Assignment: Week 5

Name: Ramani, Aarti

Date: 2023-01-14

```
library(readxl)
setwd("C:/Masters/GitHub/Winter2022/Ramani-DSC520")
housing_df <- read_excel(path = "C:/Masters/GitHub/Winter2022/Ramani-DSC520/data/week-6-housing.xlsx",
                         .name_repair = function(col){ gsub(" ", "_", col) })

#Using the dplyr package, use the 6 different operations to analyze/transform
#the data - GroupBy, Summarize, Mutate, Filter, Select, and Arrange
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# GROUP BY and GROUP KEYS
#GROUP_KEYS is depricated as of dplyr 1.0.0
price_df <-  housing_df %>% group_by(zip5, ctyname)
price_df
```

```
## # A tibble: 12,865 x 24
## # Groups:   zip5, ctyname [6]
##    Sale_Date           Sale_Price sale_r~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##    <dttm>                   <dbl>    <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
##  1 2006-01-03 00:00:00     698000        1       3 <NA>    R1      17021 ~ 98052
##  2 2006-01-03 00:00:00     649990        1       3 <NA>    R1      11927 ~ 98052
##  3 2006-01-03 00:00:00     572500        1       3 <NA>    R1      13315 ~ 98052
##  4 2006-01-03 00:00:00     420000        1       3 <NA>    R1      3303 1~ 98052
##  5 2006-01-03 00:00:00     369900        1       3 15      R1      16126 ~ 98052
##  6 2006-01-03 00:00:00     184667        1      15 18 51   R1      8101 2~ 98053
##  7 2006-01-04 00:00:00    1050000        1       3 <NA>    R1      21634 ~ 98053
##  8 2006-01-04 00:00:00     875000        1       3 <NA>    R1      21404 ~ 98053
##  9 2006-01-04 00:00:00     660000        1       3 <NA>    R1      7525 2~ 98053
## 10 2006-01-04 00:00:00     650000        1       3 <NA>    R1      17703 ~ 98052
## # ... with 12,855 more rows, 16 more variables: ctyname <chr>,
## #   postalctyn <chr>, lon <dbl>, lat <dbl>, building_grade <dbl>,
## #   square_feet_total_living <dbl>, bedrooms <dbl>, bath_full_count <dbl>,
## #   bath_half_count <dbl>, bath_3qtr_count <dbl>, year_built <dbl>,
```

```
## #   year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## #   prop_type <chr>, present_use <dbl>, and abbreviated variable names
## #   1: sale_reason, 2: sale_instrument, 3: sale_warning, 4: sitetype, ...
```

```
price_df <-  housing_df %>% group_by(zip5, ctyname) %>% tally()
price_df
```

```
## # A tibble: 6 x 3
## # Groups:   zip5 [4]
##    zip5 ctyname       n
##   <dbl> <chr>     <int>
## 1 98052 REDMOND    6721
## 2 98052 <NA>        731
## 3 98053 <NA>       5339
## 4 98059 <NA>          1
## 5 98074 SAMMAMISH    66
## 6 98074 <NA>          7
```

```
#price_df <-  housing_df %>% group_keys(zip5, ctyname)
#price_df <-  housing_df %>% group_keys(zip5, sale_reason)
price_df <-  housing_df %>% group_by(zip5, sale_reason) %>% tally()
price_df
```

```
## # A tibble: 31 x 3
## # Groups:   zip5 [4]
##     zip5 sale_reason     n
##    <dbl>       <dbl> <int>
##  1 98052           0     2
##  2 98052           1  7051
##  3 98052           4    81
##  4 98052           6     1
##  5 98052           7     1
##  6 98052           8    94
##  7 98052          10    10
##  8 98052          11     1
##  9 98052          12    37
## 10 98052          13     2
## # ... with 21 more rows
```

```
# SUMMARIZE
price_df <-  housing_df %>% group_by(zip5, ctyname) %>% summarize(AvgPrice = mean(Sale_Price))
```

```
## `summarise()` has grouped output by 'zip5'. You can override using the
## `.groups` argument.
```

```
price_df
```

```
## # A tibble: 6 x 3
## # Groups:   zip5 [4]
##    zip5 ctyname   AvgPrice
##   <dbl> <chr>        <dbl>
```

```
## 1 98052 REDMOND    644803.
## 2 98052 <NA>       691413.
## 3 98053 <NA>       672624.
## 4 98059 <NA>       645000
## 5 98074 SAMMAMISH  972480.
## 6 98074 <NA>       754143.
```

```
# SELECT
price_df <-  housing_df %>%
  select(one_of ('Sale_Price', 'sale_reason', 'zip5','ctyname', 'square_feet_total_living',
                 'bedrooms', 'sq_ft_lot' ))
price_df
```

```
## # A tibble: 12,865 x 7
##    Sale_Price sale_reason  zip5 ctyname square_feet_total_living bedro~1 sq_ft~2
##         <dbl>       <dbl> <dbl> <chr>                      <dbl>   <dbl>   <dbl>
##  1     698000           1 98052 REDMOND                     2810       4    6635
##  2     649990           1 98052 REDMOND                     2880       4    5570
##  3     572500           1 98052 <NA>                        2770       4    8444
##  4     420000           1 98052 REDMOND                     1620       3    9600
##  5     369900           1 98052 REDMOND                     1440       3    7526
##  6     184667           1 98053 <NA>                        4160       4    7280
##  7    1050000           1 98053 <NA>                        3960       5   97574
##  8     875000           1 98053 <NA>                        3720       4   30649
##  9     660000           1 98053 <NA>                        4160       4   42688
## 10     650000           1 98052 REDMOND                     2760       4   94889
## # ... with 12,855 more rows, and abbreviated variable names 1: bedrooms,
## #   2: sq_ft_lot
```

```
# MUTATE
price_df <-  housing_df %>%
  mutate(Sq_yards_total_living = square_feet_total_living/9)
names(price_df)
```

```
##  [1] "Sale_Date"               "Sale_Price"
##  [3] "sale_reason"             "sale_instrument"
##  [5] "sale_warning"            "sitetype"
##  [7] "addr_full"               "zip5"
##  [9] "ctyname"                 "postalctyn"
## [11] "lon"                     "lat"
## [13] "building_grade"          "square_feet_total_living"
## [15] "bedrooms"                "bath_full_count"
## [17] "bath_half_count"         "bath_3qtr_count"
## [19] "year_built"              "year_renovated"
## [21] "current_zoning"          "sq_ft_lot"
## [23] "prop_type"               "present_use"
## [25] "Sq_yards_total_living"
```

```
price_df
```

```
## # A tibble: 12,865 x 25
##    Sale_Date           Sale_Price sale_r~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
```

```
##    <dttm>                  <dbl>    <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
##  1 2006-01-03 00:00:00     698000        1       3 <NA>    R1      17021 ~ 98052
##  2 2006-01-03 00:00:00     649990        1       3 <NA>    R1      11927 ~ 98052
##  3 2006-01-03 00:00:00     572500        1       3 <NA>    R1      13315 ~ 98052
##  4 2006-01-03 00:00:00     420000        1       3 <NA>    R1      3303 1~ 98052
##  5 2006-01-03 00:00:00     369900        1       3 15      R1      16126 ~ 98052
##  6 2006-01-03 00:00:00     184667        1      15 18 51   R1      8101 2~ 98053
##  7 2006-01-04 00:00:00    1050000        1       3 <NA>    R1      21634 ~ 98053
##  8 2006-01-04 00:00:00     875000        1       3 <NA>    R1      21404 ~ 98053
##  9 2006-01-04 00:00:00     660000        1       3 <NA>    R1      7525 2~ 98053
## 10 2006-01-04 00:00:00     650000        1       3 <NA>    R1      17703 ~ 98052
## # ... with 12,855 more rows, 17 more variables: ctyname <chr>,
## #   postalctyn <chr>, lon <dbl>, lat <dbl>, building_grade <dbl>,
## #   square_feet_total_living <dbl>, bedrooms <dbl>, bath_full_count <dbl>,
## #   bath_half_count <dbl>, bath_3qtr_count <dbl>, year_built <dbl>,
## #   year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## #   prop_type <chr>, present_use <dbl>, Sq_yards_total_living <dbl>, and
## #   abbreviated variable names 1: sale_reason, 2: sale_instrument, ...
```

```
# MUTATE AND SELECT
price_df <-  housing_df %>%
  select(one_of ('Sale_Price', 'sale_reason', 'zip5','ctyname', 'square_feet_total_living',
                 'bedrooms', 'sq_ft_lot' )) %>%
  mutate(Sq_yards_total_living = square_feet_total_living/9)
price_df
```

```
## # A tibble: 12,865 x 8
##    Sale_Price sale_reason  zip5 ctyname square_feet_to~1 bedro~2 sq_ft~3 Sq_ya~4
##         <dbl>       <dbl> <dbl> <chr>              <dbl>   <dbl>   <dbl>   <dbl>
##  1     698000           1 98052 REDMOND             2810       4    6635    312.
##  2     649990           1 98052 REDMOND             2880       4    5570    320
##  3     572500           1 98052 <NA>                2770       4    8444    308.
##  4     420000           1 98052 REDMOND             1620       3    9600    180
##  5     369900           1 98052 REDMOND             1440       3    7526    160
##  6     184667           1 98053 <NA>                4160       4    7280    462.
##  7    1050000           1 98053 <NA>                3960       5   97574    440
##  8     875000           1 98053 <NA>                3720       4   30649    413.
##  9     660000           1 98053 <NA>                4160       4   42688    462.
## 10     650000           1 98052 REDMOND             2760       4   94889    307.
## # ... with 12,855 more rows, and abbreviated variable names
## #   1: square_feet_total_living, 2: bedrooms, 3: sq_ft_lot,
## #   4: Sq_yards_total_living
```

```
# FILTER
price_df <-  housing_df %>%
  select(one_of ('Sale_Price', 'sale_reason', 'zip5','ctyname', 'square_feet_total_living',
                 'bedrooms', 'sq_ft_lot' )) %>%
  filter(is.na(ctyname))
price_df
```

```
## # A tibble: 6,078 x 7
##    Sale_Price sale_reason  zip5 ctyname square_feet_total_living bedro~1 sq_ft~2
##         <dbl>       <dbl> <dbl> <chr>                      <dbl>   <dbl>   <dbl>
```

```
##  1     572500           1 98052 <NA>                          2770       4    8444
##  2     184667           1 98053 <NA>                          4160       4    7280
##  3    1050000           1 98053 <NA>                          3960       5   97574
##  4     875000           1 98053 <NA>                          3720       4   30649
##  5     660000           1 98053 <NA>                          4160       4   42688
##  6     165000           1 98053 <NA>                          1850       3  278891
##  7     803000           1 98053 <NA>                          3180       3   95013
##  8     765000           1 98053 <NA>                          4000       4    7611
##  9     372500           1 98053 <NA>                          1620       3   47480
## 10     513262           1 98053 <NA>                          1930       2    4958
## # ... with 6,068 more rows, and abbreviated variable names 1: bedrooms,
## #   2: sq_ft_lot
```

```
# ARRANGE
price_df <-  housing_df %>%
  select(one_of ('Sale_Price', 'sale_reason', 'zip5','ctyname', 'square_feet_total_living',
                 'bedrooms', 'sq_ft_lot' )) %>%
  filter(is.na(ctyname)) %>%
  mutate(Sq_yards_total_living = square_feet_total_living/9) %>%
  arrange(desc(Sq_yards_total_living))
price_df
```

```
## # A tibble: 6,078 x 8
##    Sale_Price sale_reason  zip5 ctyname square_feet_to~1 bedro~2 sq_ft~3 Sq_ya~4
##         <dbl>       <dbl> <dbl> <chr>              <dbl>   <dbl>   <dbl>   <dbl>
##  1    2300000           4 98053 <NA>               13540       7  307752   1504.
##  2    1300000           1 98053 <NA>               13540       7  307752   1504.
##  3    2280000           1 98053 <NA>               13540       7  307752   1504.
##  4    3995000           1 98052 <NA>               11810       7  139392   1312.
##  5    2988000           1 98053 <NA>               10630       5  207781   1181.
##  6    1775000           7 98053 <NA>                9720       4   81335   1080
##  7     349999           1 98052 <NA>                9360       4   45738   1040
##  8    1050000           1 98053 <NA>                9070       6  186525   1008.
##  9      14000           1 98053 <NA>                8750       5 1631322    972.
## 10     350000           1 98053 <NA>                8490       7  118483    943.
## # ... with 6,068 more rows, and abbreviated variable names
## #   1: square_feet_total_living, 2: bedrooms, 3: sq_ft_lot,
## #   4: Sq_yards_total_living
```

```
#Using the purrr package - perform 2 functions on your dataset.
library(purrr)
#You could use zip_n, keep, discard, compact, etc.
#MAP
#housing_df %>% map_dbl(mean)
housing_df %>% map(mean)
```

```
## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA


## $Sale_Date
## [1] "2011-07-28 15:07:32 UTC"
##
## $Sale_Price
## [1] 660737.7
##
## $sale_reason
## [1] 1.550019
##
## $sale_instrument
## [1] 3.67773
##
## $sale_warning
## [1] NA
##
## $sitetype
## [1] NA
##
## $addr_full
## [1] NA
##
## $zip5
## [1] 98052.54
##
## $ctyname
## [1] NA
##
## $postalctyn
## [1] NA
##
## $lon
## [1] -122.0792
##
## $lat
## [1] 47.68358
##
## $building_grade
## [1] 8.24042
##
## $square_feet_total_living
```

```
## [1] 2539.506
##
## $bedrooms
## [1] 3.478663
##
## $bath_full_count
## [1] 1.798445
##
## $bath_half_count
## [1] 0.6133696
##
## $bath_3qtr_count
## [1] 0.4939759
##
## $year_built
## [1] 1993.003
##
## $year_renovated
## [1] 26.24431
##
## $current_zoning
## [1] NA
##
## $sq_ft_lot
## [1] 22228.57
##
## $prop_type
## [1] NA
##
## $present_use
## [1] 6.597746
```

```
#KEEP

#sale_price_1m <- housing_df$Sale_Price %>% map(mean) %>%
#  keep(~mean(.x) >= 1000000)
sale_price_4380542 <- housing_df$Sale_Price %>%
  keep(housing_df$Sale_Price == 4380542)
length(sale_price_4380542)
```

```
## [1] 14
```

```
#DISCARD
#sale_price_discard_1m <- housing_df$Sale_Price %>% map(mean) %>%
#  discard(~mean(.x) >= 1000000)
#length(sale_price_discard_1m)
sale_price_not_4380542 <- housing_df$Sale_Price %>%
  discard(housing_df$Sale_Price == 4380542)
length(sale_price_not_4380542)
```

```
## [1] 12851
```

```r
# Use the cbind and rbind function on your dataset
#CBIND
price_loc_df = data.frame("SalePrice"=housing_df$Sale_Price, "Zip"=housing_df$zip5,"CityName"=housing_d
length(price_loc_df)
```

```
## [1] 3
```

```r
house_df <- data.frame("Bedrooms"=housing_df$bedrooms, "Sq.Ft"=housing_df$square_feet_total_living)
length(house_df)
```

```
## [1] 2
```

```r
cbind_housing_df <- cbind(price_loc_df,house_df )
nrow(cbind_housing_df)
```

```
## [1] 12865
```

```r
length(cbind_housing_df)
```

```
## [1] 5
```

```r
head(cbind_housing_df)
```

```
##   SalePrice   Zip CityName Bedrooms Sq.Ft
## 1    698000 98052  REDMOND        4  2810
## 2    649990 98052  REDMOND        4  2880
## 3    572500 98052     <NA>        4  2770
## 4    420000 98052  REDMOND        3  1620
## 5    369900 98052  REDMOND        3  1440
## 6    184667 98053     <NA>        4  4160
```

```r
#RBIND
price_4380542 <- housing_df[housing_df$Sale_Price == 4380542,]
price_not_4380542 <- housing_df[housing_df$Sale_Price != 4380542,]
nrow(price_4380542)
```

```
## [1] 14
```

```r
nrow(price_not_4380542)
```

```
## [1] 12851
```

```r
rbind_housing_df <- rbind(price_4380542,price_not_4380542)
nrow(rbind_housing_df)
```

```
## [1] 12865
```

```
rbind_housing_df[order(housing_df$zip5, housing_df$ctyname),]
```

```
## # A tibble: 12,865 x 24
##    Sale_Date           Sale_Price sale_r~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##    <dttm>                   <dbl>    <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
##  1 2011-11-17 00:00:00    4380542        1      22 11 45   R1      17137 ~ 98052
##  2 2011-11-17 00:00:00    4380542        1      22 11 45   R1      11818 ~ 98052
##  3 2011-11-17 00:00:00    4380542        1      22 11 45   R1      16943 ~ 98052
##  4 2011-11-17 00:00:00    4380542        1      22 11 45   R1      16944 ~ 98052
##  5 2011-11-17 00:00:00    4380542        1      22 11 45   R1      11719 ~ 98052
##  6 2011-11-17 00:00:00    4380542        1      22 11 45   R1      16955 ~ 98052
##  7 2011-11-17 00:00:00    4380542        1      22 11 45   R1      11703 ~ 98052
##  8 2011-11-17 00:00:00    4380542        1      22 11 45   R1      16906 ~ 98052
##  9 2006-01-03 00:00:00     649990        1       3 <NA>    R1      11927 ~ 98052
## 10 2006-01-03 00:00:00     420000        1       3 <NA>    R1      3303 1~ 98052
## # ... with 12,855 more rows, 16 more variables: ctyname <chr>,
## #   postalctyn <chr>, lon <dbl>, lat <dbl>, building_grade <dbl>,
## #   square_feet_total_living <dbl>, bedrooms <dbl>, bath_full_count <dbl>,
## #   bath_half_count <dbl>, bath_3qtr_count <dbl>, year_built <dbl>,
## #   year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## #   prop_type <chr>, present_use <dbl>, and abbreviated variable names
## #   1: sale_reason, 2: sale_instrument, 3: sale_warning, 4: sitetype, ...
```

```
#Split a string, then concatenate the results back together
#SPLIT
split_str <- strsplit(housing_df[housing_df$Sale_Price>3000000,]$sale_warning, " ")
split_str
```

```
## [[1]]
## [1] NA
##
## [[2]]
## [1] "45"
##
## [[3]]
## [1] "45"
##
## [[4]]
## [1] "45"
##
## [[5]]
## [1] "45"
##
## [[6]]
## [1] "45"
##
## [[7]]
## [1] "45"
##
## [[8]]
## [1] "45"
##
```

```
## [[9]]
## [1] "45"
##
## [[10]]
## [1] "45"
##
## [[11]]
## [1] "45"
##
## [[12]]
## [1] "45"
##
## [[13]]
## [1] "45"
##
## [[14]]
## [1] "45"
##
## [[15]]
## [1] "45"
##
## [[16]]
## [1] "45"
##
## [[17]]
## [1] "45"
##
## [[18]]
## [1] "45"
##
## [[19]]
## [1] "45"
##
## [[20]]
## [1] "45"
##
## [[21]]
## [1] "45"
##
## [[22]]
## [1] "45"
##
## [[23]]
## [1] "45"
##
## [[24]]
## [1] "45"
##
## [[25]]
## [1] "45"
##
## [[26]]
## [1] "45"
##
```

```
## [[27]]
## [1] "45"
##
## [[28]]
## [1] "45"
##
## [[29]]
## [1] "45"
##
## [[30]]
## [1] "45"
##
## [[31]]
## [1] "45"
##
## [[32]]
## [1] "45"
##
## [[33]]
## [1] "45"
##
## [[34]]
## [1] "45"
##
## [[35]]
## [1] "45"
##
## [[36]]
## [1] "45"
##
## [[37]]
## [1] "45"
##
## [[38]]
## [1] "45"
##
## [[39]]
## [1] "45"
##
## [[40]]
## [1] "45"
##
## [[41]]
## [1] "45"
##
## [[42]]
## [1] "45"
##
## [[43]]
## [1] "45"
##
## [[44]]
## [1] "45"
##
```

```
## [[45]]
## [1] "45"
##
## [[46]]
## [1] "45"
##
## [[47]]
## [1] "45"
##
## [[48]]
## [1] "45"
##
## [[49]]
## [1] "45"
##
## [[50]]
## [1] "45"
##
## [[51]]
## [1] "45"
##
## [[52]]
## [1] "45"
##
## [[53]]
## [1] "45"
##
## [[54]]
## [1] "45"
##
## [[55]]
## [1] "45"
##
## [[56]]
## [1] "45"
##
## [[57]]
## [1] "45"
##
## [[58]]
## [1] "45"
##
## [[59]]
## [1] "45"
##
## [[60]]
## [1] "45"
##
## [[61]]
## [1] "45"
##
## [[62]]
## [1] "45"
##
```

```
## [[63]]
## [1] "45"
##
## [[64]]
## [1] "45"
##
## [[65]]
## [1] "45"
##
## [[66]]
## [1] "45"
##
## [[67]]
## [1] "45"
##
## [[68]]
## [1] "45"
##
## [[69]]
## [1] "45"
##
## [[70]]
## [1] "45"
##
## [[71]]
## [1] "35" "45"
##
## [[72]]
## [1] "35" "45"
##
## [[73]]
## [1] "11" "45"
##
## [[74]]
## [1] "11" "45"
##
## [[75]]
## [1] "11" "45"
##
## [[76]]
## [1] "11" "45"
##
## [[77]]
## [1] "11" "45"
##
## [[78]]
## [1] "11" "45"
##
## [[79]]
## [1] "11" "45"
##
## [[80]]
## [1] "11" "45"
##
```

```
## [[81]]
## [1] "11" "45"
##
## [[82]]
## [1] "11" "45"
##
## [[83]]
## [1] "11" "45"
##
## [[84]]
## [1] "11" "45"
##
## [[85]]
## [1] "11" "45"
##
## [[86]]
## [1] "11" "45"
##
## [[87]]
## [1] "11" "45"
##
## [[88]]
## [1] "11" "45"
##
## [[89]]
## [1] "11" "45"
##
## [[90]]
## [1] "11" "45"
##
## [[91]]
## [1] "11" "45"
##
## [[92]]
## [1] "11" "45"
##
## [[93]]
## [1] "11" "45"
##
## [[94]]
## [1] "11" "45"
##
## [[95]]
## [1] "11" "45"
##
## [[96]]
## [1] "11" "45"
##
## [[97]]
## [1] "11" "45"
##
## [[98]]
## [1] "11" "45"
##
```

```
## [[99]]
## [1] "11" "45"
##
## [[100]]
## [1] "11" "45"
##
## [[101]]
## [1] "11" "45"
##
## [[102]]
## [1] "15"
##
## [[103]]
## [1] "45"
##
## [[104]]
## [1] "45"
##
## [[105]]
## [1] "45"
##
## [[106]]
## [1] "45"
##
## [[107]]
## [1] "45"
##
## [[108]]
## [1] "45"
##
## [[109]]
## [1] "45"
##
## [[110]]
## [1] "45"
##
## [[111]]
## [1] "45"
##
## [[112]]
## [1] "45"
##
## [[113]]
## [1] "45"
##
## [[114]]
## [1] "45"
##
## [[115]]
## [1] "45"
##
## [[116]]
## [1] "45"
##
```

```
## [[117]]
## [1] "45"
##
## [[118]]
## [1] "45"
##
## [[119]]
## [1] "45"
##
## [[120]]
## [1] "45"
##
## [[121]]
## [1] NA
##
## [[122]]
## [1] NA
##
## [[123]]
## [1] "45"
##
## [[124]]
## [1] NA
##
## [[125]]
## [1] NA
##
## [[126]]
## [1] NA
```

```
#PASTE
str_paste <- paste(split_str)
str_paste
```

```
##   [1] "NA"              "45"              "45"
##   [4] "45"              "45"              "45"
##   [7] "45"              "45"              "45"
##  [10] "45"              "45"              "45"
##  [13] "45"              "45"              "45"
##  [16] "45"              "45"              "45"
##  [19] "45"              "45"              "45"
##  [22] "45"              "45"              "45"
##  [25] "45"              "45"              "45"
##  [28] "45"              "45"              "45"
##  [31] "45"              "45"              "45"
##  [34] "45"              "45"              "45"
##  [37] "45"              "45"              "45"
##  [40] "45"              "45"              "45"
##  [43] "45"              "45"              "45"
##  [46] "45"              "45"              "45"
##  [49] "45"              "45"              "45"
##  [52] "45"              "45"              "45"
##  [55] "45"              "45"              "45"
##  [58] "45"              "45"              "45"
```

```
##  [61] "45"                "45"                "45"
##  [64] "45"                "45"                "45"
##  [67] "45"                "45"                "45"
##  [70] "45"                "c(\"35\", \"45\")" "c(\"35\", \"45\")"
##  [73] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [76] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [79] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [82] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [85] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [88] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [91] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [94] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
##  [97] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "c(\"11\", \"45\")"
## [100] "c(\"11\", \"45\")" "c(\"11\", \"45\")" "15"
## [103] "45"                "45"                "45"
## [106] "45"                "45"                "45"
## [109] "45"                "45"                "45"
## [112] "45"                "45"                "45"
## [115] "45"                "45"                "45"
## [118] "45"                "45"                "45"
## [121] "NA"                "NA"                "45"
## [124] "NA"                "NA"                "NA"
```