Assignment 5

Problem 1

Perform the following tasks related to dates and times in R:

a. Use Sys.Date() to get the current date and assign it to a variable named today.

```
today <- Sys.Date()
today
```

```
## [1] "2025-02-17"
```

b. Convert the following string to a Date object using as.Date(): “12/31/2023”. Use the format “%m/%d/%Y”.

```
Date <- as.Date("12/31/2023", format = "%m/%d/%Y")
Date
```

```
## [1] "2023-12-31"
```

c. Create a sequence of dates starting from January 1, 2024, to January 10, 2024, using the seq.Date() function with a daily interval.

```
date_sequence <- seq.Date(from = as.Date("2024-01-01"),
                          to = as.Date("2024-01-10"), by = "day")
date_sequence
```

```
##  [1] "2024-01-01" "2024-01-02" "2024-01-03" "2024-01-04" "2024-01-05"
##  [6] "2024-01-06" "2024-01-07" "2024-01-08" "2024-01-09" "2024-01-10"
```

d. Calculate the number of days between today and “1986-01-28”.

```
days_difference <- today - as.Date("1986-01-28")
days_difference
```

```
## Time difference of 14265 days
```

```
class(days_difference)
```

```
## [1] "difftime"
```

e. Create a POSIXct date-time object for “2023-12-25 15:30:00” in the UTC time zone.

```
datetime_utc <- as.POSIXct("2023-12-25 15:30:00", tz = "UTC")
datetime_utc
```

```
## [1] "2023-12-25 15:30:00 UTC"
```

f. Add 7 days and 12 hours to the date-time object created in part (e).

```
new_datetime <- datetime_utc + (7 * 24 + 12) * 3600
new_datetime
```

```
## [1] "2024-01-02 03:30:00 UTC"
```

g. Using difftime(), calculate the difference between "2023-12-31 23:59:59" and "2023-01-01 00:00:00" in terms of weeks.

```
start_time <- as.POSIXct("2023-01-01 00:00:00", tz = "UTC")
end_time <- as.POSIXct("2023-12-31 23:59:59", tz = "UTC")
time_diff <- difftime(end_time, start_time, units = "weeks")
time_diff
```

```
## Time difference of 52.14286 weeks
```

h. Convert the following times into a difftime object: "3:45" as hours and minutes. "2:30:15" as hours, minutes, and seconds.

```
time_1 <- as.difftime("3:45", format = "%H:%M", units = "mins")
time_1
```

```
## Time difference of 225 mins
```

```
time_2 <- as.difftime("2:30:15", format = "%H:%M:%S", units = "secs")
time_2
```

```
## Time difference of 9015 secs
```

i. Write an R script to determine if a given date (e.g., "2024-02-29") is valid and falls on a leap year.

```
date_validity <- function(date_str) {
  date_input <- as.Date(date_str, format = "%Y-%m-%d")
  if (is.na(date_input)) {
    return(cat(date_str, "is NOT a valid date.\n"))
  }
  return(cat(date_str, "is a valid date.\n"))
}

# If the year is divisible by 4 and not divisible by 100, then it is a leap year
# If the year is divisible by both 4 and 100, then it should also be divisible by
# 400 to be considered a leap year

is_leap_year <- function(date_str) {
  year <- as.numeric(format(as.Date(date_str), "%Y"))
  return ((year %% 4 == 0) & (year %% 100 != 0)) | (year %% 400 == 0)
}
```

j. Using the current system time, extract and print the year, month, and day components individually.

```
year <- format(today, "%Y")
month <- format(today, "%m")
day <- format(today, "%d")

cat("Year:", year, "Month:", month, "Day:", day)
```

```
## Year: 2025 Month: 02 Day: 17
```

Problem 2

USArrests

a. Determine the number of rows and columns.

```
nrow(USArrests)
```

```
## [1] 50
```

```
ncol(USArrests)
```

```
## [1] 4
```

b. Show the row corresponding to Florida. Access the row by name, not number.

```
FL_row <- USArrests["Florida", ]
FL_row
```

```
##         Murder Assault UrbanPop Rape
## Florida   15.4     335       80 31.9
```

c. Show the last three rows.

```
last_three_rows <- tail(USArrests, 3)
last_three_rows
```

```
##               Murder Assault UrbanPop Rape
## West Virginia    5.7      81       39  9.3
## Wisconsin        2.6      53       66 10.8
## Wyoming          6.8     161       60 15.6
```

d. Find the name of the states in rows 20, 30, and 40.

```
state_names <- rownames(USArrests)[c(20, 30, 40)]
state_names
```

```
## [1] "Maryland"       "New Jersey"     "South Carolina"
```

e. Create two new data frames (keeping the same row names as USArrests): USArrests2 containing columns Murder, Assault, and Rape. Urban that only contains the column UrbanPop.

```
USArrests2 <- USArrests[, c("Murder", "Assault", "Rape")]
USArrests2
```

```
##                Murder Assault Rape
## Alabama          13.2     236 21.2
## Alaska           10.0     263 44.5
## Arizona           8.1     294 31.0
## Arkansas          8.8     190 19.5
## California        9.0     276 40.6
## Colorado          7.9     204 38.7
## Connecticut       3.3     110 11.1
## Delaware          5.9     238 15.8
## Florida          15.4     335 31.9
## Georgia          17.4     211 25.8
## Hawaii            5.3      46 20.2
## Idaho             2.6     120 14.2
## Illinois         10.4     249 24.0
## Indiana           7.2     113 21.0
## Iowa              2.2      56 11.3
## Kansas            6.0     115 18.0
## Kentucky          9.7     109 16.3
## Louisiana        15.4     249 22.2
## Maine             2.1      83  7.8
## Maryland         11.3     300 27.8
## Massachusetts     4.4     149 16.3
## Michigan         12.1     255 35.1
## Minnesota         2.7      72 14.9
## Mississippi      16.1     259 17.1
## Missouri          9.0     178 28.2
## Montana           6.0     109 16.4
## Nebraska          4.3     102 16.5
## Nevada           12.2     252 46.0
## New Hampshire     2.1      57  9.5
## New Jersey        7.4     159 18.8
## New Mexico       11.4     285 32.1
## New York         11.1     254 26.1
## North Carolina   13.0     337 16.1
## North Dakota      0.8      45  7.3
## Ohio              7.3     120 21.4
## Oklahoma          6.6     151 20.0
## Oregon            4.9     159 29.3
## Pennsylvania      6.3     106 14.9
## Rhode Island      3.4     174  8.3
## South Carolina   14.4     279 22.5
## South Dakota      3.8      86 12.8
## Tennessee        13.2     188 26.9
## Texas            12.7     201 25.5
## Utah              3.2     120 22.9
## Vermont           2.2      48 11.2
## Virginia          8.5     156 20.7
## Washington        4.0     145 26.2
## West Virginia     5.7      81  9.3
## Wisconsin         2.6      53 10.8
```

```
## Wyoming           6.8     161 15.6
```

```r
Urban <- USArrests[, "UrbanPop", drop = FALSE]
Urban
```

```
##                UrbanPop
## Alabama              58
## Alaska               48
## Arizona              80
## Arkansas             50
## California           91
## Colorado             78
## Connecticut          77
## Delaware             72
## Florida              80
## Georgia              60
## Hawaii               83
## Idaho                54
## Illinois             83
## Indiana              65
## Iowa                 57
## Kansas               66
## Kentucky             52
## Louisiana            66
## Maine                51
## Maryland             67
## Massachusetts        85
## Michigan             74
## Minnesota            66
## Mississippi          44
## Missouri             70
## Montana              53
## Nebraska             62
## Nevada               81
## New Hampshire        56
## New Jersey           89
## New Mexico           70
## New York             86
## North Carolina       45
## North Dakota         44
## Ohio                 75
## Oklahoma             68
## Oregon               67
## Pennsylvania         72
## Rhode Island         87
## South Carolina       48
## South Dakota         45
## Tennessee            59
## Texas                80
## Utah                 80
## Vermont              32
## Virginia             63
## Washington           73
## West Virginia        39
```

```
## Wisconsin             66
## Wyoming               60
```

f. In Urban, rename the column UrbanPop to "UrbanPercent".

```
colnames(Urban)[colnames(Urban) == "UrbanPop"] <- "UrbanPercent"
colnames(Urban)
```

```
## [1] "UrbanPercent"
```

g. Find the median of each column in USArrests2.

```
column_medians <- apply(USArrests2, 2, median)
column_medians
```

```
##   Murder Assault    Rape
##     7.25  159.00   20.10
```

h. In USArrests2, add a column called "Total", which gives the total sum of Murder, Assault, and Rape.

```
USArrests2$Total <- rowSums(USArrests2)
USArrests2
```

```
##                  Murder Assault Rape Total
## Alabama            13.2     236 21.2 270.4
## Alaska             10.0     263 44.5 317.5
## Arizona             8.1     294 31.0 333.1
## Arkansas            8.8     190 19.5 218.3
## California          9.0     276 40.6 325.6
## Colorado            7.9     204 38.7 250.6
## Connecticut         3.3     110 11.1 124.4
## Delaware            5.9     238 15.8 259.7
## Florida            15.4     335 31.9 382.3
## Georgia            17.4     211 25.8 254.2
## Hawaii              5.3      46 20.2  71.5
## Idaho               2.6     120 14.2 136.8
## Illinois           10.4     249 24.0 283.4
## Indiana             7.2     113 21.0 141.2
## Iowa                2.2      56 11.3  69.5
## Kansas              6.0     115 18.0 139.0
## Kentucky            9.7     109 16.3 135.0
## Louisiana          15.4     249 22.2 286.6
## Maine               2.1      83  7.8  92.9
## Maryland           11.3     300 27.8 339.1
## Massachusetts       4.4     149 16.3 169.7
## Michigan           12.1     255 35.1 302.2
## Minnesota           2.7      72 14.9  89.6
## Mississippi        16.1     259 17.1 292.2
## Missouri            9.0     178 28.2 215.2
## Montana             6.0     109 16.4 131.4
## Nebraska            4.3     102 16.5 122.8
```

```
## Nevada           12.2     252 46.0 310.2
## New Hampshire     2.1      57  9.5  68.6
## New Jersey        7.4     159 18.8 185.2
## New Mexico       11.4     285 32.1 328.5
## New York         11.1     254 26.1 291.2
## North Carolina   13.0     337 16.1 366.1
## North Dakota      0.8      45  7.3  53.1
## Ohio              7.3     120 21.4 148.7
## Oklahoma          6.6     151 20.0 177.6
## Oregon            4.9     159 29.3 193.2
## Pennsylvania      6.3     106 14.9 127.2
## Rhode Island      3.4     174  8.3 185.7
## South Carolina   14.4     279 22.5 315.9
## South Dakota      3.8      86 12.8 102.6
## Tennessee        13.2     188 26.9 228.1
## Texas            12.7     201 25.5 239.2
## Utah              3.2     120 22.9 146.1
## Vermont           2.2      48 11.2  61.4
## Virginia          8.5     156 20.7 185.2
## Washington        4.0     145 26.2 175.2
## West Virginia     5.7      81  9.3  96.0
## Wisconsin         2.6      53 10.8  66.4
## Wyoming           6.8     161 15.6 183.4
```

i. Using the order() function, list the states in USArrests2 according to Murder in decreasing order.

```
ordered_states <- rownames(USArrests2)[order(USArrests2$Murder, decreasing = TRUE)]
ordered_states
```

```
##  [1] "Georgia"        "Mississippi"    "Florida"        "Louisiana"
##  [5] "South Carolina" "Alabama"        "Tennessee"      "North Carolina"
##  [9] "Texas"          "Nevada"         "Michigan"       "New Mexico"
## [13] "Maryland"       "New York"       "Illinois"       "Alaska"
## [17] "Kentucky"       "California"     "Missouri"       "Arkansas"
## [21] "Virginia"       "Arizona"        "Colorado"       "New Jersey"
## [25] "Ohio"           "Indiana"        "Wyoming"        "Oklahoma"
## [29] "Pennsylvania"   "Kansas"         "Montana"        "Delaware"
## [33] "West Virginia"  "Hawaii"         "Oregon"         "Massachusetts"
## [37] "Nebraska"       "Washington"     "South Dakota"   "Rhode Island"
## [41] "Connecticut"    "Utah"           "Minnesota"      "Idaho"
## [45] "Wisconsin"      "Iowa"           "Vermont"        "Maine"
## [49] "New Hampshire"  "North Dakota"
```

j. Show the subset of the data frame USArrests2 containing all states having Murder $<$ 10, Assault $<$ 100, and Rape $<$ 10.

```
subset_USArrests2 <- USArrests2[USArrests2$Murder < 10 & USArrests2$Assault < 100 & USArrests2$Rape < 1
subset_USArrests2
```

```
##               Murder Assault Rape Total
## Maine            2.1      83  7.8  92.9
## New Hampshire    2.1      57  9.5  68.6
## North Dakota     0.8      45  7.3  53.1
## West Virginia    5.7      81  9.3  96.0
```

k. Using data frame USArrests, find the across-state average murder rate (Murder) in regions where the percentage of the population living in urban areas (UrbanPop) exceeds 77%. Compare this with the average murder rate where the urban area population is less than 50%.

```
high_urban_avg_murder <- mean(USArrests$Murder[USArrests$UrbanPop > 77])
low_urban_avg_murder <- mean(USArrests$Murder[USArrests$UrbanPop < 50])
cat("Average Murder Rate (UrbanPop > 77%):", high_urban_avg_murder, "\n")
```

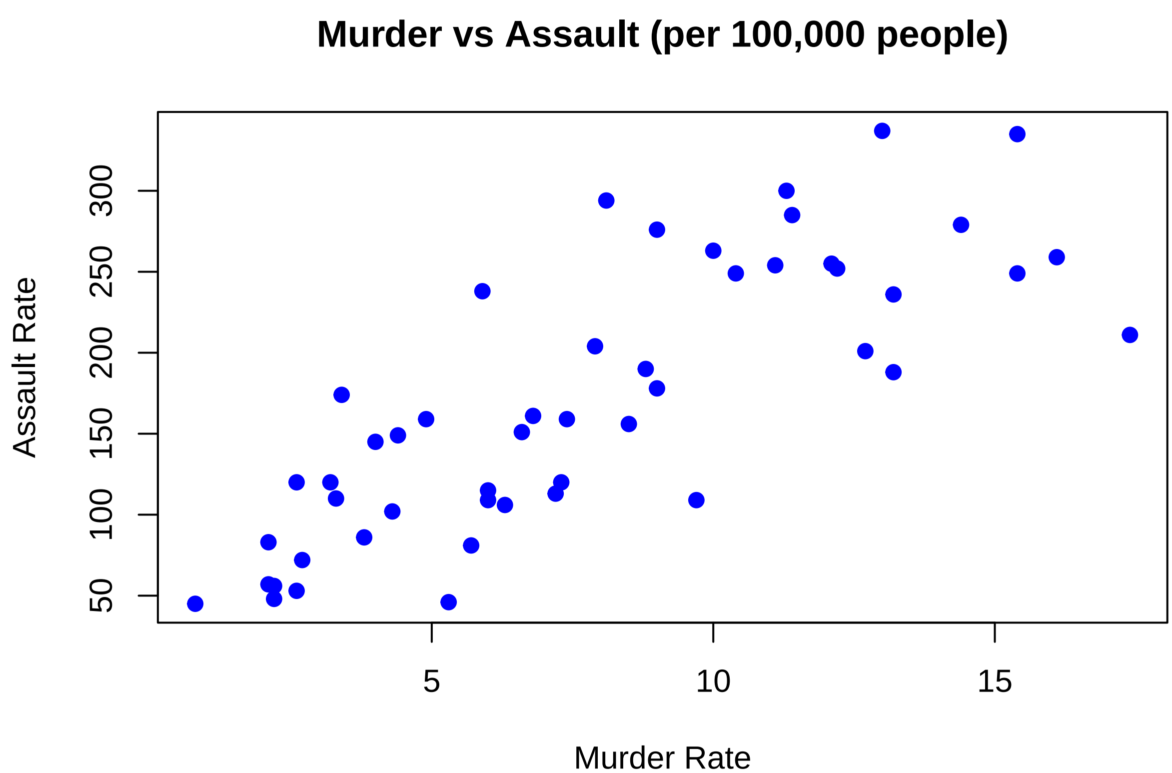
```
## Average Murder Rate (UrbanPop > 77%): 8.5
```

```
cat("Average Murder Rate (UrbanPop < 50%):", low_urban_avg_murder, "\n")
```

```
## Average Murder Rate (UrbanPop < 50%): 8.25
```

l. Use the plot() function to show Murder vs Assault. Identify and explain any relationships you see.

```
plot(USArrests$Murder, USArrests$Assault,
     main = "Murder vs Assault (per 100,000 people)",
     xlab = "Murder Rate",
     ylab = "Assault Rate",
     pch = 19, col = "blue")
```



```r
# States with higher rates of murder also tend to have higher rates of assault.
# More states fall toward the lower end of the murder & assault axes.
```

Problem 3

Install the package COUNT. Then run the following:

```r
library(COUNT)
```

```
## Warning: package 'COUNT' was built under R version 4.3.3

## Loading required package: msme

## Warning: package 'msme' was built under R version 4.3.3

## Loading required package: MASS

## Loading required package: lattice

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.3.3
```

```r
data(fishing)
pois.glm <- glm(totabund ~ meandepth, data = fishing, family = poisson)
```

a. Verify that the object pois.glm is a list.

```r
typeof(pois.glm)
```

```
## [1] "list"
```

b. Find the number of components in pois.glm and display their names.

```r
length(names(pois.glm))
```

```
## [1] 30
```

```r
names(pois.glm)
```

```
##  [1] "coefficients"      "residuals"         "fitted.values"
##  [4] "effects"           "R"                 "rank"
##  [7] "qr"                "family"            "linear.predictors"
## [10] "deviance"          "aic"               "null.deviance"
## [13] "iter"              "weights"           "prior.weights"
## [16] "df.residual"       "df.null"           "y"
## [19] "converged"         "boundary"          "model"
## [22] "call"              "formula"           "terms"
## [25] "data"              "offset"            "control"
## [28] "method"            "contrasts"         "xlevels"
```

c. Among all the components of pois.glm, is there any component which is a list itself?

```r
typeof(pois.glm$qr)
```

```
## [1] "list"
```

```r
# qr is an example of a component which is a list itself
```

d. Find out the number of elements contained in the component residuals, as well as the maximum and minimum values.

```r
pois.glm$residuals
```

```
##            1            2            3            4            5            6 
## -0.836114413 -0.651944054 -0.915635101 -0.090993164 -0.606335210  0.653698796 
##            7            8            9           10           11           12 
## -0.153411594  0.626146106  0.508364370  0.780219249 -0.189841858  0.303981513 
##           13           14           15           16           17           18 
## -0.124375070 -0.559010969 -0.851258198 -0.314923808 -0.865200098 -0.333871842 
##           19           20           21           22           23           24 
## -0.738133994 -0.098093594 -0.560578496 -0.452174832  2.405158939 -0.722283112 
##           25           26           27           28           29           30 
## -0.073898914  0.779702619  0.908411746 -0.409461368  0.130650880 -0.591511730 
##           31           32           33           34           35           36 
##  0.655127940 -0.454488030  0.194391331  0.598612581 -0.778385975 -0.342925463 
##           37           38           39           40           41           42 
##  0.313938092 -0.254223249 -0.197966896  2.321485266 -0.399583375 -0.206353177 
##           43           44           45           46           47           48 
## -0.732377205  0.638690566 -0.488770254 -0.911862542  2.406513504  1.313876114 
##           49           50           51           52           53           54 
## -0.231475486  0.843493094  1.258251026 -0.401391780 -0.008458713 -0.886146281 
##           55           56           57           58           59           60 
##  0.158361429 -0.495030232  0.426991935  1.571223611 -0.255653487  0.216361179 
##           61           62           63           64           65           66 
##  0.875904849 -0.817014467  0.631935935  0.353667907 -0.313759944 -0.255130253 
##           67           68           69           70           71           72 
## -0.281199541 -0.785400657  1.346084116 -0.977865977 -0.534598627  0.060845906 
##           73           74           75           76           77           78 
##  0.214461460 -0.465801881 -0.319447100 -0.369975461 -0.491008591  0.098352081 
##           79           80           81           82           83           84 
## -0.467775839  1.322927717  0.078562827 -0.796579045  0.595958708 -0.148067841 
##           85           86           87           88           89           90 
## -0.250136462  1.008944786  0.425564592 -0.175112405 -0.457256995 -0.451750456 
##           91           92           93           94           95           96 
## -0.122272226  0.803129150  0.458195414  0.197223139  0.691173597  0.430730311 
##           97           98           99          100          101          102 
##  0.255775671 -0.532886456  0.494947870  0.693681527 -0.116594818  0.046064690 
##          103          104          105          106          107          108 
##  0.385913758  0.310837694  0.332541368 -0.451910696  0.704767520  0.091718187 
##          109          110          111          112          113          114 
## -0.486755447 -0.332412738 -0.332880680  0.083438787 -0.933944771  0.385994196 
##          115          116          117          118          119          120
```

```
## -0.640813852  0.338204527 -0.438437493  0.001063848 -0.721853840  1.132374646
##          121          122          123          124          125          126
## -0.546948031 -0.814723234 -0.372456219 -0.389282343 -0.052384367  0.192473390
##          127          128          129          130          131          132
## -0.514333574 -0.962262009  0.058665630 -0.084735167 -0.394025110 -0.920716833
##          133          134          135          136          137          138
## -0.528025010 -0.731859279 -0.398514684 -0.737546578 -0.068574023  1.817290540
##          139          140          141          142          143          144
## -0.675986122 -0.810872619 -0.891859005 -0.729647512 -0.449658269 -0.641826206
##          145          146          147
## -0.337085210  0.049615083 -0.189318024
## attr(,"label")
## [1] "total number fish/site"
## attr(,"class")
## [1] "labelled" "integer"
## attr(,"format")
## [1] "%9.0g"
```

```r
length(pois.glm$residuals)
```

```
## [1] 147
```

```r
max(pois.glm$residuals)
```

```
## [1] 2.406514
```

```r
min(pois.glm$residuals)
```

```
## [1] -0.977866
```

e. Add a new component named "extra" to pois.glm, containing 10 random numbers following a Poisson(2) distribution.

```r
new_component <- rpois(10, lambda = 2)
pois.glm$extra <- new_component
```