objects.R

Laptop

2025-02-23

# 1: Choose dataset  
judgeRatings <- USJudgeRatings  
  
# For this assignment to work, I chose a dataset that was structured, and where  
# each row represents an object or in this case a person  
  
  
  
  
# Dataset information  
# https://rdrr.io/r/datasets/USJudgeRatings.html  
  
# Description: Lawyers' ratings of state judges in the US Superior Court.  
  
# Format: A data frame containing 43 observations on 12 numeric variables.  
  
# Variables  
  
# 1: CONT Number of contacts of lawyer with judge.  
# 2: INTG Judicial integrity  
# 3: DMNR Demeanor  
# 4: DILG Diligence.  
# 5: CFMG Case flow managing.  
# 6: DECI Prompt decisions.  
# 7: PREP Preparation for trial.  
# 8: FAMI Familiarity with law.  
# 9: ORAL Sound oral rulings.  
# 10: WRIT Sound written rulings.  
# 11: PHYS Physical ability.  
# 12: RTEN Worthy of retention.   
  
  
  
  
# 2: Can generic object functions be applied to dataset?  
  
# Because of the structure of the data, generic object functions should be able  
# to be applied to the dataset which will be demonstrated in the next section.  
  
  
  
  
# 3: S3 objects vs S4 objects  
  
  
  
  
# Applying S3 classes to the dataset  
  
# Create judge1\_S3 object using first row from dataset  
judge1\_S3 <- list(  
 name = rownames(judgeRatings[1, ]),   
 contacts = judgeRatings$CONT[1],  
 integrity = judgeRatings$INTG[1],  
 demeanor = judgeRatings$DMNR[1],  
 dilligence = judgeRatings$DILG[1],  
 caseFlow = judgeRatings$CFMG[1],  
 promptDecisions = judgeRatings$DECI[1],  
 prep = judgeRatings$PREP[1],  
 familiarity = judgeRatings$FAMI[1],  
 oral = judgeRatings$ORAL[1],  
 written = judgeRatings$WRIT[1],  
 physical = judgeRatings$PHYS[1],  
 retention = judgeRatings$RTEN[1]  
 )  
  
# Assign the class "judge" to the judge1\_S3 object  
class(judge1\_S3) <- "judge"  
  
# Inspect the attributes of judge1\_S3  
attributes(judge1\_S3)

## $names  
## [1] "name" "contacts" "integrity" "demeanor"   
## [5] "dilligence" "caseFlow" "promptDecisions" "prep"   
## [9] "familiarity" "oral" "written" "physical"   
## [13] "retention"   
##   
## $class  
## [1] "judge"

# Get name  
judge1\_S3$name

## [1] "AARONSON,L.H."

# Applying S4 objects to the dataset  
  
# Use setClass function to create a judge class  
  
# !!!  
# The representation method has been deprecated in favor of the slots argument  
# The following use of setClass() does not work  
#   
# setClass("judge",  
# representation(  
# name = "character",  
# contacts = "numeric",  
# integrity = "numeric",  
# demeanor = "numeric",  
# dilligence = "numeric",  
# caseFlow = "numeric",  
# promptDecisions = "numeric",  
# prep = "numeric",  
# familiarity = "numeric",  
# oral = "numeric",  
# written = "numeric",  
# physical = "numeric",  
# retention = "numeric",  
# )  
# )  
  
# Create a judge class using the slots method instead of representation  
setClass("judge",  
 slots = list(  
 name = "character",  
 contacts = "numeric",  
 integrity = "numeric",  
 demeanor = "numeric",  
 dilligence = "numeric",  
 caseFlow = "numeric",  
 promptDecisions = "numeric",  
 prep = "numeric",  
 familiarity = "numeric",  
 oral = "numeric",  
 written = "numeric",  
 physical = "numeric",  
 retention = "numeric"  
 )  
)  
  
# Create judge1\_S4 object using new()  
judge1\_S4 <- new("judge",   
 name = rownames(judgeRatings[1, ]),   
 contacts = judgeRatings$CONT[1],  
 integrity = judgeRatings$INTG[1],  
 demeanor = judgeRatings$DMNR[1],  
 dilligence = judgeRatings$DILG[1],  
 caseFlow = judgeRatings$CFMG[1],  
 promptDecisions = judgeRatings$DECI[1],  
 prep = judgeRatings$PREP[1],  
 familiarity = judgeRatings$FAMI[1],  
 oral = judgeRatings$ORAL[1],  
 written = judgeRatings$WRIT[1],  
 physical = judgeRatings$PHYS[1],  
 retention = judgeRatings$RTEN[1]  
)  
  
# View object  
judge1\_S4

## An object of class "judge"  
## Slot "name":  
## [1] "AARONSON,L.H."  
##   
## Slot "contacts":  
## [1] 5.7  
##   
## Slot "integrity":  
## [1] 7.9  
##   
## Slot "demeanor":  
## [1] 7.7  
##   
## Slot "dilligence":  
## [1] 7.3  
##   
## Slot "caseFlow":  
## [1] 7.1  
##   
## Slot "promptDecisions":  
## [1] 7.4  
##   
## Slot "prep":  
## [1] 7.1  
##   
## Slot "familiarity":  
## [1] 7.1  
##   
## Slot "oral":  
## [1] 7.1  
##   
## Slot "written":  
## [1] 7  
##   
## Slot "physical":  
## [1] 8.3  
##   
## Slot "retention":  
## [1] 7.8

# Get name  
judge1\_S4@name

## [1] "AARONSON,L.H."

# Blog questions  
  
  
  
  
# 1: How do you tell what OO system (S3 vs. S4) an object is associated with?  
  
# We can determine if an object is S3 or S4 by using the isS4() function  
isS4(judge1\_S3)

## [1] FALSE

isS4(judge1\_S4)

## [1] TRUE

# 2: How do you determine the base type (like integer or list) of an object?  
  
# The typeof() method can be used to determine the base type of an object  
typeof(judge1\_S3)

## [1] "list"

typeof(judge1\_S4)

## [1] "S4"

# 3: What is a generic function?  
  
# As per the textbook: "R is polyporphic, which means that a single function  
# can be applied to different types of inputs, which the function processes in  
# the appropriate way. Such a function is called a generic function."  
# (Matloff, pg. xxi). The process that the function uses is known as the  
# "method". With S4 classes we can define our own method that is specific to  
# a certain class.  
  
  
  
  
# 4: What are the main differences between S3 and S4?  
  
# S3 – older, simpler, more dynamic, less structured version, single dispatch  
# S4 – newer, more structured, more rigorous, multi-dispatch  
  
  
  
  
# 5: In your GitHub, create two examples of S3 and S4.  
  
# Done