

# R Notebook

## Contents

Load Libraries	1
Load initial data	2
Summary Statistics	7
Load and flesh out metadata	7
Lag the data	8
Add in the weather data	9
Split the data	10
Create the trees	11

## Load Libraries

```
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(tidync)

library(tree)
library(randomForest)

## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##      combine

## The following object is masked from 'package:ggplot2':
##
##      margin

```

```

library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

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

```

```

library(rpart.plot)

## Loading required package: rpart

# 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 initial data

```

# load the wait time files
files <-
  data.table("filePath" = grep("*\\.csv", list.files("data/"), value = TRUE)) %>%
  .[, "rideName" := sub("(\\_old)?\\.csv", "", filePath)] %>%
  .[, rideName := toupper(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(as.Date(date, format = "%m/%d/%Y")),
  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(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
##      1: 2012-01-01      1  1 08:25:00                NA

```

##	2:	2012-01-01	1	1	08:45:00	NA
##	3:	2012-01-01	1	1	09:00:00	NA
##	4:	2012-01-01	1	1	09:10:00	NA
##	5:	2012-01-01	1	1	09:15:00	NA
##	---					
##	539491:	2021-12-28	12	28	22:40:00	NA
##	539492:	2021-12-28	12	28	22:45:00	NA
##	539493:	2021-12-28	12	28	22:50:00	NA
##	539494:	2021-12-28	12	28	22:55:00	NA
##	539495:	2021-12-28	12	28	23:00:00	NA
##	ALIEN_SAUCERS_SPOSTMIN DINOSAUR_SACTMIN DINOSAUR_SPOSTMIN					
##	1:		NA		NA	5
##	2:		NA		NA	10
##	3:		NA		NA	NA
##	4:		NA		NA	NA
##	5:		NA		NA	10
##	---					
##	539491:		NA		NA	NA
##	539492:		NA		NA	NA
##	539493:		NA		NA	NA
##	539494:		NA		NA	NA
##	539495:		NA		NA	NA
##	DWARFS_TRAIN_SACTMIN DWARFS_TRAIN_SPOSTMIN EXPEDITION_EVEREST_SACTMIN					
##	1:		NA		NA	NA
##	2:		NA		NA	NA
##	3:		NA		NA	NA
##	4:		NA		NA	NA
##	5:		NA		NA	NA
##	---					
##	539491:		NA		45	NA
##	539492:		NA		45	NA
##	539493:		10		45	NA
##	539494:		NA		35	NA
##	539495:		NA		NA	NA
##	EXPEDITION_EVEREST_SPOSTMIN FLIGHT_OF_PASSAGE_SACTMIN					
##	1:		NA		NA	
##	2:		NA		NA	
##	3:		5		NA	
##	4:		5		NA	
##	5:		NA		NA	
##	---					
##	539491:		NA		NA	
##	539492:		NA		NA	
##	539493:		NA		NA	
##	539494:		NA		NA	
##	539495:		NA		NA	
##	FLIGHT_OF_PASSAGE_SPOSTMIN KILIMANJARO_SAFARIS_SACTMIN					
##	1:		NA		NA	
##	2:		NA		NA	
##	3:		NA		NA	
##	4:		NA		NA	
##	5:		NA		NA	
##	---					
##	539491:		NA		NA	

##	539492:	NA	NA
##	539493:	NA	NA
##	539494:	NA	NA
##	539495:	NA	NA
##	KILIMANJARO_SAFARIS_SPOSTMIN NAVI_RIVER_SACTMIN NAVI_RIVER_SPOSTMIN		
##	1:	NA	NA
##	2:	NA	NA
##	3:	NA	NA
##	4:	NA	NA
##	5:	NA	NA
##	---		
##	539491:	NA	NA
##	539492:	NA	NA
##	539493:	NA	NA
##	539494:	NA	NA
##	539495:	NA	NA
##	PIRATES_OF_CARIBBEAN_SACTMIN PIRATES_OF_CARIBBEAN_SPOSTMIN		
##	1:	NA	NA
##	2:	NA	NA
##	3:	NA	NA
##	4:	NA	NA
##	5:	NA	NA
##	---		
##	539491:	NA	5
##	539492:	NA	5
##	539493:	NA	5
##	539494:	NA	NA
##	539495:	NA	5
##	ROCK_N_ROLLERCOASTER_SACTMIN ROCK_N_ROLLERCOASTER_SPOSTMIN		
##	1:	NA	NA
##	2:	NA	NA
##	3:	NA	NA
##	4:	NA	NA
##	5:	NA	NA
##	---		
##	539491:	NA	NA
##	539492:	NA	NA
##	539493:	NA	NA
##	539494:	NA	NA
##	539495:	NA	NA
##	SLINKY_DOG_SACTMIN SLINKY_DOG_SPOSTMIN SOARIN_SACTMIN SOARIN_SPOSTMIN		
##	1:	NA	NA
##	2:	NA	NA
##	3:	NA	NA
##	4:	NA	NA
##	5:	NA	NA
##	---		
##	539491:	NA	NA
##	539492:	NA	NA
##	539493:	NA	NA
##	539494:	NA	NA
##	539495:	NA	NA
##	SPACESHIP_EARTH_SACTMIN SPACESHIP_EARTH_SPOSTMIN		
##	1:	NA	NA

```

##      2:      NA      NA
##      3:      NA      NA
##      4:      NA      NA
##      5:      NA      NA
##      ---
## 539491:      NA      NA
## 539492:      NA      NA
## 539493:      NA      NA
## 539494:      NA      NA
## 539495:      NA      NA
##      SPLASH_MOUNTAIN_SACTMIN SPLASH_MOUNTAIN_SPOSTMIN
##      1:      NA      NA
##      2:      NA      NA
##      3:      NA      NA
##      4:      NA      NA
##      5:      NA      NA
##      ---
## 539491:      NA      15
## 539492:      NA      15
## 539493:      NA      15
## 539494:      NA      NA
## 539495:      NA      15
##      TOY_STORY_MANIA_SACTMIN TOY_STORY_MANIA_SPOSTMIN
##      1:      NA      NA
##      2:      NA      NA
##      3:      NA      NA
##      4:      NA      NA
##      5:      NA      NA
##      ---
## 539491:      NA      NA
## 539492:      NA      NA
## 539493:      NA      NA
## 539494:      NA      NA
## 539495:      NA      NA

```

```

RIDENAME <- toupper(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"))
OPENDATE <- 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,
  FALSE, FALSE, TRUE, FALSE)
INDOOR <- c(FALSE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE, TRUE, FALSE,
  TRUE, TRUE, FALSE, TRUE)
AGEHIERARCHY <- c(10, 13, 4, 8, 11, 5, 12, 1, 6, 14, 7, 2, 3, 9)
DURATION <- c(3, 2.5, 3.5, 4, 6, 20, 5, 7.5, 1.5, 3, 8, 16, 18, 6.5)
WAITPERHUNDRED <- c(5, 10, 3, 4, 4, 4, 5, 1.5, 2.5, 5, 3, 3, 3.5, 4.5)
PARK <- toupper(c("mk", "hs", "ak", "ak", "ak", "ak", "ak", "mk", "hs", "hs", "ep",
  "ep", "mk", "hs"))
dtMeta <- data.table(RIDENAME, OPENDATE, AGEHIERARCHY, SPLASH, INDOOR, PARK, DURATION, WAITPERHUNDRED)

```

## Summary Statistics

```
removeOutliers <- function(x) {  
  # get quantiles  
  q <- quantile(as.numeric(x), na.rm = TRUE)  
  # get lower limit  
  lower <- q["25%"]  
  # get upper limit  
  upper <- q["75%"]  
  # produce T/F if x is in range  
  q <- x >= lower & x <= upper  
  # if q is.na, change it to true, else keep q  
  q <- ifelse(is.na(q), TRUE, q)  
  q <- ifelse(!is.na(q) & !q, NA, q)  
  return(x[q])  
}  
  
cols <- grep(paste0(dtMeta$RIDENAME, collapse = "|"), colnames(dt), value = TRUE)  
dt <- dt[, (cols) := lapply(.SD, removeOutliers), .SDcols = cols]
```

```
library(GGally)  
  
summary(dt)  
  
sapply(unique(dtMeta$PARK), function(x) {  
  rides <- dtMeta$RIDENAME[dtMeta$PARK == x]  
  cols <- (1:ncol(dt))[grepl(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)))  
})
```

## Load and flesh out metadata

```
# metadata <- unique(rbindlist(list(fread("data/metadata/metadata.csv", na.strings = c("")),  
#                                fread("data/metadata/metadata_old.csv", na.strings = c("")),  
#                                fill = TRUE)) %>%  
# .[, DATE := as.ordered(format(as.Date(DATE, format = "%m/%d/%y")))] %>%  
# fwrite("newMetadata.csv")  
metadata <- fread("data/metadata/newMetadata.csv", na.strings = "") %>%  
  .[, DATE := as.ordered(format(as.Date(DATE, format = "%m/%d/%y")))]  
colnames(metadata) <- toupper(colnames(metadata))  
  
tmp <- grep("OPEN|CLOSE|PRDDT[1-2]{1}|SHWNT[1-2]{1}|FIRET[1-2]{1}|PRDNT[1-2]{1}|SUNSET",  
  colnames(metadata), value = TRUE)  
metadata <- metadata[!duplicated(metadata$DATE)]  
metadata[, (tmp) := lapply(.SD, as.ITime), .SDcols = tmp]  
# which(metadata$MKFIRET1 == metadata$MKFIRET2)
```

```

datetime <- data.table("DATE" = rep(metadata$DATE, each = 288),
  "TIME" = as.ITime(rep(seq(0*3600, 24*3600-1, by = 60*5))))
shows <- grep("PRDDT[1-2]{1}|SHWNT[1-2]{1}|FIRET[1-2]{1}|PRDNT[1-2]{1}",
  colnames(metadata), value = TRUE)
showType <- c("PRDDT", "SHWNT", "FIRET", "PRDNT")
tmp <- lapply(toupper(unique(dtMeta$PARK)), function(x) {
  type <- grep(paste0(x, showType, collapse = "|"), shows, value = TRUE)
  type <- unique(str_extract(type, paste0(showType, collapse = "|")))
  lapply(type, function(y) {
    cols <- c("DATE", grep(paste0(x, y), shows, value = TRUE))
    y <- melt(metadata[, ..cols],
      measure.vars = cols[-1],
      variable.name = paste0(x, y),
      value.name = paste0(x, y, "TIME"))
    y <- merge(datetime, y,
      by.x = c("DATE", "TIME"),
      by.y = c("DATE", grep("TIME", names(y), value = TRUE)),
      all.x = TRUE)
    y <- y[, !c("DATE", "TIME")]
    return(y)
  })
})
tmp <- rlist::list.cbind(unlist(tmp, recursive = FALSE))
cols <- unique(colnames(tmp))
tmp <- tmp[, ..cols]
tmp <- cbind(datetime, tmp)

cols <- !grepl(paste0(shows, collapse = "|"), colnames(metadata))
metadata <- merge(tmp, metadata[, ..cols], all.x = TRUE, by = "DATE")

metadata <- metadata[TIME >= as.ITime("06:00:00") | TIME <= as.ITime("03:00:00")]

tmp <- merge(dt, metadata, by = c("DATE", "TIME"), all.y = TRUE) %>%
  .[, `:=`(YEAR = NULL, MONTH = NULL, DAY = NULL)]

```

```

## Warning in rbindlist(l, use.names, fill, idcol): Column 1 of item 2 is an
## ordered factor but level 2080 ['2017-09-10'] is missing from the ordered levels
## from column 1 of item 1. Each set of ordered factor levels should be an ordered
## subset of the first longest. A regular factor will be created for this column.

```

```

dt <- tmp
rm(tmp)
rm(metadata)

```

## Lag the data

```

# lag the actual data
cols <- paste0(dtMeta$RIDENAME, c("_SACTMIN"))
dt <- dt[, (cols) := shift(.SD, type = "lead"), by = .(DATE), .SDcols = cols]

```



## Add in the weather data

```
nc <- tidync("data/w2012_2018.nc")
w2012_2018 <- nc %>%
  hyper_tibble() %>%
  data.table() %>%
  .[, `:=`(TIME = as.POSIXct(time * 60 * 60,
                             origin = "1900/01/01", tz = "EST"),
        temp = ((t2m - 273.15) * 9/5) + 32,
        rain = tp,
        longitude = NULL, latitude = NULL,
        t2m = NULL, tp = NULL, time = NULL)] %>%
  .[, `:=`(DATE = as.ordered(as.Date(TIME, format = "%m/%d/%y", tz = "EST")),
        hour = hour(as.ITime(TIME)),
        TIME = NULL)]
head(w2012_2018)
```

##	temp	rain	DATE	hour
## 1:	68.25431	8.673617e-19	2011-12-31	19
## 2:	64.37574	8.673617e-19	2011-12-31	20
## 3:	62.21426	8.673617e-19	2011-12-31	21
## 4:	58.55789	8.673617e-19	2011-12-31	22
## 5:	55.87502	8.673617e-19	2012-01-01	1
## 6:	56.98711	8.673617e-19	2012-01-01	2

```
nc <- tidync("data/w2019_2021.nc")
w2019_2021 <- nc %>%
  hyper_tibble() %>%
  data.table() %>%
  .[, `:=`(TIME = as.POSIXct(time * 60 * 60,
                             origin = "1900/01/01", tz = "EST"),
        temp = ((t2m - 273.15) * 9/5) + 32,
        rain = tp,
        longitude = NULL, latitude = NULL,
        t2m = NULL, tp = NULL, time = NULL)] %>%
  .[, `:=`(DATE = as.ordered(as.Date(TIME, format = "%m/%d/%y", tz = "EST")),
        hour = hour(as.ITime(TIME)),
        TIME = NULL)]
head(w2019_2021)
```

##	temp	rain	DATE	hour
## 1:	75.68875	0.000000e+00	2018-12-31	19
## 2:	74.18016	0.000000e+00	2018-12-31	20
## 3:	72.44805	0.000000e+00	2018-12-31	21
## 4:	71.34443	0.000000e+00	2018-12-31	22
## 5:	69.46173	1.577227e-06	2019-01-01	1
## 6:	68.77805	5.914599e-07	2019-01-01	2

```
rbind(w2012_2018, w2019_2021)
```

```
## Warning in rbindlist(l, use.names, fill, idcol): Column 3 of item 2 is an
```

```
## ordered factor but level 2 ['2019-01-01'] is missing from the ordered levels
## from column 3 of item 1. Each set of ordered factor levels should be an ordered
## subset of the first longest. A regular factor will be created for this column.
```

```
##          temp          rain        DATE hour
##      1: 68.25431 8.673617e-19 2011-12-31   19
##      2: 64.37574 8.673617e-19 2011-12-31   20
##      3: 62.21426 8.673617e-19 2011-12-31   21
##      4: 58.55789 8.673617e-19 2011-12-31   22
##      5: 55.87502 8.673617e-19 2012-01-01    1
##      ---
## 80362: 79.92739 7.886133e-07 2021-12-31   14
## 80363: 80.74575 1.971533e-07 2021-12-31   15
## 80364: 81.03101 5.914599e-07 2021-12-31   16
## 80365: 78.36923 0.000000e+00 2021-12-31   17
## 80366: 75.39134 0.000000e+00 2021-12-31   18
```

```
# w2019_2021 <- w2019_2021 %>%
#   .[, `:=`(hour = hour(TIME), TIME = NULL)]

nc <- merge(datetime[, `:=`(hour = hour(TIME))],
            w2019_2021, by = c("DATE", "hour"), all.x = TRUE) %>%
  .[, hour := NULL] %>%
  .[rowSums(is.na(.)) == 0]
nc
```

```
##          DATE        TIME      temp rain
##      1: 2018-12-31 19:00:00 75.68875    0
##      2: 2018-12-31 19:05:00 75.68875    0
##      3: 2018-12-31 19:10:00 75.68875    0
##      4: 2018-12-31 19:15:00 75.68875    0
##      5: 2018-12-31 19:20:00 75.68875    0
##      ---
## 163196: 2021-08-31 22:35:00 82.04110    0
## 163197: 2021-08-31 22:40:00 82.04110    0
## 163198: 2021-08-31 22:45:00 82.04110    0
## 163199: 2021-08-31 22:50:00 82.04110    0
## 163200: 2021-08-31 22:55:00 82.04110    0
```

```
dt <- merge(dt, nc, by = c("DATE", "TIME"), all.x = TRUE) %>%
  .[, DATE := as.ordered(format(as.Date(DATE), format = "%m-%d"))]
dt

rm("datetime", "files", "nc", "w2012_2018", "w2019_2021")
```

## Split the data

```
saveRDS(dt, "BIG_DATA.rds")
set.seed(2022)
rowPicker <- sample(c(TRUE, FALSE), nrow(dt), replace = TRUE, prob = c(.5, .5))
```

```

train <- dt[rowPicker,]
train[, TIME := as.numeric(as.numeric(TIME) / 3600)]
test <- dt[!rowPicker,]
colnames(train) <- toupper(colnames(train))

```

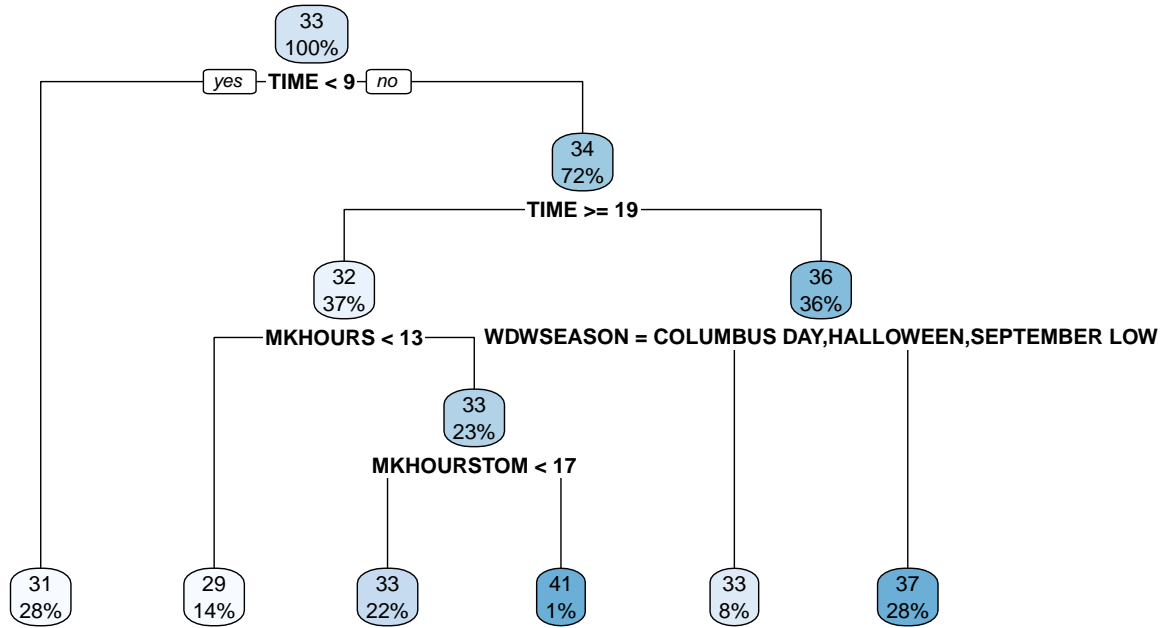
## Create the trees

```

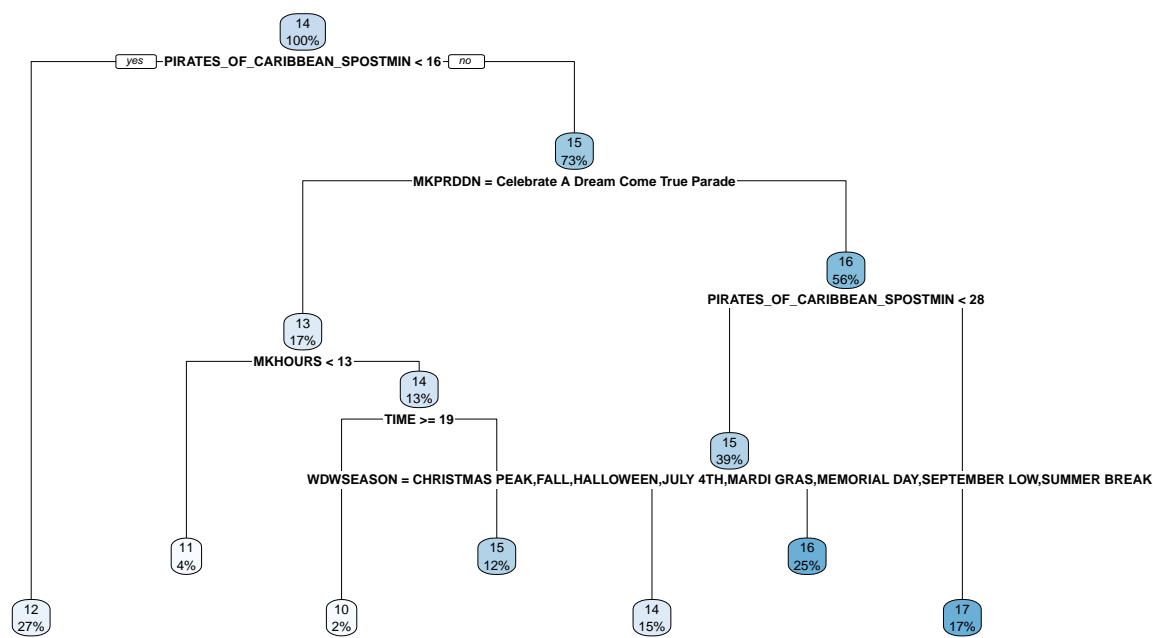
lapply(unique(dtMeta$PARK), function(x) {
  # get columns for the park
  # drop the columns for rides not in the park
  rides <-
    colnames(train)[!grepl(paste0(unique(dtMeta$PARK[!dtMeta$PARK %in% x]),
                                collapse = "|"), colnames(train))]
  rides <- rides[!grepl(paste0(dtMeta$RIDENAME[dtMeta$PARK != x],
                                collapse = "|"), rides)]
  # get the actual ride time variables
  actuals <- rides[grepl("(SACTMIN)", rides)]
  # return decision trees for the actual ride time variables
  return(sapply(actuals, function(y) {
    tr <- rpart(as.formula(paste(y, "~ .")), train[, ..rides])
    rpart.plot(tr, main = y)
    return(tr)
  }))
})

```

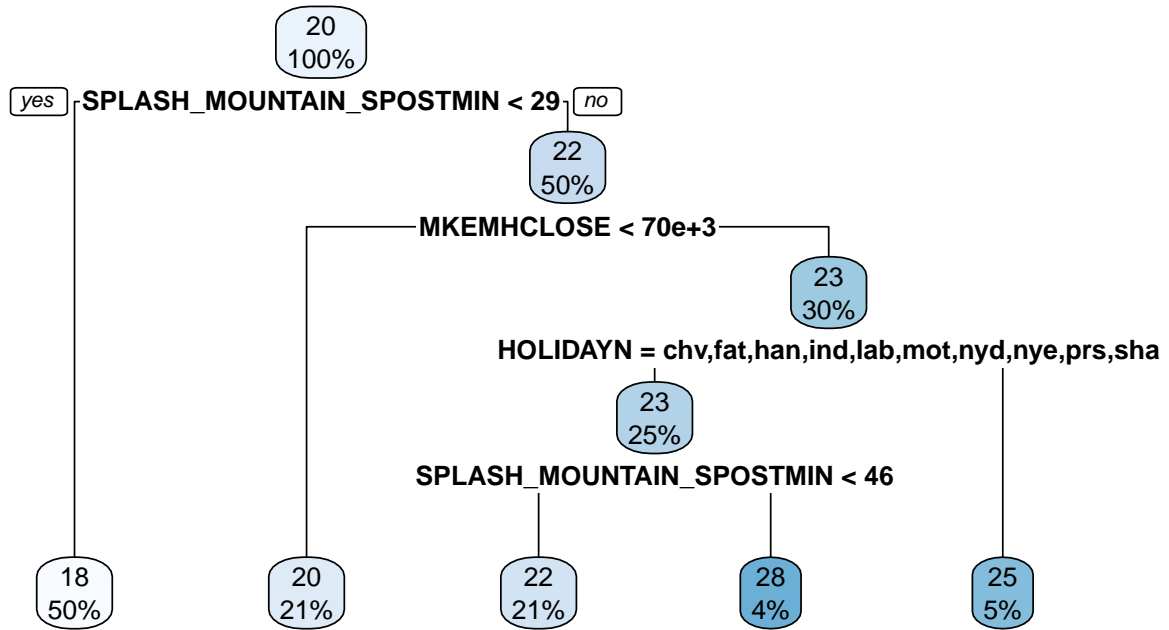
## DWARFS\_TRAIN\_SACTMIN



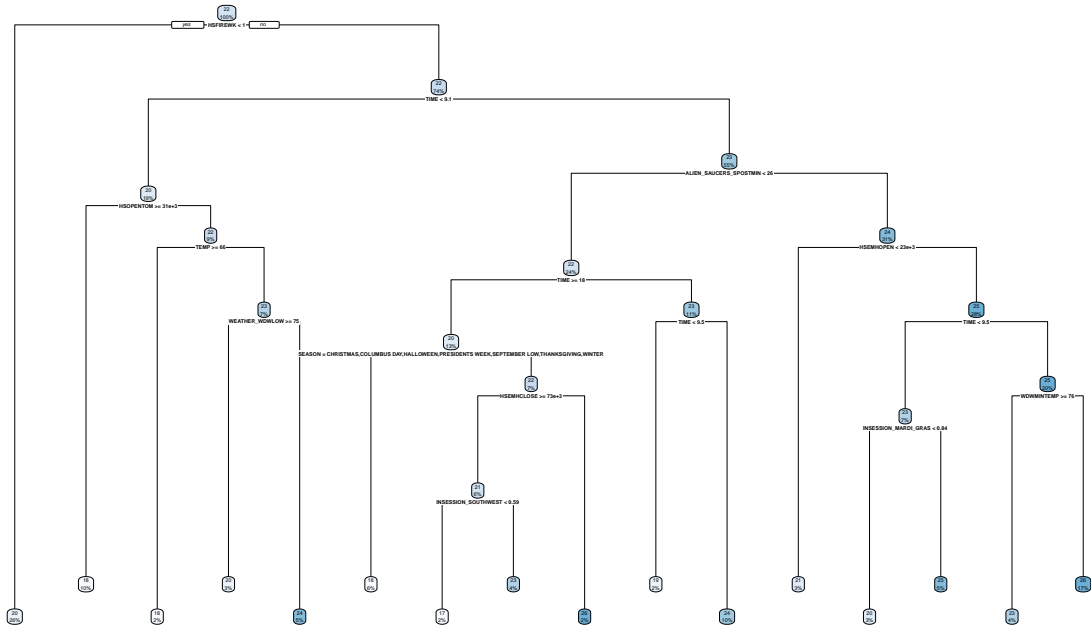
PIRATES\_OF\_CARIBBEAN\_SACTMIN



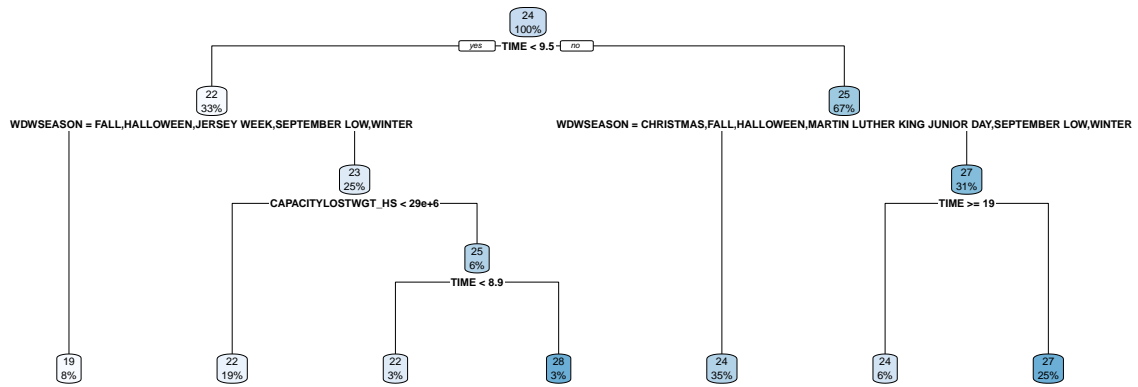
## SPLASH\_MOUNTAIN\_SACTMIN



# ALIEN\_SAUCERS\_SACTMIN

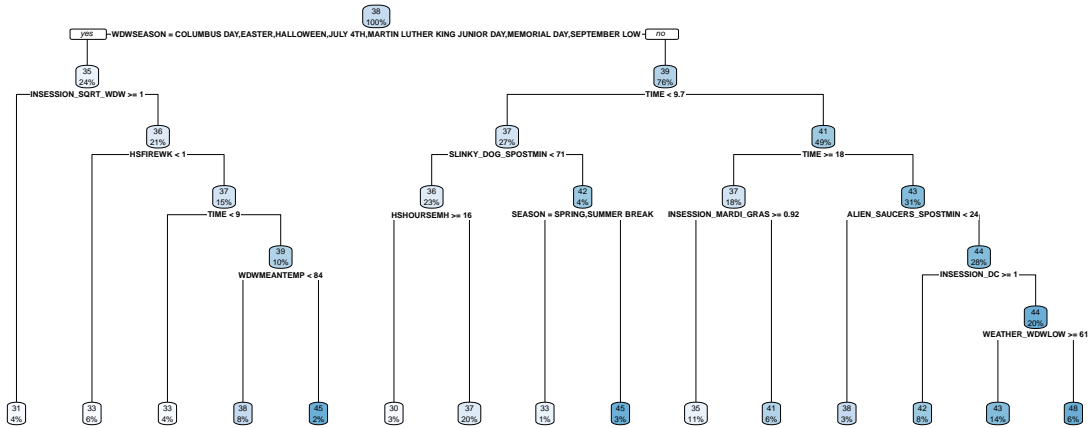


# ROCK\_N\_ROLLERCOASTER\_SACTMIN

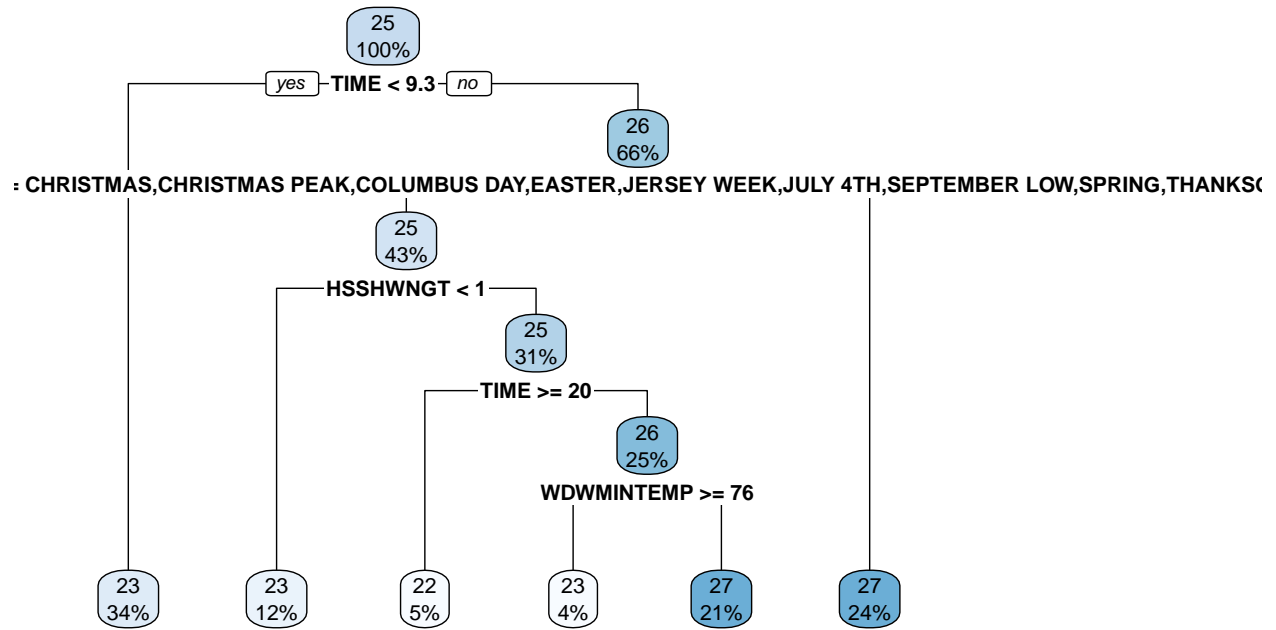




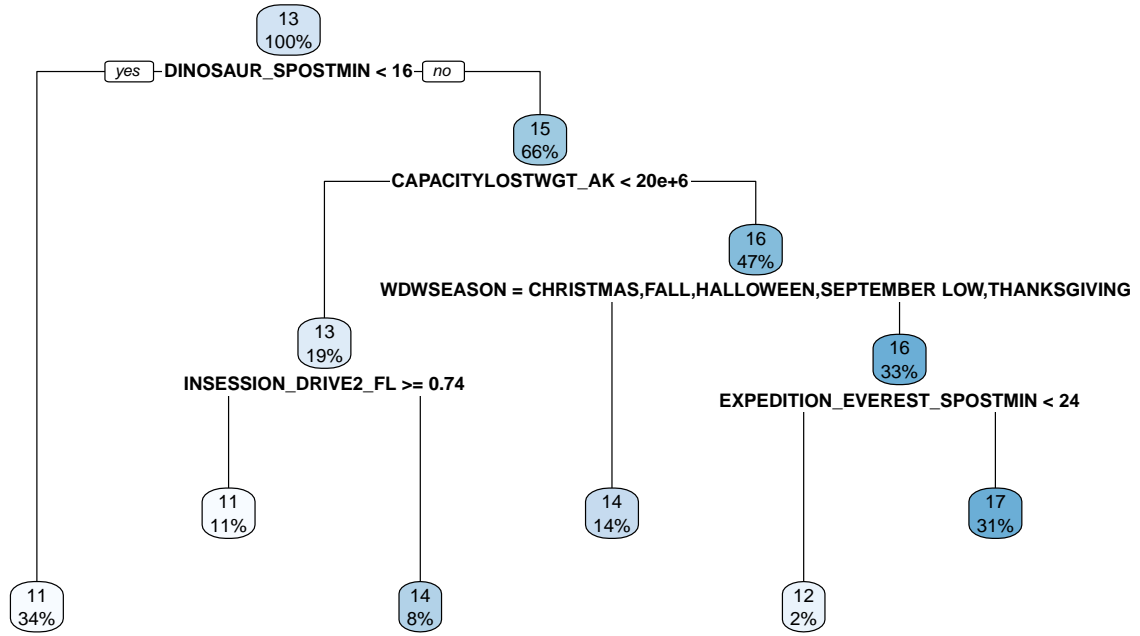
# SLINKY\_DOG\_SACTMIN



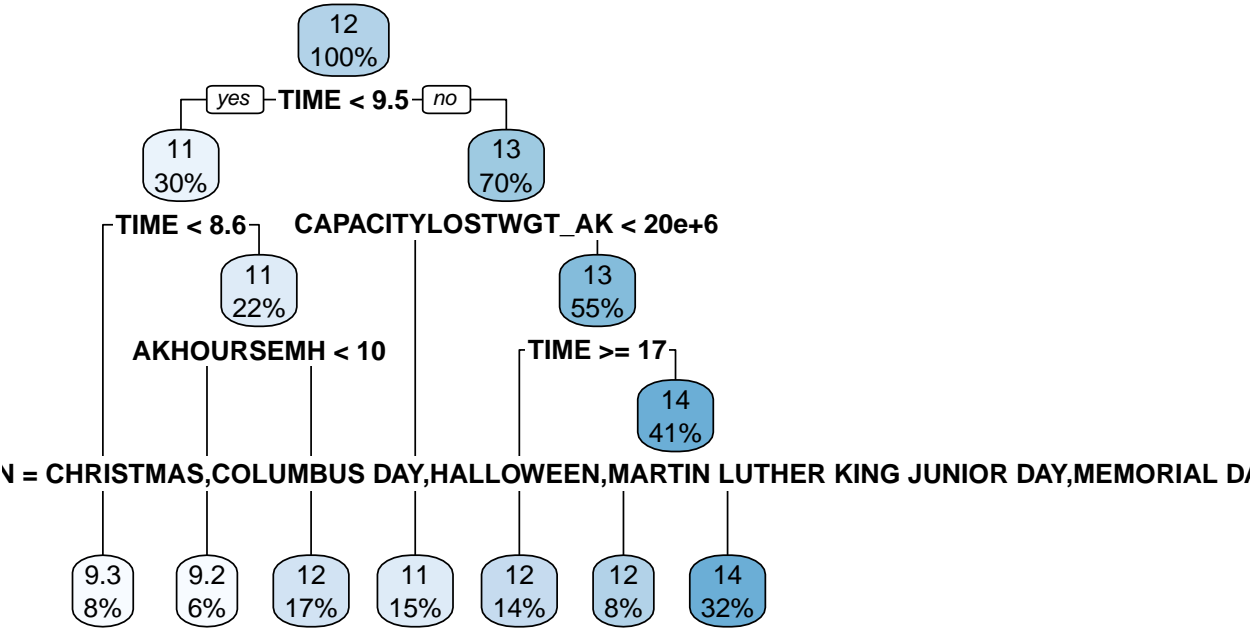
## TOY\_STORY\_MANIA\_SACTMIN

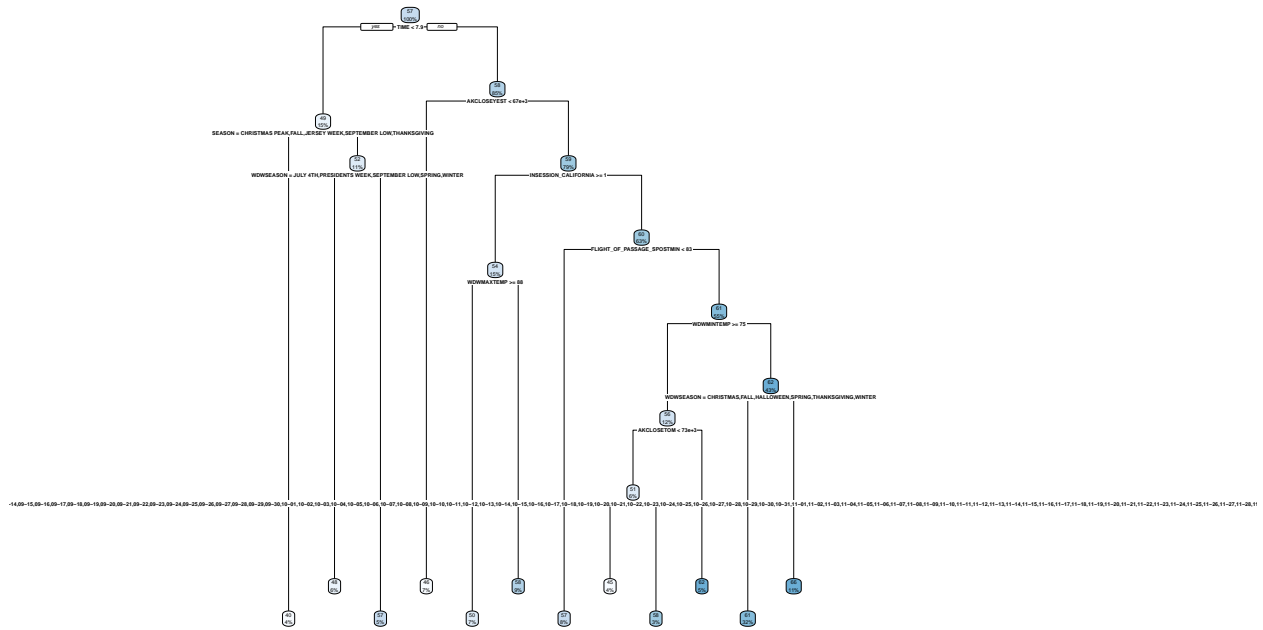


# DINOSAUR\_SACTMIN

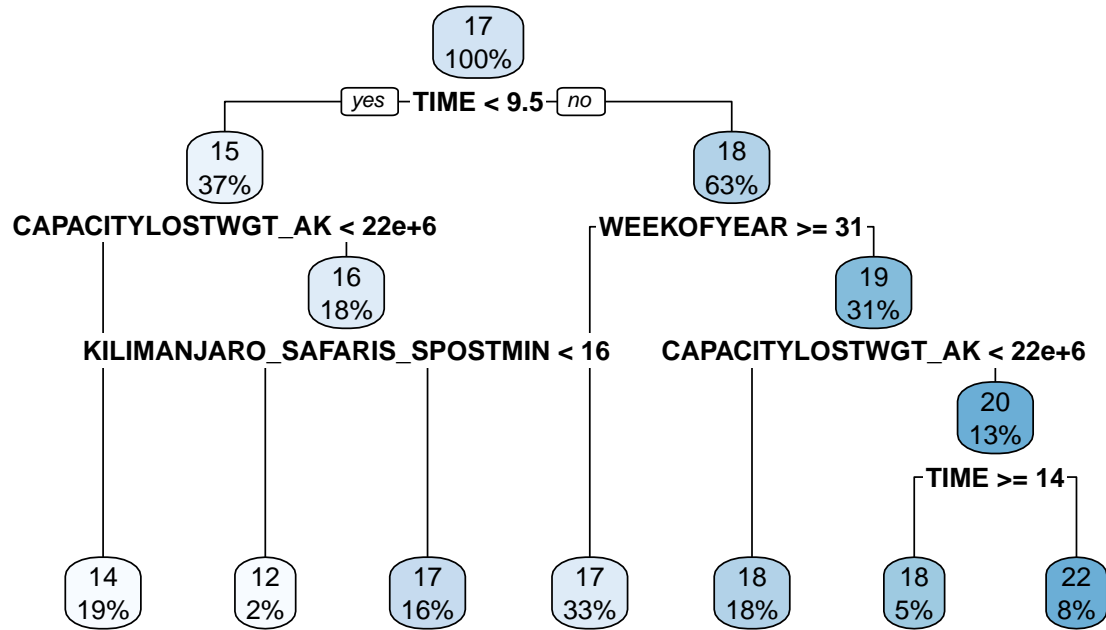


EXPEDITION\_EVEREST\_SACTMIN



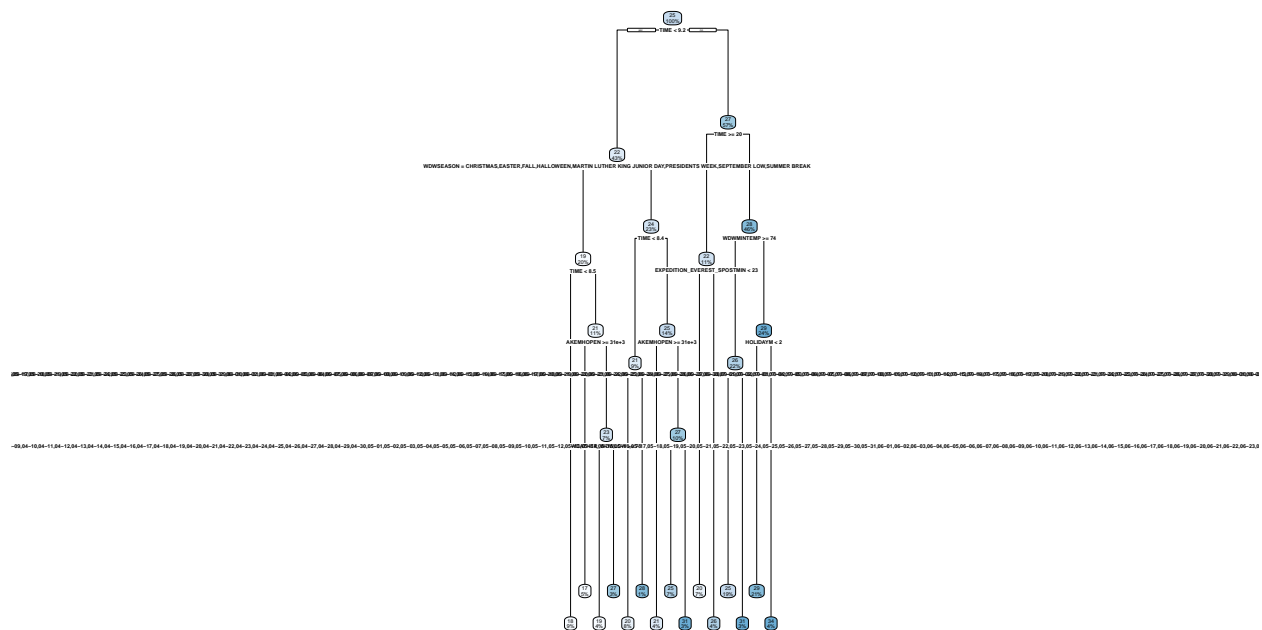
**FLIGHT\_OF\_PASSAGE\_SACTMIN**

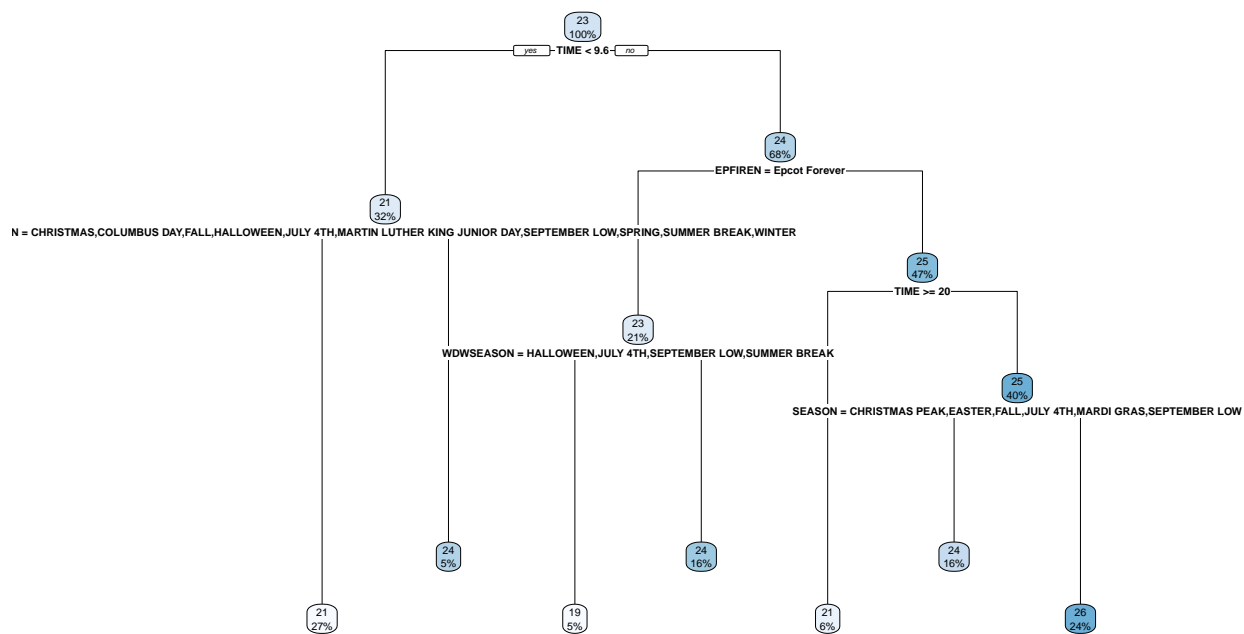
## KILIMANJARO\_SAFARIS\_SACTMIN



## Warning: labs do not fit even at cex 0.15, there may be some overplotting

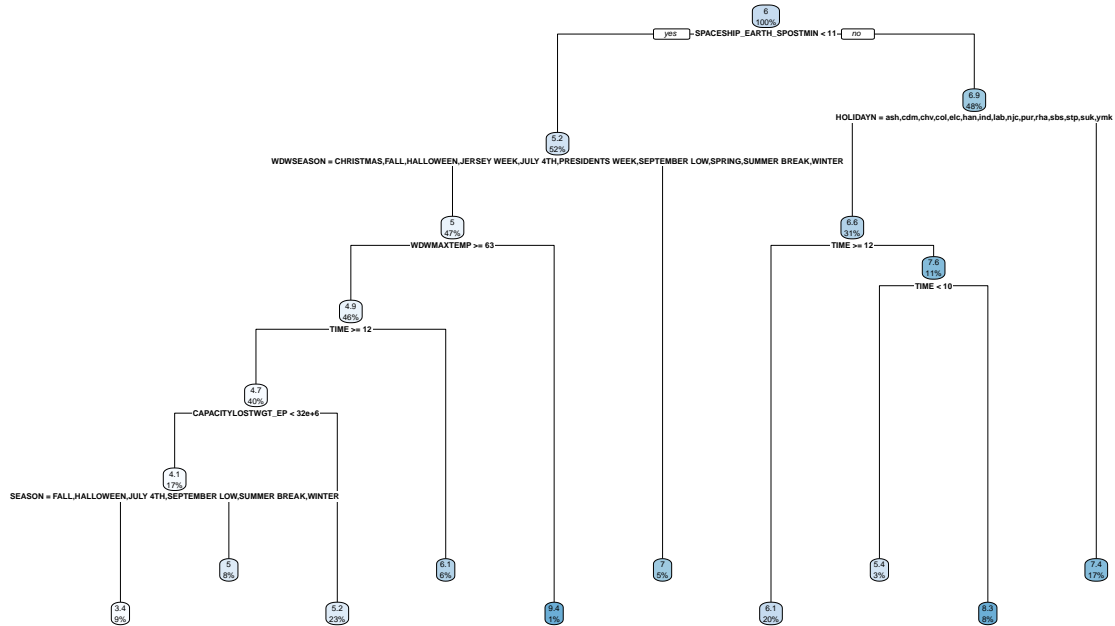
# NAVI RIVER SACTMIN



**SOARIN\_SACTMIN**



## SPACESHIP\_EARTH\_SACTMIN



## [[1]]		
##	DWARFS_TRAIN_SACTMIN	PIRATES_OF_CARIBBEAN_SACTMIN
## frame	data.frame,8	data.frame,8
## where	integer,1602	integer,3528
## call	expression	expression
## terms	terms,3	terms,3
## cptable	numeric,25	numeric,30
## method	"anova"	"anova"
## parms	NULL	NULL
## control	list,9	list,9
## functions	list,2	list,2
## numresp	1	1
## splits	numeric,235	numeric,280
## csplit	integer,4392	integer,5856
## variable.importance	numeric,17	numeric,22
## y	numeric,1602	numeric,3528
## ordered	logical,93	logical,93
## na.action	na.rpart,400211	na.rpart,398285
##	SPLASH_MOUNTAIN_SACTMIN	
## frame	data.frame,8	
## where	integer,1707	
## call	expression	
## terms	terms,3	
## cptable	numeric,20	
## method	"anova"	
## parms	NULL	

```

## control          list,9
## functions        list,2
## numresp          1
## splits           numeric,185
## csplit            integer,259
## variable.importance numeric,14
## y                numeric,1707
## ordered           logical,93
## na.action        na.rpart,400106
##
## [[2]]
##
## ALIEN_SAUCERS_SACTMIN ROCK_N_ROLLERCOASTER_SACTMIN
## frame            data.frame,8          data.frame,8
## where            integer,521            integer,1841
## call             expression             expression
## terms            terms,3                terms,3
## cptable          numeric,55              numeric,30
## method           "anova"                "anova"
## parms            NULL                    NULL
## control          list,9                  list,9
## functions        list,2                  list,2
## numresp          1                      1
## splits           numeric,730              numeric,300
## csplit            integer,8418            integer,5124
## variable.importance numeric,43            numeric,25
## y                numeric,521              numeric,1841
## ordered           logical,95              logical,95
## na.action        na.rpart,401292          na.rpart,399972
##
## SLINKY_DOG_SACTMIN TOY_STORY_MANIA_SACTMIN
## frame            data.frame,8          data.frame,8
## where            integer,820            integer,2242
## call             expression             expression
## terms            terms,3                terms,3
## cptable          numeric,50              numeric,20
## method           "anova"                "anova"
## parms            NULL                    NULL
## control          list,9                  list,9
## functions        list,2                  list,2
## numresp          1                      1
## splits           numeric,675              numeric,250
## csplit            integer,4392            integer,2196
## variable.importance numeric,41            numeric,24
## y                numeric,820              numeric,2242
## ordered           logical,95              logical,95
## na.action        na.rpart,400993          na.rpart,399571
##
## [[3]]
##
## DINOSAUR_SACTMIN EXPEDITION_EVEREST_SACTMIN
## frame            data.frame,8          data.frame,8
## where            integer,1752            integer,2158
## call             expression             expression
## terms            terms,3                terms,3
## cptable          numeric,30              numeric,30
## method           "anova"                "anova"

```

## parms	NULL	NULL
## control	list,9	list,9
## functions	list,2	list,2
## numresp	1	1
## splits	numeric,225	numeric,285
## csplit	integer,1464	integer,3660
## variable.importance	numeric,21	numeric,21
## y	numeric,1752	numeric,2158
## ordered	logical,93	logical,93
## na.action	na.rpart,400061	na.rpart,399655
##	FLIGHT_OF_PASSAGE_SACTMIN	KILIMANJARO_SAFARIS_SACTMIN
## frame	data.frame,8	data.frame,8
## where	integer,819	integer,1628
## call	expression	expression
## terms	terms,3	terms,3
## cptable	numeric,35	numeric,35
## method	"anova"	"anova"
## parms	NULL	NULL
## control	list,9	list,9
## functions	list,2	list,2
## numresp	1	1
## splits	numeric,525	numeric,285
## csplit	integer,5490	integer,4392
## variable.importance	numeric,30	numeric,26
## y	numeric,819	numeric,1628
## ordered	logical,93	logical,93
## na.action	na.rpart,400994	na.rpart,400185
##	NAVI_RIVER_SACTMIN	
## frame	data.frame,8	
## where	integer,767	
## call	expression	
## terms	terms,3	
## cptable	numeric,60	
## method	"anova"	
## parms	NULL	
## control	list,9	
## functions	list,2	
## numresp	1	
## splits	numeric,680	
## csplit	integer,10248	
## variable.importance	numeric,32	
## y	numeric,767	
## ordered	logical,93	
## na.action	na.rpart,401046	
##		
## [[4]]		
##	SOARIN_SACTMIN	SPACESHIP_EARTH_SACTMIN
## frame	data.frame,8	data.frame,8
## where	integer,2084	integer,1639
## call	expression	expression
## terms	terms,3	terms,3
## cptable	numeric,25	numeric,30
## method	"anova"	"anova"
## parms	NULL	NULL

## control	list,9	list,9
## functions	list,2	list,2
## numresp	1	1
## splits	numeric,300	numeric,410
## csplit	integer,6954	integer,7686
## variable.importance	numeric,27	numeric,32
## y	numeric,2084	numeric,1639
## ordered	logical,84	logical,84
## na.action	na.rpart,399729	na.rpart,400174