

Midterm Project

Stat 133, Fall 2016, Prof. Sanchez

Due Date: October 21, 2016

Abstract: The purpose of this project is to put in practice the main concepts that we've seen so far in the course:

- working with various types of data objects
- using control flow structures
- writing functions
- manipulation of character strings with "stringr"
- data manipulation with "dplyr"
- data visualization with "ggplot2"

It is possible that you may need to learn new concepts (and consult topics) not covered during lecture or lab.

General Instructions

Please use the provided .Rmd and .R file templates available in bCourses. And don't forget to include your name!

When writing commands and functions for this assignment:

- Use a consistent naming style.
- Include a description of what the function does.
- Include a description for the expected input(s) of a function.
- Include a description for the returned output of a function.
- Avoid writing long functions, and strive to limit the number of lines in the body of a function to less than 10 lines of code.
- Likewise, limit the line-width of your code to a max of 80 characters. If you have long lines, then split them into shorter lines, e.g.:

```
# avoid long line
dat = read.csv("~/Documents/stat133/project1/data/dataset.csv", stringsAsFactors = FALSE)

# better (split long lines)
dat = read.csv(
  "~/Documents/stat133/project1/data/dataset.csv",
  stringsAsFactors = FALSE)
```

- Include comments but don't belabor the obvious.
- Use white spaces:

```
# avoid lack of spaces
dat=read.csv("dataset.csv",header=FALSE,row.names=1,dec=';',skip=1)
a=seq(from=1,to=100,by=1)

# better (use spaces!)
dat = read.csv("dataset.csv", header = FALSE, row.names = 1, dec = ';', skip = 1)
a = seq(from = 1, to = 100, by = 1)
```

- Use **indentation** (RStudio usually does this for you).
- Lack of indentation, lack of **white spaces**, inconsistent **naming style**, lines **exceeding 80 chars-width**, and code that in general is **hard to “read”** or **“review”** will be penalized.
- If you are not sure about the appearance of your code, apply code **reformatting** by using the **Reformat Code** option from **Code** in RStudio’s menu bar.

In addition to checking whether you meet all the listed project requirements, we will evaluate the following core competencies of your report:

- **Computation:** Perform computations correctly.
- **Analysis:** Carry out **analysis appropriate** for data and context.
- **Synthesis:** Identify **key features of the analysis**, and interpret results.
- **Visual presentation:** **communicate findings graphically clearly.**
- **Verbal:** **communicate findings in writing clearly, precisely and concisely.**

This project is an **individual** project. You may discuss its different parts with other students, but you must **independently write your code** and **solutions**. For example, suggesting a function or a package to another student is acceptable, whereas simply giving him or her your own code is not. If you are not clear about the expectations for completing this assignment, be sure to seek clarification from the instructor or GSI beforehand.








We reserve the right to **meet with you** and **ask you questions** (verbally, in written form, or coding in Rstudio) about your submitted project, your code, and the approach that you use to solve the questions. Failing to have such a meeting may cause losing all points in this project.








About the Data Sets

High Jump World Records

You will be working with data about **High Jump World Records** for both **women and men** (see screenshots below). The original sources are available in wikipedia:

- https://en.wikipedia.org/wiki/Women%27s_high_jump_world_record_progression
- https://en.wikipedia.org/wiki/Men%27s_high_jump_world_record_progression

Height ↕	Athlete ↕	Date ↕	Place ↕
1.46 m (4 ft 9½ in)	 Nancy Voorhees (USA)	20 May 1922	Simsbury ^[1]
1.485 m (4 ft 10½ in)	 Elizabeth Stine (USA)	26 May 1923	Leonia ^[1]
1.485 m (4 ft 10½ in)	 Sophie Elliott-Lynn (GBR)	6 August 1923	Brentwood ^[1]
1.524 m (5 ft 0 in)	 Phyllis Green (GBR)	11 July 1925	London ^[1]
1.552 m (5 ft 1⅛ in)	 Phyllis Green (GBR)	2 August 1926	London ^[1]
1.58 m (5 ft 2¼ in)	 Ethel Catherwood (CAN)	6 September 1926	Regina ^[1]
1.58 m (5 ft 2¼ in)	 Lien Gisolf (NED)	3 July 1928	Brussels ^[1]

Height	Athlete	Venue	Date
2.00 m (6 ft 6 ³ / ₄ in)	 George Horine (USA)	Palo Alto, California	18 May 1912 ^[1]
2.022 m (6 ft 7 ⁵ / ₈ in)	 Edward Beeson (USA)	Berkeley, California	2 May 1914 ^[3]
2.038 m (6 ft 8 ¹ / ₄ in)	 Harold Osborn (USA)	Urbana, Illinois	27 May 1924 ^[4]
2.04 m (6 ft 8 ³ / ₈ in)	 Walter Marty (USA)	Fresno, California	13 May 1933 ^[1]
2.06 m (6 ft 9 ¹ / ₈ in)	 Walter Marty (USA)	Palo Alto, California	28 April 1934 ^[1]
2.07 m (6 ft 9 ¹ / ₂ in)	 Cornelius Johnson (USA)	New York	12 July 1936 ^[1]
2.07 m (6 ft 9 ¹ / ₂ in)	 Dave Albritton (USA)	New York	12 July 1936 ^[1]

I've scraped the data tables and saved them in two CSV files available in the course's github repository:

- <https://raw.githubusercontent.com/ucb-stat133/stat133-fall-2016/master/data/womens-high-jump-raw.csv>
- <https://raw.githubusercontent.com/ucb-stat133/stat133-fall-2016/master/data/mens-high-jump-raw.csv>

About the Problems

This project is divided in **five parts** (please follow further instructions in the **.Rmd** template file available on bCourses):

- **Functions:** programming auxiliary functions that will help you clean the raw data sets.
- **Data cleaning:** this part involves working with the raw messy data sets in order to produce two clean data frames: one for women, one for men.
- **Exploratory Data Analysis:** this has to do with computing a series of **summary measures**, and summary tables.
- **Data Visualization:** production of various plots that allow the audience to look at the progression of high jump world records.
- **Model Fitting:** consists of fitting a **least squares regression line** to model the height values in terms of time. This model will be used to extrapolate what the world records could have been various years.

It is possible you may need to **learn about functions** and/or **other R capabilities** that we didn't cover in lecture, labs, or HW. If you believe there is a concept that falls in this category, google it first. If you don't find any information, or the concepts are still unclear, then ask the lab assistants, GSIs, or the instructor.

Functions

To **clean the raw data sets** you must write various functions. **All the code for the functions must be written** in an **.R script using the provided template file**. Some functions that you may find useful are **str_extract()**, **str_split()**, **str_replace()**, and/or **str_sub()**, from the package **"stringr"**.

Extract Height: write a **function to extract the numbers** corresponding to the **height value in meters**. This function will help you **clean the column Height**. The output must be a **numeric vector** with the values for the records (corresponding to meters).

Extract Athlete's Name: write a function to extract the names of the athletes. This function will help you clean the column `Athlete`. This column contains the name of the athlete, together with the country (inside parenthesis). Your function must return a character vector with just the first and last names of athletes.

Extract Country: write a function to extract the name of the countries. This function will help you clean the column `Athlete` by extracting just the abbreviation of the countries. Your function must return a character vector containing the initials of the countries (without no parenthesis): e.g. "USA", "USA", "GBR",

Remove Brackets: write a function to remove the brackets (and the numbers inside them), that appear in some of the columns of both data sets (e.g. column `Date` for mens, column `Place` for womens). This function must return a "clean" character vector with no brackets (and no numbers inside the brackets).

Extract Day: write a function to extract the day number of the column `Date`. This function must return a numeric vector with such day numbers.

Extract Month: write a function to extract the name of the month from the column `Date`. This function must return a character vector with the names of the months.

Reformat Date: write a function to reformat the date. This function must return a vector of class "Date" with format "%d %B %Y".

Extract City: write a function to extract the city name from the column `Place` (for womens data). This function must return a character vector with just the name fo the city.

Clean data sets:

Your clean womens data frame should look like this one (6 first rows displayed)

	height	athlete	gender	country	city	date	day	month	year
1	1.460	Nancy Voorhees	female	USA	Simsbury	1922-05-20	20	May	1922
2	1.485	Elizabeth Stine	female	USA	Leonia	1923-05-26	26	May	1923
3	1.485	Sophie Elliott	female	GBR	Brentwood	1923-08-06	6	August	1923
4	1.524	Phyllis Green	female	GBR	London	1925-07-11	11	July	1925
5	1.552	Phyllis Green	female	GBR	London	1926-08-02	2	August	1926

Your clean mens data frame should look like this one (6 first rows displayed)

	height	athlete	gender	country	city	date	day	month	year
1	2.000	George Horine	male	USA	Palo Alto	1912-05-18	18	May	1912
2	2.022	Edward Beeson	male	USA	Berkeley	1914-05-02	2	May	1914
3	2.038	Harold Osborn	male	USA	Urbana	1924-05-27	27	May	1924
4	2.040	Walter Marty	male	USA	Fresno	1933-05-13	13	May	1933
5	2.060	Walter Marty	male	USA	Palo Alto	1934-04-28	28	April	1934

Your merged `records data frame` should have this structure:

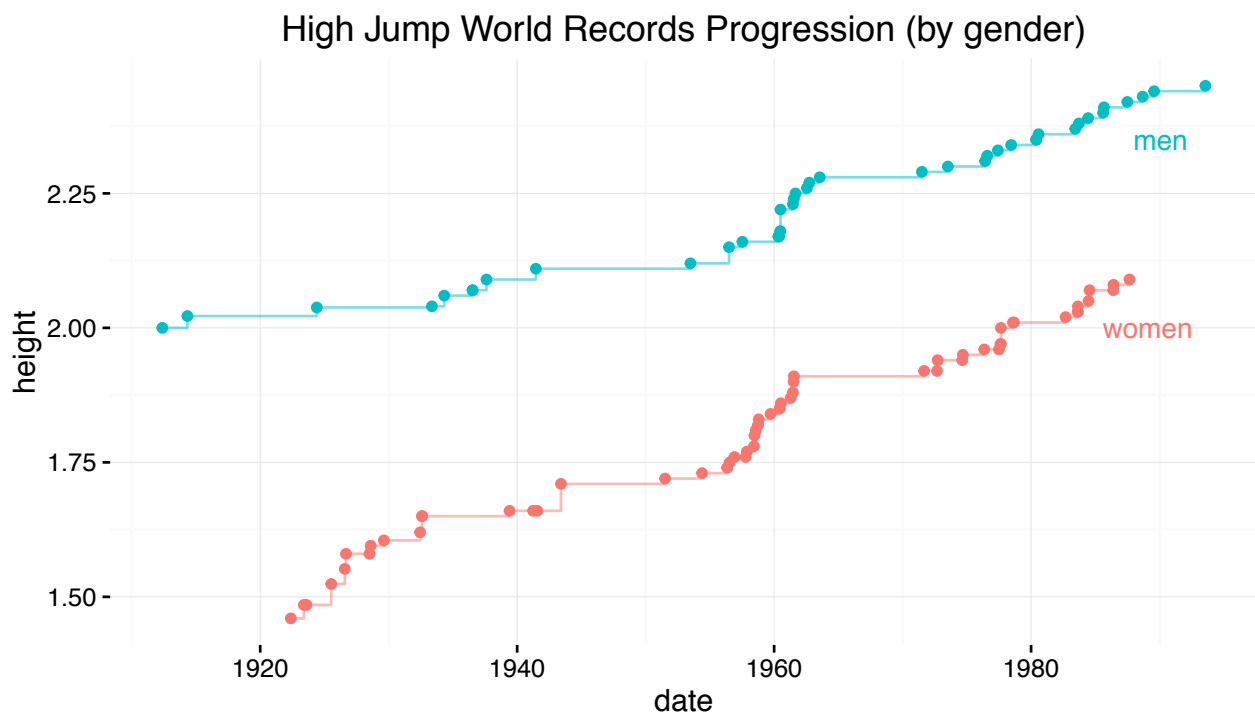
```
'data.frame': 96 obs. of 9 variables:
 $ height : num 1.46 ...
 $ athlete: chr "Nancy Voorhees" ...
 $ gender : Factor w/ 2 levels "female","male": 1 1 ...
 $ country: chr "USA" ...
 $ city : chr "Simsbury" ...
```

```
$ date : Date, format: "1922-05-20" ...  
$ day  : num 20 26 ...  
$ month: chr "May" ...  
$ year : num 1922 ...
```

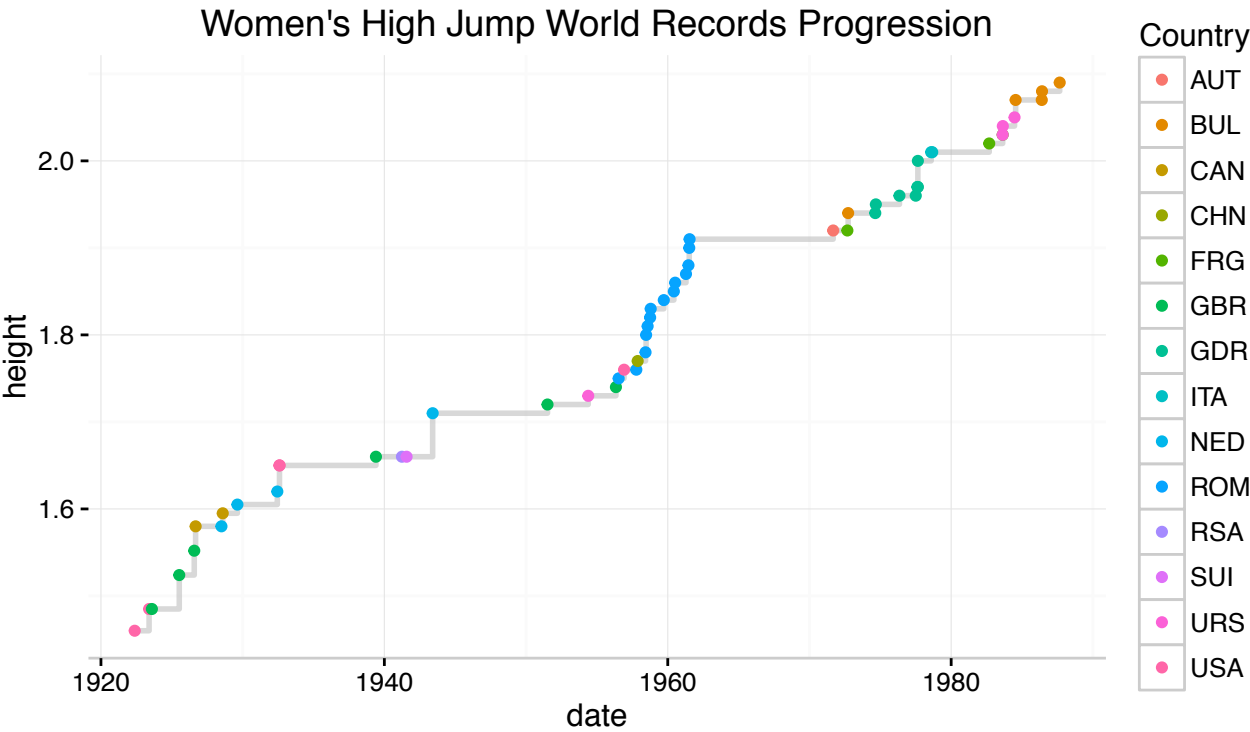
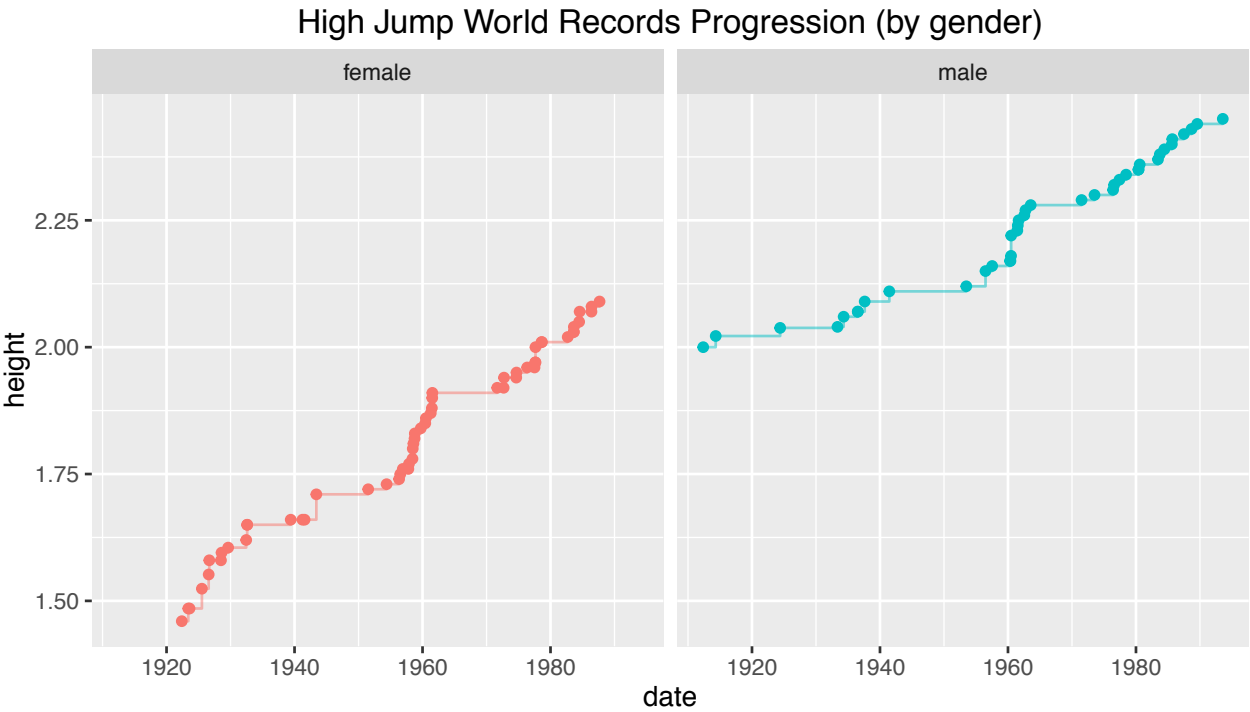
Data Visualization

Your `plots` should look like the charts shown below one.

High Jump World Records Progression (by gender) version 1



High Jump World Records Progression (by gender) version 2



Men's High Jump World Records Progression

