1.

ask <- function() {

num <- readline("enter a positive number")

num <- as.numeric(num)

return(1:num)

}

|  |
| --- |
| > ask()  enter a positive number24  [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  [24] 24 |
|  |
| |  | | --- | | > | |

2.

ask <- function() {

num <- readline("enter a number")

num <- as.numeric(num)

return(seq(2, num, by = 2))

}

|  |
| --- |
| > ask()  enter a number18  [1] 2 4 6 8 10 12 14 16 18 |
|  |
| |  | | --- | | > | |

3.

ask <- function() {

num <- readline("enter a number")

num <- as.numeric(num)

ans <- 1 + 1:num

return(ans)

}

|  |
| --- |
| > ask()  enter a number7  [1] 2 3 4 5 6 7 8 |
|  |
| |  | | --- | | > | |

4.

ask <- function() {

num <- readline("The total of the bill was: ")

num <- as.numeric(num)

ans <- round(num \* .15, 2)

return(ans)

}

|  |
| --- |
| > ask()  The total of the bill was: 28.57  [1] 4.29 |
|  |
| |  | | --- | | > | |

5. ask <- function() {

num <- readline("Enter the year of your birth: ")

num <- as.numeric(num)

ans <- 2016 - num

paste("You turn", ans, "this year")

}

|  |
| --- |
| Enter the year of your birth: 1988  [1] "You turn 28 this year" |
|  |
| |  | | --- | | > | |

6.

ask <- function() {

num <- readline("Enter a number: ")

num <- as.numeric(num)

if (num == round(num)) print("This is a whole number")

if (num != round(num)) print("This is not whole number")

}

|  |
| --- |
| > ask()  Enter a number: 67.56  [1] "This is not whole number" |
|  |
| |  | | --- | | > | |

7.

ask <- function() {

num <- readline("Enter a number: ")

num <- as.numeric(num)

ans <- prod(num:1)

print(ans)

}

|  |
| --- |
| > ask()  Enter a number: 4  [1] 24  > ask()  Enter a number: 3  [1] 6 |
|  |
| |  | | --- | | > | |

8. ask <- function() {

num <- readline("Enter a number between 1 and 10: ")

num <- as.numeric(num)

if (num > 3) paste(num, "th", sept = "")

if (num == 1) paste(num, "st", sept = "")

if (num == 2) paste(num, "nd", sept = "")

if (num == 3) paste(num, "rd", sept = "")

> ask()

Enter a number between 1 and 10: 3

[1] "3 rd "

9.

ask <- function() {

num <- readline("Enter a number between 1 and 10: ")

finaldigit <- as.numeric(substr(num, nchar(num), nchar(num)))

if (finaldigit > 3) ans <- paste(num, "th", sept = "")

if (finaldigit == 1) ans <- paste(num, "st", sept = "")

if (finaldigit == 2) ans <- paste(num, "nd", sept = "")

if (finaldigit == 3) ans <- paste(num, "rd", sept = "")

return(ans)

}

|  |
| --- |
| > ask()  Enter a number between 1 and 10: 4.56  [1] "4.56 th " |
|  |
| |  | | --- | | > | |

10.

ask <- function() {

abrv <- (readline("Enter a state abbreviation: "))

ans <-abrv %in% state.abb

return(ans)

}

|  |
| --- |
| > ask()  Enter a state abbreviation: SD  [1] TRUE  > ask()  Enter a state abbreviation: mocha  [1] FALSE |
|  |
| |  | | --- | | > | |

11.

ask <- function() {

abrv <- (readline("Enter a state abbreviation: "))

abrv <- toupper(abrv)

ans <-abrv %in% state.abb

return(ans)

}

|  |
| --- |
| > ask()  Enter a state abbreviation: MD  [1] TRUE  > ask()  Enter a state abbreviation: md  [1] TRUE  > ask()  Enter a state abbreviation: Md  [1] TRUE |
|  |
| |  | | --- | | > | |

12.

ask <- function() {

abrv <- (readline("Enter a state abbreviation: "))

abrv <- toupper(abrv)

ans <- state.name[abrv == state.abb]

if (length(ans) == 0) ans = "Not a proper State abbreviation"

return(ans)

}

|  |
| --- |
| > ask()  Enter a state abbreviation: SD  [1] "South Dakota"  > ask()  Enter a state abbreviation: Peanut  [1] "Not a proper State abbreviation" |
|  |
| |  | | --- | | > | |

13.

ask <- function() {

state <- (readline("Enter a state name or abbreviation: "))

if (any(state == state.name)) {

ans <- state.abb[state == state.name]

} else if(any(state == state.abb)) {

ans <- state.name[state == state.abb]

} else {

ans <- "Neither a proper state name nor an official abbreviation"

}

return(ans)

}

|  |
| --- |
| > ask()  Enter a state name or abbreviation: Wyoming  [1] "WY"  > ask()  Enter a state name or abbreviation: SD  [1] "South Dakota"  > ask()  Enter a state name or abbreviation: Fandango  [1] "Neither a proper state name nor an official abbreviation" |
|  |
| |  | | --- | | > | |

14. sort(state.x77[,1] \* state.x77[,6], decreasing = TRUE)

California New York Pennsylvania Illinois Texas

1326994.8 952605.2 595372.0 588962.2 580033.8

Ohio Michigan Florida New Jersey Massachusetts

571102.0 481060.8 435370.2 384982.5 340119.0

Indiana Wisconsin Virginia Missouri Washington

281057.7 250100.5 238091.8 232629.6 225996.5

Minnesota Maryland North Carolina Georgia Tennessee

225849.6 215580.6 209478.5 200198.6 174431.4

Connecticut Iowa Colorado Louisiana Alabama

173600.0 168799.0 162369.9 160613.2 149299.5

Oklahoma Oregon Kansas Kentucky Arizona

140094.0 137040.0 136572.0 130399.5 128517.2

South Carolina Mississippi Nebraska Arkansas Utah

106444.8 95981.0 91559.2 84189.0 80961.9

West Virginia New Mexico Maine Hawaii Idaho

74838.4 63148.8 57872.6 53729.2 48373.5

New Hampshire Montana Rhode Island Nevada South Dakota

46771.2 44163.2 43198.4 38468.0 36297.3

North Dakota Delaware Vermont Alaska Wyoming

32041.1 31613.4 26951.2 24345.5 23650.4

15. sort(state.x77[,6] / state.x77[,8], decreasing = TRUE)

Rhode Island Delaware Connecticut Hawaii Massachusetts

0.0442326025 0.0275479314 0.0115178939 0.0096342412 0.0074750831

New Jersey New Hampshire Vermont Maryland Maine

0.0069804547 0.0063808574 0.0061616489 0.0052876352 0.0017690815

West Virginia Indiana Ohio South Carolina Virginia

0.0017282925 0.0014654957 0.0012983527 0.0012506203 0.0012016088

Pennsylvania New York Iowa Tennessee Wisconsin

0.0011163991 0.0011017959 0.0010546826 0.0010114208 0.0010006610

Florida Kentucky Washington Illinois Louisiana

0.0009724533 0.0009709962 0.0009538831 0.0009435316 0.0009392388

Michigan Mississippi Utah Alabama North Carolina

0.0009292993 0.0008668809 0.0008197720 0.0008144671 0.0007889668

Nebraska Arkansas Oklahoma Kansas Minnesota

0.0007753357 0.0007681201 0.0007501963 0.0007323902 0.0007264564

North Dakota Idaho Missouri South Dakota Georgia

0.0007261126 0.0007196681 0.0007072976 0.0007017313 0.0006991201

Wyoming Oregon Colorado Nevada Arizona

0.0006470994 0.0006238044 0.0006158086 0.0005933260 0.0005122689

New Mexico Montana California Texas Alaska

0.0004546503 0.0004066297 0.0004003556 0.0001808235 0.0001177546

16.

ask <- function() {

abb <- readline("Enter a state abbreviation:")

index <- which(abb == state.abb)

ans <- state.x77[index,1] \* state.x77[index, 6]

return(ans)

}

|  |
| --- |
| > ask()  Enter a state abbreviation:MD  [1] 215580.6  > ask()  Enter a state abbreviation:SD  [1] 36297.3 |
|  |
| |  | | --- | | > | |

17.

|  |
| --- |
| > rownames(state.x77)[order(state.x77 [,"Illiteracy"], decreasing=TRUE)]  [1] "Louisiana" "Mississippi" "South Carolina" "New Mexico"  [5] "Texas" "Alabama" "Georgia" "Arkansas"  [9] "Hawaii" "Arizona" "North Carolina" "Tennessee"  [13] "Kentucky" "Alaska" "New York" "Virginia"  [17] "West Virginia" "Florida" "Rhode Island" "California"  [21] "Connecticut" "Massachusetts" "New Jersey" "Oklahoma"  [25] "Pennsylvania" "Delaware" "Illinois" "Maryland"  [29] "Michigan" "Missouri" "North Dakota" "Ohio"  [33] "Colorado" "Indiana" "Maine" "New Hampshire"  [37] "Wisconsin" "Idaho" "Kansas" "Minnesota"  [41] "Montana" "Nebraska" "Oregon" "Utah"  [45] "Vermont" "Washington" "Wyoming" "Iowa"  [49] "Nevada" "South Dakota"  18. |
|  |
| |  | | --- | |  | |

> index <- order(state.x77[,3], decreasing=TRUE)[1:10]

> Illiteracy <- rownames(state.x77)[index]

> index <- order(state.x77[,4])[1:10]

> Life\_expectancy <- rownames(state.x77)[index]

> index <- order(state.x77[,5], decreasing=TRUE)[1:10]

> Murder <- rownames(state.x77)[index]

> index <- order(state.x77[,6])[1:10]

> HS\_grad <- rownames(state.x77)[index]

> data.frame(Illiteracy, Life\_expectancy, Murder, HS\_grad, stringsAsFactors = FALSE)

Illiteracy Life\_expectancy Murder HS\_grad

1 Louisiana South Carolina Alabama South Carolina

2 Mississippi Mississippi Georgia Kentucky

3 South Carolina Georgia Louisiana North Carolina

4 New Mexico Louisiana Mississippi Arkansas

5 Texas Nevada Texas Georgia

6 Alabama Alabama South Carolina Mississippi

7 Georgia North Carolina Nevada Alabama

8 Arkansas Alaska Alaska West Virginia

9 Hawaii West Virginia Michigan Tennessee

10 Arizona Delaware North Carolina Louisiana

19.

> ans <- data.frame(Illiteracy, Life\_expectancy, Murder, HS\_grad, stringsAsFactors = FALSE)

> modifiedans<- ans$Illiteracy[which(ans$Illiteracy %in% ans$Murde)]

> modiefiedans

Error: object 'modiefiedans' not found

> modifiedans

[1] "Louisiana" "Mississippi" "South Carolina" "Texas"

[5] "Alabama" "Georgia"

20.

> ask <- function() {

+ cl <- readline("enter class: ")

+

+ tiSubset <- ti[ti$Class == cl,]

+ deaths <- sum(tiSubset$Freq[tiSubset$Survived == "No"])

+ survived <- sum(tiSubset$Freq[tiSubset$Survived == "Yes"])

+ print(paste("Total deaths", deaths))

+ print(paste("Total survived", survived))

+ }

> ask()

enter class: 1st

[1] "Total deaths 122"

[1] "Total survived 203"

> ask()

enter class: 2nd

[1] "Total deaths 167"

[1] "Total survived 118"

> ask()

enter class: 3rd

[1] "Total deaths 528"

[1] "Total survived 178"

> ask()

enter class: Crew

[1] "Total deaths 673"

[1] "Total survived 212"

21.

|  |
| --- |
| + cl <- readline("enter class: ")  +  + tiSubset <- ti[ti$Class == cl,]  + write.csv(tiSubset, "titanicdata.csv")  +  + print(paste("Saved: ", getwd(), "/titanicdata.csv", sep=""))  + }  > ask()  enter class: 1st  [1] "Saved: C:/Users/Thomas/Documents/titanicdata.csv" |
|  |
| |  | | --- | |  | |

22.

> ask <- function() {

+ wd <- readline("Enter the directory that contains Titanic Data: ")

+

+ setwd(wd)

+ tiInput <- read.csv("titanicdata.csv", as.is=TRUE)

+

+ return(tiInput)

+ }

> ask()

Enter the directory that contains Titanic Data: C:/Users/Thomas/Documents

X Class Sex Age Survived Freq

1 1 1st Male Child No 0

2 5 1st Female Child No 0

3 9 1st Male Adult No 118

4 13 1st Female Adult No 4

5 17 1st Male Child Yes 5

6 21 1st Female Child Yes 1

7 25 1st Male Adult Yes 57

8 29 1st Female Adult Yes 140

23.

|  |
| --- |
| > ask <- function() {  + wd <- readline("Enter the directory that contains Titanic Data: ")  +  + setwd(wd)  + tiInput <- read.csv("titanicdata.csv", as.is=TRUE)  +  + print(paste("You saved data on class: ", tiInput$Class[1]))  + }  > ask()  Enter the directory that contains Titanic Data: C:/Users/Thomas/Documents  [1] "You saved data on class: 1st" |
|  |
| |  | | --- | |  | |

24.

|  |
| --- |
| > (WorldPhones[7,] - WorldPhones[1,]) / WorldPhones[1,]  N.Amer Europe Asia S.Amer Oceania Africa Mid.Amer  0.7377609 1.0011588 2.1477747 0.8391185 0.9586877 21.5280899 0.9387387 |
|  |
| |  | | --- | | > | |

25.

ask <- function() {

year <- readline("Enter the year (1951-1961) you want phone data for: ")

allyears <- as.numeric(rownames(WorldPhones))

index <- which(allyears == year)

index <- index[length(index)]

ans <- WorldPhones[index, 2] \* 1000

return(ans)

}

> ask()

Enter the year (1951-1961) you want phone data for: 1958

[1] 35218000