Advanced R

Chapter 5

R4DS Reading Group

CONTROL FLOW

- `if`
- `ifelse`
- `case_when`
- `switch`
- `for`
- `while`
- `repeat`

But First... BEER!

- State (abbreviated)
- Year
- Barrels (barrels produced)
- Type (On premise, Bottles/Cans, Kegs/Barrels)

state	year	barrels	type
AK	2008	2067.69	On Premises
AK	2009	2263.65	On Premises
AK	2010	1929.15	On Premises
AK	2011	2251.02	On Premises
AK	2012	2312.43	On Premises
AK	2013	2155.60	On Premises

If and ifelse

Let's see if a random state in our dataset is my home town, NV

IF

```
if (sample(beer_states$state, 1) == "NV") print("My Home State")
```

IF ELSE

```
if (sample(beer_states$state, 1) == "NV") print("My Home State") else print("Not my home")
```

```
## [1] "Not my home"
```

IFELSE

```
ifelse((sample(beer_states$state, 1) == "NV"), print("My Home State"), print("Not my home"))
```

```
## [1] "Not my home"
```

```
## [1] "Not my home"
```

*Why does the **`ifelse`** statement print "Not my home" twice?*

CASE WHEN

Let's change the barrels column to categorical

```
beer_states %>%
  mutate(
    barrel_cat =
      if (barrels >= 100000000) {
        "A lot"
      } else if (barrels >= 10000000) {
        "Many"
      } else if (barrels >= 1000000) {
        "A few"
      } else {
        "Not much"
      }
    )
```

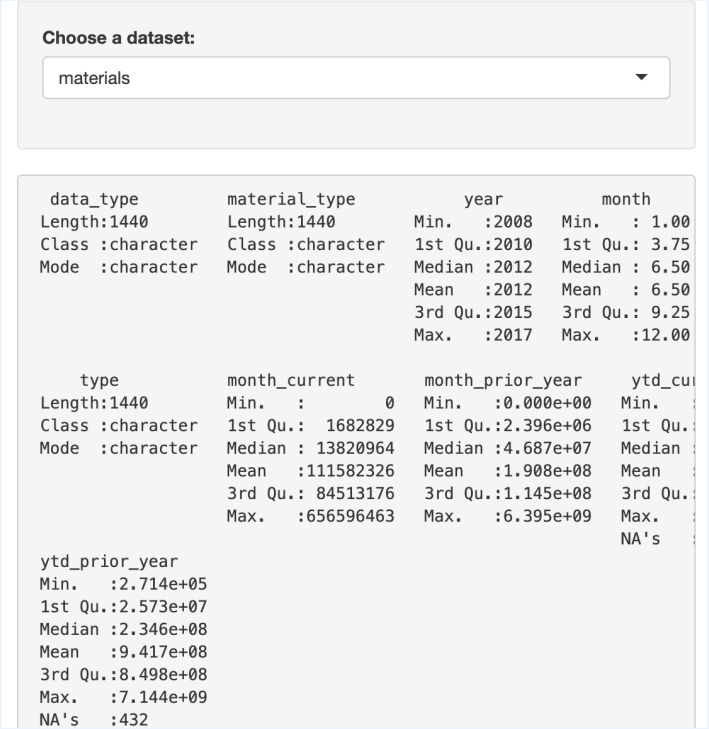
```
beer_states %>%
  mutate(
    barrel_cat = case_when(
      barrels >= 100000000 ~ "A LOT!",
      barrels >= 10000000 ~ "Many",
      barrels >= 1000000 ~ "A few",
      TRUE ~ "Not much"
    )
  )
```

barrel_cat	n
A LOT!	12
Many	98
Not much	1569
A few	193

SWITCH

Let's make a small shiny app to see the number of barrels per state

```
library(shiny)
library(tidyverse)

brewing_materials <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidy
beer_taxed <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/
brewer_size <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday
beer_states <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday


ui <- fluidPage(
    sidebarLayout(
        sidebarPanel(
            selectInput(inputId = "dataset",
                        label = "Choose a dataset:",
                        choices = c("materials", "size", "states", "taxed"))
        ),
        mainPanel(
            verbatimTextOutput("summary")
        )
    )
)

server <- function(input, output) {

    datasetInput <- reactive({
        switch(input$dataset,
               "materials" = brewing_materials,
               "size" = brewer_size,
               "states" = beer_states,
               "taxed" = beer_taxed)
    })

    output$summary <- renderPrint({
        dataset <- datasetInput()
        summary(dataset)
    })

}

shinyApp(ui = ui, server = server)
```

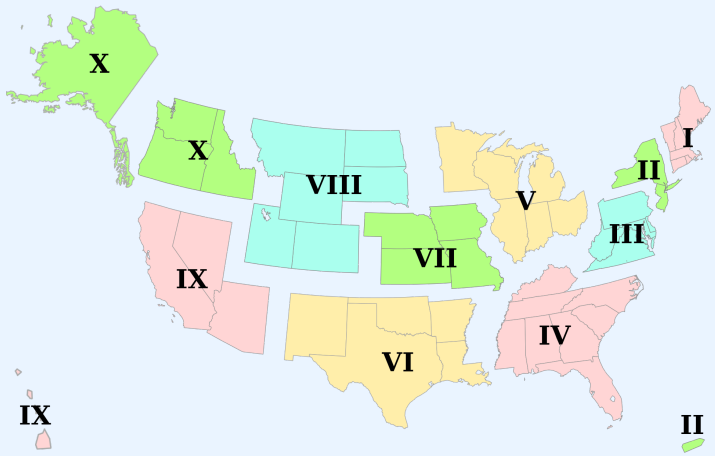



```
runApp("Presentations/Week5/switch_app/app.R", display.mode = "showcase")
```

FOR

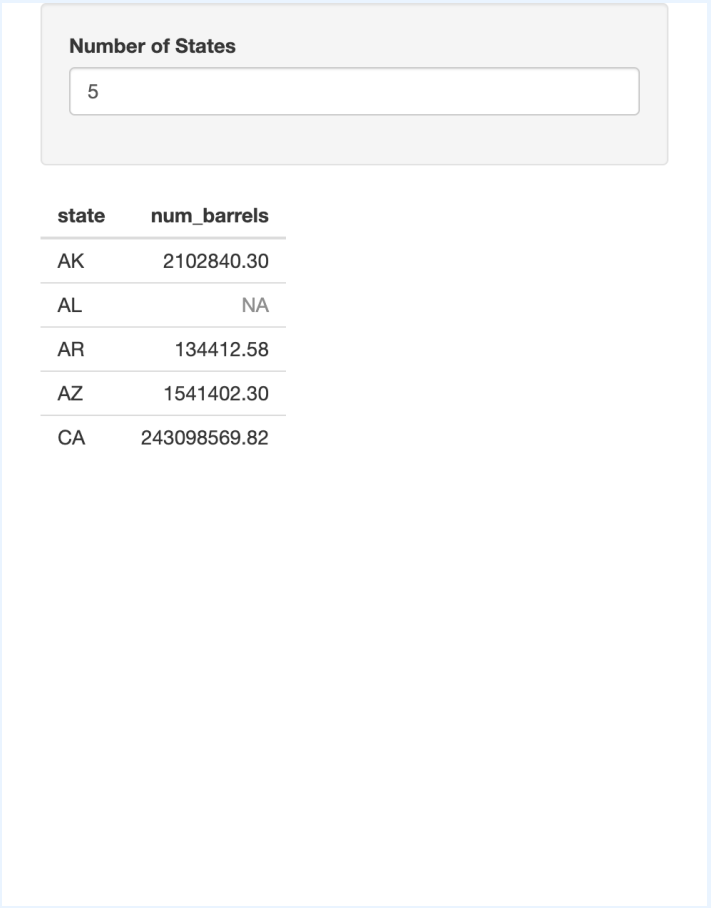
Create a "region" column based on each observatation's state

```
for (i in 1:nrow(beer_states)) {
  if (beer_states$state[i] %in% Region1) {
    beer_states$region[i] <- "Region 1"
  } else if (beer_states$state[i] %in% Region2) {
   beer_states$region[i] <- "Region 2"
  } else if (beer_states$state[i] %in% Region3) {
    beer_states$region[i] <- "Region 3"
  } else if (beer_states$state[i] %in% Region4) {
    beer_states$region[i] <- "Region 4"
  } else if (beer_states$state[i] %in% Region5) {
     beer_states$region[i] <- "Region 5"
  } else if (beer_states$state[i] %in% Region6) {
     beer_states$region[i] <- "Region 6"
  } else if (beer_states$state[i] %in% Region7) {
    beer_states$region[i] <- "Region 7"
  } else if (beer_states$state[i] %in% Region8) {
    beer_states$region[i] <- "Region 8"
  } else if (beer_states$state[i] %in% Region9) {
    beer_states$region[i] <- "Region 9"
  } else if (beer_states$state[i] %in% Region10) {
    beer_states$region[i] <- "Region 10"
  } else {
    beer_states$region[i] <- "Missing"
  }
}
```



WHILE

```
library(shiny)

beer_states <- readr::read_csv('https://raw.githubu

ui <- fluidPage(
    sidebarPanel(numericInput('xqty', 'Number of St
    mainPanel(tableOutput("while_debug")))

server <- function(input, output, session) {

   states <- unique(beer_states$state)

   my_vector <- reactive({
        i <- 0
        my_vector <- vector()
        while (i <= input$xqty) {
            my_vector[i] <- i
            i = i+1
        }
        return(my_vector)
    })

    output$while_debug <- renderTable({

        beer_states %>%
            filter(state %in% unique(beer_states$st
            filter(state != "total") %>%
            group_by(state) %>%
            summarise(num_barrels = sum(barrels))
    })
}
shinyApp(ui = ui, server = server)
```



state	num_barrels
AK	2102840.30
AL	NA
AR	134412.58
AZ	1541402.30
CA	243098569.82

REPEAT

Let's revist beer in Nevada - rather than take one sample, we can use repeat to continue sampling until we find beer!

```
repeat {
  if (sample(beer_states$state, 1) == "NV") {
    print("Go grab a beer!");
    break
  } else print("Maya doesn't live here");
}
```

```
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Maya doesn't live here"
[1] "Go grab a beer!"
```

ALL TOGETHER!

This diagram is incomplete - let's improve it together!