This exam is closed book, closed notes, and closed R. However, you may use your “Cheat sheets” for ggplots2, dplyr, tidyr, readr, stringr, forcats, lubridate, and Base R as well as the “Learn SQL” Basics sheet. Be sure to clearly label answers and turn in all supporting work (to maximize your chance of partial credit).

Note that while this exam is cumulative (from day 1 of the course), the emphasis will be on the material since Exam 1. (However, recall that dplyr and ggplots have been used in just about every lesson we’ve now seen.)

What will definitely not be on the exam?

* The “intro stat inference” material (e.g., t-tests)
* Data Ethics (as that is in Project 1 only)
* Shiny
* Nested JSON files (although simple, non-nested, JSON files are fair game)

1. I recommend going back over exercises and trying them “by hand’ wherever possible.
2. Suppose you have imported a dataset into an object called **grades** with the following variables. You may assume that you have already loaded the appropriate packages for this quiz.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| quiz | The cumulative quiz scores for the student (scale 0 – 100) |
| exam | The cumulative exams scores for the student (scale 0 – 100) |

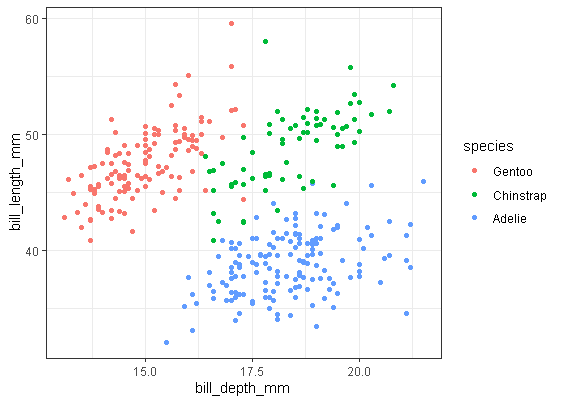
Write the appropriate dplyr code to calculate the average quiz score and the median exam score for this class. Piping is required. You may assume that there are no missing values in this data.

1. Suppose that you have a dataset called ***penguins*** in your environment with the structure displayed below.

str(penguins)

## tibble [344 x 3] (S3: tbl\_df/tbl/data.frame)  
## $ species : chr [1:344] "Chinstrap" "Gentoo" "Adelie" "Chinstrap" ...  
## $ bill\_length\_mm: num [1:344] 49 45.8 39 43.2 48.8 49.1 40.9 37.3 46.5 50.7 ...  
## $ bill\_depth\_mm : num [1:344] 19.6 14.6 17.1 16.6 16.2 14.8 16.6 17.8 17.9 19.7

Provide the commands to produce the following plot. You may assume that you have loaded any required packages. Further, for full credit you must use at least one function from the forcats package. Note: The colors for the three species are just the default palette, when the *species* variable is handled appropriately. (i.e., there is no need to include the scale\_\*\_manual in your code)



1. Suppose you have a data table of the following form, stored as a object called **country\_rates**. It is important to note that this is just a snippet of a much larger dataset. (i.e., the full dataset has more years and more countries in it within the “…” parts)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | Algeria | Brazil | Columbia |  | Zimbabwe |
| 2000 | 7% | 12% | 16% |  | 8% |
| 2001 | 9% | 14% | 18% |  | 8% |
|  |  |  |  |  |  |
| 2021 | 8% | 15% | 19% |  | 6% |

Again, the “…” is not data itself, just an indication that there are more rows and columns of a similar structure between.

1. Provide the R code to reshape and tidy the above such that you have three variables: the ***year***, ***country***, and ***rate***. Country should be stored as a character variable, year and rate should be numeric.
2. Provide the R code to reshape it into the following. (Tip: Use you resulting data frame from part a as the starting point.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Country | 2000 | 2001 |  | 2021 |
| Algeria | 7 | 9 |  | 8 |
| Brazil | 12 | 14 |  | 15 |
| Columbia | 16 | 18 |  | 19 |
|  |  |  |  |  |
| Zimbabwe | 8 | 8 |  | 6 |

1. Suppose you have an R environment with the data frames **cats** and **dogs**. A printout of each is below. Note the first row in each are simply the variable names.

**cats**

person Cat

Jessica Calypso

Ivan Appa

Mara Lil Brother

**dogs**

human Dog

Grampy Edgar

Jessica Gordy

Nana Buster

Mara Lady

Assuming you have already loaded the **dplyr** package, use the appropriate **join** command to create a data frame that combines **cats** and **dogs** such that it looks like

1. this:

human Dog Cat

Jessica Gordy Calypso

Mara Lady Lil Brother

1. this:

person Cat Dog

Jessica Calypso Gordy

Ivan Appa <NA>

Mara Lil Brother Lady

Grampy <NA> Edgar

Nana <NA> Buster

1. You may assume that you have already loaded all necessary packages.

Suppose you have the run the following line of code

**quizStr <- c("I","would","be","interested","in","a","text","mining","class")**

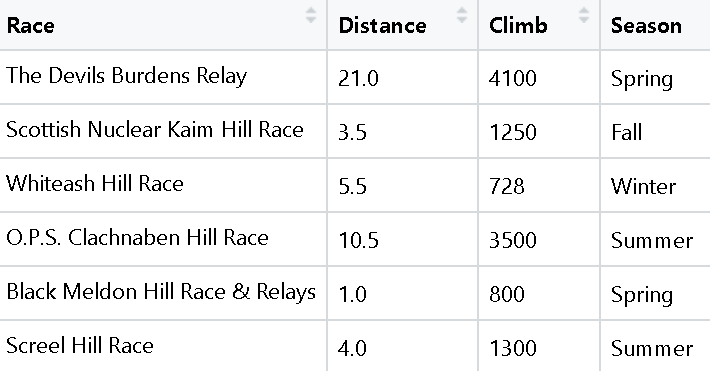
Provide the appropriate code to determine which elements in the vector contain either an I (either upper or lower case).

You must use the function(s) from the stringr package.

1. SLU keeps track of diversity of faculty through time and makes this data public on the following website: <https://www.stlawu.edu/ir/diversity/faculty> Use `rvest` to scrape the data tables into R. The two tables are stored within the first two objects in the list of tables you would get from rvest. Store each of these in separate objects.
2. Hill running – races up and down hills – has a written history in Scotland dating back to the year 1040. Races are held throughout the year at different locations around Scotland. A recent compilation of information for 71 races includes the distance (miles), the total amount of climb (ft), the season the race was in (recorded as Fall, Spring, Summer, and Winter) and the record time (seconds) for males and females. This information in stored in three separate data frames in your R project. Screenshots of the first 6 races in each data frame are provided below (and on the next page). You may assume that each of the 71 races shows up exactly once in each data frame (but are not guaranteed to be in the same order in each). **Note:** These data are available in the Exam 2: Additional Study Materials module on canvas (and on the GitHub site for the class). Feel free to test out your code you’ve written in R on these data.
3. Create a new data frame (called hill\_running) that contains the race information and both the male and female record times.
4. Calculate the number of races that have the word hill in their name. (Be wary of capitalization differences.)
5. Calculate the number of races that have the substring hill in their name. (This should result in more races than the previous part)
6. Calculate the number of races that have either *race*, *trot*, or *dash* in their name.
7. Compare the average climb for races with and without the word uphill in their name.
8. Calculate the mean record times (for both males and females) comparing races with the term race versus those without it in their name.
9. Convert all race names to lower case.
10. Replace any instances of *race*, *trot*, or *dash* with *gallop*.
11. Remove all punctuation from the race names.
12. Produce a histogram of the difference in record times (male – female) and write a sentence or two summarizing the distribution.
13. Create a “Seasons” variable with Winter being Jan – March, Spring being April & May, Summer being June – Sept, and Fall being Oct – Dec.
14. Make side-by-side boxplots of the record time vs season with facets for each gender. Write a few sentences comparing and contrasting the distributions. (Tip: For this part, you might need to rethink the shape of your data frame from what you created in part a.)

|  |  |
| --- | --- |
| Male Record’s dataset (called **males** in your R project) | Female Record’s dataset (called **females** in your R project) |
|  |  |

Race Information dataset (called **race\_info** in your R project)



1. The object ***mcu*** contains data on movies from the Marvel Comic Universe. Each row represents one of the movies in the mega-series and contains the following variables.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Example Entry of a Single Row** |
| Name | Name of the film | Avengers: Infinity War |
| US\_release\_date | Release date in the United States | 27-Apr-18 |
| Directors | The director(s) of the movie | Anthony Russo, Joe Russo |
| Producers | The producer(s) of the movie | Kevin Feige |
| Duration | The duration of the movie | 2h 29min |
| Genre | The main genre(s) associated with the movie | Action , Adventure, Sci-Fi |
| IMDB\_rating | The IMDb score | 8.4 |
| Metascore | The Metacritic score | 68 |
| Main\_cast | The main cast members for the movie | Robert Downey Jr. , Chris Evans , Chris Hemsworth , Mark Ruffalo , Josh Brolin  **see note below** |
| Total\_gross | World-wide gross revenue in millions of USD | 2048 |
| Oscar\_nominations | Number of Oscar nominations | 1 |

**A note on the *Main\_cast* variable:** This variable contains anywhere from 3 to 5 actors/actresses for each movie. While many members of the cast were in multiple MCU movies, no single movie has them all listed in the Main\_cast variable.

1. What if Captain America (played by the actor, Chris Evans) was actually played by Hayley Atwell (the actress best known as Peggy Carter in the MC Universe)? Provide the appropriate tidyverse code to replace all references to Chris Evans with Hayley Atwell.
2. Calculate the number of movies released in each of the months of the year. Your code should be structured in such a way that the resulting output is a single tibble (data frame) with 12 rows (i.e., one for each month) and 2 columns (one for the month’s *name* and one for the *number of movies released*)
3. For each main cast member, determine the average Metacritic score for movies in which they appeared (as a main member of the cast). Recall that the *main\_cast* variable, as written above, contains between 3 – 5 people and not just the five seen in the example above.

For each of the following, provide a brief answer (typically a single word). Note, that when asking for a package, it is not appropriate to say “tidyverse” – I want to know the specific package you would use

1. Which R package did we load to scrape data from the web?
2. Which R package did we use to import CSV files?
3. Which function is used to stack data tables on top of each other (assuming they have the same column names)?
4. Which function can be used to make all characters in a string UPPERCASE?
5. Which function can be used to split a column into two (or more) different columns?
6. ~~I want to learn about what starting salaries listed in entry level jobs for statisticians/data scientists. Suppose that we have a data set of 100 job ads stored in the object~~ *~~statjobs.~~*  ~~This data contains a variable called~~ *~~starting\_salary~~* ~~which contains the starting salary of the position – if the company provided it in the ad (and is blank otherwise). The goal is to use this data to construct a 95% bootstrap confidence interval, based on 1000 bootstrap samples, for the median starting salary of entry level “statistician/data scientist” jobs. I have provided most of the code below. There are three places where you need to provide the appropriate line of code (highlighted and marked by a # in the code chunk). Provide your answers in the space at the bottom of the page.~~

~~#a) Provide the line of code to clean statjobs object~~

~~B <- 1000 # number of bootstrap samples~~

~~all\_median\_boots <- rep(NA, B) # a blank vector to store the bootstrap statistics~~

~~#b) Provide the line of code missing from this line~~

~~# 1. Resample Data~~

~~resampled\_data <- slice\_sample(statjobs\_clean, prop = 1, replace = TRUE)~~

~~#2. Calculate statistic~~

~~one\_median\_boot <- median(resampled\_data$starting\_salary)~~

~~#c) Provide the line of code missing from this line~~

~~}~~

~~conf.level <- 0.95~~

~~alpha <- 1 - conf.level~~

~~quantile(all\_median\_boots, probs = c(alpha/2, 1-alpha/2))~~

~~Your answers:~~

~~a)~~

~~b)~~

~~c)~~

1. Suppose you have a file, called **car\_insurance.json** is a JSON file with the following structure.

[{"name":"John","age":30,"isMarried":true},

{"name":"Jane","age":25,"isMarried":false},

{"name":"Bob","age":40,"isMarried":true}]

You should assume that there are numerous other rows in the full data, but each has the same three pieces of information (name, age, and marriage status).

Provide the code to read this in to R and convert it into a tibble (or data frame) with three variables.

1. Suppose we have a SQLIte database called **employee\_data.sqlite** with the following tables:

* **employees** with fields: id, name, age, department\_id, title, and hire\_date (stored as a numerical year)
* **departments** with fields: id and name
* **salaries** with fields: employee\_id and salary

Write the SQL query to answer each of the following

* 1. What is the average salary of all employees in the employees database?
  2. What is the total number of employees hired from 01 Jan 2010 to 31 Dec 2019?
  3. What is the highest salary of all employees who have the job title "Manager"?

1. For additional practice, go back over the notes and complete any examples and exercises that we skipped over. (e.g., in the “Working with dates” lesson)