Homework 03 - Nonstandard Evaluation and Git

Nonstandard Evaluation

Question 1

Imagine we have a data frame called `data`, with a `type` column. Which one works and why?
Function 1:

```
group_and_tally <- function(df, column){
    df %>% group_by({{ column }}) %>% tally();
}
group_and_tally(data, type);
```

Function 2:

```
group_and_tally <- function(df, column){
    df %>% group_by(column) %>% tally();
}
group_and_tally(data, type);
```

Q1 Answer

Function 1 works because it includes the embrace symbols `{{ }}`. The embrace symbols are needed in tidyverse because otherwise the funciton will try to process `column` as an object instead of looking for the column name `type` (which is what is happening in Funciton 2).

Git

For the questions below, please add the commands you used to complete these steps.

Question 2

Set up your git repo on your local computer. If you already make a git repo on GitHub, but it isn't on your local computer - clone it.

mkdir -p ~/git/BIOS512 cd ~/git/BIOS512 cat <<EOF > README.md #BIOS512 Course This is for the BIOS512 Course, Fall 2025. author: Jama-Brookes EOF git init git add README.md git commit -m "First commit" git branch -M main git remote add origin git@github.com:Jama-Brookes/BIOS512.git git push -u origin main

Question 3

Set up your SSH key.

cd /tmp/ mkdir ssh-keys cd ssh-keys/ ssh-keyge cat ~/.ssh/id_ed25519.pub

Then this SSH key was copied added to Git Hub via Settings > SSH and GPG keys.

Question 4

a) Add a HW2 directory to your git repo through the terminal with a HW.md file that says "This is for homework 2."

In my BIOS512 git repo from my terminal, I added the following code:

mkdir HW2 cd HW2 echo "This is for homework 2." > HW2.md

b) *Add* HW2.md to the staging area. Then, use the command to see which files have been modified, staged for commit, or are untracked. What does it show? They should copy paste the terminal response after git status, and show that key used the commands below.

git add HW2.md git status

Status showed:

```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   HW2.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be
committed)
  (use "git restore <file>..." to discard changes in
working directory)
        modified:
../Homework/BIOS512_HW3_Brookes.ipynb
```

c) Save file changes to the main branch.

git commit -m "Add HW2 folder and HW2.md file" git push -u origin main

d) Now, edit the HW2.md file to give it a title.

cat <<EOF > HW2.md # Homework 2 Example of editing documents in Git in HW3. EOF

e) Use the command that compares current, unsaved changes to the main branch. What does it say?\

Command:

```
git diff
```

Output:

diff --git a/HW2/HW2.md b/HW2/HW2.md index 1a010d3..2eaef26 100644 --- a/HW2/HW2.md +++ b/HW2/HW2.md @@ -1 +1,2 @@ -This is for homework 2. +# Homework 2 +Example of editing documents in Git in HW3.

f) Use the command that checks the status of the working directory and the staging area *again*. What does it say? Command:

```
git status
```

Output:

On branch main Your branch is up to date with 'origin/main'. Changes not staged for commit: (use "git add <file>..." to update what will be committed) (use "git restore <file>..." to discard changes in working directory) modified: HW2.md modified: ../Homework/.ipynb_checkpoints/BIOS512_HW3_Brookes-checkpoint.ipynb modified: ../Homework/BIOS512_HW3_Brookes.ipynb no changes added to commit (use "git add" and/or "git commit -a")

g) Once again, add HW2.md to the staging area and save the file changes to the main branch. Then, get use the command that gives you project history and paste the output in your homework.

Commands:

```
git add HW2.md
```

```
git commit -m "Updated HW2.md with a Title"
```

```
git push
```

```
git log
```

Output:

commit 94454d90ac235a3054c7ee1a92689f12bbf763c0 (HEAD -> main, origin/main) Author: Jama Brookes <brookesjj@Mac.lan> Date: Tue Sep 9 17:11:20 2025 -0400 Updated HW2.md with a Title commit 584b86f809f79f4a97968926035b79dea37eb2d2 Author: Jama Brookes <brookesjj@Mac.lan> Date: Tue Sep 9 16:24:38 2025 -0400 Add HW2 folder and HW2.md file commit 794a793d3187f32d0837fe93055a5d0c14549976 Author: Jama Brookes <brookesjj@Mac.lan> Date: Tue Sep 9 16:08:46 2025 -0400 Add homework folder commit 4115c83e72013300cf8afe0988a419295ddb17de Author: Jama Brookes <brookesjj@Mac.lan> Date: Tue Sep 9 15:14:58 2025 -0400 First commit

h) Do some searching... What `git` command will provide you documentation on other commands? Use that command to find documentation on `git log` and `git show`. What does `--since` mean in regards to `git log`? Copy and

paste what is written in the documentation.

Command:

`git log --help`

`git show --help`

Output:

--since=<date>, --after=<date> Show commits more recent than <date>.

Tidyverse

Note: Please make sure Binder is set up correctly to run this section. You can follow the instructions here: https://github.com/rjenki/BIOS512.

Please show your code for this section! Before completing this section, please run the following.

In [105…
```
library(tidyverse)
if (!dir.exists("intermediate")) dir.create("intermediate", recursive
if (!exists("mdpre")) mdpre <- function(x) { print(x) }
if (!exists("ggmd"))  ggmd  <- function(p) { print(p) }
```

Question 5

Download the patient_names.csv and patient_properties.csv files from Canvas and read them into R. Manually set the date columns to be date variables. Print the first 10 observations of each.

In [95]:
```
patient_names <- read.csv(file = "/Users/brookesjj/git/BIOS512/data/pa
patient_properties <- read.csv(file = "/Users/brookesjj/git/BIOS512/da
#changing date columns to date variables)
patient_names$BIRTHDATE <- as.Date(patient_names$BIRTHDATE, "%m/%d/%y"
patient_names$DEATHDATE <- as.Date(patient_names$DEATHDATE, "%m/%d/%y"
head(patient_names, n = 10)
head(patient_properties, n = 10)
```

A data.frame: 10 × 7

	ID	BIRTHDATE	DEATHDATE	FIRST	LAST	Cl
	<chr>	<date>	<date>	<chr>	<chr>	<ch
1	5605b66b-e92d-c16c-1b83-b8bf7040d51f	1977-03-19	NA	Nikita578	Erdman779	Quin
2	6e5ae27c-8038-7988-e2c0-25a103f01bfa	2040-02-19	NA	Zane918	Hodkiewicz467	Bost
3	8123d076-0886-9007-e956-d5864aa121a7	2058-06-04	NA	Quinn173	Marquardt819	Quin
4	770518e4-6133-648e-60c9-071eb2f0e2ce	2028-12-25	2017-09-29	Abel832	Smitham825	Bost
5	f96addf5-81b9-0aab-7855-d208d3d352c5	2028-12-25	2014-02-23	Edwin773	Labadie908	Bost
6	8e9650d1-788a-78f9-4a28-d08f7f95354a	2028-12-25	NA	Frankie174	Oberbrunner298	Bost
7	183df435-4190-060e-8f8e-bf63c572b266	2057-11-08	NA	Eilene124	Walsh511	Cambric
8	720560d4-51da-c38c-ee90-c15935278df1	1972-06-27	NA	Lowell343	Price929	Quin
9	217851b0-5f47-d376-18b9-0fe4ba77207e	2054-03-06	NA	Adrian111	Gleason633	Bost
10	ff331e5c-ab16-e218-f39a-63e11de1ed75	2027-07-10	NA	Eugene421	Abernathy524	Bost

A data.frame: 10 × 3

	ID	property	value
	<chr>	<chr>	<chr>
1	5605b66b-e92d-c16c-1b83-b8bf7040d51f	MARITAL	M
2	5605b66b-e92d-c16c-1b83-b8bf7040d51f	RACE	white
3	5605b66b-e92d-c16c-1b83-b8bf7040d51f	ETHNICITY	nonhispanic
4	5605b66b-e92d-c16c-1b83-b8bf7040d51f	GENDER	F
5	6e5ae27c-8038-7988-e2c0-25a103f01bfa	MARITAL	M
6	6e5ae27c-8038-7988-e2c0-25a103f01bfa	RACE	white
7	6e5ae27c-8038-7988-e2c0-25a103f01bfa	ETHNICITY	nonhispanic
8	6e5ae27c-8038-7988-e2c0-25a103f01bfa	GENDER	M
9	8123d076-0886-9007-e956-d5864aa121a7	MARITAL	M
10	8123d076-0886-9007-e956-d5864aa121a7	RACE	white

Question 6

In the data frame pulled from patient_properties, you'll notice that the data is long, not wide. Do a pivot to make the properties their own columns. Print the first 10 observations after you do so.

In [97]:
```
properties_wide <- patient_properties %>%
                        pivot_wider(id_cols = ID,
                                    names_from = property,
                                    values_from = value)
head(properties_wide, n = 10)
```

A tibble: 10 × 5

ID	MARITAL	RACE	ETHNICITY	GENDER
<chr>	<chr>	<chr>	<chr>	<chr>
5605b66b-e92d-c16c-1b83-b8bf7040d51f	M	white	nonhispanic	F
6e5ae27c-8038-7988-e2c0-25a103f01bfa	M	white	nonhispanic	M
8123d076-0886-9007-e956-d5864aa121a7	M	white	nonhispanic	M
770518e4-6133-648e-60c9-071eb2f0e2ce	M	white	hispanic	M
f96addf5-81b9-0aab-7855-d208d3d352c5	M	white	hispanic	M
8e9650d1-788a-78f9-4a28-d08f7f95354a	M	white	hispanic	M
183df435-4190-060e-8f8e-bf63c572b266	M	asian	nonhispanic	F
720560d4-51da-c38c-ee90-c15935278df1	M	white	nonhispanic	M
217851b0-5f47-d376-18b9-0fe4ba77207e	S	black	hispanic	M
ff331e5c-ab16-e218-f39a-63e11de1ed75	M	native	hispanic	M

Question 7

Perform a left join of the names and properties_wide data frames by the ID column and print the first 10 rows.

In [98]:
```
patients_left <- patient_names %>% left_join(properties_wide,
                                  by = "ID")
head(patients_left, n=10)
```

A data.frame: 10 × 11

	ID	BIRTHDATE	DEATHDATE	FIRST	LAST	CI
	<chr>	<date>	<date>	<chr>	<chr>	<ch
1	5605b66b-e92d-c16c-1b83-b8bf7040d51f	1977-03-19	NA	Nikita578	Erdman779	Quin
2	6e5ae27c-8038-7988-e2c0-25a103f01bfa	2040-02-19	NA	Zane918	Hodkiewicz467	Bost
3	8123d076-0886-9007-e956-d5864aa121a7	2058-06-04	NA	Quinn173	Marquardt819	Quin
4	770518e4-6133-648e-60c9-071eb2f0e2ce	2028-12-25	2017-09-29	Abel832	Smitham825	Bost
5	f96addf5-81b9-0aab-7855-d208d3d352c5	2028-12-25	2014-02-23	Edwin773	Labadie908	Bost
6	8e9650d1-788a-78f9-4a28-d08f7f95354a	2028-12-25	NA	Frankie174	Oberbrunner298	Bost
7	183df435-4190-060e-8f8e-bf63c572b266	2057-11-08	NA	Eilene124	Walsh511	Cambric
8	720560d4-51da-c38c-ee90-c15935278df1	1972-06-27	NA	Lowell343	Price929	Quin
9	217851b0-5f47-d376-18b9-0fe4ba77207e	2054-03-06	NA	Adrian111	Gleason633	Bost
10	ff331e5c-ab16-e218-f39a-63e11de1ed75	2027-07-10	NA	Eugene421	Abernathy524	Bost

Question 8

Notice something interesting about the names in our data set. Fix the name formatting and print the first 10 observations.

In [99]:
```
patients_left$FIRST <- gsub("[0-9]", "", patients_left$FIRST)
patients_left$LAST <- gsub("[0-9]", "", patients_left$LAST)
head(patients_left, 10)
```

A data.frame: 10 × 11

	ID	BIRTHDATE	DEATHDATE	FIRST	LAST	CITY	
	<chr>	<date>	<date>	<chr>	<chr>	<chr>	
1	5605b66b-e92d-c16c-1b83-b8bf7040d51f	1977-03-19	NA	Nikita	Erdman	Quincy	Ma
2	6e5ae27c-8038-7988-e2c0-25a103f01bfa	2040-02-19	NA	Zane	Hodkiewicz	Boston	Ma
3	8123d076-0886-9007-e956-d5864aa121a7	2058-06-04	NA	Quinn	Marquardt	Quincy	Ma
4	770518e4-6133-648e-60c9-071eb2f0e2ce	2028-12-25	2017-09-29	Abel	Smitham	Boston	Ma
5	f96addf5-81b9-0aab-7855-d208d3d352c5	2028-12-25	2014-02-23	Edwin	Labadie	Boston	Ma
6	8e9650d1-788a-78f9-4a28-d08f7f95354a	2028-12-25	NA	Frankie	Oberbrunner	Boston	Ma
7	183df435-4190-060e-8f8e-bf63c572b266	2057-11-08	NA	Eilene	Walsh	Cambridge	Ma
8	720560d4-51da-c38c-ee90-c15935278df1	1972-06-27	NA	Lowell	Price	Quincy	Ma
9	217851b0-5f47-d376-18b9-0fe4ba77207e	2054-03-06	NA	Adrian	Gleason	Boston	Ma
10	ff331e5c-ab16-e218-f39a-63e11de1ed75	2027-07-10	NA	Eugene	Abernathy	Boston	Ma

Question 9

Using a for statement to loop through the categorical variables (excluding name and ID), print the counts of each unique value in descending order, using the mdpre() function for formatting.

In [100…

```
#definding mdpre

pat_charact <- subset(patients_left, select = MARITAL:GENDER)
#patients_left %>% group_by(MARITAL) %>% tally()


group_and_tally <- function(df, column){
  df %>%
    group_by({{ column }}) %>%
      tally()
}

for (i in colnames(pat_charact)) {
  cat("\nCounts for column:", i, "\n")
  mdpre(group_and_tally(patients_left, !!sym(i)))
}
```

```
Counts for column: MARITAL
# A tibble: 5 × 2
  MARITAL     n
  <chr>   <int>
1 Fine        1
2 M         782
3 S         189
4 male        1
5 NA          1

Counts for column: RACE
# A tibble: 7 × 2
  RACE         n
  <chr>    <int>
1 asian       90
2 asiann       1
3 black      163
4 hawaiian    13
5 native      11
6 other       16
7 white      680

Counts for column: ETHNICITY
# A tibble: 4 × 2
  ETHNICITY       n
  <chr>       <int>
1 hispani         1
2 hispanic      190
3 nonhispani      2
4 nonhispanic   781

Counts for column: GENDER
# A tibble: 5 × 2
  GENDER     n
  <chr>  <int>
1 F        478
2 Female     1
3 M        493
4 Male       1
5 female     1
```

Question 10

If you see any weird values, get rid of the ones that don't make sense, and combine the ones that are formatted wrong. Don't forget ot check the dates! Print the new tables for categorical values, and print the date ranges.

In [101…

```
#noticing that some ages are incorrectly, so fixing this
patients_left$BIRTHDATE <- as.Date(ifelse(patients_left$BIRTHDATE > Sy
  format(patients_left$BIRTHDATE, "19%y-%m-%d"),
  format(patients_left$BIRTHDATE)))
```

```
patients_left$DEATHDATE <- as.Date(ifelse(patients_left$DEATHDATE > Sy
  format(patients_left$DEATHDATE, "19%y-%m-%d"),
  format(patients_left$DEATHDATE)))

#removing values that do not make sense for MARITAL (aka, not M or S)
patients_left$MARITAL <- ifelse(patients_left$MARITAL %in% c("M", "S")
                                patients_left$MARITAL, NA)
patients_left <- patients_left %>% mutate(MARITAL = recode(MARITAL,
                            M = "Married",
                            S = "Single"))

#fixing "asiann" to be "asian" in the RACE variable
patients_left$RACE <- ifelse(patients_left$RACE == "asiann",
                             "asian", patients_left$RACE)

#fixing Ethnicity column values of hispani and nonhispani
patients_left <- patients_left %>% mutate(ETHNICITY = case_when(
                                                  ETHNICITY %in% c("
                                                  ETHNICITY %in% c("
                                                  TRUE ~ ETHNICITY)
                                          )

#fixing Gender columns to all be "Female" or "Male"
patients_left <- patients_left %>% mutate(GENDER = case_when(
                                                  GENDER %in% c("F",
                                                  GENDER %in% c("M",
                                                  TRUE ~ GENDER)
                                          )

for (i in colnames(pat_charact)) {
  cat("\nCounts for column:", i, "\n")
  mdpre(group_and_tally(patients_left, !!sym(i)))
}
```

```
Counts for column: MARITAL
# A tibble: 3 × 2
  MARITAL     n
  <chr>   <int>
1 Married   782
2 Single    189
3 NA          3

Counts for column: RACE
# A tibble: 6 × 2
  RACE         n
  <chr>    <int>
1 asian       91
2 black      163
3 hawaiian    13
4 native      11
5 other       16
6 white      680

Counts for column: ETHNICITY
# A tibble: 2 × 2
  ETHNICITY       n
  <chr>       <int>
1 Hispanic      191
2 Nonhispanic   783

Counts for column: GENDER
# A tibble: 2 × 2
  GENDER     n
  <chr>  <int>
1 Female   480
2 Male     494
```

Question 11

Make a histogram of the ages of patients by gender.

In [102…

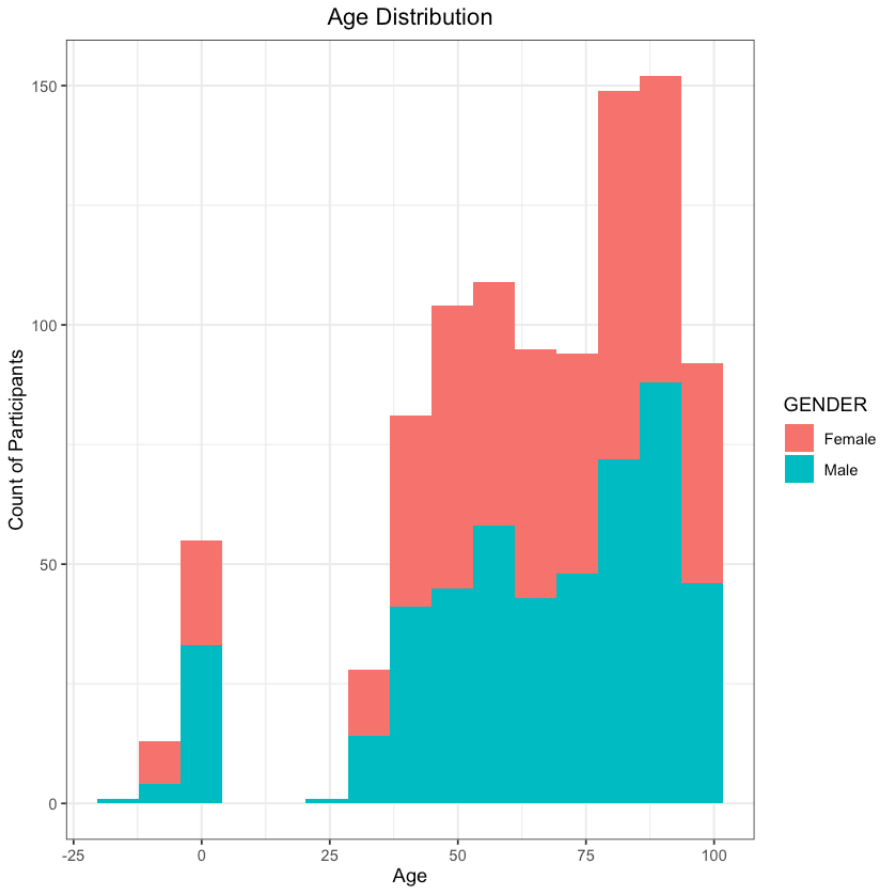
```
#creating age variable

patients_left <- patients_left %>%
                    mutate(AGE = round(ifelse(!is.na(patients_left$DEA
                                        as.numeric(patients_left$DEATHDAT
                                        as.numeric(Sys.Date() - patients_
                                ))
#checking age variable
#head(patients_left, 10)

#creating histogram
library(ggplot2)

patients_left %>% ggplot(aes(x=AGE, fill = GENDER)) +
```

```
  geom_histogram(bins = 15) +
    theme_bw() +
     labs(x="Age",y="Count of Participants",title="Age Distribution")
     theme(plot.title = element_text(hjust = 0.5))
```

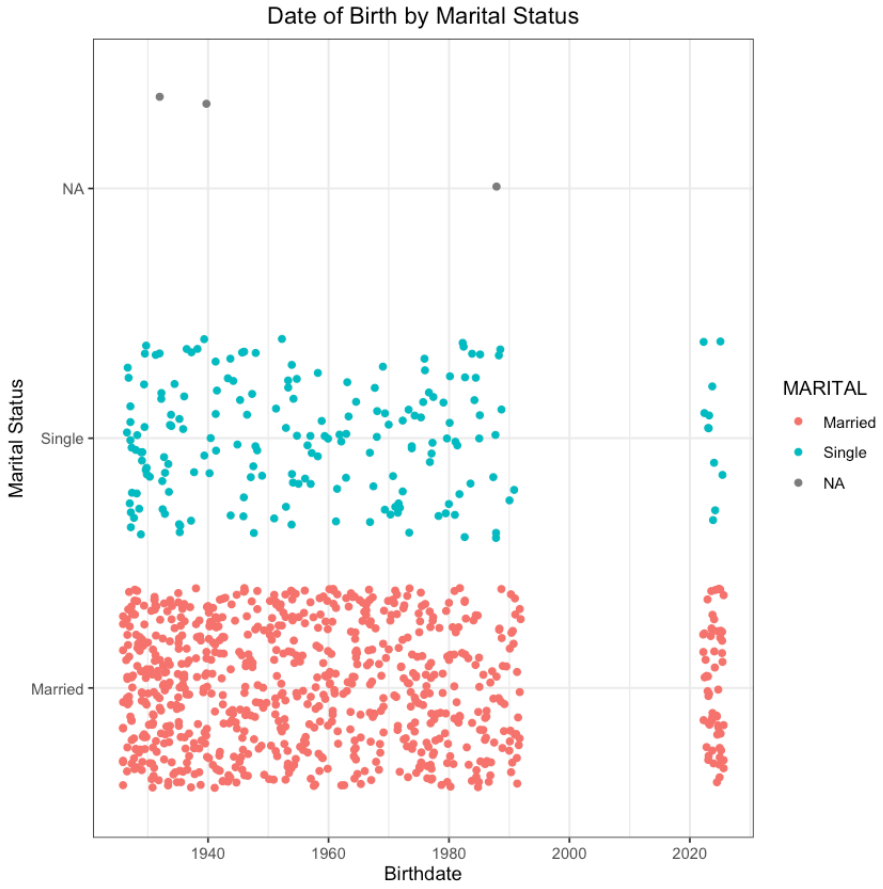


Question 12

Make a scatterplot of birthdate by martial status.

In [103…

```
patients_left %>% ggplot(aes(x=BIRTHDATE, y= MARITAL, color = MARITAL)
  geom_jitter() +
    theme_bw() +
     labs(x="Birthdate",y="Marital Status",title="Date of Birth by Mar
     theme(plot.title = element_text(hjust = 0.5))
```


Date of Birth by Marital Status
Marital Status
NA
Single
Married
1940
1960
1980
2000
2020
Birthdate
MARITAL
Married
Single
NA