week13 exercise

Xiang Li

2023/12/31

```
library(brolgar)
library(ptmixed)
library(reshape2)
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
```

```
library(magrittr)
```

Exercise 1

```
df_long = as.data.frame(heights)
```

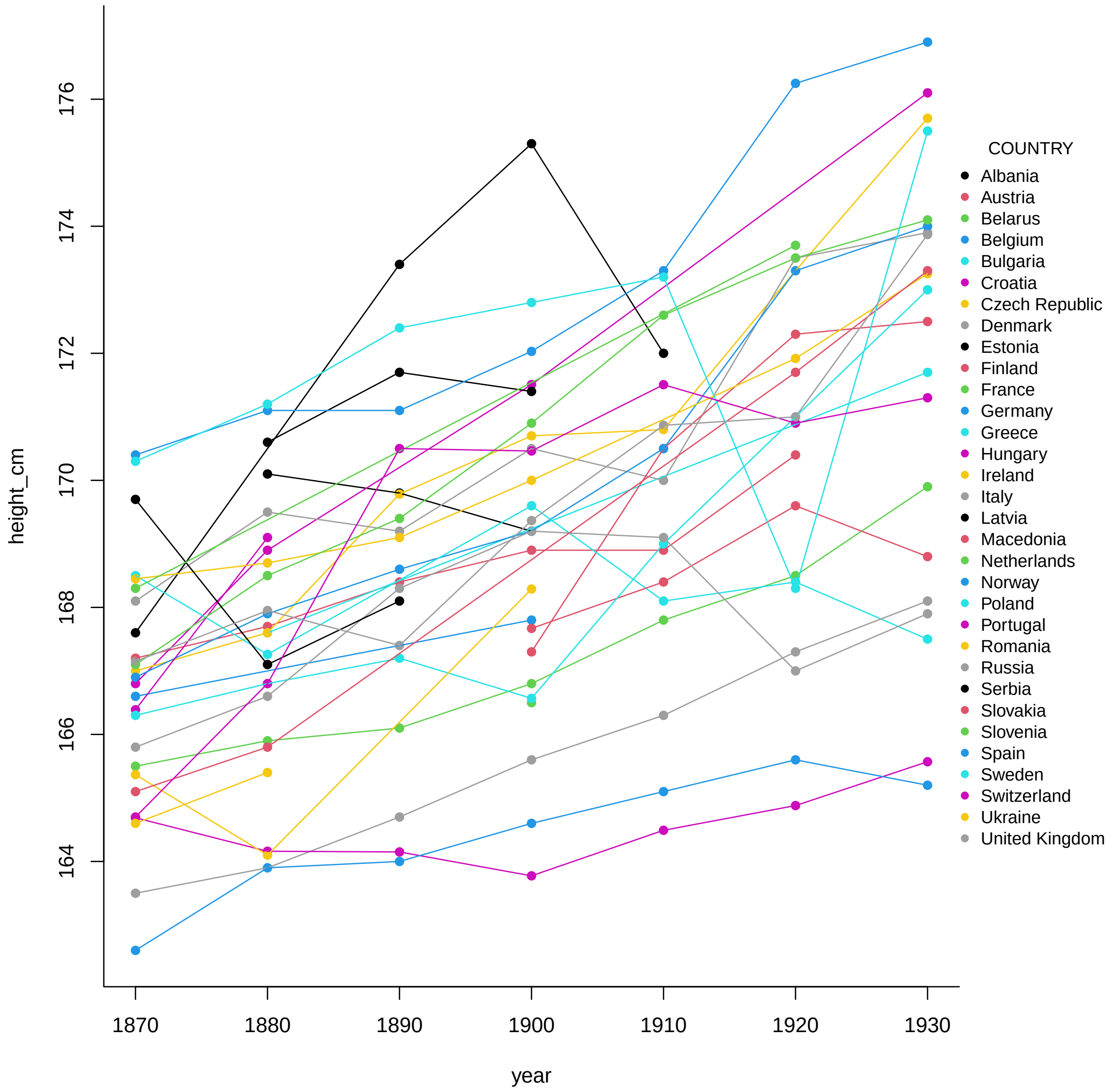
1

```
df_long1 = df_long[(df_long$continent == "Europe") & (df_long$year >=
    1870) & (df_long$year <= 1930), ]
head(df_long1, 5)
```

```
##    country continent year height_cm
## 6  Albania    Europe 1880     170.1
## 7  Albania    Europe 1890     169.8
## 8  Albania    Europe 1900     169.2
## 74 Austria    Europe 1870     167.2
## 75 Austria    Europe 1880     167.7
```

2

```
coun_n = length(unique(df_long1$country))
make.spaghetti(year, height_cm, id = country, group = country,
    data = df_long1, legend.title = "COUNTRY", col = 1:coun_n,
    cex.leg = 0.85, legend.inset = -0.18)
```



Exercise 2

1

```
df_long_wd = dcast(df_long, country + continent ~ year,
    value.var = "height_cm")
head(df_long_wd, 5)
```

```
##       country continent 1550 1650 1660 1670 1680 1690 1700 1710 1720 1730 1740
## 1 Afghanistan      Asia   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2     Albania    Europe   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3     Algeria    Africa   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 4      Angola    Africa   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 5   Argentina  Americas   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1750 1760  1770  1780  1790  1800  1810  1820  1830  1840  1850  1860  1870
## 1   NA   NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA 168.4
## 2   NA   NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 3   NA   NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 4   NA   NA    NA    NA 160.4 158.6 160.5    NA    NA    NA    NA    NA    NA
## 5   NA   NA 170.3 168.2 168.0 168.0 168.8 169.9 170.9 169.6 168.2 167.4 167.6
##      1880    1890    1900  1910    1920  1930  1940  1950 1960 1970 1980  1990
## 1 165.690      NA      NA    NA      NA 166.8    NA    NA   NA   NA   NA 167.1
## 2 170.100 169.800 169.200    NA      NA    NA    NA    NA   NA   NA   NA    NA
## 3      NA      NA      NA 168.8 166.241 169.0    NA    NA   NA   NA   NA 171.3
## 4 168.800 169.100 168.100 168.0 165.700 166.7    NA    NA   NA   NA   NA    NA
## 5 167.565 167.792 167.868 168.2 169.000 169.8 170.6 170.8   NA   NA   NA 174.4
##    2000
## 1 161.4
## 2 167.9
## 3 169.5
## 4    NA
## 5    NA
```

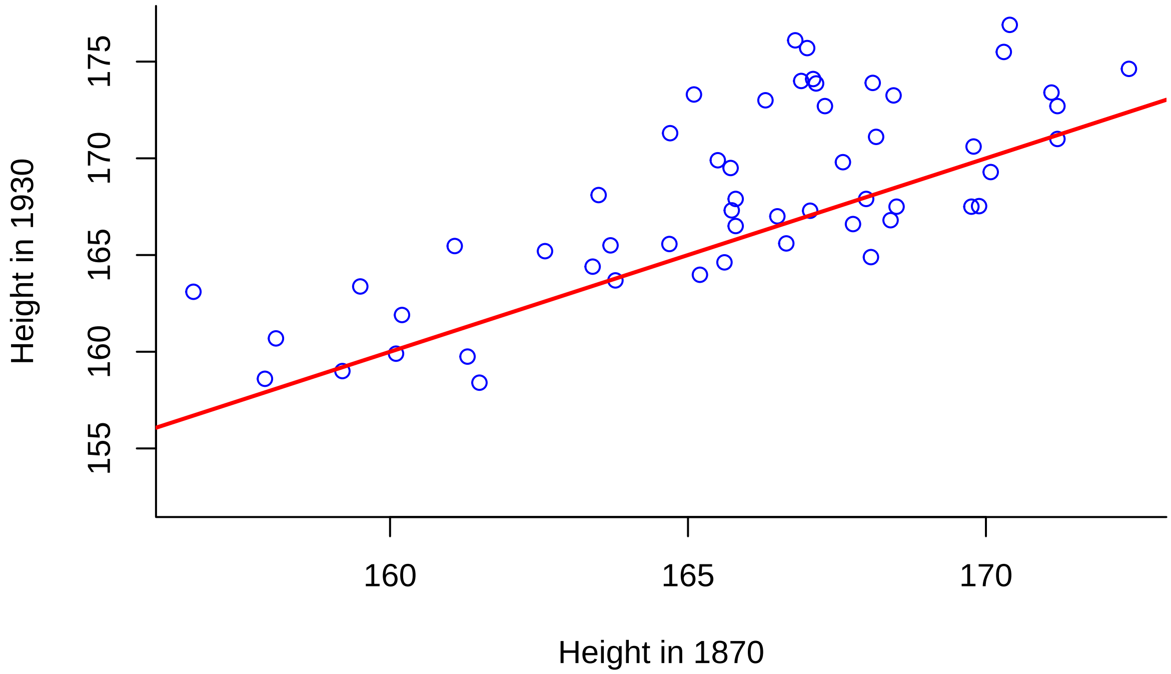
2

```
rename_f = function(x) {
    paste0("height_", as.character(x))
}
colnames(df_long_wd)[3:ncol(df_long_wd)] = sapply(colnames(df_long_wd)[3:ncol(df_long_wd)],
    rename_f)
head(df_long_wd, 5)
```

```
##       country continent height_1550 height_1650 height_1660 height_1670
## 1 Afghanistan      Asia          NA          NA          NA          NA
## 2     Albania    Europe          NA          NA          NA          NA
## 3     Algeria    Africa          NA          NA          NA          NA
## 4      Angola    Africa          NA          NA          NA          NA
## 5   Argentina  Americas          NA          NA          NA          NA
##   height_1680 height_1690 height_1700 height_1710 height_1720 height_1730
## 1          NA          NA          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA          NA
## 3          NA          NA          NA          NA          NA          NA
## 4          NA          NA          NA          NA          NA          NA
## 5          NA          NA          NA          NA          NA          NA
##   height_1740 height_1750 height_1760 height_1770 height_1780 height_1790
## 1          NA          NA          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA          NA
## 3          NA          NA          NA          NA          NA          NA
## 4          NA          NA          NA          NA          NA       160.4
## 5          NA          NA          NA       170.3       168.2       168.0
##   height_1800 height_1810 height_1820 height_1830 height_1840 height_1850
## 1          NA          NA          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA          NA
## 3          NA          NA          NA          NA          NA          NA
## 4       158.6       160.5          NA          NA          NA          NA
## 5       168.0       168.8       169.9       170.9       169.6       168.2
##   height_1860 height_1870 height_1880 height_1890 height_1900 height_1910
## 1          NA       168.4     165.690          NA          NA          NA
## 2          NA          NA     170.100     169.800     169.200          NA
## 3          NA          NA          NA          NA          NA       168.8
## 4          NA          NA     168.800     169.100     168.100       168.0
## 5       167.4       167.6     167.565     167.792     167.868       168.2
##   height_1920 height_1930 height_1940 height_1950 height_1960 height_1970
## 1          NA       166.8          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA          NA
## 3     166.241       169.0          NA          NA          NA          NA
## 4     165.700       166.7          NA          NA          NA          NA
## 5     169.000       169.8       170.6       170.8          NA          NA
##   height_1980 height_1990 height_2000
## 1          NA       167.1       161.4
## 2          NA          NA       167.9
## 3          NA       171.3       169.5
## 4          NA          NA          NA
## 5          NA       174.4          NA
```

3

```
par(bty = "l")
plot(x = df_long_wd$height_1870, y = df_long_wd$height_1930,
    type = "p", col = "blue", xlab = "Height in 1870", ylab = "Height in 1930")
abline(a = 0, b = 1, col = "red", lwd = 2)
```



4

For most countries, height increase during 1870 to 1930. But there are some exceptions.

Exercise 3

1

```
load("data/data_metadata_country.RData")
```

2

Horizontal merge.

3

Country.Code can be key for the merge.

4

```
population = select(population, !X)
metadata = select(metadata, !X)
data_df = merge(population, metadata, by = "Country.Code",
    all = T)
head(data_df, 5)
```

```
##   Country.Code                Country.Name    Indicator.Name Indicator.Code
## 1          ABW                       Aruba Population, total    SP.POP.TOTL
## 2          AFE Africa Eastern and Southern Population, total    SP.POP.TOTL
## 3          AFG                 Afghanistan Population, total    SP.POP.TOTL
## 4          AFW  Africa Western and Central Population, total    SP.POP.TOTL
## 5          AGO                      Angola Population, total    SP.POP.TOTL
##       X1960     X1961     X1962     X1963     X1964     X1965     X1966
## 1     54608     55811     56682     57475     58178     58782     59291
## 2 130692579 134169237 137835590 141630546 145605995 149742351 153955516
## 3   8622466   8790140   8969047   9157465   9355514   9565147   9783147
## 4  97256290  99314028 101445032 103667517 105959979 108336203 110798486
## 5   5357195   5441333   5521400   5599827   5673199   5736582   5787044
##       X1967     X1968     X1969     X1970     X1971     X1972     X1973
## 1     59522     59471     59330     59106     58816     58855     59365
## 2 158313235 162875171 167596160 172475766 177503186 182599092 187901657
## 3  10010030  10247780  10494489  10752971  11015857  11286753  11575305
## 4 113319950 115921723 118615741 121424797 124336039 127364044 130563107
## 5   5827503   5868203   5928386   6029700   6177049   6364731   6578230
##       X1974     X1975     X1976     X1977     X1978     X1979     X1980
## 1     60028     60715     61193     61465     61738     62006     62267
## 2 193512956 199284304 205202669 211120911 217481420 224315978 230967858
## 3  11869879  12157386  12425267  12687301  12938862  12986369  12486631
## 4 133953892 137548613 141258400 145122851 149206663 153459665 157825609
## 5   6802494   7032713   7266780   7511895   7771590   8043218   8330047
##       X1981     X1982     X1983     X1984     X1985     X1986     X1987
## 1     62614     63116     63683     64174     64478     64553     64450
## 2 237937461 245386717 252779730 260209149 267938123 276035920 284490394
## 3  11155195  10088289   9951449  10243686  10512221  10448442  10322758
## 4 162323313 167023385 171566640 176054495 180817312 185720244 190759952
## 5   8631457   8947152   9276707   9617702   9970621  10332574  10694057
##       X1988     X1989     X1990     X1991     X1992     X1993     X1994
## 1     64332     64596     65712     67864     70192     72360     74710
## 2 292795186 301124880 309890664 318544083 326933522 335625136 344418362
## 3  10383460  10673168  10694796  10745167  12057433  14003760  15455555
## 4 195969722 201392200 206739024 212172888 217966101 223788766 229675775
## 5  11060261  11439498  11828638  12228691  12632507  13038270  13462031
##       X1995     X1996     X1997     X1998     X1999     X2000     X2001
## 1     77050     79417     81858     84355     86867     89101     90691
## 2 353466601 362985802 372352230 381715600 391486231 401600588 412001885
## 3  16418912  17106595  17788819  18493132  19262847  19542982  19688632
## 4 235861484 242200260 248713095 255482918 262397030 269611898 277160097
## 5  13912253  14383350  14871146  15366864  15870753  16394062  16941587
##       X2002     X2003     X2004     X2005     X2006     X2007     X2008
## 1     91781     92701     93540     94483     95606     96787     97996
## 2 422741118 433807484 445281555 457153837 469508516 482406426 495748900
## 3  21000256  22645130  23553551  24411191  25442944  25903301  26427199
## 4 284952322 292977949 301265247 309824829 318601484 327612838 336893835
## 5  17516139  18124342  18771125  19450959  20162340  20909684  21691522
##       X2009     X2010     X2011     X2012     X2013     X2014     X2015
## 1     99212    100341    101288    102112    102880    103594    104257
## 2 509410477 523459657 537792950 552530654 567892149 583651101 600008424
## 3  27385307  28189672  29249157  30466479  31541209  32716210  33753499
## 4 346475221 356337762 366489204 376797999 387204553 397855507 408690375
## 5  22507674  23364185  24259111  25188292  26147002  27128337  28127721
##       X2016     X2017     X2018     X2019     X2020     X2021     X2022
## 1    104874    105439    105962    106442    106585    106537    106445
## 2 616377605 632746570 649757148 667242986 685112979 702977106 720839314
## 3  34636207  35643418  36686784  37769499  38972230  40099462  41128771
## 4 419778384 431138704 442646825 454306063 466189102 478185907 490330870
## 5  29154746  30208628  31273533  32353588  33428486  34503774  35588987
##                      Region         IncomeGroup
## 1 Latin America & Caribbean         High income
## 2
## 3                South Asia          Low income
## 4
## 5        Sub-Saharan Africa Lower middle income
##
## 1
## 2
## 3 The reporting period for national accounts data is designated as either calendar year basis (CY) o
## 4
## 5
##                     TableName
## 1                       Aruba
## 2 Africa Eastern and Southern
## 3                 Afghanistan
## 4  Africa Western and Central
## 5                      Angola
```

5

```
data_df = filter(data_df, !Region == "")
```

6

a)

```
summarize(group_by(data_df, Region), large2020 = max(X2020))
```

```
## # A tibble: 7 x 2
##   Region                    large2020
##   <chr>                         <dbl>
## 1 East Asia & Pacific        1411100000
## 2 Europe & Central Asia       144073139
## 3 Latin America & Caribbean   213196304
## 4 Middle East & North Africa  107465134
## 5 North America               331511512
## 6 South Asia                 1396387127
## 7 Sub-Saharan Africa          208327405
```

b)

```
summarize(group_by(data_df, Region), large2020_country = Country.Name[which.max(X2020)])
```

```
## # A tibble: 7 x 2
##   Region                     large2020_country 
##   <chr>                      <chr>             
## 1 East Asia & Pacific        China             
## 2 Europe & Central Asia      Russian Federation
## 3 Latin America & Caribbean  Brazil            
## 4 Middle East & North Africa Egypt, Arab Rep.  
## 5 North America              United States     
## 6 South Asia                 India             
## 7 Sub-Saharan Africa         Nigeria
```

Exercise 4

```
data(iris)
df.list = split(iris, iris$Species)
lapply(df.list, summary)
```

```
## $setosa
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100  
##  1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200  
##  Median :5.000   Median :3.400   Median :1.500   Median :0.200  
##  Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246  
##  3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300  
##  Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600  
##        Species  
##  setosa    :50  
##  versicolor: 0  
##  virginica : 0  
##                 
##                 
##                 
## 
## $versicolor
##   Sepal.Length    Sepal.Width     Petal.Length   Petal.Width          Species  
##  Min.   :4.900   Min.   :2.000   Min.   :3.00   Min.   :1.000   setosa    : 0  
##  1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.00   1st Qu.:1.200   versicolor:50  
##  Median :5.900   Median :2.800   Median :4.35   Median :1.300   virginica : 0  
##  Mean   :5.936   Mean   :2.770   Mean   :4.26   Mean   :1.326
##  3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.60   3rd Qu.:1.500
##  Max.   :7.000   Max.   :3.400   Max.   :5.10   Max.   :1.800
##
## $virginica
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.900   Min.   :2.200   Min.   :4.500   Min.   :1.400
##  1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100   1st Qu.:1.800
##  Median :6.500   Median :3.000   Median :5.550   Median :2.000
##  Mean   :6.588   Mean   :2.974   Mean   :5.552   Mean   :2.026
##  3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875   3rd Qu.:2.300
##  Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    : 0
##  versicolor: 0
##  virginica :50
##
##
##
```

1

```
iris |>
    split(iris$Species) |>
    lapply(summary)
```

```
## $setosa
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100
##  1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200
##  Median :5.000   Median :3.400   Median :1.500   Median :0.200
##  Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246
##  3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300
##  Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600
##        Species
##  setosa    :50
##  versicolor: 0
##  virginica : 0
##
##
##
##
## $versicolor
##   Sepal.Length    Sepal.Width     Petal.Length   Petal.Width          Species
##  Min.   :4.900   Min.   :2.000   Min.   :3.00   Min.   :1.000   setosa    : 0
##  1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.00   1st Qu.:1.200   versicolor:50
##  Median :5.900   Median :2.800   Median :4.35   Median :1.300   virginica : 0
##  Mean   :5.936   Mean   :2.770   Mean   :4.26   Mean   :1.326
##  3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.60   3rd Qu.:1.500
##  Max.   :7.000   Max.   :3.400   Max.   :5.10   Max.   :1.800
##
## $virginica
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.900   Min.   :2.200   Min.   :4.500   Min.   :1.400  
##  1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100   1st Qu.:1.800  
##  Median :6.500   Median :3.000   Median :5.550   Median :2.000  
##  Mean   :6.588   Mean   :2.974   Mean   :5.552   Mean   :2.026  
##  3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875   3rd Qu.:2.300  
##  Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    : 0  
##  versicolor: 0  
##  virginica :50  
##                 
##                 
## 
```

2

```
iris %>%
    split(iris$Species) %>%
    lapply(summary)
```

```
## $setosa
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100  
##  1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200  
##  Median :5.000   Median :3.400   Median :1.500   Median :0.200  
##  Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246  
##  3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300  
##  Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600  
##        Species  
##  setosa    :50  
##  versicolor: 0  
##  virginica : 0  
##                 
##                 
##                 
## 
## $versicolor
##   Sepal.Length    Sepal.Width     Petal.Length   Petal.Width          Species  
##  Min.   :4.900   Min.   :2.000   Min.   :3.00   Min.   :1.000   setosa    : 0  
##  1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.00   1st Qu.:1.200   versicolor:50  
##  Median :5.900   Median :2.800   Median :4.35   Median :1.300   virginica : 0  
##  Mean   :5.936   Mean   :2.770   Mean   :4.26   Mean   :1.326                  
##  3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.60   3rd Qu.:1.500                  
##  Max.   :7.000   Max.   :3.400   Max.   :5.10   Max.   :1.800                  
## 
## $virginica
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.900   Min.   :2.200   Min.   :4.500   Min.   :1.400  
##  1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100   1st Qu.:1.800  
##  Median :6.500   Median :3.000   Median :5.550   Median :2.000  
##  Mean   :6.588   Mean   :2.974   Mean   :5.552   Mean   :2.026  
##  3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875   3rd Qu.:2.300  
##  Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    : 0
##  versicolor: 0
##  virginica :50
##
##
##
```

Exercise 5

1

```
data_df = read.csv("data/irish_polls.csv")
data_df[data_df == "Not Available"] = NA
to_decimal = function(x) {
    return(sub("%", "", x))
}
data_df[, 10:21] = lapply(data_df[, 10:21], to_decimal)
data_df[, 10:21] = as.numeric(unlist(data_df[, 10:21]))/100
```

2

```
data_df = filter(data_df, (Fieldwork.End >= "2021-05-27") &
    (Fieldwork.End <= "2021-09-09"))
colna = c(colnames(data_df)[1:9], colnames(data_df)[10:21][!apply(is.na(data_df[,
    10:21]), 2, any)])
data_df = data_df[, colna]
```

3

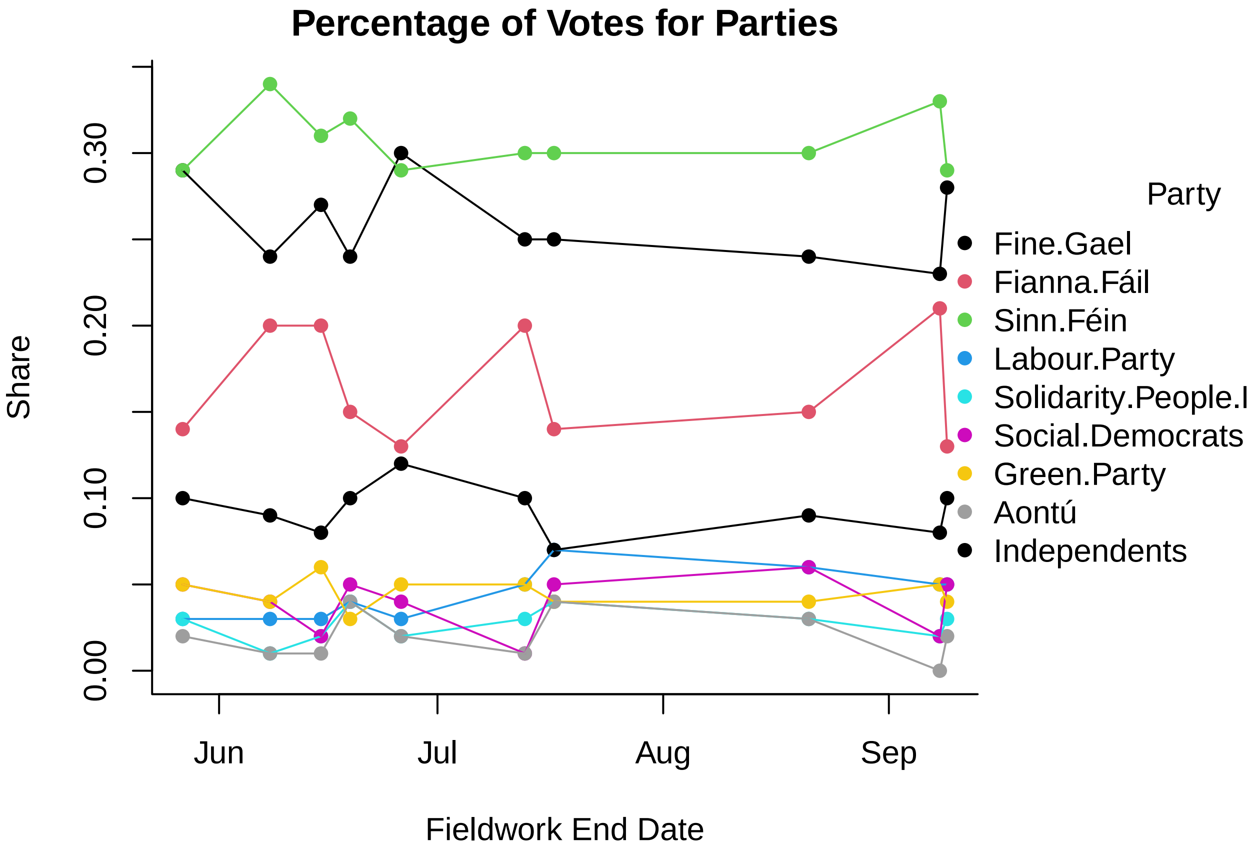
```
data_df = mutate(data_df, Fieldwork.End = as.Date(Fieldwork.End,
    "%Y-%m-%d"))
data_df_long = melt(data_df, id.vars = colnames(data_df)[1:9])
head(data_df_long, 5)
```

```
##              Polling.Firm        Commissioners Fieldwork.Start Fieldwork.End
## 1                   Red C         Business Post      2021-09-03    2021-09-09
## 2 Behaviour and Attitudes      The Sunday Times      2021-08-26    2021-09-08
## 3          Ireland Thinks Irish Mail on Sunday      2021-08-21    2021-08-21
## 4          Ireland Thinks Irish Mail on Sunday      2021-07-17    2021-07-17
## 5 Behaviour and Attitudes      The Sunday Times      2021-07-01    2021-07-13
##      Scope Sample.Size Sample.Size.Qualification Participation Precision
## 1 National        1031                  Provided          <NA>        1%
## 2 National         922                  Provided          <NA>        1%
## 3 National        1203                  Provided          <NA>        1%
## 4 National        1001                    Provided          <NA>         1%
## 5 National        1001                    Provided          <NA>         1%
##    variable value
## 1 Fine.Gael  0.28
## 2 Fine.Gael  0.23
## 3 Fine.Gael  0.24
## 4 Fine.Gael  0.25
## 5 Fine.Gael  0.25
```

```
data_df_long = rename(data_df_long, Party = variable, Share = value)
```

4

```
coun_n = length(unique(data_df_long$Party))
make.spaghetti(Fieldwork.End, Share, id = Party, group = Party,
    data = data_df_long, legend.title = "Party", col = 1:coun_n,
    cex.leg = 1, legend.inset = -0.55, xlab = "Fieldwork End Date",
    ylab = "Share", title = "Percentage of Votes for Parties")
```



5

```
mean_share = data_df_long |>
    group_by(Party) |>
    summarize(mean = mean(Share))
mean_share
```

```
## # A tibble: 9 x 2
##   Party                            mean
##   <fct>                           <dbl>
## 1 Fine.Gael                       0.259
## 2 Fianna.Fáil                     0.165
## 3 Sinn.Féin                       0.307
## 4 Labour.Party                    0.044
## 5 Solidarity.People.Before.Profit 0.027
## 6 Social.Democrats                0.039
## 7 Green.Party                     0.045
## 8 Aontú                           0.02
## 9 Independents                    0.093
```

6

```
mean_share$Party[which.max(mean_share$mean)]
```

```
## [1] Sinn.Féin
## 9 Levels: Fine.Gael Fianna.Fáil Sinn.Féin ... Independents
```

Yes, the line of Sinn.Fein is in the highest position.