

~~> min_crat <- quantile(diamonds\$crat, probs=0.99)~~

> lang:expr(dplyr::filter(diams, carat >= !!min_carat))

`dp1yr$filter(diamonds, carat >= quantile(diamonds$carat, probs = 0.99))`

Use the unquoting operator to selectively replace quoted subexpressions (like `min_carat`) with **other quoted expressions**

The unquoting operator (!!)

The unquoting operator (!!)

```
> min_carat <- quote(quantile(diamonds$carat, probs = 0.99))  
  
> rlang::expr(dplyr::filter(diamonds, carat >= !!min_carat))  
dplyr::filter(diamonds, carat >= quantile(diamonds$carat, probs = 0.99))
```

Use the unquoting operator to selectively replace quoted subexpressions (like `min_carat`) with **other quoted expressions**

Further reading

- That's all we'll need to know about metaprogramming for the rest of this talk, but shinymeta works best if you have a solid grasp on the following:
 - Advanced R (Wickham) - chapters on Non-standard evaluation and Expressions
 - The dplyr vignette Programming with dplyr, specifically the section on quasiquotation