Month = month(Date)
gen Day = day(Date)
gen Workday = 1-OffDay



foreach var of varlist SunshineHrs Employment Olympic OffDay Workday TransDayOff CECRegression
Weekday WeekdayD1 WeekdayD2 /*
*/ WeekdayD3 WeekdayD4 WeekdayD5 WeekdayD6 WeekdayD7 SchoolVac Holiday{
replace `var'= `var' if DST ==0
replace `var'= `var'[_n+2] if DST ==1
}


drop if Date>=d(1Jan2003)
drop Hour
reshape wide  Demand Price DST Temp Humidity Wind Pressure Precip PctTwilight PctDaylight
TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I TreatPer1999II  Treat /*
*/ TempShort TempShortQuadr TempShortCubic TempLongH TempLongW TempLongC SunTemp, /*
*/ i(Province Year Month  Date  Day) j(DST_Hour)

* Check to see missing data
* egen b=rmiss(Province-WeekdayD7)
* tab b
* browse if b>1 & Province=="VIC1"





drop if Month <=3
drop if Month >=10
drop if Month == 7
drop if Month ==8 & Day <=14






* Instead of "Visibility" and "Cloud Cover" used in the 2001 CEC study, we us "SunshineHrs" and
"SunTemp0..SunTemp47" as a proxy.

/* Define the Regressors into local `X0' to `X47' as locals */
forvalues i=0(1)47 {
local X`i' TempShort`i'       TempShortQuadr`i'      TempShortCubic`i'      TempLongH`i'
TempLongW`i' TempLongC`i' /*
*/      SunTemp`i' Humidity`i' Precip`i' Pressure`i'  Wind`i' PctTwilight`i' PctDaylight`i' /*
*/    SunshineHrs WeekdayD*  Employment Workday
}




/*SUR Estimation */
sureg   (       Demand0 `X0'    )       /*
*/      (       Demand1 `X1'    )       /*
*/      (       Demand2 `X2'    )       /*
*/      (       Demand3 `X3'    )       /*
*/      (       Demand4 `X4'    )       /*
*/      (       Demand5 `X5'    )       /*
*/      (       Demand6 `X6'    )       /*
*/      (       Demand7 `X7'    )       /*
*/      (       Demand8 `X8'    )       /*
*/      (       Demand9 `X9'    )       /*
*/      (       Demand10       `X10'   )       /*
*/      (       Demand11       `X11'   )       /*
*/      (       Demand12       `X12'   )       /*
*/      (       Demand13       `X13'   )       /*
*/      (       Demand14       `X14'   )       /*
*/      (       Demand15       `X15'   )       /*
*/      (       Demand16       `X16'   )       /*
*/      (       Demand17       `X17'   )       /*
*/      (       Demand18       `X18'   )       /*
*/      (       Demand19       `X19'   )       /*
*/      (       Demand20        `X20'   )       /*
*/      (       Demand21        `X21'   )       /*
*/      (       Demand22        `X22'   )       /*
*/      (       Demand23        `X23'   )       /*
*/      (       Demand24        `X24'   )       /*
*/      (       Demand25        `X25'   )       /*
*/      (       Demand26        `X26'   )       /*
*/      (       Demand27        `X27'   )       /*
*/      (       Demand28        `X28'   )       /*
*/      (       Demand29        `X29'   )       /*
*/      (       Demand30        `X30'   )       /*
*/      (       Demand31        `X31'   )       /*
*/      (       Demand32        `X32'   )       /*
*/      (       Demand33        `X33'   )       /*
*/      (       Demand34        `X34'   )       /*
*/      (       Demand35        `X35'   )       /*
*/      (       Demand36        `X36'   )       /*
*/      (       Demand37        `X37'   )       /*
*/      (       Demand38        `X38'   )       /*
*/      (       Demand39        `X39'   )       /*
*/      (       Demand40        `X40'   )       /*
*/      (       Demand41        `X41'   )       /*
*/      (       Demand42        `X42'   )       /*
*/      (       Demand43        `X43'   )       /*
*/      (       Demand44        `X44'   )       /*
*/      (       Demand45        `X45'   )       /*
*/      (       Demand46        `X46'   )       /*
*/      (       Demand47        `X47'   )       /*
*/  if CECRegression==1 & Province=="VIC1"


estimates store CEC

drop if Date >=d(1Jan2003)

/*Precict, what would have happened with DST, two month earlier in year 2000 (simulated DST)*/
* Here DSim`i'p         : predicted Demand_hat by SUR model
*          DSim`i'r     : predicted residuals from SUR errors: Attention, in "out of sample"
DSim`i'r  is difference between y_hat and y_observed!
*          DSim`i'wW    : Demand without Weather: predicted Demand whereby all Weather and
Lighting coefficients from SUR are set to 0
*          DSim`i'      : Simulated Demand whereby all Weather and Lighting variables are laged by
one hour
*          DSimwR`i'    : Simulated Demand + residuals from SUR, whereby all Weather and Lighting
variables are laged by one hour



/* For HalfHour 0 , we have to wrap around the day (for DSim), rest remains the same as below
forvalues loop */
predict DSim0p, eq(Demand0)
predict DSim0r, resid eq(Demand0)
gen DSim0wW = DSim0p  - ([Demand0]_b[TempShort0]*TempShort0
+[Demand0]_b[TempShortQuadr0]*TempShortQuadr0
+[Demand0]_b[TempShortCubic0]*TempShortCubic0         +[Demand0]_b[TempLongH0]*TempLongH0
+[Demand0]_b[TempLongW0]*TempLongW0         +[Demand0]_b[TempLongC0]*TempLongC0
+[Demand0]_b[SunTemp0]*SunTemp0         +[Demand0]_b[Humidity0]*Humidity0
+[Demand0]_b[Precip0]*Precip0          +[Demand0]_b[Pressure0]*Pressure0
+[Demand0]_b[Wind0]*Wind0        +[Demand0]_b[PctTwilight0]*PctTwilight0+
[Demand0]_b[PctDaylight0]*PctDaylight0)
gen DSim0 = DSim0wW    +([Demand0]_b[TempShort0]*TempShort46[_n-
1]+[Demand0]_b[TempShortQuadr0]*TempShortQuadr46[_n-
1]+[Demand0]_b[TempShortCubic0]*TempShortCubic46[_n-1]+[Demand0]_b[TempLongH0]*TempLongH46[_n-
1]+[Demand0]_b[TempLongW0]*TempLongW46[_n-1]+[Demand0]_b[TempLongC0]*TempLongC46[_n-
1]+[Demand0]_b[SunTemp0]*SunTemp46[_n-1]+[Demand0]_b[Humidity0]*Humidity46[_n-
1]+[Demand0]_b[Precip0]*Precip46[_n-1]+[Demand0]_b[Pressure0]*Pressure46[_n-
1]+[Demand0]_b[Wind0]*Wind46[_n-1]+[Demand0]_b[PctTwilight0]*PctTwilight46[_n-1]+
[Demand0]_b[PctDaylight0]*PctDaylight46[_n-1])
gen DSimwR0 = DSim0 + DSim0r
drop DSim0wW DSim0r DSim0p
```

```
/* For HalfHour 0 , we have to wrap around the day (for DSim), rest remains the same as below
forvalues loop */
predict DSim1p, eq(Demand1)
predict DSim1r, resid eq(Demand1)
gen DSim1wW = DSim1p  - ([Demand1]_b[TempShort1]*TempShort1
+[Demand1]_b[TempShortQuadr1]*TempShortQuadr1
+[Demand1]_b[TempShortCubic1]*TempShortCubic1         +[Demand1]_b[TempLongH1]*TempLongH1
+[Demand1]_b[TempLongW1]*TempLongW1         +[Demand1]_b[TempLongC1]*TempLongC1
+[Demand1]_b[SunTemp1]*SunTemp1         +[Demand1]_b[Humidity1]*Humidity1
+[Demand1]_b[Precip1]*Precip1          +[Demand1]_b[Pressure1]*Pressure1
+[Demand1]_b[Wind1]*Wind1        +[Demand1]_b[PctTwilight1]*PctTwilight1+
[Demand1]_b[PctDaylight1]*PctDaylight1)
gen DSim1 = DSim1wW    +([Demand1]_b[TempShort1]*TempShort47[_n-
1]+[Demand1]_b[TempShortQuadr1]*TempShortQuadr47[_n-
1]+[Demand1]_b[TempShortCubic1]*TempShortCubic47[_n-1]+[Demand1]_b[TempLongH1]*TempLongH47[_n-
1]+[Demand1]_b[TempLongW1]*TempLongW47[_n-1]+[Demand1]_b[TempLongC1]*TempLongC47[_n-
1]+[Demand1]_b[SunTemp1]*SunTemp47[_n-1]+[Demand1]_b[Humidity1]*Humidity47[_n-
1]+[Demand1]_b[Precip1]*Precip47[_n-1]+[Demand1]_b[Pressure1]*Pressure47[_n-
1]+[Demand1]_b[Wind1]*Wind47[_n-1]+[Demand1]_b[PctTwilight1]*PctTwilight47[_n-1]+
[Demand1]_b[PctDaylight1]*PctDaylight47[_n-1])
gen DSimwR1 = DSim1 + DSim1r
drop DSim1wW DSim1r DSim1p

/* Finally do predictions for the half-hour 2-47, more efficient in loop */
forvalues i = 2(1)47 {
local j = `i'-2
predict DSim`i'p, eq(Demand`i')
predict DSim`i'r, resid eq(Demand`i')
gen DSim`i'wW = DSim`i'p  - ([Demand`i']_b[TempShort`i']*TempShort`i'
+[Demand`i']_b[TempShortQuadr`i']*TempShortQuadr`i'
+[Demand`i']_b[TempShortCubic`i']*TempShortCubic`i'
+[Demand`i']_b[TempLongH`i']*TempLongH`i'         +[Demand`i']_b[TempLongW`i']*TempLongW`i'
+[Demand`i']_b[TempLongC`i']*TempLongC`i'         +[Demand`i']_b[SunTemp`i']*SunTemp`i'
+[Demand`i']_b[Humidity`i']*Humidity`i'         +[Demand`i']_b[Precip`i']*Precip`i'
+[Demand`i']_b[Pressure`i']*Pressure`i'         +[Demand`i']_b[Wind`i']*Wind`i'
+[Demand`i']_b[PctTwilight`i']*PctTwilight`i'+
[Demand`i']_b[PctDaylight`i']*PctDaylight`i')
gen DSim`i' = DSim`i'wW
+([Demand`i']_b[TempShort`i']*TempShort`j'+[Demand`i']_b[TempShortQuadr`i']*TempShortQuadr`j'+[De
mand`i']_b[TempShortCubic`i']*TempShortCubic`j'+[Demand`i']_b[TempLongH`i']*TempLongH`j'+[Demand`
i']_b[TempLongW`i']*TempLongW`j'+[Demand`i']_b[TempLongC`i']*TempLongC`j'+[Demand`i']_b[SunTemp`i
']*SunTemp`j'+[Demand`i']_b[Humidity`i']*Humidity`j'+[Demand`i']_b[Precip`i']*Precip`j'+[Demand`i
']_b[Pressure`i']*Pressure`j'+[Demand`i']_b[Wind`i']*Wind`j'+[Demand`i']_b[PctTwilight`i']*PctTwi
light`j'+ [Demand`i']_b[PctDaylight`i']*PctDaylight`j')
gen DSimwR`i' = DSim`i' + DSim`i'r
drop DSim`i'wW DSim`i'p DSim`i'r
}

gen index = _n
sort index
save startAnalysis.dta, replace

matrix covb = e(V)
matrix coef = e(b)'
drop *
svmat covb
svmat coef
gen index = _n
sort index
save matricesSUR.dta, replace



/*

*********** start the looop !
forvalues looop=1(2)7 {
clear
use matricesSUR.dta
sort index
mkmat covb*, matrix(covb)
mkmat coef, matrix(coef)
drop covb* coef

merge index using startAnalysis.dta
drop _merge index
local loop = 1
* here loop just with 2 o (above with 3!)
gen test = `looop'

if test[1]==1 {
keep if Year==1999 & Month ==9 & Province =="VIC1"
                   }
if test[1]==2 {
keep if Year==1999 & Month ==10 & Date<=d(31Oct1999) & Province =="VIC1"
                   }
if test[1]==3 {
keep if (TreatPer1999I1 == 1  | TreatPer1999I47 == 1)
                }
if test[1]==4 {
keep if (TreatPer2001II1 == 1 | TreatPer2001II47 == 1)
                }
if test[1]==5 {
keep if Year==2001 & Month ==9 & Province =="VIC1"
                   }
if test[1]==6 {
keep if Year==2001 & Month ==10 & Date <=d(28Oct2001) & Province =="VIC1"
                   }
if test[1]==7 {
keep if (TreatPer2001I1 == 1  | TreatPer2001I47 == 1)
                }



matrix rbhh = rowsof(coef)/48
svmat rbhh


reshape long Demand Price DST Temp Humidity Wind Pressure Precip PctTwilight PctDaylight
TreatPerI TreatPerII TreatPer1999I TreatPer1999II TreatPer2001I TreatPer2001II Treat /*
                   */   TempShort TempShortQuadr TempShortCubic TempLongH TempLongW TempLongC
SunTemp DSim DSimwR, /*
*/ i(Province Year Month Date Day) j(DST_Hour)

local WandL TempShort TempShortQuadr  TempShortCubic TempLongH      TempLongW      TempLongC
       SunTemp Humidity       Precip  Pressure       Wind    PctTwilight    PctDaylight

foreach var of varlist `WandL' {
gen Sim`var'=`var'[_n-2]
}

global SimWandL SimTempShort SimTempShortQuadr      SimTempShortCubic      SimTempLongH
       SimTempLongW   SimTempLongC   SimSunTemp     SimHumidity    SimPrecip      SimPressure
       SimWind SimPctTwilight SimPctDaylight


reshape wide  Demand Price DST Temp Humidity Wind Pressure Precip PctTwilight PctDaylight
TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I TreatPer1999II  Treat /*
*/ TempShort TempShortQuadr TempShortCubic TempLongH TempLongW TempLongC SunTemp DSim DSimwR
$SimWandL , /*
*/ i(Province Year Month  Date  Day) j(DST_Hour)


******************************************
/* Calculate the Standard Errors associated with the predictions! */

gen ONE = 1
global rbb=rbhh[1]
/* generate rbhh vectors of zeroes (for the block-diagonal)*/
forvalues i = 1/$rbb {
gen ZEROS`i'=0
}
```

```
mkmat ZEROS*, matrix(Z)


/* Define Weather and Light Simulation Values into local `SWL' */
forvalues i=0(1)47 {
local SWL`i'   SimTempShort`i'       SimTempShortQuadr`i'   SimTempShortCubic`i'
        SimTempLongH`i'       SimTempLongW`i'       SimTempLongC`i'       SimSunTemp`i'
        SimHumidity`i' SimPrecip`i'   SimPressure`i' SimWind`i'     SimPctTwilight`i'
        SimPctDaylight`i'
}

/* Make 48 Matrices XSim (Simulated Weather and Light & daily regressors) from WandL by adding
the Daily Regressors plus constant*/
forvalues i=0(1)47 {
mkmat  `SWL`i'' SunshineHrs WeekdayD*  Employment Workday ONE, matrix(XSim`i')
}


/* partition coefficent vector into 48 subvectors */
forvalues i=0(1)47 {
 matrix coef`i' = coef[(1+`i'*rbhh[1])..(`i'+1)*rbhh[1],1]
}



/* Test if Matrix OK: Predict DSim and compare to above variables result D */

forvalues i=0(1)47 {
 matrix ds`i' = XSim`i'*coef`i'
svmat ds`i'
rename ds`i'1 ds`i'
gen control`i' =ds`i'-DSim`i'
}

matrix Xtild0 = XSim0, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild1 = Z, XSim1, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild2 = Z, Z, XSim2, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild3 = Z, Z, Z, XSim3, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild4 = Z, Z, Z, Z, XSim4, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild5 = Z, Z, Z, Z, Z, XSim5, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild6 = Z, Z, Z, Z, Z, Z, XSim6, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild7 = Z, Z, Z, Z, Z, Z, Z, XSim7, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild8 = Z, Z, Z, Z, Z, Z, Z, Z, XSim8, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild9 = Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim9, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild10 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim10, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild11 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim11, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild12 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim12, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild13 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim13, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild14 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim14, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild15 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim15, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild16 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim16, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild17 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim17, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
```

```
 matrix Xtild18 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim18, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild19 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim19, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild20 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim20, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild21 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim21, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild22 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim22, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild23 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim23, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild24 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim24,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild25 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
XSim25, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild26 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
XSim26, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild27 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, XSim27, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild28 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, XSim28, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild29 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, XSim29, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild30 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, XSim30, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild31 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, XSim31, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild32 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, XSim32, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild33 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, XSim33, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild34 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, XSim34, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild35 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim35, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild36 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim36, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild37 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim37, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild38 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim38, Z, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild39 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim39, Z, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild40 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim40, Z, Z, Z, Z, Z, Z, Z
 matrix Xtild41 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim41, Z, Z, Z, Z, Z, Z
 matrix Xtild42 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim42, Z, Z, Z, Z, Z
 matrix Xtild43 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim43, Z, Z, Z, Z
 matrix Xtild44 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim44, Z, Z, Z
 matrix Xtild45 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim45, Z, Z
 matrix Xtild46 = Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim46, Z
 matrix Xtild47 = Z , Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z,
Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, Z, XSim47

matrix Xtild =
Xtild0\Xtild1\Xtild2\Xtild3\Xtild4\Xtild5\Xtild6\Xtild7\Xtild8\Xtild9\Xtild10\Xtild11\Xtild12\Xti
ld13\ /*
*/ Xtild14     \Xtild15       \Xtild16       \Xtild17       \Xtild18       \Xtild19
        \Xtild20       \Xtild21       \Xtild22       \Xtild23 /*
*/ \Xtild24    \Xtild25       \Xtild26       \Xtild27       \Xtild28       \Xtild29
        \Xtild30       \Xtild31       \Xtild32       \Xtild33 /*
*/ \Xtild34    \Xtild35       \Xtild36       \Xtild37       \Xtild38       \Xtild39
        \Xtild40       \Xtild41       \Xtild42       \Xtild43 /*
*/ \Xtild44    \Xtild45       \Xtild46       \Xtild47
```

```
matrix COV=Xtild*covb*Xtild'

*******************************************************

reshape long Demand Price DST Temp Humidity Wind Pressure Precip PctTwilight PctDaylight
TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I TreatPer1999II  Treat /*
                   */   TempShort TempShortQuadr TempShortCubic TempLongH TempLongW TempLongC
SunTemp DSim DSimwR, /*
*/ i(Province Year Month Date Day) j(DST_Hour)


svmat COV
egen rCOVtot= rowtotal(COV*)
egen DSimtot_Var=total(rCOVtot)

egen DSimTreat  = sum(DSim)
egen DSimTreatwR = sum(DSimwR)
egen DObsTreat =  sum(Demand)

gen DSimwRtot_t_neutral =  (DSimTreatwR-DObsTreat)/sqrt(DSimtot_Var)
gen DSimtot_t_neutral =  (DSimTreat-DObsTreat)/sqrt(DSimtot_Var)

gen DSimwRtot_t_m6pc =  (DSimTreatwR-(DObsTreat*0.994))/sqrt(DSimtot_Var)
gen DSimtot_t_m6pc =  (DSimTreat-(DObsTreat*0.994))/sqrt(DSimtot_Var)

gen DSimwRtot_t_mhpc =  (DSimTreatwR-(DObsTreat*0.995))/sqrt(DSimtot_Var)
gen DSimtot_t_mhpc =  (DSimTreat-(DObsTreat*0.995))/sqrt(DSimtot_Var)

gen DSimwRtot_t_m1pc =  (DSimTreatwR-(DObsTreat*0.99))/sqrt(DSimtot_Var)
gen DSimtot_t_m1pc =  (DSimTreat-(DObsTreat*0.99))/sqrt(DSimtot_Var)

gen PerCh  =(DSimTreat-DObsTreat)/DObsTreat*100
gen PerChwR=(DSimTreatwR-DObsTreat)/DObsTreat*100



** The following Table displays the percentage changes between the CEC-simulation outcomes and
the observed Demand
** for the Treatment Period ,  first for DSim and then for wR (With Residual) respectively
sum PerCh PerChwR

** The following Table displays the t-values for electricity neutrality, -1% and -0.5%:
** difference between the CEC-simulation outcomes and the observed Demand
** for the Treatment , first for DSim and then for wR (With Residual) respectively
sum DSimtot_t_neutral DSimtot_t_mhpc DSimtot_t_m1pc
sum DSimwRtot_t_neutral DSimwRtot_t_mhpc DSimwRtot_t_m1pc

save data`looop'output.dta , replace
}


/*
clear
set mem 800m
set more off
set matsize 2000
set maxvar 7000
use startAnalysis.dta, clear

reshape long Demand Price DST Temp Humidity Wind Pressure Precip PctTwilight PctDaylight
TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I TreatPer1999II  Treat /*
                   */   TempShort TempShortQuadr TempShortCubic TempLongH TempLongW TempLongC
SunTemp DSim DSimwR, /*
*/ i(Province Year Month Date Day) j(DST_Hour)

sort Province Year Month Day DST_Hour
gen a = _n


***********************
```

```
* Daily Load Shape Graphs for 2000
***********************

twoway line Demand a if Province=="VIC1" &  TreatPerI==1  || /*
        */ line DSim a if Province=="VIC1" &  TreatPerI==1, yaxis(2) clcolor(orange) /*
        */ title("DST plot--Vicoria TreatPerI") /*
        */ legend(on order(1 "Demand" 2 "DSim"))
graph export VICDemandsTreatPerI.tif, replace

twoway line Demand a if Province=="VIC1" &  TreatPerII==1  || /*
        */ line DSim a if Province=="VIC1" &  TreatPerII==1, yaxis(2) clcolor(orange) /*
        */ title("DST plot--Vicoria TreatPerII") /*
        */ legend(on order(1 "Demand" 2 "DSim"))
graph export VICDemandsTreatPerII.tif, replace

***********************
* Daily Load Shape Graphs for 2001
***********************

* September Graph
***********************
twoway line Demand a if Year==2001 & Month ==9 & Province =="VIC1"|| /*
        */ line DSimwR a if Year==2001 & Month ==9 & Province =="VIC1", clcolor(orange) /*
        */ title("DST plot--Vicoria Sept2001 WITH RESIDUAL") /*
        */ legend(on order(1 "Demand" 2 "DSimwR"))
graph export VICDemandswRSep01.tif, replace

twoway line Demand a if Year==2001 & Month ==9 & Province =="VIC1"|| /*
        */ line DSim a if Year==2001 & Month ==9 & Province =="VIC1", clcolor(orange) /*
        */ title("DST plot--Vicoria Sept2001 NO Residual") /*
        */ legend(on order(1 "Demand" 2 "DSim"))
graph export VICDemandsSep01.tif, replace

* October Graph
***********************
twoway line Demand a  if Year==2001 & Month ==10 & Day<=28  &Province =="VIC1"|| /*
        */ line DSimwR a if Year==2001 & Month ==10 & Day<=28  &Province =="VIC1",
clcolor(orange) /*
        */ title("DST plot--Vicoria October 2001 WITH RESIDUAL") /*
        */ legend(on order(1 "Demand" 2 "DSimwR"))
graph export VICDemandswROct01.tif, replace
twoway line Demand a if Year==2001 & Month ==10 & Day<=28  &Province =="VIC1"|| /*
        */ line DSim a if Year==2001 & Month ==10 & Day<=28  &Province =="VIC1", clcolor(orange)
/*
        */ title("DST plot--Vicoria October 2001 NO Residual") /*
        */ legend(on order(1 "Demand" 2 "DSim"))
graph export VICDemandsOct01.tif, replace
save TempLong.dta, replace

***********************
* Average Load Shape Graphs for 2000
***********************
*/
use TempLong.dta, clear
sort Province Year Month Day DST_Hour
gen DSimStandarTime = DSim[_n-2]
sort Province DST_Hour Year Month
collapse(mean) Demand DSimStandarTime DSim DSimwR, by(Province DST_Hour Year Month)


sort Province Year Month DST_Hour


twoway line Demand DST_Hour if Month==9 & Year==2000& Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSimStandarTime DST_Hour if Month==9 & Year==2000 & Province=="VIC1",
clcolor(red) /*
        */ title("DST plot--Vicoria September 2000") /*
        */ legend(on order(1 "2000 Demand" 2 "2000 DSim in Standard Time"))
graph export VICAverDemandsStandardTime00.tif, replace
```

```
twoway line Demand DST_Hour if Month==9 & Year==2000& Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSim DST_Hour if Month==9 & Year==2000 & Province=="VIC1", clcolor(red) /*
        */ title("DST plot--Vicoria September 2000") /*
        */ legend(on order(1 "2000 Demand" 2 "2000 DSim"))
graph export VICAverDemands00.tif, replace


/*
twoway line Demand DST_Hour if Month==9 & Year==1999 & Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSimwR DST_Hour if Month==9 & Year==1999 & Province=="VIC1", clcolor(red) /*
        */ title("DST plot--Vicoria September 1999") /*
        */ legend(on order(1 "1999 Demand" 2 "1999 DSimWR"))
graph export VICAverDemandswR99.tif, replace







twoway line Demand DST_Hour if Month==9 & Year==2001 & Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSimwR DST_Hour if Month==9 & Year==2001 & Province=="VIC1", clcolor(red)  /*
        */ title("DST plot--Vicoria September") /*
        */ legend(on order(1 "Sep2001 Demand" 2 "Sep2001 DSimWR"))
graph export VICAverDemandswRSep01.tif, replace
twoway line Demand DST_Hour if Month==10  & Year==2001 & Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSimwR DST_Hour if Month==10 & Year==2001 & Province=="VIC1", clcolor(red)  /*
        */ title("DST plot--Vicoria October") /*
        */ legend(on order(1 "Oct2001 Demand" 2 "Oct2001 DSimWR"))
graph export VICAverDemandswROct01.tif, replace
twoway line Demand DST_Hour if Month==9 & Year==2001 & Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSim DST_Hour if Month==9 & Year==2001 & Province=="VIC1", clcolor(red)  /*
        */ title("DST plot--Vicoria September") /*
        */ legend(on order(1 "Sep2001 Demand" 2 "Sep2001 DSim (NO Residual)"))
graph export VICAverDemandsSep01.tif, replace

***********
*this following graph does not make sense due to adding of Residial at DSim in the period were
Standard Time is not observed in 2000.
*rather consistency check,
***********
/*
twoway line Demand DST_Hour if Month==9 & Year==2000& Province=="VIC1", clcolor(blue)
clpat(dash)|| /*
        */ line DSimwR DST_Hour if Month==9 & Year==2000 & Province=="VIC1", clcolor(red) /*
        */ title("DST plot--Vicoria September 2000") /*
        */ legend(on order(1 "2000 Demand" 2 "2000 DSimWR"))
graph export VICAverDemandswR00.tif, replace
*/

 cd "C:\My Documents\Berkeley\ARE\Diss\Ideas\DST\AustraliaData\Analysis"
*log close
*log using "C:\My Documents\Berkeley\ARE\Diss\Ideas\DST\AustraliaData\Analysis\logCEC.smcl",
replace

set mem 800m
set more off
set matsize 2000
set maxvar 7000

clear
use TempLong.dta


sort Province Year Month Day DST_Hour
```

```
drop TreatPer1999I TreatPer1999II
gen TreatPer1999I=0
gen TreatPer1999II=0
replace TreatPer1999I=1 if (Province=="VIC1" | Province=="NSW1") & Year==1999 & Month ==8 & Day
==30 & DST_Hour >=4
replace TreatPer1999I=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(31Aug1999) &
Date<=d(11Sep1999)
replace TreatPer1999I=1 if Province=="VIC1" & Date>=d(31Aug1999) & Date<=d(18Sep1999)
replace TreatPer1999II=1 if (Province=="VIC1" | Province=="NSW1") & Date>=d(3Oct1999) &
Date<=d(30Oct1999)
replace TreatPer1999II=1 if (Province=="VIC1" | Province=="NSW1") & Year==1999 & Month ==10 & Day
<=31 & DST_Hour <=3


gen DSimStandarTime = DSim[_n-2]
gen DSimwRStandarTime = DSimwR[_n-2]
gen DemandLeft = Demand[_n+2]
sort Province DST_Hour Year TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I
TreatPer1999II
collapse(mean) Demand DSimStandarTime DSimwRStandarTime DemandLeft DSim DSimwR, by(Province
DST_Hour Year TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I TreatPer1999II )

gen Time = DST_Hour/2



twoway line Demand Time if TreatPer1999I==1 & Province=="VIC1" & Time >=6 & Time <= 23,
clcolor(black) ||/*
        */ line DSimwR Time if TreatPer1999I==1 & Province=="VIC1" & Time >=6 & Time <= 23,
clcolor(black) clpat(dash) || /*
        */ line Demand Time if TreatPer2001I==1 & Province=="VIC1" & Time >=6 & Time <= 23,
clcolor(blue) ||/*
      */ line DSimwR Time if TreatPer2001I==1 & Province=="VIC1" & Time >=6 & Time <= 23,
clcolor(blue)  clpat(dash)|| /*
        */ line Demand Time if TreatPerI==1 & Province=="VIC1" & Time >=6 & Time <= 23,
clcolor(red)   lwidth(thick)  /*
        */ legend(on order(1 "1999 Actual" 2 "1999 Simulated" 3 "2001 Actual" 4 "2001 Simulated"
5 "2000 Actual" )) /*
      */ ,  ytitle(Consumption in Megawatts) xtitle(Hour (Clock Time)) xlabel(6 7 to 23)
scheme(s1manual) graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export FIG5clockTreatIRefurb.tif, replace



twoway line Demand Time if TreatPer1999I==1 & Province=="VIC1"   & Time >=3 & Time <= 23,
clcolor(black) ||/*
        */ line DSimwR Time if TreatPer1999I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) clpat(dash)/*
        */ legend(on order(1 "1999 Actual" 2 "1999 Simulated" )) /*
      */ ,  ytitle(Consumption in Megawatts) xtitle(Hour (Clock Time)) xlabel(3 4 to 23)
scheme(s1manual) graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export clockTreatI99Refurb3_23.tif, replace


twoway line Demand Time if TreatPer2001I==1 & Province=="VIC1"   & Time >=3 & Time <= 23,
clcolor(blue) ||/*
        */ line DSimwR Time if TreatPer2001I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue) clpat(dash)/*
        */ legend(on order(1 "2001 Actual" 2 "2001 Simulated" )) /*
      */ ,  ytitle(Consumption in Megawatts) xtitle(Hour (Clock Time)) xlabel(3 4 to 23)
scheme(s1manual) graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export clockTreatI01Refurb3_23.tif, replace
```

```
twoway   line DemandLeft Time if TreatPerI==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(red)   lwidth(thick)|| /*
       */ line DSimwR Time if TreatPer1999I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(green) clpat(dash) || /*
       */ line Demand Time if TreatPer1999I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(green) ||/*
      */ line DSimwR Time if TreatPer2001I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue)  clpat(dash)|| /*
       */ line Demand Time if TreatPer2001I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue) /*
       */ title("Vicoria Treatment Period I, 1999 to 2001") /*
       */ legend(on order(1 "2000 Demand" 2 "1999 Simulated Demand" 3 "1999 Demand" 4 "2001
Simulated Demand" 5 "2001 Demand" )) /*
      */ ,  ytitle(Demand in MW) xtitle(Hour) xlabel(3 4 to 23) scheme(s1manual)
graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export FIG5.tif, replace

twoway   line Demand Time if TreatPerI==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(red)   lwidth(thick)|| /*
       */ line DSimwR Time if TreatPer1999I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) clpat(dash) || /*
       */ line Demand Time if TreatPer1999I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) ||/*
      */ line DSimwR Time if TreatPer2001I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue)  clpat(dash)|| /*
       */ line Demand Time if TreatPer2001I==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue) /*
       */ title("Vicoria Treatment Period I, 1999 to 2001") /*
       */ legend(on order(1 "2000 Demand" 2 "1999 Simulated Demand" 3 "1999 Demand" 4 "2001
Simulated Demand" 5 "2001 Demand" )) /*
      */ ,  ytitle(Demand in MW) xtitle(Hour) xlabel(3 4 to 23) scheme(s1manual)
graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export FIG5clockTreatIb.tif, replace

twoway   line Demand Time if TreatPerII==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(red)   lwidth(thick)|| /*
       */ line DSimwR Time if TreatPer1999II==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) clpat(dash) || /*
       */ line Demand Time if TreatPer1999II==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) ||/*
      */ line DSimwR Time if TreatPer2001II==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue)  clpat(dash)|| /*
       */ line Demand Time if TreatPer2001II==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue) /*
       */ title("Vicoria Treatment Period II, 1999 to 2001") /*
       */ legend(on order(1 "2000 Demand" 2 "1999 Simulated Demand" 3 "1999 Demand" 4 "2001
Simulated Demand" 5 "2001 Demand" )) /*
      */ ,  ytitle(Demand in MW) xtitle(Hour) xlabel(3 4 to 23) scheme(s1manual)
graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export FIG5clockTreatII.tif, replace


twoway   line DemandLeft Time if TreatPerI==1 & Province=="VIC1" , clcolor(red)   lwidth(thick)||
/*
       */ line DSimwR Time if TreatPer1999I==1 & Province=="VIC1" , clcolor(green) clpat(dash)
|| /*
       */ line Demand Time if TreatPer1999I==1 & Province=="VIC1" , clcolor(green) ||/*
      */ line DSimwR Time if TreatPer2001I==1 & Province=="VIC1" , clcolor(blue)  clpat(dash)||
/*
       */ line Demand Time if TreatPer2001I==1 & Province=="VIC1" , clcolor(blue) /*
       */ title("Vicoria Treatment Period I, 1999 to 2001") /*
       */ legend(on order(1 "2000 Demand" 2 "1999 Simulated Demand" 3 "1999 Demand" 4 "2001
Simulated Demand" 5 "2001 Demand" )) /*
      */ ,  ytitle(Demand in MW) xtitle(Hour) scheme(s1manual) graphregion(fcolor(none)
ifcolor(none)) plotregion(fcolor(none) lcolor(none))
graph export FIG5_AllHours.tif, replace
```

```
twoway   line Demand Time if TreatPerI==1 & Province=="VIC1" , clcolor(red)   lwidth(thick)|| /*
        */ line DSimwR Time if TreatPer1999I==1 & Province=="VIC1" , clcolor(green) clpat(dash)
|| /*
        */ line Demand Time if TreatPer1999I==1 & Province=="VIC1" , clcolor(green) ||/*
      */ line DSimwR Time if TreatPer2001I==1 & Province=="VIC1" , clcolor(blue)  clpat(dash)||
/*
        */ line Demand Time if TreatPer2001I==1 & Province=="VIC1" , clcolor(blue) /*
        */ title("Vicoria Treatment Period I, 1999 to 2001") /*
        */ legend(on order(1 "2000 Demand" 2 "1999 Simulated Demand" 3 "1999 Demand" 4 "2001
Simulated Demand" 5 "2001 Demand" )) /*
      */ ,  ytitle(Demand in MW) xtitle(Hour) scheme(s1manual) graphregion(fcolor(none)
ifcolor(none)) plotregion(fcolor(none) lcolor(none))
graph export FIG5_AllHoursClock.tif, replace


twoway line Demand DST_Hour if TreatPerI==1 & Province=="VIC1", clcolor(blue)  clpat(dash)|| /*
        */ line DSimwR DST_Hour if TreatPer2001I==1 & Province=="VIC1", clcolor(red) || /*
        */ line DSimwR DST_Hour if TreatPer1999I==1 & Province=="VIC1", clcolor(green) /*
        */ title("DST plot--Vicoria Treatment Period I in 1999, 2000 and 2001") /*
        */ legend(on order(1 "2000 Demand" 2 "2001 DSimWR" 3 "1999 DSimWR"))
graph export VICAverDemandswR00_01_99.tif, replace


twoway line Demand DST_Hour if TreatPerI==1 & Province=="VIC1", clcolor(blue)  clpat(dash)|| /*
        */ line DSim DST_Hour if TreatPerI==1 & Province=="VIC1", clcolor(red)  /*
        */ title("DST plot--Vicoria Treatment Period I") /*
        */ legend(on order(1 "2000 Demand" 2 "2000 DSim" ))
graph export VICAverDemand00_Treat1.tif, replace

gen Hour = DST_Hour - 2
twoway line Demand Hour if TreatPerI==1 & Province=="VIC1", clcolor(blue)  clpat(dash)|| /*
        */ line DSimStandarTime Hour if TreatPerI==1 & Province=="VIC1", clcolor(red)  /*
        */ title("DST plot--Vicoria Treatment Period I, in Standard Time") /*
        */ legend(on order(1 "2000 Demand" 2 "2000 DSim" ))
graph export VICAverDemand00_Treat1_StandardTime.tif, replace

clear
use TempLong.dta
sort Province Year Month Day DST_Hour
gen DSimStandarTime = DSim[_n-2]
gen DSimwRStandarTime = DSimwR[_n-2]
gen DemandLeft = Demand[_n+2]
sort Province DST_Hour Year Date TreatPerI TreatPerII TreatPer2001I TreatPer2001II TreatPer1999I
TreatPer1999II
gen TreatIandII = 0
replace TreatIandII = 1 if TreatPerI ==1 | TreatPerII ==1
gen Treat1999IandII = 0
replace Treat1999IandII = 1 if TreatPer1999I ==1 | TreatPer1999II ==1
gen Treat2001IandII = 0
replace Treat2001IandII = 1 if TreatPer2001I ==1 | TreatPer2001II ==1
collapse(mean) Demand DSimStandarTime DSimwRStandarTime DemandLeft DSim DSimwR, by(Province
DST_Hour Year TreatIandII Treat1999IandII Treat2001IandII )

gen Time = DST_Hour/2


twoway   line Demand Time if TreatIandII==1     & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(red)   lwidth(thick)|| /*
        */ line DSimwR Time if Treat1999IandII==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) clpat(dash) || /*
        */ line Demand Time if Treat1999IandII==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(black) ||/*
      */ line DSimwR Time if Treat2001IandII==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue)  clpat(dash)|| /*
        */ line Demand Time if Treat2001IandII==1 & Province=="VIC1" & Time >=3 & Time <= 23,
clcolor(blue) /*
        */ title("Vicoria Treatment Period I&II, 1999 to 2001") /*
        */ legend(on order(1 "2000 Demand" 2 "1999 Simulated Demand" 3 "1999 Demand" 4 "2001
Simulated Demand" 5 "2001 Demand" )) /*
```

```
      */ ,  ytitle(Demand in MW) xtitle(Hour) xlabel(3 4 to 23) scheme(s1manual)
graphregion(fcolor(none) ifcolor(none)) plotregion(fcolor(none) lcolor(none))

graph export FIG5clock_treatIandII.tif, replace
```

References

Australian Bureau of Statistics (2001a): Tourism Indicators, Report 8634.0, December Quarter 2000, Canberra.

Australian Bureau of Statistics (2001b): Tourist accommodation: an analysis over the Olympic period. Tourism Indicators, December Quarter 2000.

Table B: Regression discontinuity estimates of the effect of switching to DST, by half-hour: late-October switches in VIC and SA in 1999 and 2001-2005

Half-hour	11:00-	11:30-	12:00-	12:30-	13:00-	13:30-	14:00-	14:30-	15:00-
	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30
Percent change in demand	1.41	0.44	0.66	0.19	-0.24	0.40	1.35	1.33	1.31
Standard error	(1.74)	(1.43)	(1.56)	(1.54)	(1.41)	(1.64)	(1.90)	(1.37)	(1.56)

Standard errors are clustered on year

Table C: Estimated treatment effects by half-hour

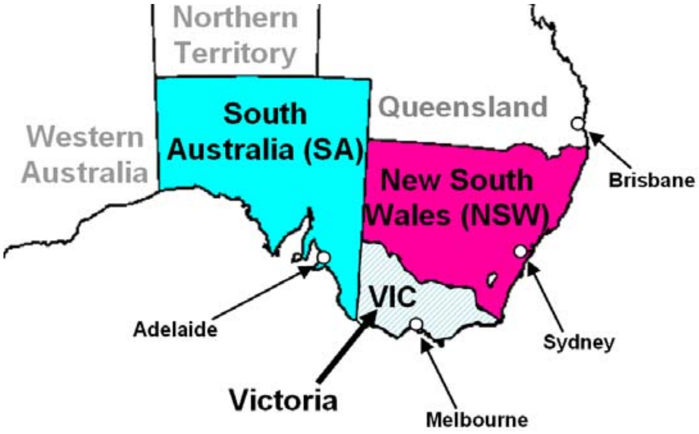
Half-hour		Standard	t-		Half-hour		Standard	t-	
beginning at	β_h	error	statistic	exp(β_h)-1	beginning at	β_h	error	statistic	exp(β_h)-1
00:00	-0.015	0.007	-2.30	-0.015	12:00	-	-	-	-
00:30	0.016	0.007	2.45	0.016	12:30	-	-	-	-
01:00	-0.052	0.006	-8.28	-0.051	13:00	-	-	-	-
01:30	-0.046	0.006	-7.17	-0.045	13:30	-	-	-	-
02:00	0.054	0.006	8.96	0.055	14:00	-	-	-	-
02:30	0.074	0.006	12.68	0.077	14:30	0.014	0.003	4.89	0.014
03:00	0.071	0.006	12.23	0.074	15:00	0.010	0.004	2.92	0.010
03:30	0.066	0.006	10.82	0.069	15:30	0.008	0.004	2.19	0.008
04:00	0.056	0.006	9.29	0.058	16:00	0.008	0.004	1.85	0.008
04:30	0.045	0.006	7.63	0.046	16:30	0.002	0.005	0.38	0.002
05:00	0.032	0.006	5.38	0.032	17:00	-0.015	0.006	-2.51	-0.015
05:30	0.024	0.005	4.42	0.024	17:30	-0.028	0.007	-3.99	-0.027
06:00	0.019	0.006	3.30	0.019	18:00	-0.050	0.007	-7.18	-0.049
06:30	0.016	0.005	3.01	0.016	18:30	-0.068	0.007	-9.50	-0.066
07:00	0.080	0.006	13.94	0.084	19:00	-0.058	0.008	-7.51	-0.056
07:30	0.083	0.006	14.20	0.087	19:30	-0.028	0.007	-3.85	-0.028
08:00	0.028	0.006	5.04	0.029	20:00	-0.011	0.007	-1.54	-0.011
08:30	0.013	0.005	2.63	0.013	20:30	-0.007	0.007	-1.06	-0.007
09:00	0.009	0.004	2.14	0.009	21:00	-0.002	0.007	-0.30	-0.002
09:30	0.006	0.004	1.65	0.006	21:30	0.002	0.007	0.25	0.002
10:00	0.003	0.003	0.87	0.003	22:00	-0.009	0.006	-1.45	-0.009
10:30	0.006	0.003	1.66	0.006	22:30	-0.030	0.006	-5.30	-0.029
11:00	0.004	0.003	1.25	0.004	23:00	-0.128	0.006	-20.60	-0.120
11:30	0.002	0.002	1.09	0.002	23:30	-0.131	0.007	-19.07	-0.123

Standard errors are clustered on date

No effects are estimated for the control period of 12:00-14:30

The large effects shown during the overnight hours are driven by centralized off-peak water heating that is activated by automatic timers, set to standard time

Figure 1: Southeastern Australia states and major cities



NSW, VIC, and SA in mainland Australia regularly begin DST on the last Sunday in October each year. In 2000, however, NSW and VIC began DST on 27 August, whereas SA did not begin DST until 29 October.

Figure D1: Quarterly Room Nights Occupied in VIC (left panel) and SA (right panel)

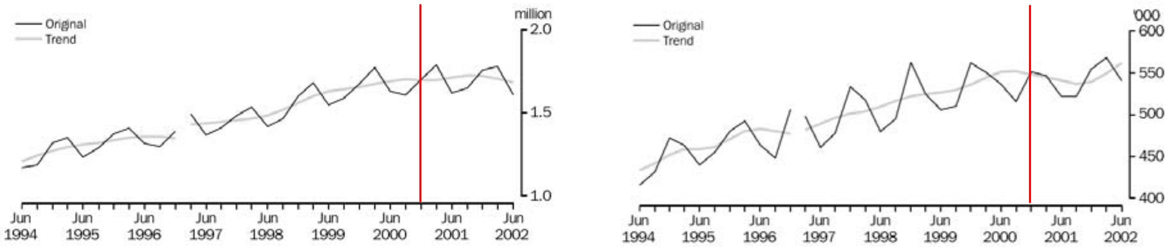
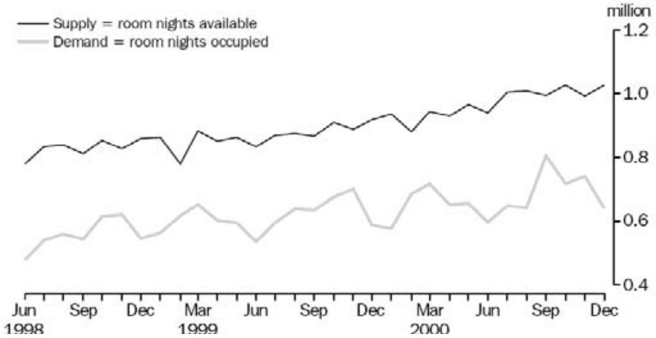


Figure: D2: Supply and Demand for Tourist Accommodations in Sydney



Source: Australian Bureau of Statistics, 2001. The vertical line indicates the 4th quarter in 2000 (December quarter). The treatment period "September" falls within the 3rd quarter 2000 and the treatment period "October" in the 4th quarter.