

Madison Wozniak: Assignment 3

Assignment 3

You are asked to submit both the R Markdown file and its pdf output.

Q1. Write an if-else statement:

1. If the number is greater than 0 and less than 10, print: "This number is between 0 and 10"
2. If the number is greater than 10, print: "This number is greater than 10"
3. If the number is less than 0, print: "This number is a negative number"
4. Otherwise print: "This number is either 0 or 10"

```
x<-12
if(x>0){
  print("This number is between 0 and 10")
}else if(x<10){
  print("This number is between 0 and 10")
} else if(x>10){
  print("This number is greater than 10")
}else if(x<0){
  print("This number is a negative number")
}else{
  print("This number is either 0 or 10")
}
```

```
## [1] "This number is between 0 and 10"
```

Q2. Write a function that gets a vector as its input and returns the mean and the standard deviation of the vector using the formula below:

$$\left(\frac{\sum(x_i - \bar{x})^2}{n - 1}\right)^{1/2}$$

where \bar{x} is the mean and n is the length of the vector.

```
x<-c(1,5,4,9,1)
n<-length(x)
((sum(x-mean(x))^2)/(n-1))^(1/2)
```

```
## [1] 0
```

```
sd_by_hand<-function(x){
  x_bar<-mean(x)
  n<-length(x)
  sd_<-(((sum(x-x_bar^2))/(n-1))^(1/2))
  mean_<-mean(x)
  c(mean_,sd_)
```

```
}
sd_by_hand(c(1,5,4,9,1))
```

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

Q3.

- (a) Write a function that returns Euclidian Distance between two k-dimensional vectors:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_k - y_k)^2}$$

```
Euclidian_Distance<-function(x){
  sqrt((sum(x-y))^2)
}
x<-c(4,5,7,3,56)
y<-c(3,4,5,6,7)
Euclidian_Distance(x)
```

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

- (b) Write a function that will input the vectors x, y and p and will return the distance between two k-dimensional vectors:

$$d_p(x, y) = ((x_1 - y_1)^p + (x_2 - y_2)^p + \dots + (x_k - y_k)^p)^{1/p}$$

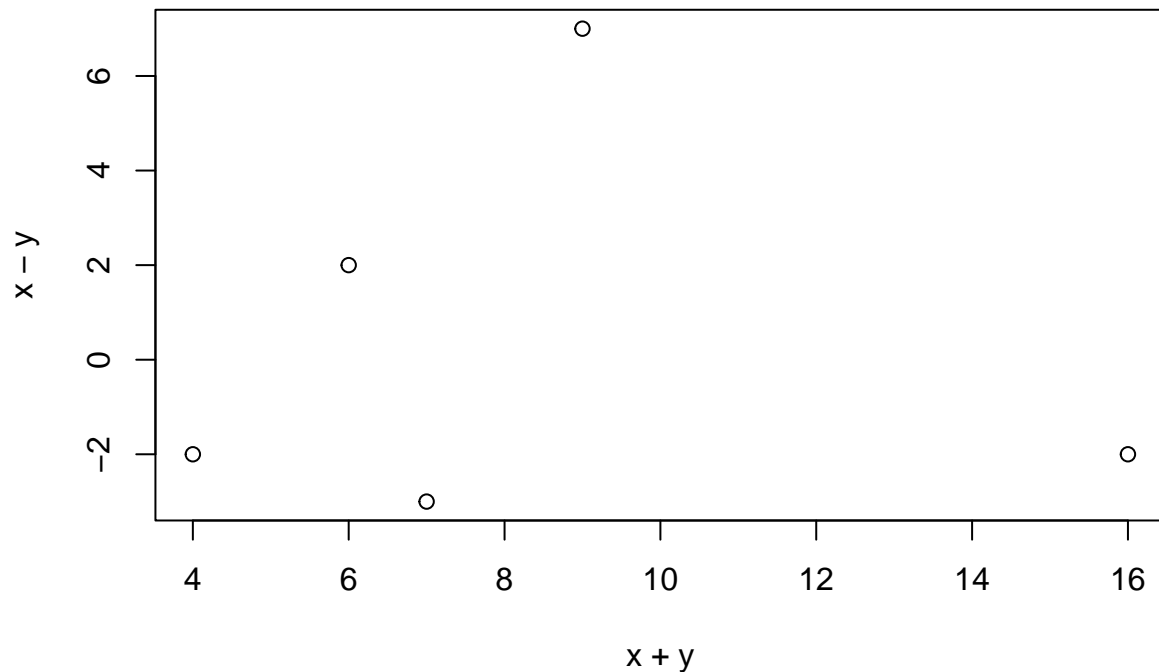
Pick the default value for p as 2.

```
distance_function<-function(x,y,p){
  p<-2
  x<-c(3,4,5,6,7)
  y<-c(1,2,3,4,5)
  x_k<-length(x)
  y_k<-length(y)
  (sum((x-y)^p))^(1/p)
}
distance_function(x,y,p)
```

```
## [1] 4.472136
```

- Q4. Create a function altman_plot that takes two arguments, x and y, and plots the difference against the sum.

```
altman_plot<-function(x,y){
  plot(x+y,x-y)
}
x<-c(1,2,4,7,8)
y<-c(3,5,2,9,1)
altman_plot(x,y)
```



Q5. Write a function `compute_s_n_2` that for any given n , computes the sum:

$$1^2 + 2^2 + 3^2 + 4^2 + \dots + (n-1)^2 + n^2$$

```
compute_s_n_2<-function(n){
  x<-1:n
  sum(x)
}
```

(a) Find `compute_s_n_2(30)`.

```
compute_s_n_2<-function(n){
  x<-1:n
  sum(x)
}
compute_s_n_2(30)
```

```
## [1] 465
```

(b) Confirm that the formula for this sum is $= \frac{n(n+1)(2n+1)}{6}$.

Q6. Which of the following built-in datasets is tidy (you can pick more than one):

- (a) BJsales
- (b) EuStockMarkets
- (c) DNase
- (d) Formaldehyde
- (e) Orange
- (f) UCBA admissions

```
head(BJsales)
```

```
## [1] 200.1 199.5 199.4 198.9 199.0 200.2
```

```
# not tidy
head(EuStockMarkets)
```

```
##           DAX      SMI      CAC      FTSE
## [1,] 1628.75 1678.1 1772.8 2443.6
## [2,] 1613.63 1688.5 1750.5 2460.2
## [3,] 1606.51 1678.6 1718.0 2448.2
## [4,] 1621.04 1684.1 1708.1 2470.4
## [5,] 1618.16 1686.6 1723.1 2484.7
## [6,] 1610.61 1671.6 1714.3 2466.8
```

```
#not tidy
head(DNase)
```

```
##      Run      conc density
## 1      1 0.04882812  0.017
## 2      1 0.04882812  0.018
## 3      1 0.19531250  0.121
## 4      1 0.19531250  0.124
## 5      1 0.39062500  0.206
## 6      1 0.39062500  0.215
```

```
#tidy
head(Formaldehyde)
```

```
##      carb optden
## 1      0.1  0.086
## 2      0.3  0.269
## 3      0.5  0.446
## 4      0.6  0.538
## 5      0.7  0.626
## 6      0.9  0.782
```

```
#tidy
head(Orange)
```

```
##      Tree  age circumference
## 1      1   118                30
## 2      1   484                58
## 3      1   664                87
## 4      1  1004               115
## 5      1  1231               120
## 6      1  1372               142
```

```
#tidy
head(UCBAdmissions)
```

```
## , , Dept = A
##
##           Gender
## Admit      Male Female
## Admitted  512      89
## Rejected  313      19
##
## , , Dept = B
##
##           Gender
## Admit      Male Female
## Admitted  353      17
## Rejected  207       8
```

```
##
## , , Dept = C
##
##           Gender
## Admit      Male Female
##   Admitted  120    202
##   Rejected  205    391
##
## , , Dept = D
##
##           Gender
## Admit      Male Female
##   Admitted  138    131
##   Rejected  279    244
##
## , , Dept = E
##
##           Gender
## Admit      Male Female
##   Admitted   53     94
##   Rejected  138    299
##
## , , Dept = F
##
##           Gender
## Admit      Male Female
##   Admitted   22     24
##   Rejected  351    317
```

#tidy

Q7. Load the dplyr package and the murders dataset.

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5    v purrr  0.3.4
## v tibble  3.1.4    v dplyr  1.0.7
## v tidyr   1.1.3    v stringr 1.4.0
## v readr   2.0.1    v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dslabs)
```

```
murders
```

```
##           state abb      region population total
## 1      Alabama  AL      South    4779736    135
## 2      Alaska   AK      West      710231     19
## 3      Arizona  AZ      West    6392017    232
## 4      Arkansas AR      South    2915918     93
## 5      California CA     West    37253956   1257
## 6      Colorado CO      West    5029196     65
## 7      Connecticut CT    Northeast  3574097     97
```

## 8	Delaware	DE	South	897934	38
## 9	District of Columbia	DC	South	601723	99
## 10	Florida	FL	South	19687653	669
## 11	Georgia	GA	South	9920000	376
## 12	Hawaii	HI	West	1360301	7
## 13	Idaho	ID	West	1567582	12
## 14	Illinois	IL	North Central	12830632	364
## 15	Indiana	IN	North Central	6483802	142
## 16	Iowa	IA	North Central	3046355	21
## 17	Kansas	KS	North Central	2853118	63
## 18	Kentucky	KY	South	4339367	116
## 19	Louisiana	LA	South	4533372	351
## 20	Maine	ME	Northeast	1328361	11
## 21	Maryland	MD	South	5773552	293
## 22	Massachusetts	MA	Northeast	6547629	118
## 23	Michigan	MI	North Central	9883640	413
## 24	Minnesota	MN	North Central	5303925	53
## 25	Mississippi	MS	South	2967297	120
## 26	Missouri	MO	North Central	5988927	321
## 27	Montana	MT	West	989415	12
## 28	Nebraska	NE	North Central	1826341	32
## 29	Nevada	NV	West	2700551	84
## 30	New Hampshire	NH	Northeast	1316470	5
## 31	New Jersey	NJ	Northeast	8791894	246
## 32	New Mexico	NM	West	2059179	67
## 33	New York	NY	Northeast	19378102	517
## 34	North Carolina	NC	South	9535483	286
## 35	North Dakota	ND	North Central	672591	4
## 36	Ohio	OH	North Central	11536504	310
## 37	Oklahoma	OK	South	3751351	111
## 38	Oregon	OR	West	3831074	36
## 39	Pennsylvania	PA	Northeast	12702379	457
## 40	Rhode Island	RI	Northeast	1052567	16
## 41	South Carolina	SC	South	4625364	207
## 42	South Dakota	SD	North Central	814180	8
## 43	Tennessee	TN	South	6346105	219
## 44	Texas	TX	South	25145561	805
## 45	Utah	UT	West	2763885	22
## 46	Vermont	VT	Northeast	625741	2
## 47	Virginia	VA	South	8001024	250
## 48	Washington	WA	West	6724540	93
## 49	West Virginia	WV	South	1852994	27
## 50	Wisconsin	WI	North Central	5686986	97
## 51	Wyoming	WY	West	563626	5

(a) By using dplyr's mutate function, add a new column:

```
population_in_millions = population / 10^6
murders_added<-mutate(murders,population_in_millions=population/(10^6))
murders_added
```

##	state	abb	region	population	total
## 1	Alabama	AL	South	4779736	135
## 2	Alaska	AK	West	710231	19
## 3	Arizona	AZ	West	6392017	232

## 4	Arkansas	AR	South	2915918	93
## 5	California	CA	West	37253956	1257
## 6	Colorado	CO	West	5029196	65
## 7	Connecticut	CT	Northeast	3574097	97
## 8	Delaware	DE	South	897934	38
## 9	District of Columbia	DC	South	601723	99
## 10	Florida	FL	South	19687653	669
## 11	Georgia	GA	South	9920000	376
## 12	Hawaii	HI	West	1360301	7
## 13	Idaho	ID	West	1567582	12
## 14	Illinois	IL	North Central	12830632	364
## 15	Indiana	IN	North Central	6483802	142
## 16	Iowa	IA	North Central	3046355	21
## 17	Kansas	KS	North Central	2853118	63
## 18	Kentucky	KY	South	4339367	116
## 19	Louisiana	LA	South	4533372	351
## 20	Maine	ME	Northeast	1328361	11
## 21	Maryland	MD	South	5773552	293
## 22	Massachusetts	MA	Northeast	6547629	118
## 23	Michigan	MI	North Central	9883640	413
## 24	Minnesota	MN	North Central	5303925	53
## 25	Mississippi	MS	South	2967297	120
## 26	Missouri	MO	North Central	5988927	321
## 27	Montana	MT	West	989415	12
## 28	Nebraska	NE	North Central	1826341	32
## 29	Nevada	NV	West	2700551	84
## 30	New Hampshire	NH	Northeast	1316470	5
## 31	New Jersey	NJ	Northeast	8791894	246
## 32	New Mexico	NM	West	2059179	67
## 33	New York	NY	Northeast	19378102	517
## 34	North Carolina	NC	South	9535483	286
## 35	North Dakota	ND	North Central	672591	4
## 36	Ohio	OH	North Central	11536504	310
## 37	Oklahoma	OK	South	3751351	111
## 38	Oregon	OR	West	3831074	36
## 39	Pennsylvania	PA	Northeast	12702379	457
## 40	Rhode Island	RI	Northeast	1052567	16
## 41	South Carolina	SC	South	4625364	207
## 42	South Dakota	SD	North Central	814180	8
## 43	Tennessee	TN	South	6346105	219
## 44	Texas	TX	South	25145561	805
## 45	Utah	UT	West	2763885	22
## 46	Vermont	VT	Northeast	625741	2
## 47	Virginia	VA	South	8001024	250
## 48	Washington	WA	West	6724540	93
## 49	West Virginia	WV	South	1852994	27
## 50	Wisconsin	WI	North Central	5686986	97
## 51	Wyoming	WY	West	563626	5
##	population_in_millions				
## 1	4.779736				
## 2	0.710231				
## 3	6.392017				
## 4	2.915918				
## 5	37.253956				

```
## 6          5.029196
## 7          3.574097
## 8          0.897934
## 9          0.601723
## 10         19.687653
## 11          9.920000
## 12          1.360301
## 13          1.567582
## 14         12.830632
## 15          6.483802
## 16          3.046355
## 17          2.853118
## 18          4.339367
## 19          4.533372
## 20          1.328361
## 21          5.773552
## 22          6.547629
## 23          9.883640
## 24          5.303925
## 25          2.967297
## 26          5.988927
## 27          0.989415
## 28          1.826341
## 29          2.700551
## 30          1.316470
## 31          8.791894
## 32          2.059179
## 33         19.378102
## 34          9.535483
## 35          0.672591
## 36         11.536504
## 37          3.751351
## 38          3.831074
## 39         12.702379
## 40          1.052567
## 41          4.625364
## 42          0.814180
## 43          6.346105
## 44         25.145561
## 45          2.763885
## 46          0.625741
## 47          8.001024
## 48          6.724540
## 49          1.852994
## 50          5.686986
## 51          0.563626
```

(b) If `rank(x)` gives you the ranks of `x` from lowest to highest, `rank(-x)` gives you the ranks from highest to lowest. Use the function `mutate` to add a column `rank` containing the rank, from highest to lowest murder rate. Make sure you redefine `murders` so we can keep using this variable.

```
murders_ranked<-murders%>%mutate(murd_rank=rank(-(total/(population)*100000)))
murders_ranked
```

```
##          state abb      region population total murd_rank
```


## 1	Alabama	AL	South	4779736	135	23
## 2	Alaska	AK	West	710231	19	27
## 3	Arizona	AZ	West	6392017	232	10
## 4	Arkansas	AR	South	2915918	93	17
## 5	California	CA	West	37253956	1257	14
## 6	Colorado	CO	West	5029196	65	38
## 7	Connecticut	CT	Northeast	3574097	97	25
## 8	Delaware	DE	South	897934	38	6
## 9	District of Columbia	DC	South	601723	99	1
## 10	Florida	FL	South	19687653	669	13
## 11	Georgia	GA	South	9920000	376	9
## 12	Hawaii	HI	West	1360301	7	49
## 13	Idaho	ID	West	1567582	12	46
## 14	Illinois	IL	North Central	12830632	364	22
## 15	Indiana	IN	North Central	6483802	142	31
## 16	Iowa	IA	North Central	3046355	21	47
## 17	Kansas	KS	North Central	2853118	63	30
## 18	Kentucky	KY	South	4339367	116	28
## 19	Louisiana	LA	South	4533372	351	2
## 20	Maine	ME	Northeast	1328361	11	44
## 21	Maryland	MD	South	5773552	293	4
## 22	Massachusetts	MA	Northeast	6547629	118	32
## 23	Michigan	MI	North Central	9883640	413	7
## 24	Minnesota	MN	North Central	5303925	53	40
## 25	Mississippi	MS	South	2967297	120	8
## 26	Missouri	MO	North Central	5988927	321	3
## 27	Montana	MT	West	989415	12	39
## 28	Nebraska	NE	North Central	1826341	32	33
## 29	Nevada	NV	West	2700551	84	19
## 30	New Hampshire	NH	Northeast	1316470	5	50
## 31	New Jersey	NJ	Northeast	8791894	246	24
## 32	New Mexico	NM	West	2059179	67	15
## 33	New York	NY	Northeast	19378102	517	29
## 34	North Carolina	NC	South	9535483	286	20
## 35	North Dakota	ND	North Central	672591	4	48
## 36	Ohio	OH	North Central	11536504	310	26
## 37	Oklahoma	OK	South	3751351	111	21
## 38	Oregon	OR	West	3831074	36	42
## 39	Pennsylvania	PA	Northeast	12702379	457	11
## 40	Rhode Island	RI	Northeast	1052567	16	35
## 41	South Carolina	SC	South	4625364	207	5
## 42	South Dakota	SD	North Central	814180	8	41
## 43	Tennessee	TN	South	6346105	219	12
## 44	Texas	TX	South	25145561	805	16
## 45	Utah	UT	West	2763885	22	45
## 46	Vermont	VT	Northeast	625741	2	51
## 47	Virginia	VA	South	8001024	250	18
## 48	Washington	WA	West	6724540	93	37
## 49	West Virginia	WV	South	1852994	27	36
## 50	Wisconsin	WI	North Central	5686986	97	34
## 51	Wyoming	WY	West	563626	5	43

(c) Select the columns `state`, `population` and give it a name `new_df`.

```
library(tidyverse)
new_df<-murders%>%
  select(state,population)
new_df
```

```
##           state population
## 1      Alabama    4779736
## 2       Alaska     710231
## 3      Arizona    6392017
## 4      Arkansas    2915918
## 5     California   37253956
## 6      Colorado    5029196
## 7    Connecticut    3574097
## 8      Delaware     897934
## 9 District of Columbia    601723
## 10     Florida   19687653
## 11     Georgia    9920000
## 12     Hawaii    1360301
## 13     Idaho     1567582
## 14     Illinois  12830632
## 15     Indiana    6483802
## 16      Iowa     3046355
## 17     Kansas     2853118
## 18     Kentucky    4339367
## 19     Louisiana    4533372
## 20      Maine     1328361
## 21     Maryland    5773552
## 22    Massachusetts    6547629
## 23      Michigan    9883640
## 24     Minnesota    5303925
## 25     Mississippi    2967297
## 26      Missouri    5988927
## 27      Montana     989415
## 28     Nebraska    1826341
## 29      Nevada     2700551
## 30    New Hampshire    1316470
## 31     New Jersey    8791894
## 32     New Mexico    2059179
## 33      New York   19378102
## 34    North Carolina    9535483
## 35     North Dakota     672591
## 36      Ohio     11536504
## 37     Oklahoma     3751351
## 38      Oregon     3831074
## 39     Pennsylvania  12702379
## 40     Rhode Island    1052567
## 41    South Carolina    4625364
## 42     South Dakota     814180
## 43      Tennessee    6346105
## 44      Texas     25145561
## 45      Utah       2763885
## 46      Vermont     625741
## 47      Virginia    8001024
## 48     Washington    6724540
```

```
## 49      West Virginia    1852994
## 50      Wisconsin      5686986
## 51      Wyoming        563626
```

(d) Filter the observations with only `state== 'New York'` and call it as `murders_ny`.

```
murders_ny<-murders%>%
  filter(state=="New York")
murders_ny
```

```
##      state abb    region population total
## 1 New York  NY Northeast   19378102    517
```

(e) Remove all the observations with `state=='Florida'` and call that dataframe as `murders_no_fl`.

```
murders_no_fl<-murders%>%
  filter(state!="Florida")
murders_no_fl
```

```
##      state abb    region population total
## 1      Alabama AL      South    4779736    135
## 2      Alaska  AK      West     710231     19
## 3      Arizona AZ      West    6392017    232
## 4      Arkansas AR     South    2915918     93
## 5      California CA     West   37253956   1257
## 6      Colorado CO     West    5029196     65
## 7      Connecticut CT    Northeast  3574097     97
## 8      Delaware DE     South     897934     38
## 9 District of Columbia DC     South     601723     99
## 10     Georgia  GA     South    9920000    376
## 11     Hawaii   HI     West    1360301     7
## 12     Idaho    ID     West    1567582    12
## 13     Illinois IL North Central 12830632   364
## 14     Indiana IN North Central  6483802   142
## 15     Iowa     IA North Central  3046355    21
## 16     Kansas  KS North Central  2853118    63
## 17     Kentucky KY     South    4339367   116
## 18     Louisiana LA     South    4533372   351
## 19     Maine    ME    Northeast  1328361    11
## 20     Maryland MD     South    5773552   293
## 21     Massachusetts MA    Northeast  6547629   118
## 22     Michigan MI North Central  9883640   413
## 23     Minnesota MN North Central  5303925    53
## 24     Mississippi MS     South    2967297   120
## 25     Missouri MO North Central  5988927   321
## 26     Montana  MT     West     989415    12
## 27     Nebraska NE North Central  1826341    32
## 28     Nevada   NV     West    2700551    84
## 29     New Hampshire NH    Northeast  1316470     5
## 30     New Jersey NJ    Northeast  8791894   246
## 31     New Mexico NM     West    2059179    67
## 32     New York  NY    Northeast 19378102   517
## 33     North Carolina NC     South    9535483   286
## 34     North Dakota ND North Central  672591     4
## 35     Ohio     OH North Central 11536504   310
## 36     Oklahoma OK     South    3751351   111
```

```
## 37          Oregon OR          West 3831074 36
## 38      Pennsylvania PA      Northeast 12702379 457
## 39      Rhode Island RI      Northeast 1052567 16
## 40      South Carolina SC          South 4625364 207
## 41      South Dakota SD North Central 814180 8
## 42      Tennessee TN          South 6346105 219
## 43      Texas TX          South 25145561 805
## 44      Utah UT          West 2763885 22
## 45      Vermont VT      Northeast 625741 2
## 46      Virginia VA          South 8001024 250
## 47      Washington WA          West 6724540 93
## 48      West Virginia WV          South 1852994 27
## 49      Wisconsin WI North Central 5686986 97
## 50      Wyoming WY          West 563626 5
```

(f) Filter the murders dataset using `%in%` to filter the observations with `state=='New York'` or `state=='Texas'`.

```
murders_tex_ny<-murders%>%
  filter(state%in%c("New York"))
murders_tex_ny
```

```
##      state abb    region population total
## 1 New York NY Northeast 19378102 517
```

(g) Suppose you want to live in the Northeast or West and want the murder rate to be less than 1. How many options do you have?

```
safe_northeast_west<-murders%>%
  mutate(rate=(total/population*100000))%>%
  filter((region%in%c("Northeast","West")&(rate<1)))
safe_northeast_west
```

```
##      state abb    region population total    rate
## 1      Hawaii HI      West 1360301      7 0.5145920
## 2      Idaho ID      West 1567582     12 0.7655102
## 3      Maine ME Northeast 1328361     11 0.8280881
## 4 New Hampshire NH Northeast 1316470      5 0.3798036
## 5      Oregon OR      West 3831074     36 0.9396843
## 6      Utah UT      West 2763885     22 0.7959810
## 7      Vermont VT Northeast 625741      2 0.3196211
## 8      Wyoming WY      West 563626      5 0.8871131
```

Q8. Use a pipe to create a new data frame called `my_states` that considers only states in the Northeast or West which have a murder rate lower than 1, and contains only the state, rate and rank columns.

```
my_states <- murders %>%
  mutate(rate=(total/population*100000),murd_rank=rank(-(total/population*100000))) %>%
  filter((region%in%c("Northeast","West")&(rate<1))) %>%
  select(state,rate,murd_rank)
my_states
```

```
##      state    rate murd_rank
## 1      Hawaii 0.5145920      49
## 2      Idaho 0.7655102      46
## 3      Maine 0.8280881      44
## 4 New Hampshire 0.3798036      50
## 5      Oregon 0.9396843      42
```

```
## 6      Utah 0.7959810      45
## 7      Vermont 0.3196211      51
## 8      Wyoming 0.8871131      43
```

Q9. Install the NHANES package, load the data NHANES.

```
#install.packages("NHANES")
library(NHANES)
data(NHANES)
```

Observe that NHANES data has many missing values.

- (a) Find the mean and the standard deviation of the variable `Age`. Remember to exclude the missing values. Hint: Add `na.rm = TRUE` inside the mean and the sd functions.

```
mean(NHANES$Age, na.rm=TRUE)
```

```
## [1] 36.7421
```

```
sd(NHANES$Age, na.rm=TRUE)
```

```
## [1] 22.39757
```

- (b) First select the group as 20-to-29-year-old females. `AgeDecade` is a categorical variable with these ages. Note that the category is coded like " 20-29", with a space in front! What is the average and standard deviation of systolic blood pressure as saved in the `BPSysAve` variable? Save it to a variable called `ref`.

```
filtered_age<-NHANES%>%
  filter(AgeDecade==" 20-29")
filtered_age
```

```
## # A tibble: 1,356 x 76
##   ID SurveyYr Gender   Age AgeDecade AgeMonths Race1   Race3 Education
##   <int> <fct>   <fct> <int> <fct>         <int> <fct>   <fct> <fct>
## 1 51710 2009_10 female   26 " 20-29"         319 White   <NA> College Grad
## 2 51723 2009_10 male    28 " 20-29"         336 Black   <NA> Some College
## 3 51731 2009_10 female   28 " 20-29"         346 Black   <NA> High School
## 4 51734 2009_10 male    25 " 20-29"         310 White   <NA> High School
## 5 51741 2009_10 female   21 " 20-29"         253 Black   <NA> Some College
## 6 51741 2009_10 female   21 " 20-29"         253 Black   <NA> Some College
## 7 51760 2009_10 female   27 " 20-29"         334 Hispanic <NA> 9 - 11th Grade
## 8 51764 2009_10 female   29 " 20-29"         357 White   <NA> College Grad
## 9 51764 2009_10 female   29 " 20-29"         357 White   <NA> College Grad
## 10 51764 2009_10 female   29 " 20-29"         357 White   <NA> College Grad
## # ... with 1,346 more rows, and 67 more variables: MaritalStatus <fct>,
## #   HHIIncome <fct>, HHIIncomeMid <int>, Poverty <dbl>, HomeRooms <int>,
## #   HomeOwn <fct>, Work <fct>, Weight <dbl>, Length <dbl>, HeadCirc <dbl>,
## #   Height <dbl>, BMI <dbl>, BMICatUnder20yrs <fct>, BMI_WHO <fct>,
## #   Pulse <int>, BPSysAve <int>, BPDiaAve <int>, BPSys1 <int>, BPDia1 <int>,
## #   BPSys2 <int>, BPDia2 <int>, BPSys3 <int>, BPDia3 <int>, Testosterone <dbl>,
## #   DirectChol <dbl>, TotChol <dbl>, UrineVol1 <int>, UrineFlow1 <dbl>, ...
```

```
ref<-filtered_age%>%
  filter(BPSysAve!="NA")%>%
  summarize(avg=mean(BPSysAve, na.rm=TRUE), standev=sd(BPSysAve, na.rm=TRUE))
ref
```

```
## # A tibble: 1 x 2
##   avg standev
```

```
##    <dbl>    <dbl>
## 1  113.    11.7
```

Hint: Use `filter` and `summarize` and use the `na.rm = TRUE` argument when computing the average and standard deviation. You can also `filter` the NA values using `filter`.

- (c) Using a pipe, assign the average to a numeric variable `ref_avg`. Hint: Use the code similar to above and then `pull`.

```
ref_avg<-filtered_age%>%
  filter(BPSysAve!="NA")%>%
  summarize(avg=mean(BPSysAve,na.rm=TRUE),standev=sd(BPSysAve,na.rm=TRUE))%>%
  pull(avg)
ref_avg
```

```
## [1] 113.1583
```

- (d) Compute the average and standard deviation for females, but for each age group separately rather than a selected decade as in the earlier question. Note that the age groups are defined by `AgeDecade`. Hint: rather than filtering by age and gender, `filter` by `Gender` and then use `group_by`.

```
filtered_female_age<-NHANES%>%
  filter(Gender=="female")%>%
  group_by(AgeDecade)%>%
  filter(BPSysAve!="NA")%>%
  summarize(avg=mean(BPSysAve,na.rm=TRUE),standev=sd(BPSysAve,na.rm=TRUE))
filtered_female_age
```

```
## # A tibble: 9 x 3
##   AgeDecade   avg standev
##   <fct>     <dbl>   <dbl>
## 1 " 0-9"    100.    9.07
## 2 " 10-19"  104.    9.46
## 3 " 20-29"  108.   10.1
## 4 " 30-39"  111.   12.3
## 5 " 40-49"  115.   14.5
## 6 " 50-59"  122.   16.2
## 7 " 60-69"  127.   17.1
## 8 " 70+"   134.   19.8
## 9 "<NA>"   142.   22.9
```

- (e) Repeat exercise (d) for males.

```
filtered_male_age<-NHANES%>%
  filter(Gender=="male")%>%
  group_by(AgeDecade)%>%
  filter(BPSysAve!="NA")%>%
  summarize(avg=mean(BPSysAve,na.rm=TRUE),standev=sd(BPSysAve,na.rm=TRUE))
filtered_male_age
```

```
## # A tibble: 5 x 3
##   AgeDecade   avg standev
##   <fct>     <dbl>   <dbl>
## 1 " 0-9"    97.4    8.32
## 2 " 10-19" 110.    11.2
## 3 " 20-29" 118.    11.3
## 4 " 30-39" 119.    12.3
## 5 " 40-49" 121.    14.0
```

```
## 6 " 50-59" 126.    17.8
## 7 " 60-69" 127.    17.5
## 8 " 70+"   130.    18.7
## 9 <NA>     136.    23.5
```

- (f) We can actually combine both summaries for exercises 4 and 5 into one line of code. This is because `group_by` permits us to group by more than one variable. Obtain one big summary table using `group_by(AgeDecade, Gender)`.

```
combined_summary<-NHANES%>%
group_by(AgeDecade,Gender)%>%
  summarize(avg=mean(BPSysAve,na.rm=TRUE),standev=sd(BPSysAve,na.rm=TRUE))
```

``summarise()`` has grouped output by 'AgeDecade'. You can override using the ``.groups`` argument.

```
combined_summary
```

```
## # A tibble: 18 x 4
## # Groups:   AgeDecade [9]
##   AgeDecade Gender   avg standev
##   <fct>      <fct> <dbl>   <dbl>
## 1 " 0-9"     female 100.    9.07
## 2 " 0-9"     male   97.4    8.32
## 3 " 10-19"   female 104.    9.46
## 4 " 10-19"   male  110.   11.2
## 5 " 20-29"   female 108.   10.1
## 6 " 20-29"   male  118.   11.3
## 7 " 30-39"   female 111.   12.3
## 8 " 30-39"   male  119.   12.3
## 9 " 40-49"   female 115.   14.5
## 10 " 40-49"  male  121.   14.0
## 11 " 50-59"   female 122.   16.2
## 12 " 50-59"   male  126.   17.8
## 13 " 60-69"   female 127.   17.1
## 14 " 60-69"   male  127.   17.5
## 15 " 70+"     female 134.   19.8
## 16 " 70+"     male  130.   18.7
## 17 <NA>       female 142.   22.9
## 18 <NA>       male  136.   23.5
```

- (g) For males between the ages of 40-49, compare systolic blood pressure `BPSysAve` across race as reported in the `Race1` variable. Order the resulting table from lowest to highest average systolic blood pressure.

```
male_filter_age<-NHANES%>%
  filter(AgeDecade==" 40-49")
male_comparison<-male_filter_age%>%
  summarize(Race1,BPSysAve)%>%
  arrange(BPSysAve)
male_comparison
```

```
## # A tibble: 1,398 x 2
##   Race1 BPSysAve
##   <fct>   <int>
## 1 White      84
## 2 White      84
## 3 White      86
## 4 White      86
```

```
## 5 White      86
## 6 White      86
## 7 White      86
## 8 White      87
## 9 White      87
## 10 White     88
## # ... with 1,388 more rows
```