

Handling missingness

CASE STUDIES: MANIPULATING TIME SERIES DATA IN R



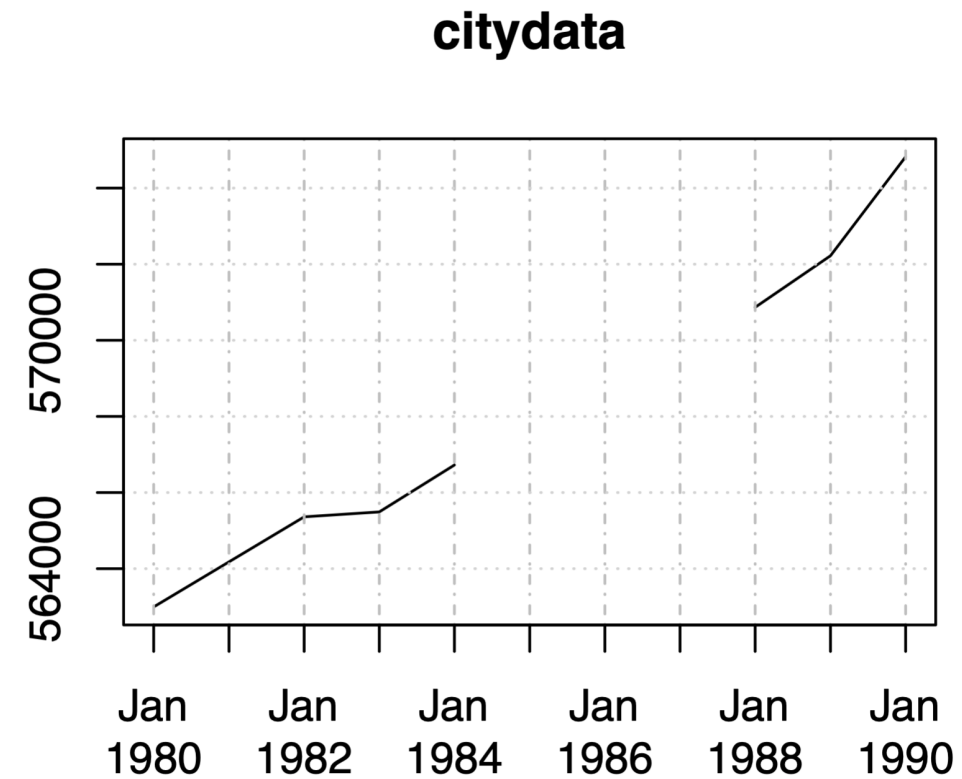
Lore Dirick

Manager of Data Science Curriculum at
Flatiron School

Missingness

citydata

	pop
1980-01-01	562994
1981-01-01	564179
1982-01-01	565361
1983-01-01	565491
1984-01-01	566723
1985-01-01	NA
1986-01-01	NA
1987-01-01	NA
1988-01-01	570867
1989-01-01	572222
1990-01-01	574823



Fill NAs with last observation

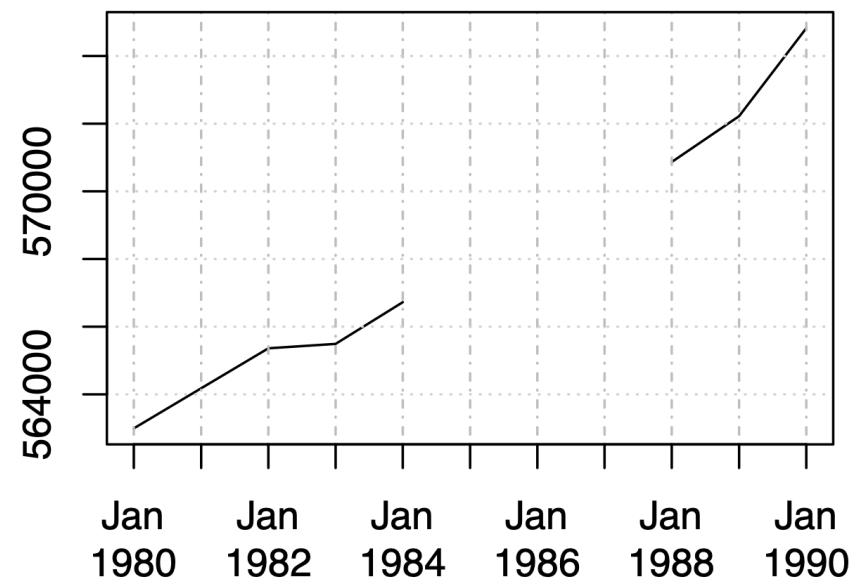
- Last observation carried forward (LOCF)

```
citydata_locf <- na.locf(citydata)
```

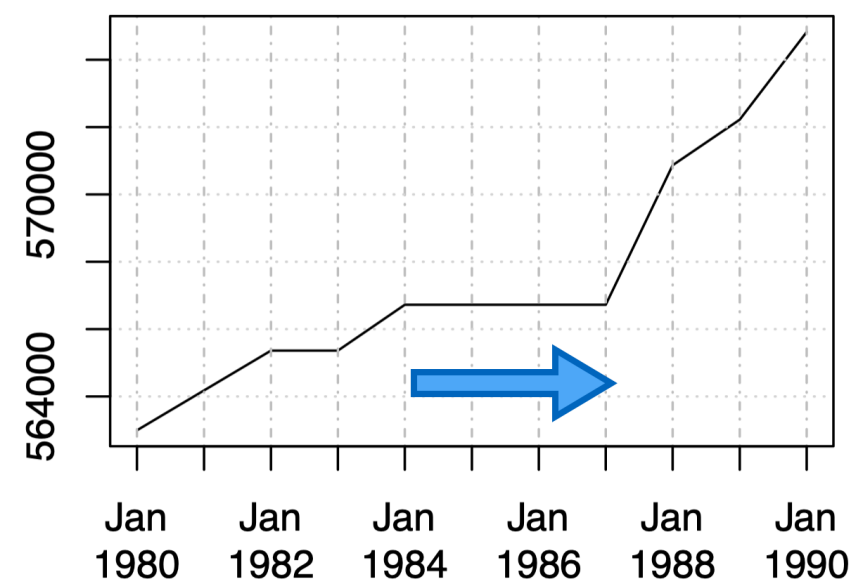
```
plot.xts(citydata)
```

```
plot.xts(citydata_locf)
```

citydata



citydata_locf



Fill NAs with next observation

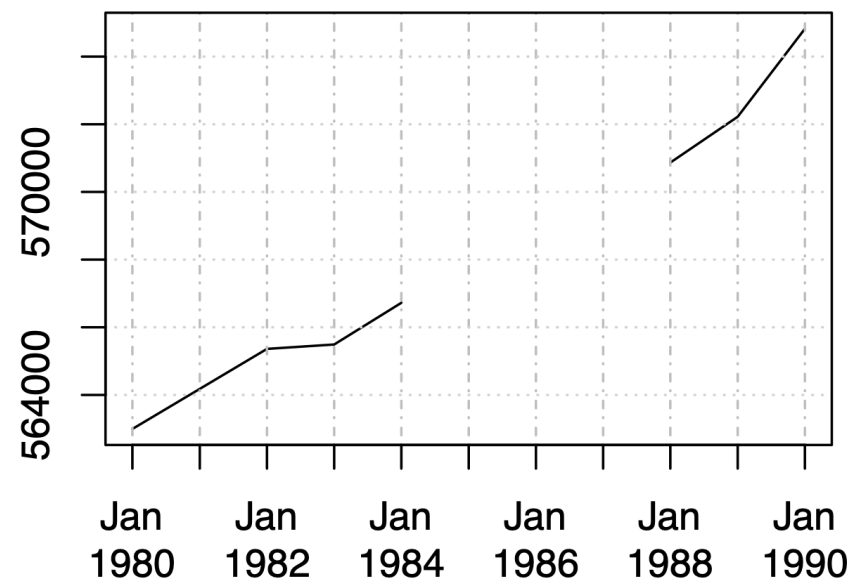
- Next observation carried backward (NOCB)

```
citydata_nocb <- na.locf(citydata, fromLast = TRUE)
```

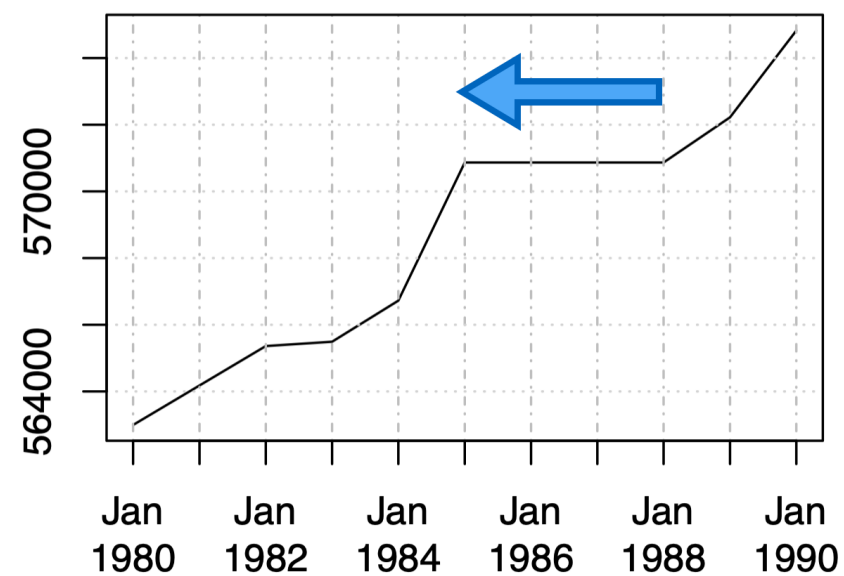
```
plot.xts(citydata)
```

```
plot.xts(citydata_nocb)
```

citydata



citydata_nocb



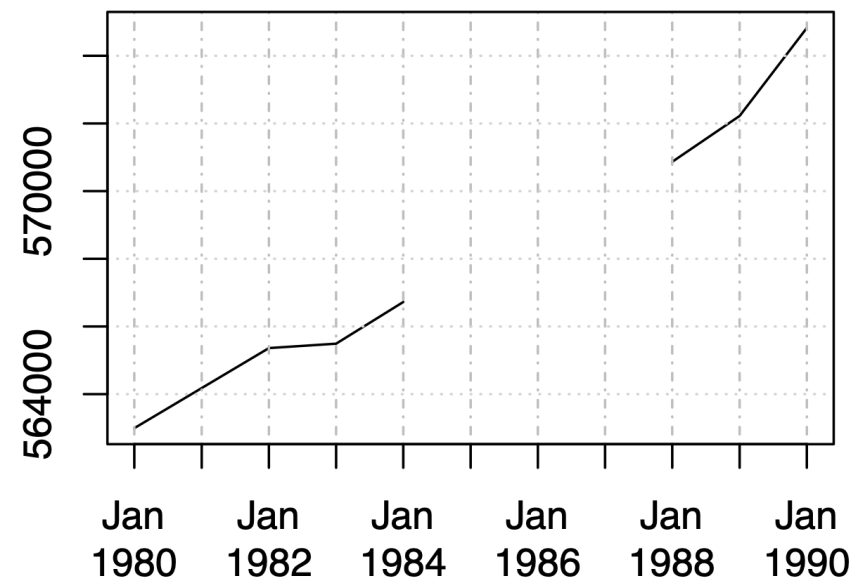
Linear interpolation

```
citydata_approx <- na.approx(citydata)
```

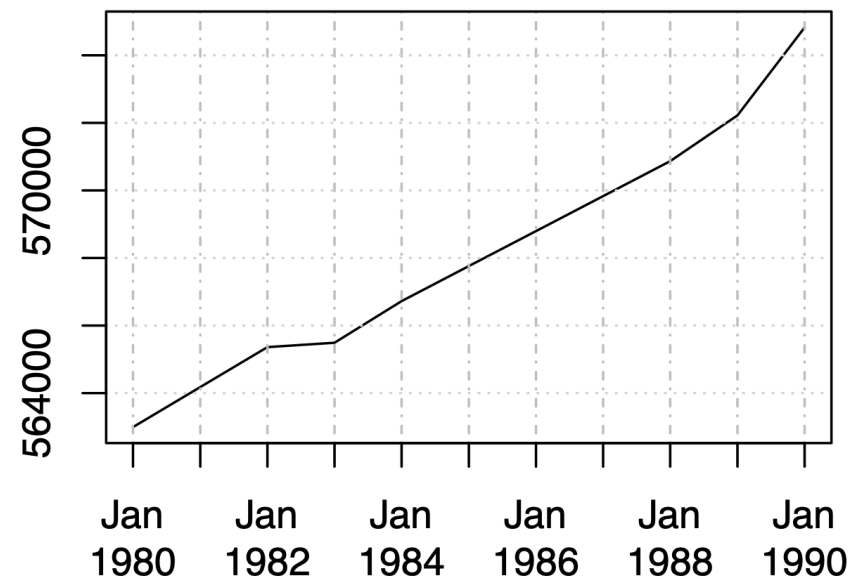
```
plot.xts(citydata)
```

```
plot.xts(citydata_nocb)
```

citydata



citydata_approx



Let's practice!

CASE STUDIES: MANIPULATING TIME SERIES DATA IN R

Lagging and differencing

CASE STUDIES: MANIPULATING TIME SERIES DATA IN R



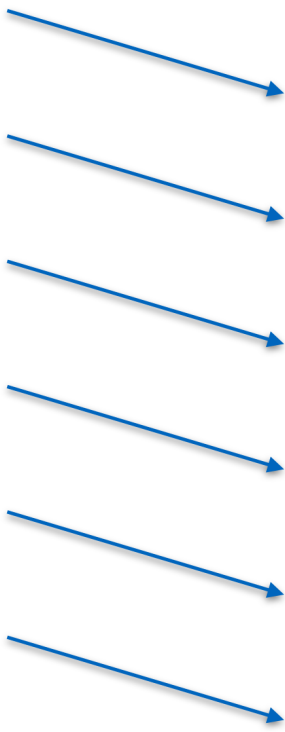
Lore Dirick

Manager of Data Science Curriculum at
Flatiron School

Lagging

- `lag()` offsets observations in time

```
lag(unemployment, k = 1, ...)
```



Jan 2010	9,6	-
Feb 2010	9,2	9,6
March 2010	8,9	9,2
April 2010	8,3	8,9
May 2010	8,2	8,3
June 2010	8,4	8,2
July 2010	8,3	8,4

Differencing

- `diff()` measures change between periods

```
diff(unemployment, lag = 1, ...)
```

Jan 2010	9,6	-
Feb 2010	9,2	-0,4
March 2010	8,9	-0,3
April 2010	8,3	-0,6
May 2010	8,2	-0,1
June 2010	8,4	0,2
July 2010	8,3	-0,1

Let's practice!

CASE STUDIES: MANIPULATING TIME SERIES DATA IN R

Rolling functions

CASE STUDIES: MANIPULATING TIME SERIES DATA IN R



Lore Dirick

Manager of Data Science Curriculum at
Flatiron School

Discrete windows

- Split the data according to period

```
unemployment_yrs <- split(unemployment, f = "years")
```

- Apply function within period

```
unemployment_yrs <- lapply(unemployment_yrs, cummax)
```

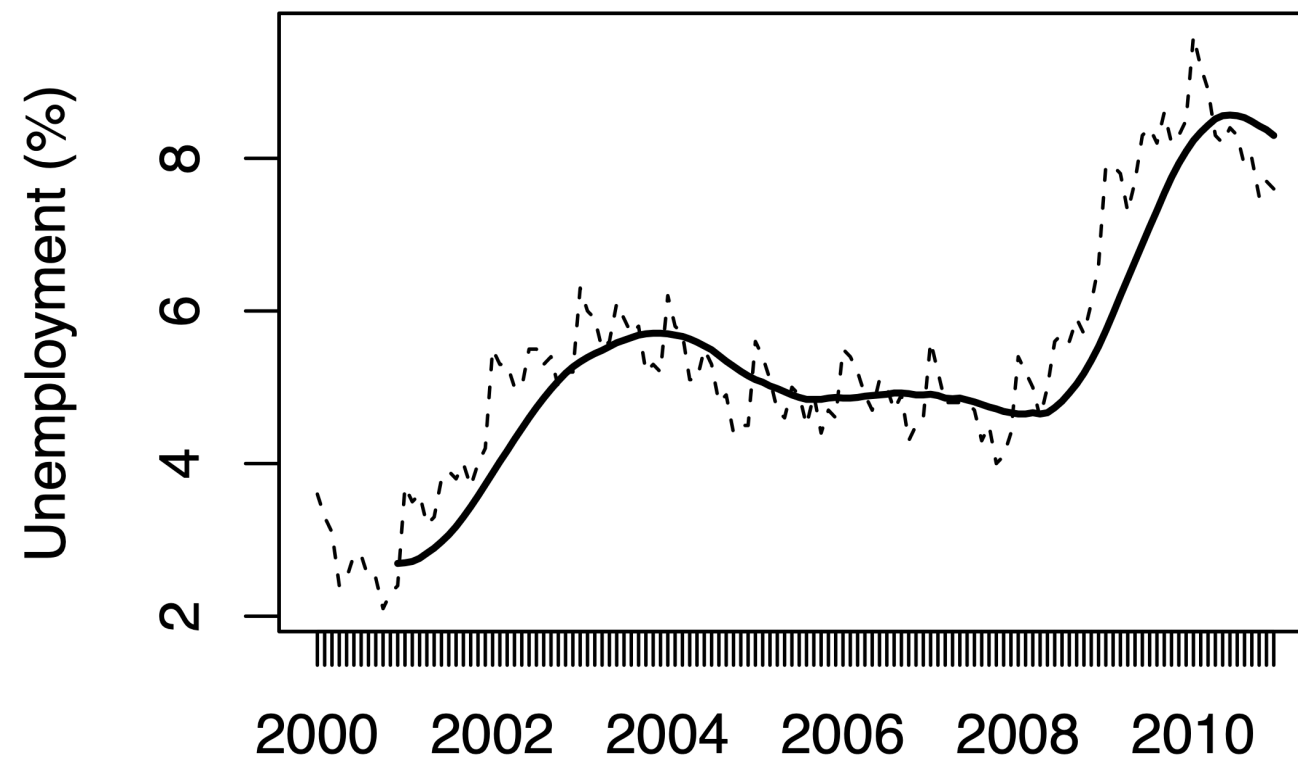
- Bind new data into xts object

```
unemployment_ytd <- do.call(rbind, unemployment_yrs)
```

Rolling windows

- `rollapply()` applies a function to a rolling window

```
unemployment_avg <- rollapply(unemployment,  
                               width = 12,  
                               FUN = mean)
```



Let's practice!

CASE STUDIES: MANIPULATING TIME SERIES DATA IN R