

2. De-reference reactively using !!!

Besides inlining *values*, unquoting has a second essential function: inlining *meta-reactive objects* as code

```
> downloads <- metaReactive({  
+   cranlogs::cran_downloads(!!input$package,  
+   from = Sys.Date() - 365, to = Sys.Date())  
+ })
```

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

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

2. De-reference reactive exprs using !!

Besides inlining *values*, unquoting has a second essential function: inlining *meta-reactive objects* as code

```
> downloads <- metaReactive({
+   cranlogs::cran_downloads(!!input$package,
+     from = Sys.Date() - 365, to = Sys.Date())
+ })

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

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

2. De-reference reactive exprs using !!

Besides inlining *values*, unquoting has a second essential function: inlining *meta-reactive objects* as code

```
> downloads <- metaReactive({  
+   cranlogs::cran_downloads(!!input$package,  
+   from = Sys.Date() - 365, to = Sys.Date())  
+ })
```

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

Bang bang!

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