R - Practice 03

Sangkon Han(sangkon@pusan.ac.kr)

2023-09-01

Contents

Data Wrangle: strings - factors (stringr & forcats)

Strings inside tidyverse

```r
# Create a string
s1 <- "Double quotes string"
s2 <- 'Single quotes string'
s3 <- "Double quotes outside, 'single quotes inside' a string"
s4 <- 'Single quotes outside, "double quotes inside" a string'
# Not possible!
# s5 <- "Double quotes inside "double" quotes"
# s6 <- 'Single quotes inside 'single' quotes'
# s7 <-  "not "working"
# s8 <-  'not 'working'
# s9 <- "Missing closing quote
```

```r
# Create a vector of strings
vec <- c("a", "b", "c")
# Character vector inside a tibble
df <- tibble(letters = vec)
```

```r
# How to escape a character ~ Regular expressions

## Literal single or double quotes
```

```
"\""   # escape a special charatcer with back slash - \
```

```
## [1] "\""
```

```
'\''
```

```
## [1] "'"
```

```
## New line
"\n"
```

```
## [1] "\n"
```

```
## Tabulator
"\t"
```

```
## [1] "\t"
```

```
##   Unicode non-english characters
"\u03B1"
```

```
## [1] "α"
```

```
## See raw content of the string (omiited escape characters and outside quotes)
s <- "string"
writeLines(s)
```

```
## string
```

```
s <- "\""
writeLines(s)
```

```
## "
```

```
s <- "line 1 \nline 2"
writeLines(s)
```

```
## line 1
## line 2
```

Strings matching

```
# load strings dataset
load("./data/strings.RData")
```

```
# str_detect() - Detect s pattern
# similar base R: grepl()
# Find a fruit containing letter "a" (anywhere in word)
fruit
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "bell pepper"       "bilberry"
##  [7] "blackberry"        "blackcurrant"      "blood orange"
## [10] "blueberry"         "boysenberry"       "breadfruit"
## [13] "canary melon"      "cantaloupe"        "cherimoya"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "cranberry"
## [22] "cucumber"          "currant"           "damson"
## [25] "date"              "dragonfruit"       "durian"
## [28] "eggplant"          "elderberry"        "feijoa"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grape"             "grapefruit"        "guava"
## [37] "honeydew"          "huckleberry"       "jackfruit"
## [40] "jambul"            "jujube"            "kiwi fruit"
## [43] "kumquat"           "lemon"             "lime"
## [46] "loquat"            "lychee"            "mandarine"
## [49] "mango"             "mulberry"          "nectarine"
## [52] "nut"               "olive"             "orange"
## [55] "pamelo"            "papaya"            "passionfruit"
## [58] "peach"             "pear"              "persimmon"
## [61] "physalis"          "pineapple"         "plum"
## [64] "pomegranate"       "pomelo"            "purple mangosteen"
## [67] "quince"            "raisin"            "rambutan"
## [70] "raspberry"         "redcurrant"        "rock melon"
## [73] "salal berry"       "satsuma"           "star fruit"
## [76] "strawberry"        "tamarillo"         "tangerine"
## [79] "ugli fruit"        "watermelon"
```

```
ind <- str_detect(string = fruit, pattern = "a") # returns TRUE / FALSE
fruit[ind]
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "blackberry"        "blackcurrant"
##  [7] "blood orange"      "breadfruit"        "canary melon"
## [10] "cantaloupe"        "cherimoya"         "cranberry"
## [13] "currant"           "damson"            "date"
## [16] "dragonfruit"       "durian"            "eggplant"
## [19] "feijoa"            "grape"             "grapefruit"
## [22] "guava"             "jackfruit"         "jambul"
## [25] "kumquat"           "loquat"            "mandarine"
## [28] "mango"             "nectarine"         "orange"
## [31] "pamelo"            "papaya"            "passionfruit"
## [34] "peach"             "pear"              "physalis"
## [37] "pineapple"         "pomegranate"       "purple mangosteen"
## [40] "raisin"            "rambutan"          "raspberry"
## [43] "redcurrant"        "salal berry"       "satsuma"
## [46] "star fruit"        "strawberry"        "tamarillo"
## [49] "tangerine"         "watermelon"
```

```
# grepl
fruit[grepl(pattern = "a", x = fruit)]
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "blackberry"        "blackcurrant"
##  [7] "blood orange"      "breadfruit"        "canary melon"
## [10] "cantaloupe"        "cherimoya"         "cranberry"
## [13] "currant"           "damson"            "date"
## [16] "dragonfruit"       "durian"            "eggplant"
## [19] "feijoa"            "grape"             "grapefruit"
## [22] "guava"             "jackfruit"         "jambul"
## [25] "kumquat"           "loquat"            "mandarine"
## [28] "mango"             "nectarine"         "orange"
## [31] "pamelo"            "papaya"            "passionfruit"
## [34] "peach"             "pear"              "physalis"
## [37] "pineapple"         "pomegranate"       "purple mangosteen"
## [40] "raisin"            "rambutan"          "raspberry"
## [43] "redcurrant"        "salal berry"       "satsuma"
## [46] "star fruit"        "strawberry"        "tamarillo"
## [49] "tangerine"         "watermelon"
```

```
# Find a fruit not containing any letter "a" !

# we use negation
fruit[str_detect(fruit, "a", negate = T)]
```

```
##  [1] "bell pepper"  "bilberry"     "blueberry"    "boysenberry"  "cherry"
##  [6] "chili pepper" "clementine"   "cloudberry"   "coconut"      "cucumber"
## [11] "elderberry"   "fig"          "goji berry"   "gooseberry"   "honeydew"
## [16] "huckleberry"  "jujube"       "kiwi fruit"   "lemon"        "lime"
## [21] "lychee"       "mulberry"     "nut"          "olive"        "persimmon"
## [26] "plum"         "pomelo"       "quince"       "rock melon"   "ugli fruit"
```

```
fruit[!str_detect(fruit, "a")]
```

```
##  [1] "bell pepper"  "bilberry"     "blueberry"    "boysenberry"  "cherry"
##  [6] "chili pepper" "clementine"   "cloudberry"   "coconut"      "cucumber"
## [11] "elderberry"   "fig"          "goji berry"   "gooseberry"   "honeydew"
## [16] "huckleberry"  "jujube"       "kiwi fruit"   "lemon"        "lime"
## [21] "lychee"       "mulberry"     "nut"          "olive"        "persimmon"
## [26] "plum"         "pomelo"       "quince"       "rock melon"   "ugli fruit"
```

```
# Inside tibble add flag if fruit contains letter "a" or if it doesn't contain letter "a"
fruit.df %>%
  mutate(flag = case_when(str_detect(fruit, pattern = "a") ~ "contains 'a'", T ~ "does not contain 'a'"
```

```
## # A tibble: 80 x 2
##    fruit        flag
##    <chr>        <chr>
##  1 apple        contains 'a'
##  2 apricot      contains 'a'
##  3 avocado      contains 'a'
##  4 banana       contains 'a'
##  5 bell pepper  does not contain 'a'
##  6 bilberry     does not contain 'a'
##  7 blackberry   contains 'a'
##  8 blackcurrant contains 'a'
##  9 blood orange contains 'a'
## 10 blueberry    does not contain 'a'
## # i 70 more rows
```

```
# Find fruits starting or ending with letter "a"
ind.start.a <- str_detect(fruit, pattern = "^a")
fruit[ind.start.a]
```

```
## [1] "apple"   "apricot" "avocado"
```

```
ind.end.a <- str_detect(fruit, pattern = "a$")
fruit[ind.end.a]
```

```
## [1] "banana"    "cherimoya" "feijoa"    "guava"     "papaya"    "satsuma"
```

```
# str_which() - Detect s pattern return index position
# similar base R: grep()
#   Find a fruit containing letter "a" (anywhere in word)
```

```r
ind <- str_which(string = fruit, pattern = "a") # returns index position
fruit[ind]
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "blackberry"        "blackcurrant"
##  [7] "blood orange"      "breadfruit"        "canary melon"
## [10] "cantaloupe"        "cherimoya"         "cranberry"
## [13] "currant"           "damson"            "date"
## [16] "dragonfruit"       "durian"            "eggplant"
## [19] "feijoa"            "grape"             "grapefruit"
## [22] "guava"             "jackfruit"         "jambul"
## [25] "kumquat"           "loquat"            "mandarine"
## [28] "mango"             "nectarine"         "orange"
## [31] "pamelo"            "papaya"            "passionfruit"
## [34] "peach"             "pear"              "physalis"
## [37] "pineapple"         "pomegranate"       "purple mangosteen"
## [40] "raisin"            "rambutan"          "raspberry"
## [43] "redcurrant"        "salal berry"       "satsuma"
## [46] "star fruit"        "strawberry"        "tamarillo"
## [49] "tangerine"         "watermelon"
```

```r
fruit[grep(pattern = "a", x = fruit)]
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "blackberry"        "blackcurrant"
##  [7] "blood orange"      "breadfruit"        "canary melon"
## [10] "cantaloupe"        "cherimoya"         "cranberry"
## [13] "currant"           "damson"            "date"
## [16] "dragonfruit"       "durian"            "eggplant"
## [19] "feijoa"            "grape"             "grapefruit"
## [22] "guava"             "jackfruit"         "jambul"
## [25] "kumquat"           "loquat"            "mandarine"
## [28] "mango"             "nectarine"         "orange"
## [31] "pamelo"            "papaya"            "passionfruit"
## [34] "peach"             "pear"              "physalis"
## [37] "pineapple"         "pomegranate"       "purple mangosteen"
## [40] "raisin"            "rambutan"          "raspberry"
## [43] "redcurrant"        "salal berry"       "satsuma"
## [46] "star fruit"        "strawberry"        "tamarillo"
## [49] "tangerine"         "watermelon"
```

```r
# str_count() - Count number of pattern matches in string

# Add count of letter "a" in each fruit (use table)
fruit.df1 <- fruit.df %>%
  mutate(`count a` = str_count(fruit, pattern = "a"))

# Show counts of letter "a" in fruits
fruit.df1 %>%
  count(`count a`)
```

```
## # A tibble: 4 x 2
##   `count a`     n
##       <int> <int>
## 1         0    30
## 2          1    37
## 3          2    11
## 4          3     2
```

```
# Show fruit with 3 "a" letters
fruit.df1 %>%
  filter(`count a` == 3)
```

```
## # A tibble: 2 x 2
##   fruit  `count a`
##   <chr>      <int>
## 1 banana         3
## 2 papaya         3
```

```
# str_locate() / str_locate_all() - Locate position(s) of pattern match in string

# Locate position of first letter "a" in each fruit (matrix is returned)
str_locate(fruit, pattern = "a")
```

```
##       start end
##  [1,]     1   1
##  [2,]     1   1
##  [3,]     1   1
##  [4,]     2   2
##  [5,]    NA  NA
##  [6,]    NA  NA
##  [7,]     3   3
##  [8,]     3   3
##  [9,]     9   9
## [10,]    NA  NA
## [11,]    NA  NA
## [12,]     4   4
## [13,]     2   2
## [14,]     2   2
## [15,]     9   9
## [16,]    NA  NA
## [17,]    NA  NA
## [18,]    NA  NA
## [19,]    NA  NA
## [20,]    NA  NA
## [21,]     3   3
## [22,]    NA  NA
## [23,]     5   5
## [24,]     2   2
## [25,]     2   2
## [26,]     3   3
## [27,]     5   5
## [28,]     6   6
## [29,]    NA  NA
## [30,]     6   6
## [31,]    NA  NA
## [32,]    NA  NA
## [33,]    NA  NA
## [34,]     3   3
## [35,]     3   3
## [36,]     3   3
## [37,]    NA  NA
## [38,]    NA  NA
## [39,]     2   2
## [40,]     2   2
## [41,]    NA  NA
## [42,]    NA  NA
## [43,]     6   6
## [44,]    NA  NA
## [45,]    NA  NA
## [46,]     5   5
## [47,]    NA  NA
## [48,]     2   2
## [49,]     2   2
## [50,]    NA  NA
## [51,]     5   5
## [52,]    NA  NA
## [53,]    NA  NA
## [54,]     3   3
## [55,]     2   2
## [56,]     2   2
## [57,]     2   2
## [58,]     3   3
## [59,]     3   3
## [60,]    NA  NA
## [61,]     5   5
## [62,]     5   5
## [63,]    NA  NA
## [64,]     7   7
## [65,]    NA  NA
## [66,]     9   9
## [67,]    NA  NA
## [68,]     2   2
## [69,]     2   2
## [70,]     2   2
## [71,]     8   8
## [72,]    NA  NA
## [73,]     2   2
## [74,]     2   2
## [75,]     3   3
## [76,]     4   4
## [77,]     2   2
## [78,]     2   2
## [79,]    NA  NA
## [80,]     2   2
```

```
fruit.df1 <- str_locate(fruit, pattern = "a") %>%
  as_tibble() %>% # convert matrix of positions to tibble
  mutate(fruit = fruit) %>% # add fruit name column
  select(fruit, start, end) # re-arrange columns

# Locate position of all letters "a" in each fruit (list is returned)
str_locate_all(fruit, pattern = "a")
```

```
## [[1]]
##      start end
## [1,]     1   1
##
## [[2]]
##      start end
## [1,]     1   1
##
## [[3]]
##      start end
## [1,]     1   1
## [2,]     5   5
##
## [[4]]
##      start end
## [1,]     2   2
## [2,]     4   4
## [3,]     6   6
##
## [[5]]
##      start end
##
## [[6]]
##      start end
##
## [[7]]
##      start end
## [1,]     3   3
##
## [[8]]
##      start end
## [1,]     3   3
## [2,]    10  10
##
## [[9]]
##      start end
## [1,]     9   9
##
## [[10]]
##      start end
##
## [[11]]
##      start end
##
## [[12]]
##      start end
## [1,]     4   4
##
## [[13]]
##      start end
## [1,]     2   2
## [2,]     4   4
##
## [[14]]
##      start end
## [1,]     2   2
## [2,]     5   5
##
## [[15]]
##      start end
## [1,]     9   9
##
## [[16]]
##      start end
##
## [[17]]
##      start end
##
## [[18]]
##      start end
##
## [[19]]
##      start end
##
## [[20]]
##      start end
##
## [[21]]
##      start end
## [1,]     3   3
##
## [[22]]
##      start end
##
## [[23]]
##      start end
## [1,]     5   5
##
## [[24]]
##      start end
## [1,]     2   2
##
## [[25]]
##      start end
## [1,]     2   2
##
## [[26]]
##      start end
## [1,]     3   3
##
## [[27]]
##      start end
## [1,]     5   5
##
## [[28]]
##      start end
## [1,]     6   6
##
## [[29]]
##      start end
```

```
##
## [[30]]
##      start end
## [1,]     6   6
##
## [[31]]
##      start end
##
## [[32]]
##      start end
##
## [[33]]
##      start end
##
## [[34]]
##      start end
## [1,]     3   3
##
## [[35]]
##      start end
## [1,]     3   3
##
## [[36]]
##      start end
## [1,]     3   3
## [2,]     5   5
##
## [[37]]
##      start end
##
## [[38]]
##      start end
##
## [[39]]
##      start end
## [1,]     2   2
##
## [[40]]
##      start end
## [1,]     2   2
##
## [[41]]
##      start end
##
## [[42]]
##      start end
##
## [[43]]
##      start end
## [1,]     6   6
##
## [[44]]
##      start end
##
```

```
## [[45]]
##      start end
##
## [[46]]
##      start end
## [1,]     5   5
##
## [[47]]
##      start end
##
## [[48]]
##      start end
## [1,]     2   2
## [2,]     5   5
##
## [[49]]
##      start end
## [1,]     2   2
##
## [[50]]
##      start end
##
## [[51]]
##      start end
## [1,]     5   5
##
## [[52]]
##      start end
##
## [[53]]
##      start end
##
## [[54]]
##      start end
## [1,]     3   3
##
## [[55]]
##      start end
## [1,]     2   2
##
## [[56]]
##      start end
## [1,]     2   2
## [2,]     4   4
## [3,]     6   6
##
## [[57]]
##      start end
## [1,]     2   2
##
## [[58]]
##      start end
## [1,]     3   3
##
```

```
## [[59]]
##      start end
## [1,]     3   3
##
## [[60]]
##      start end
##
## [[61]]
##      start end
## [1,]     5   5
##
## [[62]]
##      start end
## [1,]     5   5
##
## [[63]]
##      start end
##
## [[64]]
##      start end
## [1,]     7   7
## [2,]     9   9
##
## [[65]]
##      start end
##
## [[66]]
##      start end
## [1,]     9   9
##
## [[67]]
##      start end
##
## [[68]]
##      start end
## [1,]     2   2
##
## [[69]]
##      start end
## [1,]     2   2
## [2,]     7   7
##
## [[70]]
##      start end
## [1,]     2   2
##
## [[71]]
##      start end
## [1,]     8   8
##
## [[72]]
##      start end
##
## [[73]]
##      start end
## [1,]     2   2
## [2,]     4   4
##
## [[74]]
##      start end
## [1,]     2   2
## [2,]     7   7
##
## [[75]]
##      start end
## [1,]     3   3
##
## [[76]]
##      start end
## [1,]     4   4
##
## [[77]]
##      start end
## [1,]     2   2
## [2,]     4   4
##
## [[78]]
##      start end
## [1,]     2   2
##
## [[79]]
##      start end
##
## [[80]]
##      start end
## [1,]     2   2
```

Strings subsetting

```
# str_sub() - Extract part of s string
# similar base R: substr()

# Extract first 3 letters of a fruit
str_sub(string = fruit, start = 1, end = 3)
```

```
##  [1] "app" "apr" "avo" "ban" "bel" "bil" "bla" "bla" "blo" "blu" "boy" "bre"
## [13] "can" "can" "che" "che" "chi" "cle" "clo" "coc" "cra" "cuc" "cur" "dam"
## [25] "dat" "dra" "dur" "egg" "eld" "fei" "fig" "goj" "goo" "gra" "gra" "gua"
## [37] "hon" "huc" "jac" "jam" "juj" "kiw" "kum" "lem" "lim" "loq" "lyc" "man"
## [49] "man" "mul" "nec" "nut" "oli" "ora" "pam" "pap" "pas" "pea" "pea" "per"
## [61] "phy" "pin" "plu" "pom" "pom" "pur" "qui" "rai" "ram" "ras" "red" "roc"
## [73] "sal" "sat" "sta" "str" "tam" "tan" "ugl" "wat"
```

```
substr(x = fruit, start = 1, stop = 3)
```

```
##  [1] "app" "apr" "avo" "ban" "bel" "bil" "bla" "bla" "blo" "blu" "boy" "bre"
## [13] "can" "can" "che" "che" "chi" "cle" "clo" "coc" "cra" "cuc" "cur" "dam"
## [25] "dat" "dra" "dur" "egg" "eld" "fei" "fig" "goj" "goo" "gra" "gra" "gua"
## [37] "hon" "huc" "jac" "jam" "juj" "kiw" "kum" "lem" "lim" "loq" "lyc" "man"
## [49] "man" "mul" "nec" "nut" "oli" "ora" "pam" "pap" "pas" "pea" "pea" "per"
## [61] "phy" "pin" "plu" "pom" "pom" "pur" "qui" "rai" "ram" "ras" "red" "roc"
## [73] "sal" "sat" "sta" "str" "tam" "tan" "ugl" "wat"
```

```
# Extract first letter of common word and count word frequency by first word letter
words.df %>%
  mutate(`first letter` = str_sub(word, 1, 1)) %>% # extract first letter
  count(`first letter`) %>% # count frequencies
  arrange(desc(n)) # sort from high to low frequency
```

```
## # A tibble: 42 x 2
##    `first letter`     n
##    <chr>          <int>
##  1 s                334
##  2 c                297
##  3 p                246
##  4 a                214
##  5 t                174
##  6 e                164
##  7 r                163
##  8 d                161
##  9 b                143
## 10 f                141
## # i 32 more rows
```

```
# Extract middle part of the word
str_sub(fruit, start = 3, end = 5) # from 3rd to 5th letter
```

```
##  [1] "ple" "ric" "oca" "nan" "ll " "lbe" "ack" "ack" "ood" "ueb" "yse" "ead"
## [13] "nar" "nta" "eri" "err" "ili" "eme" "oud" "con" "anb" "cum" "rra" "mso"
## [25] "te"  "ago" "ria" "gpl" "der" "ijo" "g"   "ji " "ose" "ape" "ape" "ava"
## [37] "ney" "ckl" "ckf" "mbu" "jub" "wi " "mqu" "mon" "me"  "qua" "che" "nda"
## [49] "ngo" "lbe" "cta" "t"   "ive" "ang" "mel" "pay" "ssi" "ach" "ar"  "rsi"
## [61] "ysa" "nea" "um"  "meg" "mel" "rpl" "inc" "isi" "mbu" "spb" "dcu" "ck "
## [73] "lal" "tsu" "ar " "raw" "mar" "nge" "li " "ter"
```

```
# Extract last letter / last 3 letters (use negative counters - for counting backward)
str_sub(fruit, start = -1, end = -1) # last letter
```

```
##  [1] "e" "t" "o" "a" "r" "y" "y" "t" "e" "y" "y" "t" "n" "e" "a" "y" "r" "e" "y"
## [20] "t" "y" "r" "t" "n" "e" "t" "n" "t" "y" "a" "g" "y" "y" "e" "t" "a" "w" "y"
## [39] "t" "l" "e" "t" "t" "n" "e" "t" "e" "e" "o" "y" "e" "t" "e" "e" "o" "a" "t"
## [58] "h" "r" "n" "s" "e" "m" "e" "o" "n" "e" "n" "n" "y" "t" "n" "y" "a" "t" "y"
## [77] "o" "e" "t" "n"
```

```
str_sub(fruit, start = -3, end = -1) # last 3 letters
```

```
##  [1] "ple" "cot" "ado" "ana" "per" "rry" "rry" "ant" "nge" "rry" "rry" "uit"
## [13] "lon" "upe" "oya" "rry" "per" "ine" "rry" "nut" "rry" "ber" "ant" "son"
## [25] "ate" "uit" "ian" "ant" "rry" "joa" "fig" "rry" "rry" "ape" "uit" "ava"
## [37] "dew" "rry" "uit" "bul" "ube" "uit" "uat" "mon" "ime" "uat" "hee" "ine"
## [49] "ngo" "rry" "ine" "nut" "ive" "nge" "elo" "aya" "uit" "ach" "ear" "mon"
## [61] "lis" "ple" "lum" "ate" "elo" "een" "nce" "sin" "tan" "rry" "ant" "lon"
## [73] "rry" "uma" "uit" "rry" "llo" "ine" "uit" "lon"
```

```
# str_subset() - Return only strings that match pattern
# Return fruit containing letter "c"
```

```
str_subset(string = fruit, pattern = "c")
```

```
##  [1] "apricot"      "avocado"      "blackberry"   "blackcurrant" "canary melon"
##  [6] "cantaloupe"   "cherimoya"    "cherry"       "chili pepper" "clementine"
## [11] "cloudberry"   "coconut"      "cranberry"    "cucumber"     "currant"
## [16] "huckleberry"  "jackfruit"    "lychee"       "nectarine"    "peach"
## [21] "quince"       "redcurrant"   "rock melon"
```

```
# Return fruit starting with letter "c"
str_subset(string = fruit, pattern = "^c")
```

```
##  [1] "canary melon" "cantaloupe"   "cherimoya"    "cherry"       "chili pepper"
##  [6] "clementine"   "cloudberry"   "coconut"      "cranberry"    "cucumber"
## [11] "currant"
```

```
# str_extract() / str_extract_all() - Return first or every pattern match
# Return fruit containing "a" first occurence
str_extract(string = fruit, pattern = "a")  # vector is returned
```

```
##  [1] "a" "a" "a" "a" NA  NA  "a" "a" "a" NA  NA  "a" "a" "a" "a" NA  NA  NA  NA
## [20] NA  "a" NA  "a" "a" "a" "a" "a" "a" NA  "a" NA  NA  NA  "a" "a" "a" NA  NA
## [39] "a" "a" NA  NA  "a" NA  NA  "a" NA  "a" "a" NA  "a" NA  NA  "a" "a" "a" "a"
## [58] "a" "a" NA  "a" "a" NA  "a" NA  "a" NA  "a" "a" "a" "a" NA  "a" "a" "a" "a"
## [77] "a" "a" NA  "a"
```

```
# Return fruit containing "a" all occurences
str_extract_all(string = fruit, pattern = "a") #  list is returned
```

```
## [[1]]
## [1] "a"
##
## [[2]]
## [1] "a"
##
## [[3]]
## [1] "a" "a"
##
## [[4]]
## [1] "a" "a" "a"
##
## [[5]]
## character(0)
##
## [[6]]
## character(0)
##
## [[7]]
## [1] "a"
##
## [[8]]
## [1] "a" "a"
##
## [[9]]
## [1] "a"
##
## [[10]]
## character(0)
##
## [[11]]
## character(0)
##
## [[12]]
## [1] "a"
##
## [[13]]
## [1] "a" "a"
##
## [[14]]
## [1] "a" "a"
##
## [[15]]
## [1] "a"
##
## [[16]]
## character(0)
##
## [[17]]
## character(0)
##
## [[18]]
## character(0)
##
## [[19]]
## character(0)
##
## [[20]]
## character(0)
##
## [[21]]
## [1] "a"
##
## [[22]]
## character(0)
##
## [[23]]
## [1] "a"
##
## [[24]]
## [1] "a"
##
## [[25]]
## [1] "a"
##
## [[26]]
## [1] "a"
##
## [[27]]
## [1] "a"
##
## [[28]]
## [1] "a"
##
## [[29]]
## character(0)
##
## [[30]]
## [1] "a"
##
## [[31]]
## character(0)
##
## [[32]]
## character(0)
##
## [[33]]
## character(0)
##
## [[34]]
## [1] "a"
##
## [[35]]
## [1] "a"
##
## [[36]]
## [1] "a" "a"
##
## [[37]]
## character(0)
##
## [[38]]
## character(0)
##
## [[39]]
## [1] "a"
##
## [[40]]
## [1] "a"
##
## [[41]]
## character(0)
##
## [[42]]
## character(0)
##
## [[43]]
## [1] "a"
##
## [[44]]
## character(0)
##
## [[45]]
## character(0)
##
## [[46]]
## [1] "a"
##
## [[47]]
## character(0)
##
## [[48]]
## [1] "a" "a"
##
## [[49]]
## [1] "a"
##
## [[50]]
## character(0)
##
## [[51]]
## [1] "a"
##
## [[52]]
## character(0)
##
## [[53]]
## character(0)
##
## [[54]]
## [1] "a"
##
## [[55]]
## [1] "a"
##
## [[56]]
## [1] "a" "a" "a"
##
## [[57]]
## [1] "a"
##
## [[58]]
## [1] "a"
##
## [[59]]
## [1] "a"
##
## [[60]]
## character(0)
##
## [[61]]
## [1] "a"
##
## [[62]]
## [1] "a"
##
## [[63]]
## character(0)
##
## [[64]]
## [1] "a" "a"
##
## [[65]]
## character(0)
##
## [[66]]
## [1] "a"
##
## [[67]]
## character(0)
##
## [[68]]
## [1] "a"
##
## [[69]]
## [1] "a" "a"
##
## [[70]]
## [1] "a"
##
## [[71]]
## [1] "a"
##
## [[72]]
## character(0)
##
## [[73]]
## [1] "a" "a"
##
## [[74]]
## [1] "a" "a"
##
## [[75]]
## [1] "a"
##
## [[76]]
## [1] "a"
##
## [[77]]
## [1] "a" "a"
##
## [[78]]
## [1] "a"
##
## [[79]]
## character(0)
##
## [[80]]
## [1] "a"
```

```
# str_match() / str_match_all() - Return first or every pattern match (as a matrx)
# Return fruit containing "a" first occurence
str_match(string = fruit, pattern = "a")  # matrix is returned
```

```
##       [,1]
##  [1,] "a"
##  [2,] "a"
##  [3,] "a"
##  [4,] "a"
##  [5,] NA
##  [6,] NA
##  [7,] "a"
##  [8,] "a"
##  [9,] "a"
## [10,] NA
## [11,] NA
## [12,] "a"
## [13,] "a"
## [14,] "a"
## [15,] "a"
## [16,] NA
## [17,] NA
## [18,] NA
## [19,] NA
## [20,] NA
## [21,] "a"
## [22,] NA
## [23,] "a"
## [24,] "a"
## [25,] "a"
## [26,] "a"
## [27,] "a"
## [28,] "a"
## [29,] NA
## [30,] "a"
## [31,] NA
## [32,] NA
## [33,] NA
## [34,] "a"
## [35,] "a"
## [36,] "a"
## [37,] NA
## [38,] NA
## [39,] "a"
## [40,] "a"
## [41,] NA
## [42,] NA
## [43,] "a"
## [44,] NA
## [45,] NA
## [46,] "a"
## [47,] NA
## [48,] "a"
## [49,] "a"
## [50,] NA
## [51,] "a"
## [52,] NA
## [53,] NA
## [54,] "a"
## [55,] "a"
## [56,] "a"
## [57,] "a"
## [58,] "a"
## [59,] "a"
## [60,] NA
## [61,] "a"
## [62,] "a"
## [63,] NA
## [64,] "a"
## [65,] NA
## [66,] "a"
## [67,] NA
## [68,] "a"
## [69,] "a"
## [70,] "a"
## [71,] "a"
## [72,] NA
## [73,] "a"
## [74,] "a"
## [75,] "a"
## [76,] "a"
## [77,] "a"
## [78,] "a"
## [79,] NA
## [80,] "a"
```

```
# Return fruit containing "a" all occurences
str_match_all(string = fruit, pattern = "a") #  matrix inside list is returned
```

```
## [[1]]
##      [,1]
## [1,] "a"
##
## [[2]]
##      [,1]
## [1,] "a"
##
## [[3]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[4]]
##      [,1]
## [1,] "a"
## [2,] "a"
## [3,] "a"
##
## [[5]]
##      [,1]
##
## [[6]]
##      [,1]
##
```

```
## [[7]]
##       [,1]
## [1,] "a"
##
## [[8]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[9]]
##      [,1]
## [1,] "a"
##
## [[10]]
##      [,1]
##
## [[11]]
##      [,1]
##
## [[12]]
##      [,1]
## [1,] "a"
##
## [[13]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[14]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[15]]
##      [,1]
## [1,] "a"
##
## [[16]]
##      [,1]
##
## [[17]]
##      [,1]
##
## [[18]]
##      [,1]
##
## [[19]]
##      [,1]
##
## [[20]]
##      [,1]
##
## [[21]]
##      [,1]
## [1,] "a"
##
## [[22]]
##      [,1]
##
## [[23]]
##      [,1]
## [1,] "a"
##
## [[24]]
##      [,1]
## [1,] "a"
##
## [[25]]
##      [,1]
## [1,] "a"
##
## [[26]]
##      [,1]
## [1,] "a"
##
## [[27]]
##      [,1]
## [1,] "a"
##
## [[28]]
##      [,1]
## [1,] "a"
##
## [[29]]
##      [,1]
##
## [[30]]
##      [,1]
## [1,] "a"
##
## [[31]]
##      [,1]
##
## [[32]]
##      [,1]
##
## [[33]]
##      [,1]
##
## [[34]]
##      [,1]
## [1,] "a"
##
## [[35]]
##      [,1]
## [1,] "a"
##
## [[36]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[37]]
##      [,1]
##
## [[38]]
##      [,1]
##
## [[39]]
##      [,1]
## [1,] "a"
##
## [[40]]
##      [,1]
## [1,] "a"
##
## [[41]]
##      [,1]
##
## [[42]]
##      [,1]
##
## [[43]]
##      [,1]
## [1,] "a"
##
## [[44]]
##      [,1]
##
## [[45]]
##      [,1]
##
## [[46]]
##      [,1]
## [1,] "a"
##
## [[47]]
##      [,1]
##
## [[48]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[49]]
##      [,1]
## [1,] "a"
##
## [[50]]
##      [,1]
##
## [[51]]
##      [,1]
## [1,] "a"
##
## [[52]]
##      [,1]
##
## [[53]]
##      [,1]
##
## [[54]]
##      [,1]
## [1,] "a"
##
## [[55]]
##      [,1]
## [1,] "a"
##
## [[56]]
##      [,1]
## [1,] "a"
## [2,] "a"
## [3,] "a"
##
## [[57]]
##      [,1]
## [1,] "a"
##
## [[58]]
##      [,1]
## [1,] "a"
##
## [[59]]
##      [,1]
## [1,] "a"
##
## [[60]]
##      [,1]
##
## [[61]]
##      [,1]
## [1,] "a"
##
## [[62]]
##      [,1]
## [1,] "a"
##
## [[63]]
##      [,1]
##
## [[64]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
```

```
## [[65]]
##      [,1]
##
## [[66]]
##      [,1]
## [1,] "a"
##
## [[67]]
##      [,1]
##
## [[68]]
##      [,1]
## [1,] "a"
##
## [[69]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[70]]
##      [,1]
## [1,] "a"
##
## [[71]]
##      [,1]
## [1,] "a"
##
## [[72]]
##      [,1]
##
## [[73]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[74]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[75]]
##      [,1]
## [1,] "a"
##
## [[76]]
##      [,1]
## [1,] "a"
##
## [[77]]
##      [,1]
## [1,] "a"
## [2,] "a"
##
## [[78]]
##      [,1]
## [1,] "a"
##
## [[79]]
##      [,1]
##
## [[80]]
##      [,1]
## [1,] "a"
```

String lengths

```r
# str_length() - Width of a string
# similar base R: nchar()
str_length("word")
```

```
## [1] 4
```

```r
nchar("word")
```

```
## [1] 4
```

```r
# Find all fruits with length 10 or more characters
fruit[str_length(fruit) >= 10]
```

```
##  [1] "bell pepper"       "blackberry"        "blackcurrant"
##  [4] "blood orange"      "boysenberry"       "breadfruit"
##  [7] "canary melon"      "cantaloupe"        "chili pepper"
## [10] "clementine"        "cloudberry"        "dragonfruit"
## [13] "elderberry"        "goji berry"        "gooseberry"
## [16] "grapefruit"        "huckleberry"       "kiwi fruit"
## [19] "passionfruit"      "pomegranate"       "purple mangosteen"
## [22] "redcurrant"        "rock melon"        "salal berry"
## [25] "star fruit"        "strawberry"        "ugli fruit"
## [28] "watermelon"
```

```r
# str_pad() - String padding
# Pad fruit names with symbol "X" to get a string with width = 20
str_pad(string = fruit, width = 20, side = "left", pad = "X")  # left side padding
```

```
##  [1] "XXXXXXXXXXXXXXXapple" "XXXXXXXXXXXXXapricot" "XXXXXXXXXXXXXavocado"
##  [4] "XXXXXXXXXXXXXXbanana" "XXXXXXXXXbell pepper" "XXXXXXXXXXXXbilberry"
##  [7] "XXXXXXXXXXblackberry" "XXXXXXXXblackcurrant" "XXXXXXXXblood orange"
## [10] "XXXXXXXXXXXblueberry" "XXXXXXXXXboysenberry" "XXXXXXXXXXbreadfruit"
## [13] "XXXXXXXXcanary melon" "XXXXXXXXXXcantaloupe" "XXXXXXXXXXXcherimoya"
## [16] "XXXXXXXXXXXXXXcherry" "XXXXXXXXchili pepper" "XXXXXXXXXXclementine"
## [19] "XXXXXXXXXXcloudberry" "XXXXXXXXXXXXXcoconut" "XXXXXXXXXXXcranberry"
## [22] "XXXXXXXXXXXXcucumber" "XXXXXXXXXXXXXcurrant" "XXXXXXXXXXXXXXdamson"
## [25] "XXXXXXXXXXXXXXXXdate" "XXXXXXXXXdragonfruit" "XXXXXXXXXXXXXXdurian"
## [28] "XXXXXXXXXXXXeggplant" "XXXXXXXXXXelderberry" "XXXXXXXXXXXXXXfeijoa"
## [31] "XXXXXXXXXXXXXXXXXfig" "XXXXXXXXXXgoji berry" "XXXXXXXXXXgooseberry"
## [34] "XXXXXXXXXXXXXXXgrape" "XXXXXXXXXXgrapefruit" "XXXXXXXXXXXXXXXguava"
## [37] "XXXXXXXXXXXXhoneydew" "XXXXXXXXXhuckleberry" "XXXXXXXXXXXjackfruit"
## [40] "XXXXXXXXXXXXXXjambul" "XXXXXXXXXXXXXXjujube" "XXXXXXXXXXkiwi fruit"
## [43] "XXXXXXXXXXXXXkumquat" "XXXXXXXXXXXXXXXlemon" "XXXXXXXXXXXXXXXXlime"
## [46] "XXXXXXXXXXXXXXloquat" "XXXXXXXXXXXXXXlychee" "XXXXXXXXXXXmandarine"
## [49] "XXXXXXXXXXXXXXXmango" "XXXXXXXXXXXXmulberry" "XXXXXXXXXXXnectarine"
## [52] "XXXXXXXXXXXXXXXXXnut" "XXXXXXXXXXXXXXXolive" "XXXXXXXXXXXXXXorange"
## [55] "XXXXXXXXXXXXXXpamelo" "XXXXXXXXXXXXXXpapaya" "XXXXXXXXpassionfruit"
## [58] "XXXXXXXXXXXXXXXpeach" "XXXXXXXXXXXXXXXXpear" "XXXXXXXXXXXpersimmon"
## [61] "XXXXXXXXXXXXphysalis" "XXXXXXXXXXXpineapple" "XXXXXXXXXXXXXXXXplum"
## [64] "XXXXXXXXXpomegranate" "XXXXXXXXXXXXXXpomelo" "XXXpurple mangosteen"
## [67] "XXXXXXXXXXXXXXquince" "XXXXXXXXXXXXXXraisin" "XXXXXXXXXXXXrambutan"
## [70] "XXXXXXXXXXXraspberry" "XXXXXXXXXXredcurrant" "XXXXXXXXXXrock melon"
## [73] "XXXXXXXXXsalal berry" "XXXXXXXXXXXXXsatsuma" "XXXXXXXXXXstar fruit"
## [76] "XXXXXXXXXXstrawberry" "XXXXXXXXXXXtamarillo" "XXXXXXXXXXXtangerine"
## [79] "XXXXXXXXXXugli fruit" "XXXXXXXXXXwatermelon"
```

```
str_pad(string = fruit, width = 20, side = "right", pad = "X") # right side padding
```

```
##  [1] "appleXXXXXXXXXXXXXXX" "apricotXXXXXXXXXXXXX" "avocadoXXXXXXXXXXXXX"
##  [4] "bananaXXXXXXXXXXXXXX" "bell pepperXXXXXXXXX" "bilberryXXXXXXXXXXXX"
##  [7] "blackberryXXXXXXXXXX" "blackcurrantXXXXXXXX" "blood orangeXXXXXXXX"
## [10] "blueberryXXXXXXXXXXX" "boysenberryXXXXXXXXX" "breadfruitXXXXXXXXXX"
## [13] "canary melonXXXXXXXX" "cantaloupeXXXXXXXXXX" "cherimoyaXXXXXXXXXXX"
## [16] "cherryXXXXXXXXXXXXXX" "chili pepperXXXXXXXX" "clementineXXXXXXXXXX"
## [19] "cloudberryXXXXXXXXXX" "coconutXXXXXXXXXXXXX" "cranberryXXXXXXXXXXX"
## [22] "cucumberXXXXXXXXXXXX" "currantXXXXXXXXXXXXX" "damsonXXXXXXXXXXXXXX"
## [25] "dateXXXXXXXXXXXXXXXX" "dragonfruitXXXXXXXXX" "durianXXXXXXXXXXXXXX"
## [28] "eggplantXXXXXXXXXXXX" "elderberryXXXXXXXXXX" "feijoaXXXXXXXXXXXXXX"
## [31] "figXXXXXXXXXXXXXXXXX" "goji berryXXXXXXXXXX" "gooseberryXXXXXXXXXX"
## [34] "grapeXXXXXXXXXXXXXXX" "grapefruitXXXXXXXXXX" "guavaXXXXXXXXXXXXXXX"
## [37] "honeydewXXXXXXXXXXXX" "huckleberryXXXXXXXXX" "jackfruitXXXXXXXXXXX"
## [40] "jambulXXXXXXXXXXXXXX" "jujubeXXXXXXXXXXXXXX" "kiwi fruitXXXXXXXXXX"
## [43] "kumquatXXXXXXXXXXXXX" "lemonXXXXXXXXXXXXXXX" "limeXXXXXXXXXXXXXXXX"
## [46] "loquatXXXXXXXXXXXXXX" "lycheeXXXXXXXXXXXXXX" "mandarineXXXXXXXXXXX"
## [49] "mangoXXXXXXXXXXXXXXX" "mulberryXXXXXXXXXXXX" "nectarineXXXXXXXXXXX"
## [52] "nutXXXXXXXXXXXXXXXXX" "oliveXXXXXXXXXXXXXXX" "orangeXXXXXXXXXXXXXX"
## [55] "pameloXXXXXXXXXXXXXX" "papayaXXXXXXXXXXXXXX" "passionfruitXXXXXXXX"
## [58] "peachXXXXXXXXXXXXXXX" "pearXXXXXXXXXXXXXXXX" "persimmonXXXXXXXXXXX"
## [61] "physalisXXXXXXXXXXXX" "pineappleXXXXXXXXXXX" "plumXXXXXXXXXXXXXXXX"
## [64] "pomegranateXXXXXXXXX" "pomeloXXXXXXXXXXXXXX" "purple mangosteenXXX"
## [67] "quinceXXXXXXXXXXXXXX" "raisinXXXXXXXXXXXXXX" "rambutanXXXXXXXXXXXX"
## [70] "raspberryXXXXXXXXXXX" "redcurrantXXXXXXXXXX" "rock melonXXXXXXXXXX"
## [73] "salal berryXXXXXXXXX" "satsumaXXXXXXXXXXXXX" "star fruitXXXXXXXXXX"
## [76] "strawberryXXXXXXXXXX" "tamarilloXXXXXXXXXXX" "tangerineXXXXXXXXXXX"
## [79] "ugli fruitXXXXXXXXXX" "watermelonXXXXXXXXXX"
```

```
str_pad(string = fruit, width = 20, side = "both", pad = "X")  # both side padding
```

```
##  [1] "XXXXXXXappleXXXXXXXX" "XXXXXXapricotXXXXXXX" "XXXXXXavocadoXXXXXXX"
##  [4] "XXXXXXXbananaXXXXXXX" "XXXXbell pepperXXXXX" "XXXXXXbilberryXXXXXX"
##  [7] "XXXXXblackberryXXXXX" "XXXXblackcurrantXXXX" "XXXXblood orangeXXXX"
## [10] "XXXXXblueberryXXXXXX" "XXXXboysenberryXXXXX" "XXXXXbreadfruitXXXXX"
## [13] "XXXXcanary melonXXXX" "XXXXXcantaloupeXXXXX" "XXXXXcherimoyaXXXXXX"
## [16] "XXXXXXXcherryXXXXXXX" "XXXXchili pepperXXXX" "XXXXXclementineXXXXX"
## [19] "XXXXXcloudberryXXXXX" "XXXXXXcoconutXXXXXXX" "XXXXXcranberryXXXXXX"
## [22] "XXXXXXcucumberXXXXXX" "XXXXXXcurrantXXXXXXX" "XXXXXXXdamsonXXXXXXX"
## [25] "XXXXXXXXdateXXXXXXXX" "XXXXdragonfruitXXXXX" "XXXXXXXdurianXXXXXXX"
## [28] "XXXXXXeggplantXXXXXX" "XXXXXelderberryXXXXX" "XXXXXXXfeijoaXXXXXXX"
## [31] "XXXXXXXXfigXXXXXXXXX" "XXXXXgoji berryXXXXX" "XXXXXgooseberryXXXXX"
## [34] "XXXXXXXgrapeXXXXXXXX" "XXXXXgrapefruitXXXXX" "XXXXXXXguavaXXXXXXXX"
## [37] "XXXXXXhoneydewXXXXXX" "XXXXhuckleberryXXXXX" "XXXXXjackfruitXXXXXX"
## [40] "XXXXXXXjambulXXXXXXX" "XXXXXXXjujubeXXXXXXX" "XXXXXkiwi fruitXXXXX"
## [43] "XXXXXXkumquatXXXXXXX" "XXXXXXXlemonXXXXXXXX" "XXXXXXXXlimeXXXXXXXX"
## [46] "XXXXXXXloquatXXXXXXX" "XXXXXXXlycheeXXXXXXX" "XXXXXmandarineXXXXXX"
## [49] "XXXXXXXmangoXXXXXXXX" "XXXXXXmulberryXXXXXX" "XXXXXnectarineXXXXXX"
## [52] "XXXXXXXXnutXXXXXXXXX" "XXXXXXXoliveXXXXXXXX" "XXXXXXXorangeXXXXXXX"
## [55] "XXXXXXXpameloXXXXXXX" "XXXXXXXpapayaXXXXXXX" "XXXXpassionfruitXXXX"
## [58] "XXXXXXXpeachXXXXXXXX" "XXXXXXXXpearXXXXXXXX" "XXXXXpersimmonXXXXXX"
## [61] "XXXXXXphysalisXXXXXX" "XXXXXpineappleXXXXXX" "XXXXXXXXplumXXXXXXXX"
## [64] "XXXXpomegranateXXXXX" "XXXXXXXpomeloXXXXXXX" "Xpurple mangosteenXX"
## [67] "XXXXXXXquinceXXXXXXX" "XXXXXXXraisinXXXXXXX" "XXXXXXrambutanXXXXXX"
## [70] "XXXXXraspberryXXXXXX" "XXXXXredcurrantXXXXX" "XXXXXrock melonXXXXX"
## [73] "XXXXsalal berryXXXXX" "XXXXXXsatsumaXXXXXXX" "XXXXXstar fruitXXXXX"
## [76] "XXXXXstrawberryXXXXX" "XXXXXtamarilloXXXXXX" "XXXXXtangerineXXXXXX"
## [79] "XXXXXugli fruitXXXXX" "XXXXXwatermelonXXXXX"
```

```
# Where padding is very useful in practice (ID numbers)
set.seed(123)
id.numbers <- sample(x = 1:1000, size = 25, replace = F) # generate some ID numbers
id.numbers
```

```
##  [1] 415 463 179 526 195 938 818 118 299 229 244  14 374 665 602 603 768 709  91
## [20] 953 348 649 355 840  26
```

```
str_pad(id.numbers, width = 5, side = "left", pad = "0") # add leading zeros
```

```
##  [1] "00415" "00463" "00179" "00526" "00195" "00938" "00818" "00118" "00299"
## [10] "00229" "00244" "00014" "00374" "00665" "00602" "00603" "00768" "00709"
## [19] "00091" "00953" "00348" "00649" "00355" "00840" "00026"
```

```
# str_trunc() - String truncating
# Truncate fruit names with symbol "..." to get a string with width = 5
str_trunc(string = fruit, width = 5, side = "left", ellipsis = "...")   # left side truncating
```

```
##  [1] "apple" "...ot" "...do" "...na" "...er" "...ry" "...ry" "...nt" "...ge"
## [10] "...ry" "...ry" "...it" "...on" "...pe" "...ya" "...ry" "...er" "...ne"
## [19] "...ry" "...ut" "...ry" "...er" "...nt" "...on" "date"  "...it" "...an"
## [28] "...nt" "...ry" "...oa" "fig"   "...ry" "...ry" "grape" "...it" "guava"
## [37] "...ew" "...ry" "...it" "...ul" "...be" "...it" "...at" "lemon" "lime"
## [46] "...at" "...ee" "...ne" "mango" "...ry" "...ne" "nut"   "olive" "...ge"
## [55] "...lo" "...ya" "...it" "peach" "pear"  "...on" "...is" "...le" "plum"
## [64] "...te" "...lo" "...en" "...ce" "...in" "...an" "...ry" "...nt" "...on"
## [73] "...ry" "...ma" "...it" "...ry" "...lo" "...ne" "...it" "...on"
```

```
str_trunc(string = fruit, width = 5, side = "right", ellipsis = "...")  # right side truncating
```

```
##  [1] "apple" "ap..." "av..." "ba..." "be..." "bi..." "bl..." "bl..." "bl..."
## [10] "bl..." "bo..." "br..." "ca..." "ca..." "ch..." "ch..." "ch..." "cl..."
## [19] "cl..." "co..." "cr..." "cu..." "cu..." "da..." "date"  "dr..." "du..."
## [28] "eg..." "el..." "fe..." "fig"   "go..." "go..." "grape" "gr..." "guava"
## [37] "ho..." "hu..." "ja..." "ja..." "ju..." "ki..." "ku..." "lemon" "lime"
## [46] "lo..." "ly..." "ma..." "mango" "mu..." "ne..." "nut"   "olive" "or..."
## [55] "pa..." "pa..." "pa..." "peach" "pear"  "pe..." "ph..." "pi..." "plum"
## [64] "po..." "po..." "pu..." "qu..." "ra..." "ra..." "ra..." "re..." "ro..."
## [73] "sa..." "sa..." "st..." "st..." "ta..." "ta..." "ug..." "wa..."
```

```r
str_trunc(string = fruit, width = 5, side = "center", ellipsis = "...") # center side truncating
```

```
##  [1] "apple" "a...t" "a...o" "b...a" "b...r" "b...y" "b...y" "b...t" "b...e"
## [10] "b...y" "b...y" "b...t" "c...n" "c...e" "c...a" "c...y" "c...r" "c...e"
## [19] "c...y" "c...t" "c...y" "c...r" "c...t" "d...n" "date"  "d...t" "d...n"
## [28] "e...t" "e...y" "f...a" "fig"   "g...y" "g...y" "grape" "g...t" "guava"
## [37] "h...w" "h...y" "j...t" "j...l" "j...e" "k...t" "k...t" "lemon" "lime"
## [46] "l...t" "l...e" "m...e" "mango" "m...y" "n...e" "nut"   "olive" "o...e"
## [55] "p...o" "p...a" "p...t" "peach" "pear"  "p...n" "p...s" "p...e" "plum"
## [64] "p...e" "p...o" "p...n" "q...e" "r...n" "r...n" "r...y" "r...t" "r...n"
## [73] "s...y" "s...a" "s...t" "s...y" "t...o" "t...e" "u...t" "w...n"
```

```r
# str_trim() - Trim whitespaces
# Create a string with white spaces
whitespace <- c("nospaces",
                " leftspace",
                "      leftspaces",
                "rightspace ",
                "rightspaces     ",
                " bothspace ",
                "   bothspaces   ",
                "middle space",
                " mix space ")
whitespace
```

```
## [1] "nospaces"         " leftspace"       "      leftspaces" "rightspace "
## [5] "rightspaces     " " bothspace "      "   bothspaces   " "middle space"
## [9] " mix space "
```

```r
# Trim left white space(s)
whitespace.trim.left <- str_trim(string = whitespace, side = "left")
whitespace
```

```
## [1] "nospaces"         " leftspace"       "      leftspaces" "rightspace "
## [5] "rightspaces     " " bothspace "      "   bothspaces   " "middle space"
## [9] " mix space "
```

```r
whitespace.trim.left
```

```
## [1] "nospaces"         "leftspace"        "leftspaces"       "rightspace "
## [5] "rightspaces     " "bothspace "       "bothspaces   "    "middle space"
## [9] "mix space "
```

```r
# Trim right white space(s)
whitespace.trim.right <- str_trim(string = whitespace, side = "right")
whitespace
```

```
## [1] "nospaces"         " leftspace"       "      leftspaces" "rightspace "
## [5] "rightspaces     " " bothspace "      "   bothspaces   " "middle space"
## [9] " mix space "
```

```r
whitespace.trim.right
```

```
## [1] "nospaces"         " leftspace"       "      leftspaces" "rightspace"
## [5] "rightspaces"      " bothspace"       "   bothspaces"    "middle space"
## [9] " mix space"
```

```
# Trim both side white space(s)
whitespace.trim.both <- str_trim(string = whitespace, side = "both")
whitespace
```

```
## [1] "nospaces"         " leftspace"       "      leftspaces" "rightspace "
## [5] "rightspaces     " " bothspace "      "   bothspaces   " "middle space"
## [9] " mix space "
```

```
whitespace.trim.both
```

```
## [1] "nospaces"     "leftspace"    "leftspaces"   "rightspace"   "rightspaces"
## [6] "bothspace"    "bothspaces"   "middle space" "mix space"
```

Strings mutating

```
# str_sub() - Replace a part of given string
# Replace first 3 letters of each fruit with string "FRU"
fruit
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "bell pepper"       "bilberry"
##  [7] "blackberry"        "blackcurrant"      "blood orange"
## [10] "blueberry"         "boysenberry"       "breadfruit"
## [13] "canary melon"      "cantaloupe"        "cherimoya"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "cranberry"
## [22] "cucumber"          "currant"           "damson"
## [25] "date"              "dragonfruit"       "durian"
## [28] "eggplant"          "elderberry"        "feijoa"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grape"             "grapefruit"        "guava"
## [37] "honeydew"          "huckleberry"       "jackfruit"
## [40] "jambul"            "jujube"            "kiwi fruit"
## [43] "kumquat"           "lemon"             "lime"
## [46] "loquat"            "lychee"            "mandarine"
## [49] "mango"             "mulberry"          "nectarine"
## [52] "nut"               "olive"             "orange"
## [55] "pamelo"            "papaya"            "passionfruit"
## [58] "peach"             "pear"              "persimmon"
## [61] "physalis"          "pineapple"         "plum"
## [64] "pomegranate"       "pomelo"            "purple mangosteen"
## [67] "quince"            "raisin"            "rambutan"
## [70] "raspberry"         "redcurrant"        "rock melon"
## [73] "salal berry"       "satsuma"           "star fruit"
## [76] "strawberry"        "tamarillo"         "tangerine"
## [79] "ugli fruit"        "watermelon"
```

```
fruit.sub <- fruit
fruit.sub
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "bell pepper"       "bilberry"
##  [7] "blackberry"        "blackcurrant"      "blood orange"
## [10] "blueberry"         "boysenberry"       "breadfruit"
## [13] "canary melon"      "cantaloupe"        "cherimoya"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "cranberry"
## [22] "cucumber"          "currant"           "damson"
## [25] "date"              "dragonfruit"       "durian"
## [28] "eggplant"          "elderberry"        "feijoa"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grape"             "grapefruit"        "guava"
## [37] "honeydew"          "huckleberry"       "jackfruit"
## [40] "jambul"            "jujube"            "kiwi fruit"
## [43] "kumquat"           "lemon"             "lime"
## [46] "loquat"            "lychee"            "mandarine"
## [49] "mango"             "mulberry"          "nectarine"
## [52] "nut"               "olive"             "orange"
## [55] "pamelo"            "papaya"            "passionfruit"
## [58] "peach"             "pear"              "persimmon"
## [61] "physalis"          "pineapple"         "plum"
## [64] "pomegranate"       "pomelo"            "purple mangosteen"
## [67] "quince"            "raisin"            "rambutan"
## [70] "raspberry"         "redcurrant"        "rock melon"
## [73] "salal berry"       "satsuma"           "star fruit"
## [76] "strawberry"        "tamarillo"         "tangerine"
## [79] "ugli fruit"        "watermelon"
```

```
str_sub(fruit.sub, start = 1, end = 3) <- "FRU"
fruit.sub
```

```
##  [1] "FRUle"             "FRUicot"           "FRUcado"
##  [4] "FRUana"            "FRUl pepper"       "FRUberry"
##  [7] "FRUckberry"        "FRUckcurrant"      "FRUod orange"
## [10] "FRUeberry"         "FRUsenberry"       "FRUadfruit"
## [13] "FRUary melon"      "FRUtaloupe"        "FRUrimoya"
## [16] "FRUrry"            "FRUli pepper"      "FRUmentine"
## [19] "FRUudberry"        "FRUonut"           "FRUnberry"
## [22] "FRUumber"          "FRUrant"           "FRUson"
## [25] "FRUe"              "FRUgonfruit"       "FRUian"
## [28] "FRUplant"          "FRUerberry"        "FRUjoa"
## [31] "FRU"               "FRUi berry"        "FRUseberry"
## [34] "FRUpe"             "FRUpefruit"        "FRUva"
## [37] "FRUeydew"          "FRUkleberry"       "FRUkfruit"
## [40] "FRUbul"            "FRUube"            "FRUi fruit"
## [43] "FRUquat"           "FRUon"             "FRUe"
## [46] "FRUuat"            "FRUhee"            "FRUdarine"
## [49] "FRUgo"             "FRUberry"          "FRUtarine"
## [52] "FRU"               "FRUve"             "FRUnge"
## [55] "FRUelo"            "FRUaya"            "FRUsionfruit"
## [58] "FRUch"             "FRUr"              "FRUsimmon"
## [61] "FRUsalis"          "FRUeapple"         "FRUm"
## [64] "FRUegranate"       "FRUelo"            "FRUple mangosteen"
## [67] "FRUnce"            "FRUsin"            "FRUbutan"
## [70] "FRUpberry"         "FRUcurrant"        "FRUk melon"
## [73] "FRUal berry"       "FRUsuma"           "FRUr fruit"
## [76] "FRUawberry"        "FRUarillo"         "FRUgerine"
## [79] "FRUi fruit"        "FRUermelon"
```

```
# str_replace() - Replace the first matched pattern in a string
# Replace first occurence of letter "a" with "A" in each fruit
```

```r
str_replace(string = fruit, pattern = "a", replacement = "A")
```

```
##  [1] "Apple"             "Apricot"           "Avocado"
##  [4] "bAnana"            "bell pepper"       "bilberry"
##  [7] "blAckberry"        "blAckcurrant"      "blood orAnge"
## [10] "blueberry"         "boysenberry"       "breAdfruit"
## [13] "cAnary melon"      "cAntaloupe"        "cherimoyA"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "crAnberry"
## [22] "cucumber"          "currAnt"           "dAmson"
## [25] "dAte"              "drAgonfruit"       "duriAn"
## [28] "eggplAnt"          "elderberry"        "feijoA"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grApe"             "grApefruit"        "guAva"
## [37] "honeydew"          "huckleberry"       "jAckfruit"
## [40] "jAmbul"            "jujube"            "kiwi fruit"
## [43] "kumquAt"           "lemon"             "lime"
## [46] "loquAt"            "lychee"            "mAndarine"
## [49] "mAngo"             "mulberry"          "nectArine"
## [52] "nut"               "olive"             "orAnge"
## [55] "pAmelo"            "pApaya"            "pAssionfruit"
## [58] "peAch"             "peAr"              "persimmon"
## [61] "physAlis"          "pineApple"         "plum"
## [64] "pomegrAnate"       "pomelo"            "purple mAngosteen"
## [67] "quince"            "rAisin"            "rAmbutan"
## [70] "rAspberry"         "redcurrAnt"        "rock melon"
## [73] "sAlal berry"       "sAtsuma"           "stAr fruit"
## [76] "strAwberry"        "tAmarillo"         "tAngerine"
## [79] "ugli fruit"        "wAtermelon"
```

```r
# str_replace_all() - Replace all matched patterns in a string
# Replace all occurences of letter "a" with "A" in each fruit
str_replace_all(string = fruit, pattern = "a", replacement = "A")
```

```
##  [1] "Apple"             "Apricot"           "AvocAdo"
##  [4] "bAnAnA"            "bell pepper"       "bilberry"
##  [7] "blAckberry"        "blAckcurrAnt"      "blood orAnge"
## [10] "blueberry"         "boysenberry"       "breAdfruit"
## [13] "cAnAry melon"      "cAntAloupe"        "cherimoyA"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "crAnberry"
## [22] "cucumber"          "currAnt"           "dAmson"
## [25] "dAte"              "drAgonfruit"       "duriAn"
## [28] "eggplAnt"          "elderberry"        "feijoA"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grApe"             "grApefruit"        "guAvA"
## [37] "honeydew"          "huckleberry"       "jAckfruit"
## [40] "jAmbul"            "jujube"            "kiwi fruit"
## [43] "kumquAt"           "lemon"             "lime"
## [46] "loquAt"            "lychee"            "mAndArine"
## [49] "mAngo"             "mulberry"          "nectArine"
## [52] "nut"               "olive"             "orAnge"
## [55] "pAmelo"            "pApAyA"            "pAssionfruit"
## [58] "peAch"             "peAr"              "persimmon"
## [61] "physAlis"          "pineApple"         "plum"
## [64] "pomegrAnAte"       "pomelo"            "purple mAngosteen"
## [67] "quince"            "rAisin"            "rAmbutAn"
## [70] "rAspberry"         "redcurrAnt"        "rock melon"
## [73] "sAlAl berry"       "sAtsumA"           "stAr fruit"
## [76] "strAwberry"        "tAmArillo"         "tAngerine"
## [79] "ugli fruit"        "wAtermelon"
```

```
# str_to_lower() - Convert string to lower case
string.upper <- "THIS IS A STRING"
string.lower <- str_to_lower(string = string.upper)
string.lower
```

```
## [1] "this is a string"
```

```
# str_to_upper() - Convert string to upper case
string.upper <- str_to_lower(string = string.lower)
string.upper
```

```
## [1] "this is a string"
```

```
# str_to_title() - Convert string to "upper"title" case
string.title <- str_to_title(string = string.lower)
string.title
```

```
## [1] "This Is A String"
```

Joining and splitting strings

```
# str_c() - Join multiple strings into a single string
# Let's split vector "fruit" into 4 equal in size smaller vectors
fruit1 <- fruit[1:20]
fruit2 <- fruit[21:40]
fruit3 <- fruit[41:60]
fruit4 <- fruit[61:80]

# Create one vector of strings using all 4 smaller vectors
str_c(fruit1, fruit2, fruit3, fruit4, sep = "-")
```

```
##  [1] "apple-cranberry-jujube-physalis"
##  [2] "apricot-cucumber-kiwi fruit-pineapple"
##  [3] "avocado-currant-kumquat-plum"
##  [4] "banana-damson-lemon-pomegranate"
##  [5] "bell pepper-date-lime-pomelo"
##  [6] "bilberry-dragonfruit-loquat-purple mangosteen"
##  [7] "blackberry-durian-lychee-quince"
##  [8] "blackcurrant-eggplant-mandarine-raisin"
##  [9] "blood orange-elderberry-mango-rambutan"
## [10] "blueberry-feijoa-mulberry-raspberry"
## [11] "boysenberry-fig-nectarine-redcurrant"
## [12] "breadfruit-goji berry-nut-rock melon"
## [13] "canary melon-gooseberry-olive-salal berry"
## [14] "cantaloupe-grape-orange-satsuma"
## [15] "cherimoya-grapefruit-pamelo-star fruit"
## [16] "cherry-guava-papaya-strawberry"
## [17] "chili pepper-honeydew-passionfruit-tamarillo"
## [18] "clementine-huckleberry-peach-tangerine"
## [19] "cloudberry-jackfruit-pear-ugli fruit"
## [20] "coconut-jambul-persimmon-watermelon"
```

```
# Create vector of alphabet letters: one lower and one upper case
letters
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
Letters
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
str_c(letters, Letters)
```

```
##  [1] "aA" "bB" "cC" "dD" "eE" "fF" "gG" "hH" "iI" "jJ" "kK" "lL" "mM" "nN" "oO"
## [16] "pP" "qQ" "rR" "sS" "tT" "uU" "vV" "wW" "xX" "yY" "zZ"
```

```
# str_c() - Colapse a vector of strings into  single string
# Collapse a vector of letters into a single string containing all letters
str_c(letters, collapse = "")
```

```
## [1] "abcdefghijklmnopqrstuvwxyz"
```

```
str_c(letters, collapse = " ")
```

```
## [1] "a b c d e f g h i j k l m n o p q r s t u v w x y z"
```

```
# str_dup() - Repeat a string multiple times
# Repeat one string 5 times
str_dup(string = "string", times = 5)
```

```
## [1] "stringstringstringstringstring"
```

```
# Repeat a vector of strings 2 times
str_dup(string = fruit1, times = 2)
```

```
##  [1] "appleapple"                "apricotapricot"
##  [3] "avocadoavocado"            "bananabanana"
##  [5] "bell pepperbell pepper"    "bilberrybilberry"
##  [7] "blackberryblackberry"      "blackcurrantblackcurrant"
##  [9] "blood orangeblood orange" "blueberryblueberry"
## [11] "boysenberryboysenberry"    "breadfruitbreadfruit"
## [13] "canary meloncanary melon" "cantaloupecantaloupe"
## [15] "cherimoyacherimoya"        "cherrycherry"
## [17] "chili pepperchili pepper" "clementineclementine"
## [19] "cloudberrycloudberry"      "coconutcoconut"
```

```
# str_split_fixed() - Split a vector of strings into a matrix of substrings base on pattern
# Split fruit by " " white space
str_split_fixed(string = fruit, pattern = " ", n = 2) # n - number of pieces to return!
```

```
##       [,1]           [,2]
##  [1,] "apple"        ""
##  [2,] "apricot"      ""
##  [3,] "avocado"      ""
##  [4,] "banana"       ""
##  [5,] "bell"         "pepper"
##  [6,] "bilberry"     ""
##  [7,] "blackberry"   ""
##  [8,] "blackcurrant" ""
##  [9,] "blood"        "orange"
## [10,] "blueberry"    ""
## [11,] "boysenberry"  ""
## [12,] "breadfruit"   ""
## [13,] "canary"       "melon"
## [14,] "cantaloupe"   ""
## [15,] "cherimoya"    ""
## [16,] "cherry"       ""
## [17,] "chili"        "pepper"
## [18,] "clementine"   ""
## [19,] "cloudberry"   ""
## [20,] "coconut"      ""
## [21,] "cranberry"    ""
## [22,] "cucumber"     ""
## [23,] "currant"      ""
## [24,] "damson"       ""
## [25,] "date"         ""
## [26,] "dragonfruit"  ""
## [27,] "durian"       ""
## [28,] "eggplant"     ""
## [29,] "elderberry"   ""
## [30,] "feijoa"       ""
## [31,] "fig"          ""
## [32,] "goji"         "berry"
## [33,] "gooseberry"   ""
## [34,] "grape"        ""
## [35,] "grapefruit"   ""
## [36,] "guava"        ""
## [37,] "honeydew"     ""
## [38,] "huckleberry"  ""
## [39,] "jackfruit"    ""
## [40,] "jambul"       ""
## [41,] "jujube"       ""
## [42,] "kiwi"         "fruit"
## [43,] "kumquat"      ""
## [44,] "lemon"        ""
## [45,] "lime"         ""
## [46,] "loquat"       ""
## [47,] "lychee"       ""
## [48,] "mandarine"    ""
## [49,] "mango"        ""
## [50,] "mulberry"     ""
## [51,] "nectarine"    ""
## [52,] "nut"          ""
## [53,] "olive"        ""
## [54,] "orange"       ""
## [55,] "pamelo"       ""
## [56,] "papaya"       ""
## [57,] "passionfruit" ""
## [58,] "peach"        ""
## [59,] "pear"         ""
## [60,] "persimmon"    ""
## [61,] "physalis"     ""
## [62,] "pineapple"    ""
## [63,] "plum"         ""
## [64,] "pomegranate"  ""
## [65,] "pomelo"       ""
## [66,] "purple"       "mangosteen"
## [67,] "quince"       ""
## [68,] "raisin"       ""
## [69,] "rambutan"     ""
## [70,] "raspberry"    ""
## [71,] "redcurrant"   ""
## [72,] "rock"         "melon"
## [73,] "salal"        "berry"
## [74,] "satsuma"      ""
## [75,] "star"         "fruit"
## [76,] "strawberry"   ""
## [77,] "tamarillo"    ""
## [78,] "tangerine"    ""
## [79,] "ugli"         "fruit"
## [80,] "watermelon"   ""
```

```
# Split first 5 sentences by " " white space - increase n
str_split_fixed(sentences[1:5], pattern = " ", n = 10)
```

```
##      [,1]    [,2]    [,3]    [,4]      [,5]  [,6]    [,7]     [,8]
## [1,] "The"   "birch" "canoe" "slid"    "on"  "the"   "smooth" "planks."
## [2,] "Glue"  "the"   "sheet" "to"      "the" "dark"  "blue"   "background."
## [3,] "It's"  "easy"  "to"    "tell"    "the" "depth" "of"     "a"
## [4,] "These" "days"  "a"     "chicken" "leg" "is"    "a"      "rare"
## [5,] "Rice"  "is"    "often" "served"  "in"  "round" "bowls." ""
##      [,9]    [,10]
## [1,] ""      ""
## [2,] ""      ""
## [3,] "well." ""
## [4,] "dish." ""
## [5,] ""      ""
```

```
# str_split() - Split a vector of strings into a list / matrix of substrings base on pattern
# Split first 5 sentences by " " white space
str_split(sentences[1:5], pattern = " ") # return a list
```

```
## [[1]]
## [1] "The"     "birch"   "canoe"   "slid"    "on"      "the"     "smooth"
## [8] "planks."
##
## [[2]]
## [1] "Glue"        "the"         "sheet"       "to"          "the"
## [6] "dark"        "blue"        "background."
##
## [[3]]
## [1] "It's"  "easy"  "to"    "tell"  "the"   "depth" "of"    "a"     "well."
##
## [[4]]
## [1] "These"   "days"    "a"       "chicken" "leg"     "is"      "a"
## [8] "rare"    "dish."
##
## [[5]]
## [1] "Rice"   "is"     "often"  "served" "in"     "round"  "bowls."
```

```r
str_split(sentences[1:5], pattern = " ", simplify = T) # return a matrix
```

```
##      [,1]    [,2]    [,3]    [,4]      [,5]   [,6]    [,7]     [,8]
## [1,] "The"   "birch" "canoe" "slid"    "on"   "the"   "smooth" "planks."
## [2,] "Glue"  "the"   "sheet" "to"      "the"  "dark"  "blue"   "background."
## [3,] "It's"  "easy"  "to"    "tell"    "the"  "depth" "of"     "a"
## [4,] "These" "days"  "a"     "chicken" "leg"  "is"    "a"      "rare"
## [5,] "Rice"  "is"    "often" "served"  "in"   "round" "bowls." ""
##      [,9]
## [1,] ""
## [2,] ""
## [3,] "well."
## [4,] "dish."
## [5,] ""
```

```r
# str_glue() - Glue/merge together string and expression
# Merge string and evaluated mathematical symbol
str_glue("What is the value of sqrt(2), it is {sqrt(2)}.")
```

```
## What is the value of sqrt(2), it is 1.4142135623731.
```

```r
# Merge fixed string and assigned string to a variable
name <- "Marko"
str_glue("Hi my name is {name}")
```

```
## Hi my name is Marko
```

```r
# str_glue_data() - Use data.frame / list or environment to create strings from string and expression
# Merge string and values from a data.frame
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
str_glue_data(mtcars, "The car {rownames(mtcars)}: {hp} horsepower, {cyl} number of cylinders and consu
```

```
## The car Mazda RX4: 110 horsepower, 6 number of cylinders and consumption 21 miles per gallon
## The car Mazda RX4 Wag: 110 horsepower, 6 number of cylinders and consumption 21 miles per gallon
## The car Datsun 710: 93 horsepower, 4 number of cylinders and consumption 22.8 miles per gallon
## The car Hornet 4 Drive: 110 horsepower, 6 number of cylinders and consumption 21.4 miles per gallon
## The car Hornet Sportabout: 175 horsepower, 8 number of cylinders and consumption 18.7 miles per gall
## The car Valiant: 105 horsepower, 6 number of cylinders and consumption 18.1 miles per gallon
## The car Duster 360: 245 horsepower, 8 number of cylinders and consumption 14.3 miles per gallon
## The car Merc 240D: 62 horsepower, 4 number of cylinders and consumption 24.4 miles per gallon
## The car Merc 230: 95 horsepower, 4 number of cylinders and consumption 22.8 miles per gallon
## The car Merc 280: 123 horsepower, 6 number of cylinders and consumption 19.2 miles per gallon
## The car Merc 280C: 123 horsepower, 6 number of cylinders and consumption 17.8 miles per gallon
## The car Merc 450SE: 180 horsepower, 8 number of cylinders and consumption 16.4 miles per gallon
## The car Merc 450SL: 180 horsepower, 8 number of cylinders and consumption 17.3 miles per gallon
## The car Merc 450SLC: 180 horsepower, 8 number of cylinders and consumption 15.2 miles per gallon
## The car Cadillac Fleetwood: 205 horsepower, 8 number of cylinders and consumption 10.4 miles per gal
## The car Lincoln Continental: 215 horsepower, 8 number of cylinders and consumption 10.4 miles per ga
## The car Chrysler Imperial: 230 horsepower, 8 number of cylinders and consumption 14.7 miles per gall
## The car Fiat 128: 66 horsepower, 4 number of cylinders and consumption 32.4 miles per gallon
## The car Honda Civic: 52 horsepower, 4 number of cylinders and consumption 30.4 miles per gallon
## The car Toyota Corolla: 65 horsepower, 4 number of cylinders and consumption 33.9 miles per gallon
## The car Toyota Corona: 97 horsepower, 4 number of cylinders and consumption 21.5 miles per gallon
## The car Dodge Challenger: 150 horsepower, 8 number of cylinders and consumption 15.5 miles per gallo
## The car AMC Javelin: 150 horsepower, 8 number of cylinders and consumption 15.2 miles per gallon
## The car Camaro Z28: 245 horsepower, 8 number of cylinders and consumption 13.3 miles per gallon
## The car Pontiac Firebird: 175 horsepower, 8 number of cylinders and consumption 19.2 miles per gallo
## The car Fiat X1-9: 66 horsepower, 4 number of cylinders and consumption 27.3 miles per gallon
## The car Porsche 914-2: 91 horsepower, 4 number of cylinders and consumption 26 miles per gallon
## The car Lotus Europa: 113 horsepower, 4 number of cylinders and consumption 30.4 miles per gallon
## The car Ford Pantera L: 264 horsepower, 8 number of cylinders and consumption 15.8 miles per gallon
## The car Ferrari Dino: 175 horsepower, 6 number of cylinders and consumption 19.7 miles per gallon
## The car Maserati Bora: 335 horsepower, 8 number of cylinders and consumption 15 miles per gallon
## The car Volvo 142E: 109 horsepower, 4 number of cylinders and consumption 21.4 miles per gallon
```

Other string helper functions

```
# str_order() - Return a vector of indexes after character vector is sorted.
# Let's first shuffle fruits (to get random order)
set.seed(456)
fruit.shuf <- sample(x = fruit, size = length(fruit), replace = F)
fruit.shuf
```

```
##  [1] "grapefruit"        "huckleberry"       "durian"
##  [4] "date"              "fig"               "salal berry"
##  [7] "kumquat"           "blackcurrant"      "cantaloupe"
## [10] "canary melon"      "rambutan"          "nut"
## [13] "feijoa"            "lychee"            "tamarillo"
## [16] "ugli fruit"        "jambul"            "honeydew"
## [19] "jujube"            "grape"             "raspberry"
## [22] "peach"             "tangerine"         "banana"
## [25] "eggplant"          "blueberry"         "strawberry"
## [28] "apple"             "pamelo"            "purple mangosteen"
## [31] "satsuma"           "coconut"           "raisin"
## [34] "boysenberry"       "damson"            "physalis"
## [37] "pomelo"            "mulberry"          "mandarine"
## [40] "watermelon"        "cherimoya"         "gooseberry"
## [43] "passionfruit"      "mango"             "cherry"
## [46] "nectarine"         "apricot"           "currant"
## [49] "pear"              "guava"             "orange"
## [52] "loquat"            "cucumber"          "pineapple"
## [55] "avocado"           "pomegranate"       "dragonfruit"
## [58] "jackfruit"         "olive"             "breadfruit"
## [61] "cranberry"         "chili pepper"      "lime"
## [64] "cloudberry"        "bell pepper"       "rock melon"
## [67] "lemon"             "elderberry"        "blood orange"
## [70] "papaya"            "bilberry"          "kiwi fruit"
## [73] "persimmon"         "redcurrant"        "quince"
## [76] "goji berry"        "blackberry"        "clementine"
## [79] "star fruit"        "plum"
```

```
# Now get order index and use to sort shuffled fruits
str_order(x = fruit.shuf) # get index
```

```
##  [1] 28 47 55 24 65 71 77  8 69 26 34 60 10  9 41 45 62 78 64 32 61 53 48 35  4
## [26] 57  3 25 68 13  5 76 42 20  1 50 18  2 58 17 19 72  7 67 63 52 14 39 44 38
## [51] 46 12 59 51 29 70 43 22 49 73 36 54 80 56 37 30 75 33 11 21 74 66  6 31 79
## [76] 27 15 23 16 40
```

```
fruit.shuf[str_order(x = fruit.shuf)] # use it for sort
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "bell pepper"       "bilberry"
##  [7] "blackberry"        "blackcurrant"      "blood orange"
## [10] "blueberry"         "boysenberry"       "breadfruit"
## [13] "canary melon"      "cantaloupe"        "cherimoya"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "cranberry"
## [22] "cucumber"          "currant"           "damson"
## [25] "date"              "dragonfruit"       "durian"
## [28] "eggplant"          "elderberry"        "feijoa"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grape"             "grapefruit"        "guava"
## [37] "honeydew"          "huckleberry"       "jackfruit"
## [40] "jambul"            "jujube"            "kiwi fruit"
## [43] "kumquat"           "lemon"             "lime"
## [46] "loquat"            "lychee"            "mandarine"
## [49] "mango"             "mulberry"          "nectarine"
## [52] "nut"               "olive"             "orange"
## [55] "pamelo"            "papaya"            "passionfruit"
## [58] "peach"             "pear"              "persimmon"
## [61] "physalis"          "pineapple"         "plum"
## [64] "pomegranate"       "pomelo"            "purple mangosteen"
## [67] "quince"            "raisin"            "rambutan"
## [70] "raspberry"         "redcurrant"        "rock melon"
## [73] "salal berry"       "satsuma"           "star fruit"
## [76] "strawberry"        "tamarillo"         "tangerine"
## [79] "ugli fruit"        "watermelon"
```

```
# str_sort() - Sort character vector
# Let' sort shuffled fruits
str_sort(x = fruit.shuf)
```

```
##  [1] "apple"             "apricot"           "avocado"
##  [4] "banana"            "bell pepper"       "bilberry"
##  [7] "blackberry"        "blackcurrant"      "blood orange"
## [10] "blueberry"         "boysenberry"       "breadfruit"
## [13] "canary melon"      "cantaloupe"        "cherimoya"
## [16] "cherry"            "chili pepper"      "clementine"
## [19] "cloudberry"        "coconut"           "cranberry"
## [22] "cucumber"          "currant"           "damson"
## [25] "date"              "dragonfruit"       "durian"
## [28] "eggplant"          "elderberry"        "feijoa"
## [31] "fig"               "goji berry"        "gooseberry"
## [34] "grape"             "grapefruit"        "guava"
## [37] "honeydew"          "huckleberry"       "jackfruit"
## [40] "jambul"            "jujube"            "kiwi fruit"
## [43] "kumquat"           "lemon"             "lime"
## [46] "loquat"            "lychee"            "mandarine"
## [49] "mango"             "mulberry"          "nectarine"
## [52] "nut"               "olive"             "orange"
## [55] "pamelo"            "papaya"            "passionfruit"
## [58] "peach"             "pear"              "persimmon"
## [61] "physalis"          "pineapple"         "plum"
## [64] "pomegranate"       "pomelo"            "purple mangosteen"
## [67] "quince"            "raisin"            "rambutan"
## [70] "raspberry"         "redcurrant"        "rock melon"
## [73] "salal berry"       "satsuma"           "star fruit"
## [76] "strawberry"        "tamarillo"         "tangerine"
## [79] "ugli fruit"        "watermelon"
```

```
str_sort(x = fruit.shuf, decreasing = T)
```

```
##  [1] "watermelon"        "ugli fruit"        "tangerine"
##  [4] "tamarillo"         "strawberry"        "star fruit"
##  [7] "satsuma"           "salal berry"       "rock melon"
## [10] "redcurrant"        "raspberry"         "rambutan"
## [13] "raisin"            "quince"            "purple mangosteen"
## [16] "pomelo"            "pomegranate"       "plum"
## [19] "pineapple"         "physalis"          "persimmon"
## [22] "pear"              "peach"             "passionfruit"
## [25] "papaya"            "pamelo"            "orange"
## [28] "olive"             "nut"               "nectarine"
## [31] "mulberry"          "mango"             "mandarine"
## [34] "lychee"            "loquat"            "lime"
## [37] "lemon"             "kumquat"           "kiwi fruit"
## [40] "jujube"            "jambul"            "jackfruit"
## [43] "huckleberry"       "honeydew"          "guava"
## [46] "grapefruit"        "grape"             "gooseberry"
## [49] "goji berry"        "fig"               "feijoa"
## [52] "elderberry"        "eggplant"          "durian"
## [55] "dragonfruit"       "date"              "damson"
## [58] "currant"           "cucumber"          "cranberry"
## [61] "coconut"           "cloudberry"        "clementine"
## [64] "chili pepper"      "cherry"            "cherimoya"
## [67] "cantaloupe"        "canary melon"      "breadfruit"
## [70] "boysenberry"       "blueberry"         "blood orange"
## [73] "blackcurrant"      "blackberry"        "bilberry"
## [76] "bell pepper"       "banana"            "avocado"
## [79] "apricot"           "apple"
```

```
# Sorting numbers stored as strings!
set.seed(567)
numbers.s <- sample(1:250, size = 20, replace = F) # generate some numbers
numbers.s <- as.character(numbers.s) # conver numbers to charatcer
numbers.s
```

```
##  [1] "141" "28"  "69"  "199" "215" "46"  "106" "95"  "21"  "76"  "20"  "211"
## [13] "54"  "82"  "47"  "29"  "62"  "182" "185" "166"
```

```
str_sort(numbers.s) # not sorted as numbers but as strings
```

```
##  [1] "106" "141" "166" "182" "185" "199" "20"  "21"  "211" "215" "28"  "29"
## [13] "46"  "47"  "54"  "62"  "69"  "76"  "82"  "95"
```

```
str_sort(numbers.s, numeric = T) # sorted as numbers
```

```
##  [1] "20"  "21"  "28"  "29"  "46"  "47"  "54"  "62"  "69"  "76"  "82"  "95"
## [13] "106" "141" "166" "182" "185" "199" "211" "215"
```

```
# str_view() / str_view_all() - Useful HTML rendering function
# very useful in the context of regular expressions
# (given context will be shown later)
# View first match
str_view(string = fruit, pattern = "a") # displays all
```

```
##  [1] | <a>pple
##  [2] | <a>pricot
##  [3] | <a>voc<a>do
##  [4] | b<a>n<a>n<a>
##  [7] | bl<a>ckberry
##  [8] | bl<a>ckcurr<a>nt
##  [9] | blood or<a>nge
## [12] | bre<a>dfruit
## [13] | c<a>n<a>ry melon
## [14] | c<a>nt<a>loupe
## [15] | cherimoy<a>
## [21] | cr<a>nberry
## [23] | curr<a>nt
## [24] | d<a>mson
## [25] | d<a>te
## [26] | dr<a>gonfruit
## [27] | duri<a>n
## [28] | eggpl<a>nt
## [30] | feijo<a>
## [34] | gr<a>pe
## ... and 30 more
```

```
str_view(string = fruit, pattern = "a", match = T) # display only matched
```

```
##  [1] | <a>pple
##  [2] | <a>pricot
##  [3] | <a>voc<a>do
##  [4] | b<a>n<a>n<a>
##  [7] | bl<a>ckberry
##  [8] | bl<a>ckcurr<a>nt
##  [9] | blood or<a>nge
## [12] | bre<a>dfruit
## [13] | c<a>n<a>ry melon
## [14] | c<a>nt<a>loupe
## [15] | cherimoy<a>
## [21] | cr<a>nberry
## [23] | curr<a>nt
## [24] | d<a>mson
## [25] | d<a>te
## [26] | dr<a>gonfruit
## [27] | duri<a>n
## [28] | eggpl<a>nt
## [30] | feijo<a>
## [34] | gr<a>pe
## ... and 30 more
```

```
str_view(string = fruit, pattern = "^a", match = T)
```

```
## [1] | <a>pple
## [2] | <a>pricot
## [3] | <a>vocado
```

```
# View all matches
str_view_all(string = fruit, pattern = "a", match = T)
```

```
##  [1] | <a>pple
##  [2] | <a>pricot
##  [3] | <a>voc<a>do
##  [4] | b<a>n<a>n<a>
##  [7] | bl<a>ckberry
##  [8] | bl<a>ckcurr<a>nt
##  [9] | blood or<a>nge
## [12] | bre<a>dfruit
## [13] | c<a>n<a>ry melon
## [14] | c<a>nt<a>loupe
## [15] | cherimoy<a>
## [21] | cr<a>nberry
## [23] | curr<a>nt
## [24] | d<a>mson
## [25] | d<a>te
## [26] | dr<a>gonfruit
## [27] | duri<a>n
## [28] | eggpl<a>nt
## [30] | feijo<a>
## [34] | gr<a>pe
## ... and 30 more
```

Regular expressions (regex)

```
# Get list of some special characters
?"'"

# Escaping paradox
string <- c("string", "word", "letter", "word.letter", "character/letter")

# Match "tr"
str_view(string, "tr")
```

```
## [1] | s<tr>ing
```

```
# Match ".t." - any character before t and any character after t
str_view(string, ".t.")
```

```
## [1] | <str>ing
## [3] | l<ett>er
## [4] | word.l<ett>er
## [5] | chara<cte>r/l<ett>er
```

```
str_view_all(string, ".t.")
```

```
## [1] | <str>ing
## [2] | word
## [3] | l<ett>er
## [4] | word.l<ett>er
## [5] | chara<cte>r/l<ett>er
```

```
# Match "." as a dot not as a metacharacter meaning:
# 1) wrong way: since . is interpreted as metacharater ~ any charatcer
#str_view(string, ".")

# 2) wrong way (single backslash \ ): escaping is applied on . but \ is not escaped!
#str_view(string, "\.")

# 3) correct way (double backslash \\ ): escaping is applied on . and \ !
#str_view(string, "\\.")

# Match "\" as a backslash character not as a metacharacter meaning:
#writeLines("\\") # \ must be escaped when written as a string
#str_view("\\", "\\\\") # double escaping is applied int he pattern ~ four \ in total at the end!
```

Regex: Special characters & Classes

```r
# Digits VS non-digits
string <- c(letters, "123", "1-5-6", "598642")
string
```

```
##  [1] "a"      "b"      "c"      "d"      "e"      "f"      "g"      "h"
##  [9] "i"      "j"      "k"      "l"      "m"      "n"      "o"      "p"
## [17] "q"      "r"      "s"      "t"      "u"      "v"      "w"      "x"
## [25] "y"      "z"      "123"    "1-5-6"  "598642"
```

```r
# Find strings with digits
str_subset(string, "\\d")
```

```
## [1] "123"    "1-5-6"  "598642"
```

```r
str_view_all(string, "\\d", match = T)
```

```
## [27] | <1><2><3>
## [28] | <1>-<5>-<6>
## [29] | <5><9><8><6><4><2>
```

```r
# Find strings without digits
str_subset(string, "\\D")
```

```
##  [1] "a"     "b"     "c"     "d"     "e"     "f"     "g"     "h"     "i"
## [10] "j"     "k"     "l"     "m"     "n"     "o"     "p"     "q"     "r"
## [19] "s"     "t"     "u"     "v"     "w"     "x"     "y"     "z"     "1-5-6"
```

```r
str_view_all(string, "\\D", match = T)
```

```
##  [1] | <a>
##  [2] | <b>
##  [3] | <c>
##  [4] | <d>
##  [5] | <e>
##  [6] | <f>
##  [7] | <g>
##  [8] | <h>
##  [9] | <i>
## [10] | <j>
## [11] | <k>
## [12] | <l>
## [13] | <m>
## [14] | <n>
## [15] | <o>
## [16] | <p>
## [17] | <q>
## [18] | <r>
## [19] | <s>
## [20] | <t>
## ... and 7 more
```

```r
# Strings with pattern "digit-digit-digit"
str_subset(string, "\\d-\\d-\\d")
```

```
## [1] "1-5-6"
```

```r
str_view_all(string, "\\d-\\d-\\d", match = T)
```

```
## [28] | <1-5-6>
```

```r
# Locate whitespace(s)
set.seed(123)
string <- c(sample(sentences, 5),
            sample(fruit, 5),
            sample(words, 5),
            "This is \nnewline",
            "String with a tab \t")
string
```

```
##  [1] "Fasten two pins on each side."
##  [2] "The bunch of grapes was pressed into wine."
##  [3] "They felt gay when the ship arrived in port."
##  [4] "Shake hands with this friendly child."
##  [5] "Sell your gift to a buyer at a good gain."
##  [6] "kiwi fruit"
##  [7] "mulberry"
##  [8] "kumquat"
##  [9] "cantaloupe"
## [10] "date"
## [11] "team"
## [12] "measure"
## [13] "trace"
## [14] "fairly"
## [15] "freedom"
## [16] "This is \nnewline"
## [17] "String with a tab \t"
```

```r
writeLines(string)
```

```
## Fasten two pins on each side.
## The bunch of grapes was pressed into wine.
## They felt gay when the ship arrived in port.
## Shake hands with this friendly child.
## Sell your gift to a buyer at a good gain.
## kiwi fruit
## mulberry
## kumquat
## cantaloupe
## date
## team
## measure
## trace
## fairly
## freedom
## This is
## newline
## String with a tab
```

```r
str_subset(string, "\\s") # only strings with white spaces
```

```
## [1] "Fasten two pins on each side."
## [2] "The bunch of grapes was pressed into wine."
## [3] "They felt gay when the ship arrived in port."
## [4] "Shake hands with this friendly child."
## [5] "Sell your gift to a buyer at a good gain."
## [6] "kiwi fruit"
## [7] "This is \nnewline"
## [8] "String with a tab \t"
```

```r
str_view_all(string, "\\s")
```

```
##  [1] | Fasten< >two< >pins< >on< >each< >side.
##  [2] | The< >bunch< >of< >grapes< >was< >pressed< >into< >wine.
##  [3] | They< >felt< >gay< >when< >the< >ship< >arrived< >in< >port.
##  [4] | Shake< >hands< >with< >this< >friendly< >child.
##  [5] | Sell< >your< >gift< >to< >a< >buyer< >at< >a< >good< >gain.
##  [6] | kiwi< >fruit
##  [7] | mulberry
##  [8] | kumquat
##  [9] | cantaloupe
## [10] | date
## [11] | team
## [12] | measure
## [13] | trace
## [14] | fairly
## [15] | freedom
## [16] | This< >is< ><
##      | >newline
## [17] | String< >with< >a< >tab< ><{\t}>
```

```r
# Locate string with new lines or tabs
str_subset(string, "\\n") # only strings with new lines
```

```
## [1] "This is \nnewline"
```

```r
str_subset(string, "\\t") # only strings with tabs
```

```
## [1] "String with a tab \t"
```

```r
# Different classes
string <- c("123abc", "abc", "123", ".,?", "ABC", "\nABC", "\tabc")
string
```

```
## [1] "123abc" "abc"    "123"    ".,?"    "ABC"    "\nABC"  "\tabc"
```

```r
# Strings with digits
str_subset(string, "[:digit:]")
```

```
## [1] "123abc" "123"
```

```r
str_view_all(string, "[:digit:]", match = T)
```

```
## [1] | <1><2><3>abc
## [3] | <1><2><3>
```

```r
# Strings with letters
str_subset(string, "[:alpha:]")
```

```
## [1] "123abc" "abc"    "ABC"    "\nABC"  "\tabc"
```

```r
str_view_all(string, "[:alpha:]", match = T)
```

```
## [1] | 123<a><b><c>
## [2] | <a><b><c>
## [5] | <A><B><C>
## [6] |
##     | <A><B><C>
## [7] | {\t}<a><b><c>
```

```r
# Strings with upper / lower case letters
str_subset(string, "[:lower:]")
```

```
## [1] "123abc" "abc"    "\tabc"
```

```r
str_view_all(string, "[:lower:]", match = T)
```

```
## [1] | 123<a><b><c>
## [2] | <a><b><c>
## [7] | {\t}<a><b><c>
```

```r
str_subset(string, "[:upper:]")
```

```
## [1] "ABC"   "\nABC"
```

```r
str_view_all(string, "[:upper:]", match = T)
```

```
## [5] | <A><B><C>
## [6] |
##     | <A><B><C>
```

```r
# Strings with letters or numbers
str_subset(string, "[:alnum:]")
```

```
## [1] "123abc" "abc"    "123"    "ABC"    "\nABC"  "\tabc"
```

```r
str_view_all(string, "[:alnum:]", match = T)
```

```
## [1] | <1><2><3><a><b><c>
## [2] | <a><b><c>
## [3] | <1><2><3>
## [5] | <A><B><C>
## [6] |
##     | <A><B><C>
## [7] | {\t}<a><b><c>
```

```r
# Strings with punctuation
str_subset(string, "[:punct:]")
```

```
## [1] ".,?"
```

```r
str_view_all(string, "[:punct:]", match = T)
```

```
## [4] | <.><,><?>
```

```r
# Strings with letters, numbers or punctuation
str_subset(string, "[:graph:]")
```

```
## [1] "123abc" "abc"    "123"    ".,?"    "ABC"    "\nABC"  "\tabc"
```

```r
str_view_all(string, "[:graph:]", match = T)
```

```
## [1] | <1><2><3><a><b><c>
## [2] | <a><b><c>
## [3] | <1><2><3>
## [4] | <.><,><?>
## [5] | <A><B><C>
## [6] |
##     | <A><B><C>
## [7] | {\t}<a><b><c>
```

```
# Strings with space characters
str_subset(string, "[:blank:]")
```

```
## [1] "\tabc"
```

```
str_view_all(string, "[:blank:]", match = T)
```

```
## [7] | <{\t}>abc
```

Regex: Alternates, anchors & groups

```
# Anchors
# Find a word starting with letter "a"
str_subset(words, "^a")
```

```
##   [1] "a"              "abandon"        "ability"        "able"
##   [5] "abortion"       "about"          "above"          "abroad"
##   [9] "absence"        "absolute"       "absolutely"     "absorb"
##  [13] "abuse"          "academic"       "accept"         "access"
##  [17] "accident"       "accompany"      "accomplish"     "according"
##  [21] "account"        "accurate"       "accuse"         "achieve"
##  [25] "achievement"    "acid"           "acknowledge"    "acquire"
##  [29] "across"         "act"            "action"         "active"
##  [33] "activist"       "activity"       "actor"          "actress"
##  [37] "actual"         "actually"       "ad"             "adapt"
##  [41] "add"            "addition"       "additional"     "address"
##  [45] "adequate"       "adjust"         "adjustment"     "administration"
##  [49] "administrator"  "admire"         "admission"      "admit"
##  [53] "adolescent"     "adopt"          "adult"          "advance"
##  [57] "advanced"       "advantage"      "adventure"      "advertising"
##  [61] "advice"         "advise"         "adviser"        "advocate"
##  [65] "affair"         "affect"         "afford"         "afraid"
##  [69] "after"          "afternoon"      "again"          "against"
##  [73] "age"            "agency"         "agenda"         "agent"
##  [77] "aggressive"     "ago"            "agree"          "agreement"
##  [81] "agricultural"   "ah"             "ahead"          "aid"
##  [85] "aide"           "aim"            "air"            "aircraft"
##  [89] "airline"        "airport"        "album"          "alcohol"
##  [93] "alive"          "all"            "alliance"       "allow"
##  [97] "ally"           "almost"         "alone"          "along"
## [101] "already"        "also"           "alter"          "alternative"
## [105] "although"       "always"         "amazing"        "among"
## [109] "amount"         "analysis"       "analyst"        "analyze"
## [113] "ancient"        "and"            "anger"          "angle"
## [117] "angry"          "animal"         "anniversary"    "announce"
## [121] "annual"         "another"        "answer"         "anticipate"
## [125] "anxiety"        "any"            "anybody"        "anymore"
## [129] "anyone"         "anything"       "anyway"         "anywhere"
## [133] "apart"          "apartment"      "apparent"       "apparently"
## [137] "appeal"         "appear"         "appearance"     "apple"
## [141] "application"    "apply"          "appoint"        "appointment"
## [145] "appreciate"     "approach"       "appropriate"    "approval"
## [149] "approve"        "approximately"  "architect"      "area"
## [153] "argue"          "argument"       "arise"          "arm"
## [157] "armed"          "army"           "around"         "arrange"
## [161] "arrangement"    "arrest"         "arrival"        "arrive"
## [165] "art"            "article"        "artist"         "artistic"
## [169] "as"             "aside"          "ask"            "asleep"
## [173] "aspect"         "assault"        "assert"         "assess"
## [177] "assessment"     "asset"          "assign"         "assignment"
## [181] "assist"         "assistance"     "assistant"      "associate"
## [185] "association"    "assume"         "assumption"     "assure"
## [189] "at"             "athlete"        "athletic"       "atmosphere"
## [193] "attach"         "attack"         "attempt"        "attend"
## [197] "attention"      "attitude"       "attorney"       "attract"
## [201] "attractive"     "attribute"      "audience"       "author"
## [205] "authority"      "auto"           "available"      "average"
## [209] "avoid"          "award"          "aware"          "awareness"
## [213] "away"           "awful"
```

```
str_view_all(words, "^a", match = T)
```

```
##  [1] | <a>
##  [2] | <a>bandon
##  [3] | <a>bility
##  [4] | <a>ble
##  [5] | <a>bortion
##  [6] | <a>bout
##  [7] | <a>bove
##  [8] | <a>broad
##  [9] | <a>bsence
## [10] | <a>bsolute
## [11] | <a>bsolutely
## [12] | <a>bsorb
## [13] | <a>buse
## [14] | <a>cademic
## [15] | <a>ccept
## [16] | <a>ccess
## [17] | <a>ccident
## [18] | <a>ccompany
## [19] | <a>ccomplish
## [20] | <a>ccording
## ... and 194 more
```

```
# Find a word ending with letter "a"
str_subset(words, "a$")
```

```
##  [1] "a"        "agenda"   "area"     "camera"   "criteria" "data"
##  [7] "drama"    "era"      "extra"    "formula"  "idea"     "media"
## [13] "sea"      "tea"      "via"
```

```
str_view_all(words, "a$", match = T)
```

```
##    [1] | <a>
##   [77] | agend<a>
##  [158] | are<a>
##  [373] │ camer<a>
##  [650] │ criteri<a>
##  [680] │ dat<a>
##  [817] │ dram<a>
##  [923] │ er<a>
##  [995] │ extr<a>
## [1103] │ formul<a>
## [1315] │ ide<a>
## [1631] │ medi<a>
## [2330] │ se<a>
## [2646] │ te<a>
## [2859] │ vi<a>
```

```
# Find exact word using ^....$
str_subset(words, "^actor$")
```

```
## [1] "actor"
```

```
str_view_all(words, "^actor$", match = T)
```

```
## [35] │ <actor>
```

```
str_subset(fruit, "^lemon$")
```

```
## [1] "lemon"
```

```
str_view_all(fruit, "^lemon$", match = T)
```

```
## [44] │ <lemon>
```

```
# Alternates
# Find words that starts with "af" or "ag"
str_subset(words, "^af|^ag")
```

```
##  [1] "affair"       "affect"       "afford"       "afraid"       "after"
##  [6] "afternoon"    "again"        "against"      "age"          "agency"
## [11] "agenda"       "agent"        "aggressive"   "ago"          "agree"
## [16] "agreement"    "agricultural"
```

```
str_view_all(words, "^af|^ag", match = T)
```

```
## [65] │ <af>fair
## [66] │ <af>fect
## [67] │ <af>ford
## [68] │ <af>raid
## [71] │ <af>ter
## [72] │ <af>ternoon
## [73] │ <ag>ain
## [74] │ <ag>ainst
## [75] │ <ag>e
## [76] │ <ag>ency
## [77] │ <ag>enda
## [78] │ <ag>ent
## [79] │ <ag>gressive
## [80] │ <ag>o
## [81] │ <ag>ree
## [82] │ <ag>reement
## [83] │ <ag>ricultural
```

```r
# Find words containing letters "x" or "y" or "z"
str_subset(words, "[xyz]")
```

```
##   [1] "ability"        "absolutely"     "accompany"      "activity"
##   [5] "actually"       "agency"         "ally"           "already"
##   [9] "always"         "amazing"        "analysis"       "analyst"
##  [13] "analyze"        "angry"          "anniversary"    "anxiety"
##  [17] "any"            "anybody"        "anymore"        "anyone"
##  [21] "anything"       "anyway"         "anywhere"       "apparently"
##  [25] "apply"          "approximately"  "army"           "attorney"
##  [29] "authority"      "away"           "baby"           "badly"
##  [33] "barely"         "basically"      "battery"        "beauty"
##  [37] "beyond"         "birthday"       "body"           "boundary"
##  [41] "box"            "boy"            "boyfriend"      "briefly"
##  [45] "bury"           "busy"           "buy"            "buyer"
##  [49] "by"             "capability"     "capacity"       "carefully"
##  [53] "carry"          "category"       "celebrity"      "century"
##  [57] "ceremony"       "certainly"      "characterize"   "charity"
##  [61] "citizen"        "city"           "clearly"        "closely"
##  [65] "comedy"         "community"      "company"        "completely"
##  [69] "complex"        "constantly"     "contemporary"   "context"
##  [73] "controversy"    "copy"           "country"        "county"
##  [77] "crazy"          "criticize"      "cry"            "currently"
##  [81] "cycle"          "daily"          "day"            "deeply"
##  [85] "definitely"     "delay"          "delivery"       "democracy"
##  [89] "deny"           "deputy"         "destroy"        "differently"
##  [93] "difficulty"     "directly"       "dirty"          "disability"
##  [97] "discovery"      "display"        "diversity"      "dozen"
## [101] "dramatically"   "dry"            "duty"           "early"
## [105] "easily"         "easy"           "economy"        "effectively"
## [109] "efficiency"     "elderly"        "electricity"    "elementary"
## [113] "emergency"      "emphasize"      "employ"         "employee"
## [117] "employer"       "employment"     "empty"          "enemy"
## [121] "energy"         "enjoy"          "entirely"       "entry"
## [125] "equally"        "especially"     "essay"          "essentially"
## [129] "eventually"     "every"          "everybody"      "everyday"
## [133] "everyone"       "everything"     "everywhere"     "exact"
## [137] "exactly"        "examination"    "examine"        "example"
## [141] "exceed"         "excellent"      "except"         "exception"
## [145] "exchange"       "exciting"       "executive"      "exercise"
## [149] "exhibit"        "exhibition"     "exist"          "existence"
## [153] "existing"       "expand"         "expansion"      "expect"
## [157] "expectation"    "expense"        "expensive"      "experience"
## [161] "experiment"     "expert"         "explain"        "explanation"
## [165] "explode"        "explore"        "explosion"      "expose"
## [169] "exposure"       "express"        "expression"     "extend"
## [173] "extension"      "extensive"      "extent"         "external"
## [177] "extra"          "extraordinary"  "extreme"        "extremely"
## [181] "eye"            "facility"       "factory"        "faculty"
## [185] "fairly"         "family"         "fantasy"        "fifty"
## [189] "finally"        "fix"            "fly"            "freeze"
## [193] "frequency"      "frequently"     "friendly"       "fully"
## [197] "funny"          "galaxy"         "gallery"        "gay"
## [201] "gaze"           "generally"      "gently"         "gradually"
## [205] "gray"           "grocery"        "guilty"         "guy"
## [209] "happy"          "hardly"         "healthy"        "heavily"
## [213] "heavy"          "hey"            "highly"         "highway"
## [217] "history"        "holiday"        "holy"           "honey"
## [221] "horizon"        "hungry"         "hypothesis"     "identify"
## [225] "identity"       "immediately"    "imply"          "increasingly"
## [229] "index"          "industry"       "initially"      "injury"
## [233] "inquiry"        "intensity"      "journey"        "joy"
## [237] "jury"           "justify"        "key"            "laboratory"
## [241] "lady"           "largely"        "lawyer"         "lay"
## [245] "layer"          "legacy"         "library"        "lifestyle"
## [249] "likely"         "literally"      "literary"       "lovely"
## [253] "lucky"          "magazine"       "mainly"         "majority"
## [257] "many"           "marry"          "may"            "maybe"
## [261] "mayor"          "memory"         "merely"         "Mexican"
## [265] "military"       "minority"       "mix"            "mixture"
## [269] "money"          "mostly"         "my"             "myself"
## [273] "mystery"        "myth"           "naturally"      "nearby"
## [277] "nearly"         "necessarily"    "necessary"      "newly"
## [281] "next"           "nobody"         "normally"       "obviously"
## [285] "occasionally"   "occupy"         "okay"           "Olympic"
## [289] "only"           "opportunity"    "ordinary"       "organization"
## [293] "organize"       "originally"     "particularly"   "partly"
## [297] "party"          "pay"            "payment"        "penalty"
## [301] "perfectly"      "personality"    "personally"     "philosophy"
## [305] "physical"       "physically"     "physician"      "play"
## [309] "player"         "plenty"         "poetry"         "policy"
## [313] "politically"    "portray"        "possibility"    "possibly"
## [317] "potentially"    "poverty"        "pray"           "prayer"
## [321] "precisely"      "pregnancy"      "pretty"         "previously"
## [325] "primarily"      "primary"        "priority"       "privacy"
## [329] "probably"       "properly"       "property"       "psychological"
## [333] "psychologist"   "psychology"     "publicly"       "qualify"
## [337] "quality"        "quickly"        "quietly"        "rapidly"
## [341] "rarely"         "ready"          "reality"        "realize"
## [345] "really"         "recently"       "recognize"      "recovery"
## [349] "regularly"      "relatively"     "relax"          "rely"
## [353] "repeatedly"     "reply"          "responsibility" "rhythm"
## [357] "roughly"        "safety"         "salary"         "satisfy"
## [361] "say"            "secretary"      "security"       "seize"
## [365] "seriously"      "sex"            "sexual"         "shortly"
## [369] "significantly"  "similarly"      "simply"         "six"
## [373] "size"           "sky"            "slightly"       "slowly"
## [377] "society"        "somebody"       "sorry"          "specifically"
## [381] "squeeze"        "stability"      "stay"           "steady"
## [385] "story"          "strategy"       "strongly"       "study"
## [389] "style"          "successfully"   "suddenly"       "supply"
## [393] "surely"         "surgery"        "surprisingly"   "survey"
## [397] "symbol"         "symptom"        "system"         "tax"
## [401] "taxpayer"       "technology"     "temporary"      "tendency"
## [405] "territory"      "testify"        "testimony"      "text"
## [409] "theory"         "therapy"        "they"           "thirty"
## [413] "tiny"           "today"          "totally"        "toy"
## [417] "tragedy"        "treaty"         "truly"          "try"
## [421] "twenty"        "type"          "typical"       "typically"
## [425] "ugly"          "ultimately"    "unfortunately" "university"
## [429] "unlikely"      "usually"       "utility"       "valley"
## [433] "variety"       "vary"          "very"          "victory"
## [437] "virtually"     "way"           "wealthy"       "weekly"
## [441] "why"           "widely"        "worry"         "yard"
## [445] "yeah"          "year"          "yell"          "yellow"
## [449] "yes"           "yesterday"     "yet"           "yield"
## [453] "you"           "young"         "your"          "yours"
## [457] "yourself"      "youth"         "zone"
```

```r
# Find words not containing letters from "a" to "x"
str_subset(words %>% str_to_lower(), "[^[a-y]]")
```

```
##  [1] "african-american" "amazing"          "analyze"          "characterize"
##  [5] "citizen"          "crazy"            "criticize"        "dozen"
##  [9] "e-mail"           "emphasize"        "freeze"           "gaze"
## [13] "horizon"          "long-term"        "magazine"         "mm-hmm"
## [17] "n't"              "organization"     "organize"         "realize"
## [21] "recognize"        "seize"            "size"             "so-called"
## [25] "squeeze"          "zone"
```

```r
str_view_all(words %>% str_to_lower(), "[^[a-y]]", match = T)
```

```
##   [70] | african<->american
##  [111] | ama<z>ing
##  [117] | analy<z>e
##  [431] | characteri<z>e
##  [460] | citi<z>en
##  [639] | cra<z>y
##  [654] | critici<z>e
##  [814] | do<z>en
##  [879] | e<->mail
##  [887] | emphasi<z>e
## [1116] | free<z>e
## [1155] | ga<z>e
## [1290] | hori<z>on
## [1562] | long<->term
## [1582] | maga<z>ine
## [1671] | mm<->hmm
## [1778] | n<'>t
## [1842] | organi<z>ation
## [1843] | organi<z>e
## [2153] | reali<z>e
## ... and 6 more
```

```r
# Find all country names beginning with letter "A" or "E"
str_subset(countries, "^A|^E")
```

```
##  [1] "Afghanistan"       "Albania"           "Algeria"
##  [4] "American Samoa"    "Andorra"           "Angola"
##  [7] "Anguilla"          "Antarctica"        "Antigua & Barbuda"
## [10] "Argentina"         "Armenia"           "Aruba"
## [13] "Australia"         "Austria"           "Austria-Hungary"
## [16] "Azerbaijan"        "Ecuador"           "Egypt"
## [19] "El Salvador"       "Equatorial Guinea" "Eritrea"
## [22] "Estonia"           "Ethiopia"
```

```
# Find all country names ending with letter "a" or "e"
str_subset(countries, "a$|e$")
```

```
##   [1] "Albania"              "Algeria"              "American Samoa"
##   [4] "Andorra"              "Angola"               "Anguilla"
##   [7] "Antarctica"           "Antigua & Barbuda"    "Argentina"
##  [10] "Armenia"              "Aruba"                "Australia"
##  [13] "Austria"              "Bavaria"              "Belize"
##  [16] "Bermuda"              "Bolivia"              "Bosnia & Herzegovina"
##  [19] "Botswana"             "Bulgaria"             "Cape Verde"
##  [22] "Cambodia"             "Canada"               "Chile"
##  [25] "China"                "Colombia"             "Congo - Brazzaville"
##  [28] "Costa Rica"           "Côte d’Ivoire"        "Croatia"
##  [31] "Cuba"                 "Czechia"              "Czechoslovakia"
##  [34] "North Korea"          "Congo - Kinshasa"     "Dominica"
##  [37] "Equatorial Guinea"    "Eritrea"              "Estonia"
##  [40] "Ethiopia"             "France"               "French Guiana"
##  [43] "French Polynesia"     "Gambia"               "Georgia"
##  [46] "Ghana"                "Greece"               "Grenada"
##  [49] "Guadeloupe"           "Guatemala"            "Guinea"
##  [52] "Guyana"               "Hong Kong SAR China"  "India"
##  [55] "Indonesia"            "Jamaica"              "Kenya"
##  [58] "Latvia"               "Liberia"              "Libya"
##  [61] "Lithuania"            "Macau SAR China"      "Malaysia"
##  [64] "Malta"                "Martinique"           "Mauritania"
##  [67] "Mayotte"              "Modena"               "Mongolia"
##  [70] "Mozambique"           "Namibia"              "New Caledonia"
##  [73] "Nicaragua"            "Nigeria"              "Niue"
##  [76] "Orange Free State"    "Panama"               "Papua New Guinea"
##  [79] "Parma"                "Piedmont-Sardinia"    "Prussia"
##  [82] "South Korea"          "Moldova"              "Romania"
##  [85] "Russia"               "Rwanda"               "St. Helena"
##  [88] "St. Lucia"            "Samoa"                "São Tomé & Príncipe"
##  [91] "Sardinia"             "Saudi Arabia"         "Serbia"
##  [94] "Sierra Leone"         "Singapore"            "Slovakia"
##  [97] "Slovenia"             "Somalia"              "South Africa"
## [100] "Sri Lanka"            "Suriname"             "Syria"
## [103] "Macedonia"            "Timor-Leste"          "Tonga"
## [106] "Tunisia"              "Uganda"               "Ukraine"
## [109] "Tanzania"             "Venezuela"            "Wallis & Futuna"
## [112] "Western Sahara"       "Yugoslavia"           "Zambia"
## [115] "Zimbabwe"
```

```
# Groups
# Find all sentences that include words: "the", "a" or "an"
str_subset(sentences, "(\\sthe\\s|\\sa\\s|\\san\\s)")
```

```
##   [1] "The birch canoe slid on the smooth planks."
##   [2] "Glue the sheet to the dark blue background."
##   [3] "It's easy to tell the depth of a well."
##   [4] "These days a chicken leg is a rare dish."
##   [5] "The box was thrown beside the parked truck."
##   [6] "The boy was there when the sun rose."
##   [7] "The source of the huge river is the clear spring."
##   [8] "Kick the ball straight and follow through."
##   [9] "Help the woman get back to her feet."
##  [10] "A pot of tea helps to pass the evening."
##  [11] "The soft cushion broke the man's fall."
##  [12] "The salt breeze came across from the sea."
##  [13] "The girl at the booth sold fifty bonds."
##  [14] "The small pup gnawed a hole in the sock."
##  [15] "The fish twisted and turned on the bent hook."
##  [16] "Press the pants and sew a button on the vest."
##  [17] "The beauty of the view stunned the young boy."
##  [18] "Two blue fish swam in the tank."
##  [19] "The colt reared and threw the tall rider."
##  [20] "It snowed, rained, and hailed the same morning."
##  [21] "Hoist the load to your left shoulder."
##  [22] "Take the winding path to reach the lake."
##  [23] "Note closely the size of the gas tank."
##  [24] "Wipe the grease off his dirty face."
##  [25] "Mend the coat before you go out."
##  [26] "The meal was cooked before the bell rang."
##  [27] "A king ruled the state in the early days."
##  [28] "The ship was torn apart on the sharp reef."
##  [29] "Sickness kept him home the third week."
##  [30] "The wide road shimmered in the hot sun."
##  [31] "The lazy cow lay in the cool grass."
##  [32] "Lift the square stone over the fence."
##  [33] "The rope will bind the seven books at once."
##  [34] "Hop over the fence and plunge in."
##  [35] "The friendly gang left the drug store."
##  [36] "The frosty air passed through the coat."
##  [37] "The crooked maze failed to fool the mouse."
##  [38] "The show was a flop from the very start."
##  [39] "A saw is a tool used for making boards."
##  [40] "March the soldiers past the next hill."
##  [41] "Place a rosebush near the porch steps."
##  [42] "Both lost their lives in the raging storm."
##  [43] "We talked of the slide show in the circus."
##  [44] "Use a pencil to write the first draft."
##  [45] "He ran half way to the hardware store."
##  [46] "The clock struck to mark the third period."
##  [47] "A small creek cut across the field."
##  [48] "The set of china hit, the floor with a crash."
##  [49] "This is a grand season for hikes on the road."
##  [50] "The dune rose from the edge of the water."
##  [51] "Those words were the cue for the actor to leave."
##  [52] "A yacht slid around the point into the bay."
##  [53] "The two met while playing on the sand."
##  [54] "The ink stain dried on the finished page."
##  [55] "The walled town was seized without a fight."
##  [56] "A tame squirrel makes a nice pet."
##  [57] "The horn of the car woke the sleeping cop."
##  [58] "The pearl was worn in a thin silver ring."
##  [59] "The Navy attacked the big task force."
##  [60] "See the cat glaring at the scared mouse."
##  [61] "The grass curled around the fence post."
##  [62] "Cut the pie into large parts."
##  [63] "Always close the barn door tight."
##  [64] "He lay prone and hardly moved a limb."
##  [65] "The slush lay deep along the street."
##  [66] "A wisp of cloud hung in the blue air."
##  [67] "The fin was sharp and cut the clear water."
##  [68] "Bail the boat, to stop it from sinking."
##  [69] "Cats and dogs each hate the other."
##  [70] "Open the crate but don't break the glass."
##  [71] "Add the sum to the product of these three."
##  [72] "The hog crawled under the high fence."
##  [73] "Move the vat over the hot fire."
##  [74] "The bark of the pine tree was shiny and dark."
##  [75] "Leaves turn brown and yellow in the fall."
##  [76] "The pennant waved when the wind blew."
##  [77] "Split the log with a quick, sharp blow."
##  [78] "Burn peat after the logs give out."
##  [79] "Weave the carpet on the right hand side."
##  [80] "Hemp is a weed found in parts of the tropics."
##  [81] "We find joy in the simplest things."
##  [82] "The harder he tried the less he got done."
##  [83] "The boss ran the show with a watchful eye."
##  [84] "Paste can cleanse the most dirty brass."
##  [85] "It caught its hind paw in a rusty trap."
##  [86] "The wharf could be seen at the farther shore."
##  [87] "Feel the heat of the weak dying flame."
##  [88] "A cramp is no small danger on a swim."
##  [89] "He said the same phrase thirty times."
##  [90] "Pluck the bright rose without leaves."
##  [91] "The glow deepened in the eyes of the sweet girl."
##  [92] "Bring your problems to the wise chief."
##  [93] "Write a fond note to the friend you cherish."
##  [94] "We frown when events take a bad turn."
##  [95] "Port is a strong wine with s smoky taste."
##  [96] "The young kid jumped the rusty gate."
##  [97] "Guess the results from the first scores."
##  [98] "The just claim got the right verdict."
##  [99] "These thistles bend in a high wind."
## [100] "The tree top waved in a graceful way."
## [101] "The spot on the blotter was made by green ink."
## [102] "Mud was spattered on the front of his white shirt."
## [103] "The cigar burned a hole in the desk top."
## [104] "The empty flask stood on the tin tray."
## [105] "He broke a new shoelace that day."
## [106] "The coffee stand is too high for the couch."
## [107] "The pirates seized the crew of the lost ship."
## [108] "We tried to replace the coin but failed."
## [109] "She sewed the torn coat quite neatly."
## [110] "The jacket hung on the back of the wide chair."
## [111] "At that high level the air is pure."
## [112] "Drop the two when you add the figures."
## [113] "An abrupt start does not win the prize."
## [114] "The office paint was a dull sad tan."
## [115] "He knew the skill of the great young actress."
## [116] "A shower of dirt fell from the hot pipes."
## [117] "Steam hissed from the broken valve."
## [118] "The child almost hurt the small dog."
## [119] "There was a sound of dry leaves outside."
## [120] "Torn scraps littered the stone floor."
## [121] "Sunday is the best part of the week."
## [122] "They felt gay when the ship arrived in port."
## [123] "Add the store's account to the last cent."
## [124] "The third act was dull and tired the players."
## [125] "Add the column and put the sum here."
## [126] "We admire and love a good cook."
## [127] "There the flood mark is ten inches."
## [128] "He carved a head from the round block of marble."
## [129] "The fruit of a fig tree is apple-shaped."
## [130] "Corn cobs can be used to kindle a fire."
## [131] "Where were they when the noise started."
## [132] "Sell your gift to a buyer at a good gain."
## [133] "The tongs lay beside the ice pail."
## [134] "The petals fall with the next puff of wind."
## [135] "Bring your best compass to the third class."
## [136] "Farmers came in to thresh the oat crop."
## [137] "The brown house was on fire to the attic."
## [138] "Float the soap on top of the bath water."
## [139] "A blue crane is a tall wading bird."
## [140] "The club rented the rink for the fifth night."
## [141] "After the dance they went straight home."
## [142] "The hostess taught the new maid to serve."
## [143] "He wrote his last novel there at the inn."
## [144] "Even the worst will beat his low score."
## [145] "The loss of the second ship was hard to take."
## [146] "The fly made its way along the wall."
## [147] "Do that with a wooden stick."
## [148] "The wreck occurred by the bank on Main Street."
## [149] "Coax a young calf to drink from a bucket."
## [150] "The lamp shone with a steady green flame."
## [151] "They took the axe and the saw to the forest."
## [152] "The shaky barn fell with a loud crash."
## [153] "Rake the rubbish up and then burn it."
## [154] "Slash the gold cloth into fine ribbons."
## [155] "Try to have the court decide the case."
## [156] "They floated on the raft to sun their white backs."
## [157] "The map had an X that meant nothing."
## [158] "Jerk the rope and the bell rings weakly."
## [159] "Madam, this is the best brand of corn."
## [160] "On the islands the sea breeze is soft and mild."
## [161] "This will lead the world to more sound and fury"
## [162] "Add salt before you fry the egg."
## [163] "The box is held by a bright red snapper."
## [164] "Jump the fence and hurry up the bank."
## [165] "Yell and clap as the curtain slides back."
## [166] "They are men nho walk the middle of the road."
## [167] "Both brothers wear the same size."
## [168] "Ducks fly north but lack a compass."
## [169] "The dark pot hung in the front closet."
## [170] "Carry the pail to the wall and spill it there."
## [171] "The train brought our hero to the big town."
## [172] "The rude laugh filled the empty room."
## [173] "Tea served from the brown jug is tasty."
## [174] "A zestful food is the hot-cross bun."
## [175] "The horse trotted around the field at a brisk pace."
## [176] "Find the twin who stole the pearl necklace."
## [177] "Cut the cord that binds the box tightly."
## [178] "The red tape bound the smuggled food."
## [179] "Look in the corner to find the tan shirt."
## [180] "The cold drizzle will halt the bond drive."
## [181] "Nine men were hired to dig the ruins."
## [182] "The junk yard had a mouldy smell."
## [183] "The flint sputtered and lit a pine torch."
## [184] "Soak the cloth and drown the sharp odor."
## [185] "A joy to every child is the swan boat."
## [186] "All sat frozen and watched the screen."
## [187] "To reach the end he needs much courage."
## [188] "Shape the clay gently into block form."
## [189] "The ridge on a smooth surface is a bump or flaw."
## [190] "Quench your thirst, then eat the crackers."
## [191] "The mute muffled the high tones of the horn."
## [192] "The gold ring fits only a pierced ear."
## [193] "Watch the log float in the wide river."
## [194] "The node on the stalk of wheat grew daily."
## [195] "The barrel of beer was a brew of malt and hops."
## [196] "Slide the box into that empty space."
## [197] "The plant grew large and green in the window."
## [198] "The beam dropped down on the workmen's head."
## [199] "Pink clouds floated JTith the breeze."
## [200] "She danced like a swan, tall and graceful."
## [201] "The tube was blown and the tire flat and useless."
## [202] "It is late morning on the old wall clock."
## [203] "Let's all join as we sing the last chorus."
## [204] "The rise to fame of a person takes luck."
## [205] "The quick fox jumped on the sleeping cat."
## [206] "The nozzle of the fire hose was bright brass."
## [207] "Screw the round cap on as tight as needed."
## [208] "Fill the ink jar with sticky glue."
## [209] "He smoke a big pipe with strong contents."
## [210] "Pack the records in a neat thin case."
## [211] "The crunch of feet in the snow was the only sound."
## [212] "The copper bowl shone in the sun's rays."
## [213] "The plush chair leaned against the wall."
## [214] "Bathe and relax in the cool green grass."
## [215] "The kitten chased the dog down the street."
## [216] "Pages bound in cloth make a book."
## [217] "Try to trace the fine lines of the painting."
## [218] "Women form less than half of the group."
## [219] "The zones merge in the central part of town."
## [220] "A gem in the rough needs work to polish."
## [221] "Most of the new is easy for us to hear."
## [222] "He used the lathe to make brass objects."
## [223] "The vane on top of the pole revolved in the wind."
## [224] "Mince pie is a dish served to children."
## [225] "A child's wit saved the day for us."
## [226] "Tack the strip of carpet to the worn floor."
## [227] "Pour the stew from the pot into the plate."
## [228] "The man went to the woods to gather sticks."
## [229] "The dirt piles were lines along the road."
## [230] "The logs fell and tumbled into the clear stream."
## [231] "A ripe plum is fit for a king's palate."
## [232] "It takes a good trap to capture a bear."
## [233] "Feed the white mouse some flower seeds."
## [234] "The thaw came early and freed the stream."
## [235] "He took the lead and kept it the whole distance."
## [236] "The key you designed will fit the lock."
## [237] "Plead to the council to free the poor thief."
## [238] "The lake sparkled in the red hot sun."
## [239] "He crawled with care along the ledge."
## [240] "Tend the sheep while the dog wanders."
## [241] "It takes a lot of help to finish these."
## [242] "Mark the spot with a sign painted red."
## [243] "Take two shares as a fair profit."
## [244] "That move means the game is over."
## [245] "He wrote down a long list of items."
## [246] "A siege will crack the strong defense."
## [247] "The drip of the rain made a pleasant sound."
## [248] "Serve the hot rum to the tired heroes."
## [249] "Much of the story makes good sense."
## [250] "The sun came up to light the eastern sky."
## [251] "Heave the line over the port side."
## [252] "It's a dense crowd in two distinct ways."
## [253] "His hip struck the knee of the next player."
## [254] "The desk was firm on the shaky floor."
## [255] "It takes heat to bring out the odor."
## [256] "Raise the sail and steer the ship northward."
## [257] "Jerk the dart from the cork target."
## [258] "We now have a new base for shipping."
## [259] "The list of names is carved around the base."
## [260] "The sheep were led home by a dog."
## [261] "Three for a dime, the young peddler cried."
## [262] "Once we stood beside the shore."
## [263] "A chink in the wall allowed a draft to blow."
## [264] "He takes the oath of office each March."
## [265] "The sand drifts over the sill of the old house."
## [266] "The point of the steel pen was bent and twisted."
## [267] "There is a lag between thought and act."
## [268] "Seed is needed to plant the spring corn."
## [269] "Draw the chart with heavy black lines."
## [270] "The chap slipped into the crowd and was lost."
## [271] "The ramp led up to the wide highway."
## [272] "Beat the dust from the rug onto the lawn."
## [273] "Screen the porch with woven straw mats."
## [274] "This horse will nose his way to the finish."
## [275] "The dry wax protects the deep scratch."
## [276] "He picked up the dice for a second roll."
## [277] "The nag pulled the frail cart along."
## [278] "Twist the valve and release hot steam."
## [279] "The vamp of the shoe had a gold buckle."
## [280] "They slice the sausage thin with a knife."
## [281] "The bloom of the rose lasts a few days."
## [282] "A gray mare walked before the colt."
## [283] "Breakfast buns are fine with a hot drink."
## [284] "The man wore a feather in his felt hat."
## [285] "He wheeled the bike past. the winding road."
## [286] "Drop the ashes on the worn old rug."
## [287] "Throw out the used paper cup and plate."
## [288] "A clean neck means a neat collar."
## [289] "The stems of the tall glasses cracked and broke."
## [290] "The clothes dried on a thin wooden rack."
## [291] "Turn on the lantern which gives us light."
## [292] "The cleat sank deeply into the soft turf."
## [293] "The bills were mailed promptly on the tenth of the month."
## [294] "The price is fair for a good antique clock."
## [295] "Dispense with a vest on a day like this."
## [296] "He sent the figs, but kept the ripe cherries."
## [297] "The hinge on the door creaked with old age."
## [298] "The screen before the fire kept in the sparks."
## [299] "Thick glasses helped him read the print."
## [300] "Birth and death mark the limits of life."
## [301] "The kite flew wildly in the high wind."
## [302] "We need an end of all such matter."
## [303] "The case was puzzling to the old and wise."
## [304] "The bright lanterns were gay on the dark lawn."
## [305] "Five years he lived with a shaggy dog."
## [306] "A fence cuts through the corner lot."
## [307] "Shut the hatch before the waves push it in."
## [308] "Crack the walnut with your sharp side teeth."
## [309] "He offered proof in the form of a lsrge chart."
## [310] "Send the stuff in a thick paper bag."
## [311] "A quart of milk is water for the most part."
## [312] "In the rear of the ground floor was a large passage."
## [313] "A man in a blue sweater sat at the desk."
## [314] "Oats are a food eaten by horse and man."
## [315] "Tuck the sheet under the edge of the mat."
## [316] "A force equal to that would move the earth."
## [317] "The work of the tailor is seen on each side."
## [318] "Take a chance and win a china doll."
## [319] "Shake the dust from your shoes, stranger."
## [320] "The dusty bench stood by the stone wall."
## [321] "We dress to suit the weather of most days."
## [322] "The water in this well is a source of good health."
## [323] "That guy is the writer of a few banned books."
## [324] "Ripe pears are fit for a queen's table."
## [325] "A big wet stain was on the round carpet."
## [326] "The room was crowded with a wild mob."
## [327] "She blushed when he gave her a white orchid."
## [328] "The beetle droned in the hot June sun."
## [329] "Press the pedal with your left foot."
## [330] "The black trunk fell from the landing."
## [331] "The bank pressed for payment of the debt."
## [332] "The theft of the pearl pin was kept secret."
## [333] "The vast space stretched into the far distance."
## [334] "Flax makes a fine brand of paper."
## [335] "Hurdle the pit with the aid of a long pole."
## [336] "Even a just cause needs power to win."
## [337] "Peep under the tent and see the clowns."
## [338] "The leaf drifts along with a slow spin."
## [339] "Flood the mails with requests for this book."
## [340] "Those last words were a strong statement."
## [341] "He wrote his name boldly at the top of tile sheet."
## [342] "Down that road is the way to the grain farmer."
## [343] "At night the alarm roused him from a deep sleep."
## [344] "Read just what the meter says."
## [345] "Fill your pack with bright trinkets for the poor."
## [346] "The line where the edges join was clean."
## [347] "Breathe deep and smell the piny air."
## [348] "A toad and a frog are hard to tell apart."
## [349] "A break in the dam almost caused a flood."
## [350] "Paint the sockets in the wall dull green."
## [351] "The child crawled into the dense grass."
## [352] "Trample the spark, else the flames will spread."
## [353] "The hilt. of the sword was carved with fine designs."
## [354] "A round hole was drilled through the thin board."
## [355] "Footprints showed the path he took up the beach."
## [356] "A vent near the edge brought in fresh air."
## [357] "Prod the old mule with a crooked stick."
## [358] "It is a band of steel three inches wide."
## [359] "The pipe ran almost the length of the ditch."
## [360] "It was hidden from sight by a mass of leaves and shrubs."
## [361] "The weight. of the package was seen on the high scale."
## [362] "Wake and rise, and step into the green outdoors."
## [363] "The green light in the brown box flickered."
## [364] "The brass tube circled the high wall."
## [365] "Hold the hammer near the end to drive the nail."
## [366] "Next Sunday is the twelfth of the month."
## [367] "He put his last cartridge into the gun and fired."
## [368] "They took their kids from the public school."
## [369] "Drive the screw straight into the wood."
## [370] "Keep the hatch tight and the watch constant."
## [371] "Sever the twine with a quick snip of the knife."
## [372] "Slide the catch back and open the desk."
## [373] "Help the weak to preserve their strength."
## [374] "Stop whistling and watch the boys march."
## [375] "Jerk the cord, and out tumbles the gold."
## [376] "Slidc the tray across the glass top."
## [377] "The cloud moved in a stately way and was gone."
## [378] "Light maple makes for a swell room."
## [379] "Set the piece here and say nothing."
## [380] "Get the trust fund to the bank early."
## [381] "Choose between the high road and the low."
## [382] "He lent his coat to the tall gaunt stranger."
## [383] "There is a strong chance it will happen once more."
## [384] "The duke left the park in a silver coach."
## [385] "Greet the new guests and leave quickly."
## [386] "When the frost has come it is time for turkey."
## [387] "A thin stripe runs down the middle."
## [388] "A six comes up more often than a ten."
## [389] "Lush fern grow on the lofty rocks."
## [390] "The ram scared the school children off."
## [391] "The team with the best timing looks good."
## [392] "The farmer swapped his horse for a brown ox."
## [393] "Sit on the perch and tell the others what to do."
## [394] "Green moss grows on the northern side."
## [395] "Tea in thin china has a sweet taste."
## [396] "Pitch the straw through the door of the stable."
## [397] "The latch on the beck gate needed a nail."
## [398] "The goose was brought straight from the old market."
## [399] "The sink is the thing in which we pile dishes."
## [400] "A whiff of it will cure the most stubborn cold."
## [401] "She flaps her cape as she parades the street."
## [402] "The loss of the cruiser was a blow to the fleet."
## [403] "Loop the braid to the left and then over."
## [404] "Plead with the lawyer to drop the lost cause."
## [405] "Tear a thin sheet from the yellow pad."
## [406] "A cruise in warm waters in a sleek yacht is fun."
## [407] "A streak of color ran down the left edge."
## [408] "It was done before the boy could see it."
## [409] "Crouch before you jump or miss the mark."
## [410] "Pack the kits and don\u0092t forget the salt."
## [411] "The square peg will settle in the round hole."
## [412] "Bad nerves are jangled by a door slam."
## [413] "They sang the same tunes at each party."
## [414] "The sky in the west is tinged with orange red."
## [415] "The horse balked and threw the tall rider."
## [416] "The hitch between the horse and cart broke."
## [417] "Pile the coal high in the shed corner."
## [418] "The rarest spice comes from the far East."
## [419] "The roof should be tilted at a sharp slant."
## [420] "The mule trod the treadmill day and night."
## [421] "The aim of the contest is to raise a great fund."
## [422] "There is a fine hard tang in salty air."
## [423] "Cod is the main business of the north shore."
## [424] "Dunk the stale biscuits into strong drink."
## [425] "Cap the jar with a tight brass cover."
## [426] "The poor boy missed the boat again."
## [427] "Be sure to set the lamp firmly in the hole."
## [428] "Pick a card and slip it. under the pack."
## [429] "A round mat will cover the dull spot."
## [430] "The first part of the plan needs changing."
## [431] "You cannot brew tea in a cold pot."
## [432] "Dots of light betrayed the black cat."
## [433] "Put the chart on the mantel and tack it down."
## [434] "The red paper brightened the dim stage."
## [435] "See the player scoot to third base."
## [436] "Slide the bill between the two leaves."
## [437] "Many hands help get the job done."
## [438] "No doubt about the way the wind blows."
## [439] "Dig deep in the earth for pirate's gold."
## [440] "The steady drip is worse than a drenching rain."
## [441] "Green ice frosted the punch bowl."
## [442] "A stuffed chair slipped from the moving van."
## [443] "A thin book fits in the side pocket."
## [444] "The hail pattered on the burnt brown grass."
## [445] "The store was jammed before the sale could start."
## [446] "It was a bad error on the part of the new judge."
## [447] "One step more and the board will collapse."
## [448] "Take the match and strike it against your shoe."
## [449] "The pot boiled, but the contents failed to jell."
## [450] "The bombs left most of the town in ruins."
## [451] "Stop and stare at the hard working man."
## [452] "The pup jerked the leash as he saw a feline shape."
## [453] "Open your book to the first page."
## [454] "Fish evade the net, and swim off."
## [455] "Dip the pail once and let it settle."
## [456] "The big red apple fell to the ground."
## [457] "The curtain rose and the show was on."
## [458] "The young prince became heir to the throne."
## [459] "He sent the boy on a short errand."
## [460] "The long journey home took a year."
## [461] "She saw a cat in the neighbor's house."
## [462] "A pink shell was found on the sandy beach."
## [463] "A severe storm tore down the barn."
## [464] "When you hear the bell, come quickly."
```

```r
str_view_all(sentences, "(\\sthe\\s|\\sa\\s|\\san\\s)", match = T)
```

```
##  [1] | The birch canoe slid on< the >smooth planks.
##  [2] | Glue< the >sheet to< the >dark blue background.
##  [3] | It's easy to tell< the >depth of< a >well.
##  [4] | These days< a >chicken leg is< a >rare dish.
##  [7] | The box was thrown beside< the >parked truck.
## [11] | The boy was there when< the >sun rose.
## [13] | The source of< the >huge river is< the >clear spring.
## [14] | Kick< the >ball straight and follow through.
## [15] | Help< the >woman get back to her feet.
## [16] | A pot of tea helps to pass< the >evening.
## [18] | The soft cushion broke< the >man's fall.
## [19] | The salt breeze came across from< the >sea.
## [20] | The girl at< the >booth sold fifty bonds.
## [21] | The small pup gnawed< a >hole in< the >sock.
## [22] | The fish twisted and turned on< the >bent hook.
## [23] | Press< the >pants and sew< a >button on< the >vest.
## [25] | The beauty of< the >view stunned< the >young boy.
## [26] | Two blue fish swam in< the >tank.
## [28] | The colt reared and threw< the >tall rider.
## [29] | It snowed, rained, and hailed< the >same morning.
## ... and 444 more
```

```r
# Find words with repeated pair of letters (two letters must be repeated): use back references
str_subset(words, "(..)\\1")   # \1 is a group reference 1st group, double backslash ~ escaping
```

```
## [1] "competition" "competitive" "crisis"      "dining"      "remaining"
## [6] "remember"    "training"
```

```r
str_view_all(words, "(..)\\1", match = T)
```

```
##  [526] | compe<titi>on
##  [527] | compe<titi>ve
##  [649] | cr<isis>
##  [763] | d<inin>g
## [2207] | rema<inin>g
## [2209] | r<emem>ber
## [2766] | tra<inin>g
```

```r
str_subset(fruit, "(..)\\1")
```

```
## [1] "banana"      "coconut"     "cucumber"    "jujube"      "papaya"     
## [6] "salal berry"
```

```r
str_view_all(fruit, "(..)\\1", match = T)
```

```
##  [4] | b<anan>a
## [20] | <coco>nut
## [22] | <cucu>mber
## [41] | <juju>be
## [56] | <papa>ya
## [73] | s<alal> berry
```

```r
# Mor ethan one greoup in back reference
string <- c("abc", "abcabc", "ababcc", "abababccc")
string
```

```
## [1] "abc"       "abcabc"    "ababcc"    "abababccc"
```

```r
str_view_all(string, "(a)(b)", match = T)             # ab
```

```
## [1] | <ab>c
## [2] | <ab>c<ab>c
## [3] | <ab><ab>cc
## [4] | <ab><ab><ab>ccc
```

```r
str_view_all(string, "(a)(b)\\1", match = T)          # aba
```

```
## [3] | <aba>bcc
## [4] | <aba>babccc
```

```r
str_view_all(string, "(a)(b)\\1\\2", match = T)       # abab
```

```
## [3] | <abab>cc
## [4] | <abab>abccc
```

```r
str_view_all(string, "(a)(b)\\1\\2\\1\\2", match = T) # ababab
```

```
## [4] | <ababab>ccc
```

Regex: Look arounds & quantifiers

```r
# Look arounds
# Find a word where letter "w" is followed by letter "a"
str_subset(words, "w(?=a)")
```

```
##  [1] "always"    "anyway"    "award"     "aware"     "awareness" "away"     
##  [7] "forward"   "highway"   "software"  "toward"    "towards"   "wage"
## [13] "wait"      "wake"      "walk"      "wall"      "wander"    "want"
## [19] "war"       "warm"      "warn"      "warning"   "wash"      "waste"
## [25] "watch"     "water"     "wave"      "way"
```

```
str_view_all(words, "w(?=a)", match = T)
```

```
##  [109] | al<w>ays
##  [136] | any<w>ay
##  [217] | a<w>ard
##  [218] | a<w>are
##  [219] | a<w>areness
##  [220] | a<w>ay
## [1106] | for<w>ard
## [1268] | high<w>ay
## [2465] | soft<w>are
## [2752] | to<w>ard
## [2753] | to<w>ards
## [2886] | <w>age
## [2887] | <w>ait
## [2888] | <w>ake
## [2889] | <w>alk
## [2890] | <w>all
## [2891] | <w>ander
## [2892] | <w>ant
## [2893] | <w>ar
## [2894] | <w>arm
## ... and 8 more
```

```
# Find a word where letter "w" is not followed by letter "a"
str_subset(words, "w(?!a)")
```

```
##   [1] "acknowledge" "allow"       "answer"      "anywhere"    "awful"
##   [6] "below"       "between"     "blow"        "borrow"      "bowl"
##  [11] "brown"       "cow"         "crew"        "crowd"       "down"
##  [16] "downtown"    "draw"        "drawing"     "elsewhere"   "everywhere"
##  [21] "fellow"      "few"         "fewer"       "flow"        "flower"
##  [26] "follow"      "following"   "framework"   "grow"        "growing"
##  [31] "growth"      "how"         "however"     "interview"   "Jew"
##  [36] "Jewish"      "know"        "knowledge"   "law"         "lawn"
##  [41] "lawsuit"     "lawyer"      "low"         "lower"       "meanwhile"
##  [46] "narrow"      "network"     "new"         "newly"       "news"
##  [51] "newspaper"   "now"         "nowhere"     "otherwise"   "owe"
##  [56] "own"         "owner"       "powder"      "power"       "powerful"
##  [61] "raw"         "review"      "row"         "shadow"      "show"
##  [66] "shower"      "slow"        "slowly"      "snow"        "somehow"
##  [71] "somewhat"    "somewhere"   "swear"       "sweep"       "sweet"
##  [76] "swim"        "swing"       "switch"      "throw"       "tomorrow"
##  [81] "tower"       "town"        "twelve"      "twenty"      "twice"
##  [86] "twin"        "two"         "unknown"     "view"        "viewer"
##  [91] "we"          "weak"        "wealth"      "wealthy"     "weapon"
##  [96] "wear"        "weather"     "wedding"     "week"        "weekend"
## [101] "weekly"      "weigh"       "weight"      "welcome"     "welfare"
## [106] "well"        "west"        "western"     "wet"         "what"
## [111] "whatever"    "wheel"       "when"        "whenever"    "where"
## [116] "whereas"     "whether"     "which"       "while"       "whisper"
## [121] "white"       "who"         "whole"       "whom"        "whose"
## [126] "why"          "wide"         "widely"       "widespread"   "wife"
## [131] "wild"         "will"         "willing"      "win"          "wind"
## [136] "window"       "wine"         "wing"         "winner"       "winter"
## [141] "wipe"         "wire"         "wisdom"       "wise"         "wish"
## [146] "with"         "withdraw"     "within"       "without"      "witness"
## [151] "woman"        "wonder"       "wonderful"    "wood"         "wooden"
## [156] "word"         "work"         "worker"       "working"      "works"
## [161] "workshop"     "world"        "worried"      "worry"        "worth"
## [166] "would"        "wound"        "wrap"         "write"        "writer"
## [171] "writing"      "wrong"        "yellow"
```

```
str_view_all(words, "w(?!a)", match = T)
```

```
##  [27] | ackno<w>ledge
##  [99] | allo<w>
## [128] | ans<w>er
## [137] | any<w>here
## [221] | a<w>ful
## [270] | belo<w>
## [281] | bet<w>een
## [302] | blo<w>
## [316] | borro<w>
## [323] | bo<w>l
## [347] | bro<w>n
## [635] | co<w>
## [646] | cre<w>
## [657] | cro<w>d
## [812] | do<w>n
## [813] | do<w>nto<w>n
## [820] | dra<w>
## [821] | dra<w>ing
## [878] | else<w>here
## [950] | every<w>here
## ... and 153 more
```

```
# Find a word where letter "a" is preceded by letter "w"
str_subset(words, "(?<=w)a")
```

```
##  [1] "always"    "anyway"    "award"     "aware"     "awareness" "away"
##  [7] "forward"   "highway"   "software"  "toward"    "towards"   "wage"
## [13] "wait"      "wake"      "walk"      "wall"      "wander"    "want"
## [19] "war"       "warm"      "warn"      "warning"   "wash"      "waste"
## [25] "watch"     "water"     "wave"      "way"
```

```
str_view_all(words, "(?<=w)a", match = T)
```

```
##  [109] | alw<a>ys
##  [136] | anyw<a>y
##  [217] | aw<a>rd
##  [218] | aw<a>re
##  [219] | aw<a>reness
##  [220] | aw<a>y
## [1106] | forw<a>rd
## [1268] | highw<a>y
## [2465] | softw<a>re
## [2752] | tow<a>rd
## [2753] | tow<a>rds
## [2886] | w<a>ge
## [2887] | w<a>it
## [2888] | w<a>ke
## [2889] | w<a>lk
## [2890] | w<a>ll
## [2891] | w<a>nder
## [2892] | w<a>nt
## [2893] | w<a>r
## [2894] | w<a>rm
## ... and 8 more
```

```
# Find a word where letter "a" is not preceded by letter "w"
str_subset(words, "(?<!w)a")
```

```
##    [1] "a"                "abandon"          "ability"
##    [4] "able"             "abortion"         "about"
##    [7] "above"            "abroad"           "absence"
##   [10] "absolute"         "absolutely"       "absorb"
##   [13] "abuse"            "academic"         "accept"
##   [16] "access"           "accident"         "accompany"
##   [19] "accomplish"       "according"        "account"
##   [22] "accurate"         "accuse"           "achieve"
##   [25] "achievement"      "acid"             "acknowledge"
##   [28] "acquire"          "across"           "act"
##   [31] "action"           "active"           "activist"
##   [34] "activity"         "actor"            "actress"
##   [37] "actual"           "actually"         "ad"
##   [40] "adapt"            "add"              "addition"
##   [43] "additional"       "address"          "adequate"
##   [46] "adjust"           "adjustment"       "administration"
##   [49] "administrator"    "admire"           "admission"
##   [52] "admit"            "adolescent"       "adopt"
##   [55] "adult"            "advance"          "advanced"
##   [58] "advantage"        "adventure"        "advertising"
##   [61] "advice"           "advise"           "adviser"
##   [64] "advocate"         "affair"           "affect"
##   [67] "afford"           "afraid"           "African"
##   [70] "African-American" "after"            "afternoon"
##   [73] "again"            "against"          "age"
##   [76] "agency"           "agenda"           "agent"
##   [79] "aggressive"       "ago"              "agree"
##   [82] "agreement"        "agricultural"     "ah"
##   [85] "ahead"            "aid"              "aide"
##   [88] "aim"              "air"              "aircraft"
##   [91] "airline"          "airport"          "album"
##   [94] "alcohol"          "alive"            "all"
##   [97] "alliance"         "allow"            "ally"
##  [100] "almost"           "alone"            "along"
##  [103] "already"          "also"             "alter"
##  [106] "alternative"      "although"         "always"
##  [109] "amazing"          "American"         "among"
##  [112] "amount"           "analysis"         "analyst"
##  [115] "analyze"          "ancient"          "and"
##  [118] "anger"            "angle"            "angry"
##  [121] "animal"           "anniversary"      "announce"
##  [124] "annual"           "another"          "answer"
##  [127] "anticipate"       "anxiety"          "any"
##  [130] "anybody"          "anymore"          "anyone"
##  [133] "anything"         "anyway"           "anywhere"
##  [136] "apart"            "apartment"        "apparent"
##  [139] "apparently"       "appeal"           "appear"
##  [142] "appearance"       "apple"            "application"
##  [145] "apply"            "appoint"          "appointment"
##  [148] "appreciate"       "approach"         "appropriate"
##  [151] "approval"         "approve"          "approximately"
##  [154] "Arab"             "architect"        "area"
##  [157] "argue"            "argument"         "arise"
##  [160] "arm"              "armed"            "army"
##  [163] "around"           "arrange"          "arrangement"
##  [166] "arrest"           "arrival"          "arrive"
##  [169] "art"              "article"          "artist"
##  [172] "artistic"         "as"               "Asian"
##  [175] "aside"            "ask"              "asleep"
##  [178] "aspect"           "assault"          "assert"
##  [181] "assess"           "assessment"       "asset"
##  [184] "assign"           "assignment"       "assist"
##  [187] "assistance"       "assistant"        "associate"
##  [190] "association"      "assume"           "assumption"
##  [193] "assure"           "at"               "athlete"
##  [196] "athletic"         "atmosphere"       "attach"
##  [199] "attack"           "attempt"          "attend"
##  [202] "attention"        "attitude"         "attorney"
##  [205] "attract"          "attractive"       "attribute"
##  [208] "audience"         "author"           "authority"
##  [211] "auto"             "available"        "average"
##  [214] "avoid"            "award"            "aware"
##  [217] "awareness"        "away"             "awful"
##  [220] "baby"             "back"             "background"
##  [223] "bad"              "badly"            "bag"
##  [226] "bake"             "balance"          "ball"
##  [229] "ban"              "band"             "bank"
##  [232] "bar"              "barely"           "barrel"
##  [235] "barrier"          "base"             "baseball"
##  [238] "basic"            "basically"        "basis"
##  [241] "basket"           "basketball"       "bathroom"
##  [244] "battery"          "battle"           "beach"
##  [247] "bean"             "bear"             "beat"
##  [250] "beautiful"        "beauty"           "because"
##  [253] "behavior"         "beneath"          "biological"
##  [256] "birthday"         "black"            "blade"
##  [259] "blame"            "blanket"          "board"
##  [262] "boat"             "boundary"         "brain"
##  [265] "branch"           "brand"            "bread"
##  [268] "break"            "breakfast"        "breast"
##  [271] "breath"           "breathe"          "brilliant"
##  [274] "broad"            "cabin"            "cabinet"
##  [277] "cable"            "cake"             "calculate"
##  [280] "call"             "camera"           "camp"
##  [283] "campaign"         "campus"           "can"
##  [286] "Canadian"         "cancer"           "candidate"
##  [289] "cap"              "capability"       "capable"
##  [292] "capacity"         "capital"          "captain"
##  [295] "capture"          "car"              "carbon"
##  [298] "card"             "care"             "career"
##  [301] "careful"          "carefully"        "carrier"
##  [304] "carry"            "case"             "cash"
##  [307] "cast"             "cat"              "catch"
##  [310] "category"         "Catholic"         "cause"
##  [313] "celebrate"        "celebration"      "central"
##  [316] "certain"          "certainly"        "chain"
##  [319] "chair"            "chairman"         "challenge"
##  [322] "chamber"          "champion"         "championship"
##  [325] "chance"           "change"           "changing"
##  [328] "channel"          "chapter"          "character"
##  [331] "characteristic"   "characterize"     "charge"
##  [334] "charity"          "chart"            "chase"
##  [337] "cheap"            "chemical"         "chocolate"
##  [340] "Christian"        "Christmas"        "cigarette"
##  [343] "circumstance"     "civilian"         "claim"
##  [346] "class"            "classic"          "classroom"
##  [349] "clean"            "clear"            "clearly"
##  [352] "climate"          "clinical"         "coach"
##  [355] "coal"             "coalition"        "coast"
##  [358] "coat"             "collapse"         "colleague"
##  [361] "colonial"         "combination"      "comfortable"
##  [364] "command"          "commander"        "commercial"
##  [367] "communicate"      "communication"    "company"
##  [370] "compare"          "comparison"       "complain"
##  [373] "complaint"        "complicated"      "concentrate"
##  [376] "concentration"    "congressional"    "conservative"
##  [379] "considerable"     "consideration"    "constant"
##  [382] "constantly"       "constitutional"   "consultant"
##  [385] "contact"          "contain"          "container"
##  [388] "contemporary"     "contract"         "contrast"
##  [391] "controversial"    "conventional"     "conversation"
##  [394] "cooperation"      "corporate"        "corporation"
##  [397] "courage"          "coverage"         "crack"
##  [400] "craft"            "crash"            "crazy"
##  [403] "cream"            "create"           "creation"
##  [406] "creative"         "creature"         "criminal"
##  [409] "criteria"         "critical"         "crucial"
##  [412] "cultural"         "dad"              "daily"
##  [415] "damage"           "dance"            "danger"
##  [418] "dangerous"        "dare"             "dark"
##  [421] "darkness"         "data"             "date"
##  [424] "daughter"         "day"              "dead"
##  [427] "deal"             "dealer"           "dear"
##  [430] "death"            "debate"           "decade"
##  [433] "declare"          "decrease"         "defeat"
##  [436] "defendant"        "delay"            "demand"
##  [439] "democracy"        "Democrat"         "democratic"
##  [442] "demonstrate"      "demonstration"    "department"
##  [445] "desperate"        "detail"           "detailed"
##  [448] "dialogue"         "digital"          "disability"
##  [451] "disagree"         "disappear"        "disaster"
##  [454] "discrimination"   "disease"          "display"
##  [457] "distance"         "distant"          "dominant"
##  [460] "dominate"         "draft"            "drag"
##  [463] "drama"            "dramatic"         "dramatically"
##  [466] "draw"             "drawing"          "dream"
##  [469] "each"             "eager"            "ear"
##  [472] "early"            "earn"             "earnings"
##  [475] "earth"            "ease"             "easily"
##  [478] "east"             "eastern"          "easy"
##  [481] "eat"              "educate"          "education"
##  [484] "educational"      "educator"         "elementary"
##  [487] "eliminate"        "e-mail"           "embrace"
##  [490] "emotional"        "emphasis"         "emphasize"
##  [493] "enable"           "encourage"        "engage"
##  [496] "enhance"          "entertainment"    "entrance"
##  [499] "environmental"    "equal"            "equally"
##  [502] "era"              "escape"           "especially"
##  [505] "essay"            "essential"        "essentially"
##  [508] "establish"        "establishment"    "estate"
##  [511] "estimate"         "European"         "evaluate"
##  [514] "evaluation"       "eventually"       "everyday"
##  [517] "exact"            "exactly"          "examination"
##  [520] "examine"          "example"          "exchange"
##  [523] "expand"           "expansion"        "expectation"
##  [526] "explain"          "explanation"      "external"
##  [529] "extra"            "extraordinary"    "fabric"
##  [532] "face"             "facility"         "fact"
##  [535] "factor"           "factory"          "faculty"
##  [538] "fade"             "fail"             "failure"
##  [541] "fair"             "fairly"           "faith"
##  [544] "fall"             "false"            "familiar"
##  [547] "family"           "famous"           "fan"
##  [550] "fantasy"          "far"              "farm"
##  [553] "farmer"           "fashion"          "fast"
##  [556] "fat"              "fate"             "father"
##  [559] "fault"            "favor"            "favorite"
##  [562] "fear"             "feature"          "federal"
##  [565] "female"           "final"            "finally"
##  [568] "finance"          "financial"        "flag"
##  [571] "flame"            "flat"             "flavor"
##  [574] "float"            "football"         "formal"
##  [577] "formation"        "formula"          "foundation"
##  [580] "frame"            "framework"        "frustration"
##  [583] "fundamental"      "funeral"          "gain"
##  [586] "galaxy"           "gallery"          "game"
##  [589] "gang"             "gap"              "garage"
##  [592] "garden"           "garlic"           "gas"
##  [595] "gate"             "gather"           "gay"
##  [598] "gaze"             "gear"             "general"
##  [601] "generally"        "generate"         "generation"
##  [604] "gentleman"        "German"           "giant"
##  [607] "glad"             "glance"           "glass"
##  [610] "global"           "goal"             "grab"
##  [613] "grade"            "gradually"        "graduate"
##  [616] "grain"            "grand"            "grandfather"
##  [619] "grandmother"      "grant"            "grass"
##  [622] "grave"            "gray"             "great"
##  [625] "greatest"         "guarantee"        "guard"
##  [628] "habit"            "habitat"          "hair"
##  [631] "half"             "hall"             "hand"
##  [634] "handful"          "handle"           "hang"
##  [637] "happen"           "happy"            "hard"
##  [640] "hardly"           "hat"              "hate"
##  [643] "have"             "head"             "headline"
##  [646] "headquarters"     "health"           "healthy"
##  [649] "hear"             "hearing"          "heart"
##  [652] "heat"             "heaven"           "heavily"
##  [655] "heavy"            "heritage"         "historian"
##  [658] "historical"       "holiday"          "hospital"
##  [661] "human"            "husband"          "idea"
##  [664] "ideal"            "identification"   "illegal"
##  [667] "illustrate"       "image"            "imagination"
##  [670] "imagine"          "immediate"        "immediately"
##  [673] "immigrant"        "immigration"      "impact"
##  [676] "implication"      "importance"       "important"
##  [679] "incorporate"      "increase"         "increased"
##  [682] "increasing"       "increasingly"     "Indian"
##  [685] "indicate"         "indication"       "individual"
##  [688] "industrial"       "infant"           "inflation"
##  [691] "information"      "initial"          "initially"
##  [694] "initiative"       "install"          "instance"
##  [697] "instead"          "institutional"    "insurance"
##  [700] "intellectual"     "interaction"      "internal"
##  [703] "international"    "interpretation"   "invasion"
##  [706] "investigate"      "investigation"    "investigator"
##  [709] "Iraqi"            "Islamic"          "island"
##  [712] "Israeli"          "Italian"          "jacket"
##  [715] "jail"             "Japanese"         "journal"
##  [718] "journalist"       "lab"              "label"
##  [721] "labor"            "laboratory"       "lack"
##  [724] "lady"             "lake"             "land"
##  [727] "landscape"        "language"         "lap"
##  [730] "large"            "largely"          "last"
##  [733] "late"             "later"            "Latin"
##  [736] "latter"           "laugh"            "launch"
##  [739] "law"              "lawn"             "lawsuit"
##  [742] "lawyer"           "lay"              "layer"
##  [745] "lead"             "leader"           "leadership"
##  [748] "leading"          "leaf"             "league"
##  [751] "lean"             "learn"            "learning"
##  [754] "least"            "leather"          "leave"
##  [757] "legacy"           "legal"            "legislation"
##  [760] "legitimate"       "liberal"          "library"
##  [763] "limitation"       "literally"        "literary"
##  [766] "literature"       "load"             "loan"
##  [769] "local"            "locate"           "location"
##  [772] "machine"          "mad"              "magazine"
##  [775] "mail"             "main"             "mainly"
##  [778] "maintain"         "maintenance"      "major"
##  [781] "majority"         "make"             "maker"
##  [784] "makeup"           "male"             "mall"
##  [787] "man"              "manage"           "management"
##  [790] "manager"          "manner"           "manufacturer"
##  [793] "manufacturing"    "many"             "map"
##  [796] "margin"           "mark"             "market"
##  [799] "marketing"        "marriage"         "married"
##  [802] "marry"            "mask"             "mass"
##  [805] "massive"          "master"           "match"
##  [808] "material"         "math"             "matter"
##  [811] "may"              "maybe"            "mayor"
##  [814] "meal"             "mean"             "meaning"
##  [817] "meanwhile"        "measure"          "measurement"
##  [820] "meat"             "mechanism"        "media"
##  [823] "medical"          "medication"       "mental"
##  [826] "message"          "metal"            "Mexican"
##  [829] "military"         "miracle"          "mistake"
##  [832] "moderate"         "moral"            "mortgage"
##  [835] "motivation"       "mountain"         "musical"
##  [838] "musician"         "mutual"           "naked"
##  [841] "name"             "narrative"        "narrow"
##  [844] "nation"           "national"         "native"
##  [847] "natural"          "naturally"        "nature"
##  [850] "near"             "nearby"           "nearly"
##  [853] "necessarily"      "necessary"        "negative"
##  [856] "negotiate"        "negotiation"      "newspaper"
##  [859] "nomination"       "normal"           "normally"
##  [862] "nuclear"          "obligation"       "observation"
##  [865] "obtain"           "occasion"         "occasionally"
##  [868] "occupation"       "ocean"            "official"
##  [871] "okay"             "operate"          "operating"
##  [874] "operation"        "operator"         "orange"
##  [877] "ordinary"         "organic"          "organization"
##  [880] "organize"         "orientation"      "original"
##  [883] "originally"       "overall"          "pace"
##  [886] "pack"             "package"          "page"
##  [889] "pain"             "painful"          "paint"
##  [892] "painter"          "painting"         "pair"
##  [895] "pale"             "Palestinian"      "palm"
##  [898] "pan"              "panel"            "pant"
##  [901] "paper"            "parent"           "park"
##  [904] "parking"          "part"             "participant"
##  [907] "participate"      "participation"    "particular"
##  [910] "particularly"     "partly"           "partner"
##  [913] "partnership"      "party"            "pass"
##  [916] "passage"          "passenger"        "passion"
##  [919] "past"             "patch"            "path"
##  [922] "patient"          "pattern"          "pause"
##  [925] "pay"              "payment"          "peace"
##  [928] "peak"             "penalty"          "percentage"
##  [931] "performance"      "perhaps"          "permanent"
##  [934] "personal"         "personality"      "personally"
##  [937] "persuade"         "phase"            "photograph"
##  [940] "photographer"     "phrase"           "physical"
##  [943] "physically"       "physician"        "piano"
##  [946] "place"            "plan"             "plane"
##  [949] "planet"           "planning"         "plant"
##  [952] "plastic"          "plate"            "platform"
##  [955] "play"             "player"           "please"
##  [958] "pleasure"         "political"        "politically"
##  [961] "politician"       "popular"          "population"
##  [964] "portrait"         "portray"          "potato"
##  [967] "potential"        "potentially"      "practical"
##  [970] "practice"         "pray"             "prayer"
##  [973] "pregnancy"        "pregnant"         "preparation"
##  [976] "prepare"          "presentation"     "presidential"
##  [979] "primarily"        "primary"          "principal"
##  [982] "privacy"          "private"          "probably"
##  [985] "professional"     "program"          "proposal"
##  [988] "psychological"    "publication"      "purchase"
##  [991] "qualify"          "quality"          "quarter"
##  [994] "quarterback"      "race"             "racial"
##  [997] "radical"          "radio"            "rail"
## [1000] "rain"             "raise"            "range"
## [1003] "rank"             "rapid"            "rapidly"
## [1006] "rare"             "rarely"           "rate"
## [1009] "rather"           "rating"           "ratio"
## [1012] "raw"              "reach"            "react"
## [1015] "reaction"         "read"             "reader"
## [1018] "reading"          "ready"            "real"
## [1021] "reality"          "realize"          "really"
## [1024] "reason"           "reasonable"       "recall"
## [1027] "recommendation"   "regard"           "regarding"
## [1030] "regardless"       "regional"         "regular"
## [1033] "regularly"        "regulate"         "regulation"
## [1036] "relate"           "relation"         "relationship"
## [1039] "relative"         "relatively"       "relax"
## [1042] "release"          "relevant"         "remain"
## [1045] "remaining"        "remarkable"       "repeat"
## [1048] "repeatedly"       "replace"          "representation"
## [1051] "representative"   "Republican"       "reputation"
## [1054] "research"         "researcher"       "reservation"
## [1057] "resistance"       "restaurant"       "retain"
## [1060] "reveal"           "road"             "romantic"
## [1063] "rural"            "Russian"          "sacred"
## [1066] "sad"              "safe"             "safety"
## [1069] "sake"             "salad"            "salary"
## [1072] "sale"             "sales"            "salt"
## [1075] "same"             "sample"           "sanction"
## [1078] "sand"             "satellite"        "satisfaction"
## [1081] "satisfy"          "sauce"            "save"
## [1084] "saving"           "say"              "scale"
## [1087] "scandal"          "scared"           "scenario"
## [1090] "scholar"          "scholarship"      "scream"
## [1093] "sea"              "search"           "season"
## [1096] "seat"             "secretary"        "Senate"
## [1099] "senator"          "separate"         "several"
## [1102] "sexual"           "shade"            "shadow"
## [1105] "shake"            "shall"            "shape"
## [1108] "share"            "sharp"            "signal"
## [1111] "significance"     "significant"      "significantly"
## [1114] "similar"          "similarly"        "situation"
## [1117] "slave"            "small"            "smart"
## [1120] "snap"             "so-called"        "social"
## [1123] "solar"            "somewhat"         "sophisticated"
## [1126] "space"            "Spanish"          "speak"
## [1129] "speaker"          "special"          "specialist"
## [1132] "specifically"     "spiritual"        "spokesman"
## [1135] "spread"           "square"           "stability"
## [1138] "stable"           "staff"            "stage"
## [1141] "stair"            "stake"            "stand"
## [1144] "standard"         "standing"         "star"
## [1147] "stare"            "start"            "state"
## [1150] "statement"        "station"          "statistics"
## [1153] "status"           "stay"             "steady"
## [1156] "steal"            "stomach"          "storage"
## [1159] "straight"         "strange"          "stranger"
## [1162] "strategic"        "strategy"         "stream"
## [1165] "substance"        "substantial"      "sugar"
## [1168] "surface"          "survival"         "sustain"
## [1171] "swear"            "table"            "tablespoon"
## [1174] "tactic"           "tail"             "take"
## [1177] "tale"             "talent"           "talk"
## [1180] "tall"             "tank"             "tap"
## [1183] "tape"             "target"           "task"
## [1186] "taste"            "tax"              "taxpayer"
## [1189] "tea"              "teach"            "teacher"
## [1192] "teaching"         "team"             "tear"
## [1195] "teaspoon"         "technical"        "teenager"
## [1198] "temperature"      "temporary"        "than"
## [1201] "thank"            "thanks"           "that"
## [1204] "theater"          "therapy"          "thousand"
## [1207] "threat"           "threaten"         "throat"
## [1210] "tobacco"          "today"            "tomato"
## [1213] "total"            "totally"          "tournament"
## [1216] "trace"            "track"            "trade"
## [1219] "tradition"        "traditional"      "traffic"
## [1222] "tragedy"          "trail"            "train"
## [1225] "training"         "transfer"         "transform"
## [1228] "transformation"   "transition"       "translate"
## [1231] "transportation"   "travel"           "treat"
## [1234] "treatment"        "treaty"           "trial"
## [1237] "typical"          "typically"        "ultimate"
## [1240] "ultimately"       "unable"           "understand"
## [1243] "understanding"    "unfortunately"    "universal"
## [1246] "unusual"          "urban"            "usual"
## [1249] "usually"          "vacation"         "valley"
## [1252] "valuable"         "value"            "variable"
## [1255] "variation"        "variety"          "various"
## [1258] "vary"             "vast"             "vegetable"
## [1261] "veteran"          "via"              "village"
## [1264] "violate"          "violation"        "virtually"
## [1267] "visual"           "vital"            "vulnerable"
## [1270] "weak"             "wealth"           "wealthy"
## [1273] "weapon"           "wear"             "weather"
## [1276] "welfare"          "what"             "whatever"
## [1279] "whereas"          "widespread"       "withdraw"
## [1282] "woman"            "wrap"             "yard"
## [1285] "yeah"             "year"             "yesterday"
```

```
str_view_all(words, "(?<!w)a", match = T)
```

```
##  [1] | <a>
##  [2] | <a>b<a>ndon
##  [3] | <a>bility
##  [4] | <a>ble
##  [5] | <a>bortion
##  [6] | <a>bout
##  [7] | <a>bove
##  [8] | <a>bro<a>d
##  [9] | <a>bsence
## [10] | <a>bsolute
## [11] | <a>bsolutely
## [12] | <a>bsorb
## [13] | <a>buse
## [14] | <a>c<a>demic
## [15] | <a>ccept
## [16] | <a>ccess
## [17] | <a>ccident
## [18] | <a>ccomp<a>ny
## [19] | <a>ccomplish
## [20] | <a>ccording
## ... and 1267 more
```

```
# Quantifiers
string <- " .A.AA.AAA.AAAA"

# zero or one "A"
str_view_all(string, "A?")
```

```
## [1] | <> <>.<A><>.<A><A><>.<A><A><A><>.<A><A><A><A><>
```

```
# zero or more "A"
str_view_all(string, "A*")
```

```
## [1] | <> <>.<A><>.<AA><>.<AAA><>.<AAAA><>
```

```
# one or more "A"
str_view_all(string, "A+")
```

```
## [1] |  .<A>.<AA>.<AAA>.<AAAA>
```

```
# exactly 2 "A"
str_view_all(string, "A{2}")
```

```
## [1] |  .A.<AA>.<AA>A.<AA><AA>
```

```
# 2 or more "A"
str_view_all(string, "A{2,}")
```

```
## [1] |  .A.<AA>.<AAA>.<AAAA>
```

```
# between 2 and 3 "A"
str_view_all(string, "A{2,3}")
```

```
## [1] |  .A.<AA>.<AAA>.<AAA>A
```

```
# Exercise with sentences
# count the number of words in each sentence
# first remove all punctuation and convert all to lower case
#   then count the number of words and show results
sentences.df1 <- sentences.df %>%
  mutate(sentence = str_remove_all(sentence, "[:punct:]"), # remove punctuation
         sentence = str_to_lower(sentence)) %>%  # convert to lower case
  mutate(`nr words` = str_count(string = sentence, "\\s+") + 1) # counts number of spaces between words

sentences.df1 %>% count(`nr words`) # show frequencies
```

```
## # A tibble: 8 x 2
##   `nr words`     n
##        <dbl> <int>
## 1          5    10
## 2          6    56
## 3          7   188
## 4          8   245
## 5          9   150
## 6         10    54
## 7         11    15
## 8         12     2
```

```
# Countries with more than 3 words in a country name
countries.df %>%
  mutate(`nr words` = str_count(string = country, "\\s+") + 1) %>%
  filter(`nr words` > 3)
```

```
## # A tibble: 13 x 2
##    country                                    `nr words`
##    <chr>                                           <dbl>
##  1 British Indian Ocean Territory                      4
##  2 Heard & McDonald Islands                            4
##  3 Hong Kong SAR China                                 4
##  4 Micronesia (Federated States of)                    4
##  5 St. Kitts & Nevis                                   4
##  6 Saint Martin (French part)                          4
##  7 St. Pierre & Miquelon                               4
##  8 St. Vincent & Grenadines                            4
##  9 São Tomé & Príncipe                                 4
## 10 South Georgia & South Sandwich Islands              6
## 11 Svalbard & Jan Mayen                                4
## 12 Turks & Caicos Islands                              4
## 13 United States Minor Outlying Islands (the)          6
```

Factors ~ forcats

```r
# First lets' create a factor variables
df <- mpg %>%
  mutate_at(.vars = c("manufacturer", "model", "trans", "class"), .funs = as_factor)
str(df)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: Factor w/ 15 levels "audi","chevrolet",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ model       : Factor w/ 38 levels "a4","a4 quattro",..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : Factor w/ 10 levels "auto(l5)","manual(m5)",..: 1 2 3 4 1 2 4 2 1 3 ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : Factor w/ 7 levels "compact","midsize",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
# Check factor levels
df$manufacturer %>% levels()
```

```
##  [1] "audi"       "chevrolet"  "dodge"      "ford"       "honda"
##  [6] "hyundai"    "jeep"       "land rover" "lincoln"    "mercury"
## [11] "nissan"     "pontiac"    "subaru"     "toyota"     "volkswagen"
```

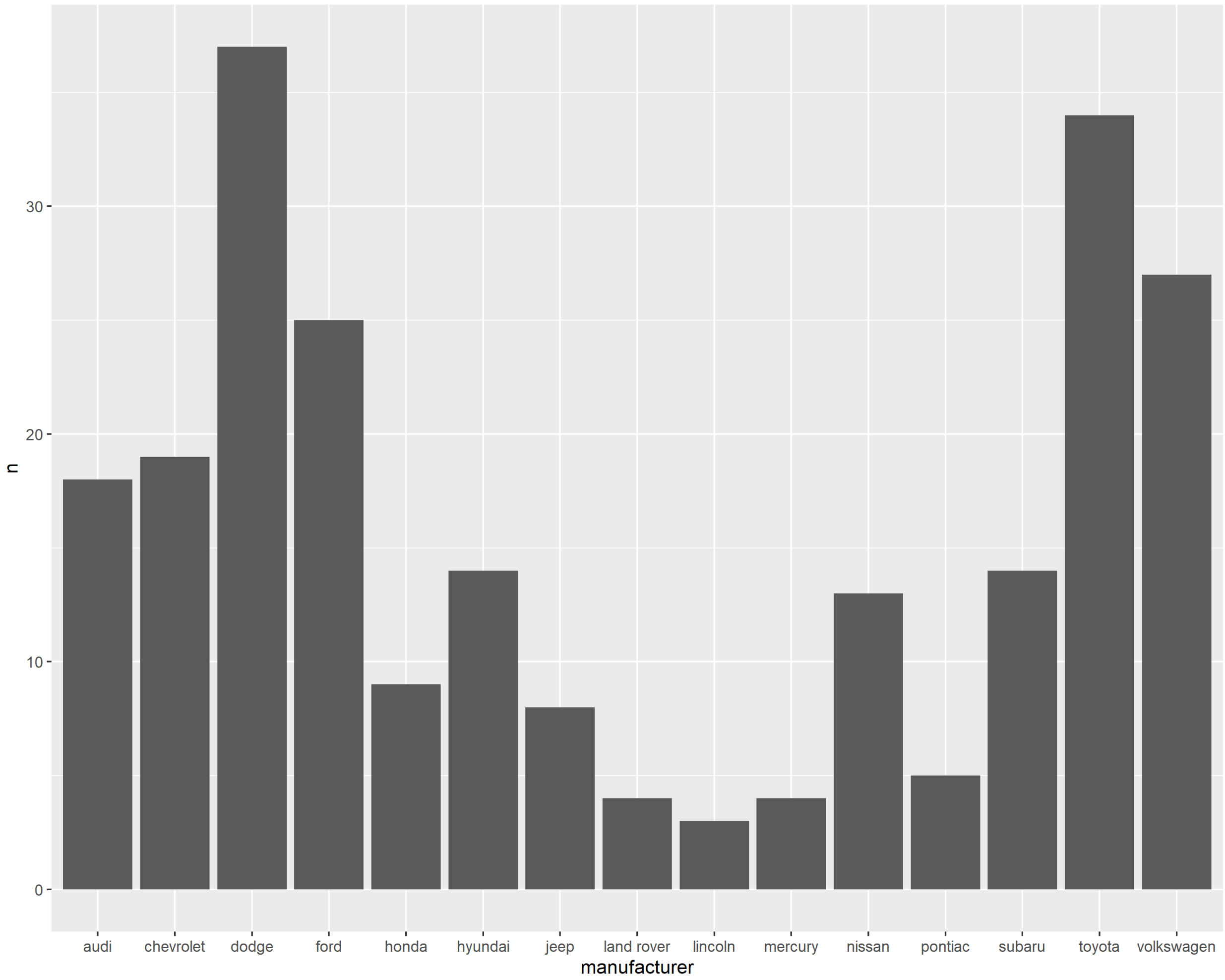
```r
# fct_count() - Count factor values
# Check car manufacturer (frequencies count)
df %>% .$manufacturer %>%  fct_count()
```

```
## # A tibble: 15 x 2
##    f              n
##    <fct>      <int>
##  1 audi          18
##  2 chevrolet     19
##  3 dodge         37
##  4 ford          25
##  5 honda          9
##  6 hyundai       14
##  7 jeep           8
##  8 land rover     4
##  9 lincoln        3
## 10 mercury        4
## 11 nissan        13
## 12 pontiac        5
## 13 subaru        14
## 14 toyota        34
## 15 volkswagen    27
```

```r
df %>% count(manufacturer)
```

```
## # A tibble: 15 x 2
##    manufacturer     n
##    <fct>        <int>
##  1 audi            18
##  2 chevrolet       19
##   3 dodge           37
##   4 ford            25
##   5 honda            9
##   6 hyundai         14
##   7 jeep             8
##   8 land rover       4
##   9 lincoln          3
##  10 mercury          4
##  11 nissan          13
##  12 pontiac          5
##  13 subaru          14
##  14 toyota          34
##  15 volkswagen      27
```

```
# Let's visualize frequencies (ggplot2 section is coming later!)
df %>%
  count(manufacturer) %>%
  ggplot(aes(x = manufacturer,
             y = n)) +
  geom_col()
```



```
# fct_unique() - Extract unique levels
# Car manufacturer unique levels
```

```r
df %>% .$manufacturer %>% fct_unique()
```

```
##  [1] audi       chevrolet  dodge      ford       honda      hyundai   
##  [7] jeep       land rover lincoln    mercury    nissan     pontiac   
## [13] subaru     toyota     volkswagen
## 15 Levels: audi chevrolet dodge ford honda hyundai jeep land rover ... volkswagen
```

```r
df %>% .$manufacturer %>% fct_unique() %>% as.character()
```

```
##  [1] "audi"       "chevrolet"  "dodge"      "ford"       "honda"     
##  [6] "hyundai"    "jeep"       "land rover" "lincoln"    "mercury"   
## [11] "nissan"     "pontiac"    "subaru"     "toyota"     "volkswagen"
```

Factors combine and order levels

```r
# fct_c() - Combine factors
# First lets split cars into 2 data frames
manufacturers <- df %>% .$manufacturer %>% fct_unique() %>% as.character() # unique manufacturers

df1 <- df %>%  # first subset
  filter(manufacturer %in% manufacturers[1:8])

df2 <- df %>%  # second subset
  filter(manufacturer %in% manufacturers[9:15])

# Extract only factor vectors
f1 <- df1 %>% pull(manufacturer)
f2 <- df2 %>% pull(manufacturer)

# Combine factors
c(f1, f2)    # with classical vector bind we lose factor level labels!
```

```
##   [1] audi       audi       audi       audi       audi       audi      
##   [7] audi       audi       audi       audi       audi       audi      
##  [13] audi       audi       audi       audi       audi       audi      
##  [19] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet 
##  [25] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet 
##  [31] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet 
##  [37] chevrolet  dodge      dodge      dodge      dodge      dodge     
##  [43] dodge      dodge      dodge      dodge      dodge      dodge     
##  [49] dodge      dodge      dodge      dodge      dodge      dodge     
##  [55] dodge      dodge      dodge      dodge      dodge      dodge     
##  [61] dodge      dodge      dodge      dodge      dodge      dodge     
##  [67] dodge      dodge      dodge      dodge      dodge      dodge     
##  [73] dodge      dodge      ford       ford       ford       ford      
##  [79] ford       ford       ford       ford       ford       ford      
##  [85] ford       ford       ford       ford       ford       ford      
##  [91] ford       ford       ford       ford       ford       ford      
##  [97] ford       ford       ford       honda      honda      honda     
## [103] honda      honda      honda      honda      honda      honda     
## [109] hyundai    hyundai    hyundai    hyundai    hyundai    hyundai   
## [115] hyundai    hyundai    hyundai    hyundai    hyundai    hyundai   
## [121] hyundai    hyundai    jeep       jeep       jeep       jeep      
## [127] jeep       jeep       jeep       jeep       land rover land rover
## [133] land rover land rover lincoln    lincoln    lincoln    mercury
## [139] mercury    mercury    mercury    nissan     nissan     nissan
## [145] nissan     nissan     nissan     nissan     nissan     nissan
## [151] nissan     nissan     nissan     nissan     pontiac    pontiac
## [157] pontiac    pontiac    pontiac    subaru     subaru     subaru
## [163] subaru     subaru     subaru     subaru     subaru     subaru
## [169] subaru     subaru     subaru     subaru     subaru     toyota
## [175] toyota     toyota     toyota     toyota     toyota     toyota
## [181] toyota     toyota     toyota     toyota     toyota     toyota
## [187] toyota     toyota     toyota     toyota     toyota     toyota
## [193] toyota     toyota     toyota     toyota     toyota     toyota
## [199] toyota     toyota     toyota     toyota     toyota     toyota
## [205] toyota     toyota     toyota     volkswagen volkswagen volkswagen
## [211] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## [217] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## [223] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## [229] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## 15 Levels: audi chevrolet dodge ford honda hyundai jeep land rover ... volkswagen
```

```
fct_c(f1,f2) # This way levels are preserved!
```

```
##   [1] audi       audi       audi       audi       audi       audi
##   [7] audi       audi       audi       audi       audi       audi
##  [13] audi       audi       audi       audi       audi       audi
##  [19] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet
##  [25] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet
##  [31] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet
##  [37] chevrolet  dodge      dodge      dodge      dodge      dodge
##  [43] dodge      dodge      dodge      dodge      dodge      dodge
##  [49] dodge      dodge      dodge      dodge      dodge      dodge
##  [55] dodge      dodge      dodge      dodge      dodge      dodge
##  [61] dodge      dodge      dodge      dodge      dodge      dodge
##  [67] dodge      dodge      dodge      dodge      dodge      dodge
##  [73] dodge      dodge      ford       ford       ford       ford
##  [79] ford       ford       ford       ford       ford       ford
##  [85] ford       ford       ford       ford       ford       ford
##  [91] ford       ford       ford       ford       ford       ford
##  [97] ford       ford       ford       honda      honda      honda
## [103] honda      honda      honda      honda      honda      honda
## [109] hyundai    hyundai    hyundai    hyundai    hyundai    hyundai
## [115] hyundai    hyundai    hyundai    hyundai    hyundai    hyundai
## [121] hyundai    hyundai    jeep       jeep       jeep       jeep
## [127] jeep       jeep       jeep       jeep       land rover land rover
## [133] land rover land rover lincoln    lincoln    lincoln    mercury
## [139] mercury    mercury    mercury    nissan     nissan     nissan
## [145] nissan     nissan     nissan     nissan     nissan     nissan
## [151] nissan     nissan     nissan     nissan     pontiac    pontiac
## [157] pontiac    pontiac    pontiac    subaru     subaru     subaru
## [163] subaru     subaru     subaru     subaru     subaru     subaru
## [169] subaru     subaru     subaru     subaru     subaru     toyota
## [175] toyota     toyota     toyota     toyota     toyota     toyota
## [181] toyota     toyota     toyota     toyota     toyota     toyota
## [187] toyota     toyota     toyota     toyota     toyota     toyota
## [193] toyota     toyota     toyota     toyota     toyota     toyota
## [199] toyota     toyota     toyota     toyota     toyota     toyota
## [205] toyota     toyota     toyota     volkswagen volkswagen volkswagen
## [211] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## [217] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## [223] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## [229] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
## 15 Levels: audi chevrolet dodge ford honda hyundai jeep land rover ... volkswagen
```

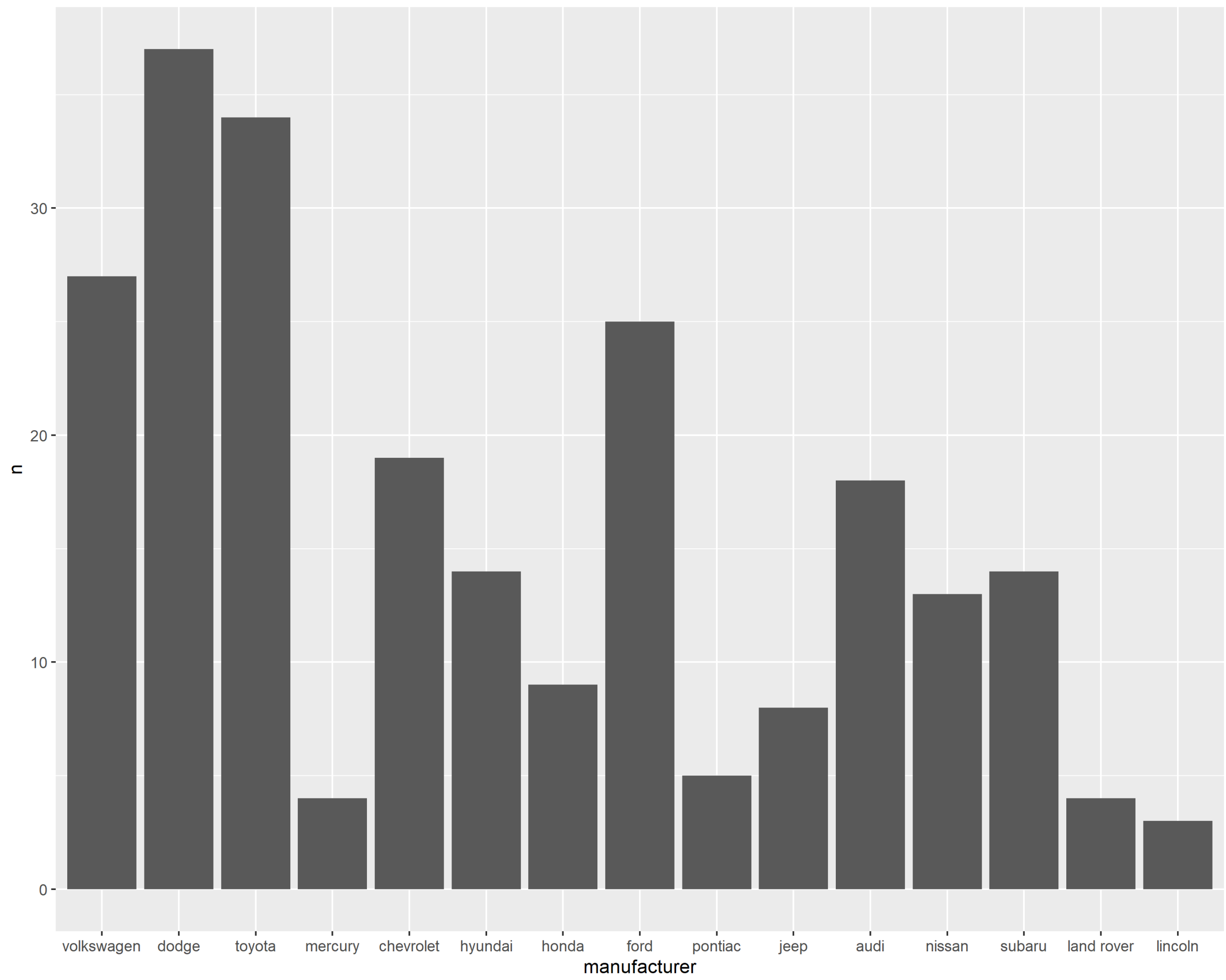
```r
# fct_relevel() - Manually reorder levels
# Lets randomly shuffle levels - manufacturers
set.seed(123)

manufacturers.rnd <- sample(manufacturers, size = length(manufacturers), replace = F)

# Now count frequencies &  create another bar plot with manually reordered levels
df %>%
  mutate(manufacturer = fct_relevel(manufacturer, manufacturers.rnd)) %>%
  count(manufacturer)
```

```
## # A tibble: 15 x 2
##    manufacturer     n
##    <fct>        <int>
##  1 volkswagen      27
##  2 dodge           37
##  3 toyota          34
##  4 mercury          4
##  5 chevrolet       19
##  6 hyundai         14
##  7 honda            9
##  8 ford            25
##  9 pontiac          5
## 10 jeep             8
## 11 audi            18
## 12 nissan          13
## 13 subaru          14
## 14 land rover       4
## 15 lincoln          3
```

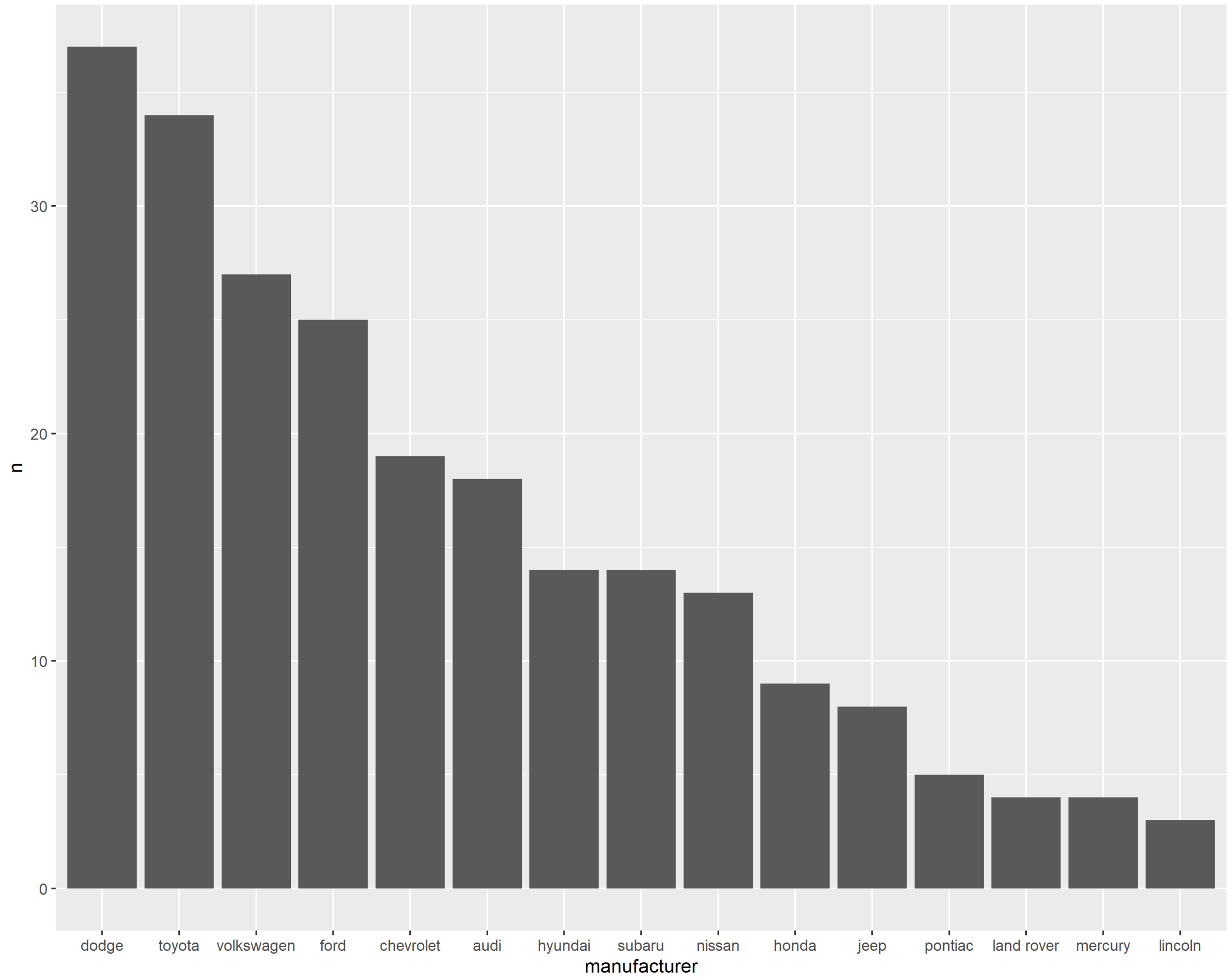
```r
df %>%
  mutate(manufacturer = fct_relevel(manufacturer, manufacturers.rnd)) %>%
  count(manufacturer) %>%
  ggplot(aes(x = manufacturer,
             y = n)) +
  geom_col()
```



```
# fct_infreq() - Levels by frequency
# Order manufacturers based on car count
df %>%
  mutate(manufacturer = fct_infreq(manufacturer)) %>%
  count(manufacturer)
```

```
## # A tibble: 15 x 2
##    manufacturer     n
##    <fct>        <int>
##  1 dodge           37
##  2 toyota          34
##  3 volkswagen      27
##  4 ford            25
##  5 chevrolet       19
##  6 audi            18
##  7 hyundai         14
##  8 subaru          14
##  9 nissan          13
## 10 honda            9
## 11 jeep             8
## 12 pontiac          5
## 13 land rover       4
## 14 mercury           4
## 15 lincoln           3
```

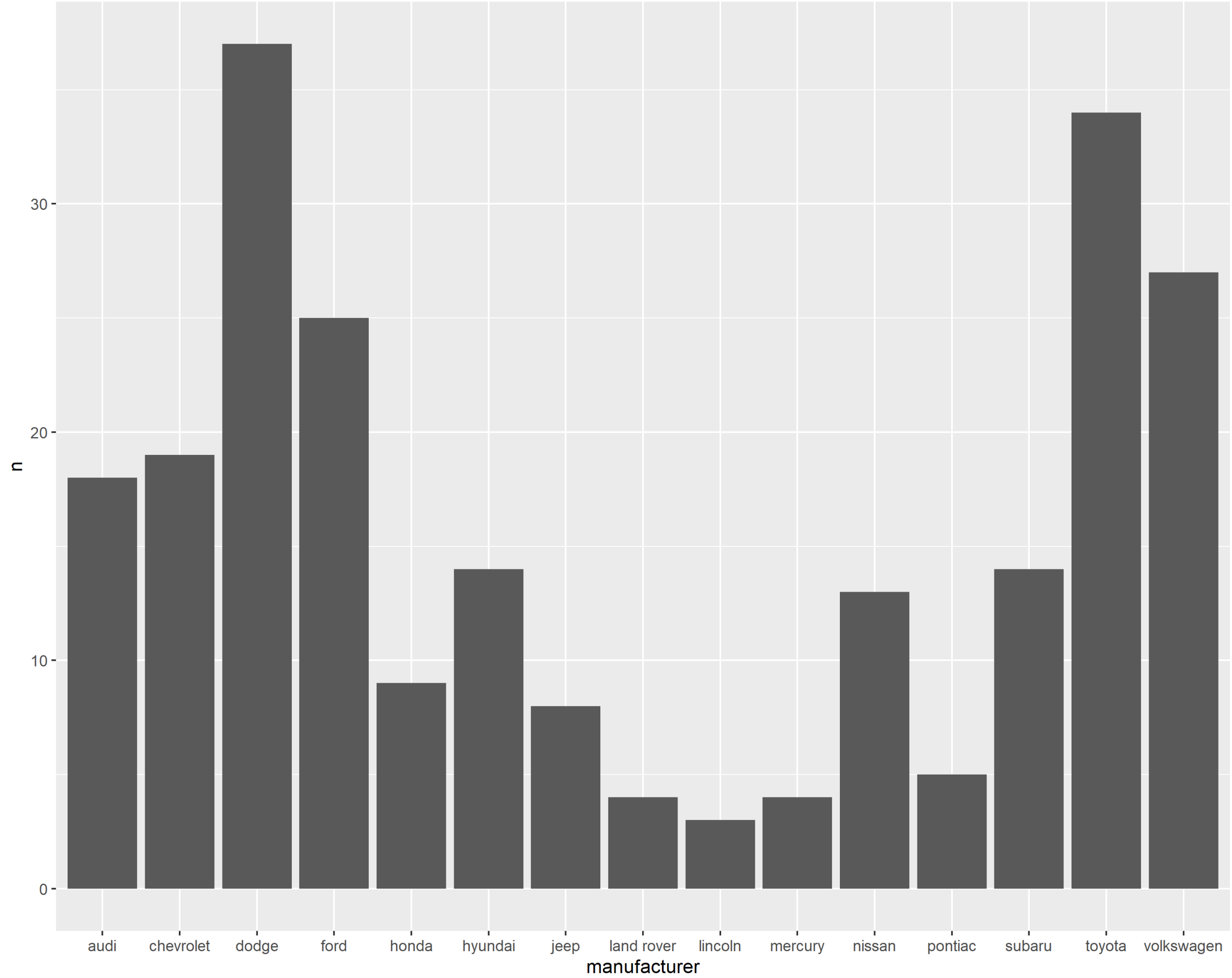
```
df %>%
  mutate(manufacturer = fct_infreq(manufacturer)) %>%
  count(manufacturer) %>%
  ggplot(aes(x = manufacturer,
             y = n)) +
  geom_col()
```



```
# fct_inorder() - Levels by order of appearance
# Order manufacturers based how they appear in the data
df %>%
  mutate(manufacturer = fct_inorder(manufacturer)) %>%
  count(manufacturer)
```

```
## # A tibble: 15 x 2
##    manufacturer     n
##    <fct>        <int>
##  1 audi            18
##  2 chevrolet       19
##  3 dodge           37
##  4 ford            25
##   5 honda           9
##   6 hyundai        14
##   7 jeep            8
##   8 land rover      4
##   9 lincoln         3
##  10 mercury         4
##  11 nissan         13
##  12 pontiac         5
##  13 subaru         14
##  14 toyota         34
##  15 volkswagen     27
```

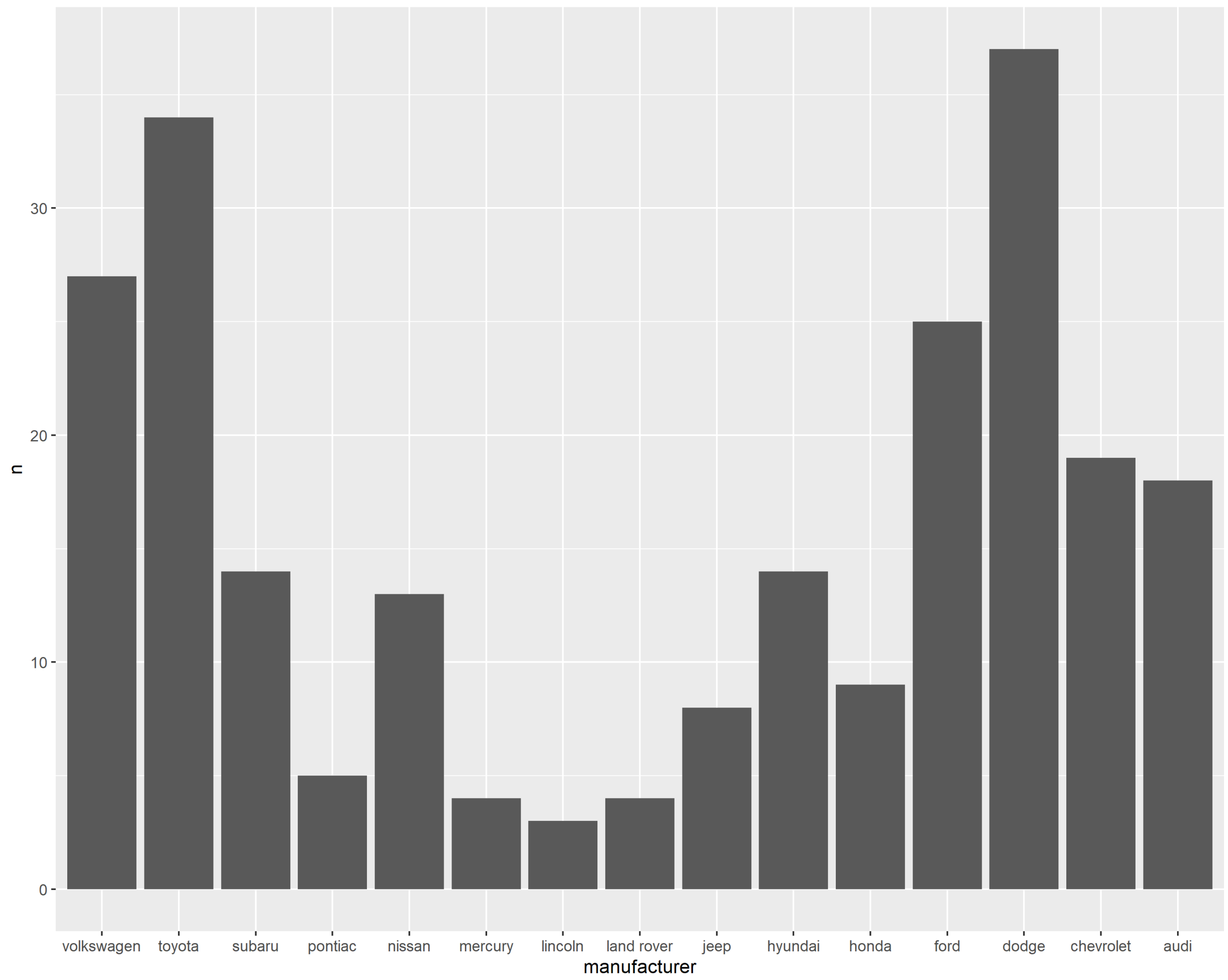
```
df %>%
  mutate(manufacturer = fct_inorder(manufacturer)) %>%
  count(manufacturer) %>%
  ggplot(aes(x = manufacturer,
             y = n)) +
  geom_col()
```



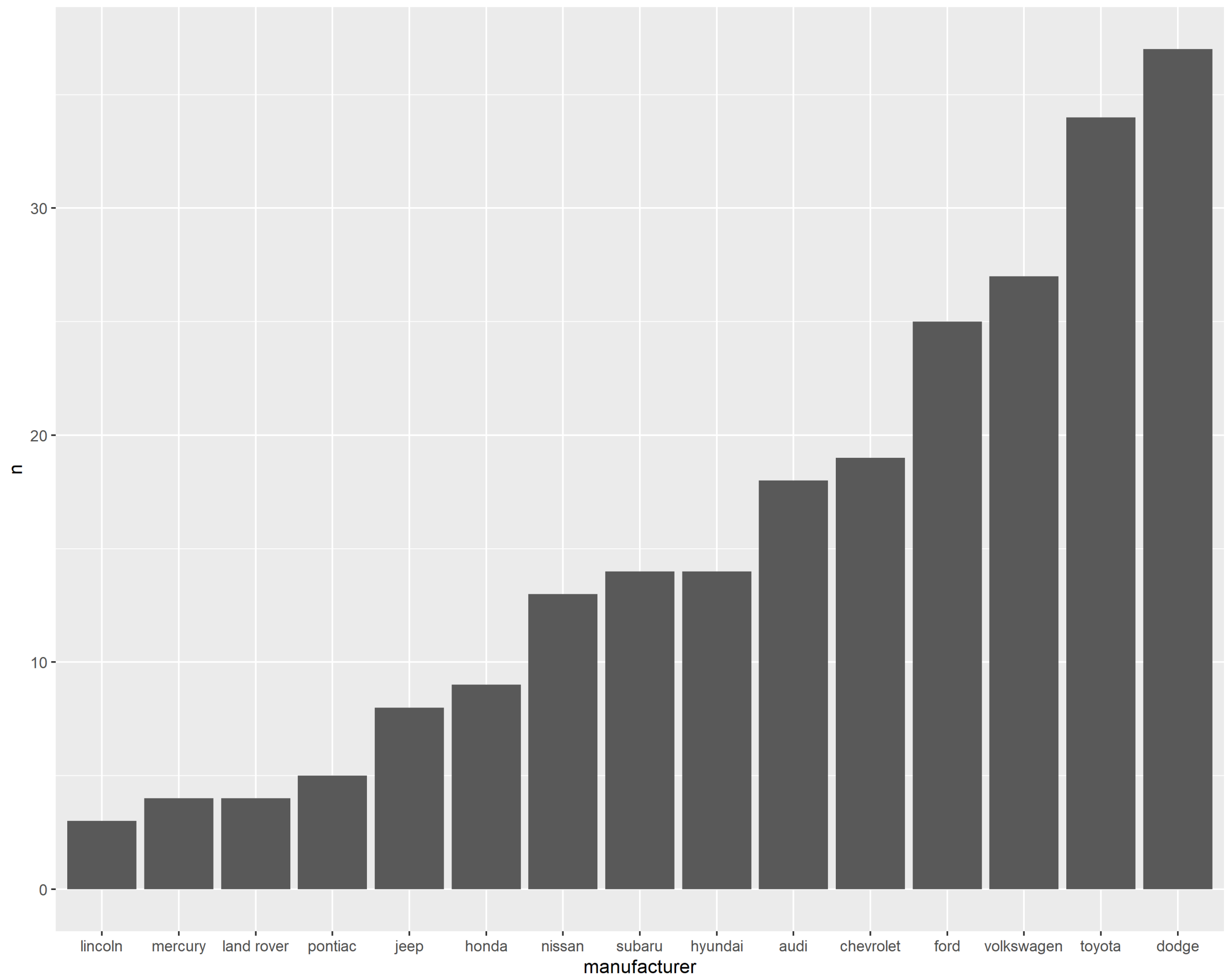
```
# fct_rev() - Reverse level of order
# Order manufacturers based on reverse appereance in the data
df %>%
  mutate(manufacturer = fct_rev(manufacturer)) %>%
  count(manufacturer)
```

```
## # A tibble: 15 x 2
##    manufacturer     n
##    <fct>        <int>
##  1 volkswagen      27
##  2 toyota          34
##  3 subaru          14
##  4 pontiac          5
##  5 nissan          13
##  6 mercury          4
##  7 lincoln          3
##  8 land rover       4
##  9 jeep             8
## 10 hyundai         14
## 11 honda            9
## 12 ford            25
## 13 dodge           37
## 14 chevrolet       19
## 15 audi            18
```

```
df %>%
  mutate(manufacturer = fct_rev(manufacturer)) %>%
  count(manufacturer) %>%
  ggplot(aes(x = manufacturer,
             y = n)) +
  geom_col()
```



```r
# To get reverse frequency count: fct_infreq + fct_rev
df %>%
  mutate(manufacturer = fct_infreq(manufacturer),
         manufacturer = fct_rev(manufacturer)) %>%
  count(manufacturer) %>%
  ggplot(aes(x = manufacturer,
             y = n)) +
  geom_col()
```



Factors change levels value and add / drop levels

```
# fct_recode() - Manually change levels
# First lets pull levels and add county of origin
df %>% pull(manufacturer) %>% fct_count()
```

```
## # A tibble: 15 x 2
##    f              n
##    <fct>      <int>
##  1 audi          18
##  2 chevrolet     19
##  3 dodge         37
##  4 ford          25
##  5 honda          9
##  6 hyundai       14
##  7 jeep           8
##  8 land rover     4
##  9 lincoln        3
## 10 mercury        4
## 11 nissan        13
## 12 pontiac        5
## 13 subaru        14
## 14 toyota        34
## 15 volkswagen    27
```

```r
levels.country <- tribble( # table of: company & country of origin
  ~company,    ~country,
   "audi",          "Germany",
   "chevrolet",     "USA",
   "dodge",         "USA",
   "ford",          "USA",
   "honda",         "Japan",
   "hyundai",       "South Korea",
   "jeep",          "USA",
   "land rover",    "England",
   "lincoln",       "USA",
   "mercury",       "USA",
   "nissan",       "Japan",
   "pontiac",       "USA",
   "subaru",        "Japan",
   "toyota",        "Japan",
   "volkswagen",    "Germany")

# Prepare pairs for re-coding factor levels
levels.country %>%
  mutate(recode = str_c(country, " = ", "'", company, "'", sep = "")) %>%
  pull(recode) %>%
  str_c(., collapse = ", ")
```

```
## [1] "Germany = 'audi', USA = 'chevrolet', USA = 'dodge', USA = 'ford', Japan = 'honda', South Korea =
```

```r
# Now re-code factor levels: companies -> countries
df.recode <- df %>%
  mutate(manufacturer = fct_recode(manufacturer,
                                   Germany = 'audi',
                                   USA = 'chevrolet',
                                   USA = 'dodge',
                                   USA = 'ford',
                                   Japan = 'honda',
                                   `South Korea` = 'hyundai',
                                   USA = 'jeep',
                                   England = 'land rover',
                                   USA = 'lincoln',
                                   USA = 'mercury',
                                   Japan = 'nissan',
                                   USA = 'pontiac',
                                   Japan = 'subaru',
                                   Japan = 'toyota',
                                   Germany = 'volkswagen'))

# Check recoded factor levels
df.recode %>%
  count(manufacturer)
```

```
## # A tibble: 5 x 2
##   manufacturer     n
##   <fct>        <int>
## 1 Germany         45
## 2 USA            101
## 3 Japan           70
## 4 South Korea     14
## 5 England          4
```

```
# fct_collapse() - Collapse levels into manually defined groups
# Let's keep only USA manufacturers, others are collapsed
non.us.manufacturers <- levels.country %>% filter(country != "USA") %>% pull(company) # vector of non-U

df.collapse <- df %>%
  mutate(manufacturer = fct_collapse(manufacturer, `non US` = non.us.manufacturers))

# Check collapsed factor levels
df.collapse %>%
  count(manufacturer)
```

```
## # A tibble: 8 x 2
##   manufacturer     n
##   <fct>        <int>
## 1 non US         133
## 2 chevrolet       19
## 3 dodge           37
## 4 ford            25
## 5 jeep             8
## 6 lincoln          3
## 7 mercury          4
## 8 pontiac          5
```

```
# fct_other() - Replace levels with other
# All non-US companies -> Other
df.other <- df %>%
  mutate(manufacturer = fct_other(manufacturer, drop = non.us.manufacturers))

# Check other factor levels
df.other %>%
  count(manufacturer)
```

```
## # A tibble: 8 x 2
##   manufacturer     n
##   <fct>        <int>
## 1 chevrolet       19
## 2 dodge           37
## 3 ford            25
## 4 jeep             8
## 5 lincoln          3
## 6 mercury          4
## 7 pontiac          5
## 8 Other          133
```

```
# fct_drop() - Drop factor levels
# Drop other level
# - first filter out rows with "Other"
df.drop <- df.other %>%
  filter(manufacturer != "Other")
```

```
# Check levels - "Other" still present!
df.drop %>% pull(manufacturer) %>%  fct_unique()
```

```
## [1] chevrolet dodge     ford      jeep      lincoln   mercury   pontiac  
## [8] Other    
## Levels: chevrolet dodge ford jeep lincoln mercury pontiac Other
```

```
# Now drop "Other" level with no more data
df.drop <- df.drop %>%
  mutate(manufacturer = fct_drop(manufacturer))

# Check levels - "Other" removed!
df.drop %>% pull(manufacturer) %>%  fct_unique()
```

```
## [1] chevrolet dodge     ford      jeep      lincoln   mercury   pontiac  
## Levels: chevrolet dodge ford jeep lincoln mercury pontiac
```

```
# fct_expand() - Add additional levels to factor
# Lets add some additional manufacturers
df.expand <- df %>%
  mutate(manufacturer = fct_expand(manufacturer, c("Ferrari", "Lamborghini")))

# Check levels - "New levels added
df.expand %>% pull(manufacturer) %>% fct_unique()
```

```
##  [1] audi        chevrolet   dodge       ford        honda       hyundai    
##  [7] jeep        land rover  lincoln     mercury     nissan      pontiac    
## [13] subaru      toyota      volkswagen  Ferrari     Lamborghini
## 17 Levels: audi chevrolet dodge ford honda hyundai jeep land rover ... Lamborghini
```

```
df.expand %>% pull(manufacturer) %>% levels()
```

```
##  [1] "audi"        "chevrolet"   "dodge"       "ford"        "honda"      
##  [6] "hyundai"     "jeep"        "land rover"  "lincoln"     "mercury"    
## [11] "nissan"      "pontiac"     "subaru"      "toyota"      "volkswagen" 
## [16] "Ferrari"     "Lamborghini"
```

```
# But at the moment there aren't any cars from "Ferrari" or "Lamborghini"
# - just levels are prepared in advance
df.expand %>% filter(manufacturer %in% c("Ferrari", "Lamborghini"))
```

```
## # A tibble: 0 x 11
## # i 11 variables: manufacturer <fct>, model <fct>, displ <dbl>, year <int>,
## #   cyl <int>, trans <fct>, drv <chr>, cty <int>, hwy <int>, fl <chr>,
## #   class <fct>
```