

R Notebook

```
library(dewey)
library(data.table)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble   3.1.6    v dplyr    1.0.8
## v tidyr    1.2.0    v stringr  1.4.0
## v readr    2.1.2    v forcats  0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::between() masks data.table::between()
## x dplyr::filter()  masks stats::filter()
## x dplyr::first()   masks data.table::first()
## x dplyr::lag()     masks stats::lag()
## x dplyr::last()    masks data.table::last()
## x purrr::transpose() masks data.table::transpose()

library(sparklyr)

##
## Attaching package: 'sparklyr'

## The following object is masked from 'package:purrr':
## 
##     invoke

## The following object is masked from 'package:stats':
## 
##     filter

Sys.setenv(JAVA_HOME = "/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home")

sc <- spark_connect(master = "local")

## Warning in sprintf(versions$pattern, version$spark, version$hadoop): 2 arguments
## not used by format 'spark-3.2.1-bin-hadoop3.2'
```

```

# if(file.exists("source")) unlink("source", TRUE)
# if(file.exists("source-out")) unlink("source-out", TRUE)
#
# stream_generate_test(iterations = 1)
# list.files("source")
#
# read_folder <- stream_read_csv(sc, "source")

# spark_disconnect(sc)

# load the wait time files
files <-
  data.table("filePath" = grep("*\\*.csv", list.files("data/"), value = TRUE)) %>%
  .[, "rideName" := sub("(\\_old)?\\*.csv", "", filePath)] %>%
  .[, rideName := ifelse(rideName == "7_dwarfs_train", "dwarfs_train", rideName)]

round_time = function(x, precision, method = round) {
  if ("POSIXct" %in% class(x) == FALSE)
    stop("x must be POSIXct")

  tz = attributes(x)$tzone
  secs_rounded = method(as.numeric(x) / precision) * precision
  as.POSIXct(secs_rounded, tz = tz, origin = "1970-01-01")
}

longerData <- function(x) {
  rbindlist(
    list(x %>%
        .[, .(RIDENAME, date, datetime, SACTMIN)] %>%
        .[, `:=`(`TYPE` = "SACTMIN", WAITTIME = SACTMIN, SACTMIN = NULL)],
      x %>%
        .[, .(RIDENAME, date, datetime, SPOSTMIN)] %>%
        .[, `:=`(`TYPE` = "SPOSTMIN", WAITTIME = SPOSTMIN, SPOSTMIN = NULL)])
  )
}

dt <- unique(rbindlist(apply(files, 1, function(x) {
  fread(paste0("data/", x["filePath"])) %>%
  .[!is.na(SPOSTMIN) & SPOSTMIN >= 0 | !is.na(SACTMIN) & SACTMIN >= 0] %>%
  .[, RIDENAME := x["rideName"]]
}), use.names = TRUE)) %>%
longerData(.) %>%
.[, `:=`(`DATE` = as.ordered(format(as.Date(date, format = "%m/%d/%Y"),
                                     format = "%m-%d")),
          date = NULL,
          DATETIME = round_time(datetime, 60*5, floor),
          datetime = NULL)] %>%
.[, `:=`(`MONTH` = month(DATETIME),
         DAY = mday(DATETIME),
         TIME = as.ITime(DATETIME),
         DATETIME = NULL)] %>%
.[, WAITTIMEmean := mean(log(WAITTIME), na.rm = TRUE),
   by = .(RIDENAME, TYPE, DATE, MONTH, DAY, TIME)] %>%

```

```

. [ , WAITTIME := NULL] %>%
. [order(RIDENAME, DATE, TYPE, TIME)] %>%
unique() %>%
dcast(., DATE + MONTH + DAY + TIME ~ RIDENAME + TYPE,
      value.var = "WAITTIMEmean")

dt[dt == "NaN" | dt == "-Inf"] <- NA

print(dt)

##           DATE MONTH DAY      TIME alien_saucers_SACTMIN alien_saucers_SPOSTMIN
## 1: 01-01     1    1 06:00:00                  NA                  NA
## 2: 01-01     1    1 06:05:00                  NA                  NA
## 3: 01-01     1    1 06:20:00                  NA                  NA
## 4: 01-01     1    1 06:25:00                  NA                  NA
## 5: 01-01     1    1 06:35:00                  NA                  NA
##   ---
## 79309: 12-31    12  31 23:35:00                  NA 2.928967
## 79310: 12-31    12  31 23:40:00                  NA 2.505318
## 79311: 12-31    12  31 23:45:00                  NA          NA
## 79312: 12-31    12  31 23:50:00                  NA 2.505318
## 79313: 12-31    12  31 23:55:00                  NA 2.505318
##           dinosaur_SACTMIN dinosaur_SPOSTMIN dwarfs_train_SACTMIN
## 1:                 NA                  NA                  NA
## 2:                 NA                  NA                  NA
## 3:                 NA                  NA                  NA
## 4:                 NA                  NA                  NA
## 5:                 NA                  NA                  NA
##   ---
## 79309:             NA 2.302585                  NA
## 79310:             NA          NA                  NA
## 79311:             NA 1.609438                  NA
## 79312:             NA 1.609438                  NA
## 79313:             NA 1.609438                  NA
##           dwarfs_train_SPOSTMIN expedition_everest_SACTMIN
## 1:                 NA                  NA
## 2:                 NA                  NA
## 3:                 NA                  NA
## 4:                 NA                  NA
## 5:                 NA                  NA
##   ---
## 79309:        4.204803                  NA
## 79310:        3.623511                  NA
## 79311:        3.516270                  NA
## 79312:        4.787492                  NA
## 79313:        3.747771                  NA
##           expedition_everest_SPOSTMIN flight_of_passage_SACTMIN
## 1:                 NA                  NA
## 2:                 NA                  NA
## 3:                 NA                  NA
## 4:                 NA                  NA
## 5:                 NA                  NA
##   ---

```

```

## 79309:           1.609438      NA
## 79310:             NA          NA
## 79311:           1.609438      NA
## 79312:           1.609438      NA
## 79313:           1.609438      NA
##     flight_of_passage_SPOSTMIN kilimanjaro_safaris_SACTMIN
## 1:                 NA          NA
## 2:                 NA          NA
## 3:                 NA          NA
## 4:                 NA          NA
## 5:                 NA          NA
##   ---
## 79309:           3.912023      NA
## 79310:             NA          NA
## 79311:           3.806662      NA
## 79312:           3.806662      NA
## 79313:           3.806662      NA
##     kilimanjaro_safaris_SPOSTMIN navi_river_SACTMIN navi_river_SPOSTMIN
## 1:                 NA          NA          NA
## 2:                 NA          NA          NA
## 3:                 NA          NA          NA
## 4:                 NA          NA          NA
## 5:                 NA          NA          NA
##   ---
## 79309:             NA          NA       1.609438
## 79310:             NA          NA          NA
## 79311:             NA          NA       1.609438
## 79312:             NA          NA       1.609438
## 79313:             NA          NA       1.609438
##     pirates_of_caribbean_SACTMIN pirates_of_caribbean_SPOSTMIN
## 1:                 NA          2.302585
## 2:                 NA          2.302585
## 3:                 NA          2.302585
## 4:                 NA          2.302585
## 5:                 NA          2.302585
##   ---
## 79309:             NA          2.129298
## 79310:             NA          2.104543
## 79311:             NA          2.071536
## 79312:             NA          NA
## 79313:             NA          2.163956
##     rock_n_rollercoaster_SACTMIN rock_n_rollercoaster_SPOSTMIN
## 1:                 NA          3.688879
## 2:                 NA          3.688879
## 3:                 NA          3.688879
## 4:                 NA          3.688879
## 5:                 NA          3.688879
##   ---
## 79309:             NA          4.343304
## 79310:             NA          3.792406
## 79311:             NA          3.891612
## 79312:             NA          3.759549
## 79313:             NA          3.401197
##     slinky_dog_SACTMIN slinky_dog_SPOSTMIN soarin_SACTMIN soarin_SPOSTMIN

```

```

##   1:      NA      NA      NA 3.912023
##   2:      NA      NA      NA 3.912023
##   3:      NA      NA      NA 3.912023
##   4:      NA      NA      NA 3.912023
##   5:      NA      NA      NA      NA
##   ---
## 79309:      NA 3.912023      NA 3.502026
## 79310:      NA 3.912023      NA 3.512027
## 79311:      NA      NA      NA 3.310037
## 79312:      NA 3.912023      NA 3.311365
## 79313:      NA 3.806662      NA 3.512027
##      spaceship_earth_SACTMIN spaceship_earth_SPOSTMIN splash_mountain_SACTMIN
##   1:          NA          1.609438      NA
##   2:          NA          1.609438      NA
##   3:          NA          1.609438      NA
##   4:          NA          1.609438      NA
##   5:          NA          NA      NA
##   ---
## 79309:          NA          2.607255      NA
## 79310:          NA          2.590455      NA
## 79311:          NA          2.158744      NA
## 79312:          NA          2.773557      NA
## 79313:          NA          2.590455      NA
##      splash_mountain_SPOSTMIN toy_story_mania_SACTMIN
##   1:          1.609438      NA
##   2:          1.609438      NA
##   3:          1.609438      NA
##   4:          1.609438      NA
##   5:          1.609438      NA
##   ---
## 79309:          2.057378      NA
## 79310:          1.983904      NA
## 79311:          1.993202      NA
## 79312:          1.609438      NA
## 79313:          1.886697      NA
##      toy_story_mania_SPOSTMIN
##   1:          NA
##   2:          NA
##   3:          NA
##   4:          NA
##   5:          NA
##   ---
## 79309:          3.329277
## 79310:          2.603401
## 79311:          NA
## 79312:          2.332031
## 79313:          1.975642

RIDENAME <- c("dwarfs_train", "alien_saucers", "dinosaur",
           "expedition_everest", "flight_of_passage", "kilimanjaro_safaris",
           "navi_river", "pirates_of_caribbean", "rock_n_rollercoaster",
           "slinky_dog", "soarin", "spaceship_earth", "splash_mountain",
           "toy_story_mania")
open_date <- as.Date(c("2014/05/28", "2018/06/30", "1998/04/22", "2006/04/09",

```

```

    "2017/05/27", "1998/04/22", "2017/05/27", "1973/12/17",
    "1999/07/29", "2018/06/30", "2005/05/15", "1982/10/01",
    "1992/07/17", "2008/05/31"))
splash <- c(FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, TRUE,
      FALSE, FALSE, TRUE, FALSE)
indoor <- c(FALSE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE,
            TRUE, TRUE, FALSE, TRUE)
age_hierarchy <- c(10, 13, 4, 8, 11, 5, 12, 1, 6, 14, 7, 2, 3, 9)
park <- c("mk", "hs", "ak", "ak", "ak", "ak", "mk", "hs", "hs", "ep",
        "ep", "mk", "hs")
dtMeta <- data.table(RIDENAME, open_date, age_hierarchy, splash, indoor, park)

```

I DID A LOG TRANSFORM

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
summary(dt)
```

	DATE	MONTH	DAY	TIME	
##	11-19	: 304	Min. : 1.000	Min. : 1.00	Min. : 0
##	11-03	: 274	1st Qu.: 4.000	1st Qu.: 8.00	1st Qu.:37200
##	10-04	: 254	Median : 7.000	Median :16.00	Median :53700
##	05-24	: 253	Mean : 6.625	Mean :15.77	Mean :52547
##	10-10	: 251	3rd Qu.:10.000	3rd Qu.:23.00	3rd Qu.:69900
##	12-30	: 251	Max. :12.000	Max. :31.00	Max. :86100
##	(Other)	:77726			
##	alien_saucers_SACTMIN	alien_saucers_SPOSTMIN	dinosaur_SACTMIN		
##	Min. :0.00	Min. :1.609	Min. :0.00		
##	1st Qu.:2.56	1st Qu.:2.804	1st Qu.:1.95		
##	Median :3.04	Median :3.314	Median :2.64		
##	Mean :2.87	Mean :3.130	Mean :2.55		
##	3rd Qu.:3.40	3rd Qu.:3.601	3rd Qu.:3.22		
##	Max. :4.37	Max. :5.247	Max. :4.56		
##	NA's :76440	NA's :25515	NA's :72989		
##	dinosaur_SPOSTMIN	dwarfs_train_SACTMIN	dwarfs_train_SPOSTMIN		
##	Min. :1.609	Min. :0.00	Min. :1.609		
##	1st Qu.:2.303	1st Qu.:3.09	1st Qu.:3.929		
##	Median :2.924	Median :3.47	Median :4.232		
##	Mean :2.843	Mean :3.42	Mean :4.155		
##	3rd Qu.:3.342	3rd Qu.:3.87	3rd Qu.:4.449		
##	Max. :5.165	Max. :5.38	Max. :5.340		
##	NA's :19585	NA's :71594	NA's :7300		
##	expedition EVEREST_SACTMIN	expedition EVEREST_SPOSTMIN			
##	Min. :0.00	Min. :1.609			
##	1st Qu.:1.95	1st Qu.:2.476			

```

## Median :2.48          Median :3.186
## Mean   :2.40          Mean   :3.000
## 3rd Qu.:3.00          3rd Qu.:3.569
## Max.   :4.92          Max.   :4.942
## NA's   :69843         NA's   :19404
## flight_of_passage_SACTMIN flight_of_passage_SPOSTMIN
## Min.   : 0.00          Min.   :1.609
## 1st Qu.: 3.56          1st Qu.:4.382
## Median : 3.97          Median :4.646
## Mean   : 3.93          Mean   :4.563
## 3rd Qu.: 4.38          3rd Qu.:4.847
## Max.   :10.78          Max.   :5.753
## NA's   :73850          NA's   :24142
## kilimanjaro_safaris_SACTMIN kilimanjaro_safaris_SPOSTMIN navi_river_SACTMIN
## Min.   :0.00            Min.   :1.609            Min.   :0.00
## 1st Qu.:2.08            1st Qu.:2.718            1st Qu.:2.64
## Median :2.83            Median :3.273            Median :3.22
## Mean   :2.70            Mean   :3.188            Mean   :3.07
## 3rd Qu.:3.37            3rd Qu.:3.680            3rd Qu.:3.69
## Max.   :5.33            Max.   :5.011            Max.   :5.31
## NA's   :72006           NA's   :22720           NA's   :75157
## navi_river_SPOSTMIN pirates_of_caribbean_SACTMIN pirates_of_caribbean_SPOSTMIN
## Min.   :1.609           Min.   :0.00           Min.   :1.609
## 1st Qu.:3.638           1st Qu.:1.79           1st Qu.:2.284
## Median :4.060           Median :2.64           Median :2.996
## Mean   :3.835           Mean   :2.43           Mean   :2.855
## 3rd Qu.:4.289           3rd Qu.:3.18           3rd Qu.:3.464
## Max.   :5.416           Max.   :4.62           Max.   :4.787
## NA's   :24433           NA's   :65567           NA's   :8910
## rock_n_rollercoaster_SACTMIN rock_n_rollercoaster_SPOSTMIN slinky_dog_SACTMIN
## Min.   :0.00             Min.   :1.609             Min.   :0.00
## 1st Qu.:2.71             1st Qu.:3.512             1st Qu.:3.22
## Median :3.22             Median :3.905             Median :3.61
## Mean   :3.11             Mean   :3.740             Mean   :3.54
## 3rd Qu.:3.64             3rd Qu.:4.152             3rd Qu.:4.01
## Max.   :5.27             Max.   :5.481             Max.   :5.28
## NA's   :71491             NA's   :13577             NA's   :73852
## slinky_dog_SPOSTMIN soarin_SACTMIN soarin_SPOSTMIN spaceship_earth_SACTMIN
## Min.   :1.609           Min.   :0.00           Min.   :1.609   Min.   :0.00
## 1st Qu.:4.007           1st Qu.:2.64           1st Qu.:3.364   1st Qu.:1.10
## Median :4.248           Median :3.14           Median :3.760   Median :1.95
## Mean   :4.152           Mean   :3.05           Mean   :3.642   Mean   :1.75
## 3rd Qu.:4.441           3rd Qu.:3.53           3rd Qu.:4.043   3rd Qu.:2.56
## Max.   :5.720           Max.   :5.26           Max.   :5.438   Max.   :4.39
## NA's   :25588           NA's   :70234           NA's   :16641   NA's   :73016
## spaceship_earth_SPOSTMIN splash_mountain_SACTMIN splash_mountain_SPOSTMIN
## Min.   :1.609           Min.   :0.00           Min.   :1.609
## 1st Qu.:1.869           1st Qu.:2.20           1st Qu.:2.438
## Median :2.390           Median :3.00           Median :3.414
## Mean   :2.457           Mean   :2.77           Mean   :3.204
## 3rd Qu.:2.993           3rd Qu.:3.53           3rd Qu.:3.970
## Max.   :4.787           Max.   :5.05           Max.   :5.043
## NA's   :17262           NA's   :71602           NA's   :10690
## toy_story_mania_SACTMIN toy_story_mania_SPOSTMIN

```

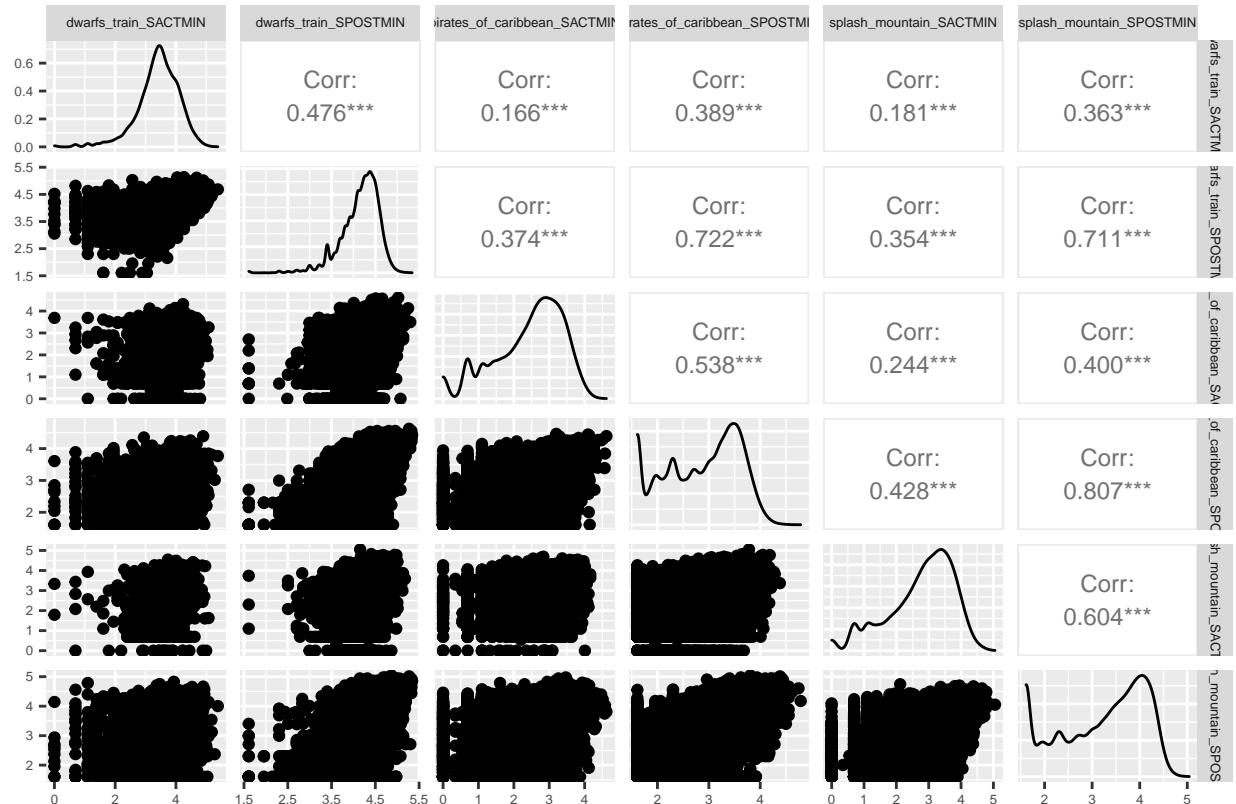
```

##  Min.   : 0.00      Min.   :1.609
##  1st Qu.: 2.71     1st Qu.:3.563
##  Median : 3.18     Median :3.948
##  Mean   : 3.09     Mean   :3.766
##  3rd Qu.: 3.58     3rd Qu.:4.155
##  Max.   :11.41     Max.   :5.704
##  NA's    :68541    NA's    :13206

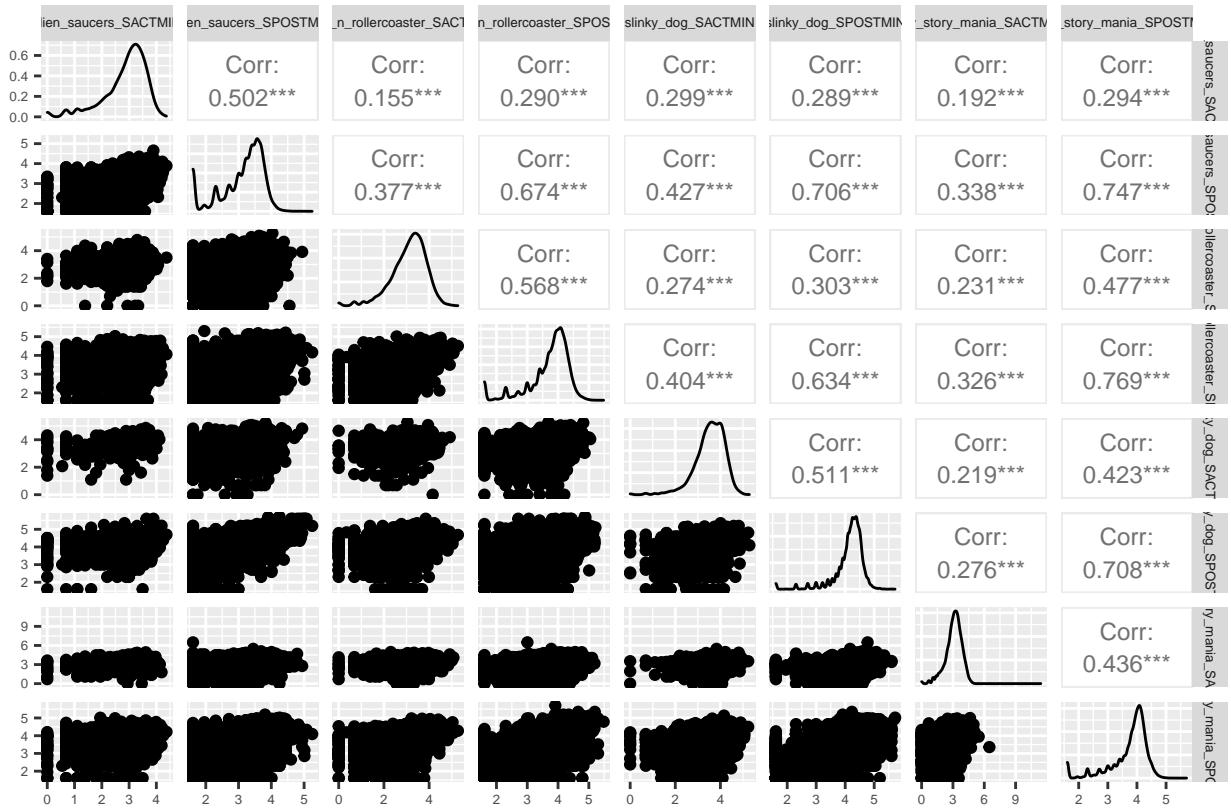
sapply(unique(dtMeta$park), function(x) {
  rides <- dtMeta$RIDENAME[dtMeta$park == x]
  cols <- (1:ncol(dt))[grep1(paste0(rides, collapse = "|"), colnames(dt))]
  print(ggpairs(dt[, ..cols], title = paste("Correlation plot for rides in", x),
                upper = list(continuous = wrap("cor", size = 3)),
                progress = FALSE) +
    theme(text = element_text(size = 6)))
})

```

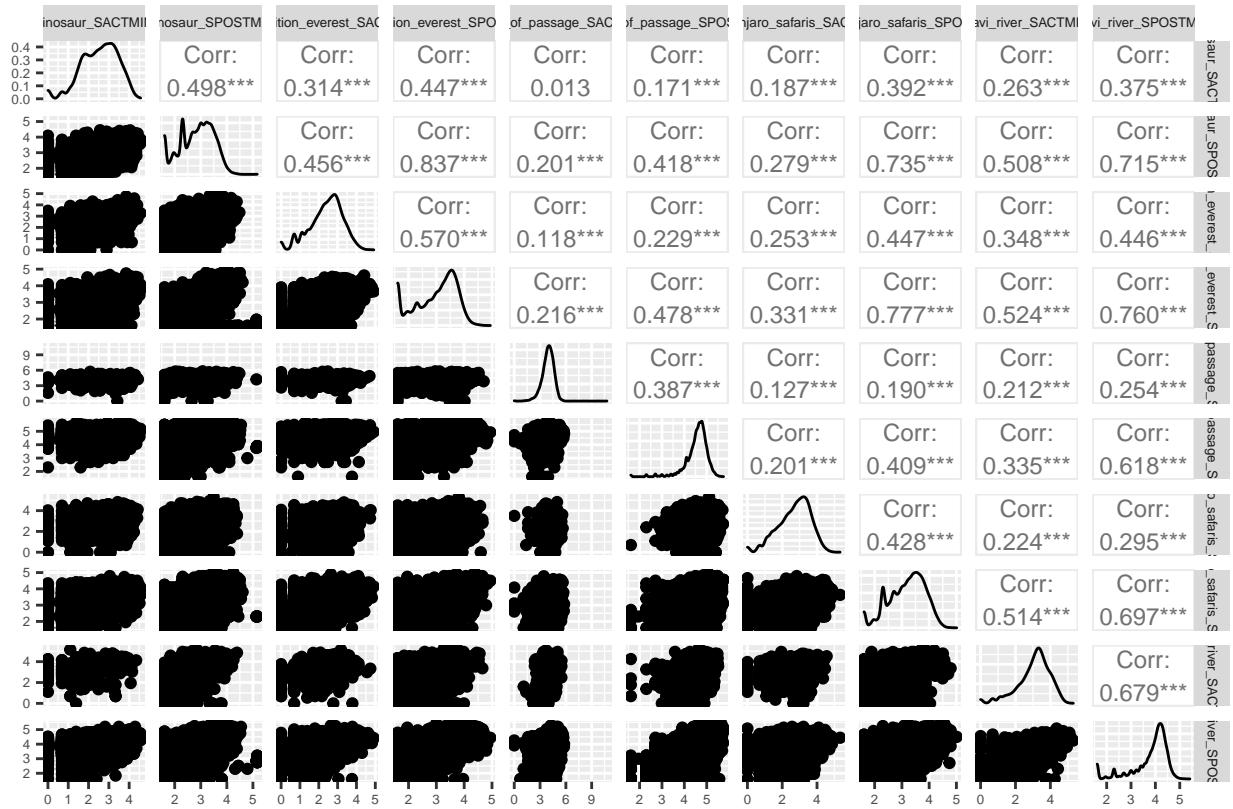
Correlation plot for rides in mk

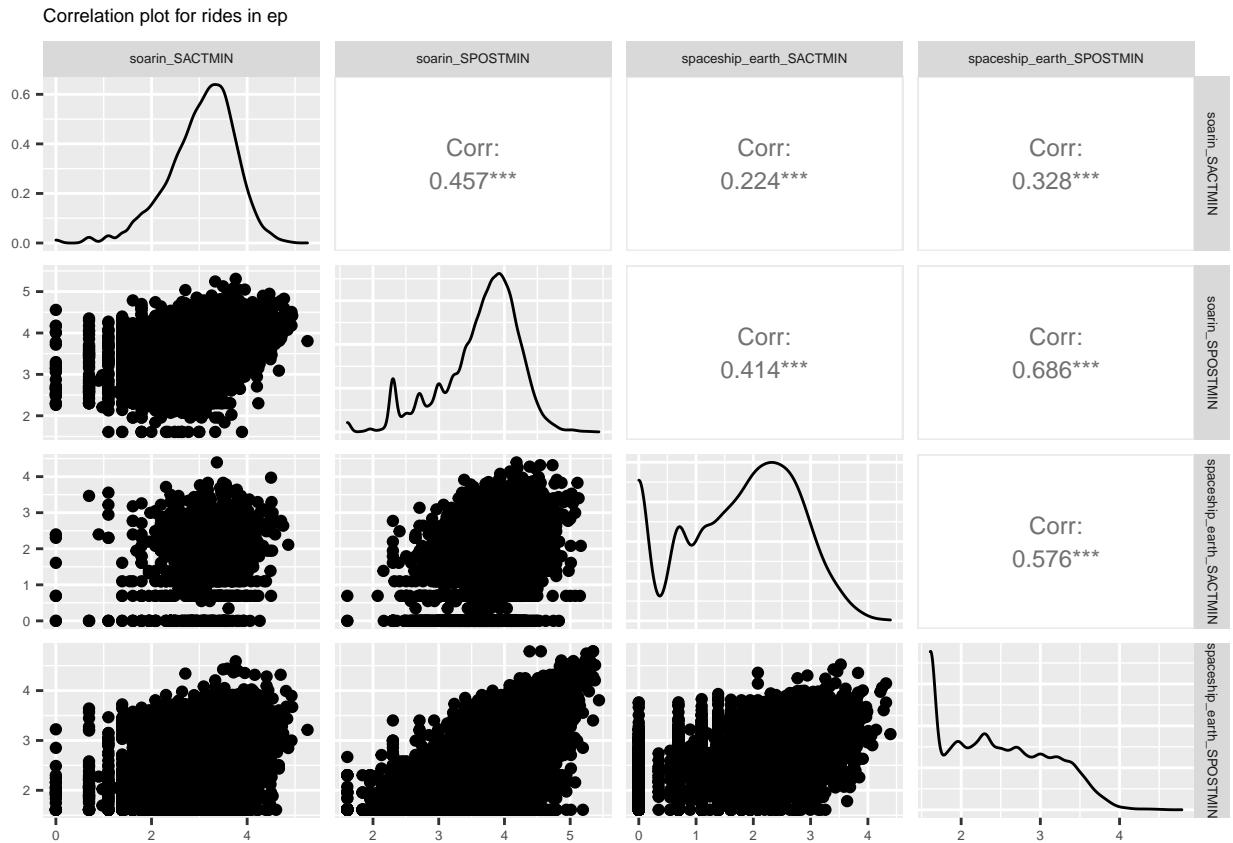


Correlation plot for rides in hs



Correlation plot for rides in ak





```

## $mk
## function (... , list = character() , package = NULL , lib.loc = NULL ,
##           verbose = getOption("verbose") , envir = .GlobalEnv , overwrite = TRUE)
## {
##   fileExt <- function(x) {
##     db <- grep("\\\\.[^.]\\\\.(gz|bz2|xz)$" , x)
##     ans <- sub("\\.*\\\\.", "" , x)
##     ans[db] <- sub("\\.*\\\\.([^.]+\\\\.)\\\\.(gz|bz2|xz)$" , "\\\\1\\\\2",
##                   x[db])
##     ans
##   }
##   my_read_table <- function(...) {
##     lcc <- Sys.getlocale("LC_COLLATE")
##     on.exit(Sys.setlocale("LC_COLLATE" , lcc))
##     Sys.setlocale("LC_COLLATE" , "C")
##     read.table(...)
##   }
##   stopifnot(is.character(list))
##   names <- c(as.character(substitute(list(...))[-1L]) , list)
##   if (!is.null(package)) {
##     if (!is.character(package))
##       stop("'package' must be a character vector or NULL")
##   }
##   paths <- find.package(package , lib.loc , verbose = verbose)
##   if (is.null(lib.loc))
##     paths <- c(path.package(package , TRUE) , if (!length(package)) getwd())
## }
```

```

##           paths)
## paths <- unique(normalizePath(paths[file.exists(paths)]))
## paths <- paths[dir.exists(file.path(paths, "data"))]
## dataExts <- tools::::make_file_exts("data")
## if (length(names) == 0L) {
##   db <- matrix(character(), nrow = 0L, ncol = 4L)
##   for (path in paths) {
##     entries <- NULL
##     packageName <- if (file_test("-f", file.path(path,
##       "DESCRIPTION")))
##       basename(path)
##     else "."
##     if (file_test("-f", INDEX <- file.path(path, "Meta",
##       "data.rds"))) {
##       entries <- readRDS(INDEX)
##     }
##     else {
##       dataDir <- file.path(path, "data")
##       entries <- tools::list_files_with_type(dataDir,
##         "data")
##       if (length(entries)) {
##         entries <- unique(tools::file_path_sans_ext(basename(entries)))
##         entries <- cbind(entries, "")
##       }
##       if (NROW(entries)) {
##         if (is.matrix(entries) && ncol(entries) == 2L)
##           db <- rbind(db, cbind(packageName, dirname(path),
##             entries))
##         else warning(gettextf("data index for package %s is invalid and will be ignored",
##           sQuote(packageName)), domain = NA, call. = FALSE)
##       }
##     }
##     colnames(db) <- c("Package", "LibPath", "Item", "Title")
##     footer <- if (missing(package))
##       paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRUE))")),
##         "\n", "to list the data sets in all *available* packages.")
##     else NULL
##     y <- list(title = "Data sets", header = NULL, results = db,
##       footer = footer)
##     class(y) <- "packageIQR"
##     return(y)
##   }
##   paths <- file.path(paths, "data")
##   for (name in names) {
##     found <- FALSE
##     for (p in paths) {
##       tmp_env <- if (overwrite)
##         envir
##       else new.env()
##       if (file_test("-f", file.path(p, "Rdata.rds"))) {
##         rds <- readRDS(file.path(p, "Rdata.rds"))
##         if (name %in% names(rds)) {
##           found <- TRUE
##         }
##       }
##     }
##     if (!found) {
##       db <- rbind(db, cbind(packageName, dirname(p),
##         entries))
##     }
##   }
## }

```

```

##           if (verbose)
##             message(sprintf("name=%s:\t found in Rdata.rds",
##                               name), domain = NA)
##             thispkg <- sub(".*/([~/]*)/data$", "\\\1", p)
##             thispkg <- sub("_.*$","", thispkg)
##             thispkg <- paste0("package:", thispkg)
##             objs <- rds[[name]]
##             lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                      filter = function(x) x %in% objs)
##             break
##           }
##         else if (verbose)
##           message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                           name, paste(names(rds), collapse = ",")),
##                           domain = NA)
##         }
##       if (file_test("-f", file.path(p, "Rdata.zip"))) {
##         warning("zipped data found for package ", sQuote(basename(dirname(p))),
##                 ".\nThat is defunct, so please re-install the package.",
##                 domain = NA)
##         if (file_test("-f", fp <- file.path(p, "filelist")))
##           files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##         else {
##           warninggettext("file 'filelist' is missing for directory %s",
##                         sQuote(p)), domain = NA)
##           next
##         }
##       }
##     else {
##       files <- list.files(p, full.names = TRUE)
##     }
##     files <- files[grep(name, files, fixed = TRUE)]
##     if (length(files) > 1L) {
##       o <- match(fileExt(files), dataExts, nomatch = 100L)
##       paths0 <- dirname(files)
##       paths0 <- factor(paths0, levels = unique(paths0))
##       files <- files[order(paths0, o)]
##     }
##     if (length(files)) {
##       for (file in files) {
##         if (verbose)
##           message("name=", name, ":\t file= ...", .Platform$file.sep,
##                  basename(file), ":\t", appendLF = FALSE,
##                  domain = NA)
##         ext <- fileExt(file)
##         if (basename(file) != paste0(name, ".", ext))
##           found <- FALSE
##         else {
##           found <- TRUE
##           zfile <- file
##           zipname <- file.path(dirname(file), "Rdata.zip")
##           if (file.exists(zipname)) {
##             Rdatadir <- tempfile("Rdata")
##             dir.create(Rdatadir, showWarnings = FALSE)

```

```

##          topic <- basename(file)
##          rc <- .External(C_unzip, zipfile, topic,
##                            Rdatadir, FALSE, TRUE, FALSE)
##          if (rc == 0L)
##              zfile <- file.path(Rdatadir, topic)
##          }
##          if (zfile != file)
##              on.exit(unlink(zfile))
##          switch(ext, R = , r = {
##              library("utils")
##              sys.source(zfile, chdir = TRUE, envir = tmp_env)
##          }, RData = , rdata = , rda = load(zfile,
##                                              envir = tmp_env), TXT = , txt = , tab = ,
##                                              tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                                              txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                                              header = TRUE, as.is = FALSE), envir = tmp_env),
##                                              CSV = , csv = , csv.gz = , csv.bz2 = ,
##                                              csv.xz = assign(name, my_read_table(zfile,
##                                              header = TRUE, sep = ";", as.is = FALSE),
##                                              envir = tmp_env), found <- FALSE)
##          }
##          if (found)
##              break
##      }
##      if (verbose)
##          message(if (!found)
##                  "*NOT* ", "found", domain = NA)
##      }
##      if (found)
##          break
##  }
##  if (!found) {
##      warning(gettextf("data set %s not found", sQuote(name)),
##              domain = NA)
##  }
##  else if (!overwrite) {
##      for (o in ls(envir = tmp_env, all.names = TRUE)) {
##          if (exists(o, envir = envir, inherits = FALSE))
##              warning(gettextf("an object named %s already exists and will not be overwritten",
##                           sQuote(o)))
##          else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                     envir = envir)
##      }
##      rm(tmp_env)
##  }
##  invisible(names)
## }

## <bytecode: 0x7fbef9af5d90>
## <environment: namespace:utils>
##
## $hs
## function (... , list = character(), package = NULL, lib.loc = NULL,
##           verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)

```

```

## {
##   fileExt <- function(x) {
##     db <- grep("\\\\.[^.]\\\\.+\\\\.(gz|bz2|xz)$", x)
##     ans <- sub("\\.*\\\\.", "", x)
##     ans[db] <- sub("\\.*\\\\.([^.]+\\\\.)\\\\.(gz|bz2|xz)$", "\\\\1\\\\2",
##                   x[db])
##     ans
##   }
##   my_read_table <- function(...) {
##     lcc <- Sys.getlocale("LC_COLLATE")
##     on.exit(Sys.setlocale("LC_COLLATE", lcc))
##     Sys.setlocale("LC_COLLATE", "C")
##     read.table(...)
##   }
##   stopifnot(is.character(list))
##   names <- c(as.character(substitute(list(...)))[-1L]), list)
##   if (!is.null(package)) {
##     if (!is.character(package))
##       stop("'package' must be a character vector or NULL")
##   }
##   paths <- find.package(package, lib.loc, verbose = verbose)
##   if (is.null(lib.loc))
##     paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##                paths)
##   paths <- unique(normalizePath(paths[file.exists(paths)]))
##   paths <- paths[dir.exists(file.path(paths, "data"))]
##   dataExts <- tools:::make_file_exts("data")
##   if (length(names) == 0L) {
##     db <- matrix(character(), nrow = 0L, ncol = 4L)
##     for (path in paths) {
##       entries <- NULL
##       packageName <- if (file_test("-f", file.path(path,
##           "DESCRIPTION")))
##         basename(path)
##       else "."
##       if (file_test("-f", INDEX <- file.path(path, "Meta",
##           "data.rds"))) {
##         entries <- readRDS(INDEX)
##       }
##       else {
##         dataDir <- file.path(path, "data")
##         entries <- tools:::list_files_with_type(dataDir,
##             "data")
##         if (length(entries)) {
##           entries <- unique(tools:::file_path_sans_ext(basename(entries)))
##           entries <- cbind(entries, "")
##         }
##       }
##       if (NROW(entries)) {
##         if (is.matrix(entries) && ncol(entries) == 2L)
##           db <- rbind(db, cbind(packageName, dirname(path),
##                               entries))
##         else warning(gettextf("data index for package %s is invalid and will be ignored",
##           sQuote(packageName)), domain = NA, call. = FALSE)
##       }
##     }
##   }
## }

```

```

##      }
##  }
## colnames(db) <- c("Package", "LibPath", "Item", "Title")
## footer <- if (missing(package))
##   paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRUE))")),
##          "\n", "to list the data sets in all *available* packages.")
## else NULL
## y <- list(title = "Data sets", header = NULL, results = db,
##           footer = footer)
## class(y) <- "packageIQR"
## return(y)
## }
## paths <- file.path(paths, "data")
## for (name in names) {
##   found <- FALSE
##   for (p in paths) {
##     tmp_env <- if (overwrite)
##                 envir
##               else new.env()
##     if (file_test("-f", file.path(p, "Rdata.rds")))
##       rds <- readRDS(file.path(p, "Rdata.rds"))
##       if (name %in% names(rds)) {
##         found <- TRUE
##         if (verbose)
##           message(sprintf("name=%s:\t found in Rdata.rds",
##                           name), domain = NA)
##         thispkg <- sub(".*/([^\t]*)/data$", "\\\1", p)
##         thispkg <- sub("_.*$","", thispkg)
##         thispkg <- paste0("package:", thispkg)
##         objs <- rds[[name]]
##         lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                  filter = function(x) x %in% objs)
##         break
##       }
##     else if (verbose)
##       message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                       name, paste(names(rds), collapse = ",")),
##               domain = NA)
##   }
##   if (file_test("-f", file.path(p, "Rdata.zip")))
##     warning("zipped data found for package ", sQuote(basename(dirname(p))),
##            ".\nThat is defunct, so please re-install the package.",
##            domain = NA)
##     if (file_test("-f", fp <- file.path(p, "filelist")))
##       files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##     else {
##       warninggettextf("file 'filelist' is missing for directory %s",
##                      sQuote(p)), domain = NA)
##       next
##     }
##   }
##   else {
##     files <- list.files(p, full.names = TRUE)
##   }

```

```

## files <- files[grep(name, files, fixed = TRUE)]
## if (length(files) > 1L) {
##   o <- match(fileExt(files), dataExts, nomatch = 100L)
##   paths0 <- dirname(files)
##   paths0 <- factor(paths0, levels = unique(paths0))
##   files <- files[order(paths0, o)]
##
## }
## if (length(files)) {
##   for (file in files) {
##     if (verbose)
##       message("name=", name, ":\t file= ...", .Platform$file.sep,
##              basename(file), ":\t", appendLF = FALSE,
##              domain = NA)
##     ext <- fileExt(file)
##     if (basename(file) != paste0(name, ".", ext))
##       found <- FALSE
##     else {
##       found <- TRUE
##       zfile <- file
##       zipname <- file.path(dirname(file), "Rdata.zip")
##       if (file.exists(zipname)) {
##         Rdatadir <- tempfile("Rdata")
##         dir.create(Rdatadir, showWarnings = FALSE)
##         topic <- basename(file)
##         rc <- .External(C_unzip, zipname, topic,
##                         Rdatadir, FALSE, TRUE, FALSE, FALSE)
##         if (rc == 0L)
##           zfile <- file.path(Rdatadir, topic)
##       }
##       if (zfile != file)
##         on.exit(unlink(zfile))
##       switch(ext, R = , r = {
##         library("utils")
##         sys.source(zfile, chdir = TRUE, envir = tmp_env)
##       }, RData = , rdata = , rda = load(zfile,
##                                            envir = tmp_env), TXT = , txt = , tab = ,
##                                            tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                                            txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                                                                           header = TRUE, as.is = FALSE), envir = tmp_env),
##                                            CSV = , csv = , csv.gz = , csv.bz2 = ,
##                                            csv.xz = assign(name, my_read_table(zfile,
##                                                                           header = TRUE, sep = ";", as.is = FALSE),
##                                                           envir = tmp_env), found <- FALSE)
##     }
##     if (found)
##       break
##   }
##   if (verbose)
##     message(if (!found)
##             "*NOT* ", "found", domain = NA)
##   }
##   if (found)
##     break
## }

```

```

##           if (!found) {
##             warning(gettextf("data set %s not found", sQuote(name)),
##                     domain = NA)
##           }
##         else if (!overwrite) {
##           for (o in ls(envir = tmp_env, all.names = TRUE)) {
##             if (exists(o, envir = envir, inherits = FALSE))
##               warning(gettextf("an object named %s already exists and will not be overwritten",
##                               sQuote(o)))
##             else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                        envir = envir)
##           }
##           rm(tmp_env)
##         }
##       invisible(names)
##     }
## <bytecode: 0x7fbef9af5d90>
## <environment: namespace:utils>
##
## $ak
## function (..., list = character(), package = NULL, lib.loc = NULL,
##           verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## {
##   fileExt <- function(x) {
##     db <- grep("\\\\.[^.]+\\\\.(gz|bz2|xz)$", x)
##     ans <- sub("\\.*\\\\.", "", x)
##     ans$db <- sub("\\.*\\\\.(^.+\\\\.)\\\\.(gz|bz2|xz)$", "\\\\1\\\\2",
##                  x$db)
##     ans
##   }
##   my_read_table <- function(...) {
##     lcc <- Sys.getlocale("LC_COLLATE")
##     on.exit(Sys.setlocale("LC_COLLATE", lcc))
##     Sys.setlocale("LC_COLLATE", "C")
##     read.table(...)
##   }
##   stopifnot(is.character(list))
##   names <- c(as.character(substitute(list(...))[-1L]), list)
##   if (!is.null(package)) {
##     if (!is.character(package))
##       stop("'package' must be a character vector or NULL")
##   }
##   paths <- find.package(package, lib.loc, verbose = verbose)
##   if (is.null(lib.loc))
##     paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##                paths)
##   paths <- unique(normalizePath(paths[file.exists(paths)]))
##   paths <- paths[dir.exists(file.path(paths, "data"))]
##   dataExts <- tools:::make_file_exts("data")
##   if (length(names) == 0L) {
##     db <- matrix(character(), nrow = 0L, ncol = 4L)
##     for (path in paths) {
##       entries <- NULL

```

```

## packageName <- if (file_test("-f", file.path(path,
##                               "DESCRIPTION")))
##   basename(path)
## else "."
## if (file_test("-f", INDEX <- file.path(path, "Meta",
##                                         "data.rds"))) {
##   entries <- readRDS(INDEX)
## }
## else {
##   dataDir <- file.path(path, "data")
##   entries <- tools::list_files_with_type(dataDir,
##                                         "data")
##   if (length(entries)) {
##     entries <- unique(tools::file_path_sans_ext(basename(entries)))
##     entries <- cbind(entries, "")
##   }
##   if (NROW(entries)) {
##     if (is.matrix(entries) && ncol(entries) == 2L)
##       db <- rbind(db, cbind(packageName, dirname(path),
##                             entries))
##     else warning(gettextf("data index for package %s is invalid and will be ignored",
##                           sQuote(packageName)), domain = NA, call. = FALSE)
##   }
##   colnames(db) <- c("Package", "LibPath", "Item", "Title")
##   footer <- if (missing(package))
##     paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRUE)"))),
##           "\n", "to list the data sets in all *available* packages.")
##   else NULL
##   y <- list(title = "Data sets", header = NULL, results = db,
##             footer = footer)
##   class(y) <- "packageIQR"
##   return(y)
## }
## paths <- file.path(paths, "data")
## for (name in names) {
##   found <- FALSE
##   for (p in paths) {
##     tmp_env <- if (overwrite)
##       envir
##     else new.env()
##     if (file_test("-f", file.path(p, "Rdata.rds")))
##       rds <- readRDS(file.path(p, "Rdata.rds"))
##     if (name %in% names(rds)) {
##       found <- TRUE
##       if (verbose)
##         message(sprintf("name=%s:\t found in Rdata.rds",
##                         name), domain = NA)
##       thispkg <- sub(".*/([^\/*])/data$", "\\\1", p)
##       thispkg <- sub("_.*$","", thispkg)
##       thispkg <- paste0("package:", thispkg)
##       objs <- rds[[name]]
##       lazyLoad(file.path(p, "Rdata"), envir = tmp_env,

```

```

##           filter = function(x) x %in% objs)
##           break
##       }
##   else if (verbose)
##     message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                     name, paste(names(rds), collapse = ",")),
##                     domain = NA)
##   }
##   if (file_test("-f", file.path(p, "Rdata.zip"))) {
##     warning("zipped data found for package ", sQuote(basename(dirname(p))),
##             ".\nThat is defunct, so please re-install the package.",
##             domain = NA)
##     if (file_test("-f", fp <- file.path(p, "filelist")))
##       files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##     else {
##       warning(gettextf("file 'filelist' is missing for directory %s",
##                       sQuote(p)), domain = NA)
##       next
##     }
##   }
##   else {
##     files <- list.files(p, full.names = TRUE)
##   }
##   files <- files[grep(name, files, fixed = TRUE)]
##   if (length(files) > 1L) {
##     o <- match(fileExt(files), dataExts, nomatch = 100L)
##     paths0 <- dirname(files)
##     paths0 <- factor(paths0, levels = unique(paths0))
##     files <- files[order(paths0, o)]
##   }
##   if (length(files)) {
##     for (file in files) {
##       if (verbose)
##         message("name=", name, ":\t file= ...",
##                 .Platform$file.sep,
##                 basename(file), ":\t", appendLF = FALSE,
##                 domain = NA)
##       ext <- fileExt(file)
##       if (basename(file) != paste0(name, ".", ext))
##         found <- FALSE
##       else {
##         found <- TRUE
##         zfile <- file
##         zipname <- file.path(dirname(file), "Rdata.zip")
##         if (file.exists(zipname)) {
##           Rdatadir <- tempfile("Rdata")
##           dir.create(Rdatadir, showWarnings = FALSE)
##           topic <- basename(file)
##           rc <- .External(C_unzip, zipname, topic,
##                           Rdatadir, FALSE, TRUE, FALSE, FALSE)
##           if (rc == 0L)
##             zfile <- file.path(Rdatadir, topic)
##           }
##           if (zfile != file)
##             on.exit(unlink(zfile))
##         }
##       }
##     }
##   }
## }

```

```

##           switch(ext, R = , r = {
##             library("utils")
##             sys.source(zfile, chdir = TRUE, envir = tmp_env)
##           }, RData = , rdata = , rda = load(zfile,
##             envir = tmp_env), TXT = , txt = , tab = ,
##             tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##             txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##               header = TRUE, as.is = FALSE), envir = tmp_env),
##             CSV = , csv = , csv.gz = , csv.bz2 = ,
##             csv.xz = assign(name, my_read_table(zfile,
##               header = TRUE, sep = ";", as.is = FALSE),
##               envir = tmp_env), found <- FALSE)
##           }
##           if (found)
##             break
##         }
##         if (verbose)
##           message(if (!found)
##             "*NOT* ", "found", domain = NA)
##         }
##         if (found)
##           break
##       }
##       if (!found) {
##         warning(gettextf("data set %s not found", sQuote(name)),
##             domain = NA)
##       }
##       else if (!overwrite) {
##         for (o in ls(envir = tmp_env, all.names = TRUE)) {
##           if (exists(o, envir = envir, inherits = FALSE))
##             warning(gettextf("an object named %s already exists and will not be overwritten",
##                 sQuote(o)))
##           else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##             envir = envir)
##         }
##         rm(tmp_env)
##       }
##       invisible(names)
##     }
##   <bytecode: 0x7fbbeb9af5d90>
##   <environment: namespace:utils>
##
## $ep
## function (..., list = character(), package = NULL, lib.loc = NULL,
##   verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## {
##   fileExt <- function(x) {
##     db <- grep("(\\.\\[^.]+\\.(gz|bz2|xz)$", x)
##     ans <- sub("\\.", "", x)
##     ans$db <- sub("\\.(\\[^.]+\\.(gz|bz2|xz)$", "\\1\\2",
##       x$db)
##     ans
##   }

```

```

## my_read_table <- function(...) {
##   lcc <- Sys.getlocale("LC_COLLATE")
##   on.exit(Sys.setlocale("LC_COLLATE", lcc))
##   Sys.setlocale("LC_COLLATE", "C")
##   read.table(...)
## }
## stopifnot(is.character(list))
## names <- c(as.character(substitute(list(...)))[-1L]), list)
## if (!is.null(package)) {
##   if (!is.character(package))
##     stop("'package' must be a character vector or NULL")
## }
## paths <- find.package(package, lib.loc, verbose = verbose)
## if (is.null(lib.loc))
##   paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##             paths)
## paths <- unique(normalizePath(paths[file.exists(paths)]))
## paths <- paths[dir.exists(file.path(paths, "data"))]
## dataExts <- tools:::make_file_exts("data")
## if (length(names) == 0L) {
##   db <- matrix(character(), nrow = 0L, ncol = 4L)
##   for (path in paths) {
##     entries <- NULL
##     packageName <- if (file_test("-f", file.path(path,
##                                                 "DESCRIPTION")))
##                   basename(path)
##                 else "."
##     if (file_test("-f", INDEX <- file.path(path, "Meta",
##                                             "data.rds"))) {
##       entries <- readRDS(INDEX)
##     }
##     else {
##       dataDir <- file.path(path, "data")
##       entries <- tools:::list_files_with_type(dataDir,
##                                                "data")
##       if (length(entries)) {
##         entries <- unique(tools:::file_path_sans_ext(basename(entries)))
##         entries <- cbind(entries, "")
##       }
##     }
##     if (NROW(entries)) {
##       if (is.matrix(entries) && ncol(entries) == 2L)
##         db <- rbind(db, cbind(packageName, dirname(path),
##                               entries))
##       else warning(gettextf("data index for package %s is invalid and will be ignored",
##                             sQuote(packageName)), domain = NA, call. = FALSE)
##     }
##   }
##   colnames(db) <- c("Package", "LibPath", "Item", "Title")
##   footer <- if (missing(package))
##             paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRUE))))),
##                   "\n", "to list the data sets in all *available* packages.")
##           else NULL
##   y <- list(title = "Data sets", header = NULL, results = db,

```

```

##           footer = footer)
##       class(y) <- "packageIQR"
##       return(y)
##   }
##   paths <- file.path(paths, "data")
##   for (name in names) {
##       found <- FALSE
##       for (p in paths) {
##           tmp_env <- if (overwrite)
##                       envir
##           else new.env()
##           if (file_test("-f", file.path(p, "Rdata.rds")))
##               rds <- readRDS(file.path(p, "Rdata.rds"))
##               if (name %in% names(rds)) {
##                   found <- TRUE
##                   if (verbose)
##                       message(sprintf("name=%s:\t found in Rdata.rds",
##                                       name), domain = NA)
##                   thispkg <- sub(".*/([/]*)/data$", "\\\1", p)
##                   thispkg <- sub("_.*$","", thispkg)
##                   thispkg <- paste0("package:", thispkg)
##                   objs <- rds[[name]]
##                   lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                            filter = function(x) x %in% objs)
##                   break
##               }
##               else if (verbose)
##                   message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                                   name, paste(names(rds), collapse = ",")),
##                           domain = NA)
##               }
##           if (file_test("-f", file.path(p, "Rdata.zip")))
##               warning("zipped data found for package ", sQuote(basename(dirname(p))),
##                      ".\nThat is defunct, so please re-install the package.", domain = NA)
##           if (file_test("-f", fp <- file.path(p, "filelist")))
##               files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##           else {
##               warning(gettextf("file 'filelist' is missing for directory %s",
##                               sQuote(p)), domain = NA)
##               next
##           }
##       }
##       else {
##           files <- list.files(p, full.names = TRUE)
##       }
##       files <- files[grep(name, files, fixed = TRUE)]
##       if (length(files) > 1L) {
##           o <- match(fileExt(files), dataExts, nomatch = 100L)
##           paths0 <- dirname(files)
##           paths0 <- factor(paths0, levels = unique(paths0))
##           files <- files[order(paths0, o)]
##       }
##       if (length(files)) {

```

```

##           for (file in files) {
##             if (verbose)
##               message("name=", name, ":\t file= ...", .Platform$file.sep,
##                      basename(file), ":\t", appendLF = FALSE,
##                      domain = NA)
##             ext <- fileExt(file)
##             if (basename(file) != paste0(name, ".", ext))
##               found <- FALSE
##             else {
##               found <- TRUE
##               zfile <- file
##               zipname <- file.path(dirname(file), "Rdata.zip")
##               if (file.exists(zipname)) {
##                 Rdatadir <- tempfile("Rdata")
##                 dir.create(Rdatadir, showWarnings = FALSE)
##                 topic <- basename(file)
##                 rc <- .External(C_unzip, zipname, topic,
##                                Rdatadir, FALSE, TRUE, FALSE, FALSE)
##                 if (rc == 0L)
##                   zfile <- file.path(Rdatadir, topic)
##               }
##               if (zfile != file)
##                 on.exit(unlink(zfile))
##               switch(ext, R = , r = {
##                 library("utils")
##                 sys.source(zfile, chdir = TRUE, envir = tmp_env)
##               }, RData = , rdata = , rda = load(zfile,
##                                                 envir = tmp_env), TXT = , txt = , tab = ,
##                                                 tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                                                 txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                                                 header = TRUE, as.is = FALSE), envir = tmp_env),
##                                                 CSV = , csv = , csv.gz = , csv.bz2 = ,
##                                                 csv.xz = assign(name, my_read_table(zfile,
##                                                 header = TRUE, sep = ";", as.is = FALSE),
##                                                 envir = tmp_env), found <- FALSE)
##               }
##               if (found)
##                 break
##             }
##             if (verbose)
##               message(if (!found)
##                       "*NOT* ", "found", domain = NA)
##           }
##           if (found)
##             break
##       }
##       if (!found) {
##         warning(gettextf("data set %s not found", sQuote(name)),
##                 domain = NA)
##       }
##       else if (!overwrite) {
##         for (o in ls(envir = tmp_env, all.names = TRUE)) {
##           if (exists(o, envir = envir, inherits = FALSE))
##             warning(gettextf("an object named %s already exists and will not be overwritten",

```

```

##           sQuote(o)))
##     else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                  envir = envir)
##   }
##   rm(tmp_env)
## }
## invisible(names)
## }
## <bytecode: 0x7fbeb9af5d90>
## <environment: namespace:utils>

## Loading required package: lattice

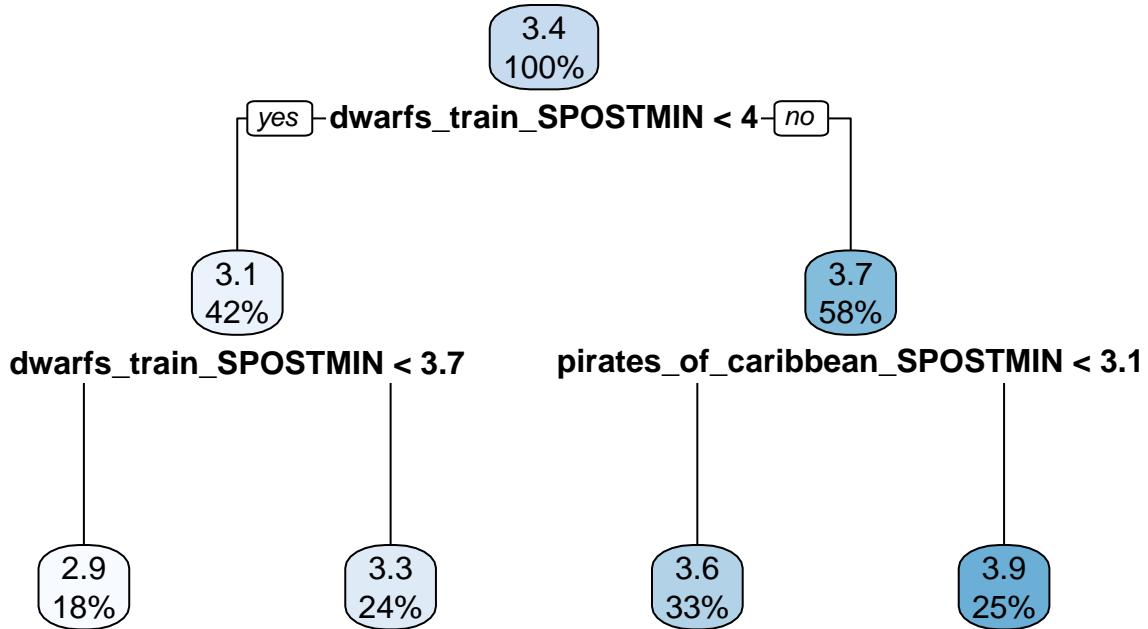
##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
## 
##     lift

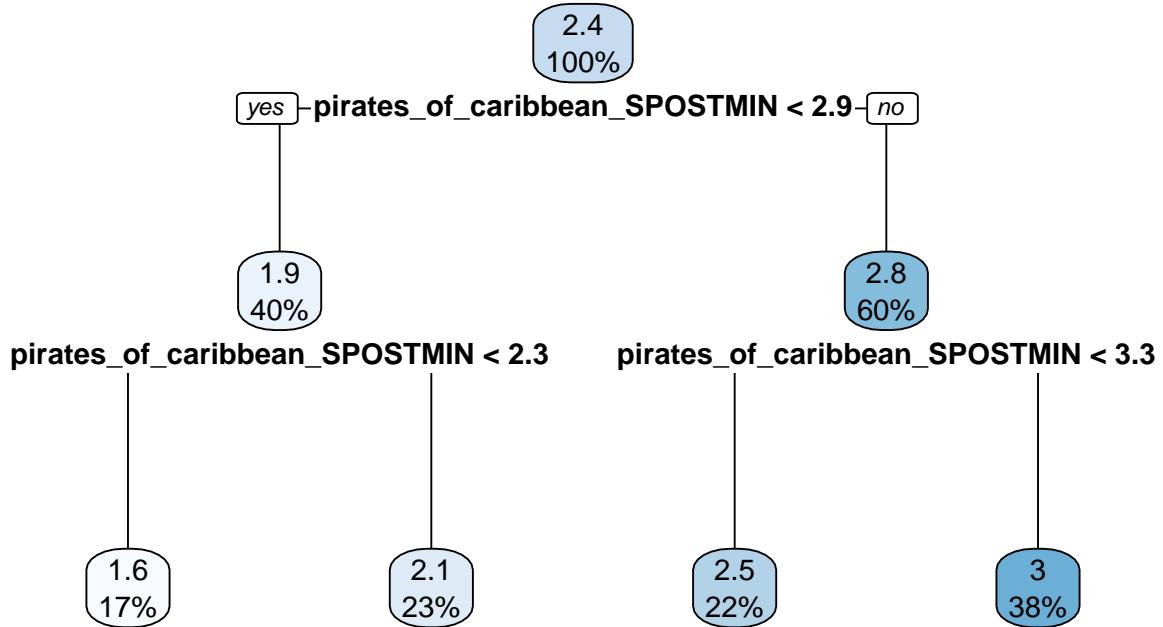
## Loading required package: rpart

```

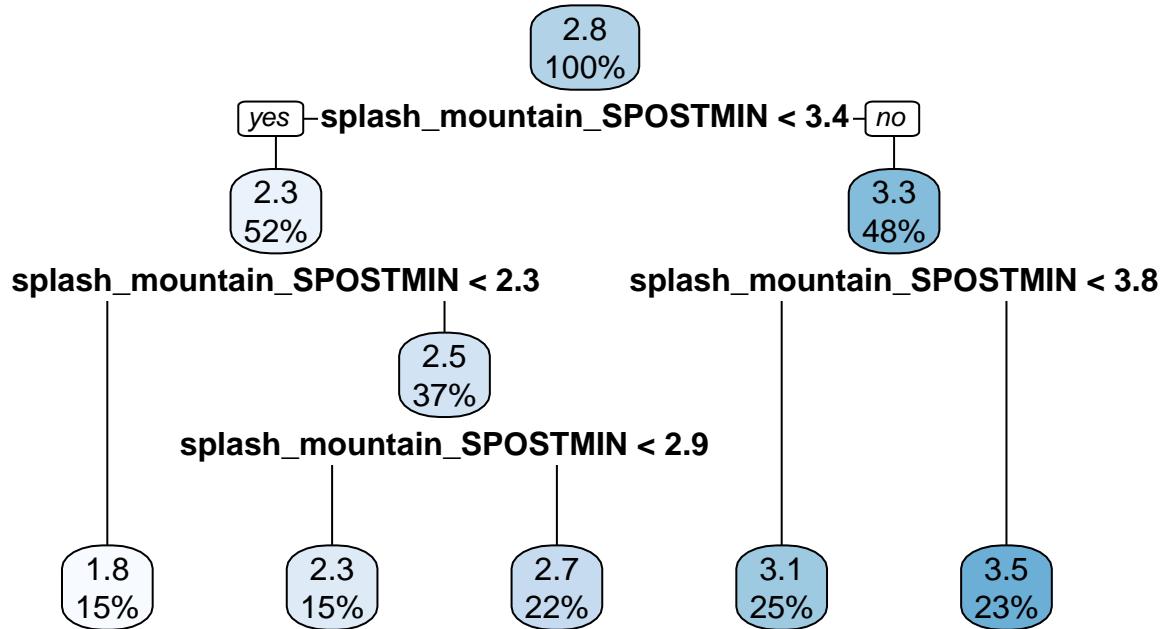
dwarfs_train_SACTMIN



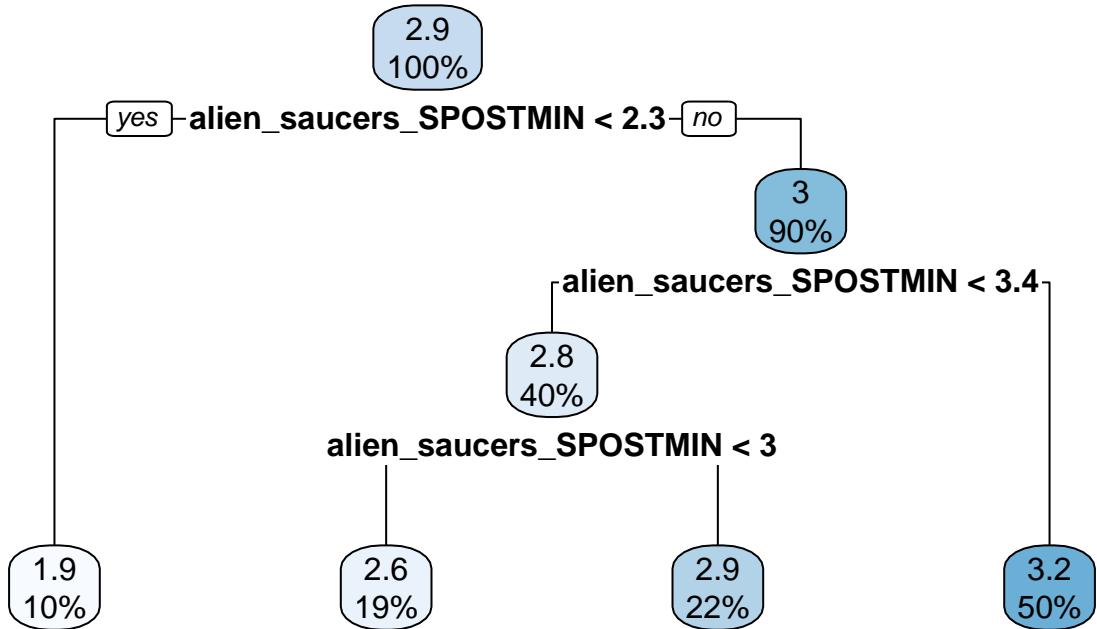
pirates_of_caribbean_SACTMIN



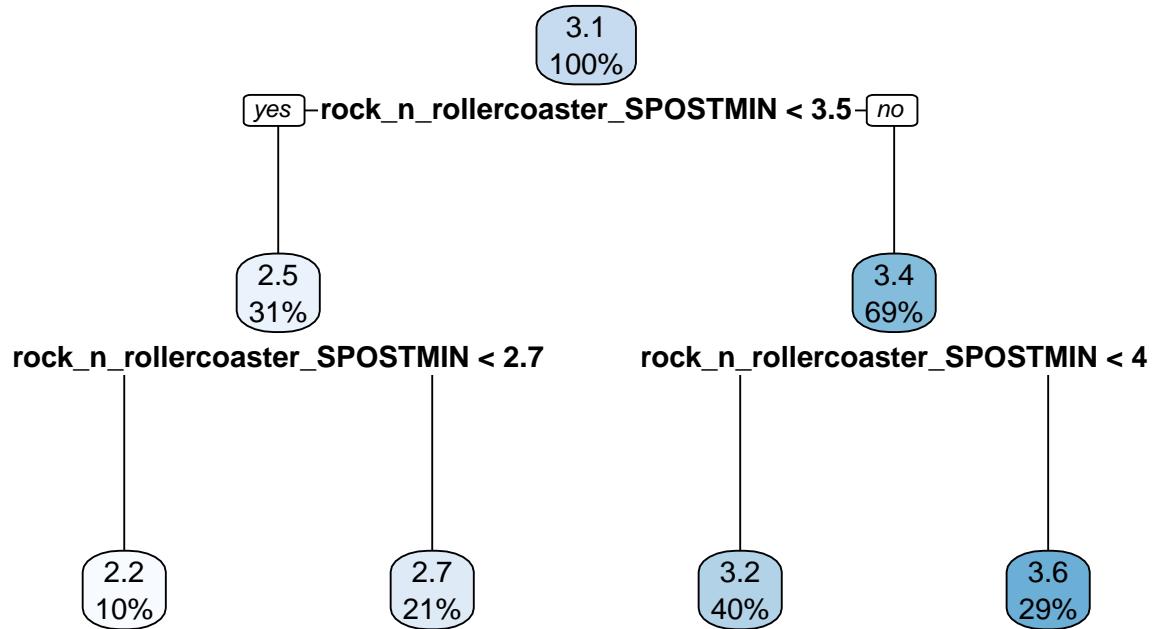
splash_mountain_SACTMIN



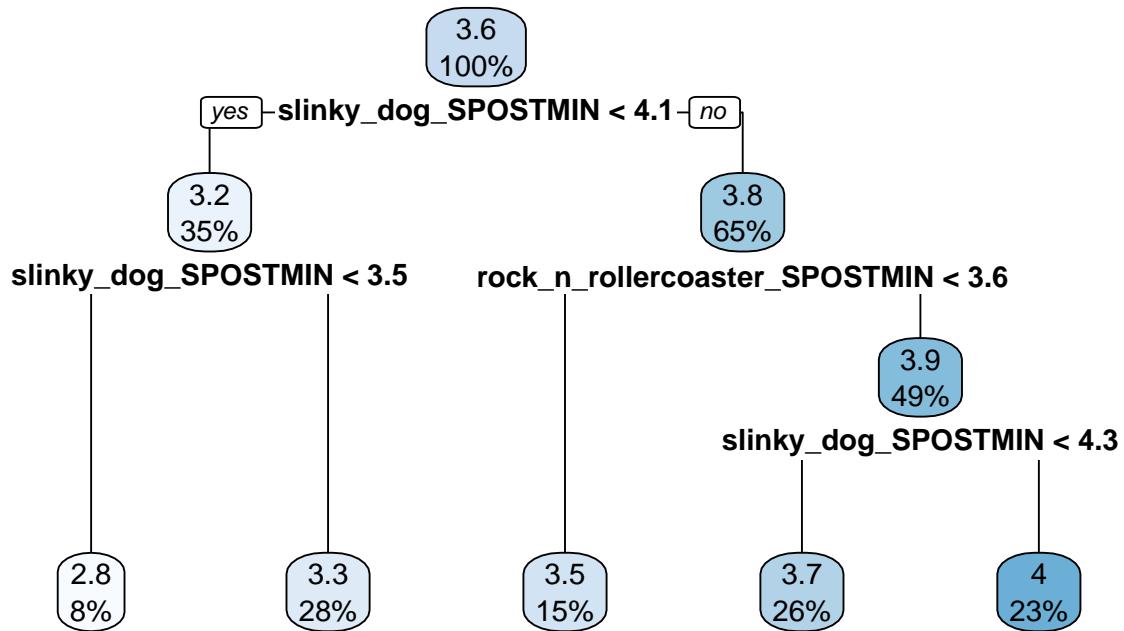
alien_saucers_SACTMIN



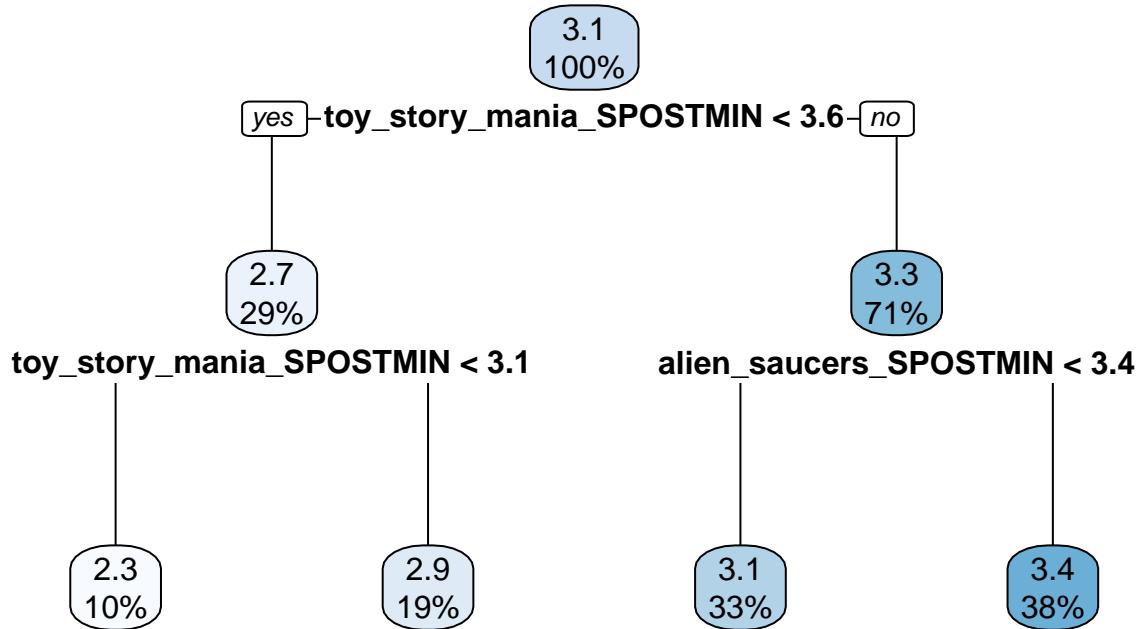
rock_n_rollercoaster_SACTMIN



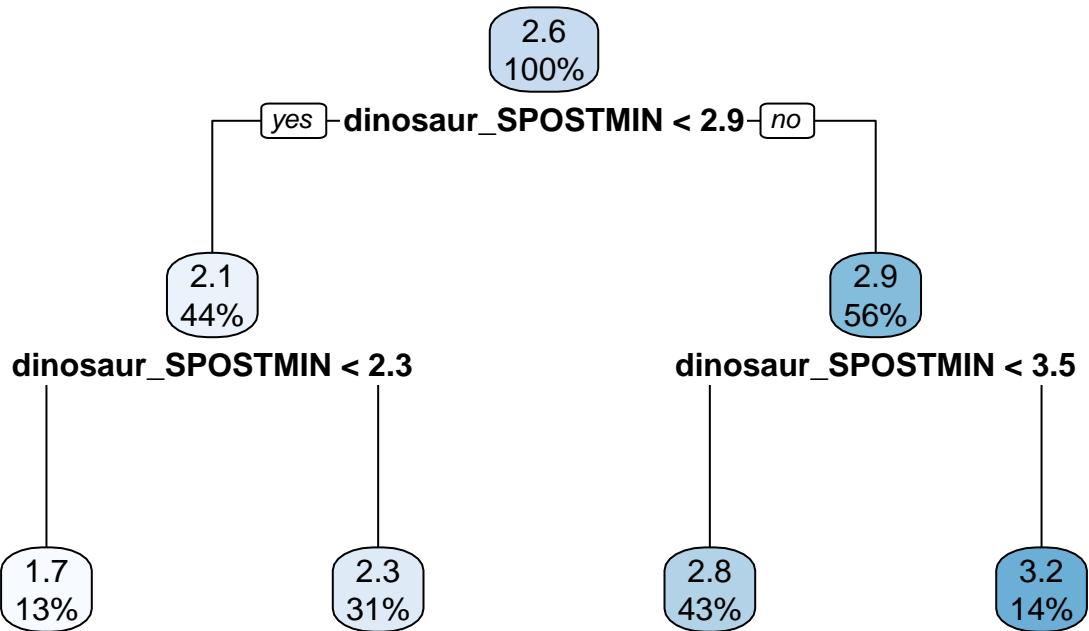
slinky_dog_SACTMIN



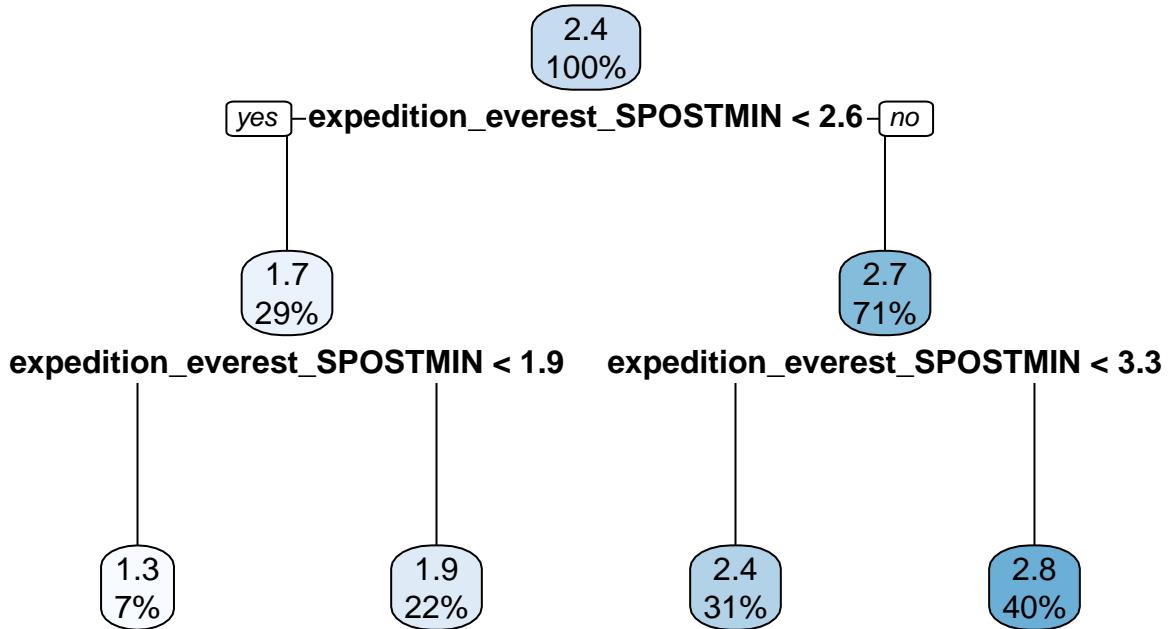
toy_story_mania_SACTMIN



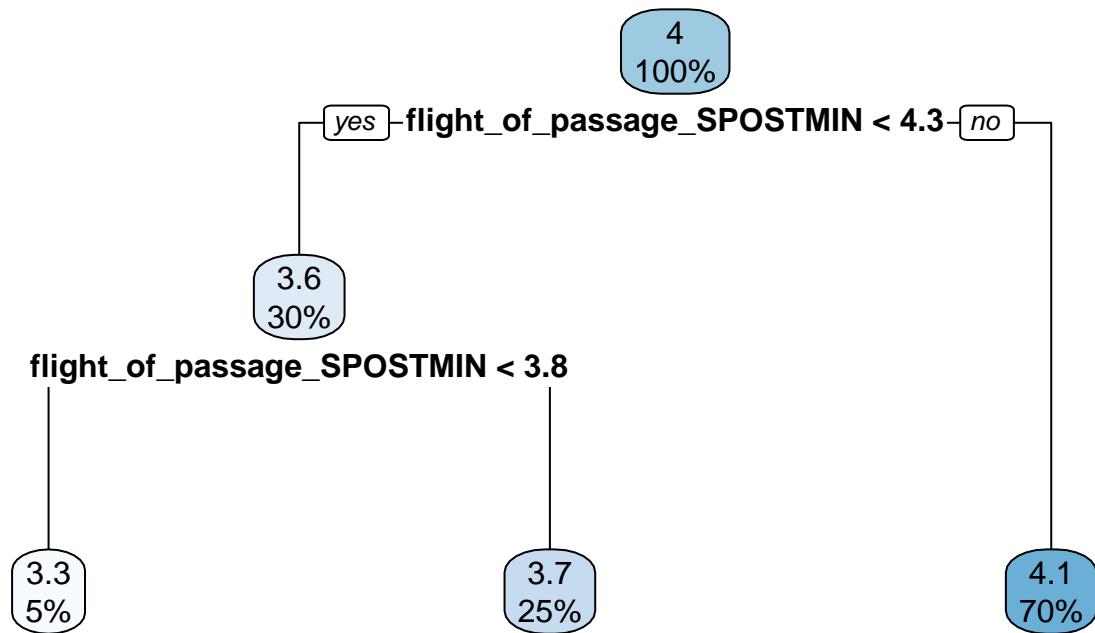
dinosaur_SACTMIN



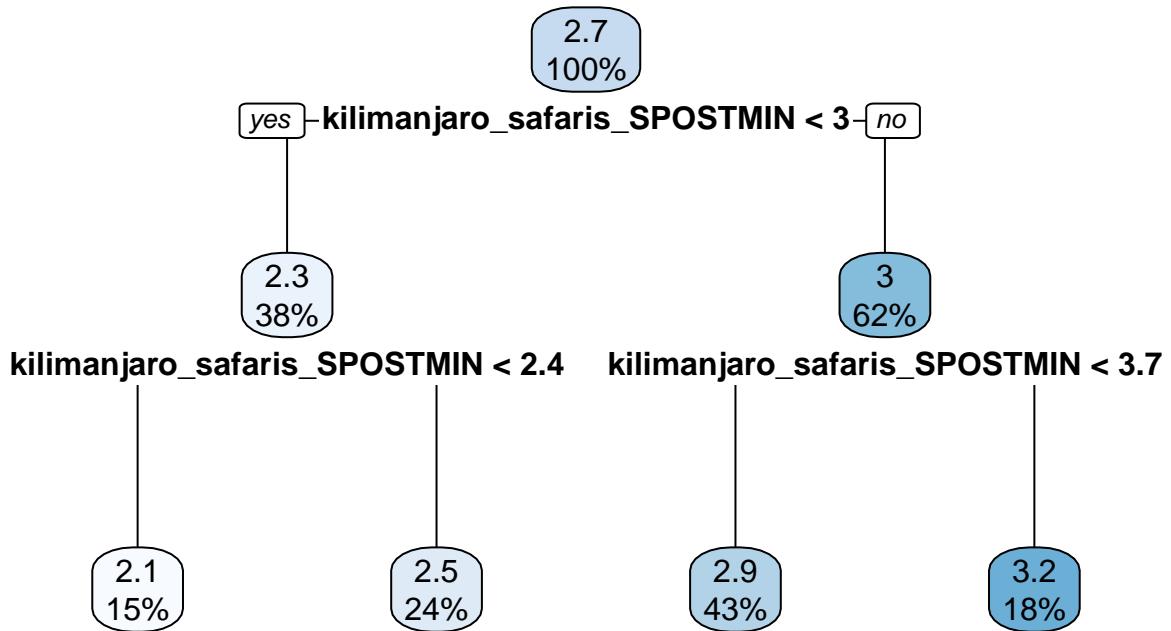
expedition_everest_SACTMIN



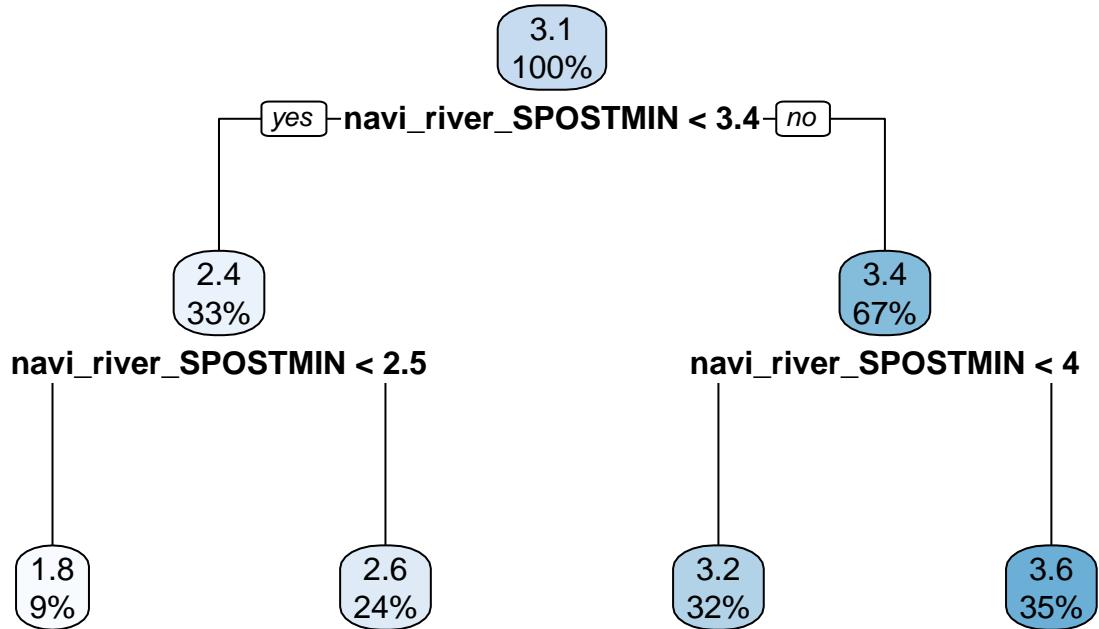
flight_of_passage_SACTMIN



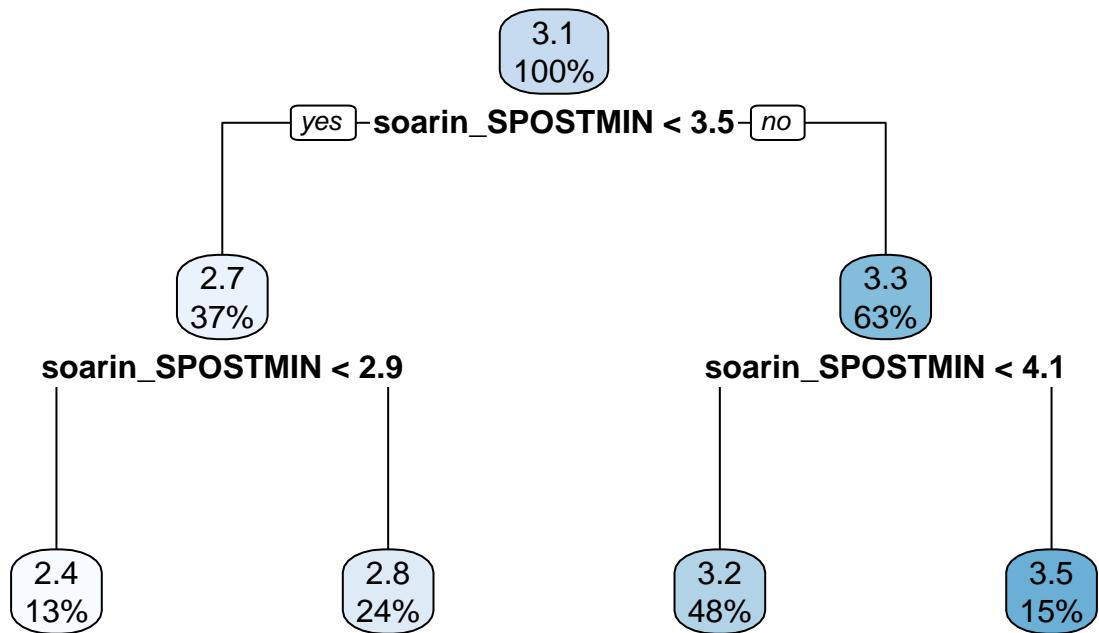
kilimanjaro_safaris_SACTMIN



navi_river_SACTMIN



soarin_SACTMIN



spaceship_earth_SACTMIN

