

Control Structures

Naomi Tague

January, 2023

Review DataTypes.Rmd

Two key “take homes”

- working with *factors*
- how to return multiple items from a function using *list*

How do you figure out the rarest fish in our simulated ocean?

Try it - generate a simulated ocean by sampling; and then uses summary to find the rarest fish

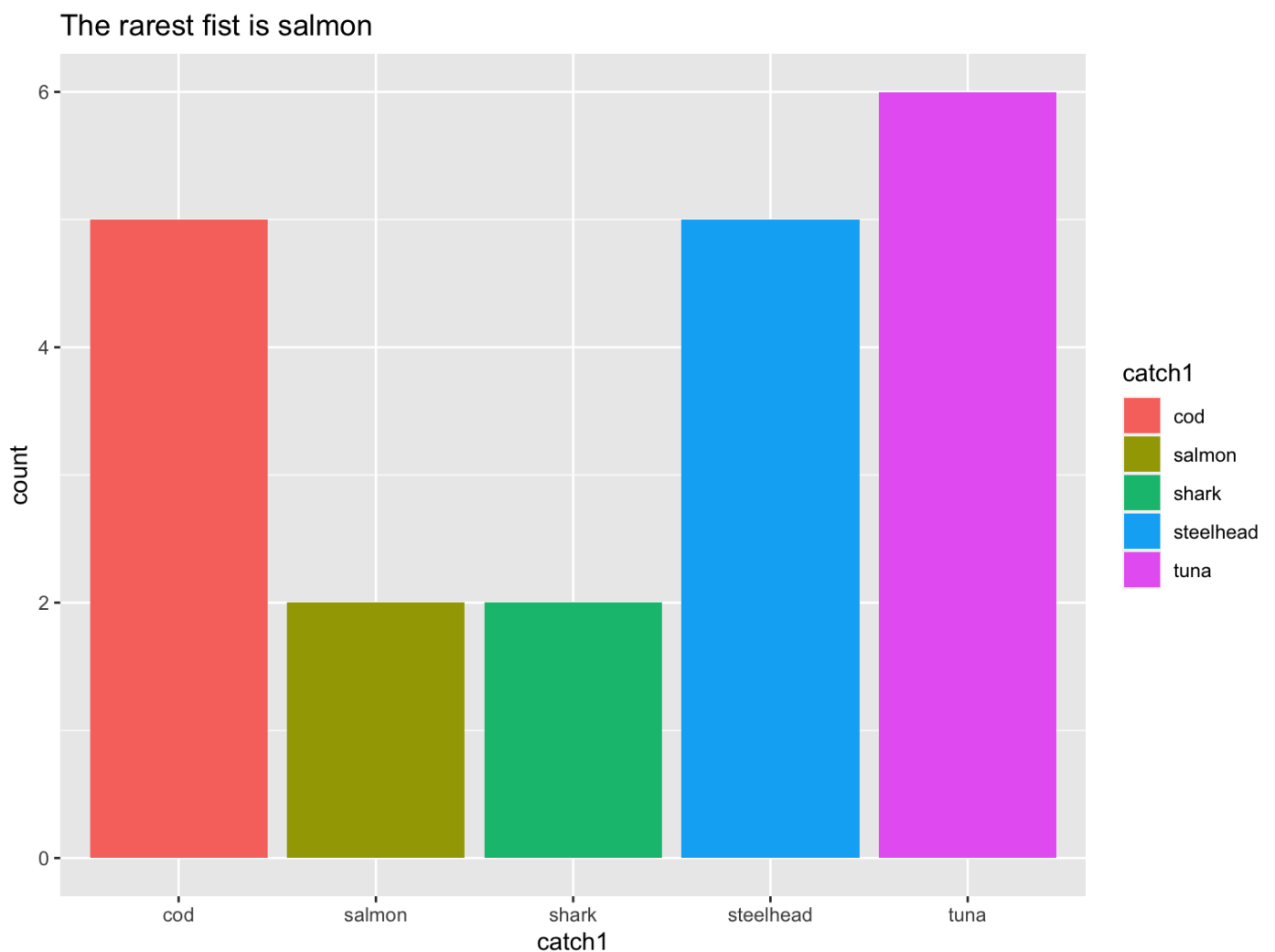
Answer

```
possible.fish = c("salmon","steelhead","shark","tuna","cod")
catch1 = base::sample(possible.fish, size=20, replace=T)
rarestfish = names(which.min(summary(as.factor(catch1))))
rarestfish
```

```
## [1] "salmon"
```

```
plottitle = sprintf("The rarest fist is %s", rarestfish)
ggplot(data.frame(catch1=catch1), aes(catch1, fill=catch1))+geom_histogram(stat="count")+labs(title=plottitle)
```

```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```



Flow Control (think of steering your program)

Another KEY concept is flow control

Allowing your function to “do” different things depending on a conditions

CLASSIC example is

IF then *ELSE*

If you have multiple conditions we can use *case_when*

Here's a silly simple example of how it works

Simple example of flow control with if

```
mycortest = function(x,y, thresh=0.8) {  
  # compute correlation  
  res = cor(x,y)  
  classification = ifelse(res > thresh, "GOOD", "NotGood")  
  return(classification)  
}  
  
a = runif(min=1, max=100, n=100)  
b = runif(min=1, max=100, n=100)  
  
mycortest(a,b)
```

```
## [1] "NotGood"
```

```
mycortest(a,a)
```

```
## [1] "GOOD"
```

```
# this doesn't work - why?  
mycortest(a, 1)
```

```
## Error in cor(x, y): incompatible dimensions
```

```
# useful to add error checking  
mycortest = function(x,y, thresh=0.8) {  
  # compute correlation  
  if(length(x) != length(y)) {stop("unequal lengths for x and y")}  
  res = cor(x,y)  
  classification = ifelse(res > thresh, "GOOD", "NotGood")  
  return(classification)  
}  
mycortest(a, 1)
```

```
## Error in mycortest(a, 1): unequal lengths for x and y
```

Flow control with a simple if

```
# Simple "IF"  
# imagine we are trying to get a tuna - Lets "fish" by sampling  
possible.fish = c("salmon","steelhead","shark","tuna","cod")  
catch1 = base::sample(possible.fish, size=1, replace=T)  
catch1
```

```
## [1] "cod"
```

```
ifelse(catch1 == "tuna", "success", "tryagain")
```

```
## [1] "tryagain"
```

```
catch1 = "tuna"  
ifelse(catch1 == "tuna", "success", "tryagain")
```

```
## [1] "success"
```

Multiple Alternatives

What if we have more than one category of fish

grade A, B, C

- steelhead are A,
- tuna are B
- everything else is C

R and other languages have ways to do this multiple alternatives flow control in R an example is

case_when

`case_when(`

`condition ~ response,`

`condition ~ response ...)`

```
fish = "steelhead"

case_when((fish == "steelhead") ~ "A", (fish == "tuna") ~ "B", !(fish == "tuna") & !(fish == "steelhead") ~ "C")
```

```
## [1] "A"
```

```
# apply to all of our ocean
# start by making a function
classify_fish = function(fish) {
  class= case_when((fish == "steelhead") ~ "A", (fish == "tuna") ~ "B", !(fish == "tuna") & !(fish == "steelhead") ~ "C")
  return(class) }

classify_fish("steelhead")
```

```
## [1] "A"
```

```
classify_fish("cod")
```

```
## [1] "C"
```

```
# apply to our ocean
possible.fish = c("salmon", "steelhead", "shark", "tuna", "cod")
catch2 = base::sample(possible.fish, size=20, replace=T)
catch2
```

```
## [1] "shark"      "salmon"      "steelhead"   "shark"      "tuna"      "steelhead"
## [7] "steelhead"  "tuna"        "salmon"      "cod"        "salmon"    "steelhead"
## [13] "shark"      "steelhead"   "salmon"      "salmon"     "tuna"      "salmon"
## [19] "tuna"      "steelhead"
```

```
classify_fish(catch2)
```

```
## [1] "C" "C" "A" "C" "B" "A" "A" "B" "C" "C" "C" "A" "C" "A" "C" "C" "B" "C" "B"
## [20] "A"
```


A more interesting example

Lets imagine that we are monitoring pollution in a lake, and we want to write a function that will let us know (flag) if risk associated with nutrient pollution are high, medium or low

From ecological studies, we know that

High Risk Conditions

- risk is high if water temperature is greater than a threshold for more than 5 days, and
- mean nutrient concentration is greater than a high threshold

Medium Risk Conditions

- Risk is medium if water temperature is greater than a threshold for more than 5 days and
- mean nutrient concentration is greater than a medium threshold

Everything else is low risk

Design the function

Inputs:

- nutrient concentration for at least 5 days
- temperature for at least 5 days
- thresholds for temperature and nutrient (with default values)

Output:

- Mean Nutrient Concentration
- Pollution Risk as “low”, “med” or “high”

Take a look at `classify_lake`

- example of while
- example of flow control with `if..else` and `case_when`

```
source("../R/classify_lake.R")

# generate some data to try the function
temperature = runif(min=4, max=35, n=30)
nutrient = runif(min=5, max=40, n=30)

# try it
res=classify_lake(temperature=temperature, nutrient=nutrient)

# Lets pick some numbers we know should give us a low value
nutrient = runif(min=0, max=5, n=30)
classify_lake(temperature=temperature, nutrient=nutrient)
```

```
## $risk
## [1] "low"
##
## $mean_nutrient
## [1] 2.419575
```

```
# now try high
# use repeat to generate the same value multiple times
nutrient = rep(50, times=30)
temperature = rep(25, times=30)
classify_lake(temperature=temperature, nutrient=nutrient)
```

```
## $risk
## [1] "high"
##
```

```
## $mean_nutrient  
## [1] 50
```

What we've learned

- how to write a function (and add error checking)
- how to generate data
- how to repeat in code (different types of looping)
- how to make choices (flow control)

Assignment

Write a function that takes a vector of fish names and always returns three items

- the most common fish,
- the rarest fish
- the total number of fish

Create an Rmarkdown to demonstrate the use of your function with fish.txt - which is under Data on ESM_262_Examples

[Data on ESM_262_Examples]

{https://github.com/naomitague/ESM_262_Examples/blob/main/Data/fish.txt}

Turn in on Gauchospace what your function returns when you run with fish.txt!

Challenge: What if we had multiple catches - how would you run your summary function for all of those catches - see below for an example to generate multiple catches You don't have to run this one in but we will go over in class

```
# generate some data if we had 10 different fishing days
# some examples of using the purr family of functions

possible.fish = c("salmon","steelhead","shark","tuna","cod")

# Lets generate random sample of the number of fish caught on each day
ndays=10
catches = round(runif(min=3, max=400, n=ndays))

# now for each fishing day (and its number of fish caught) - use sample to generate the fish
catches_res = list(catches) %>% pmap(sample, x=possible.fish, replace=TRUE)
# Look at the sample catches
catches_res[[1]]
```

```
## [1] "shark" "tuna" "salmon" "tuna" "tuna" "salmon"
## [7] "tuna" "tuna" "steelhead" "tuna" "salmon" "steelhead"
## [13] "shark" "tuna" "tuna" "steelhead" "salmon" "cod"
## [19] "steelhead" "cod" "tuna" "shark" "steelhead" "steelhead"
## [25] "steelhead" "cod" "salmon" "tuna" "tuna" "shark"
## [31] "steelhead" "cod" "salmon" "salmon" "steelhead" "shark"
## [37] "steelhead" "tuna" "salmon" "shark" "tuna" "cod"
## [43] "shark" "cod" "steelhead" "cod" "shark" "cod"
## [49] "shark" "cod" "salmon" "shark" "steelhead" "steelhead"
## [55] "salmon" "shark" "salmon" "salmon" "salmon" "tuna"
## [61] "shark" "tuna" "cod" "shark" "steelhead" "shark"
## [67] "salmon" "tuna" "cod" "cod" "shark" "shark"
## [73] "tuna" "cod" "shark" "salmon" "cod" "tuna"
## [79] "steelhead" "steelhead" "salmon" "steelhead" "cod" "shark"
## [85] "salmon" "steelhead" "salmon" "cod" "shark" "shark"
## [91] "shark" "shark" "tuna" "shark" "salmon" "shark"
## [97] "shark" "tuna" "shark" "steelhead" "shark" "shark"
```

```
## [103] "shark"      "salmon"      "cod"         "steelhead"   "shark"       "steelhead"
## [109] "tuna"       "cod"         "shark"       "steelhead"   "steelhead"   "steelhead"
## [115] "salmon"     "steelhead"   "cod"         "steelhead"   "salmon"      "tuna"
## [121] "salmon"     "shark"       "tuna"        "cod"         "cod"         "salmon"
## [127] "steelhead"  "tuna"        "tuna"        "shark"       "tuna"        "salmon"
## [133] "shark"      "shark"       "cod"         "steelhead"   "cod"         "steelhead"
## [139] "shark"      "cod"         "salmon"      "tuna"        "steelhead"   "tuna"
## [145] "steelhead"  "steelhead"   "tuna"        "cod"         "tuna"        "shark"
## [151] "salmon"     "tuna"        "steelhead"   "tuna"        "cod"         "tuna"
## [157] "salmon"     "steelhead"   "shark"       "cod"         "shark"       "tuna"
## [163] "steelhead"  "shark"       "cod"         "tuna"        "tuna"        "cod"
## [169] "shark"      "tuna"        "cod"         "cod"         "steelhead"   "steelhead"
## [175] "salmon"     "tuna"        "shark"       "cod"         "shark"       "shark"
## [181] "cod"        "cod"         "salmon"      "shark"       "tuna"
```

```
catches_res[[10]]
```

```
## [1] "steelhead" "tuna"      "cod"       "salmon"    "shark"     "steelhead"
## [7] "cod"       "steelhead" "steelhead" "cod"       "tuna"      "cod"
## [13] "cod"       "cod"       "salmon"    "steelhead" "tuna"      "salmon"
## [19] "tuna"      "tuna"      "salmon"    "shark"     "tuna"      "cod"
## [25] "salmon"    "steelhead" "shark"     "steelhead" "tuna"      "tuna"
## [31] "steelhead" "steelhead" "steelhead" "tuna"      "salmon"    "cod"
## [37] "steelhead" "steelhead" "salmon"    "steelhead" "tuna"      "shark"
## [43] "steelhead" "tuna"      "steelhead" "salmon"    "tuna"      "shark"
## [49] "cod"       "steelhead" "tuna"      "cod"       "tuna"      "steelhead"
## [55] "salmon"    "tuna"      "steelhead" "cod"       "steelhead" "cod"
## [61] "steelhead" "tuna"      "tuna"      "shark"     "cod"       "steelhead"
## [67] "tuna"      "cod"       "steelhead" "tuna"      "salmon"    "shark"
## [73] "tuna"
```

```
# ok so how do we get the summary (rarest fish, number of fish for all catches in catch_res)?
```