assignment4

Xi Cheng

February 21, 2018

10.5.1

```
## -- Attaching packages --------------- tidyverse 1.2.1 --
```

```
## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.2     v dplyr   0.7.4
## v tidyr   0.8.0     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.2.0
```

```
## -- Conflicts ------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
is.tibble(mtcars)
```

```
## [1] FALSE
```

So, if we can open the data.frame with tibble, then it's a tibble data.frame. If we have to open it with as_tibble(), then it is a regular data.frame. And also, we can use is.tibble() to find out if it is a tibble or just a normal data.frame.

10.5.2

```
## [1] a
## Levels: a
```

```
## [1] a
## Levels: a
```

```
##   abc xyz
## 1   1   a
```

if we want to subset some data from regular data.frame, we have to consider about the rows and columns. However, if we use tibble, we could just consider the columns. ##10.5.3

var <- "mpg" var[["mpg"]] var %>% .$mpg var %>% .[["mpg"]]

10.5.4

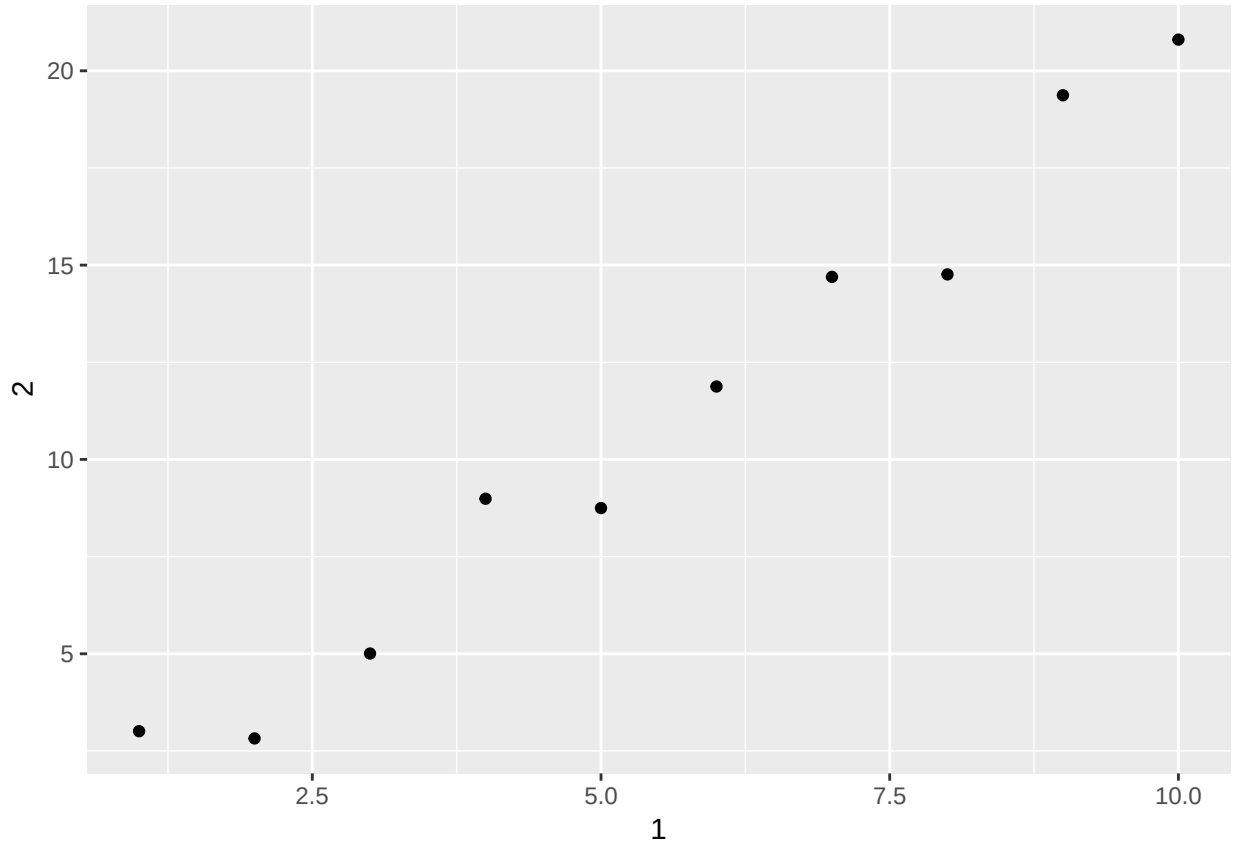
```
## # A tibble: 10 x 2
##      `1`   `2`
##    <int> <dbl>
##  1     1  3.01
##  2     2  2.82
##  3     3  5.01
##  4     4  8.99
##  5     5  8.75
##  6     6 11.9
##  7     7 14.7
```

```
##  8     8 14.8
##  9     9 19.4
## 10    10 20.8
```

10.5.4.1

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

10.5.4.2



10.5.4.3

```
## # A tibble: 10 x 3
##      `1`   `2`   `3`
##    <int> <dbl> <dbl>
##  1     1  3.01  3.01
##  2     2  2.82  1.41
##  3     3  5.01  1.67
##  4     4  8.99  2.25
##  5     5  8.75  1.75
##  6     6 11.9   1.98
##  7     7 14.7   2.10
##  8     8 14.8   1.85
```

```
##  9     9 19.4   2.15
## 10    10 20.8   2.08
```

```
## # A tibble: 10 x 3
##      `1`   `2`   `3`
##    <int> <dbl> <dbl>
##  1     1  3.01  3.01
##  2     2  2.82  1.41
##  3     3  5.01  1.67
##  4     4  8.99  2.25
##  5     5  8.75  1.75
##  6     6 11.9   1.98
##  7     7 14.7   2.10
##  8     8 14.8   1.85
##  9     9 19.4   2.15
## 10    10 20.8   2.08
```

10.5.4.4

```
## # A tibble: 10 x 3
##      one   two three
##    <int> <dbl> <dbl>
##  1     1  3.01  3.01
##  2     2  2.82  1.41
##  3     3  5.01  1.67
##  4     4  8.99  2.25
##  5     5  8.75  1.75
##  6     6 11.9   1.98
##  7     7 14.7   2.10
##  8     8 14.8   1.85
##  9     9 19.4   2.15
## 10    10 20.8   2.08
```

10.5.5

enframe() converts named atomic vectors or lists to two-column data frames. For unnamed vectors, the natural sequence is used as name column.

10.5.6

print.tbl_df

12.6.1

```
## # A tibble: 7,240 x 60
##    country     iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534
##    <chr>       <chr> <chr> <int>       <int>        <int>        <int>
##  1 Afghanistan AF    AFG    1980          NA           NA           NA
##  2 Afghanistan AF    AFG    1981          NA           NA           NA
##  3 Afghanistan AF    AFG    1982          NA           NA           NA
```

```
##  4 Afghanistan AF    AFG    1983          NA           NA           NA
##  5 Afghanistan AF    AFG    1984          NA           NA           NA
##  6 Afghanistan AF    AFG    1985          NA           NA           NA
##  7 Afghanistan AF    AFG    1986          NA           NA           NA
##  8 Afghanistan AF    AFG    1987          NA           NA           NA
##  9 Afghanistan AF    AFG    1988          NA           NA           NA
## 10 Afghanistan AF    AFG    1989          NA           NA           NA
## # ... with 7,230 more rows, and 53 more variables: new_sp_m3544 <int>,
## #   new_sp_m4554 <int>, new_sp_m5564 <int>, new_sp_m65 <int>,
## #   new_sp_f014 <int>, new_sp_f1524 <int>, new_sp_f2534 <int>,
## #   new_sp_f3544 <int>, new_sp_f4554 <int>, new_sp_f5564 <int>,
## #   new_sp_f65 <int>, new_sn_m014 <int>, new_sn_m1524 <int>,
## #   new_sn_m2534 <int>, new_sn_m3544 <int>, new_sn_m4554 <int>,
## #   new_sn_m5564 <int>, new_sn_m65 <int>, new_sn_f014 <int>,
## #   new_sn_f1524 <int>, new_sn_f2534 <int>, new_sn_f3544 <int>,
## #   new_sn_f4554 <int>, new_sn_f5564 <int>, new_sn_f65 <int>,
## #   new_ep_m014 <int>, new_ep_m1524 <int>, new_ep_m2534 <int>,
## #   new_ep_m3544 <int>, new_ep_m4554 <int>, new_ep_m5564 <int>,
## #   new_ep_m65 <int>, new_ep_f014 <int>, new_ep_f1524 <int>,
## #   new_ep_f2534 <int>, new_ep_f3544 <int>, new_ep_f4554 <int>,
## #   new_ep_f5564 <int>, new_ep_f65 <int>, newrel_m014 <int>,
## #   newrel_m1524 <int>, newrel_m2534 <int>, newrel_m3544 <int>,
## #   newrel_m4554 <int>, newrel_m5564 <int>, newrel_m65 <int>,
## #   newrel_f014 <int>, newrel_f1524 <int>, newrel_f2534 <int>,
## #   newrel_f3544 <int>, newrel_f4554 <int>, newrel_f5564 <int>,
## #   newrel_f65 <int>
```

```
## # A tibble: 56 x 2
##    key              n
##    <chr>        <int>
##  1 new_ep_f014   1032
##  2 new_ep_f1524  1021
##  3 new_ep_f2534  1021
##  4 new_ep_f3544  1021
##  5 new_ep_f4554  1017
##  6 new_ep_f5564  1017
##  7 new_ep_f65    1014
##  8 new_ep_m014   1038
##  9 new_ep_m1524  1026
## 10 new_ep_m2534  1020
## # ... with 46 more rows
```

```
## # A tibble: 1 x 2
##   new       n
##   <chr> <int>
## 1 new   76046
```

```
## # A tibble: 76,046 x 6
##    country      year type  sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1998 sp    m     014      30
##  3 Afghanistan  1999 sp    m     014       8
##  4 Afghanistan  2000 sp    m     014      52
##  5 Afghanistan  2001 sp    m     014     129
```

```
##  6 Afghanistan  2002 sp    m     014      90
##  7 Afghanistan  2003 sp    m     014     127
##  8 Afghanistan  2004 sp    m     014     139
##  9 Afghanistan  2005 sp    m     014     151
## 10 Afghanistan  2006 sp    m     014     193
## # ... with 76,036 more rows
```

we use na.rm = T in this case is reasonable, because there are lots of NA values in the data which is useless. we can use count(cases=NA) find out the all these missing values.

```
count(who,cases = NA)
```

```
## # A tibble: 1 x 2
##   cases     n
##   <lgl> <int>
## 1 NA     7240
```

12.6.2

The code will not be separated properly into the three columns new, var, and sexage.

12.6.3

```
## # A tibble: 0 x 3
## # Groups:   country [0]
## # ... with 3 variables: country <chr>, iso2 <chr>, iso3 <chr>
```

12.6.4