



MANIPULATING TIME SERIES DATA IN R WITH XTS & ZOO

Merging time series

Introducing `merge()`

- Combine series by **column**
- `cbind()` and `merge()`
- Database style joins on `index` (i.e. by time)
 - Inner, outer, left and right joins

```
> merge(..., fill = NA, join = "outer")
```

- `fill` argument handles missingness

merge() example

```
> merge(x, y) # Default join = "outer"
```

	x	y
2016-08-09	1	2
2016-08-10	1	2
2016-08-11	1	NA
2016-08-12	NA	2

```
> merge(x, y, join = "inner")
```

	x	y
2016-08-09	1	2
2016-08-10	1	2

```
> merge(x, y, join = "right", fill = na.locf)
```

	x	y
2016-08-09	1	2
2016-08-10	1	2
2016-08-12	1	2

merge() example

```
> merge(x, c(2, 3, 4))  
      x c.2..3..4.  
2016-08-09 1      2  
2016-08-10 1      3  
2016-08-11 1      4
```

```
> merge(x, 3)  
      x X3  
2016-08-09 1  3  
2016-08-10 1  3  
2016-08-11 1  3
```

```
> merge(x, as.Date(c("2016-08-14")))  
      x  
2016-08-09 1  
2016-08-10 1  
2016-08-11 1  
2016-08-14 NA
```

Introducing `rbind()`

- Combine series by **row**
- Rows are inserted in time order
- All rows in `rbind()` must have a time
- The number of columns must match

rbind() example

```
> rbind(x, y)
      x
2016-08-09 1
2016-08-09 2
2016-08-10 1
2016-08-10 2
2016-08-11 1
2016-08-12 2
```

```
> rbind(x, as.integer(y))
Error in try.xts(c(2L, 2L, 2L)) :
  Error in as.xts.integer(x, ..., .RECLASS = TRUE) : order.by
must be either 'names()' or otherwise specified
```



MANIPULATING TIME SERIES DATA IN R WITH XTS & ZOO

Let's practice!



MANIPULATING TIME SERIES DATA IN R WITH XTS & ZOO

Handling missingness

Fill NAs with last observation

- l.o.c.f. means “last observation carried forward”

```
na.locf(object,  
         na.rm = TRUE,  
         fromLast = FALSE,  
         maxgap = Inf)
```

```
> cbind(z, na.locf(z), na.locf(z, fromLast = TRUE))  
      z z.1 z.2  
2016-08-09  1  1  1  
2016-08-10 NA  1  4  
2016-08-11 NA  1  4  
2016-08-12  4  4  4
```

Other NA options

- Replace NAs

```
na.fill(object, fill, ...)
```

- Remove NAs

```
na.trim(object, ...)  
na.omit(object, ...)
```

- Interpolate NAs

```
na.approx(object, ...)
```

NA replace and remove

```
> na.fill(z, fill = -999)
```

```
      z
Aug 09, 2016 1
Aug 10, 2016 -999
Aug 11, 2016 -999
Aug 12, 2016 4
Aug 13, 2016 -999
```

```
> na.trim(z)
```

```
      z
Aug 09, 2016 1
Aug 10, 2016 NA
Aug 11, 2016 NA
Aug 12, 2016 4
```

```
> na.omit(z)
```

```
      z
Aug 09, 2016 1
Aug 12, 2016 4
```

NA interpolation

- `na.approx()` uses index spacing to linearly approximate the missing values

```
> x
              x
Aug 09, 2016  1
Aug 11, 2016 NA
Aug 12, 2016  4
```

```
> na.approx(x)
              z
Aug 09, 2016  1
Aug 11, 2016  3
Aug 12, 2016  4
```



MANIPULATING TIME SERIES DATA IN R WITH XTS & ZOO

Let's practice!



MANIPULATING TIME SERIES DATA IN R WITH XTS & ZOO

Lags and differences


Seasonality and stationarity

- Seasonality is a repeating pattern
- Stationarity refers to some bound of the series
- These patterns are often compared
- How get around misalignment of the series?

Lagging a time series

- Used to align time series for comparisons
- `lag()` will shift observations in time

```
lag(x, k = 1, na.pad = TRUE, ...)
```

- `k` controls number of lags
- `na.pad` controls NA introduction
- With `xts`, *positive* `k` shifts values forward 

Differencing series

- Convert levels to changes (i.e. deltas)

```
diff(x,  
     lag = 1, differences = 1,  
     arithmetic = TRUE,  
     log = FALSE,  
     na.pad = TRUE, ...)
```

- Lag controls which observations
- Arithmetic vs. log calculations



MANIPULATING TIME SERIES DATA IN R WITH XTS & ZOO

Let's practice!