Homework 1

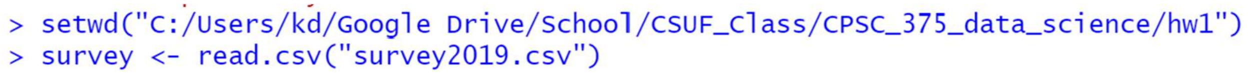
Khoa Do

Due: Thursday 9/27, 11:55pm on Titanium. Prepare your answers as a **single PDF file**.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file. Only one person in the group needs to submit to Titanium.

1. Consider the data collected from the survey conducted during the first class in the attached .csv file. Perform the following steps. Give the corresponding R code (single line), code output, and answer other questions, show plots, if asked. For this question, you may use either base R or the tidyverse library.

a. Load the survey data into a variable called "survey" Hint: use read_csv() (code, output)

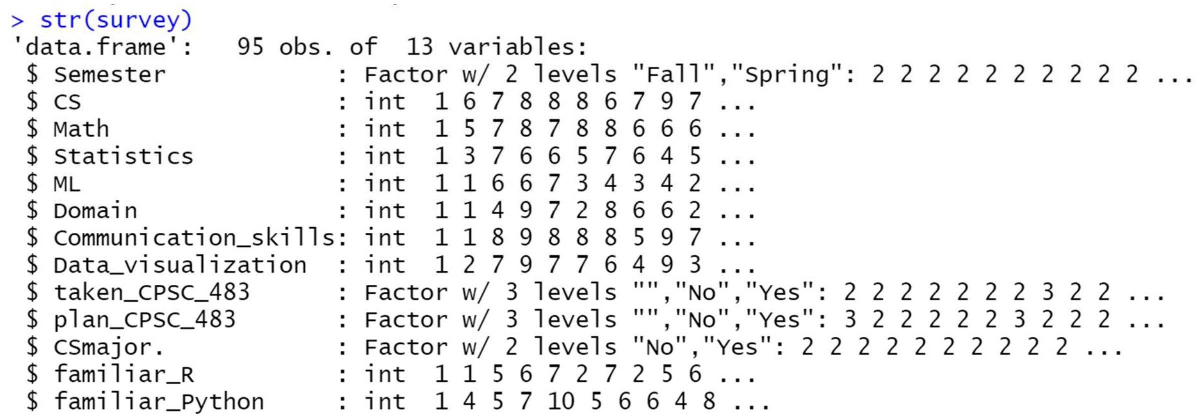
    answer: setwd("C:/Users/kd/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1")
    setwd("C:/Users/kdo1/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1")
    setwd("/Users/mikedo/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1")
    survey <- read.csv("survey2019.csv")



```
> setwd("C:/Users/kd/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1")
> survey <- read.csv("survey2019.csv")
```

b. Which variables are numeric? (code, output)

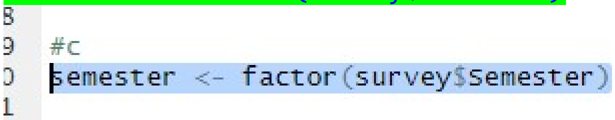
    answer: str(survey)
    CS, Math, Statistics, ML, Domain, Communication_skills, Data_visualization, familiar_R, familiar_Python



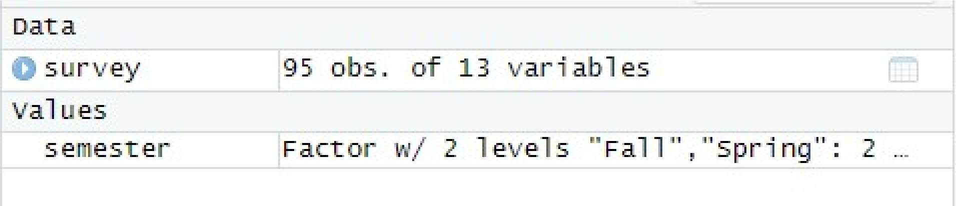
```
> str(survey)
'data.frame':   95 obs. of  13 variables:
 $ Semester            : Factor w/ 2 levels "Fall","Spring": 2 2 2 2 2 2 2 2 2 2 ...
 $ CS                  : int  1 6 7 8 8 8 6 7 9 7 ...
 $ Math                : int  1 5 7 8 7 8 8 6 6 6 ...
 $ Statistics          : int  1 3 7 6 6 5 7 6 4 5 ...
 $ ML                  : int  1 1 6 6 7 3 4 3 4 2 ...
 $ Domain              : int  1 1 4 9 7 2 8 6 6 2 ...
 $ Communication_skills: int  1 1 8 9 8 8 8 5 9 7 ...
 $ Data_visualization  : int  1 2 7 9 7 7 6 4 9 3 ...
 $ taken_CPSC_483      : Factor w/ 3 levels "","No","Yes": 2 2 2 2 2 2 2 3 2 2 ...
 $ plan_CPSC_483       : Factor w/ 3 levels "","No","Yes": 3 2 2 2 2 2 2 3 2 2 2 ...
 $ CSmajor.            : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ familiar_R          : int  1 1 5 6 7 2 7 2 5 6 ...
 $ familiar_Python     : int  1 4 5 7 10 5 6 6 4 8 ...
```

c. Convert variable Semester to a factor. (code)

```
semester <- factor(survey$Semester)
```



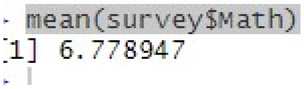
```
#c
semester <- factor(survey$Semester)
```



```
Data
survey        95 obs. of 13 variables
Values
  semester    Factor w/ 2 levels "Fall","Spring": 2 …
```

d. What is the mean value of Math skills? (code, output)

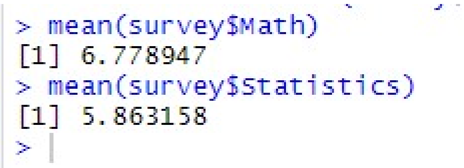
```
mean(survey$Math) = 6.778947
```



```
> mean(survey$Math)
[1] 6.778947
>
```

e. Is the mean skill level in Statistics higher than that in Math? (code, output)

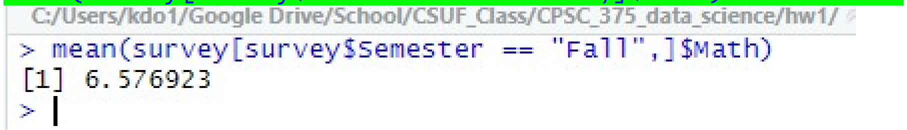
```
answer: #d
mean(survey$Math)

#e
mean(survey$Statistics)
no
```



```
> mean(survey$Math)
[1] 6.778947
> mean(survey$Statistics)
[1] 5.863158
>
```

f. What is the mean value of Math skills in Fall semester? (code, output)

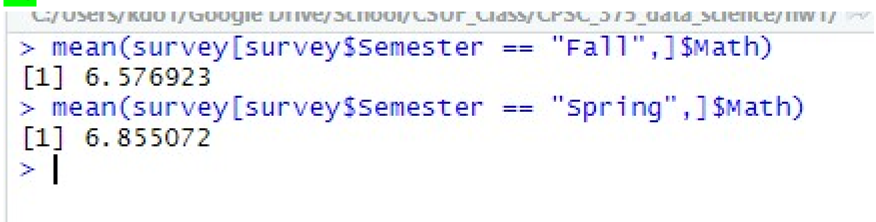
```
mean(survey[survey$Semester == "Fall",]$Math) = 6.576923
```



```
C:/Users/kdo1/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1/
> mean(survey[survey$Semester == "Fall",]$Math)
[1] 6.576923
>
```

g. Is the mean value of Math skills in Fall semester higher than that in Spring? (code, output)

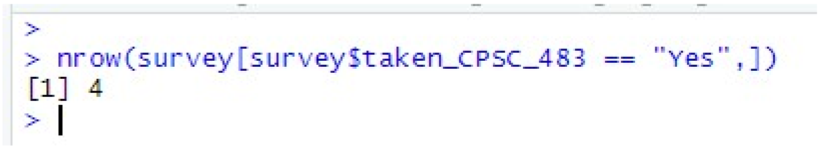
```
answer: #f
mean(survey[survey$Semester == "Fall",]$Math)

#g
mean(survey[survey$Semester == "Spring",]$Math)
no
```



```
C:/Users/kdo1/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1/
> mean(survey[survey$Semester == "Fall",]$Math)
[1] 6.576923
> mean(survey[survey$Semester == "Spring",]$Math)
[1] 6.855072
>
```

h. How many students have taken CPSC 483?

```
nrow(survey[survey$taken_CPSC_483 == "Yes",])
4
```

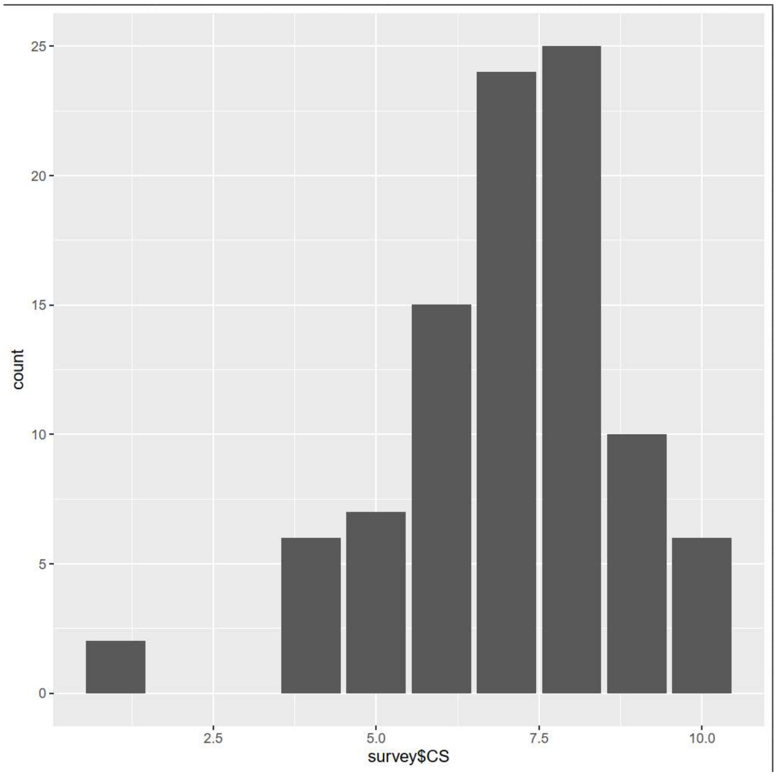


```
>
> nrow(survey[survey$taken_CPSC_483 == "Yes",])
[1] 4
> |
```

2. Using the survey data for Q1, create the following plots using ggplot. Give both code and include the plot as an image. Plots can be saved from RStudio or using R commands.

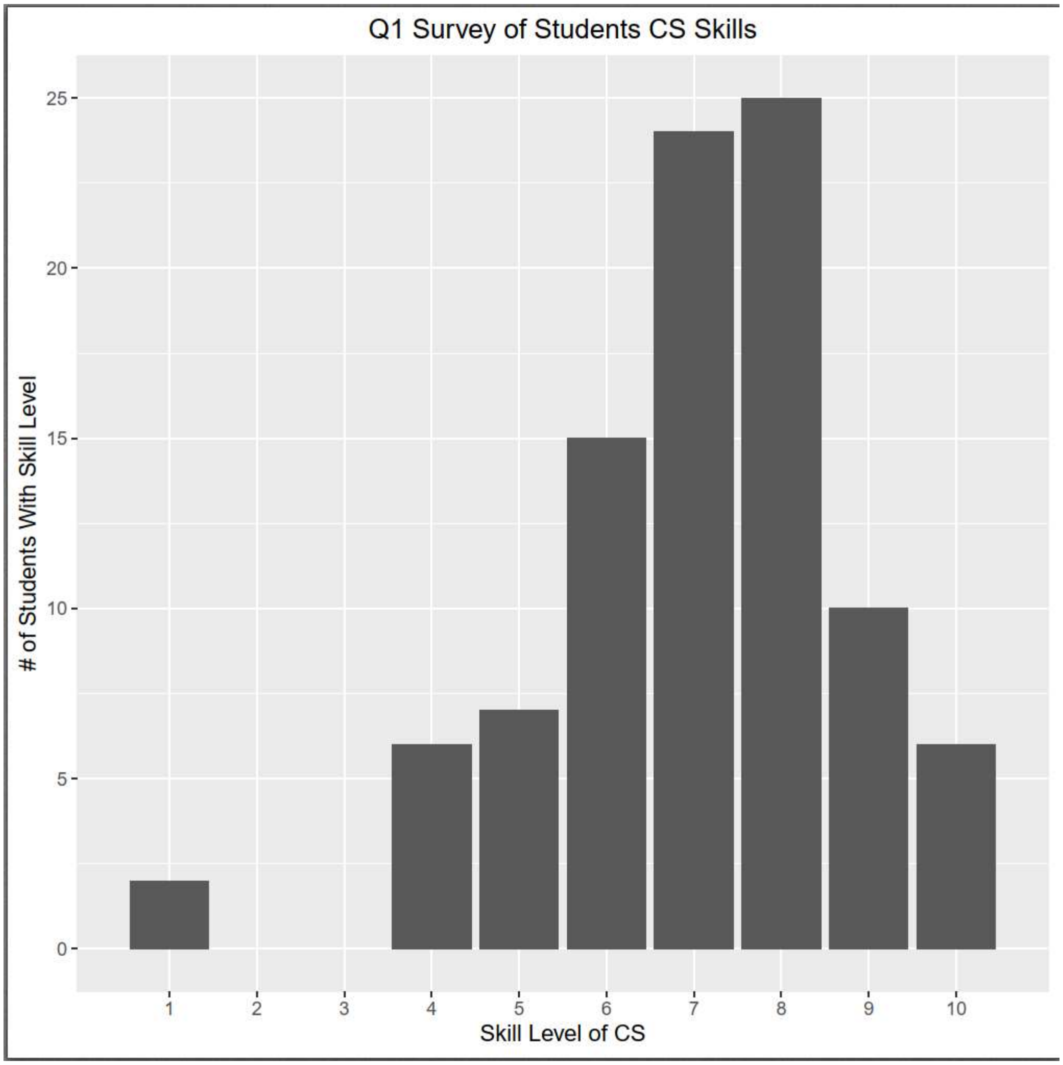
a. A bar graph of variable "CS"

```
answer: p <- ggplot(data = survey)
barGraph =
  p +
  geom_bar(mapping=aes(x=survey$CS))

pdf("question2a_graph.pdf")
print(barGraph)
dev.off()
```



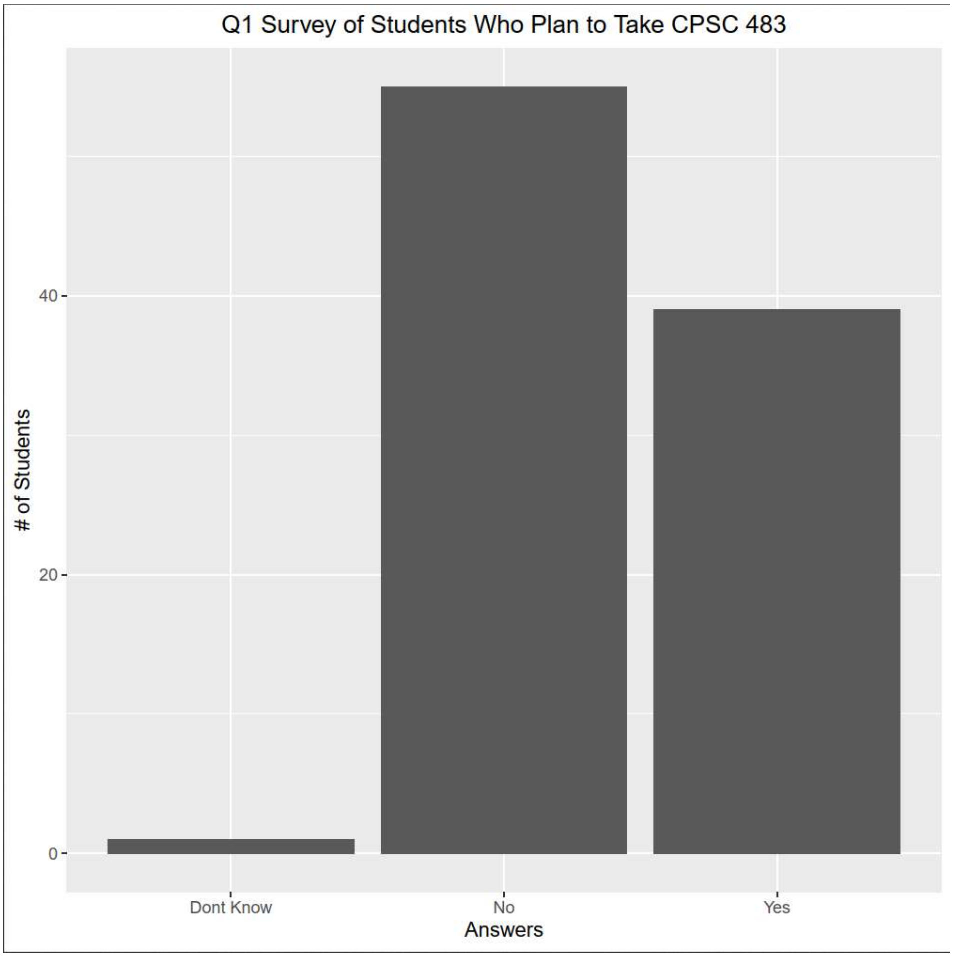
b. The plot above likely has x-axis labels not aligned with the bars. Provide your own breaks to match the variable values/bars. Also, add a plot title. Show both code and paste the plot as an image.

```
answer: #bar graph
p <- ggplot(data = survey)
barGraph =
  p +
  geom_bar(mapping=aes(x=ilist)) +
  ggtitle("Q1 Survey of Students CS Skills") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Skill Level of CS") +
  ylab("# of Students With Skill Level") +
  scale_x_discrete(breaks=breaklist, labels=labellist, limits=breaklist)

pdf("question2b_graph.pdf")
print(barGraph)
dev.off()
```



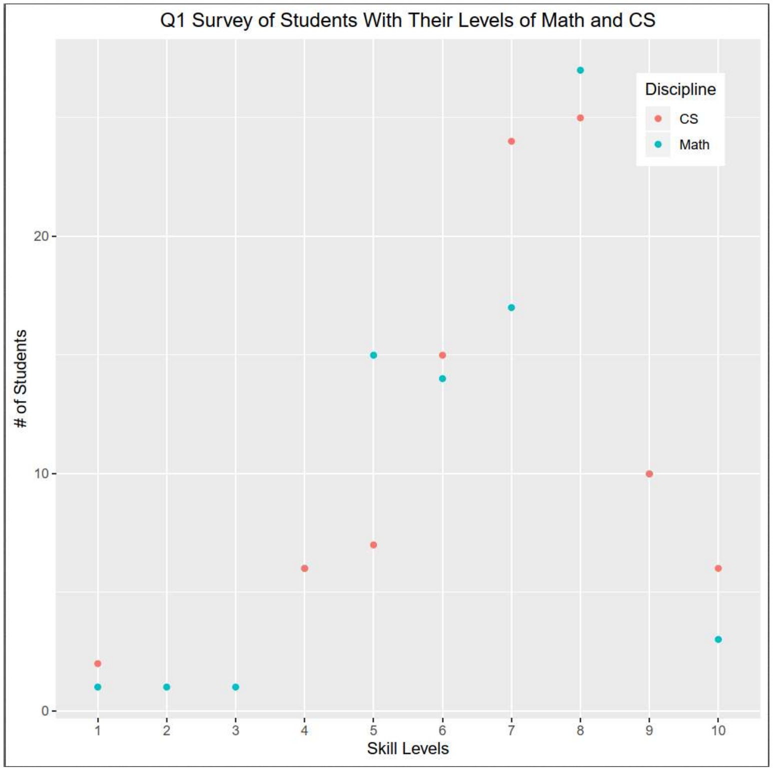
c. Plot (using ggplot) a bar graph of variable “plan_CPSC_483”. Show both code and paste the plot as an image.

```
answer: ilist = survey$plan_CPSC_483
breaklist = levels(ilist)
labellist = c("Dont Know", "No", "Yes")

#bar graph
p <- ggplot(data = survey)
barGraph =
  p +
  geom_bar(mapping=aes(x=ilist)) +
  ggtitle("Q1 Survey of Students Who Plan to Take CPSC 483") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Answers") +
  ylab("# of Students") +
  scale_x_discrete(breaks=breaklist, labels=labellist, limits=breaklist)

pdf("question2c_graph.pdf")
print(barGraph)
dev.off()
```



d. A scatterplot of variables “Math” and “CS”

```
seq <- 1:10
mathList = survey %>% group_by(Math) %>% summarise(Count=n()) %>% mutate(Item=Math,
Discipline="Math") %>% select(Item, Count, Discipline) %>% arrange(Item)
#mathList <- mathList %>% complete(Item = seq, fill = list(Count = 0, Discipline="Math"))
csList = survey %>% group_by(CS) %>% summarise(Count=n()) %>% mutate(Item=CS, Discipline="CS") %>%
select(Item, Count, Discipline) %>% arrange(Item)
#csList <- csList %>% complete(Item = seq, fill = list(Count = 0, Discipline="CS"))

#totalList <- merge(mathList, csList, by="Item")
totalList <- bind_rows(mathList, csList)

#scatter plot
p <- ggplot(data = totalList)
graph =
  p +
  geom_point(mapping=aes(x=totalList$Item, y=totalList$Count, colour=totalList$Discipline)) +
  ggtitle("Q1 Survey of Students With Their Levels of Math and CS") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Skill Levels") +
  ylab("# of Students") +
  scale_x_discrete(breaks=totalList$Item, labels=as.character(totalList$Item), limits=totalList$Item) +
  theme(legend.position = c(0.95, 0.95), legend.justification = c("right", "top")) +
  scale_color_discrete(name = "Discipline")
  #guides(color=guide_legend(title="New Legend Title Guides"))
  #labs(color="NEW LEGEND TITLE labs")

pdf("question2d_graph.pdf")
print(graph)
dev.off()
```



3. Data wrangling using the tidyverse

Data wrangling cheatsheet: http://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

Apply the tidyverse's data wrangling verbs to answer these questions about the survey data used in Q1. For each question, **give only the (one line of) code**.

a. List data only for students with skill in CS > 7

   ```
   survey %>% filter(CS > 7)
   ```

b. List data only for students with skill in CS > 7 and skill in Math > 5

   ```
   survey %>% filter(CS > 7, Math > 5)
   ```

c. List data only for students with either skill in CS > 7 or skill in Math > 5

   ```
   survey %>% filter(CS > 7 | Math > 5)
   ```

d. List data only for students who are CS majors.

   ```
   survey %>% filter(CSmajor. == "Yes")
   ```

e. Sort data in order of increasing Statistics skill

   ```
   survey %>% arrange(Statistics)
   ```

f. Sort data in order of decreasing Statistics skill

   ```
   survey %>% arrange(desc(Statistics))
   ```

g. Show only the Semester and CS major of students in order of decreasing Statistics skill

   ```
   survey %>% arrange(desc(Statistics)) %>% select(Semester, CSmajor.)
   ```

h. Add a new variable, Math_Statistics, that indicates the total Math + Statistics skill

   ```
   survey %>% mutate(Math_Statistics=Math+Statistics)
   ```

i. Show only the Semester and CS major of students in order of decreasing Math_Statistics skill

   ```
   survey %>% mutate(Math_Statistics=Math+Statistics) %>%
   arrange(desc(Math_Statistics)) %>% select(Semester, CSmajor.)
   ```

j. Show only the Semester and CS major of students with 10 highest Statistics skill (Hint: use the min_rank() function which assigns ranks 1, 2, 3, …)

   ```
   survey %>% arrange(desc(Statistics)) %>% head(10) %>% select(Semester, 'CSmajor.')
   ```

k. Show the average Math_Statistics skill for every Semester.

```
survey %>%
  mutate(Math_Statistics=Math+Statistics) %>%
  group_by(Semester) %>%
  summarise(average=mean(Math_Statistics))
```

l. Show the average Math_Statistics skill for every Semester-CS major pair

```
survey %>%
  mutate(Math_Statistics=Math+Statistics) %>%
  group_by(Semester, CSmajor.) %>%
  summarise(average=mean(Math_Statistics))
```

4. Data reshaping using the tidyverse

a. Consider the attached .csv file “horse_racing.csv” which contains data related to horse racing licensing in New York[1]. The `License` column has two types of values: license numbers and receipt numbers. Load the dataset and transform it such that this column is split into two:

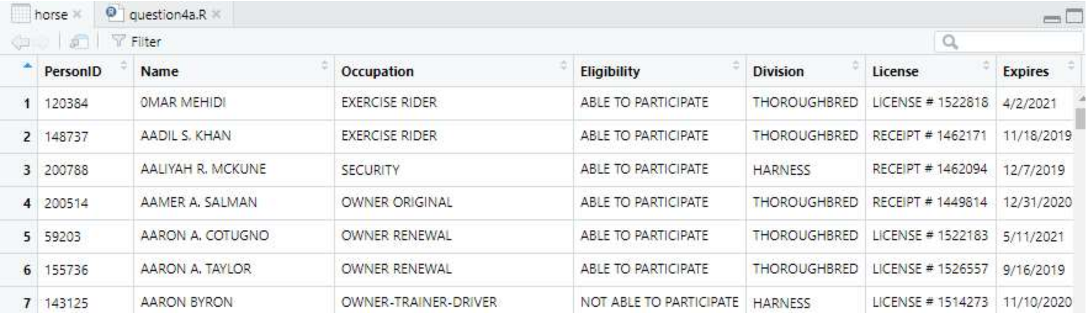
   i. `LicenseOrReceipt`: a factor with two levels “License” and “Receipt”

   ii. `Number`: numeric column with the license/receipt number

   Show (1) your code, and (2) copy & paste the output of the function **`str()`** on your final table.

1.

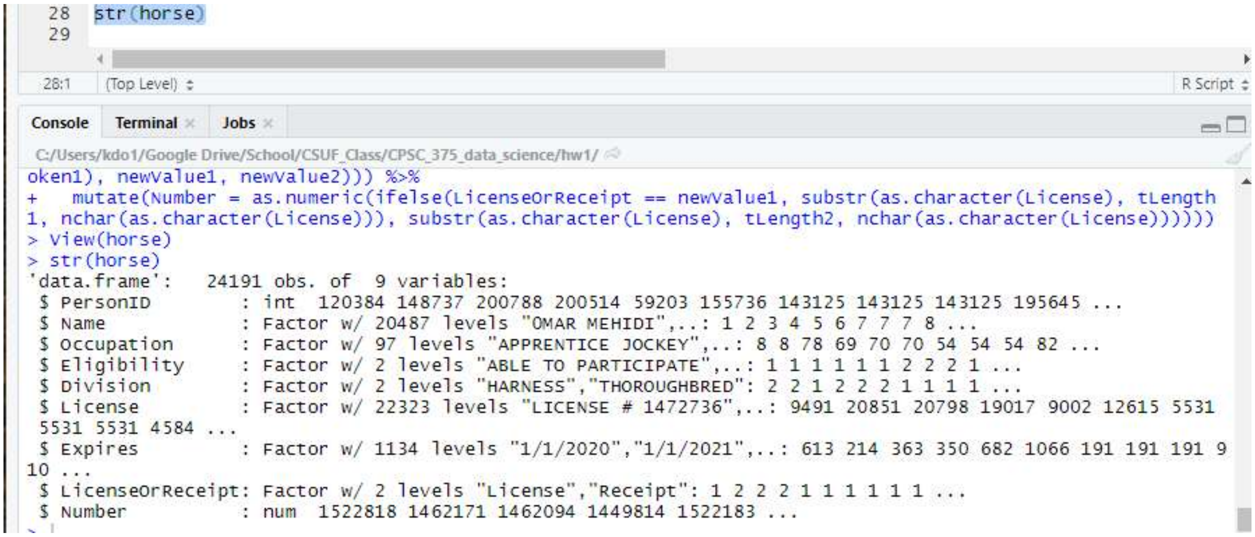
```
token1 <- 'LICENSE # '
tLength1 <- nchar(token1)

token2 <- 'RECEIPT # '
tLength2 <- nchar(token2)

newValue1 = 'License'
newValue2 = 'Receipt'

horse <- data %>%
   mutate(LicenseOrReceipt = factor(ifelse((substr(as.character(License), start = 1, stop
= tLength1)==token1), newValue1, newValue2))) %>%
  mutate(Number = as.numeric(ifelse(LicenseOrReceipt == newValue1,
substr(as.character(License), tLength1, nchar(as.character(License))),
substr(as.character(License), tLength2, nchar(as.character(License))))))
View(horse)
```



	PersonID	Name	Occupation	Eligibility	Division	License	Expires
1	120384	0MAR MEHIDI	EXERCISE RIDER	ABLE TO PARTICIPATE	THOROUGHBRED	LICENSE # 1522818	4/2/2021
2	148737	AADIL S. KHAN	EXERCISE RIDER	ABLE TO PARTICIPATE	THOROUGHBRED	RECEIPT # 1462171	11/18/2019
3	200788	AALIYAH R. MCKUNE	SECURITY	ABLE TO PARTICIPATE	HARNESS	RECEIPT # 1462094	12/7/2019
4	200514	AAMER A. SALMAN	OWNER ORIGINAL	ABLE TO PARTICIPATE	THOROUGHBRED	RECEIPT # 1449814	12/31/2020
5	59203	AARON A. COTUGNO	OWNER RENEWAL	ABLE TO PARTICIPATE	THOROUGHBRED	LICENSE # 1522183	5/11/2021
6	155736	AARON A. TAYLOR	OWNER RENEWAL	ABLE TO PARTICIPATE	THOROUGHBRED	LICENSE # 1526557	9/16/2019
7	143125	AARON BYRON	OWNER-TRAINER-DRIVER	NOT ABLE TO PARTICIPATE	HARNESS	LICENSE # 1514273	11/10/2020

[1] Original dataset: https://data.ny.gov/Government-Finance/Horse-Racing-Licensing/cz9u-yj7m/data

2. str(horse)



```
str(horse)
```

28:1
(Top Level)
R Script
Console
Terminal
Jobs
C:/Users/kdo1/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1/

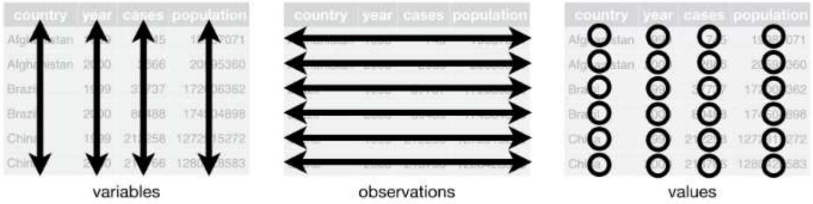
```
oken1), newValue1, newValue2))) %>%
+   mutate(Number = as.numeric(ifelse(LicenseOrReceipt == newValue1, substr(as.character(License), tLength
1, nchar(as.character(License))), substr(as.character(License), tLength2, nchar(as.character(License))))))
> View(horse)
> str(horse)
'data.frame':   24191 obs. of  9 variables:
 $ PersonID        : int  120384 148737 200788 200514 59203 155736 143125 143125 143125 195645 ...
 $ Name            : Factor w/ 20487 levels "OMAR MEHIDI",..: 1 2 3 4 5 6 7 7 7 8 ...
 $ Occupation      : Factor w/ 97 levels "APPRENTICE JOCKEY",..: 8 8 78 69 70 70 54 54 54 82 ...
 $ Eligibility     : Factor w/ 2 levels "ABLE TO PARTICIPATE",..: 1 1 1 1 1 1 2 2 2 1 ...
 $ Division        : Factor w/ 2 levels "HARNESS","THOROUGHBRED": 2 2 1 2 2 2 1 1 1 1 ...
 $ License         : Factor w/ 22323 levels "LICENSE # 1472736",..: 9491 20851 20798 19017 9002 12615 5531
 5531 5531 4584 ...
 $ Expires         : Factor w/ 1134 levels "1/1/2020","1/1/2021",..: 613 214 363 350 682 1066 191 191 191 9
10 ...
 $ LicenseOrReceipt: Factor w/ 2 levels "License","Receipt": 1 2 2 2 1 1 1 1 1 1 ...
 $ Number          : num  1522818 1462171 1462094 1449814 1522183 ...
```

b. Consider the attached .csv file, "language_diversity.csv," which contains data on the diversity of languages in different countries and other parameters[2].

i. Tidy Data is data that is easier to work with in terms of manipulation and analysis. The rules of Tidy Data are the following:

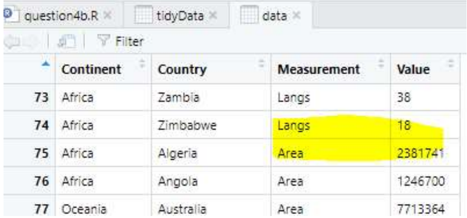
1. Each variable in the data set is placed in its own column
2. Each observation is placed in its own row
3. Each value is placed in its own cell*

ii.

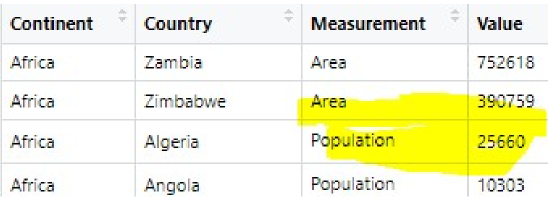


a. Is the data "tidy"? Explain your answer in 2-3 sentences.

No, if you analyze the initial data set, you will notice that the Measurement column contains multiple values that should be placed in its own column thus breaking rule 1. The Value column contains measurement values that are really for different variables thus breaking rule 3.



	Continent	Country	Measurement	Value
73	Africa	Zambia	Langs	38
74	Africa	Zimbabwe	Langs	18
75	Africa	Algeria	Area	2381741
76	Africa	Angola	Area	1246700
77	Oceania	Australia	Area	7713364



Continent	Country	Measurement	Value
Africa	Zambia	Area	752618
Africa	Zimbabwe	Area	390759
Africa	Algeria	Population	25660
Africa	Angola	Population	10303

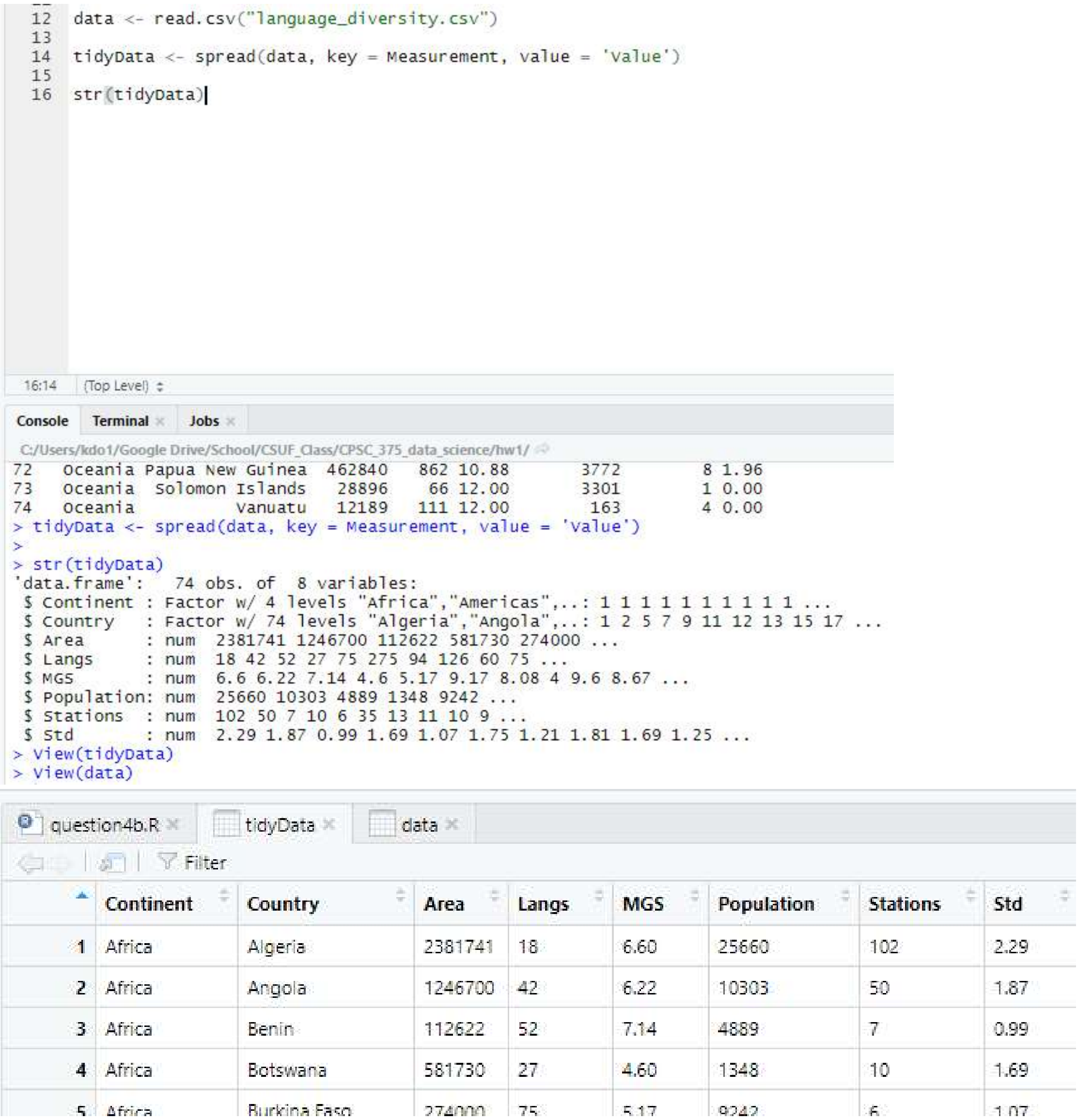
[2] Dataset from: https://github.com/jvcasillas/untidydata#language_diversity

b. Convert the data to tidy data. Show (1) your code, and (2) copy & paste the output of the function **`str`** on your final table.

data <- read.csv("language_diversity.csv")

tidyData <- spread(data, key = Measurement, value = 'Value')

str(tidyData)



```
data <- read.csv("language_diversity.csv")

tidyData <- spread(data, key = Measurement, value = 'Value')

str(tidyData)
```

16:14
(Top Level)
Console
Terminal
Jobs
C:/Users/kdo1/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1/

```
72   Oceania Papua New Guinea  462840   862 10.88       3772        8 1.96
73   Oceania  Solomon Islands   28896    66 12.00       3301        1 0.00
74   Oceania          Vanuatu   12189   111 12.00        163        4 0.00
> tidyData <- spread(data, key = Measurement, value = 'Value')
>
> str(tidyData)
'data.frame':   74 obs. of  8 variables:
 $ Continent : Factor w/ 4 levels "Africa","Americas",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Country   : Factor w/ 74 levels "Algeria","Angola",..: 1 2 5 7 9 11 12 13 15 17 ...
 $ Area      : num  2381741 1246700 112622 581730 274000 ...
 $ Langs     : num  18 42 52 27 75 275 94 126 60 75 ...
 $ MGS       : num  6.6 6.22 7.14 4.6 5.17 9.17 8.08 4 9.6 8.67 ...
 $ Population: num  25660 10303 4889 1348 9242 ...
 $ Stations  : num  102 50 7 10 6 35 13 11 10 9 ...
 $ Std       : num  2.29 1.87 0.99 1.69 1.07 1.75 1.21 1.81 1.69 1.25 ...
> View(tidyData)
> View(data)
```

	Continent	Country	Area	Langs	MGS	Population	Stations	Std
1	Africa	Algeria	2381741	18	6.60	25660	102	2.29
2	Africa	Angola	1246700	42	6.22	10303	50	1.87
3	Africa	Benin	112622	52	7.14	4889	7	0.99
4	Africa	Botswana	581730	27	4.60	1348	10	1.69
5	Africa	Burkina Faso	274000	75	5.17	9242	6	1.07

c. Consider the attached .csv file, “diseases.csv,” which contains data from Australia on hospitalizations[3].

Diseases	Patientdays_Y2015-16	Separations_Y2015-16	Patientdays_Y2016-17	Separations_Y2016-17
1 Certain infectious and parasitic diseases (A00-B99)	694,007	170,095	771,770	186,034
2 Neoplasms (C00-D48)	2,223,563	666,594	2,235,045	684,075
3 Diseases of the blood and blood‑forming organs and certain disorders involving the immune mechanism (D50-D89)	317,085	175,590	335,699	190,568

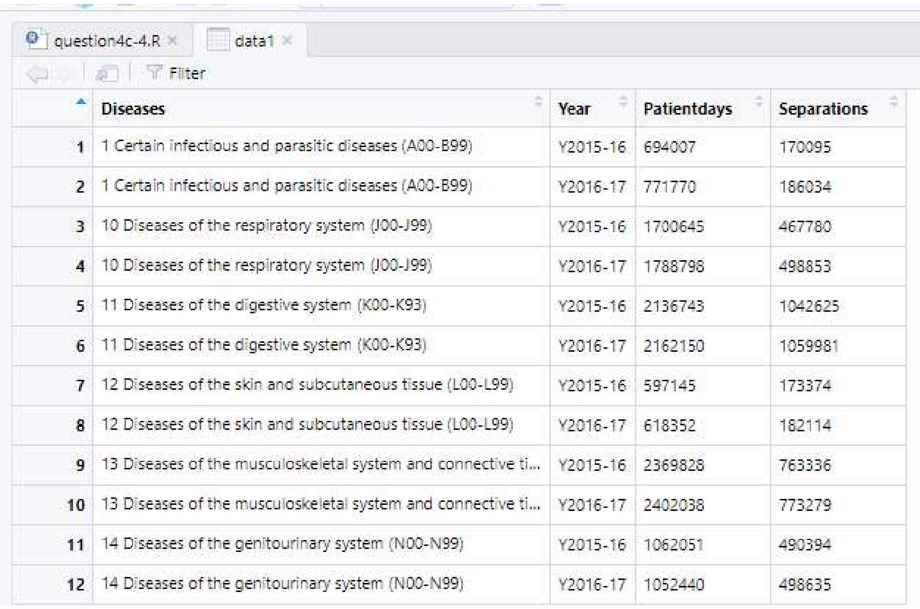
The first few rows are shown above. Load this file and convert the table to the tidy format shown below. Note the new column names. Show (1) your code, and (2) copy & paste the output of the function `str` on your final table. (*Hint: this will require multiple transforms from gather/separate/select. Read the file with* ***read_csv****, not read.csv*)

Diseases	Year	Patientdays	Separations
1 Certain infectious and parasitic diseases (A00-B99)	Y2015-16	694,007	170,095
1 Certain infectious and parasitic diseases (A00-B99)	Y2016-17	771,770	186,034
2 Neoplasms (C00-D48)	Y2015-16	2,223,563	666,594
2 Neoplasms (C00-D48)	Y2016-17	2,235,045	684,075

[3] Dataset from: https://www.aihw.gov.au/reports/hospitals/principal-diagnosis-data-cubes/contents/data-cubes

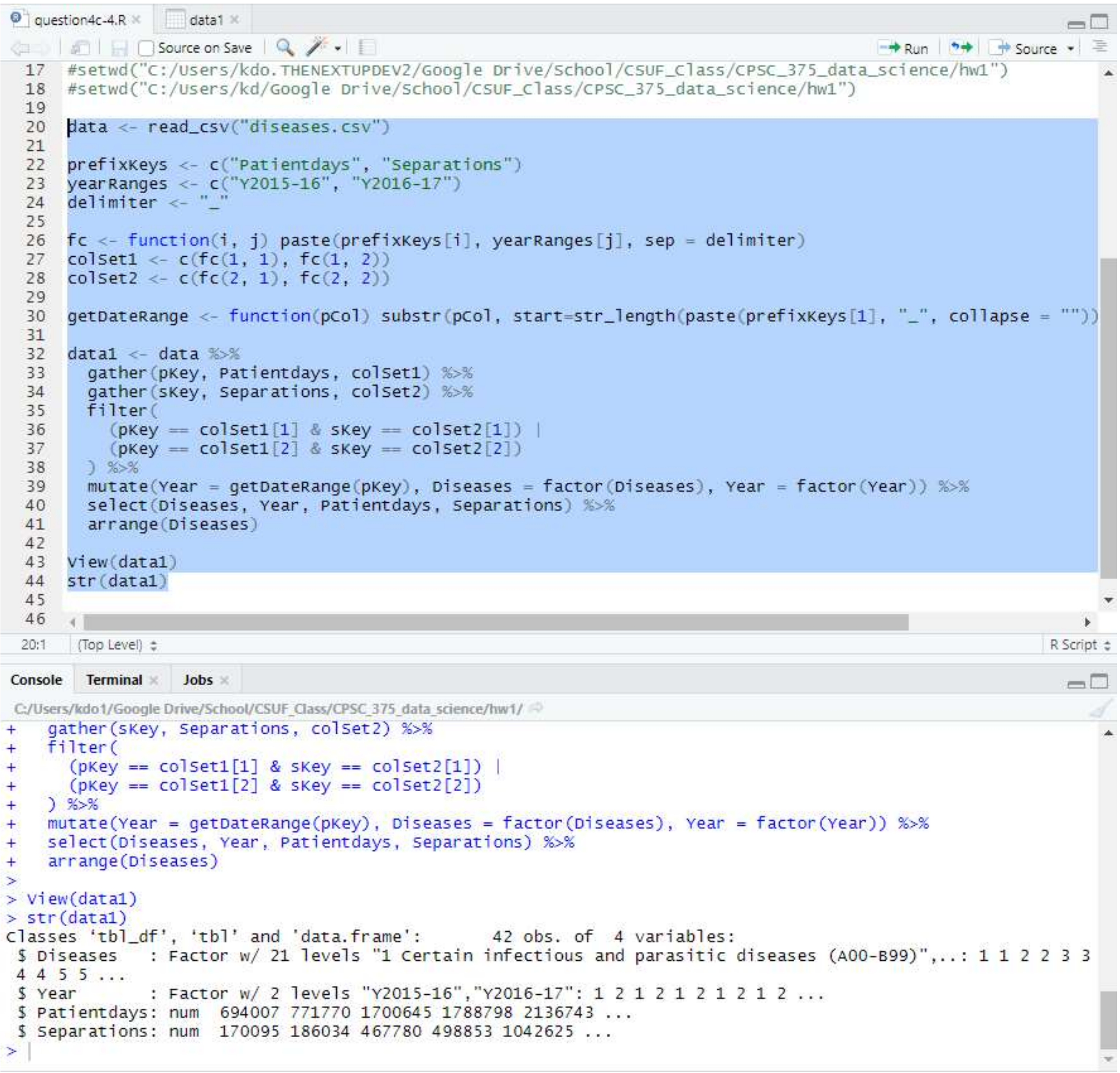
1.

```
data <- read_csv("diseases.csv")

prefixKeys <- c("Patientdays", "Separations")
yearRanges <- c("Y2015-16", "Y2016-17")
delimiter <- "_"

fc <- function(i, j) paste(prefixKeys[i], yearRanges[j], sep = delimiter)
colSet1 <- c(fc(1, 1), fc(1, 2))
colSet2 <- c(fc(2, 1), fc(2, 2))

getDateRange <- function(pCol) substr(pCol, start=str_length(paste(prefixKeys[1], "_", collapse = "")), stop=length(pCol))

data1 <- data %>%
  gather(pKey, Patientdays, colSet1) %>%
  gather(sKey, Separations, colSet2) %>%
  filter(
    (pKey == colSet1[1] & sKey == colSet2[1]) |
    (pKey == colSet1[2] & sKey == colSet2[2])
  ) %>%
  mutate(Year = getDateRange(pKey), Diseases = factor(Diseases), Year = factor(Year)) %>%
  select(Diseases, Year, Patientdays, Separations) %>%
  arrange(Diseases)

View(data1)
str(data1)
```



	Diseases	Year	Patientdays	Separations
1	1 Certain infectious and parasitic diseases (A00-B99)	Y2015-16	694007	170095
2	1 Certain infectious and parasitic diseases (A00-B99)	Y2016-17	771770	186034
3	10 Diseases of the respiratory system (J00-J99)	Y2015-16	1700645	467780
4	10 Diseases of the respiratory system (J00-J99)	Y2016-17	1788798	498853
5	11 Diseases of the digestive system (K00-K93)	Y2015-16	2136743	1042625
6	11 Diseases of the digestive system (K00-K93)	Y2016-17	2162150	1059981
7	12 Diseases of the skin and subcutaneous tissue (L00-L99)	Y2015-16	597145	173374
8	12 Diseases of the skin and subcutaneous tissue (L00-L99)	Y2016-17	618352	182114
9	13 Diseases of the musculoskeletal system and connective ti...	Y2015-16	2369828	763336
10	13 Diseases of the musculoskeletal system and connective ti...	Y2016-17	2402038	773279
11	14 Diseases of the genitourinary system (N00-N99)	Y2015-16	1062051	490394
12	14 Diseases of the genitourinary system (N00-N99)	Y2016-17	1052440	498635

2. str(data1)



```
#setwd("C:/Users/kdo.THENEXTUPDEV2/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1")
#setwd("C:/Users/kd/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1")

data <- read_csv("diseases.csv")

prefixKeys <- c("Patientdays", "Separations")
yearRanges <- c("Y2015-16", "Y2016-17")
delimiter <- "_"

fc <- function(i, j) paste(prefixKeys[i], yearRanges[j], sep = delimiter)
colSet1 <- c(fc(1, 1), fc(1, 2))
colSet2 <- c(fc(2, 1), fc(2, 2))

getDateRange <- function(pCol) substr(pCol, start=str_length(paste(prefixKeys[1], "_", collapse = ""))

data1 <- data %>%
  gather(pKey, Patientdays, colSet1) %>%
  gather(sKey, Separations, colSet2) %>%
  filter(
    (pKey == colSet1[1] & sKey == colSet2[1]) |
    (pKey == colSet1[2] & sKey == colSet2[2])
  ) %>%
  mutate(Year = getDateRange(pKey), Diseases = factor(Diseases), Year = factor(Year)) %>%
  select(Diseases, Year, Patientdays, Separations) %>%
  arrange(Diseases)

View(data1)
str(data1)
```

C:/Users/kdo1/Google Drive/School/CSUF_Class/CPSC_375_data_science/hw1/

```
+   gather(sKey, Separations, colSet2) %>%
+   filter(
+     (pKey == colSet1[1] & sKey == colSet2[1]) |
+     (pKey == colSet1[2] & sKey == colSet2[2])
+   ) %>%
+   mutate(Year = getDateRange(pKey), Diseases = factor(Diseases), Year = factor(Year)) %>%
+   select(Diseases, Year, Patientdays, Separations) %>%
+   arrange(Diseases)
>
> View(data1)
> str(data1)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':	42 obs. of  4 variables:
 $ Diseases   : Factor w/ 21 levels "1 Certain infectious and parasitic diseases (A00-B99)",..: 1 1 2 2 3 3
 4 4 5 5 ...
 $ Year       : Factor w/ 2 levels "Y2015-16","Y2016-17": 1 2 1 2 1 2 1 2 1 2 ...
 $ Patientdays: num  694007 771770 1700645 1788798 2136743 ...
 $ Separations: num  170095 186034 467780 498853 1042625 ...
>
```

5. Consider this answer posted to Quora.com to “Why is R great for Data Science?” (see attached PDF).

What are the 5 parts of the R ecosystem?

Answer:

(1) RStudio, an interactive development environment.

(2) the R “base” language itself.

(3) The tidyverse, a set of packages to develop on top of, and inspired by base R, a more consistent set of functions to wrangle data frames.

(4) The set of packages, spanning all areas of computation, statistics, and algorithms.

(5) The community, which is constantly listening to its users, fixing bugs, posting tutorials and snippets on how to do all kinds of things.

In your opinion, which of the 5 parts is most important for data science?
Justify your opinion in 2-3 sentences.

Answer:

In my opinion, the most important for data science must be the set of packages that are available for R that span to computation, statistics, and algorithms. These packages represent hard work that allow for complicated, well tested features that to be used by newbies like myself. It’s these packages that set R apart from other platform for data science.