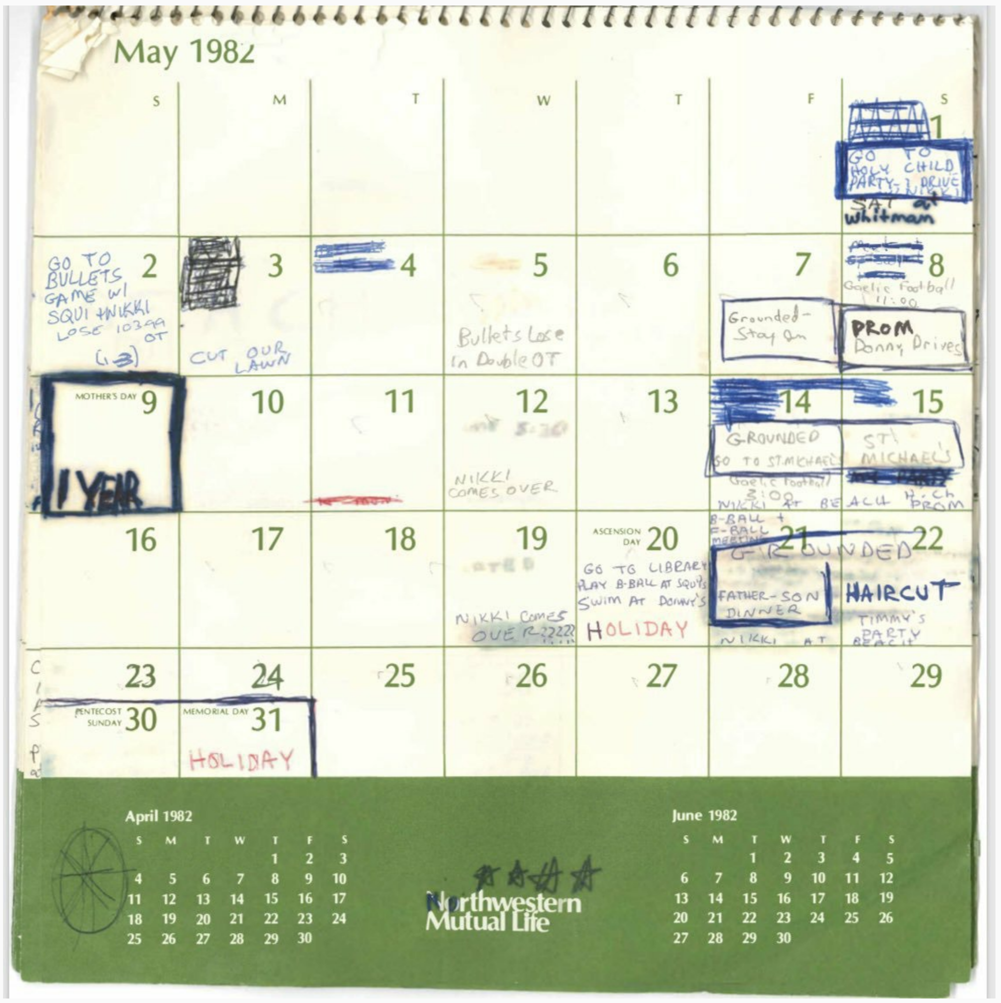
Data Analysis in R

Dealing with Dates and Times

Michael E DeWitt Jr

2018-09-20 (updated: 2018-10-18)

Dates and Times are Important!



May 1982

S	M	T	W	T	F	S
						1 GO TO HOLY CHILD PARTY - I DRIVE NIKKI SAT at Whitman
2 GO TO BULLETS GAME W/ SQUI +NIKKI LOSE 103-99 OT (1-3)	3 CUT OUR LAWN	4	5 Bullets Lose In Double OT	6	7 Grounded - Stay In	8 Gaelic Football 11:00 PROM Donny Drives
MOTHER'S DAY 9 1 YEAR	10	11	12 NIKKI COMES OVER	13	14 GROUNDED GO TO ST. MICHAEL'S Gaelic Football 3:00 NIKKI AT BE	15 ST. MICHAEL'S ACH H.Ch PROM
16	17	18	19 NIKKI COMES OVER??????	ASCENSION DAY 20 GO TO LIBRARY PLAY B-BALL AT SQUI'S SWIM AT DONNY'S HOLIDAY	21 B-BALL + F-BALL MEETING GROUNDED FATHER-SON DINNER NIKKI AT	22 HAIRCUT TIMMY'S PARTY BEACH
23	24	25	26	27	28	29
PENTECOST SUNDAY 30	MEMORIAL DAY 31 HOLIDAY					

April 1982

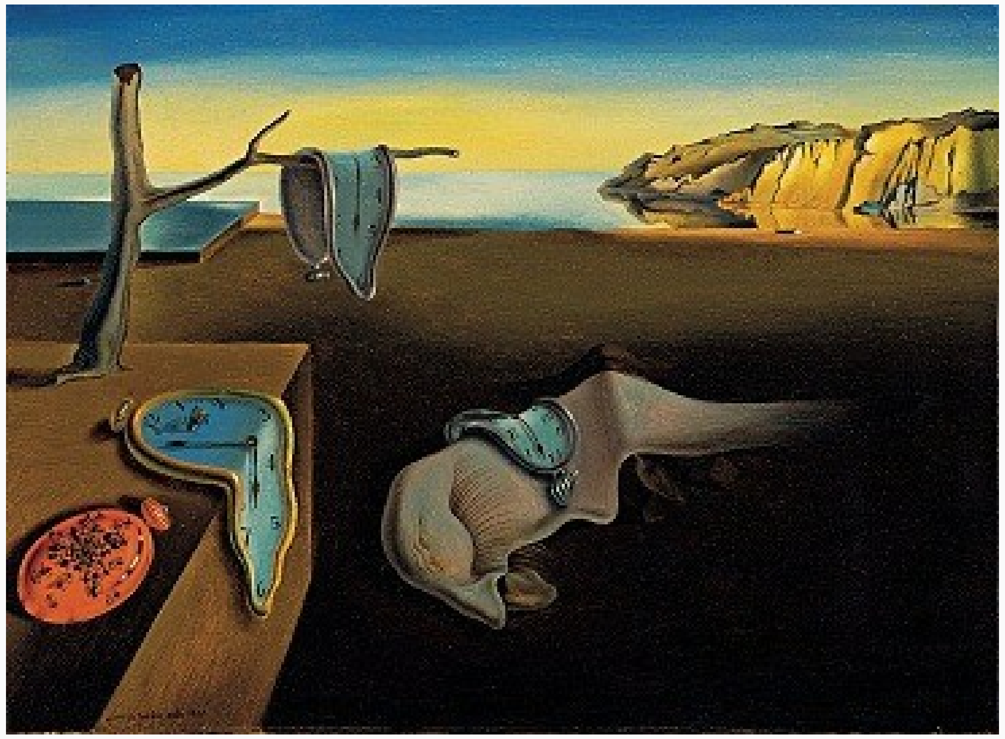
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Northwestern Mutual Life

June 1982

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

But Dates and Times are Challenging



Dates and Times are Special

- Months have different days (even particular months change their days!)
- Time has inconsistent units (hours, seconds)
- Time zones!
- They are ordered (so we can't treat them as characters)

And...we all don't use the same conventions!

And...we all don't use the same conventions!

1. YYYY-MM-DD

And...we all don't use the same conventions!

1. YYYY-MM-DD
2. MM-DD-YYYY

And...we all don't use the same conventions!

1. YYYY-MM-DD
2. MM-DD-YYYY
3. MM/DD/YYYY

And...we all don't use the same conventions!

1. YYYY-MM-DD
2. MM-DD-YYYY
3. MM/DD/YYYY
4. YYYYMMDD

And...we all don't use the same conventions!

1. YYYY-MM-DD
2. MM-DD-YYYY
3. MM/DD/YYYY
4. YYYYMMDD
5. DDMMYYYY

And...we all don't use the same conventions!

1. YYYY-MM-DD
2. MM-DD-YYYY
3. MM/DD/YYYY
4. YYYYMMDD
5. DDMMYYYY
6. MMMYY

And...we all don't use the same conventions!

1. YYYY-MM-DD
2. MM-DD-YYYY
3. MM/DD/YYYY
4. YYYYMMDD
5. DDMMYYYY
6. MMMYY
7. Somethings else?????

R Solution?

- Add a new `type` to deal with dates and times

R Solution?

- Add a new `type` to deal with dates and times

This new type will allow the R internals to handle date/ time conversions and calculations

R Solution?

- Add a new `type` to deal with dates and times

This new type will allow the R internals to handle date/ time conversions and calculations

You just need to know how to convert an object to a date to take advantage...

R Doesn't Recognize Dates Immediately Sometimes

```r
class("2017-10-17")
```

```
## [1] "character"
```

```r
class("10/17/2018")
```

```
## [1] "character"
```

Explicitly Defining Dates

R has some built in function to help define dates

Personally I find them difficult and unintuitive to use...

```
z_base <- strptime("20/2/06 11:16:16", "%d/%m/%y %H:%M")
z_base
```

```
## [1] "2006-02-20 11:16:00 EST"
```

Enter `lubridate`

`lubridate` is another `tidyverse` package designed to detail with dates

Enter `lubridate`

`lubridate` is another `tidyverse` package designed to detail with dates

It has functions that make dealing with dates more intuitive

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```r
z_lub <- dmy_hms("20/2/06 11:16:16", tz = "America/New_York")
z_lub
```

```
## [1] "2006-02-20 11:16:16 EST"
```

See the cheatsheet

`lubridate` provides several functions for dates

The function you use depends on the format in which you find your data

`ymd` for **Year** ("/"|"-") **Month** ("/"|"-") **Day**

`dmy` for **Day** ("/"|"-") **Month** ("/"|"-") **Year**

`mdy` for **Month** ("/"|"-") **Day** ("/"|"-") **Year**

Additionally We Can Specify The Timezone

```r
mdy("10/16/2018", tz = "Pacific/Auckland")
```

```
## [1] "2018-10-16 NZDT"
```

Sometimes we are interested in a part of a date

`day` returns the day number

`week` returns the week number

`wday` returns the day of the week

`month` returns the month number

`year` returns the year

Times

Times are also tricky for many of the same reasons

But `lubridate` has our solution

And `lubridate` functions are compatible with `tidyverse` workflows

Datetimes vs times

R via `lubridate` has two representations of times

`Datetimes` which have a date *and* time component

`Times` which have a time component only

The context will guide you as to how to manipulate the data

Our Functions

Date Times

For each `ymd` combination we have an associated function to parse:

- hours only `ymd_h`
- hours and minutes `ymd_hm`
- hours, minutes, seconds `ymd_hms`
- etc

Our Functions

Date Times

For each `ymd` combination we have an associated function to parse:

- hours only `ymd_h`
- hours and minutes `ymd_hm`
- hours, minutes, seconds `ymd_hms`
- etc

Times

Same as above without the date function

Calculating Differences

To calculate differences between dates and time it is best to establish an "interval"

```r
interval(ymd("20161016"), dmy("1/1/2017"))
```

```
## [1] 2016-10-16 UTC--2017-01-01 UTC
```

Calculating Differences

To calculate differences between dates and time it is best to establish an "interval"

```
interval(ymd("20161016"), dmy("1/1/2017"))
```

```
## [1] 2016-10-16 UTC--2017-01-01 UTC
```

We can then do calculations on it for example days in this interval using `d*` functions

```
interval(ymd("20161016"), dmy("1/1/2017"))/ddays(1)
```

```
## [1] 77
```

Calculating Differences

To calculate differences between dates and time it is best to establish an "interval"

```
interval(ymd("20161016"), dmy("1/1/2017"))
```

```
## [1] 2016-10-16 UTC--2017-01-01 UTC
```

We can then do calculations on it for example days in this interval using `d*` functions

```
interval(ymd("20161016"), dmy("1/1/2017"))/ddays(1)
```

```
## [1] 77
```

Or even seconds

```
interval(ymd("20161016"), dmy("1/1/2017"))/dseconds(1)
```

```
## [1] 6652800
```

Datetimes Allow Times Series Analysis

USing the `forecast` package we can implement advanced time series models (ARIMA, etc)

Using `CausalImpact` we can estimate Causality using Bayesian Structural Times Series

Using `prophet` we can forecast at scale using Bayesian estimation

And, And...

Recap

Date and Times are tricky. Period.

R has excellent faculties for dealing with dates and times

`lubridate` can help us convert our dates and times to R dates and times

We have some functions from `lubridate` to help us

Convert

`ymd` for **Year** ("/"|"-") **Month** ("/"|"-") **Day**

`dmy` for **Day** ("/"|"-") **Month** ("/"|"-") **Year**

`mdy` for **Month** ("/"|"-") **Day** ("/"|"-") **Year**

Manipulate

`day` / `week` / `wday` / `month` / `year` / `hour` / `second` to parse the desired component

`interval` to create and interval between two dates

`d*` functions to calculate the difference (in units desired)

And others depending on your need! See lubridate