week\_5

Prateek Masali

2/14/2019

Table of Contents

**No table of contents entries found.**

Introducing Tideverse package with some of the functions in it

library("tidyverse")

## ── Attaching packages ──────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.0.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.1 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ─────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

Creating a table with 4 entries with different data types

information <- read\_csv("1,Tom,23,3.4,1,1988/08/28  
 2,Dan,30,3.6,0,1990-07-20  
 3,Sam,27,3.8,0,1988-07-19  
 4,Bill,33,4.0,1,1988-Nov-02", col\_names = c("No","Name","age","GPA","Resident","DOB"))  
head(information)

## # A tibble: 4 x 6  
## No Name age GPA Resident DOB   
## <int> <chr> <int> <dbl> <int> <chr>   
## 1 1 Tom 23 3.4 1 1988/08/28   
## 2 2 Dan 30 3.6 0 1990-07-20   
## 3 3 Sam 27 3.8 0 1988-07-19   
## 4 4 Bill 33 4 1 1988-Nov-02

Working with the data: filtering on valid information using logical parsing

information$Name[parse\_logical(information$Resident) == 1]

## [1] "Tom" "Bill"

#parse\_factor returns the values (levels) those are persent in the that field and that are not valid (marked as NA)  
parse\_factor(c("Tom","Bill","Ron","Casey"), levels = information$Name)

## Warning in rbind(names(probs), probs\_f): number of columns of result is not  
## a multiple of vector length (arg 1)

## Warning: 2 parsing failures.  
## row # A tibble: 2 x 4 col row col expected actual expected <int> <int> <chr> <chr> actual 1 3 NA value in level set Ron row 2 4 NA value in level set Casey

## [1] Tom Bill <NA> <NA>  
## attr(,"problems")  
## # A tibble: 2 x 4  
## row col expected actual  
## <int> <int> <chr> <chr>   
## 1 3 NA value in level set Ron   
## 2 4 NA value in level set Casey   
## Levels: Tom Dan Sam Bill

#parse\_date will return the record that fails the test  
information$DOB <- parse\_date(information$DOB)

## Warning: 1 parsing failure.  
## row # A tibble: 1 x 4 col row col expected actual expected <int> <int> <chr> <chr> actual 1 4 NA "date like " 1988-Nov-02

#problems function will list the record that fails the parsed field  
problems(information$DOB)

## # A tibble: 1 x 4  
## row col expected actual   
## <int> <int> <chr> <chr>   
## 1 4 NA "date like " 1988-Nov-02

head(information)

## # A tibble: 4 x 6  
## No Name age GPA Resident DOB   
## <int> <chr> <int> <dbl> <int> <date>   
## 1 1 Tom 23 3.4 1 1988-08-28  
## 2 2 Dan 30 3.6 0 1990-07-20  
## 3 3 Sam 27 3.8 0 1988-07-19  
## 4 4 Bill 33 4 1 NA

guess\_encoding(charToRaw(information$Name))

## Warning in charToRaw(information$Name): argument should be a character vector of length 1  
## all but the first element will be ignored

## # A tibble: 1 x 2  
## encoding confidence  
## <chr> <dbl>  
## 1 ASCII 1

library(hms)  
information$DOB[information$No == 4] = '1988-11-02'  
#parsing date through formatted pieces  
parse\_date(information$DOB[information$No == 4], "%Y-%m-%d")

## [1] "1988-11-02"

write\_csv(information, "~/Desktop/Harrisburg University/ANLY 506-90-O/Exploratory-Data-Analytics/Data/information.csv")