Homework 1 Solution

```
library(tidyverse)
```

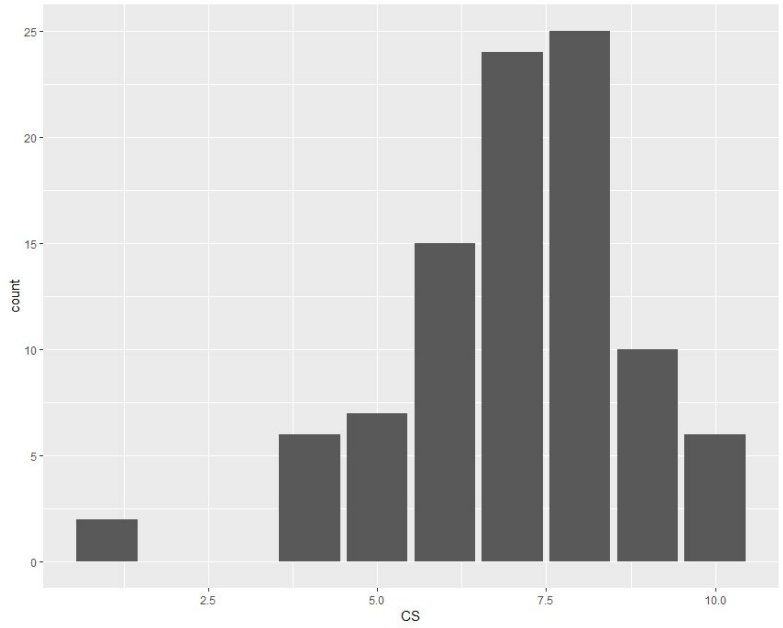
Q1 (8 points)

```
#a.
survey <- read_csv("survey2019.csv")

#b.
str(survey)
# CS, Math, Statistics, ML, Domain, Communication_skills, Data_visualization, familiar_R, familiar_Python
#
# Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    95 obs. of  13 variables:
#   $ Semester          : chr  "Spring" "Spring" "Spring" "Spring" ...
# $ CS                : num  1 6 7 8 8 8 6 7 9 7 ...
# $ Math              : num  1 5 7 8 7 8 8 6 6 6 ...
# $ Statistics        : num  1 3 7 6 6 5 7 6 4 5 ...
# $ ML                : num  1 1 6 6 7 3 4 3 4 2 ...
# $ Domain            : num  1 1 4 9 7 2 8 6 6 2 ...
# $ Communication_skills: num  1 1 8 9 8 8 8 5 9 7 ...
# $ Data_visualization  : num  1 2 7 9 7 7 6 4 9 3 ...
# $ taken_CPSC_483          : chr  "No" "No" "No" "No" ...
# $ plan_CPSC_483           : chr  "Yes" "No" "No" "No" ...
# $ CSmajor?          : chr  "Yes" "Yes" "Yes" "Yes" ...
# $ familiar_R        : num  1 1 5 6 7 2 7 2 5 6 ...
# $ familiar_Python   : num  1 4 5 7 10 5 6 6 4 8 ...

# CS, Math, Statistics, ML, Domain, Communication_skills, Data_visualization, familiar_R, familiar_Python are numeric

#c.
survey$Semester <- factor(survey$Semester)

#d.
mean(survey$Math)
6.778947

#e.
> mean(survey$Statistics)
5.86 #No.
```

```
# OR
> mean(survey$Statistics) > mean(survey$Math)
[1] FALSE

#f.
> survey %>% filter(Semester=="Fall") %>% summarise(mean(Math))
# OR
> mean(survey$Math[survey$Semester=="Fall"])
6.58

#g.
> survey %>% filter(Semester=="Spring") %>% summarise(mean(Math))
6.86
No
#OR
> survey %>% group_by(Semester) %>% summarise(mean(Math))
# Semester `mean(Math)`
# <fct>          <dbl>
# 1 Fall         6.58
# 2 Spring       6.86

#h.
> survey %>% filter(taken_CPSC_483 =="Yes") %>% summarise(n())
# A tibble: 1 x 1
`n()`
<int>
  1       4
#OR
> nrow(survey[survey$taken_CPSC_483 =="Yes" & !is.na(survey$taken_CPSC_483),])
[1] 4
#OR
length(which(survey$taken_CPSC_483 =="Yes"))
```

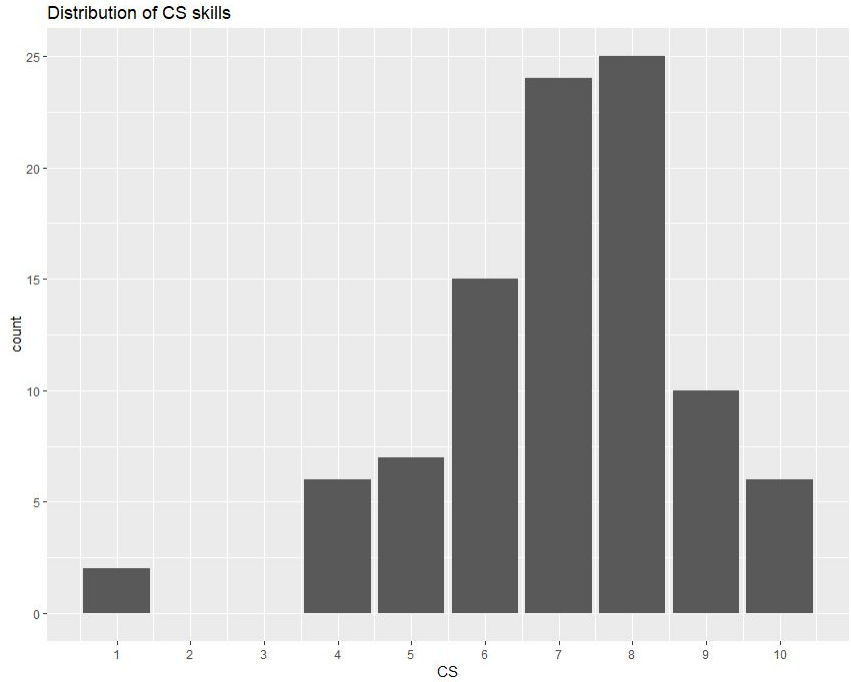
Q2 (8 points)

```
#a.
ggplot(data=survey) + geom_bar(mapping = aes(x=CS))
```



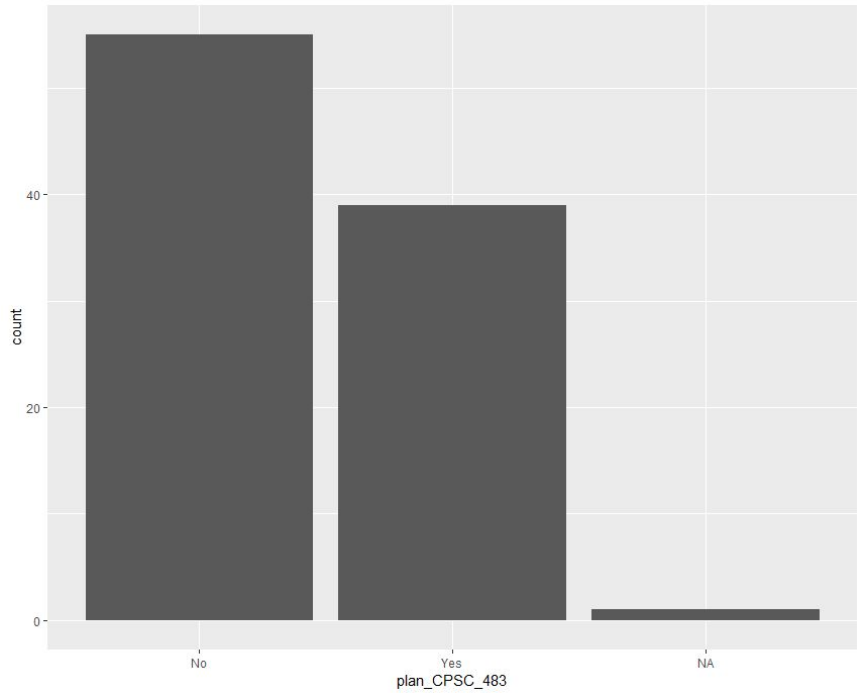
#b.

```
ggplot(data=survey) + geom_bar(mapping = aes(x=CS)) + scale_x_continuous(breaks = 1:10) +
labs(title="Distribution of CS skills")
```



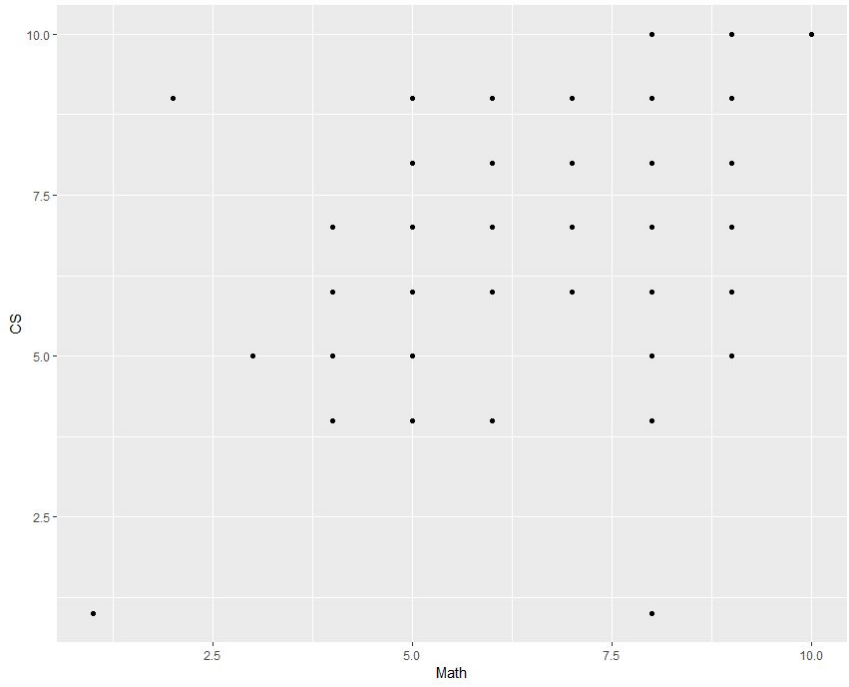
#c.

```
ggplot(data=survey) + geom_bar(mapping = aes(x=plan_CPSC_483))
```



#d.

ggplot(data=survey) + geom_point(mapping = aes(x=Math, y=CS))



3. Data wrangling using the tidyverse (12 points)

One point each. There are multiple solutions for each question. Any solution gets one point.

#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
survey %>% mutate(Rank_Statistics=min_rank(desc(Statistics))) %>%
filter(Rank_Statistics < 10) %>% select(Semester, `CSmajor?`)
#Or in one fewer step combining the filter and min_rank:
survey %>% filter(min_rank(desc(Statistics)) < 10) %>%
select(Semester, `CSmajor?`)
#Or using top_n (wrapper that uses filter() and min_rank())
survey %>% top_n(10, Statistics) %>% select(Semester, `CSmajor?`)
#Or
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(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(mean(Math_Statistics))
```

4. Data reshaping using the tidyverse (9 points)

a. [3 points] 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

```
> horse <- read.csv("horse_racing.csv")
> horsetidy <- horse %>% separate(col=License, into=c("LicenseOrReceipt", "Number"))
> horsetidy$LicenseOrReceipt <- as.factor(horsetidy$LicenseOrReceipt)
> horsetidy$Number <- as.numeric(horsetidy$Number)
> str(horsetidy)
'data.frame':    24191 obs. of  8 variables:
 $ PersonID       : int  120384 148737 200788 200514 59203 155736 143125 143125 143125
195645 ...
 $ Name           : Factor w/ 20487 levels "0MAR 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 ...
 $ LicenseOrReceipt: Factor w/ 2 levels "LICENSE","RECEIPT": 1 2 2 2 1 1 1 1 1 1 ...
 $ Number         : num  1522818 1462171 1462094 1449814 1522183 ...
 $ Expires        : Factor w/ 1134 levels "1/1/2020","1/1/2021",..: 613 214 363 350 682 1066 191
191 191 910 ...
```

b. [3 points] Consider the attached .csv file, "language_diversity.csv," which contains data on the diversity of languages in different countries and other parameters[2].

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

   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.

---

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

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

The data is not tidy. Each observation should contain the values of each type of measurement for each country. However, these measurements are distributed in 6 rows, specified by the value of the Measurement column.

```
> untidy <- read.csv("language_diversity.csv")
> tidy <- untidy %>% spread(key=Measurement, value=Value)
> str(tidy)
'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 ...
```

c. [3 points] 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

```
diseases %>% gather('Patientdays_Y2015-16', 'Patientdays_Y2016-17',
'Separations_Y2015-16', 'Separations_Y2016-17', key="TypeAndYear",
value="Value") %>% separate(TypeAndYear, into= c("Type", "Year"),
sep="_") %>% spread(key=Type, value=Value)
```

OR

```
> Ptable <- diseases %>% gather(`Patientdays_Y2015-16`,
`Patientdays_Y2016-17`, key="PYear", value="Patientdays") %>%
separate("PYear", into=c(NA, "Year"), sep="_")
> Stable <- diseases %>% gather(`Separations_Y2015-16`,
`Separations_Y2016-17`, key="SYear", value="Separations") %>%
separate("SYear", into=c(NA,"Year"), sep = "_")
> PStable <- inner_join(Ptable, Stable)PStable <- inner_join(Ptable,
Stable) %>% select(Diseases, Year, Patientdays, Separations)
```

```
> str(PStable)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':  42 obs. of  4 variables:
 $ Diseases   : chr  "1 Certain infectious and parasitic diseases
(A00-B99)" "2 Neoplasms (C00-D48)" "3 Diseases of the blood and
blood-forming organs and certain disorders involving the immune
mechanism (D50-D89)" "4 Endocrine, nutritional and metabolic
diseases (E00-E89)" ...
 $ Year       : chr  "Y2015-16" "Y2015-16" "Y2015-16" "Y2015-16"
...
 $ Patientdays: num  694007 2223563 317085 582936 3778574 ...
 $ Separations: num  170095 666594 175590 169247 429244 ...
```

```
>
```