More NSE

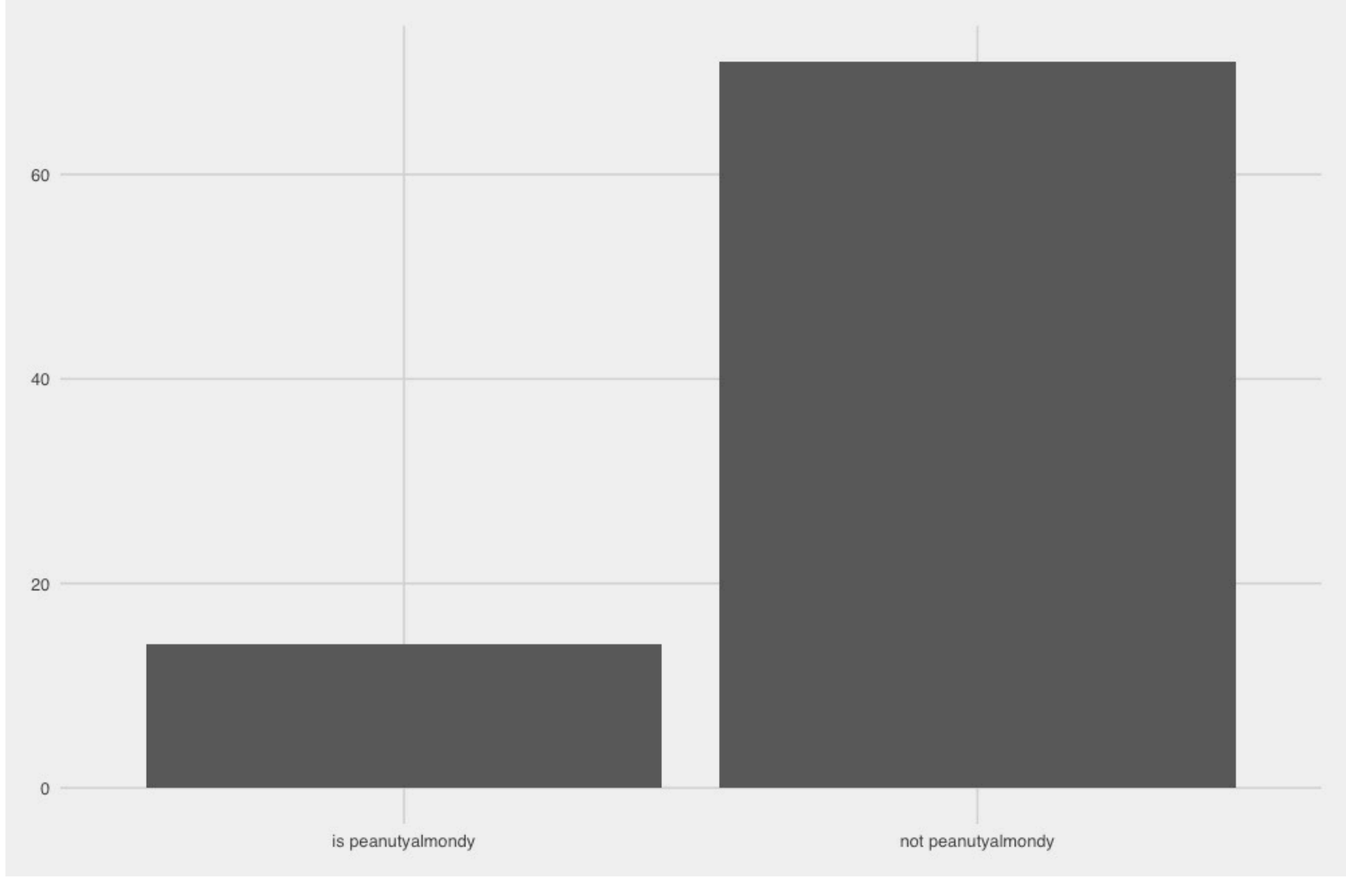
Recall

function	dplyr	base R
Get an unevaluated expression/call object	`quo()`	`quote()`
Substitute into an expression in a particular environment	`enquo()`	`substitute()`
Evaluate an R expression in a particular environment	`!!`	`eval()`

We'll be working with the fivethirtyeight dataset candy_rankings

We want to create a function that allows us to pass in a variable name and get out a bar chart

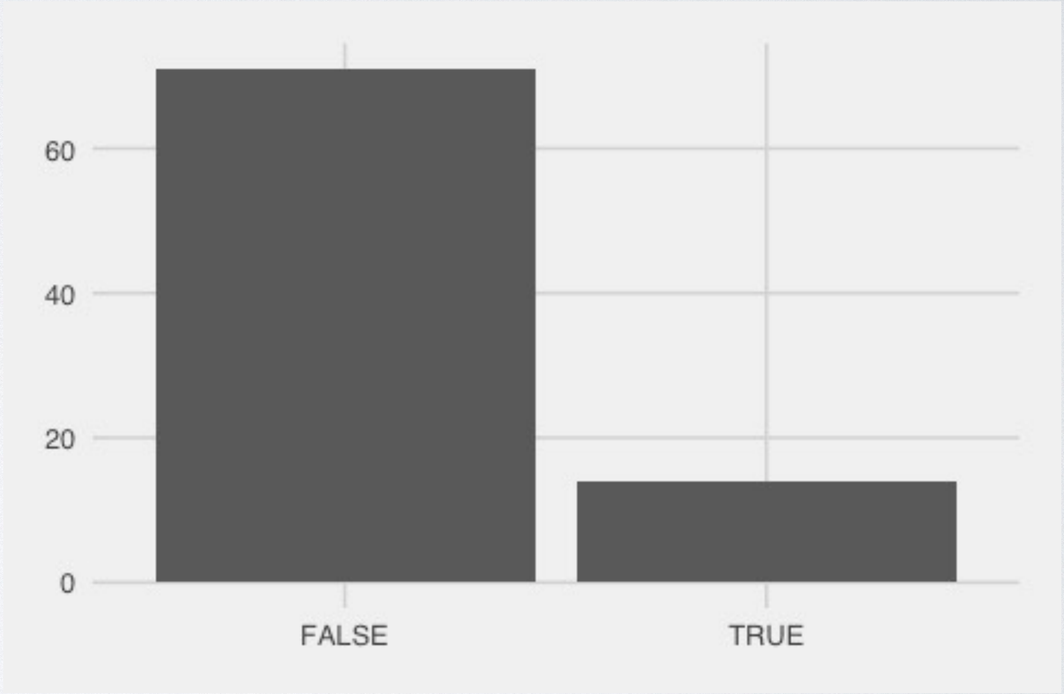
```
candy_rankings %>%

  candy_bar(peanutyalmondy)
```



Your Turn

Get the appropriate packages loaded and produce one bar chart of a variable in the `candy_rankings` dataset

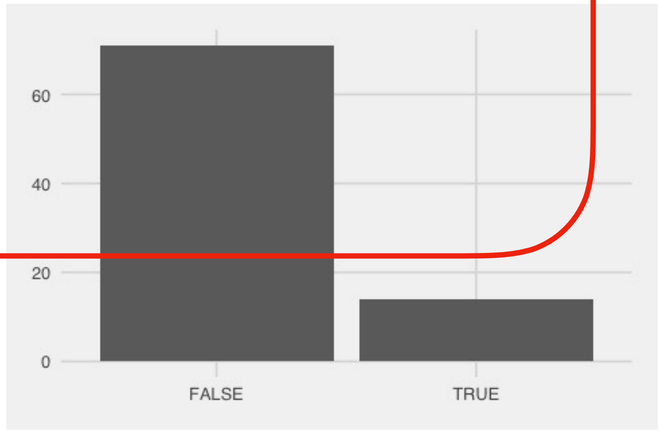
Remember the `ggthemes` package



```
library(ggplot2)
library(ggthemes)
ggplot(candy_rankings) + geom_bar(aes(x=chocolate)) +
theme_fivethirtyeight()
ggplot(candy_rankings) + geom_bar(aes(x=fruity)) +
theme_fivethirtyeight()
ggplot(candy_rankings) +
geom_bar(aes(x=peanutyalmondy)) +
theme_fivethirtyeight()
```

This is repetitive, so we want to write a function

```
library(ggplot2)
library(ggthemes)
ggplot(candy_rankings) + geom_bar(aes(x=chocolate)) +
theme_fivethirtyeight()
ggplot(candy_rankings) + geom_bar(aes(x=fruity)) +
theme_fivethirtyeight()
ggplot(candy_rankings) +
geom_bar(aes(x=peanutyalmondy)) +
theme_fivethirtyeight()
```



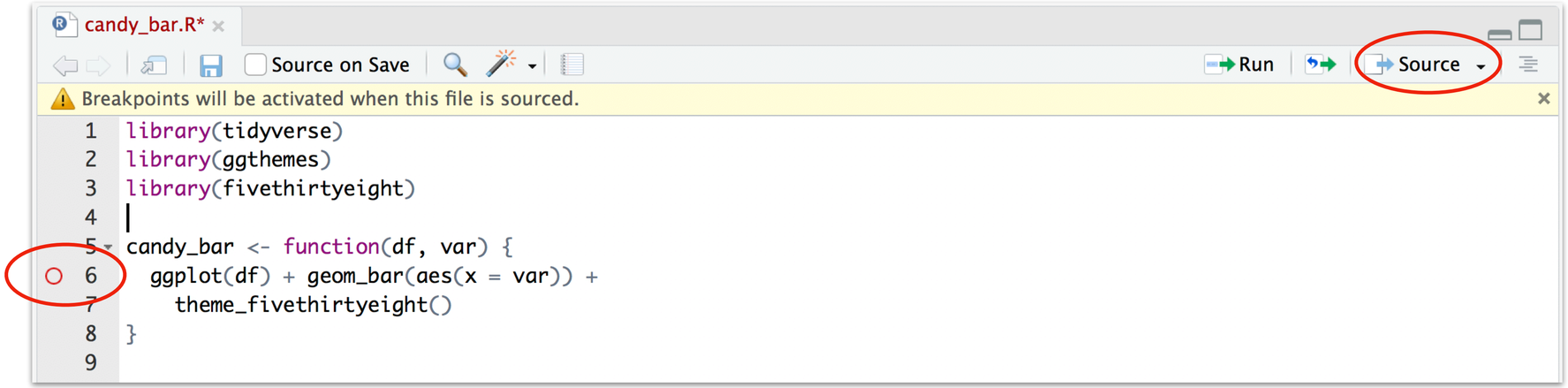
Your Turn

Start working on a function to do this task. We want it to take two arguments, a data frame and a variable name. Get at least the frame of the function and modify the ggplot code, it doesn't have to work yet.

This doesn't work yet, but I'm hoping you got to here

```
candy_bar <- function(df, var) {

  ggplot(df) + geom_bar(aes(x = var)) +

    theme_fivethirtyeight()

}
```

The reason it doesn't work is because of non-standard evaluation. Let's put this in a .R file, give it a name, and use a Breakpoint to debug it



Then click Source

Click next to a line
of the function

Once we're in Browse mode, we can see what the function is seeing

candy_bar.R

```
1  library(tidyverse)
2  library(ggthemes)
3  library(fivethirtyeight)
4
5  candy_bar <- function(df, var) {
6    ggplot(df) + geom_bar(aes(x = var)) +
7      theme_fivethirtyeight()
8  }
9
10
11
```

7:28 candy_bar(df, var) R Script

Console Terminal Compile PDF

~/STAT360/www/

Next Continue Stop

```
✖ dplyr::matches() masks testthat::matches()
> debugSource('~/STAT360/candy_bar.R')
> candy_bar(candy_rankings, chocolate)
Called from: eval(expr, p)
Browse[1]> n
debug at ~/STAT360/candy_bar.R#6: ggplot(df) + geom_bar(aes(x = var)) + theme_fivethirtyeight()
Browse[2]> var
Error: object 'chocolate' not found
In addition: Warning message:
restarting interrupted promise evaluation
Browse[2]> var
Error: object 'chocolate' not found
In addition: Warning messages:
1: In get(object, envir = currentEnv, inherits = TRUE) :
  restarting interrupted promise evaluation
2: restarting interrupted promise evaluation
Browse[2]> df
# A tibble: 85 x 13
   competitorname chocolate fruity caramel peanutyalmondy nougat crispedricewafer hard  bar   pluribus sugarpercent
   <chr>          <lgl>     <lgl>  <lgl>   <lgl>          <lgl>  <lgl>            <lgl> <lgl> <lgl>           <dbl>
 1 100 Grand      TRUE      FALSE  TRUE    FALSE          FALSE  TRUE             FALSE TRUE  FALSE           0.732
 2 3 Musketeers   TRUE      FALSE  FALSE   FALSE          TRUE   FALSE            FALSE TRUE  FALSE           0.604
 3 One dime       FALSE     FALSE  FALSE   FALSE          FALSE  FALSE            FALSE FALSE FALSE           0.011
 4 One quarter    FALSE     FALSE  FALSE   FALSE          FALSE  FALSE            FALSE FALSE FALSE           0.011
 5 Air Heads      FALSE     TRUE   FALSE   FALSE          FALSE  FALSE            FALSE FALSE FALSE           0.906
 6 Almond Joy     TRUE      FALSE  FALSE   TRUE           FALSE  FALSE            FALSE TRUE  FALSE           0.465
 7 Baby Ruth      TRUE      FALSE  TRUE    TRUE           TRUE   FALSE            FALSE TRUE  FALSE           0.604
 8 Boston Baked … FALSE     FALSE  FALSE   TRUE           FALSE  FALSE            FALSE FALSE TRUE            0.313
 9 Candy Corn     FALSE     FALSE  FALSE   FALSE          FALSE  FALSE            FALSE FALSE TRUE            0.906
10 Caramel Apple… FALSE     TRUE   TRUE    FALSE          FALSE  FALSE            FALSE FALSE FALSE           0.604
# ... with 75 more rows, and 2 more variables: pricepercent <dbl>, winpercent <dbl>
Browse[2]>
```

Your Turn

Try to figure out which of these functions will help us here. You may want to look at the Programming with dplyr vignette again.

Test out a few of them, and use debugging to see if you're right.

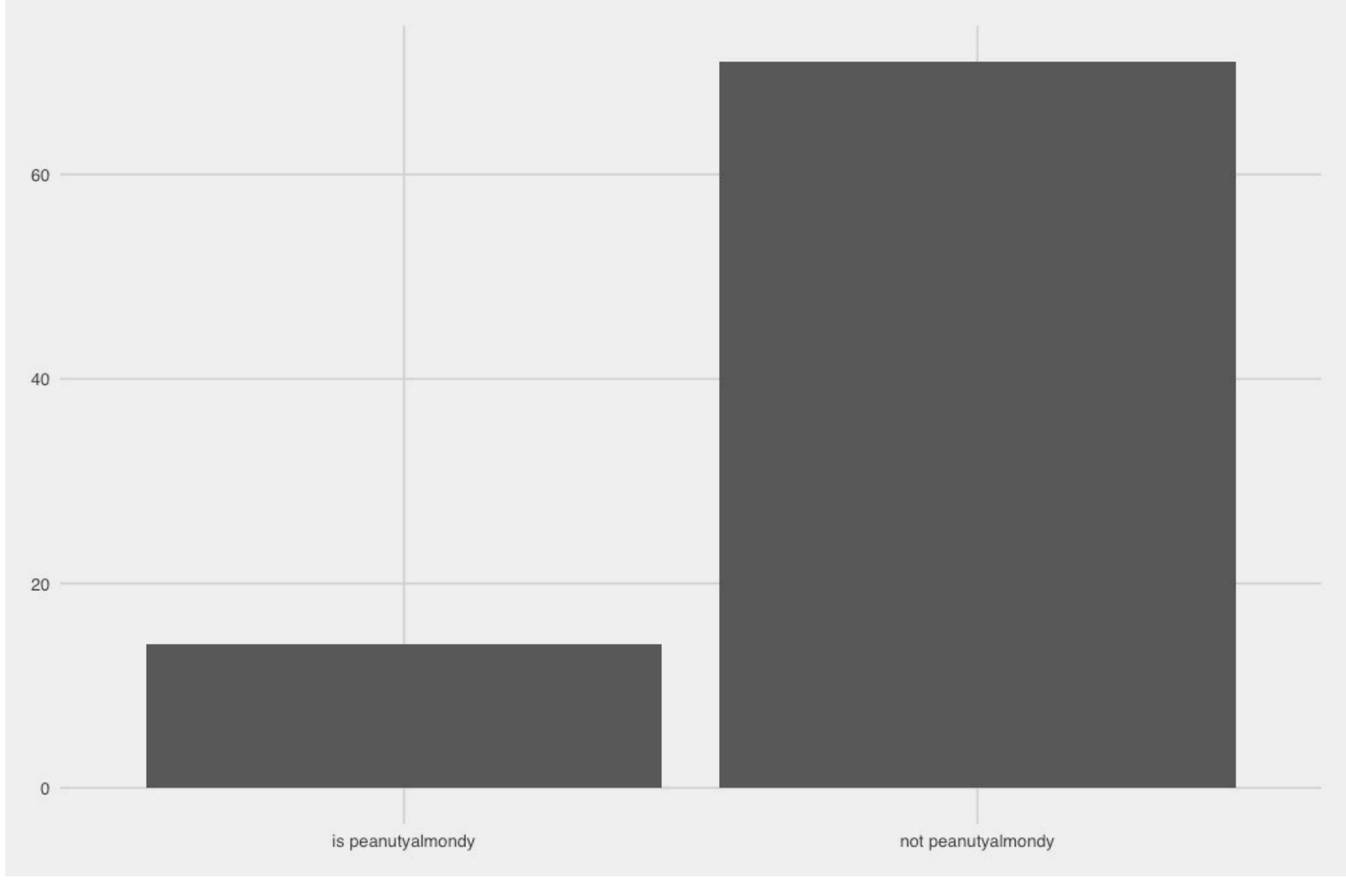
function	dplyr	base R
Get an unevaluated expression/call object	`quo()`	`quote()`
Substitute into an expression in a particular environment	`enquo()`	`substitute()`
Evaluate an R expression in a particular environment	`!!`	`eval()`

One solution (there may be better ones)

```
candy_bar <- function(df, var) {
  ggplot(df) + geom_bar(aes(x = !! enquo(var))) +
    theme_fivethirtyeight()
}
```

The final aspect is we'd like to be able to change the labels on the plot so it reflects the variable, like we can see below.

```
candy_rankings %>%

  candy_bar(peanutyalmondy)
```



Your Turn

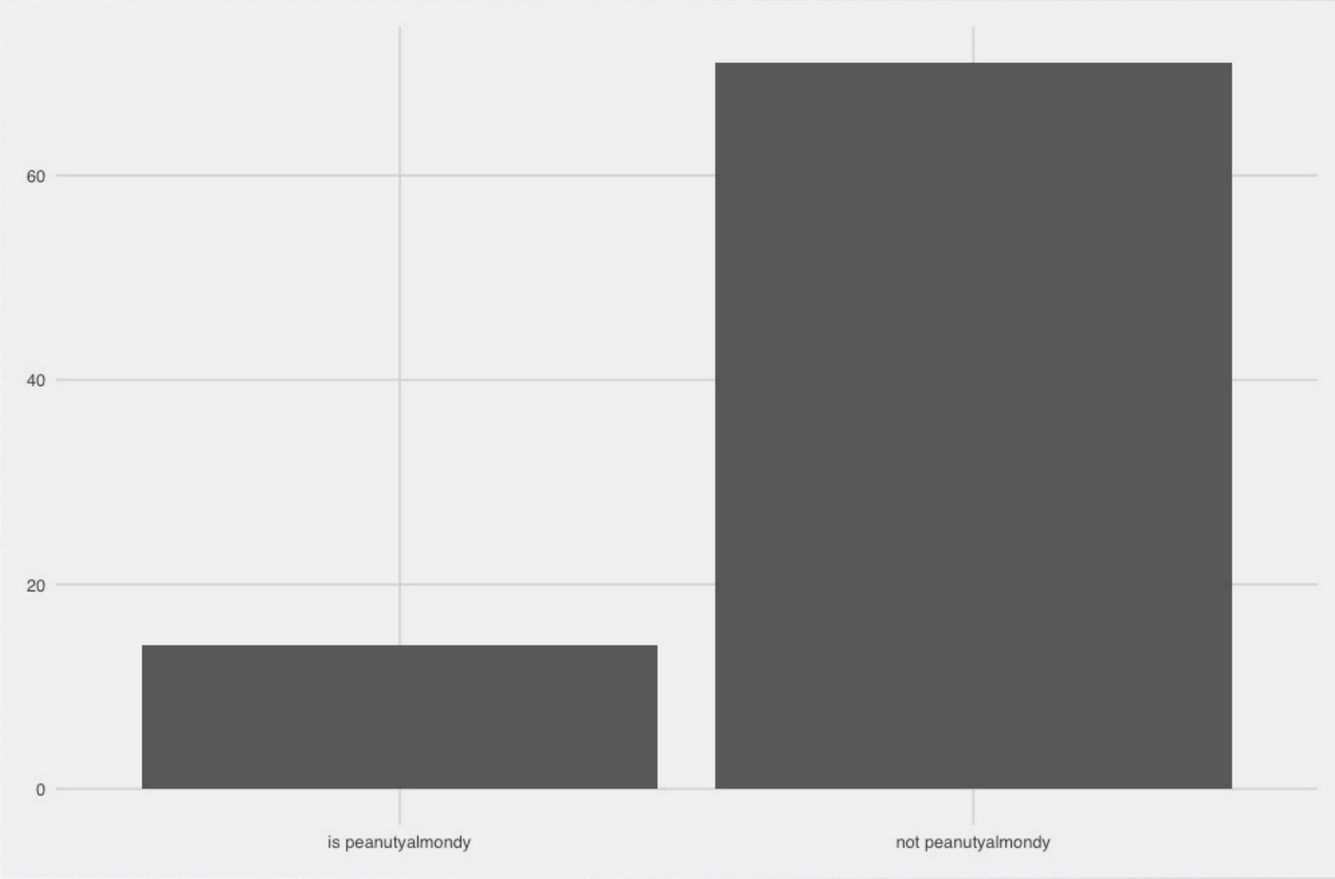
Work on getting the labeling to customize. Again, there are many ways to do this but the way I did it was:

- made a new variable using mutate that said “is peanutyalmondy” for TRUE and “not peanutyalmondy” for FALSE

- this required the use of the `if_else()` and `paste()` functions

- it also required the use of NSE in the `mutate()` call

- used the new variable as-is (no NSE) in my plotting function



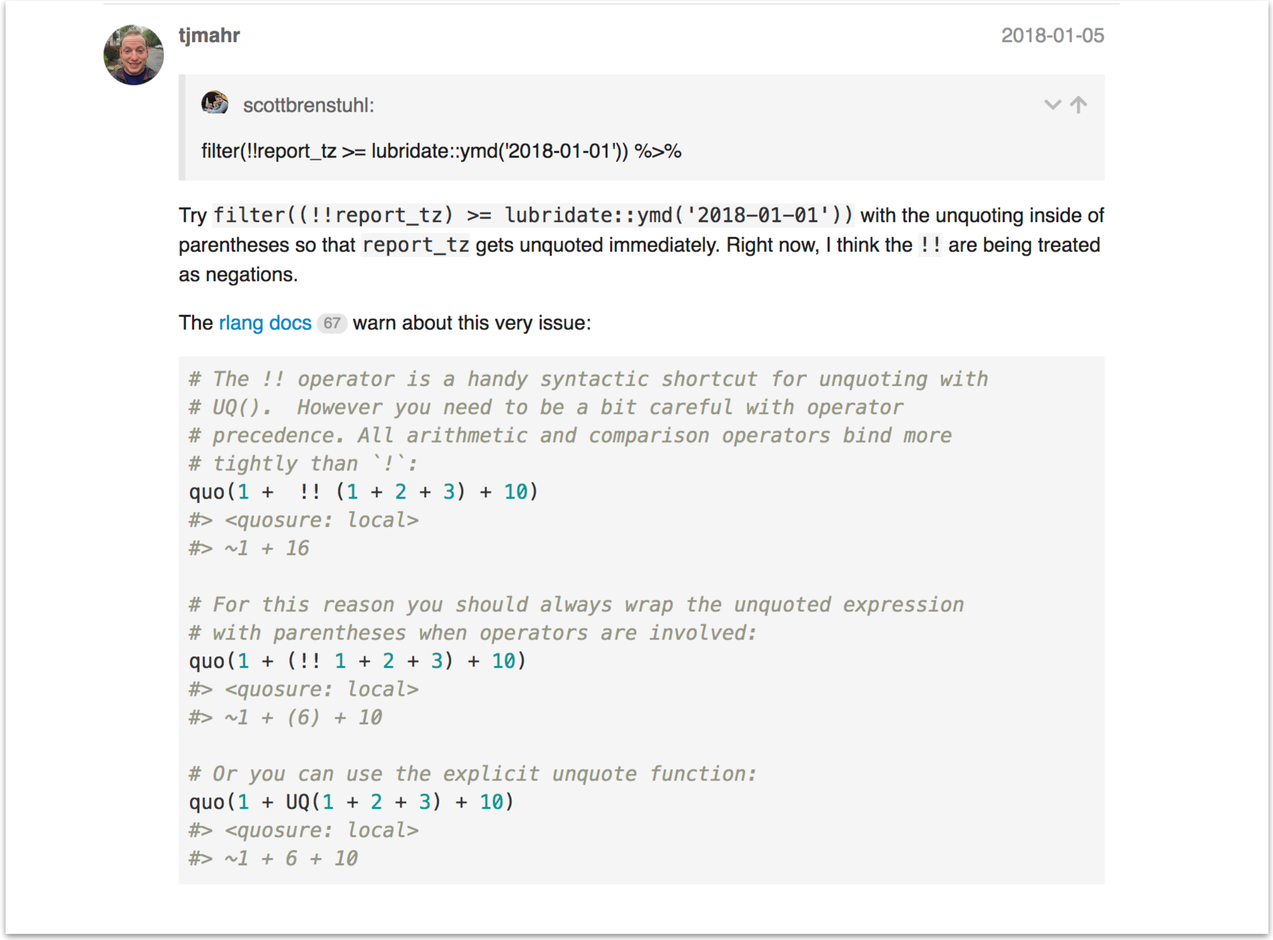
A (probably sub-optimal) solution

```
candy_bar <- function(df, var) {
  variable <- enquo(var)
  what <- substitute(var)
  df <- df %>%
    mutate(new_var = if_else(!!variable,
                             paste("is", what), paste("not", what)))
  ggplot(df) + geom_bar(aes(x = new_var)) +
     theme_fivethirtyeight()
}
```

Another (maybe better?) solution

```
candy_bar <- function(df, var) {
  variable <- substitute(var)
  ggplot(df) + geom_bar(aes(x = !! variable)) +
    theme_fivethirtyeight() +
    scale_x_discrete(labels = c(paste("not", variable),
paste("is", variable)))
}
```

A word of warning— sometimes “bang bang” gets interpreted as negation



tjmahr 2018-01-05

> scottbrenstuhl:
>
> filter(!!report_tz >= lubridate::ymd('2018-01-01')) %>%

Try `filter((!!report_tz) >= lubridate::ymd('2018-01-01'))` with the unquoting inside of parentheses so that `report_tz` gets unquoted immediately. Right now, I think the `!!` are being treated as negations.

The rlang docs 67 warn about this very issue:

```
# The !! operator is a handy syntactic shortcut for unquoting with
# UQ().  However you need to be a bit careful with operator
# precedence. All arithmetic and comparison operators bind more
# tightly than `!`:
quo(1 +  !! (1 + 2 + 3) + 10)
#> <quosure: local>
#> ~1 + 16

# For this reason you should always wrap the unquoted expression
# with parentheses when operators are involved:
quo(1 + (!! 1 + 2 + 3) + 10)
#> <quosure: local>
#> ~1 + (6) + 10

# Or you can use the explicit unquote function:
quo(1 + UQ(1 + 2 + 3) + 10)
#> <quosure: local>
#> ~1 + 6 + 10
```