CptS 475/575: Data Science, Fall 2018

Assignment 3: Data Transformation and Tidying

Release Date: September 19, 2018 **Due Date:** September 27, 2018 (11:59 pm)

This assignment has two questions. You are encouraged to use R Markdown to generate your report (in PDF).

Question 1. For this question you will be using the dplyr package to manipulate and clean up a dataset called msleep (mammals sleep) that is available on the course webpage (at https://scads.eecs.wsu.edu/wp-content/uploads/2017/10/msleep_ggplot2.csv). The dataset contains the sleep times and weights for a set of mammals. It has 83 rows and 11 variables. Here is a description of the variables:

Name	Description
name	common name
genus	taxonomic rank
vore	carnivore, omnivore or herbivore?
order	taxonomic rank
conservation	the conservation status of the mammal
sleep_total	total amount of sleep, in hours
sleep_rem	rem sleep, in hours
sleep_cycle	length of sleep cycle, in hours
awake	amount of time spent awake, in hours
brainwt	brain weight in kilograms
bodywt	body weight in kilograms

Load the data into R, and check the first few rows for abnormalities. You will likely notice several.

Below are the tasks to perform. Use select() to print the head of the columns with a title including "sleep".

a) Use filter() to count the number of animals which weigh over 50 kilograms and sleep more than 6 hours a day.
b) Use piping (%>%), select() and arrange() to print the name, order, sleep time and bodyweight of the animals with the top 6 sleep times, in order of sleep time.
c) Use mutate to add two new columns to the dataframe; wt_ratio with the ratio of brain size to body weight, rem_ratio with the ratio of rem sleep to sleep time. If you think they might be useful, feel free to extract more features than these, and describe what they are.
d) Use group_by() and summarize() to display the average, min and max sleep times for each order. Remember to use ungroup() when you are done.
e) Make a copy of your dataframe, and use group_by() and mutate() to impute the missing brain weights as the average wt_ratio for that animal's order times the animal's weight. Make a

second copy of your dataframe, but this time use group_by() and mutate() to impute missing brain weights with the average brain weight for that animal's order. What assumptions do these data filling methods make? Which is the best way to impute the data, or do you see a better way, and why? You may impute or remove other variables as you find appropriate. Briefly explain your decisions.

Question 2. For this question, you will first need to read section 12.6 in the R for Data Science book, here (http://r4ds.had.co.nz/tidy-data.html#case-study). Grab the dataset from the tidyr package, and tidy it as shown in the case study before answering the following questions.

a) Explain why this line

```
> mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
```

is necessary to properly tidy the data. What happens if you skip this line?

b) How many entries are removed from the dataset when you set na.rm to true in the gather command (in this dataset). How else could those NA values be handled? Among these options, which do you think is the best way to handle those missing values for this dataset, and why?

c) Explain the difference between an explicit and implicit missing value, in general. Can you find any implicit missing values in this dataset, if so where?

d) Looking at the features (country, year, var, sex, age, cases) in the tidied data, are they all appropriately typed? Are there any features you think would be better suited as a different type? Why or why not?

e) Explain in your own words what a gather operation is, and give an example of a situation when it might be useful. Do the same for spread.

f) Generate an informative visualization, which shows something about the data. Give a brief description of what it shows, and why you thought it was interesting.

Assignment 3

Sheryl Mathew (11627236)

27 September, 2018

1. Mammals Sleep Dataset

1.1 Load the data into R

```
library(kableExtra)
msleep = read.csv("https://scads.eecs.wsu.edu/wp-content/uploads/2017/10/msleep_ggplot2.csv")
kable(head(msleep, n=30), format = "latex", booktabs = T,
      caption="Table with Mammals and their Sleep Patterns") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 1: Table with Mammals and their Sleep Patterns

name	genus	vore	order	conservation	sleep_total	sleep_rem	sleep_cycle	awake	brainwt	bodywt
Cheetah	Acinonyx	carni	Carnivora	lc	12.1	NA	NA	11.9	NA	50.000
Owl monkey	Aotus	omni	Primates	NA	17.0	1.8	NA	7.0	0.01550	0.480
Mountain beaver	Aplodontia	herbi	Rodentia	nt	14.4	2.4	NA	9.6	NA	1.350
Greater short-tailed shrew	Blarina	omni	Soricomorpha	lc	14.9	2.3	0.1333333	9.1	0.00029	0.019
Cow	Bos	herbi	Artiodactyla	domesticated	4.0	0.7	0.6666667	20.0	0.42300	600.000
Three-toed sloth	Bradypus	herbi	Pilosa	NA	14.4	2.2	0.7666667	9.6	NA	3.850
Northern fur seal	Callorhinus	carni	Carnivora	vu	8.7	1.4	0.3833333	15.3	NA	20.490
Vesper mouse	Calomys	NA	Rodentia	NA	7.0	NA	NA	17.0	NA	0.045
Dog	Canis	carni	Carnivora	domesticated	10.1	2.9	0.3333333	13.9	0.07000	14.000
Roe deer	Capreolus	herbi	Artiodactyla	lc	3.0	NA	NA	21.0	0.09820	14.800
Goat	Capri	herbi	Artiodactyla	lc	5.3	0.6	NA	18.7	0.11500	33.500
Guinea pig	Cavis	herbi	Rodentia	domesticated	9.4	0.8	0.2166667	14.6	0.00550	0.728
Grivet	Cercopithecus	omni	Primates	lc	10.0	0.7	NA	14.0	NA	4.750
Chinchilla	Chinchilla	herbi	Rodentia	domesticated	12.5	1.5	0.1166667	11.5	0.00640	0.420
Star-nosed mole	Condylura	omni	Soricomorpha	lc	10.3	2.2	NA	13.7	0.00100	0.060
African giant pouched rat	Cricetomys	omni	Rodentia	NA	8.3	2.0	NA	15.7	0.00660	1.000
Lesser short-tailed shrew	Cryptotis	omni	Soricomorpha	lc	9.1	1.4	0.1500000	14.9	0.00014	0.005
Long-nosed armadillo	Dasypus	carni	Cingulata	lc	17.4	3.1	0.3833333	6.6	0.01080	3.500
Tree hyrax	Dendrohyrax	herbi	Hyracoidea	lc	5.3	0.5	NA	18.7	0.01230	2.950
North American Opossum	Didelphis	omni	Didelphimorphia	lc	18.0	4.9	0.3333333	6.0	0.00630	1.700
Asian elephant	Elephas	herbi	Proboscidea	en	3.9	NA	NA	20.1	4.60300	2547.000
Big brown bat	Eptesicus	insecti	Chiroptera	lc	19.7	3.9	0.1166667	4.3	0.00030	0.023
Horse	Equus	herbi	Perissodactyla	domesticated	2.9	0.6	1.0000000	21.1	0.65500	521.000
Donkey	Equus	herbi	Perissodactyla	domesticated	3.1	0.4	NA	20.9	0.41900	187.000
European hedgehog	Erinaceus	omni	Erinaceomorpha	lc	10.1	3.5	0.2833333	13.9	0.00350	0.770
Patas monkey	Erythrocebus	omni	Primates	lc	10.9	1.1	NA	13.1	0.11500	10.000
Western american chipmunk	Eutamias	herbi	Rodentia	NA	14.9	NA	NA	9.1	NA	0.071
Domestic cat	Felis	carni	Carnivora	domesticated	12.5	3.2	0.4166667	11.5	0.02560	3.300
Galago	Galago	omni	Primates	NA	9.8	1.1	0.5500000	14.2	0.00500	0.200
Giraffe	Giraffa	herbi	Artiodactyla	cd	1.9	0.4	NA	22.1	NA	899.995

1.2 Print the head of the columns with a title including "sleep"

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
msleepColumns = select(msleep,contains("sleep"))
kable(head(msleepColumns), format = "latex", booktabs = T,
      caption="Table with Column names Sleep") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 2: Table with Column names Sleep

sleep_total	sleep_rem	sleep_cycle
12.1	NA	NA
17.0	1.8	NA
14.4	2.4	NA
14.9	2.3	0.1333333
4.0	0.7	0.6666667
14.4	2.2	0.7666667

1.3 Count the number of animals which weigh over 50 kilograms and sleep more than 6 hours a day

```
countMammals1=msleep%>%
  select("name","genus","order","sleep_total","bodywt","brainwt")%>%
  filter(bodywt>50 & sleep_total>6)
kable(countMammals1, format = "latex", booktabs = T,
      caption="Table with Mammals which weigh over 50 kilograms and sleep more than
      6 hours a day") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 3: Table with Mammals which weigh over 50 kilograms and sleep more than 6 hours a day

name	genus	order	sleep_total	bodywt	brainwt
Gray seal	Haliochoerus	Carnivora	6.2	85.000	0.325
Human	Homo	Primates	8.0	62.000	1.320
Chimpanzee	Pan	Primates	9.7	52.200	0.440
Tiger	Panthera	Carnivora	15.8	162.564	NA
Jaguar	Panthera	Carnivora	10.4	100.000	0.157
Lion	Panthera	Carnivora	13.5	161.499	NA
Giant armadillo	Priodontes	Cingulata	18.1	60.000	0.081
Pig	Sus	Artiodactyla	9.1	86.250	0.180

```
countMammals2=msleep%>%
  select("name","genus","order","sleep_total","bodywt","brainwt")%>%
  filter((bodywt+brainwt)>50 & sleep_total>6)
kable(countMammals2, format = "latex", booktabs = T,
      caption="Table with Mammals which weigh over 50 kilograms and sleep more than
      6 hours a day") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 4: Table with Mammals which weigh over 50 kilograms and sleep more than 6 hours a day

name	genus	order	sleep_total	bodywt	brainwt
Gray seal	Haliochoerus	Carnivora	6.2	85.00	0.325
Human	Homo	Primates	8.0	62.00	1.320
Chimpanzee	Pan	Primates	9.7	52.20	0.440
Jaguar	Panthera	Carnivora	10.4	100.00	0.157
Giant armadillo	Priodontes	Cingulata	18.1	60.00	0.081
Pig	Sus	Artiodactyla	9.1	86.25	0.180

Total number of mammals (body weight) is 8
Total number of mammals (body and brain weight) is 6

1.4 Print the name, order, sleep time and bodyweight of the animals with the top 6 sleep times, in order of sleep time

```
orderMammals=msleep%>%
  select("name","genus","vore","order","sleep_total","bodywt")%>%
  arrange(desc(sleep_total))
kable(head(orderMammals,n=6), format = "latex", booktabs = T,
      caption="Table with Mammals which Top 6 Sleep Times") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 5: Table with Mammals which Top 6 Sleep Times

name	genus	vore	order	sleep_total	bodywt
Little brown bat	Myotis	insecti	Chiroptera	19.9	0.010
Big brown bat	Eptesicus	insecti	Chiroptera	19.7	0.023
Thick-tailed opposum	Lutreolina	carni	Didelphimorphia	19.4	0.370
Giant armadillo	Priodontes	insecti	Cingulata	18.1	60.000
North American Opossum	Didelphis	omni	Didelphimorphia	18.0	1.700
Long-nosed armadillo	Dasypus	carni	Cingulata	17.4	3.500

1.5 Add two new columns to the dataframe; wt_ratio with the ratio of brain size to body weight, rem_ratio with the ratio of rem sleep to sleep time. If you think they might be useful, feel free to extract more features than these, and describe what they are

```
newMammalsColumns= msleep%>%
  select("name","order","sleep_total","sleep_rem","awake","bodywt","brainwt")%>%
  mutate(
    wt_ratio = brainwt/bodywt,
    rem_ratio = sleep_rem/sleep_total,
    awake_ratio=awake/sleep_total,
    total_wt=brainwt+bodywt
  )
kable(head(newMammalsColumns), format = "latex", booktabs = T,
      caption="Table with New Columns Added") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 6: Table with New Columns Added

name	order	sleep_total	sleep_rem	awake	bodywt	brainwt	wt_ratio	rem_ratio	awake_ratio	total_wt
Cheetah	Carnivora	12.1	NA	11.9	50.000	NA	NA	NA	0.9834711	NA
Owl monkey	Primates	17.0	1.8	7.0	0.480	0.01550	0.0322917	0.1058824	0.4117647	0.49550
Mountain beaver	Rodentia	14.4	2.4	9.6	1.350	NA	NA	0.1666667	0.6666667	NA
Greater short-tailed shrew	Soricomorpha	14.9	2.3	9.1	0.019	0.00029	0.0152632	0.1543624	0.6107383	0.01929
Cow	Artiodactyla	4.0	0.7	20.0	600.000	0.42300	0.0007050	0.1750000	5.0000000	600.42300
Three-toed sloth	Pilosa	14.4	2.2	9.6	3.850	NA	NA	0.1527778	0.6666667	NA

Description:
wt_ratio: Ratio of brain size to body weight
rem_ratio : Ratio of rem sleep to sleep time
awake_ratio: Ratio of awake time to sleep time
total_wt: Body weight + Brain weight

1.6 Display the average, min and max sleep times for each order

```
mammalStats=msleep%>%
  group_by(order)%>%
  summarise( aveargе=mean(sleep_total),minimum=min(sleep_total),maximum=max(sleep_total),
             total=n())%>%
  ungroup(order)
kable(mammalStats, format = "latex", booktabs = T,
      caption="Table with average, min and max sleep times for each order") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 7: Table with average, min and max sleep times for each order

order	aveargе	minimum	maximum	total
Afrosoricida	15.600000	15.6	15.6	1
Artiodactyla	4.516667	1.9	9.1	6
Carnivora	10.116667	3.5	15.8	12
Cetacea	4.500000	2.7	5.6	3
Chiroptera	19.800000	19.7	19.9	2
Cingulata	17.750000	17.4	18.1	2
Didelphimorphia	18.700000	18.0	19.4	2
Diprotodontia	12.400000	11.1	13.7	2
Erinaceomorpha	10.200000	10.1	10.3	2
Hyracoidea	5.666667	5.3	6.3	3
Lagomorpha	8.400000	8.4	8.4	1
Monotremata	8.600000	8.6	8.6	1
Perissodactyla	3.466667	2.9	4.4	3
Pilosa	14.400000	14.4	14.4	1
Primates	10.500000	8.0	17.0	12
Proboscidea	3.600000	3.3	3.9	2
Rodentia	12.468182	7.0	16.6	22
Scandentia	8.900000	8.9	8.9	1
Soricomorpha	11.100000	8.4	14.9	5

1.7 Make a copy of your dataframe, and use group_by() and mutate() to impute the missing brain weights as the average wt_ratio for that animal's order times the animal's weight. Make a second copy of your dataframe, but this time use group_by() and mutate() to impute missing brain weights with the average brain weight for that animal's order. What assumptions do these data filling methods make? Which is the best way to impute the data, or do you see a better way, and why? You may impute or remove other variables as you find appropriate. Briefly explain your decisions.

```
missingAVgWtRatio=msleep%>%
  group_by(order)%>%
  select("name","genus","vore","order","brainwt","bodywt")%>%
  mutate(
    brainwt=ifelse(
      is.na(brainwt),
      ifelse(is.nan(mean(brainwt,na.rm = TRUE)),0,mean(brainwt,na.rm = TRUE))/mean(bodywt,na.rm = TRUE)*
  )%>%
  ungroup(order)
kable(head(missingAVgWtRatio), format = "latex", booktabs = T,
      caption="Missing weight computed as avearage weight ratio times animal weight") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 8: Missing weight computed as avearage weight ratio times animal weight

name	genus	vore	order	brainwt	bodywt
Cheetah	Acinonyx	carni	Carnivora	0.0854094	50.000
Owl monkey	Aotus	omni	Primates	0.0155000	0.480
Mountain beaver	Aplodontia	herbi	Rodentia	0.0167118	1.350
Greater short-tailed shrew	Blarina	omni	Soricomorpha	0.0002900	0.019
Cow	Bos	herbi	Artiodactyla	0.4230000	600.000
Three-toed sloth	Bradypus	herbi	Pilosa	0.0000000	3.850

```
missingAVgWt=msleep%>%
  group_by(order)%>%
  select("name","genus","vore","order","brainwt","bodywt")%>%
  mutate(
    brainwt=ifelse(
      is.na(brainwt),
      ifelse(is.nan(mean(brainwt,na.rm = TRUE)),0,mean(brainwt,na.rm = TRUE)),brainwt)
  )%>%
  ungroup(order)
kable(head(missingAVgWt), format = "latex", booktabs = T,
      caption="Missing brain weight is replaced as average brain weight")%>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 9: Missing brain weight is replaced as average brain weight

name	genus	vore	order	brainwt	bodywt
Cheetah	Acinonyx	carni	Carnivora	0.0985714	50.000
Owl monkey	Aotus	omni	Primates	0.0155000	0.480
Mountain beaver	Aplodontia	herbi	Rodentia	0.0035680	1.350
Greater short-tailed shrew	Blarina	omni	Soricomorpha	0.0002900	0.019
Cow	Bos	herbi	Artiodactyla	0.4230000	600.000
Three-toed sloth	Bradypus	herbi	Pilosa	0.0000000	3.850

```
missingSleepRem=msleep%>%
  group_by(order)%>%
  select("name","genus","vore","order","sleep_rem","sleep_cycle")%>%
  mutate(
    sleep_rem=ifelse(
      is.na(sleep_rem),
      ifelse(is.nan(mean(sleep_rem,na.rm = TRUE)),0,mean(sleep_rem,na.rm = TRUE)),sleep_rem)
  )%>%
  ungroup(order)
kable(head(missingSleepRem), format = "latex", booktabs = T,
      caption="Missing sleep rem is replaced as average sleep rem")%>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 10: Missing sleep rem is replaced as average sleep rem

name	genus	vore	order	sleep_rem	sleep_cycle
Cheetah	Acinonyx	carni	Carnivora	1.871429	NA
Owl monkey	Aotus	omni	Primates	1.800000	NA
Mountain beaver	Aplodontia	herbi	Rodentia	2.400000	NA
Greater short-tailed shrew	Blarina	omni	Soricomorpha	2.300000	0.1333333
Cow	Bos	herbi	Artiodactyla	0.700000	0.6666667
Three-toed sloth	Bradypus	herbi	Pilosa	2.200000	0.7666667

```
missingSleepCycle=msleep%>%
  group_by(order)%>%
  select("name","genus","vore","order","sleep_rem","sleep_cycle")%>%
  mutate(
    sleep_cycle=ifelse(
      is.na(sleep_cycle),
      ifelse(is.nan(mean(sleep_cycle,na.rm = TRUE)),0,mean(sleep_cycle,na.rm = TRUE)),sleep_cycle)
  )%>%
  ungroup(order)
kable(head(missingSleepCycle), format = "latex", booktabs = T,
      caption="Missing sleep cycle is replaced as average sleep cycle") %>%
kable_styling(latex_options = c("striped","hold_position","scale_down"))
```

Table 11: Missing sleep cycle is replaced as average sleep cycle

name	genus	vore	order	sleep_rem	sleep_cycle
Cheetah	Acinonyx	carni	Carnivora	NA	0.3708333
Owl monkey	Aotus	omni	Primates	1.8	0.9766667
Mountain beaver	Aplodontia	herbi	Rodentia	2.4	0.1809524
Greater short-tailed shrew	Blarina	omni	Soricomorpha	2.3	0.1333333
Cow	Bos	herbi	Artiodactyla	0.7	0.6666667
Three-toed sloth	Bradypus	herbi	Pilosa	2.2	0.7666667

The best way to replace missing values is by taking the mean of the brain weight since replacing the mean will not affect the data when we perform statistical operations on it. Therefore the missing values of sleep_rem and sleep_cycle has been replaced by the corresponding mean of their orders.

2. WHO Tuberclosis Dataset

2.1 Load the data into R

```
library(kableExtra)
library(tidyr)
who = tidyr::who
kable(head(who), format = "latex", booktabs = T,
      caption="Table with Tuberclosis Details") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 12: Table with Tuberclosis Details

country	iso2	iso3	year	new_sp_m014	new_sp_m1524	new_sp_m2534	new_sp_m3544	new_sp_m4
Afghanistan	AF	AFG	1980	NA	NA	NA	NA	
Afghanistan	AF	AFG	1981	NA	NA	NA	NA	
Afghanistan	AF	AFG	1982	NA	NA	NA	NA	
Afghanistan	AF	AFG	1983	NA	NA	NA	NA	
Afghanistan	AF	AFG	1984	NA	NA	NA	NA	
Afghanistan	AF	AFG	1985	NA	NA	NA	NA	

2.2 Explain why this line mutate(key = stringr::str_replace(key, "newrel", "new_rel")) is necessary to properly tidy the data. What happens if you skip this line?

This line is used to replace all the strings which contain newrel as column name to new_rel. This is done because when we try to separate the data using separate(key, c("new", "type", "sexage"), sep = "_"), if we retain the column name as newrel itself, both the details of whether is a new case of TB (new) and the type of TB (rel) will both be present in the new column itself. If we change newrel to new_rel there will be consistency in all the column names which not only makes the separate function execute as expected other functions will also work as expected.

2.3 How many entries are removed from the dataset when you set na.rm to true in the gather command (in this dataset). How else could those NA values be handled? Among these options, which do you think is the best way to handle those missing values for this dataset, and why?

```
gatherWithNa=who %>%
  gather(key, value, new_sp_m014:newrel_f65, na.rm = FALSE) %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel")) %>%
  separate(key, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
kable(head(gatherWithNa), format = "latex", booktabs = T,
      caption="Table with Tuberclosis Details containing NA") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 13: Table with Tuberclosis Details containing NA

country	year	var	sex	age	value
Afghanistan	1980	sp	m	014	NA
Afghanistan	1981	sp	m	014	NA
Afghanistan	1982	sp	m	014	NA
Afghanistan	1983	sp	m	014	NA
Afghanistan	1984	sp	m	014	NA
Afghanistan	1985	sp	m	014	NA

```
countWithNa=count(gatherWithNa)
gatherWithoutNa=who %>%
  gather(key, value, new_sp_m014:newrel_f65, na.rm = TRUE) %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel")) %>%
  separate(key, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
countWithoutNa=count(gatherWithoutNa)
kable(head(gatherWithoutNa), format = "latex", booktabs = T,
      caption="Table with Tuberclosis Details without NA") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 14: Table with Tuberclosis Details without NA

country	year	var	sex	age	value
Afghanistan	1997	sp	m	014	0
Afghanistan	1998	sp	m	014	30
Afghanistan	1999	sp	m	014	8
Afghanistan	2000	sp	m	014	52
Afghanistan	2001	sp	m	014	129
Afghanistan	2002	sp	m	014	90

Total Details Including NA = 405440
Total Details without NA = 76046
Total Details dropped = 329394

The better way to handle the data is to replace the missing values based on the mean for the country.

Removing NA values will be better because all irrelevant data is removed from the dataset and we are working with a smaller subset which improves efficiency while performing analysis.

2.4 Explain the difference between an explicit and implicit missing value, in general. Can you find any implicit missing values in this dataset, if so where?

Explicit missing values means there is a specific representation that will indicate the row has missing value (row=NA)
Implicit missing values means the value is not present(row=" ") or might be represented differently (row = 0)

```
whoNew=who
implictCount=whoNew %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = TRUE)%>%
  filter(cases == 0) %>%
  nrow()
```

Total Implicit Missing Value = 11080

2.5 Looking at the features (country, year, var, sex, age, cases) in the tidied data, are they all appropriately typed? Are there any features you think would be better suited as a different type? Why or why not?

```
whoTidiedData=who %>%
  gather(key, value, new_sp_m014:newrel_f65, na.rm = TRUE) %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel")) %>%
  separate(key, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
kable(head(whoTidiedData), format = "latex", booktabs = T,
      caption="Table with Tuberclosis Formatted Details") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 15: Table with Tuberclosis Formatted Details

country	year	var	sex	age	value
Afghanistan	1997	sp	m	014	0
Afghanistan	1998	sp	m	014	30
Afghanistan	1999	sp	m	014	8
Afghanistan	2000	sp	m	014	52
Afghanistan	2001	sp	m	014	129
Afghanistan	2002	sp	m	014	90

```
sapply(whoTidiedData,class)
```

```
##     country        year         var         sex         age       value
## "character"   "integer" "character" "character" "character"   "integer"
```

The column *age* can be changed from character to integer since it contains only integer values

2.6 Explain in your own words what a gather operation is, and give an example of a situation when it might be useful. Do the same for spread.

Gather is used to push data that is present in columns into rows ie we make wide data to long
Spread is used to pull data that is present in rows into columns ie we make long data to wide

```
details = data.frame(
  Id = c(1:10),
  Data1 = c(411,723,325,456,579,612,709,513,527,379),
  Data2 = c(123,300,400,500,600,654,789,906,413,567),
  Data3 = c(1457,1000,569,896,956,2345,780,599,1023,678)
)
kable(head(details), format = "latex", booktabs = T,
      caption="Table with all the Details") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 16: Table with all the Details

Id	Data1	Data2	Data3
1	411	123	1457
2	723	300	1000
3	325	400	569
4	456	500	896
5	579	600	956
6	612	654	2345

```
gatherData = gather(details,Data, ResponseTime, Data1:Data3)
kable(head(gatherData), format = "latex", booktabs = T,
      caption="Table after performing gather") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 17: Table after performing gather

Id	Data	ResponseTime
1	Data1	411
2	Data1	723
3	Data1	325
4	Data1	456
5	Data1	579
6	Data1	612

```
spreadData = spread(gatherData,Data, ResponseTime)
kable(head(spreadData), format = "latex", booktabs = T,
      caption="Table after performing spread") %>%
kable_styling(latex_options = c("striped","hold_position"))
```

Table 18: Table after performing spread

Id	Data1	Data2	Data3
1	411	123	1457
2	723	300	1000
3	325	400	569
4	456	500	896
5	579	600	956
6	612	654	2345

2.7 Generate an informative visualization, which shows something about the data. Give a brief description of what it shows, and why you thought it was interesting.

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     msleep
```

```
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

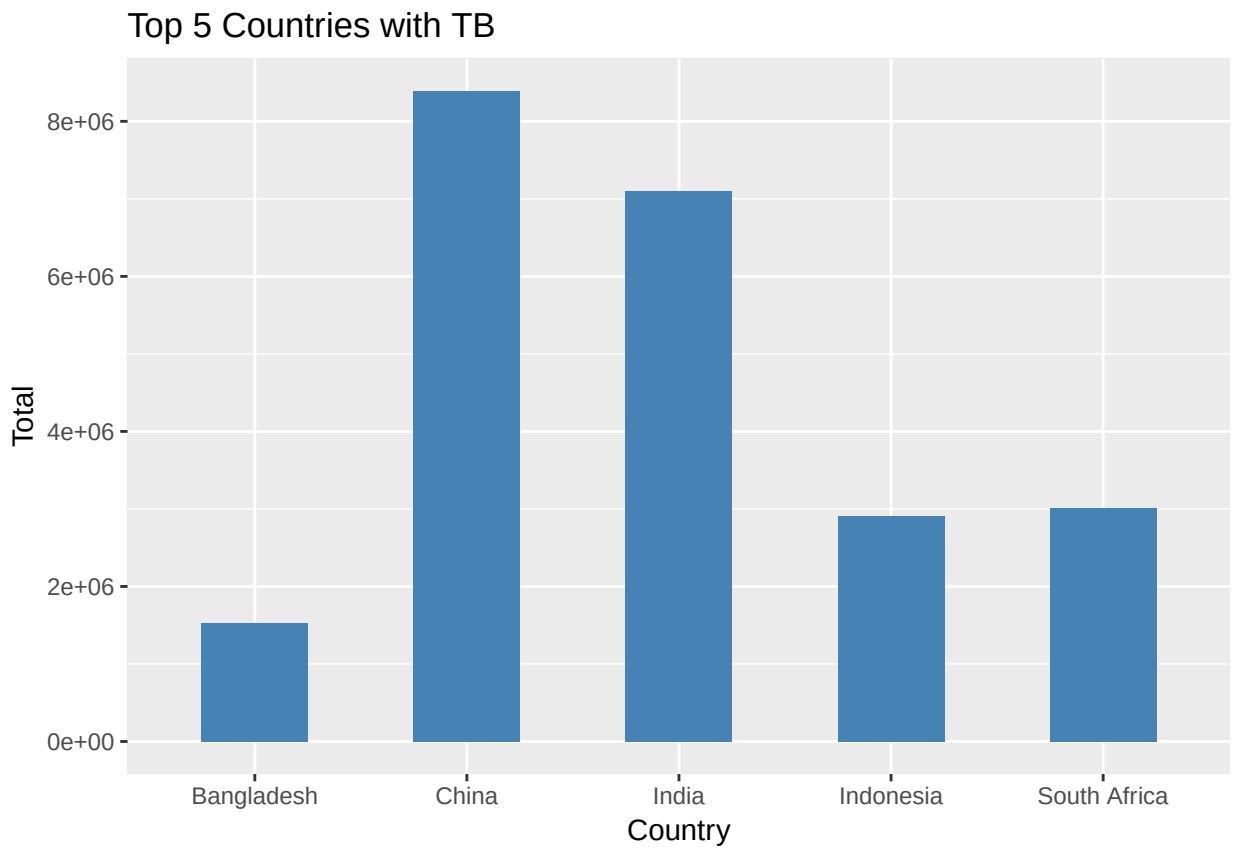
```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
whoVis= who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = TRUE)%>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))%>%
  separate(key, c("new", "type", "sexage"), sep = "_")%>%
  select(-new, -iso2, -iso3)%>%
  separate(sexage, c("sex", "age"), sep = 1)
```

```
countryCases=whoVis%>%
  group_by(country)%>%
  tally(cases)%>%
  top_n(5)
```

```
## Selecting by n
```

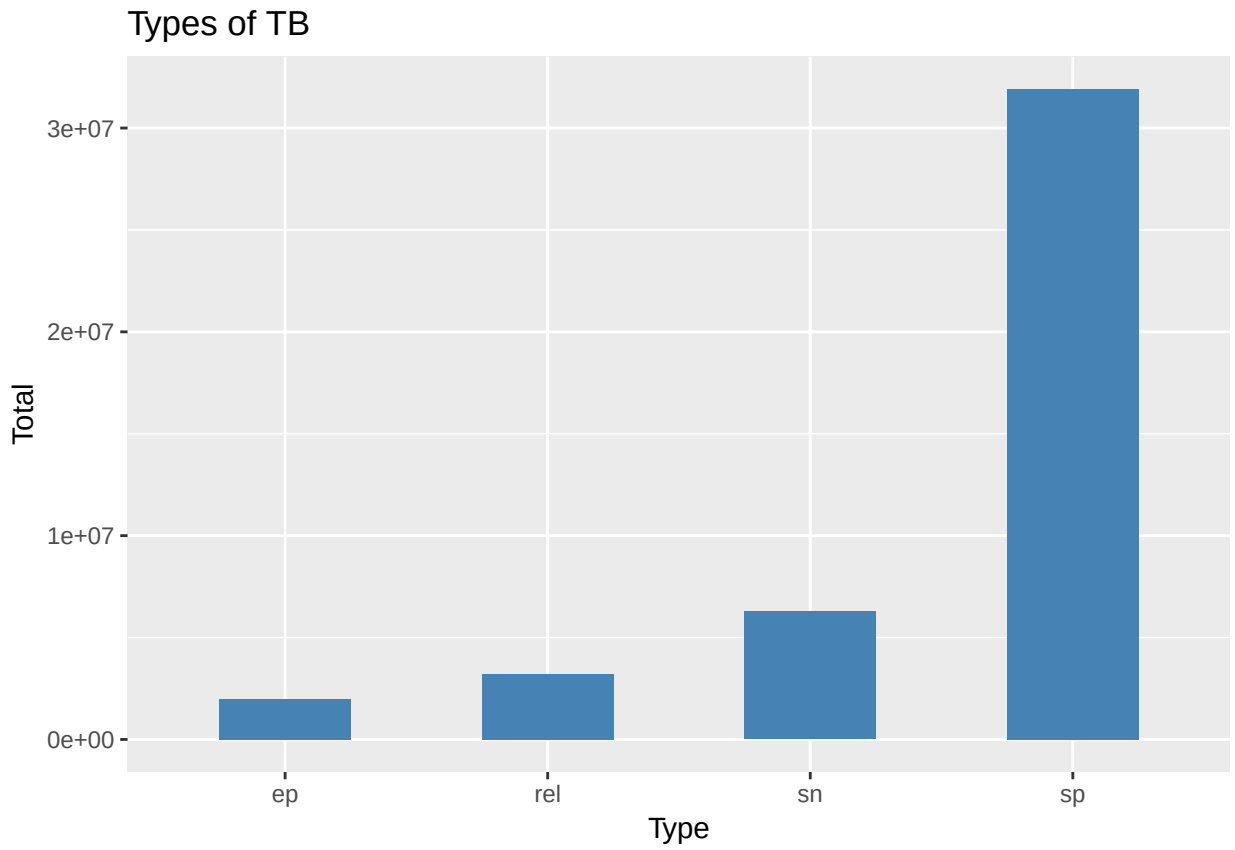
```
ggplot(data=countryCases, aes(x=country,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Top 5 Countries with TB")+
  xlab("Country")+ylab("Total")
```



Graph 1: Shows the countries with most TB cases. From the plot we can understand Asia and South Africa are the most affected.

```r
typeCases=whoVis%>%
  group_by(type)%>%
  tally(cases)

ggplot(data=typeCases, aes(x=type,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Types of TB")+
  xlab("Type")+ylab("Total")
```



Graph 2: Shows the different types of TB. From the plot we can find that sp is the most common type of TB.

```r
yearCases=whoVis%>%
  group_by(year)%>%
  tally(cases)%>%
  top_n(5)
```

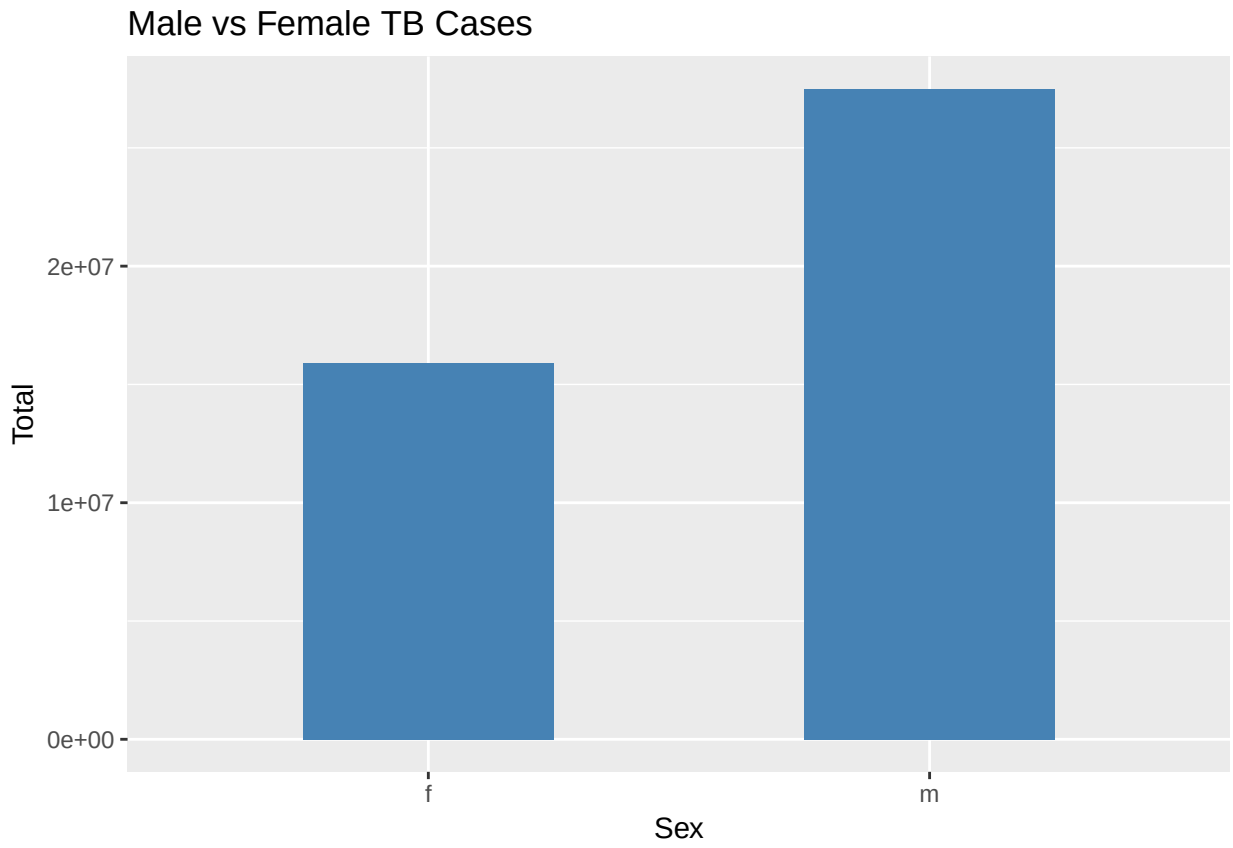
```
## Selecting by n
```

```r
ggplot(data=yearCases, aes(x=year,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Top 5 Years with most TB Cases")+
  xlab("Year")+ylab("Total")
```

Graph 3: Shows the years with most TB cases. From the plot we can see that from 2007-2012 has seen the most TB cases and the number of TB cases in these years are almost same.

```r
sexCases=whoVis%>%
  group_by(sex)%>%
  tally(cases)

ggplot(data=sexCases, aes(x=sex,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Male vs Female TB Cases")+
  xlab("Sex")+ylab("Total")
```



Graph 4: Show the number of TB cases for men and women. From the plot we know that men are more prone to TB when compared to women.