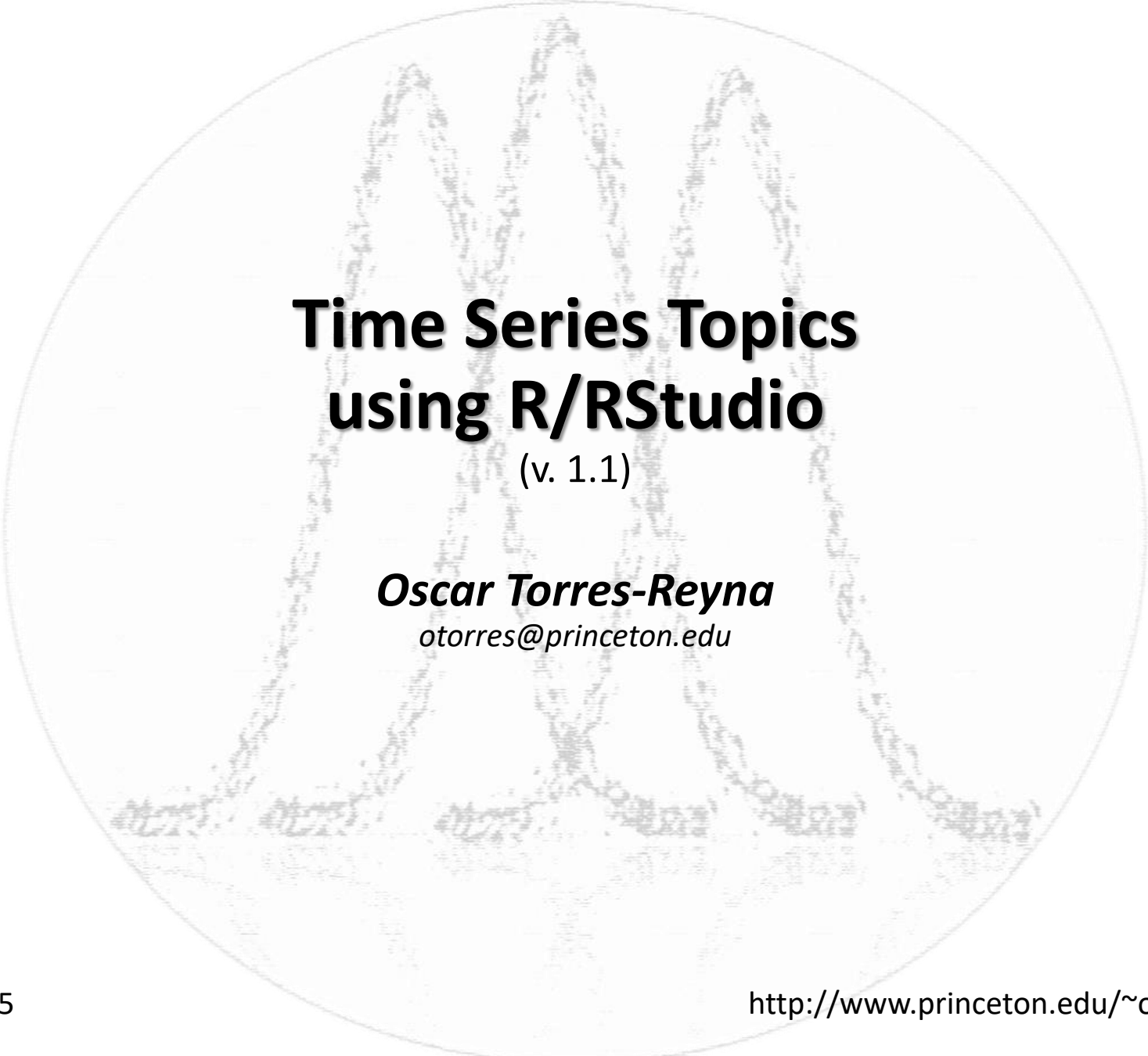




Time Series Topics using R/RStudio

(v. 1.1)

Oscar Torres-Reyna
otorres@princeton.edu

July 2015

<http://www.princeton.edu/~otorres/>

To convert date in string or integers to date variables use the function `as.Date()`

	date1	date2	date3	date4
1	1-Jan-90	1/1/1990	19900101	199011
2	2-Jan-90	1/2/1990	19900102	199012
3	3-Jan-90	1/3/1990	19900103	199013
4	4-Jan-90	1/4/1990	19900104	199014
5	5-Jan-90	1/5/1990	19900105	199015
6	6-Jan-90	1/6/1990	19900106	199016
7	7-Jan-90	1/7/1990	19900107	199017
8	8-Jan-90	1/8/1990	19900108	199018
9	9-Jan-90	1/9/1990	19900109	199019
10	10-Jan-90	1/10/1990	19900110	1990110
11	11-Jan-90	1/11/1990	19900111	1990111
12	12-Jan-90	1/12/1990	19900112	1990112
13	13-Jan-90	1/13/1990	19900113	1990113

↑ ↑ ↑ ↑
 Character Character Integer Integer

Symbol	Meaning	Example
%d	day as a number (0-31)	01-31
%a	abbreviated weekday	Mon
%A	unabbreviated weekday	Monday
%m	month (00-12)	00-12
%b	abbreviated month	Jan
%B	unabbreviated month	January
%y	2-digit year	07
%Y	4-digit year	2007

Source of table: <http://www.statmethods.net/input/dates.html>

Converting the string 'date1' to a date variable called 'new.date1'

```
mydata$new.date1 = as.Date(mydata$date1, "%d-%b-%y")
```

Converting the string 'date2' to a date variable called 'new.date2'

```
mydata$new.date2 = as.Date(mydata$date2, "%m/%d/%Y")
```

Converting the integer 'date3' to a date variable called 'new.date3'

```
mydata$new.date3 = as.Date(as.character(mydata$date3), "%Y%m%d")
```

Converting the integer 'date4' to a date variable called 'new.date4'

```
mydata$len.date4 = nchar(mydata$date4) # Need to identify the length first
```

```
mydata$date4 = as.character(mydata$date4) # Need to convert to character
```

```
mydata$date4b = ifelse(mydata$len.date4==6, paste0(substr(mydata$date4,1,4),0,
                                                    substr(mydata$date4,5,5),0,
                                                    substr(mydata$date4,6,6)),
                        paste0(substr(mydata$date4,1,4),0,
                                substr(mydata$date4,5,5),
                                substr(mydata$date4,6,7)))
```

```
mydata$new.date4 = as.Date(mydata$date4b, "%Y%m%d")
```

Need to specify the format of the original variable

See the converted variables on the following slide

See the previous slide for the conversion process

mydata	date1	date2	date3	date4	new.date1	new.date2	new.date3	len.date4	date4b	new.date4
1	1-Jan-90	1/1/1990	19900101	199011	1990-01-01	1990-01-01	1990-01-01	6	19900101	1990-01-01
2	2-Jan-90	1/2/1990	19900102	199012	1990-01-02	1990-01-02	1990-01-02	6	19900102	1990-01-02
3	3-Jan-90	1/3/1990	19900103	199013	1990-01-03	1990-01-03	1990-01-03	6	19900103	1990-01-03
4	4-Jan-90	1/4/1990	19900104	199014	1990-01-04	1990-01-04	1990-01-04	6	19900104	1990-01-04
5	5-Jan-90	1/5/1990	19900105	199015	1990-01-05	1990-01-05	1990-01-05	6	19900105	1990-01-05
6	6-Jan-90	1/6/1990	19900106	199016	1990-01-06	1990-01-06	1990-01-06	6	19900106	1990-01-06
7	7-Jan-90	1/7/1990	19900107	199017	1990-01-07	1990-01-07	1990-01-07	6	19900107	1990-01-07
8	8-Jan-90	1/8/1990	19900108	199018	1990-01-08	1990-01-08	1990-01-08	6	19900108	1990-01-08
9	9-Jan-90	1/9/1990	19900109	199019	1990-01-09	1990-01-09	1990-01-09	6	19900109	1990-01-09
10	10-Jan-90	1/10/1990	19900110	1990110	1990-01-10	1990-01-10	1990-01-10	7	19900110	1990-01-10
11	11-Jan-90	1/11/1990	19900111	1990111	1990-01-11	1990-01-11	1990-01-11	7	19900111	1990-01-11
12	12-Jan-90	1/12/1990	19900112	1990112	1990-01-12	1990-01-12	1990-01-12	7	19900112	1990-01-12
13	13-Jan-90	1/13/1990	19900113	1990113	1990-01-13	1990-01-13	1990-01-13	7	19900113	1990-01-13

↑

Character

↑

Character

↑

Integer

↑

Integer

To convert back to string characters use the function `as.character()`

```
mydata$string.date1 = as.character(mydata$new.date1)
```

From string/numeric/factor to date variable

Converting quarterly 'yearq#' to date variable

```
library(zoo)
mydata$quarter = as.yearqtr(mydata$quarterly, format="%Yq%q")
```

Formatting quarter data as daily

```
mydata$qvar = as.Date(mydata$quarter)
```

	quarterly	quarter	qvar
1	1957q1	1957 Q1	1957-01-01
2	1957q2	1957 Q2	1957-04-01
3	1957q3	1957 Q3	1957-07-01
4	1957q4	1957 Q4	1957-10-01
5	1958q1	1958 Q1	1958-01-01
6	1958q2	1958 Q2	1958-04-01
7	1958q3	1958 Q3	1958-07-01
8	1958q4	1958 Q4	1958-10-01
9	1959q1	1959 Q1	1959-01-01
10	1959q2	1959 Q2	1959-04-01
11	1959q3	1959 Q3	1959-07-01
12	1959q4	1959 Q4	1959-10-01
13	1960q1	1960 Q1	1960-01-01

Extracting year, month and day using base functions

```
# Extracting year (from a variable in date format)
```

```
mydata$year = as.numeric(format(mydata$date, "%Y"))
```

```
# Extracting month (from a variable in date format)
```

```
mydata$month = as.numeric(format(mydata$date, "%m"))
```

```
# Extracting day (from a variable in date format)
```

```
mydata$day = as.numeric(format(mydata$date, "%d"))
```

```
mydata
      date year month day
1 2005-01-22 2005     1  22
2 1990-11-18 1990    11  18
3 1990-01-26 1990     1  26
4 1999-10-23 1999    10  23
5 1996-04-28 1996     4  28
6 1993-02-02 1993     2   2
7 2015-06-29 2015     6  29
8 1990-12-30 1990    12  30
9 2010-05-27 2010     5  27
10 2002-01-16 2002     1  16
11 2007-11-13 2007    11  13
12 1998-10-23 1998    10  23
13 2013-07-03 2013     7   3
```

Converting to monthly, quarterly, weekly dates

Extracting monthly (using -zoo-), see variable 'monthly' below.

```
library(zoo)
mydata$monthly = as.yearmon(mydata$date)
```

Extracting quarterly (using -zoo-), see variable 'quarter' below.

```
mydata$quarter = as.yearqtr(mydata$date)
```

Getting days of week (using -lubridate-), see variable 'weekday' below.

```
library(lubridate)
mydata$weekday = wday(mydata$date, label = TRUE)
```

Getting the week of date (using -lubridate-), see variable 'weekly' below.

```
mydata$weekly = week(mydata$date)
```

```
mydata
      date  monthly quarter weekday weekly
1 2005-01-22   Jan 2005 2005 Q1      Sat      4
2 1990-11-18   Nov 1990 1990 Q4      Sun     46
3 1990-01-26   Jan 1990 1990 Q1      Fri      4
4 1999-10-23   Oct 1999 1999 Q4      Sat     43
5 1996-04-28   Apr 1996 1996 Q2      Sun     17
6 1993-02-02   Feb 1993 1993 Q1     Tues      5
7 2015-06-29   Jun 2015 2015 Q2      Mon     26
8 1990-12-30   Dec 1990 1990 Q4      Sun     52
9 2010-05-27   May 2010 2010 Q2    Thurs     21
10 2002-01-16   Jan 2002 2002 Q1      Wed      3
11 2007-11-13   Nov 2007 2007 Q4     Tues     46
12 1998-10-23   Oct 1998 1998 Q4      Fri     43
13 2013-07-03   Jul 2013 2013 Q3      Wed     27
```

Lags and forwards (or leads)

Getting the sample data

```
usa = read.csv("http://dss.princeton.edu/training/us.csv", header=TRUE)
```

Lag 1 of 'gdppcgr', see variable 'l1.gdp' below.

```
usa$l1.gdp <- c(NA,usa$gdppcgr[1:nrow(usa)-1])
```

Forward 1 of 'gdppcgr', see variable 'f1.gdp' below.

```
usa$f1.gdp <- c(usa$gdppcgr[2:nrow(usa)],NA)
```

	year	gdppcgr	tradegr	px	unemp	l1.gdp	f1.gdp
1	1990	0.8	-1.7	59.9	5.5	NA	-1.4
2	1991	-1.4	3.0	62.5	6.3	0.8	2.1
3	1992	2.1	7.0	64.3	11.1	-1.4	1.4
4	1993	1.4	6.0	66.2	11.5	2.1	2.8
5	1994	2.8	10.5	68.0	12.2	1.4	1.5
6	1995	1.5	9.1	69.9	9.7	2.8	2.6
7	1996	2.6	8.5	71.9	9.5	1.5	3.2
8	1997	3.2	12.7	73.6	8.7	2.6	3.2
9	1998	3.2	7.3	74.8	8.0	3.2	3.6
10	1999	3.6	8.4	76.4	6.8	3.2	2.9
11	2000	2.9	11.1	79.0	6.0	3.6	0.0
12	2001	0.0	-4.1	81.2	6.1	2.9	0.8
13	2002	0.8	1.4	82.5	8.5	0.0	1.9
14	2003	1.9	3.4	84.4	11.8	0.8	2.8
15	2004	2.8	10.8	86.6	12.7	1.9	2.4
16	2005	2.4	6.3	89.6	11.8	2.8	1.7
17	2006	1.7	7.4	92.4	10.0	2.4	0.8
18	2007	0.8	5.2	95.1	10.0	1.7	-1.2
19	2008	-1.2	0.9	98.7	10.6	0.8	-3.7
20	2009	-3.7	-11.6	98.4	16.3	-1.2	1.7
21	2010	1.7	12.3	100.0	29.0	-3.7	0.9
22	2011	0.9	6.1	103.2	31.3	1.7	1.6
23	2012	1.6	2.7	105.3	29.3	0.9	1.5
24	2013	1.5	2.0	106.8	12.3	1.6	NA

Lag variables in panel data using base R

Creating a dataset

```
set.seed(12345)
mydata = data.frame(country = rep(toupper(letters[1:3]), each=5),
                     year = rep(2000:2004,3),
                     var1 = rnorm(15))
```

Function to get the lags

```
lag = function(x) c(NA,x[1:(length(x)-1)])
```

Getting the lags in the data

```
mydata$lag.var1 = ave(mydata$var1, mydata$country, FUN=lag)
mydata
```

	country	year	var1	lag.var1
1	A	2000	0.5855288	NA
2	A	2001	0.7094660	0.5855288
3	A	2002	-0.1093033	0.7094660
4	A	2003	-0.4534972	-0.1093033
5	A	2004	0.6058875	-0.4534972
6	B	2000	-1.8179560	NA
7	B	2001	0.6300986	-1.8179560
8	B	2002	-0.2761841	0.6300986
9	B	2003	-0.2841597	-0.2761841
10	B	2004	-0.9193220	-0.2841597
11	C	2000	-0.1162478	NA
12	C	2001	1.8173120	-0.1162478
13	C	2002	0.3706279	1.8173120
14	C	2003	0.5202165	0.3706279
15	C	2004	-0.7505320	0.5202165

Forward or lead variables in panel data using base R

Creating a dataset

```
set.seed(12345)
```

```
mydata = data.frame(country = rep(toupper(letters[1:3]), each=5),  
                    year = rep(2000:2004,3),  
                    var1 = rnorm(15))
```

Function to get the forward or lead values

```
lead = function(x) c(x[2:length(x)],NA)
```

Getting the forward/leads in the data

```
mydata$lead.var1 = ave(mydata$var1, mydata$country, FUN=lead)  
mydata
```

	country	year	var1	lead.var1
1	A	2000	0.5855288	0.7094660
2	A	2001	0.7094660	-0.1093033
3	A	2002	-0.1093033	-0.4534972
4	A	2003	-0.4534972	0.6058875
5	A	2004	0.6058875	NA
6	B	2000	-1.8179560	0.6300986
7	B	2001	0.6300986	-0.2761841
8	B	2002	-0.2761841	-0.2841597
9	B	2003	-0.2841597	-0.9193220
10	B	2004	-0.9193220	NA
11	C	2000	-0.1162478	1.8173120
12	C	2001	1.8173120	0.3706279
13	C	2002	0.3706279	0.5202165
14	C	2003	0.5202165	-0.7505320
15	C	2004	-0.7505320	NA

Lag variables in panel data using `-plm-`

Creating a dataset

```
set.seed(12345)
mydata = data.frame(country = rep(toupper(letters[1:3]), each=5),
                    year = rep(2000:2004,3),
                    var1 = rnorm(15))
```

Getting the lags

```
library(plm)
mydata = pdata.frame(mydata, index = c("country", "year"))
mydata$ lag.var1 = lag(mydata$var1)
mydata
```

	country	year	var1	lag.var1
A-2000	A	2000	0.5855288	NA
A-2001	A	2001	0.7094660	0.5855288
A-2002	A	2002	-0.1093033	0.7094660
A-2003	A	2003	-0.4534972	-0.1093033
A-2004	A	2004	0.6058875	-0.4534972
B-2000	B	2000	-1.8179560	NA
B-2001	B	2001	0.6300986	-1.8179560
B-2002	B	2002	-0.2761841	0.6300986
B-2003	B	2003	-0.2841597	-0.2761841
B-2004	B	2004	-0.9193220	-0.2841597
C-2000	C	2000	-0.1162478	NA
C-2001	C	2001	1.8173120	-0.1162478
C-2002	C	2002	0.3706279	1.8173120
C-2003	C	2003	0.5202165	0.3706279
C-2004	C	2004	-0.7505320	0.5202165

Replacing missing values with previous non-missing

Creating a dataset

```
set.seed(12345)
mydata = data.frame(country = rep(toupper(letters[1:3]), each=5),
                     year = rep(2000:2004,3),
                     var1 = rnorm(15))
mydata$var1 = ifelse(mydata$year<2003,mydata$var1,NA)
```

Replacing missing values with previous non-missing

```
library(zoo)
mydata$var2 <- na.locf(mydata$var1)
mydata
```

	country	year	var1	var2
1	A	2000	0.5855288	0.5855288
2	A	2001	0.7094660	0.7094660
3	A	2002	-0.1093033	-0.1093033
4	A	2003	NA	-0.1093033
5	A	2004	NA	-0.1093033
6	B	2000	-1.8179560	-1.8179560
7	B	2001	0.6300986	0.6300986
8	B	2002	-0.2761841	-0.2761841
9	B	2003	NA	-0.2761841
10	B	2004	NA	-0.2761841
11	C	2000	-0.1162478	-0.1162478
12	C	2001	1.8173120	1.8173120
13	C	2002	0.3706279	0.3706279
14	C	2003	NA	0.3706279
15	C	2004	NA	0.3706279

Rolling sum

```
library(zoo)
```

```
# Getting the sample data
```

```
mydata = read.csv("http://www.princeton.edu/~otorres/quarterly.csv", header = TRUE)
```

```
# Getting the quarterly as date variable
```

```
mydata$quarter = as.yearqtr(mydata$quarterly, format="%Yq%q")
```

```
# Sorting the data by quarters
```

```
mydata = mydata[ order(mydata$quarterly), ]
```

```
# Getting the rolling sum every four quarters for variable 'unemp'
```

```
mydata$sum <- rollapply(mydata$unemp, 4, sum, na.rm=TRUE, fill = NA, align = "right")  
head(mydata, 13)
```

	date	unemp	cpi	interest	gdp	date1	quarterly	year	quarter	sum
1	1957:01	3.933333	27.77667	2.96	NA	1957:01	1957q1	1957	1957 Q1	NA
2	1957:02	4.100000	28.01333	3.00	7.4391565	1957:02	1957q2	1957	1957 Q2	NA
3	1957:03	4.233333	28.26333	3.47	2.9988995	1957:03	1957q3	1957	1957 Q3	NA
4	1957:04	4.933333	28.40000	2.98	-1.3890126	1957:04	1957q4	1957	1957 Q4	17.20000
5	1958:01	6.300000	28.73667	1.20	-0.9209932	1958:01	1958q1	1958	1958 Q1	19.56667
6	1958:02	7.366667	28.93000	0.93	3.2106061	1958:02	1958q2	1958	1958 Q2	22.83333
7	1958:03	7.333334	28.91333	1.76	3.6096158	1958:03	1958q3	1958	1958 Q3	25.93333
8	1958:04	6.366667	28.94333	2.42	2.9486711	1958:04	1958q4	1958	1958 Q4	27.36667
9	1959:01	5.833334	28.99333	2.80	1.1829405	1959:01	1959q1	1959	1959 Q1	26.90000
10	1959:02	5.100000	29.04333	3.39	5.2851181	1959:02	1959q2	1959	1959 Q2	24.63333
11	1959:03	5.266667	29.19333	3.76	6.3019662	1959:03	1959q3	1959	1959 Q3	22.56667
12	1959:04	5.600000	29.37000	3.99	1.0160819	1959:04	1959q4	1959	1959 Q4	21.80000
13	1960:01	5.133333	29.39667	3.84	9.7185040	1960:01	1960q1	1960	1960 Q1	21.10000

Rolling sum in panel data

Creating a dataset

```
set.seed(12345)
mydata = data.frame(country = rep(toupper(letters[1:3]), each=5),
                     year = rep(2000:2004,3),
                     var1 = rnorm(15))
```

Sort data by country and year

```
mydata = mydata[ order(mydata$country, mydata$year), ]
```

Rolling sum every four years

```
library(zoo)
rolsum = function(x) rollapply(x, 4, sum, na.rm=TRUE, fill = NA, align = "right")
mydata$sum = ave(mydata$var1, mydata$country, FUN=rolsum)
mydata
```

	country	year	var1	sum
1	A	2000	0.5855288	NA
2	A	2001	0.7094660	NA
3	A	2002	-0.1093033	NA
4	A	2003	-0.4534972	0.7321943
5	A	2004	0.6058875	0.7525530
6	B	2000	-1.8179560	NA
7	B	2001	0.6300986	NA
8	B	2002	-0.2761841	NA
9	B	2003	-0.2841597	-1.7482013
10	B	2004	-0.9193220	-0.8495673
11	C	2000	-0.1162478	NA
12	C	2001	1.8173120	NA
13	C	2002	0.3706279	NA
14	C	2003	0.5202165	2.5919086
15	C	2004	-0.7505320	1.9576244

Regression with lags

```
library(midasr)
library(dynlm)
library(stargazer)

reg1 = midas_u(gdppcgr ~ mls(tradegr,1,1), data = usa, na.action="na.exclude")
reg2 = dynlm(gdppcgr ~ mls(tradegr,1,1), data = usa, na.action="na.exclude")
stargazer(reg1, reg2, type="text")
```

=====		
	Dependent variable:	

	gdppcgr	
	OLS	dynamic
		linear
	(1)	(2)

mls(tradegr, 1, 1)	0.089 (0.063)	0.089 (0.063)
Constant	0.965* (0.483)	0.965* (0.483)

Observations	23	23
R2	0.088	0.088
Adjusted R2	0.044	0.044
Residual Std. Error (df = 21)	1.673	1.673
F Statistic (df = 1; 21)	2.022	2.022
=====		
Note:	*p<0.1; **p<0.05; ***p<0.01	

Predicting values after regression

```
usa$gdp.hat1 = c(NA,predict(reg1)) # Using reg1 running midas  
usa$gdp.hat2 = c(NA,predict(reg2)) # Using reg2 running dylm
```

```
usa  
  year gdppcgr tradegr    px unemp    gdp.hat1    gdp.hat2  
1 1990     0.8    -1.7  59.9   5.5         NA         NA  
2 1991    -1.4     3.0  62.5   6.3  0.81361487  0.81361487  
3 1992     2.1     7.0  64.3  11.1  1.23152731  1.23152731  
4 1993     1.4     6.0  66.2  11.5  1.58719747  1.58719747  
5 1994     2.8    10.5  68.0  12.2  1.49827993  1.49827993  
6 1995     1.5     9.1  69.9   9.7  1.89840886  1.89840886  
7 1996     2.6     8.5  71.9   9.5  1.77392430  1.77392430  
8 1997     3.2    12.7  73.6   8.7  1.72057378  1.72057378  
9 1998     3.2     7.3  74.8   8.0  2.09402745  2.09402745  
10 1999     3.6     8.4  76.4   6.8  1.61387273  1.61387273  
11 2000     2.9    11.1  79.0   6.0  1.71168202  1.71168202  
12 2001     0.0    -4.1  81.2   6.1  1.95175938  1.95175938  
13 2002     0.8     1.4  82.5   8.5  0.60021278  0.60021278  
14 2003     1.9     3.4  84.4  11.8  1.08925925  1.08925925  
15 2004     2.8    10.8  86.6  12.7  1.26709433  1.26709433  
16 2005     2.4     6.3  89.6  11.8  1.92508412  1.92508412  
17 2006     1.7     7.4  92.4  10.0  1.52495519  1.52495519  
18 2007     0.8     5.2  95.1  10.0  1.62276448  1.62276448  
19 2008    -1.2     0.9  98.7  10.6  1.42714590  1.42714590  
20 2009    -3.7   -11.6  98.4  16.3  1.04480048  1.04480048  
21 2010     1.7    12.3 100.0  29.0 -0.06666877 -0.06666877  
22 2011     0.9     6.1 103.2  31.3  2.05846043  2.05846043  
23 2012     1.6     2.7 105.3  29.3  1.50717168  1.50717168  
24 2013     1.5     2.0 106.8  12.3  1.20485205  1.20485205
```

References

A Little Book of R For Time Series, release 0.2, Avril Coghlan, <https://media.readthedocs.org/pdf/a-little-book-of-r-for-time-series/latest/a-little-book-of-r-for-time-series.pdf>

CrossValidated, <http://stats.stackexchange.com/>

Quick-R, <http://www.statmethods.net/>

R bloggers, <http://www.r-bloggers.com/>

Stackoverflow, <http://stackoverflow.com/>

World Development Indicators, <http://data.worldbank.org/data-catalog/world-development-indicators>

R packages:

- lubridate, <https://cran.r-project.org/web/packages/lubridate/lubridate.pdf>
- MIDAS, <https://cran.r-project.org/web/packages/midasr/midasr.pdf>
- plm, <https://cran.r-project.org/web/packages/plm/plm.pdf>
- stargazer, <https://cran.r-project.org/web/packages/stargazer/stargazer.pdf>,
<http://dss.princeton.edu/training/NiceOutputR.pdf>
- zoo, <https://cran.r-project.org/web/packages/zoo/zoo.pdf>