Assignment 1 Questions
Please put in the question and the number in ‘numerical’ order. Thanks!

Diana Lu
Gaberiel Palomarez
Kshitij Jain
Kayla Klaus

SAS

Q1

Stone Leiker

```
/*Create a library to store the health dataset*/
libname health 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment 1\';
run;
/* Import the dataset and save in the health library as health1*/
run;PROC IMPORT DATAFILE='C:\Users\leiker-s\Desktop\msa608_2024\Assignment
1\Heath Data.csv'
        OUT=health.health1
        DBMS=csv
        REPLACE;
        GETNAMES=YES;
RUN;
/* Create a dataset called t_health*/
```

```
data t_health;
	set health.health1;
run;
/* Calculate mean and standard deviation of all variables*/
proc means data=t_health mean std;
run;
/* Print the value of RBC for subject 11005*/
proc print data=t_health;
	where subj=110055;
	var rbc;
run;
/* Get summary statistics for RBC, Hcrit, WBC, and MCHC for each hospital*/
proc means data=t_health mean std min max;
	class hosp;
	var rbc hcrit wbc mchc;
run;
/* Output mean and medican for each Hospital*/
proc means data=t_health mean median;
	class hosp;
	var rbc wbc hcrit;
	output out=Hospitals_mean_median mean=mean_rbc mean_wbc mean_hcrit
                                                                median=median_rbc
median_wbc medican_hcrit;
run;
/* Create a histogram for WBC*/
proc sgplot data=t_health;
	histogram wbc;
	title "Histogram of WBC";
run;
/* Create a boxplot for WBC*/
proc sgplot data=t_health;
	vbox wbc;
	title "Boxplot of WBC";
run;
/* Create a scatterplot where X-axis is RBC and Y-axis is WBC*/
proc sgplot data=t_health;
	scatter x=rbc y=wbc;
	title "Scatterplot of RBC vs WBC";
run;
/*Create separate datasets for subject 210006, 310032, and 410010*/
```

```
data s210006;
        set t_health;
        where subj=210006;
run;

data s310032;
        set t_health;
        where subj=310032;
run;

data s410010;
        set t_health;
        where subj=410010;
Run;
```

KAYLA'S CODE

```
/*Question 1.1*/

libname health "C:\Users\kayla\OneDrive\Desktop\MSA608_2024\Health";run;

PROC IMPORT DATAFILE = "C:\Users\kayla\OneDrive\Desktop\MSA608_2024\Assignment
1\Heath Data.csv"
OUT=Health1
DBMS=csv
REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
RUN;

/*Question 1.2*/

PROC MEANS DATA=Health1 MEAN STD; RUN;

/*What is the value of rbc for subj=110055? 4.34*/

/*Question 1.3*/

PROC MEANS DATA=Health1;
class hosp;
var rbc hcrit wbc mchc;
RUN;

/*Question 1.4*/
PROC MEANS DATA=Health1 MEAN MEDIAN;
class hosp;
var rbc hcrit wbc mchc;
title "Hospitals' Mean and Median of RBC, WBC, HCRIT.";
output out=Health.summary_stats;
RUN;

/*Question 1.5*/
/*Histogram*/
```

```
TITLE "WBC Histogram";
PROC UNIVARIATE DATA = Health1 NOPRINT;
HISTOGRAM wbc/NORMAL;
RUN;
/*Box Plot*/
PROC SQL;
Create table wbc1 as
select wbc from Health1;
run;

PROC SGPLOT DATA=Health1;
VBOX wbc;
TITLE "WBC Box Plot";
RUN;

/*Scatter Plot*/
proc sgplot data=Health1;
   scatter x = rbc  y = wbc;
run;

/*Question 1.6*/
data health.s1; set health.Health1;
if subj=210006;
title "The s210006 data set";
run;


data health.s2; set health.Health1;
if subj=3100032;
title "The s3100032 data set";
run;

data health.s3; set health.Health1;
if subj=410010;
title "The s410010 data set";
Run;
```

DIANA's CODE

```
/*1.1 Create a library and name it as "health." */
libname health "C:\Users\diana\OneDrive - Texas A&M
University\Desktop\Data_Analytics\Fall_2024\ANLY608\Assignment\Assignment1";
run;
quit;

/*1.2 Import the dataset and store it in heath library as health1. */
proc import datafile = "C:\Users\diana\OneDrive - Texas A&M
University\Desktop\Data_Analytics\Fall_2024\ANLY608\Assignment\Assignment1\Data\Heath
Data.csv"
out = health.health1
dbms = csv
replace;
getnames = yes;
run;

/*1.3 Create a temporary dataset named "t_health1."*/
data t_health1;
set health.health1;
run;
quit;

/*1.4 Find the means, standard deviation of all the variables. What is the value of rbc for
subj=110055?*/
proc means data=t_health1 mean std;
run;
quit;

proc print data= t_health1;
var subj rbc;
where subj = 110055;
run;

/*1.5 Find the summary statistics (i.e., number of observations, mean, std. dev, minimum and
maximum)
of rbc, hcrit, wbc, and mchc for each hospital (hosp).*/
proc univariate data = t_health1;
var rbc hcrit wbc mchc;
by hosp;
run;
```

```
/*1.6 Create an output dataset that contains mean and median of RBC, WBC, and HCRIT for
each hospital.
Title the output dataset as "Hospitals' Mean and Median of RBC, WBC, HCRIT.*/

proc means data = t_health1 mean median;
var rbc wbc hcrit;
by hosp;
output out= Hospital_Mean_Median mean=mean_rbc mean_wbc mean_hcrit median =
median_rbc median_wbc median_hcrit;
run;

/*1.7 Create a histogram and a boxplot for WBC. Also create a scatterplot where y axis=wbc
and x-axis=rbc.*/
proc univariate data = t_health1 noprint; /*stop printing results viewer of univariate tables (ex.
sum of obs, median, mean, etc)*/
histogram wbc;
run;

proc sgplot data = t_health1;
vbox wbc;
run;

proc sgplot data = t_health1;
scatter y = wbc x = rbc;
run;

/*1.8 Create three datasets-s1, s2, s3 for subj=210006, 3100032,410010
(name them as The s210006 data set, The s310032 data set, and The s410010 data set,
respectively.*/

data s210006;
set t_health1;
where subj = 210006;
run;

data s310032;
set t_health1;
where subj = 3100032;
run;

data s410010;
set t_health1;
where subj = 410010;
```

```
run;
```

Q2

STONE’S CODE

```
/*Create a library to store the grades dataset*/
libname grades 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment 1';
run;
/* Import the dataset and save in the grades library as student_grades*/
PROC IMPORT DATAFILE= 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment
1\Student Grades.csv'
        OUT=grades.student_grades
        DBMS=csv
        REPLACE;
        GETNAMES=YES;
RUN;
/*Sort the data by student ID and grade*/
Proc sort data=grades.student_grades out=sorted_grades;
        by idno grade;
run;
/* Create a dataset with the lowest grade for each student*/
data lowest_grade;
        set sorted_grades;
        by idno;
        if first.idno then output;
run;
/* Print the lowest grades and the corresponding semesters*/
proc print data=lowest_grade;
        title 'Lowest Grades and Semester for each Student';
run;
/* Transpose the dataset from long to wide format*/
proc transpose data=grades.student_grades out=wide_grades;
        by idno;
        id gtype;
        var grade;
run;
/* Print the transposed student grades*/
```

```
proc print data=wide_grades;
	title 'Transposed Student Grades';
Run;
```

KAYLA'S CODE

```
/*Question 2.1*/
PROC IMPORT DATAFILE = "C:\Users\kayla\OneDrive\Desktop\MSA608_2024\Assignment
1\Student Grades.csv"
OUT=StudentGrades
DBMS=csv
REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
RUN;

PROC MEANS MIN DATA=StudentGrades;
var grade;
RUN;
/*Question 2.2*/
PROC transpose data=StudentGrades out=sgwide;
	var _all_;
run;
```

DIANA’s CODE

```
/* Bring the CSV file into the Work folder*/
proc import datafile = "C:\Users\diana\OneDrive - Texas A&M
University\Desktop\Data_Analytics\Fall_2024\ANLY608\Assignment\Assignment1\Data\
Student Grades.csv"
out = grades1
dbms = csv
REPLACE;
GETNAMES=YES;
RUN;

/*Lowest grade of the students along with the semester.*/
proc sql;
        select l_name, gtype, min(grade)
        from grades1;
        group by l_name;
quit;

/*Transpose data using DATA step */
proc transpose data = grades1
out = grades2;
var _all_;
run;
```

Q3

STONE’S CODE

```
/*Create a library to store the weather dataset*/
```

```
libname weather 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment 1';
run;
/* Import the atmosphere dataset*/
PROC IMPORT DATAFILE= 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment
1\Atmosphere.csv'
	OUT=weather.atmosphere
	DBMS=csv
	REPLACE;
	GETNAMES=YES;
RUN;
/* Convert Celsius to Fahrenheit using a DO loop for each month*/
data fahrenheit;
	set weather.atmosphere;
	array months {*} jan--dec;
	do i = 1 to dim(months);
		months[i] = 1.8 * months[i] + 32;
	end;
run;
/* Print the converted temperatire data*/
proc print data=fahrenheit;
	title 'Temperature Converted to Fahrenheit';
run;

data fahrenheit2;
	set weather.atmosphere;
	array temps {*} jan--dec;
	do i = 1 to dim(temps);
		temps[i] = 1.8 * temps[i] + 32;
	end;
run;
/* Print the fahrenheit*/
proc print data=fahrenheit2;
	title 'Celsius to Fahrenheit Conversion (fahrenheit2 dataset)';
Run;
```

KAYLA'S CODE:

```
/*Problem Q3*/

/*Question 3 - Convert Celsius to Fahrenheit*/

PROC IMPORT DATAFILE = "C:\Users\kayla\OneDrive\Desktop\MSA608_2024\Assignment
1\Atmosphere.csv"
OUT=Atmosphere
DBMS=csv
REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
RUN;

data fahrenheit2 (drop = i);
   set Work.Atmosphere;
        array amonths [12] jan feb mar apr may jun jul aug sep oct nov dec;
        do i = 1 to 12;
        amonths[i] = (1.8 * amonths[i]) + 32;
        end;
run;
```

DIANA'S CODE

```
/*Q3: Please refer to atmosphere dataset. This dataset records the temperature in
Celsius across cities over months.

Convert the temperature from Celsius to Fahrenheit by using the formula Fahrenheit =
1.8*Celsius + 32.
Create a do loop to covert the Celsius to Fahrenheit using the formula Fahrenheit.
Name the output dataset as fahrenheit2.*/

proc import datafile = "C:\Users\diana\OneDrive - Texas A&M
University\Desktop\Data_Analytics\Fall_2024\ANLY608\Assignment\Assignment1\Data\
Atmosphere.csv"
out = atmosphere
dbms = csv
replace;
getnames = yes;
run;
```

```
data farenheit2;
set WORK.ATMOSPHERE;
array month[12] jan feb mar apr may jun jul aug sep oct nov dec;
do i = 1 to 12;
        month[i] = (1.8 * month[i]) + 32;
        end;
        drop i;
run;
```

Q4

STONE'S CODE

```
/*Create a library to store the patient dataset*/
libname patients 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment 1';
run;
/* Import datasets d1 and d2*/
PROC IMPORT DATAFILE= 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment
1\d1.csv'
        OUT=patients.d1
        DBMS=csv
        REPLACE;
        GETNAMES=YES;
RUN;

PROC IMPORT DATAFILE= 'C:\Users\leiker-s\Desktop\msa608_2024\Assignment
1\d2.csv'
        OUT=patients.d2
        DBMS=csv
        REPLACE;
        GETNAMES=YES;
RUN;
/* Sort both datasets by OD for merging*/
proc sort data=patients.d1; by id; run;
proc sort data=patients.d2; by id; run;
/* Merge two datasets by ID*/
data merged;
        merge patients.d1 patients.d2;
```

```
        by id;
Run;

/* Merge two datasets by ID*/
data merged;
        merge patients.d1 (in=a) patients.d2 (in=b);
        by id;
        if a and b;
run;
/*Display the structure of the merged dataset*/
proc contents data=merged; run;
proc print data=merged (obs=10); run;
/* Calculate the means of numeric variables for later use*/
proc means data=merged noprint;
        var _numeric_;
        output out=means mean= / autoname;
run;
/* Replace missing values with the mean of the corresponding variable*/
data cleaned_data;
        set merged;
        if _n_ = 1 then set means;

        array vars {*} _numeric_;
        array means_arr {*} _numeric_mean_;

        do i = 1 to dim(vars);
                if missing(vars[1]) then vars[i] = means_arr[i];
        end;
Run;

/* Print the cleaned dataset with missing values replaced*/
proc print data=cleaned_data (obs=10);
        title 'Merged Dataset with Missing Values Replaced by Mean';
run;
```

KAYLA'S CODE:

```
/*Question 4.1*/
```

```
PROC IMPORT DATAFILE = "C:\Users\kayla\OneDrive\Desktop\MSA608_2024\Assignment
1\d1.csv"
OUT=d1
DBMS=csv
REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
RUN;

PROC IMPORT DATAFILE = "C:\Users\kayla\OneDrive\Desktop\MSA608_2024\Assignment
1\d2.csv"
OUT=d2
DBMS=csv
REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
RUN;

DATA dmerged; MERGE d1 d2;BY id;RUN;

/*Question 4.2*/
proc stdize data=dmerged out=demerged1 reponly method=mean;
   var id age visit outcome;
run;

proc print data=demerged1;
run;
```

DIANA'S CODE

```
/*Q4: Dataset 'd1' represents the id and age of patients,
and d2 represents id, visit, and outcome.

Merge the two datasets.
Replace the missing observations (if any) by the mean of the variable.*/

proc import datafile = "C:\Users\diana\OneDrive - Texas A&M
University\Desktop\Data_Analytics\Fall_2024\ANLY608\Assignment\Assignment1\Data\d1.csv"
out = d1
dbms = csv
replace;
getnames = yes;
run;
```

```
proc import datafile = "C:\Users\diana\OneDrive - Texas A&M
University\Desktop\Data_Analytics\Fall_2024\ANLY608\Assignment\Assignment1\Data\d2.csv"
out = d2
dbms = csv
replace;
getnames = yes;
run;

proc sort data = d1;
by id;
run;

proc sort data = d2;
by id;
run;

data d1_d2_merge;
merge d1 d2;
by id;
run;

/*Calculate mean for each column*/

proc means data = d1_d2_merge;
var age visit outcome;
output out = means (drop = _type_ _freq_) mean = /autoname;
/*autoname - automatcally names them to age_MEAN etc instead of manually specify names*/
run;

proc stdize data=d1_d2_merge out=merged1 reponly method=mean;
   var id age visit outcome;
run;
```

R

5Q

STONE'S CODE

```
# Set the working directory and list files
```

```
setwd("C:/Users/leiker-s/Desktop/msa608_2024/Assignment 1")
list.files()

# Create a sequence of weights from 7 to 40, spaced by 1.5
weights_of_babies <- seq(7, 40, by = 1.5)
mean_weights <- mean(weights_of_babies)

# Calculate the mean and standard deviation of the weights
sd_weights <- sd(weights_of_babies)
print(weights_of_babies)

#print the sequence, mean, and standard deviation
print(mean_weights)
print(sd_weights)

#Create and display a histogram of the weights
hist(weights_of_babies, main="Histogram of Weights of Babies", xlab="Weight",
col="lightblue")
```

KAYLA'S CODE:

```
#Question 5
weights_of_babies <- seq(7,40,1.5)

mean(weights_of_babies)

sd(weights_of_babies)

#Create Histogram:
hist(weights_of_babies)
```

DIANA'S CODE

```
#Q5: Create a variable named "weights of babies" with starting value=7
#and ending value=40, spaced by 1.5.


weights_of_babies <- seq(7, 40, 1.5)

#Find the mean and standard deviation of weights of babies.
mean(weights_of_babies)
sd(weights_of_babies)

#Create a histogram for weights of babies.
hist(weights_of_babies)
```

Q6

STONE'S CODE

```
# Create a sales vector with mixed values and missing values
sales <- c(NA, "TP", 4, 6.7, 'c', NA, 12)

# Find the positions of NA values in the sales vector
na_positions <- which(is.na(sales))
print(na_positions)

# Count the total number of NA values in the sales vector
total_nas <- sum(is.na(sales))
print(total_nas)
```

KAYLA'S CODE:

```
#Question 6
sales <- c(NA, "TP", 4, 6.7, 'c', NA, 12)
#Find NA in the variable:
is.na(sales)
#Identify NA's in Vector

which(is.na(sales))
```

```
#Identify total number of NA's:
sum(is.na(sales))
```

DIANA'S CODE

```
#Q6: The information on sales of a startup looks as follows

sales<- c (NA, "TP", 4, 6.7, 'c', NA, 12)
print(sales)

#Find "NA" in the variable.
find_na <- is.na(sales)
print(find_na)

#Identify NAs in Vector (i.e., position in the variable where we have NA).
na_location <- which(is.na(sales))
print(na_location)
#outcome: 1 6


#Identify total number of NAs.
na_sum <- sum(is.na(sales))
print(na_sum)
#total = 2
```

Q7

STONE'S CODE

```
# Create a data frame with missing values
dataframe <- data.frame(
  Name = c("Bell", "Dia", "KKN", "Nia"),
  Physics = c(98, 87, 91, 94),
  Chemistry = c(NA, 84, 93, 87),
  Mathematics = c(91, 86, NA, NA))

# Replace missing values in Chemistry and Mathematics columns with the column
mean
```

```
dataframe$Chemistry[is.na(dataframe$Chemistry)] <- mean(dataframe$Chemistry,
na.rm = TRUE)
dataframe$Mathematics[is.na(dataframe$Mathematics)] <-
mean(dataframe$Mathematics, na.rm = TRUE)
# Print the updated data frame
print(dataframe)
```

KAYLA'S CODE:

```
#Question 7

dataframe <- data.frame( Name = c("Bell", "Dia", "KKN", "Nia"),
                         Physics = c(98, 87, 91, 94),
                         Chemistry = c(NA, 84, 93, 87),
                         Mathematics = c(91, 86, NA, NA) )

View(dataframe)

complete.cases(dataframe)

#install packages
install.packages("dplyr")
install.packages("plyr")
install.packages("tidyr")
install.packages("magrittr")

#Remove NA Values - can't get %>% to work
dataframe %>% drop_na(Chemistry, Mathematics)

#Replace missing values with mean of column. - This isn't working either.
dataframe$Chemistry[is.null(dataframe$Chemistry)] <- mean(dataframe$Chemistry)
dataframe$Mathematics[is.null(dataframe$Mathematics)] <- mean(dataframe$Mathematics)
```

DIANA'S CODE

```
#Q7: A data frame representing the scores of Bell, Dia, KKN, and Nia ON Physics, Chemistry,
and Mathematics looks as follows

dataframe <- data.frame( Name = c("Bell", "Dia", "KKN", "Nia"),
  Physics = c(98, 87, 91, 94),
  Chemistry = c(NA, 84, 93, 87),
  Mathematics = c(91, 86, NA, NA) )
```

```
print(dataframe)
#Find the missing values, and replace them with the mean of the respective column/row.
class_na_find <- is.na(dataframe)
class_na_location <- which(is.na(dataframe))
print(class_na_find)
print(class_na_location)

#Replaced Na with Mean
dataframe$Chemistry[is.na(dataframe$Chemistry)] <- mean(dataframe$Chemistry, na.rm =
TRUE)
#Round Mean to zero decimal points
dataframe$Chemistry <- round(dataframe$Chemistry, digits = 0)

#Repeat process with Mathmatics
dataframe$Mathematics[is.na(dataframe$Mathematics)] <- mean(dataframe$Mathematics,
na.rm = TRUE)
dataframe$Mathematics <- round(dataframe$Mathematics, digits = 0)
print(dataframe)
```

KSHITIJ'S CODE

```
# Create the dataframe
dataframe <- data.frame(
  Name = c("Bell", "Dia", "KKN", "Nia"),
  Physics = c(98, 87, 91, 94),
  Chemistry = c(NA, 84, 93, 87),
  Mathematics = c(91, 86, NA, NA)
)

# Function to replace missing values with the mean of the respective column
replace_na_with_mean <- function(column) {
  # Replace NA with the mean of the column
  column[is.na(column)] <- mean(column, na.rm = TRUE)
  return(column)
}

# Apply the function to all columns (except the 'Name' column)
dataframe$Physics <- replace_na_with_mean(dataframe$Physics)
dataframe$Chemistry <- replace_na_with_mean(dataframe$Chemistry)
dataframe$Mathematics <- replace_na_with_mean(dataframe$Mathematics)

# Print the dataframe with missing values replaced by the column mean
print(dataframe)
```

```
# Alternatively, to replace missing values with the mean of the respective row:
# Function to replace NA in each row by the row mean
replace_na_with_row_mean <- function(row) {
  row[is.na(row)] <- mean(row, na.rm = TRUE)
  return(row)
}

# Apply the function to each row (excluding the 'Name' column)
dataframe[, -1] <- t(apply(dataframe[, -1], 1, replace_na_with_row_mean))

# Print the dataframe with missing values replaced by row mean
print(dataframe)
```

Q8

STONE'S CODE

```
# Import the Titanic dataset
t1 <- read.csv('titanic.csv')

# Find the total number of survivors and print the result
num_survived <- sum(t1$Survived == 1, na.rm = TRUE)
print(paste("Total survived:", num_survived))

# Find the total number of deaths and print the result

num_dead <- sum(t1$Survived == 0, na.rm = TRUE)
print(paste("Total dead:", num_dead))

# Count the number of males and print the result
num_males <- sum(t1$Sex == "male", na.rm = TRUE)
print(paste("Number of males:", num_males))

# Count the number of females and print the result
num_females <- sum(t1$Sex == "female", na.rm = TRUE)
print(paste("Number of females:", num_females))

# Find the maximum age and print the result
max_age <- max(t1$Age, na.rm = TRUE)
```

```
print(paste("Maximum age:", max_age))

# Find the medium age and print the result
median_age <- median(t1$Age, na.rm = TRUE)
print(paste("Median age:", median_age))

# Count the number of missing age values and print the result
missing_ages <- sum(is.na(t1$Age))
print(paste("Number of missing age values:", missing_ages))

# Drop all rows with missing values
t2 <- na.omit(t1)


# Create a pie chart showing the proportion of survivors vs deaths
pie(table(t1$Survived), labels = c("Dead", "Survived"), main="Survivors vs Deaths",
col=c("red", "green"))

# Create a histogram of the survived people based on gender
survived_data <- t1[t1$Survived == 1, ]
hist(as.numeric(survived_data$Sex == "male"), breaks=2, main="Survived by Gender",
xlab="Gender (0=Female, 1=Male)", col="blue")
```

KAYLA'S CODE:

```
#Question 8
#import data
t1 <- read.csv("C:/Users/kayla/OneDrive/Desktop/MSA608_2024/Assignment 1/titanic.csv",
header = T)
View(t1)
#Total Survivors:
sum(t1$Survived == 1, na.rm = TRUE)
#Total dead:
sum(t1$Survived == 0, na.rm = TRUE)
#Total Males
sum(t1$Sex == 'male', na.rm = TRUE)
#Total Females
sum(t1$Sex == 'female', na.rm = TRUE)
#Max Age
max(t1$Age, na.rm = TRUE)
#Median Age
median(t1$Age, na.rm = TRUE)
```

```
#Count of missing observations
sum(is.na(t1))

#Drop missing observations and create new dataset.
t2 <- na.omit(t1)
View(t2)

#Create pie diagram
pie(table(t2$Survived), labels = c("Died", "Survived"), main="Pie Chart of Survivors")

#Create histogram
survivors <- subset(t2, Survived==1)
View(survivors)
survivors$Sex <- ifelse(survivors$Sex == 'female',c(1),c(0))
hist(survivors$Sex)
‘
```

DIANA’S CODE

```
#Q8: Import titanic dataset (name it as t1)
titanic <- read.csv("C:/Users/diana/OneDrive - Texas A&M
University/Desktop/Data_Analytics/Fall_2024/ANLY608/Assignment/Assignment1/Data/titanic.cs
v")
print(titanic)
names(titanic)
#Find the following.

#The number of total people who survived.
sum(titanic$Survived)

#Number of total people dead
sum(titanic$Survived == 0)

#Number of males in the titanic
sum(titanic$Sex=='male')

#Number of females in the titanic
sum(titanic$Sex=='female')

#Maximum age among all people in titanic
max(titanic$Age, na.rm = TRUE)

#Median age
```

```
median(titanic$Age, na.rm = TRUE)

#How many missing observations are there in the dataset?
sum(is.na(titanic))
#ans: 87

#Drop all the missing observations, and create a new dataset (name it as t2)
install.packages("tidyverse")
t2 <- drop_na(titanic)

#Divide survived and dead people into a separate list.
survived <- subset(titanic, Survived == 1)

deceased <- subset(titanic, Survived == 0)

#Create a pie diagram that shows proportion or number of people survived vs died.
#Determine the frequency
survive_portion <- table(titanic$Survived)
print(survive_portion)

#Create labels for the pie chart
labels <- c('Deceased', 'Survived')

pie(survive_portion, labels = paste(labels, survive_portion), main = "Number of people who
survived vs died on the Titanic")

#Create a histogram for the survived people based on gender.
install.packages("ggplot2")
library(ggplot2)

survive_gender <- table(survived$Sex)
gender_label <- c('male', 'female')
print(survive_gender)


#The function factor is used to encode a vector as a factor
#(the terms ‘category’ and ‘enumerated type’ are also used for factors).

#levels = an optional vector of the unique values (as character strings)
#that x might have taken.

gender_counts <- table(factor(survived$Sex, levels = gender_label))
print(gender_counts)
barplot(gender_counts)
```

KSHITIJ'S CODE

```
# Load necessary libraries
library(ggplot2)

# Import the Titanic dataset
t1 <- read.csv('C:/Users/jaink/OneDrive/Desktop/msa608_2024/R/titanic.csv')

# View the structure of the dataset to understand column names
str(t1)

# Number of total people who survived
total_survived <- sum(t1$Survived == 1, na.rm = TRUE)

# Number of total people who died
total_dead <- sum(t1$Survived == 0, na.rm = TRUE)

# Number of males in the Titanic
total_males <- sum(t1$Sex == 'male', na.rm = TRUE)

# Number of females in the Titanic
total_females <- sum(t1$Sex == 'female', na.rm = TRUE)

# Maximum age among all people in Titanic
max_age <- max(t1$Age, na.rm = TRUE)

# Median age
median_age <- median(t1$Age, na.rm = TRUE)

# How many missing observations are there in the dataset?
missing_observations <- sum(is.na(t1))

# Drop all the missing observations, and create a new dataset (t2)
t2 <- na.omit(t1)

# Divide survived and dead people into separate lists
survived_list <- t2[t2$Survived == 1, ]
dead_list <- t2[t2$Survived == 0, ]

# Create a pie diagram showing proportion or number of people survived vs died
survival_counts <- c(Survived = total_survived, Dead = total_dead)
pie(survival_counts, labels = names(survival_counts), main = "Proportion of Survived vs Dead")
```

```
# Create a histogram for the survived people based on gender
survived_gender <- survived_list$Sex
ggplot(survived_list, aes(x = survived_gender)) +
  geom_bar(aes(fill = survived_gender), stat = "count") +
  labs(title = "Histogram of Survived People by Gender", x = "Gender", y = "Count")
```

Q9

STONE'S CODE

```
# Create the data frame with player stats: player names, positions, points, and assists
df <- data.frame(
  player = c('A', 'B', 'C', 'D', 'E', 'F'),
  position = c('R1', 'R2', 'R3', 'R4', 'R5', NA),
  points = c(102, 105, 219, 322, 232, NA),
  assists = c(405, 407, 527, 412, 211, NA))

# Create a quality variable based on point
# high if points > 215, medium if points > 120, otherwise low
df$quality <- ifelse(df$points > 215, "high",
                                    ifelse(df$points > 120, "medium", "low"))

# Create a performance variable based on points and assists
# Great if points > 215 and assists > 10, good if points > 215 and assists > 5, otherwise
average
df$performance <- ifelse(df$points > 215 & df$assists > 10, "great",
                         ifelse(df$points > 215 & df$assists > 5, "good", "average"))

# Print the updated data frame with the new variables
print(df)
```

KAYLA'S CODE:

```
#Question 9
df <- data.frame(player = c('A', 'B', 'C', 'D', 'E', 'F'),
                 position = c('R1', 'R2', 'R3', 'R4', 'R5', NA),
                 points = c(102, 105, 219, 322, 232, NA),
                 assists = c(405, 407, 527, 412, 211, NA))
```

```
View(df)
#Create new variable "quality" represented by "high" when points>120 and
# "medium" when points>215, else low.

df$quality <- as.factor(ifelse(df$points > 215, 'high',
                    ifelse(df$points > 120, 'medium', 'low')))

#Create a new variable called "performance", representing "great" when points>215
#and assists>10, and "good" when points>215 and assists>5; else average.

df$performance <- as.factor(ifelse(df$points > 215 & df$assists > 10, 'great',
                        ifelse(df$points > 215 & df$assists > 5, 'good', 'average')))

View(df)
```

DIANA'S CODE

```
#Q9: Use the following data frame.

df <- data.frame(player = c('A', 'B', 'C', 'D', 'E', 'F'),

                 position = c('R1', 'R2', 'R3', 'R4', 'R5', NA),

                 points = c(102, 105, 219, 322, 232, NA),

                 assists = c(405, 407, 527, 412, 211, NA))

#Create a new variable "quality" represented by "high" when points>120 and "medium" when
points>215, else low.

installed.packages("dplyr")
library(dplyr)

df <- df %>%
  mutate(quality = case_when(points > 215 ~ 'high',
                          points >120 ~ 'medium',
                          TRUE ~ 'low'))

print(df)
#Create a new variable called "performance",
#representing "great" when points>215 and assists>10,
#and "good" when points>215 and assists>5; else average.
```

```
df <- df %>%
  mutate(performance = case_when(points > 215 & assists >10 ~ 'great',
                         points >215 & assists > 5 ~ 'medium',
                         TRUE ~ 'average'))
print(df)
```