

### Step 3: Generate code with expandChain()

```
> expandChain(output$plot)
```

```
downloads <-  
  cranlogs::cran_downloads(  
    "shiny",  
    from = ..(format(Sys.Date() - 365)),  
    to = Sys.Date()  
  )
```

```
downloads_rolling <-  
  downloads %>%  
    mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))
```

```
ggplot(downloads_rolling, aes(date, count)) + geom_line()
```

Other code wrapped in `..()` is  
evaluated (i.e. unquoted)

### Step 3: Generate code with expandChain()

```
> expandChain(output$plot)
```

```
downloads <-  
  cranlogs::cran_downloads(  
    "shiny",  
    from = "2019-08-01",  
    to = Sys.Date()  
  )
```

```
downloads_rolling <-  
  downloads %>%  
    mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))
```

```
ggplot(downloads_rolling, aes(date, count)) + geom_line()
```

This allows dynamic results  
to be 'hard coded'

