**R code – Unexecuted**

**------------#Step1: Create a function readStates--------------------**

readStates <- read.csv(url("http://www2.census.gov/programs-surveys/popest/tables/2010-2011/state/totals/nst-est2011-01.csv"))

head(readStates,-1)

View(readStates)

**-------------#Step2: Clean the dataframe---------------------**

readStates <- readStates[-1:-8,]

readStates <- readStates[-52:-58,]

rownames(readStates) <- NULL

readStates <- readStates[,-6:-10]

View(readStates)

colnames(readStates) <- c("stateName", "base2010", "base2011", "Jul2010", "Jul2011")

readStates$base2010 <- as.numeric(readStates$base2010)

readStates$base2011 <- as.numeric(readStates$base2011)

readStates$Jul2010 <- as.numeric(readStates$Jul2010)

readStates$Jul2011 <- as.numeric(readStates$Jul2011)

View(readStates)

**--------------#Step3: Store and Explore dataset------------------------**

States <- function(){

#Step1: Create a function readStates

readStates <- read.csv(url("http://www2.census.gov/programs-surveys/popest/tables/2010-2011/state/totals/nst-est2011-01.csv"))

#Step2: Clean the dataframe

readStates <- readStates[-1:-8,]

readStates <- readStates[-52:-58,]

rownames(readStates) <- NULL

readStates <- readStates[,-6:-10]

colnames(readStates) <- c("stateName", "base2010", "base2011", "Jul2010", "Jul2011")

readStates$base2010 <- as.numeric(readStates$base2010)

readStates$base2011 <- as.numeric(readStates$base2011)

readStates$Jul2010 <- as.numeric(readStates$Jul2010)

readStates$Jul2011 <- as.numeric(readStates$Jul2011)

return (readStates)

}

dfStates <- States()

View(dfStates)

mean(dfStates$Jul2011)

**-----------------------------------------------------------------------------------**

**-------------------#Step4: Find the state with the Highest Population----------------------**

max(dfStates$Jul2011)

dfStates[order(dfStates$Jul2011),]

**------------------------#Step5: Explore distribution of the states-------------------------------**

a <- dfStates$Jul2011

b <- mean(dfStates$Jul2011)

distStates <- function(a,b){

count = 0

for (x in a){

if(x < b)

count = count+1

}

return(count / length(a))

}

print(distStates(a,b))

**R code – Executed**

#Step1: Create a function readStates

#Step1: Create a function readStates

> readStates <- read.csv(url("http://www2.census.gov/programs-surveys/popest/tables/2010-2011/state/totals/nst-est2011-01.csv"))

> head(readStates,-1)

stateName base2010 base2011 Jul2010

1 .District of Columbia 45 45 46

2 .Florida 13 13 13

3 .Georgia 56 56 57

4 .Hawaii 5 5 5

5 .Idaho 6 6 6

6 .Illinois 12 12 12

7 .Indiana 42 42 43

8 .Iowa 22 22 23

9 .Kansas 18 18 18

10 .Kentucky 29 29 30

11 .Louisiana 30 30 31

12 .Maine 4 4 4

13 .Maryland 36 36 37

14 .Massachusetts 43 43 44

15 .Michigan 57 57 58

16 .Minnesota 34 34 35

17 .Mississippi 20 20 20

18 .Missouri 37 37 38

19 .Montana 58 58 59

20 .Nebraska 7 7 7

21 .Nevada 16 16 16

22 .New Hampshire 3 3 3

23 .New Jersey 52 52 53

24 .New Mexico 15 15 15

25 .New York 14 14 14

26 .North Carolina 55 55 56

27 .North Dakota 48 48 49

28 .Ohio 9 9 9

29 .Oklahoma 25 25 26

30 .Oregon 26 26 27

31 .Pennsylvania 11 11 11

32 .Rhode Island 2 2 2

33 .South Carolina 31 31 32

34 .South Dakota 53 53 54

35 .Tennessee 40 40 41

36 .Texas 21 21 22

37 .Utah 17 17 17

38 .Vermont 46 46 47

39 .Virginia 51 51 52

40 .Washington 44 44 45

41 .West Virginia 8 8 8

42 .Wisconsin 35 35 36

Jul2011

1 46

2 13

3 56

4 5

5 6

6 12

7 43

8 23

9 18

10 30

11 31

12 4

13 37

14 44

15 57

16 35

17 20

18 40

19 59

20 7

21 16

22 3

23 53

24 15

25 14

26 55

27 49

28 9

29 26

30 27

31 11

32 2

33 32

34 54

35 41

36 22

37 17

38 47

39 52

40 45

41 8

42 36

> View(readStates)

> #Step2: Clean the dataframe

> readStates <- readStates[-1:-8,]

> readStates <- readStates[-52:-58,]

> rownames(readStates) <- NULL

> readStates <- readStates[,-6:-10]

> View(readStates)

> colnames(readStates) <- c("stateName", "base2010", "base2011", "Jul2010", "Jul2011")

> readStates$base2010 <- as.numeric(readStates$base2010)

> readStates$base2011 <- as.numeric(readStates$base2011)

> readStates$Jul2010 <- as.numeric(readStates$Jul2010)

> readStates$Jul2011 <- as.numeric(readStates$Jul2011)

> View(readStates)

> #Step3: Store and Explore dataset

>

> States <- function(){

+ #Step1: Create a function readStates

+ readStates <- read.csv(url("http://www2.census.gov/programs-surveys/popest/tables/2010-2011/state/totals/nst-est2011-01.csv"))

+

+ #Step2: Clean the dataframe

+ readStates <- readStates[-1:-8,]

+ readStates <- readStates[-52:-58,]

+ rownames(readStates) <- NULL

+ readStates <- readStates[,-6:-10]

+

+ colnames(readStates) <- c("stateName", "base2010", "base2011", "Jul2010", "Jul2011")

+ readStates$base2010 <- as.numeric(readStates$base2010)

+ readStates$base2011 <- as.numeric(readStates$base2011)

+ readStates$Jul2010 <- as.numeric(readStates$Jul2010)

+ readStates$Jul2011 <- as.numeric(readStates$Jul2011)

+

+ return (readStates)

+

+ }

> dfStates <- States()

> View(dfStates)

>

> mean(dfStates$Jul2011)

[1] 30.37255

>

> #Step4: Find the state with the Highest Population

> max(dfStates$Jul2011)

[1] 59

> dfStates[order(dfStates$Jul2011),]

stateName base2010 base2011 Jul2010

40 .Rhode Island 2 2 2

30 .New Hampshire 3 3 3

20 .Maine 4 4 4

12 .Hawaii 5 5 5

13 .Idaho 6 6 6

28 .Nebraska 7 7 7

49 .West Virginia 8 8 8

36 .Ohio 9 9 9

39 .Pennsylvania 11 11 11

14 .Illinois 12 12 12

10 .Florida 13 13 13

33 .New York 14 14 14

32 .New Mexico 15 15 15

29 .Nevada 16 16 16

45 .Utah 17 17 17

17 .Kansas 18 18 18

4 .Arkansas 19 19 19

25 .Mississippi 20 20 20

44 .Texas 21 21 22

16 .Iowa 22 22 23

7 .Connecticut 23 23 24

37 .Oklahoma 25 25 26

38 .Oregon 26 26 27

5 .California 28 28 29

18 .Kentucky 29 29 30

19 .Louisiana 30 30 31

41 .South Carolina 31 31 32

1 .Alabama 32 32 33

6 .Colorado 33 33 34

24 .Minnesota 34 34 35

50 .Wisconsin 35 35 36

21 .Maryland 36 36 37

51 .Wyoming 39 39 40

26 .Missouri 37 37 38

43 .Tennessee 40 40 41

3 .Arizona 41 41 42

15 .Indiana 42 42 43

22 .Massachusetts 43 43 44

48 .Washington 44 44 45

9 .District of Columbia 45 45 46

46 .Vermont 46 46 47

35 .North Dakota 48 48 49

2 .Alaska 50 50 50

47 .Virginia 51 51 52

31 .New Jersey 52 52 53

42 .South Dakota 53 53 54

34 .North Carolina 55 55 56

11 .Georgia 56 56 57

23 .Michigan 57 57 58

8 .Delaware 54 54 55

27 .Montana 58 58 59

Jul2011

40 2

30 3

20 4

12 5

13 6

28 7

49 8

36 9

39 11

14 12

10 13

33 14

32 15

29 16

45 17

17 18

4 19

25 20

44 22

16 23

7 24

37 26

38 27

5 29

18 30

19 31

41 32

1 33

6 34

24 35

50 36

21 37

51 39

26 40

43 41

3 42

15 43

22 44

48 45

9 46

46 47

35 49

2 51

47 52

31 53

42 54

34 55

11 56

23 57

8 58

27 59

>

> #Step5: Explore distribution of the states

>

> a <- dfStates$Jul2011

> b <- mean(dfStates$Jul2011)

>

> distStates <- function(a,b){

+ count = 0

+ for (x in a){

+ if(x < b)

+ count = count+1

+ }

+ return(count / length(a))

+ }

>

> print(distStates(a,b))

[1] 0.4901961

>