MATH2349 Semester 1, 2018

Code

Assignment 3

S3335814- Minh Phan

Required packages

Hide

```
# loadding packages
install.packages("readr")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/readr_1.1.1.zip'
Content type 'application/zip' length 1377603 bytes (1.3 MB)
downloaded 1.3 MB
```

```
package readr successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("dplyr")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("tidyr")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/tidyr_0.8.1.zip'
Content type 'application/zip' length 943474 bytes (921 KB)
downloaded 921 KB
```

```
package tidyr successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("dplyr")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/dplyr_0.7.5.zip'
Content type 'application/zip' length 3047294 bytes (2.9 MB)
downloaded 2.9 MB
```

```
package dplyr successfully unpacked and MD5 sums checked
Warning in install.packages :
  cannot remove prior installation of package dplyr

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("knitr")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("lubridate")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/lubridate_1.7.4.zip'
Content type 'application/zip' length 1567999 bytes (1.5 MB)
downloaded 1.5 MB
```

```
package lubridate successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("knitr")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/knitr_1.20.zip'
Content type 'application/zip' length 1181589 bytes (1.1 MB)
downloaded 1.1 MB
```

```
package knitr successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("stringr")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/stringr_1.3.1.zip'
Content type 'application/zip' length 194506 bytes (189 KB)
downloaded 189 KB
```

```
package stringr successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("outliers")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("MVN")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/MVN_5.3.zip'
Content type 'application/zip' length 385833 bytes (376 KB)
downloaded 376 KB
```

```
package MVN successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("datasets")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("MASS")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+393C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
```

```
  There is a binary version available but the source version is later:
```

	binary <fctr>	source <fctr>	needs_compilation <lgl>
MASS	7.3-49	7.3-50	TRUE

1 row

```
  Binaries will be installed
```

```
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/MASS_7.3-49.zip'
Content type 'application/zip' length 1171437 bytes (1.1 MB)
downloaded 1.1 MB
```

```
package MASS successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("outliers")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
```

```
Warning in install.packages :
  package outliers is in use and will not be installed
```

Hide

```
install.packages("datasets")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
```

```
Warning in install.packages :
  package datasets is not available (for R version 3.5.0)
Warning in install.packages :
  package datasets is a base package, and should not be updated
```

Hide

```
install.packages("deductive")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+39
3C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/deductive_0.1.2.zip'
Content type 'application/zip' length 56096 bytes (54 KB)
downloaded 54 KB
```

```
package deductive successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Minh\AppData\Local\Temp\RtmpGOhcVV\downloaded_packages
```

Hide

```
install.packages("validate")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("Hmisc")
```

```
Installing package into <U+393C><U+3E31>C:/Users/Minh/Documents/R/win-library/3.5<U+393C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/Hmisc_4.1-1.zip'
Content type 'application/zip' length 3012325 bytes (2.9 MB)
```

```
Restarting R session...
```

Executive Summary

The final dataset was a productof 7 different data sets. Firstly, all the excels files downloaded from http://www.abs.gov.au/ (http://www.abs.gov.au/), were converted into a single page excel files, because some of them have 3 sheets in a file. All these data sets were imported to R-studio using various of packages. Variables in some data sets were renamed for easier access. Subsequently, all the data sets relating to consumption of meat were joined using left joint. I did not used merge function because the data sets dont match in numbers of observation and not all the information was interested. The select() function was used to select wanted variables from population and consumption related datasets; this helped to reduced repeated or unwanted variables. All the variables in each data set were checked and make sure that they are appropriate. The popolation dataset requires some tidy up before joining to the meat_cons dataset. After joining the meat_cons and population datasets, the proprocess of checking variables were repeated; some variables were renamed.The dataset is now meat_cons_pop, the next step was scanning for missing values and outliers. The last step was tranforming some of the numerical variables appropriately. In addition, a subset of the dataset was made only including Australia meat consumption. This dataset was joined with the datasets realating to AUstralian meat production and export in order to find out interested information. A statement about Australian meat consumption and wastage was made at the end, using statistical values from the combined dataset.

Data

The data included 7 data sets. Three of them were download from https://data.oecd.org/agroutput/meat-consumption.htm (https://data.oecd.org/agroutput/meat-consumption.htm), including: beef and veal.csv, pork.csv,sheep.csv. One was download from https://data.worldbank.org/indicator/SP.POP.TOTL (https://data.worldbank.org/indicator/SP.POP.TOTL) ,australia population.xls. Three was downloaded from http://www.abs.gov.au/ (http://www.abs.gov.au/), including pork production aus.xls,meat export aus.xls,pork production aus.xls

Hide

```
#import/ scapping data
beef_and_veal_consumption <- read_csv("beef and veal.csv")
```

```
Parsed with column specification:
cols(
  LOCATION = col_character(),
  INDICATOR = col_character(),
  SUBJECT = col_character(),
  MEASURE = col_character(),
  FREQUENCY = col_character(),
  TIME = col_integer(),
  Value = col_double(),
  `Flag Codes` = col_character()
)
```

Hide

```
 pork_consumption <- read_csv("pork.csv")
```

```
Parsed with column specification:
cols(
  LOCATION = col_character(),
  INDICATOR = col_character(),
  SUBJECT = col_character(),
  MEASURE = col_character(),
  FREQUENCY = col_character(),
  TIME = col_integer(),
  Value = col_double(),
  `Flag Codes` = col_character()
)
```

Hide

```
 pork_production<-read_excel("pork production aus.xls")
 redmeat_production<-read_excel("red meat production aus.xls")
 lamb_consumption<-read_csv("sheep.csv")
```

```
Parsed with column specification:
cols(
  LOCATION = col_character(),
  INDICATOR = col_character(),
  SUBJECT = col_character(),
  MEASURE = col_character(),
  FREQUENCY = col_character(),
  TIME = col_integer(),
  Value = col_double(),
  `Flag Codes` = col_character()
)
```

Hide

```
meat_export<- read_excel("meat export aus.xls")
population<-read_excel("australia population.xls",skip = 3)
head(beef_and_veal_consumption)
```

LOCATION <chr>	INDICATOR <chr>	SUBJE… <chr>	MEAS… <chr>	FREQUEN… <chr>	TI… <int>	Value <dbl>	Flag Codes <chr>
AUS	MEATCONSUMP	BEEF	KG_CAP	A	1991	27.72182	*NA*
AUS	MEATCONSUMP	BEEF	KG_CAP	A	1992	26.19959	*NA*
AUS	MEATCONSUMP	BEEF	KG_CAP	A	1993	26.16909	*NA*
AUS	MEATCONSUMP	BEEF	KG_CAP	A	1994	25.45613	*NA*
AUS	MEATCONSUMP	BEEF	KG_CAP	A	1995	25.34023	*NA*
AUS	MEATCONSUMP	BEEF	KG_CAP	A	1996	27.25910	*NA*

6 rows

Hide

```
head(pork_consumption)
```

LOCATION <chr>	INDICATOR <chr>	SUBJE… <chr>	MEAS… <chr>	FREQUEN… <chr>	TI… <int>	Value <dbl>	Flag Codes <chr>
AUS	MEATCONSUMP	PIG	KG_CAP	A	1991	14.39648	*NA*
AUS	MEATCONSUMP	PIG	KG_CAP	A	1992	15.12925	*NA*
AUS	MEATCONSUMP	PIG	KG_CAP	A	1993	14.87068	*NA*
AUS	MEATCONSUMP	PIG	KG_CAP	A	1994	15.59669	*NA*
AUS	MEATCONSUMP	PIG	KG_CAP	A	1995	15.50472	*NA*
AUS	MEATCONSUMP	PIG	KG_CAP	A	1996	14.45952	*NA*

6 rows

Hide

```
head(lamb_consumption)
```

LOCATION <chr>	INDICATOR <chr>	SUBJE… <chr>	MEAS… <chr>	FREQUEN… <chr>	TI… <int>	Value <dbl>	Flag Codes <chr>
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1991	19.26239	*NA*
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1992	17.82159	*NA*
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1993	17.35676	*NA*

LOCATION <chr>	INDICATOR <chr>	SUBJE… <chr>	MEAS… <chr>	FREQUEN… <chr>	TI… <int>	Value <dbl>	Flag Codes <chr>
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1994	18.36657	*NA*
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1995	15.00546	*NA*
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1996	14.69113	*NA*

6 rows

Hide

```
head(redmeat_production)
```

X__1 <S3: POSIXct>	Meat Produced ; Total Red Meat ; Total (State) ; <dbl>
1972-07-01	209456
1972-08-01	216225
1972-09-01	192245
1972-10-01	204724
1972-11-01	212670
1972-12-01	187932

6 rows | 1-2 of 24 columns

Hide

```
head(pork_production)
```

X__1 <S3: POSIXct>	Meat Produced ; PIGS ; Total (State) ; <dbl>	Meat Produced ; PIG
1972-07-01	18028	
1972-08-01	20091	
1972-09-01	18718	
1972-10-01	20007	
1972-11-01	20629	
1972-12-01	17828	

6 rows | 1-3 of 24 columns

Hide

```
head(meat_export)
```

X__1 <S3: POSIXct>	Quantity ; Beef Bone In ; <dbl>	Quantity ; Beef Bone Out ; <dbl>	Quantit
1988-03-01	3937	154288	
1988-06-01	4204	152078	
1988-09-01	4410	136346	
1988-12-01	12297	132532	
1989-03-01	20984	98429	
1989-06-01	11978	131160	

6 rows | 1-4 of 8 columns

Hide

```
head(population)
```

Country Name <chr>	Country Code <chr>	Indicator Name <chr>	Indicator Code <chr>	1960 <dbl>	1961 <dbl>	19 <c
Aruba	ABW	Population, total	SP.POP.TOTL	54211	55438	562
Afghanistan	AFG	Population, total	SP.POP.TOTL	8996351	9166764	93458
Angola	AGO	Population, total	SP.POP.TOTL	5643182	5753024	58660
Albania	ALB	Population, total	SP.POP.TOTL	1608800	1659800	17113
Andorra	AND	Population, total	SP.POP.TOTL	13411	14375	153
Arab World	ARB	Population, total	SP.POP.TOTL	92490932	95044497	976822

6 rows | 1-8 of 62 columns

Hide

```
#change value variables to the according consumption
colnames(pork_consumption)
```

```
[1] "LOCATION"   "INDICATOR"  "SUBJECT"    "MEASURE"    "FREQUENCY"  "TIME"       "Val
ue"
[8] "Flag Codes"
```

Hide

```
 pork_consumption<-rename(pork_consumption, Pork_cons=Value)
beef_and_veal_consumption<-  rename(beef_and_veal_consumption, beef_and_veal_cons=Valu
e)
lamb_consumption<- rename(lamb_consumption, lamb_cons=Value)

 #mearging
meat_cons_draft<- lamb_consumption %>% left_join(beef_and_veal_consumption, by =c("TIM
E","LOCATION")) %>%left_join(pork_consumption,by=c("TIME","LOCATION"))
head(meat_cons_draft)
```

LOCAT… <chr>	INDICATOR.x <chr>	SUBJEC… <chr>	MEASU… <chr>	FREQUEN… <chr>	T… <int>	lamb_co… <dbl>	Flag Codes.x <chr>	IND <ch
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1991	19.26239	*NA*	ME
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1992	17.82159	*NA*	ME
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1993	17.35676	*NA*	ME
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1994	18.36657	*NA*	ME
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1995	15.00546	*NA*	ME
AUS	MEATCONSUMP	SHEEP	KG_CAP	A	1996	14.69113	*NA*	ME

6 rows | 1-9 of 20 columns

Hide

```
meat_cons<-dplyr::select(meat_cons_draft,TIME,LOCATION, beef_and_veal_cons,lamb_cons,P
ork_cons, MEASURE)

 head(meat_cons)
```

TIME <int>	LOCATION <chr>	beef_and_veal_cons <dbl>	lamb_cons <dbl>	Pork_cons <dbl>	MEASURE <chr>
1991	AUS	27.72182	19.26239	14.39648	KG_CAP
1992	AUS	26.19959	17.82159	15.12925	KG_CAP
1993	AUS	26.16909	17.35676	14.87068	KG_CAP
1994	AUS	25.45613	18.36657	15.59669	KG_CAP
1995	AUS	25.34023	15.00546	15.50472	KG_CAP
1996	AUS	27.25910	14.69113	14.45952	KG_CAP

6 rows

Hide

```
 #meat production dataset, merging pork_production and beef_production
 meat_prod_draft<- pork_production %>% left_join(redmeat_production,by=c("X__1"))
head(meat_prod_draft)
```

X__1 <S3: POSIXct>	Meat Produced ; PIGS ; Total (State) ; <dbl>	Meat Produced ; PIG
1972-07-01	18028	
1972-08-01	20091	
1972-09-01	18718	
1972-10-01	20007	
1972-11-01	20629	
1972-12-01	17828	

6 rows | 1-3 of 47 columns

Hide

```
str(meat_prod_draft)
```

```
Classes tbl_df, tbl and 'data.frame':   549 obs. of  47 variables:
 $ X__1                                                  : POSIXct, format:
"1972-07-01" "1972-08-01" "1972-09-01" ...
 $ Meat Produced ;  PIGS ;  Total (State) ;              : num  18028 20091
18718 20007 20629 ...
 $ Meat Produced ;  PIGS ;  New South Wales ;            : num  4384 5117 4
911 4877 5417 ...
 $ Meat Produced ;  PIGS ;  Victoria ;                   : num  4655 5140 4
755 5744 5194 ...
 $ Meat Produced ;  PIGS ;  Queensland ;                 : num  4018 4623 3
927 3756 4499 ...
 $ Meat Produced ;  PIGS ;  South Australia ;            : num  2102 2403 2
218 2360 2301 ...
 $ Meat Produced ;  PIGS ;  Western Australia ;          : num  2275 2126 2
229 2512 2437 ...
 $ Meat Produced ;  PIGS ;  Tasmania ;                   : num  523 614 600
690 693 694 600 627 657 545 ...
 $ Meat Produced ;  PIGS ;  Northern Territory ;         : num  7 10 12 10
11 19 14 19 19 5 ...
 $ Meat Produced ;  PIGS ;  Australian Capital Territory ;  : num  64 58 66 58
77 76 70 79 88 80 ...
 $ Meat Produced ;  PIGS ;  Total (State) ;__1           : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  New South Wales ;__1         : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Victoria ;__1                : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Queensland ;__1              : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  South Australia ;__1         : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Western Australia ;__1       : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Tasmania ;__1                : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Total (State) ;__2           : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  New South Wales ;__2         : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Victoria ;__2                : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Queensland ;__2              : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  South Australia ;__2         : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Western Australia ;__2       : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  PIGS ;  Tasmania ;__2                : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Total (State) ;    : num  209456 2162
```

```
25 192245 204724 212670 ...
 $ Meat Produced ;  Total Red Meat ;  New South Wales ;             : num  58469 58462
52360 53258 60018 ...
 $ Meat Produced ;  Total Red Meat ;  Victoria ;                    : num  63992 67262
59556 70780 75729 ...
 $ Meat Produced ;  Total Red Meat ;  Queensland ;                  : num  48697 52003
43267 35750 30851 ...
 $ Meat Produced ;  Total Red Meat ;  South Australia ;             : num  13698 13766
13790 17850 16856 ...
 $ Meat Produced ;  Total Red Meat ;  Western Australia ;           : num  16182 16248
16077 19675 20936 ...
 $ Meat Produced ;  Total Red Meat ;  Tasmania ;                    : num  5521 5474 4
947 5414 6583 ...
 $ Meat Produced ;  Total Red Meat ;  Northern Territory ;          : num  2265 2377 1
679 1388 1005 ...
 $ Meat Produced ;  Total Red Meat ;  Australian Capital Territory ;: num  632 633 569
609 692 593 753 602 658 622 ...
 $ Meat Produced ;  Total Red Meat ;  Total (State) ;__1            : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  New South Wales ;__1          : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Victoria ;__1                 : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Queensland ;__1               : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  South Australia ;__1          : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Western Australia ;__1        : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Tasmania ;__1                 : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Total (State) ;__2            : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  New South Wales ;__2          : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Victoria ;__2                 : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Queensland ;__2               : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  South Australia ;__2          : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Western Australia ;__2        : num  NA NA NA NA
NA NA NA NA NA NA ...
 $ Meat Produced ;  Total Red Meat ;  Tasmania ;__2                 : num  NA NA NA NA
NA NA NA NA NA NA ...
```

Hide

```
meat_prod_draft_2 <- dplyr::select(meat_prod_draft,X__1,"Meat Produced ;  PIGS ;  Tota
l (State) ;","Meat Produced ;  Total Red Meat ;  Total (State) ;" )
meat_prod_aus<- rename(meat_prod_draft_2, red_meat_prod="Meat Produced ;  PIGS ;  Tota
l (State) ;",pork_prod="Meat Produced ;  Total Red Meat ;  Total (State) ;")
head(meat_prod_aus)
```

X__1 <S3: POSIXct>	red_meat_prod <dbl>	pork_prod <dbl>
1972-07-01	18028	209456
1972-08-01	20091	216225
1972-09-01	18718	192245
1972-10-01	20007	204724
1972-11-01	20629	212670
1972-12-01	17828	187932

6 rows

Hide

```
population<- population %>% dplyr::select("Country Name","Country Code","1991":"2017")
meat_cons<- rename(meat_cons, 'Country Code'='LOCATION')
meat_cons<- rename(meat_cons, 'year'='TIME')
#the rest of the report will be working on these 2 data sets, they will be joined late
r for some insights analyis
```

Understand

Hide

```
str(meat_cons)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  6 variables:
 $ year             : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ Country Code     : chr  "AUS" "AUS" "AUS" "AUS" ...
 $ beef_and_veal_cons: num  27.7 26.2 26.2 25.5 25.3 ...
 $ lamb_cons        : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork_cons        : num  14.4 15.1 14.9 15.6 15.5 ...
 $ MEASURE          : chr  "KG_CAP" "KG_CAP" "KG_CAP" "KG_CAP" ...
```

Hide

```
meat_cons$beef_and_veal_cons<-as.double(meat_cons$beef_and_veal_cons)
meat_cons$lamb_cons<-as.double(meat_cons$lamb_cons)
meat_cons$Pork_cons<-as.double(meat_cons$Pork_cons)
str(meat_cons)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  6 variables:
 $ year             : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ Country Code     : chr  "AUS" "AUS" "AUS" "AUS" ...
 $ beef_and_veal_cons: num  27.7 26.2 26.2 25.5 25.3 ...
 $ lamb_cons        : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork_cons        : num  14.4 15.1 14.9 15.6 15.5 ...
 $ MEASURE          : chr  "KG_CAP" "KG_CAP" "KG_CAP" "KG_CAP" ...
```

Hide

```
str(population)
```

```
Classes tbl_df, tbl and 'data.frame':   264 obs. of  29 variables:
 $ Country Name: chr  "Aruba" "Afghanistan" "Angola" "Albania" ...
 $ Country Code: chr  "ABW" "AFG" "AGO" "ALB" ...
 $ 1991        : num  64622 12993657 12553446 3266790 56671 ...
 $ 1992        : num  68235 13981231 12968345 3247039 58888 ...
 $ 1993        : num  72504 15095099 13403734 3227287 60971 ...
 $ 1994        : num  76700 16172719 13841301 3207536 62677 ...
 $ 1995        : num  80324 17099541 14268994 3187784 63850 ...
 $ 1996        : num  83200 17822884 14682284 3168033 64360 ...
 $ 1997        : num  85451 18381605 15088981 3148281 64327 ...
 $ 1998        : num  87277 18863999 15504318 3128530 64142 ...
 $ 1999        : num  89005 19403676 15949766 3108778 64370 ...
 $ 2000        : num  90853 20093756 16440924 3089027 65390 ...
 $ 2001        : num  92898 20966463 16983266 3060173 67341 ...
 $ 2002        : num  94992 21979923 17572649 3051010 70049 ...
 $ 2003        : num  97017 23064851 18203369 3039616 73182 ...
 $ 2004        : num  98737 24118979 18865716 3026939 76244 ...
 $ 2005        : num  100031 25070798 19552542 3011487 78867 ...
 $ 2006        : num  100832 25893450 20262399 2992547 80991 ...
 $ 2007        : num  101220 26616792 20997687 2970017 82683 ...
 $ 2008        : num  101353 27294031 21759420 2947314 83861 ...
 $ 2009        : num  101453 28004331 22549547 2927519 84462 ...
 $ 2010        : num  101669 28803167 23369131 2913021 84449 ...
 $ 2011        : num  102053 29708599 24218565 2905195 83751 ...
 $ 2012        : num  102577 30696958 25096150 2900401 82431 ...
 $ 2013        : num  103187 31731688 25998340 2895092 80788 ...
 $ 2014        : num  103795 32758020 26920466 2889104 79223 ...
 $ 2015        : num  104341 33736494 27859305 2880703 78014 ...
 $ 2016        : num  104822 34656032 28813463 2876101 77281 ...
 $ 2017        : logi  NA NA NA NA NA NA ...
```

Hide

```
population$`2017`<- as.numeric(population$`2017`)
str(population)
```

```
Classes tbl_df, tbl and 'data.frame':   264 obs. of  29 variables:
 $ Country Name: chr  "Aruba" "Afghanistan" "Angola" "Albania" ...
 $ Country Code: chr  "ABW" "AFG" "AGO" "ALB" ...
 $ 1991        : num  64622 12993657 12553446 3266790 56671 ...
 $ 1992        : num  68235 13981231 12968345 3247039 58888 ...
 $ 1993        : num  72504 15095099 13403734 3227287 60971 ...
 $ 1994        : num  76700 16172719 13841301 3207536 62677 ...
 $ 1995        : num  80324 17099541 14268994 3187784 63850 ...
 $ 1996        : num  83200 17822884 14682284 3168033 64360 ...
 $ 1997        : num  85451 18381605 15088981 3148281 64327 ...
 $ 1998        : num  87277 18863999 15504318 3128530 64142 ...
 $ 1999        : num  89005 19403676 15949766 3108778 64370 ...
 $ 2000        : num  90853 20093756 16440924 3089027 65390 ...
 $ 2001        : num  92898 20966463 16983266 3060173 67341 ...
 $ 2002        : num  94992 21979923 17572649 3051010 70049 ...
 $ 2003        : num  97017 23064851 18203369 3039616 73182 ...
 $ 2004        : num  98737 24118979 18865716 3026939 76244 ...
 $ 2005        : num  100031 25070798 19552542 3011487 78867 ...
 $ 2006        : num  100832 25893450 20262399 2992547 80991 ...
 $ 2007        : num  101220 26616792 20997687 2970017 82683 ...
 $ 2008        : num  101353 27294031 21759420 2947314 83861 ...
 $ 2009        : num  101453 28004331 22549547 2927519 84462 ...
 $ 2010        : num  101669 28803167 23369131 2913021 84449 ...
 $ 2011        : num  102053 29708599 24218565 2905195 83751 ...
 $ 2012        : num  102577 30696958 25096150 2900401 82431 ...
 $ 2013        : num  103187 31731688 25998340 2895092 80788 ...
 $ 2014        : num  103795 32758020 26920466 2889104 79223 ...
 $ 2015        : num  104341 33736494 27859305 2880703 78014 ...
 $ 2016        : num  104822 34656032 28813463 2876101 77281 ...
 $ 2017        : num  NA NA NA NA NA NA NA NA NA NA ...
```

Tidy & Manipulate Data I

Hide

```
pop<-population %>% gather('1991','1992','1993','1994','1995','1996','1997','1998','19
99','2000','2001','2002','2003','2004','2005','2006','2007','2008','2009','2010','2011
','2012','2013','2014','2015','2016','2017', key = "year", value = "population")
str(pop)
```

```
Classes tbl_df, tbl and 'data.frame':   7128 obs. of  4 variables:
 $ Country Name: chr  "Aruba" "Afghanistan" "Angola" "Albania" ...
 $ Country Code: chr  "ABW" "AFG" "AGO" "ALB" ...
 $ year        : chr  "1991" "1991" "1991" "1991" ...
 $ population  : num  64622 12993657 12553446 3266790 56671 ...
```

Hide

```
pop$year<- as.integer(pop$year)
str(pop)
```

```
Classes tbl_df, tbl and 'data.frame':   7128 obs. of  4 variables:
 $ Country Name: chr  "Aruba" "Afghanistan" "Angola" "Albania" ...
 $ Country Code: chr  "ABW" "AFG" "AGO" "ALB" ...
 $ year        : int  1991 1991 1991 1991 1991 1991 1991 1991 1991 1991 ...
 $ population  : num  64622 12993657 12553446 3266790 56671 ...
```

Hide

```
# joining meat consumtion to countries population by year and Country Code
meat_cons_pop<- meat_cons %>% left_join(pop, by= c("year","Country Code"))
str(meat_cons_pop)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  8 variables:
 $ year              : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ Country Code      : chr  "AUS" "AUS" "AUS" "AUS" ...
 $ beef_and_veal_cons: num  27.7 26.2 26.2 25.5 25.3 ...
 $ lamb_cons         : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork_cons         : num  14.4 15.1 14.9 15.6 15.5 ...
 $ MEASURE           : chr  "KG_CAP" "KG_CAP" "KG_CAP" "KG_CAP" ...
 $ Country Name      : chr  "Australia" "Australia" "Australia" "Australia" ...
 $ population        : num  17284000 17495000 17667000 17855000 18072000 ...
```

Hide

```
str(meat_cons_pop)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  8 variables:
 $ year              : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ Country Code      : chr  "AUS" "AUS" "AUS" "AUS" ...
 $ beef_and_veal_cons: num  27.7 26.2 26.2 25.5 25.3 ...
 $ lamb_cons         : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork_cons         : num  14.4 15.1 14.9 15.6 15.5 ...
 $ MEASURE           : chr  "KG_CAP" "KG_CAP" "KG_CAP" "KG_CAP" ...
 $ Country Name      : chr  "Australia" "Australia" "Australia" "Australia" ...
 $ population        : num  17284000 17495000 17667000 17855000 18072000 ...
```

Hide

```
#change pork consumption to numeric
meat_cons_pop$Pork_cons<- as.numeric(meat_cons_pop$Pork_cons)
#change country code and MEASURE to factors
meat_cons_pop$`Country Code`<-as.factor(meat_cons_pop$`Country Code`)
meat_cons_pop$MEASURE<-as.factor(meat_cons_pop$MEASURE)
str(meat_cons_pop)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  8 variables:
 $ year              : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ Country Code      : Factor w/ 46 levels "ARG","AUS","BGD",..: 2 2 2 2 2 2 2 2 2 .
..
 $ beef_and_veal_cons: num  27.7 26.2 26.2 25.5 25.3 ...
 $ lamb_cons         : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork_cons         : num  14.4 15.1 14.9 15.6 15.5 ...
 $ MEASURE           : Factor w/ 1 level "KG_CAP": 1 1 1 1 1 1 1 1 1 1 ...
 $ Country Name      : chr  "Australia" "Australia" "Australia" "Australia" ...
 $ population        : num  17284000 17495000 17667000 17855000 18072000 ...
```

Hide

```
head(meat_cons_pop)
```

y…	Country Code	beef_and_veal_cons	lamb_co…	Pork_co…	MEA…	Country Name	popu
<int>	<fctr>	<dbl>	<dbl>	<dbl>	<fctr>	<chr>	
1991	AUS	27.72182	19.26239	14.39648	KG_CAP	Australia	172
1992	AUS	26.19959	17.82159	15.12925	KG_CAP	Australia	174
1993	AUS	26.16909	17.35676	14.87068	KG_CAP	Australia	176
1994	AUS	25.45613	18.36657	15.59669	KG_CAP	Australia	178
1995	AUS	25.34023	15.00546	15.50472	KG_CAP	Australia	180
1996	AUS	27.25910	14.69113	14.45952	KG_CAP	Australia	183

6 rows

Tidy & Manipulate Data II

Hide

```
meat_cons_pop<-mutate(meat_cons_pop,total_meat_cons_per_capita= beef_and_veal_cons+ la
mb_cons+ Pork_cons, volume_of_meat_cons_in_tonne= total_meat_cons_per_capita*populatio
n/1000)
meat_cons_pop<- rename(meat_cons_pop, 'Year'='year')
meat_cons_pop<- rename(meat_cons_pop, 'Population (people)'='population')
meat_cons_pop<- rename(meat_cons_pop, 'Beef and veal consumption (kg/capita)'='beef_an
d_veal_cons')
meat_cons_pop<- rename(meat_cons_pop, 'Lamb consumption (kg/capita)'='lamb_cons')
meat_cons_pop<- rename(meat_cons_pop, 'Pork consumption (kg/capita)'='Pork_cons')
meat_cons_pop<- rename(meat_cons_pop, 'Meat consumption (kg/capita)'='total_meat_cons_
per_capita')
meat_cons_pop<- rename(meat_cons_pop, 'Country meat consumption (tonnes)'='volume_of_m
eat_cons_in_tonne')
head(meat_cons_pop)
```

Y…	Country Code	Beef and veal consumption (kg/capita)	Lamb consumption (kg/c

6 rows | 1-4 of 10 columns

Hide

```
#drop the variable MEASURE and reaggrage all the variables
meat_cons_pop<-meat_cons_pop[,-6]
head(meat_cons_pop)
```

Y…	Country Code	Beef and veal consumption (kg/capita)	Lamb consumption (kg/c
<int>	<fctr>	<dbl>	
1991	AUS	27.72182	19.
1992	AUS	26.19959	17.
1993	AUS	26.16909	17.
1994	AUS	25.45613	18.
1995	AUS	25.34023	15.
1996	AUS	27.25910	14.

6 rows | 1-4 of 9 columns

Hide

```
#rearrange variables for easier reading
meat_cons_pop <- meat_cons_pop[c(6,1,7,3,4,5,8,9,2)]
str(meat_cons_pop)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  9 variables:
 $ Country Name                          : chr  "Australia" "Australia" "Australia" "Au
stralia" ...
 $ Year                                  : int  1991 1992 1993 1994 1995 1996 1997 1998
1999 2000 ...
 $ Population (people)                   : num  17284000 17495000 17667000 17855000 180
72000 ...
 $ Beef and veal consumption (kg/capita): num  27.7 26.2 26.2 25.5 25.3 ...
 $ Lamb consumption (kg/capita)          : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork consumption (kg/capita)          : num  14.4 15.1 14.9 15.6 15.5 ...
 $ Meat consumption (kg/capita)          : num  61.4 59.2 58.4 59.4 55.9 ...
 $ Country meat consumption (tonnes)     : num  1060904 1034837 1031691 1060933 1009328
...
 $ Country Code                          : Factor w/ 46 levels "ARG","AUS","BGD",..: 2
2 2 2 2 2 2 2 2 ...
```

Hide

```
meat_cons_pop$`Pork consumption (kg/capita)`<- as.numeric(meat_cons_pop$`Pork consumption (kg/capita)`)
head(meat_cons_pop)
```

Country Name <chr>	Y… <int>	Population (people) <dbl>	Beef and veal consumption (kg/capita) <dbl>
Australia	1991	17284000	27.72182
Australia	1992	17495000	26.19959
Australia	1993	17667000	26.16909
Australia	1994	17855000	25.45613
Australia	1995	18072000	25.34023
Australia	1996	18311000	27.25910

6 rows | 1-4 of 9 columns

Hide

```
str(meat_cons_pop)
```

```
Classes tbl_df, tbl and 'data.frame':   1235 obs. of  9 variables:
 $ Country Name                          : chr  "Australia" "Australia" "Australia" "Australia" ...
 $ Year                                  : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ Population (people)                   : num  17284000 17495000 17667000 17855000 18072000 ...
 $ Beef and veal consumption (kg/capita): num  27.7 26.2 26.2 25.5 25.3 ...
 $ Lamb consumption (kg/capita)          : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork consumption (kg/capita)          : num  14.4 15.1 14.9 15.6 15.5 ...
 $ Meat consumption (kg/capita)          : num  61.4 59.2 58.4 59.4 55.9 ...
 $ Country meat consumption (tonnes)     : num  1060904 1034837 1031691 1060933 1009328 ...
 $ Country Code                          : Factor w/ 46 levels "ARG","AUS","BGD",..: 2 2 2 2 2 2 2 2 2 2 ...
```

Scan I

Hide

```
sum(is.na(meat_cons_pop))
```

```
[1] 392
```

Hide

```
#missing countries name
which(is.na(meat_cons_pop$`Country Name`))
```

```
 [1] 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1
172 1173
[19] 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1
190 1191
[37] 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1
208 1209
[55] 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1
226 1227
[73] 1228 1229 1230 1231 1232 1233 1234 1235
```

Hide

```
#Chose to delete all observation with missing countries names, because all these oberv
ation represents groups of coutries such as : EU, OCEC,etc
meat_cons_pop<-meat_cons_pop[-which(is.na(meat_cons_pop$`Country Name`)),]
sum(is.na(meat_cons_pop$`Country Name`))
```

```
[1] 0
```

Hide

```
which(is.na(meat_cons_pop$Year))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Beef and veal consumption (kg/capita)`))
```

```
[1] 1103 1104 1105 1106 1107 1108 1109 1110
```

Hide

```
which(is.na(meat_cons_pop$`Lamb consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Pork consumption (kg/capita)`))
```

```
[1]  835 1103
```

Hide

```
#the missing values in beef_and_veal_cons belong the wolrd, since this will create som
e confusion for analysis such as calculating total amount of meat consumpted, I choose
to delete these observations.
meat_cons_pop<-meat_cons_pop[-(1103:1155), ]
which(is.na(meat_cons_pop$Year))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Beef and veal consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Lamb consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Pork consumption (kg/capita)`))
```

```
[1] 835
```

Hide

```
#in Pork_cons from row 835, relacing with the mean consumprion of that country over th
e year
russian<- meat_cons_pop %>% filter(`Country Name`=="Russian Federation")
mean_pork<- impute(russian$`Pork consumption (kg/capita)`, fun = mean)
meat_cons_pop$`Pork consumption (kg/capita)`[is.na(meat_cons_pop$`Pork consumption (kg
/capita)`)]=14.990506
which(is.na(meat_cons_pop$`Pork consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Country Code`))
```

```
integer(0)
```

Hide

```
which(is.na(meat_cons_pop$`Meat consumption (kg/capita)`))
```

```
[1] 835
```

Hide

```
#Russian Ferdaration is missing because the value of Pork consumption (kg/capita) was
recently added, I will use validate package at this step to make sure rules are apply
and fill the missing values
which(is.na(meat_cons_pop$`Country meat consumption (tonnes)`))
```

```
 [1]   27   54   81  108  135  162  189  216  243  270  297  324  351  378  405  432
457  484
[19]  511  538  565  592  619  645  672  699  726  753  780  807  834  835  860  887
914  941
[37]  968  995 1021 1048 1075 1102
```

Hide

```
which(is.na(meat_cons_pop$`Population (people)`))
```

```
 [1]   27   54   81  108  135  162  189  216  243  270  297  324  351  378  405  432
457  484
[19]  511  538  565  592  619  645  672  699  726  753  780  807  834  860  887  914
941  968
[37]  995 1021 1048 1075 1102
```

Hide

```
#ignoring Population (people) and Country meat consumption (tonnes)  for the year 2017
- will attempt to solve this later
#Russian Ferdaration is missing because the value of Pork consumption (kg/capita) was
recently added, I will use validate package at this step to make sure rules are apply
and fill the missing values
which(is.na(meat_cons_pop$`Meat consumption (kg/capita)`))
```

```
[1] 835
```

Hide

```
Rules <- validator(`Beef and veal consumption (kg/capita)` + `Pork consumption (kg/cap
ita)` + `Lamb consumption (kg/capita)` == `Meat consumption (kg/capita)`, `Beef and ve
al consumption (kg/capita)` >= 0,`Lamb consumption (kg/capita)` >= 0, `Pork consumptio
n (kg/capita)` >= 0, `Population (people)`*`Meat consumption (kg/capita)`/1000==`Count
ry meat consumption (tonnes)`,`Population (people)`>0)
str(meat_cons_pop)
```

```
Classes tbl_df, tbl and 'data.frame':   1102 obs. of  9 variables:
 $ Country Name                          : chr  "Australia" "Australia" "Australia" "Au
stralia" ...
 $ Year                                  : int  1991 1992 1993 1994 1995 1996 1997 1998
1999 2000 ...
 $ Population (people)                   : num  17284000 17495000 17667000 17855000 180
72000 ...
 $ Beef and veal consumption (kg/capita): num  27.7 26.2 26.2 25.5 25.3 ...
 $ Lamb consumption (kg/capita)          : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork consumption (kg/capita)          : num  14.4 15.1 14.9 15.6 15.5 ...
 $ Meat consumption (kg/capita)          : num  61.4 59.2 58.4 59.4 55.9 ...
 $ Country meat consumption (tonnes)     : num  1060904 1034837 1031691 1060933 1009328
...
 $ Country Code                          : Factor w/ 46 levels "ARG","AUS","BGD",..: 2
2 2 2 2 2 2 2 2 ...
```

Hide

```
#use the rule to missing data
meat_cons_pop <- impute_lr(meat_cons_pop,Rules)
which(is.na(meat_cons_pop$`Meat consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
#check for infinite values
which(is.infinite(meat_cons_pop$`Pork consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.infinite(meat_cons_pop$Year))
```

```
integer(0)
```

Hide

```
which(is.infinite(meat_cons_pop$`Beef and veal consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.infinite(meat_cons_pop$`Lamb consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.infinite(meat_cons_pop$`Meat consumption (kg/capita)`))
```

```
integer(0)
```

Hide

```
which(is.infinite(meat_cons_pop$`Country meat consumption (tonnes)`))
```

```
integer(0)
```

Hide

```
#removing white space in Country name
meat_cons_pop$`Country Name` <- str_trim(meat_cons_pop$`Country Name`,side = "both")
#clean up Country Code and Country Namemake sure they are both factors with the same a
mount of levels- 1 country to 1 country code
table(meat_cons_pop$`Country Name`)
```

```
           Algeria          Argentina          Australia         Bangladesh
Brazil
                27                 27                 27                 27
27
            Canada              Chile              China           Colombia   Egypt, A
rab Rep.
                27                 27                 27                 27
27
          Ethiopia              Ghana              Haiti              India          I
ndonesia
                25                 27                 27                 27
27
Iran, Islamic Rep.             Israel              Japan         Kazakhstan        Kor
ea, Rep.
                27                 27                 27                 26
27
          Malaysia             Mexico         Mozambique        New Zealand
Nigeria
                27                 27                 27                 27
27
          Pakistan           Paraguay               Peru        Philippines Russian Fe
deration
                27                 27                 27                 27
26
      Saudi Arabia       South Africa              Sudan           Tanzania
Thailand
                27                 27                 27                 27
27
            Turkey            Ukraine      United States            Uruguay
Vietnam
                27                 26                 27                 27
27
            Zambia
                27
```

Hide

```
table(meat_cons_pop$`Country Code`)
```

```
  ARG   AUS   BGD   BRA BRICS   CAN   CHL   CHN   COL   DZA   EGY   ETH  EU28   GHA
HTI   IDN
   27    27    27    27     0    27    27    27    27    27    27    25     0    27
27    27
  IND   IRN   ISR   JPN   KAZ   KOR   MEX   MOZ   MYS   NGA   NZL  OECD   PAK   PER
PHL   PRY
   27    27    27    27    26    27    27    27    27    27    27     0    27    27
27    27
  RUS   SAU   SDN   SSA   THA   TUR   TZA   UKR   URY   USA   VNM   WLD   ZAF   ZMB
   26    27    27     0    27    27    27    26    27    27    27     0    27    27
```

Hide

```
meat_cons_pop$`Country Code`<- as.character(meat_cons_pop$`Country Code`)
table(meat_cons_pop$`Country Code`)
```

```
ARG AUS BGD BRA CAN CHL CHN COL DZA EGY ETH GHA HTI IDN IND IRN ISR JPN KAZ KOR MEX MO
Z MYS NGA
 27  27  27  27  27  27  27  27  27  27  25  27  27  27  27  27  27  27  26  27  27  2
7  27  27
NZL PAK PER PHL PRY RUS SAU SDN THA TUR TZA UKR URY USA VNM ZAF ZMB
 27  27  27  27  27  26  27  27  27  27  27  26  27  27  27  27  27
```

Hide

```
which(is.na(meat_cons_pop$`Country Code`))
```

```
integer(0)
```

Hide

```
meat_cons_pop$`Country Code`<- as.factor(meat_cons_pop$`Country Code`)
str(meat_cons_pop$`Country Code`)
```

```
 Factor w/ 41 levels "ARG","AUS","BGD",..: 2 2 2 2 2 2 2 2 2 2 ...
```

Hide

```
meat_cons_pop$`Country Name`<- as.factor(meat_cons_pop$`Country Name`)
str(meat_cons_pop$`Country Name`)
```

```
 Factor w/ 41 levels "Algeria","Argentina",..: 3 3 3 3 3 3 3 3 3 3 ...
```

Scan II

Hide

```
#Investigate numeric variables for outliers - except Country meat consumption, because this value has many missing value in the year 2017 due to the missing in formation of Population in 2017
zscores_beef <- meat_cons_pop$`Beef and veal consumption (kg/capita)` %>%  scores(type = "z")
zscores_beef %>% summary()
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-0.9244 -0.6657 -0.4570  0.0000  0.4054  4.7655 
```

Hide

```
which( abs(zscores_beef) >3 )
```

```
 [1]  244  245  246  247  250  252  253  260  261  262 1022 1023 1024 1025 1026 1027 1028 1029
[19] 1030 1031 1033 1036 1045 1046
```

Hide

```
zscores_pork <- meat_cons_pop$`Pork consumption (kg/capita)` %>%  scores(type = "z")
zscores_pork %>% summary()
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-0.9360 -0.9004 -0.4437  0.0000  0.6990  3.1010 
```

Hide

```
which( abs(zscores_pork) >3 )
```

```
[1] 375
```

Hide

```
zscores_lamb <- meat_cons_pop$`Lamb consumption (kg/capita)` %>%  scores(type = "z")
zscores_lamb %>% summary()
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
-0.58480 -0.46953 -0.35367  0.00000  0.01835  8.99463 
```

Hide

```
which( abs(zscores_lamb) >3 )
```

```
 [1]    1    2    3    4    5    6    7    8    9   10   11  136  137  138  139  140
141  142
[19]  143  144  145  146  147  148  150  151  152  153 1023 1024
```

Hide

```
zscores_meat <- meat_cons_pop$`Meat consumption (kg/capita)` %>%  scores(type = "z")
zscores_meat %>% summary()
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.1565 -0.7704 -0.3530  0.0000  0.6366  3.6761
```

Hide

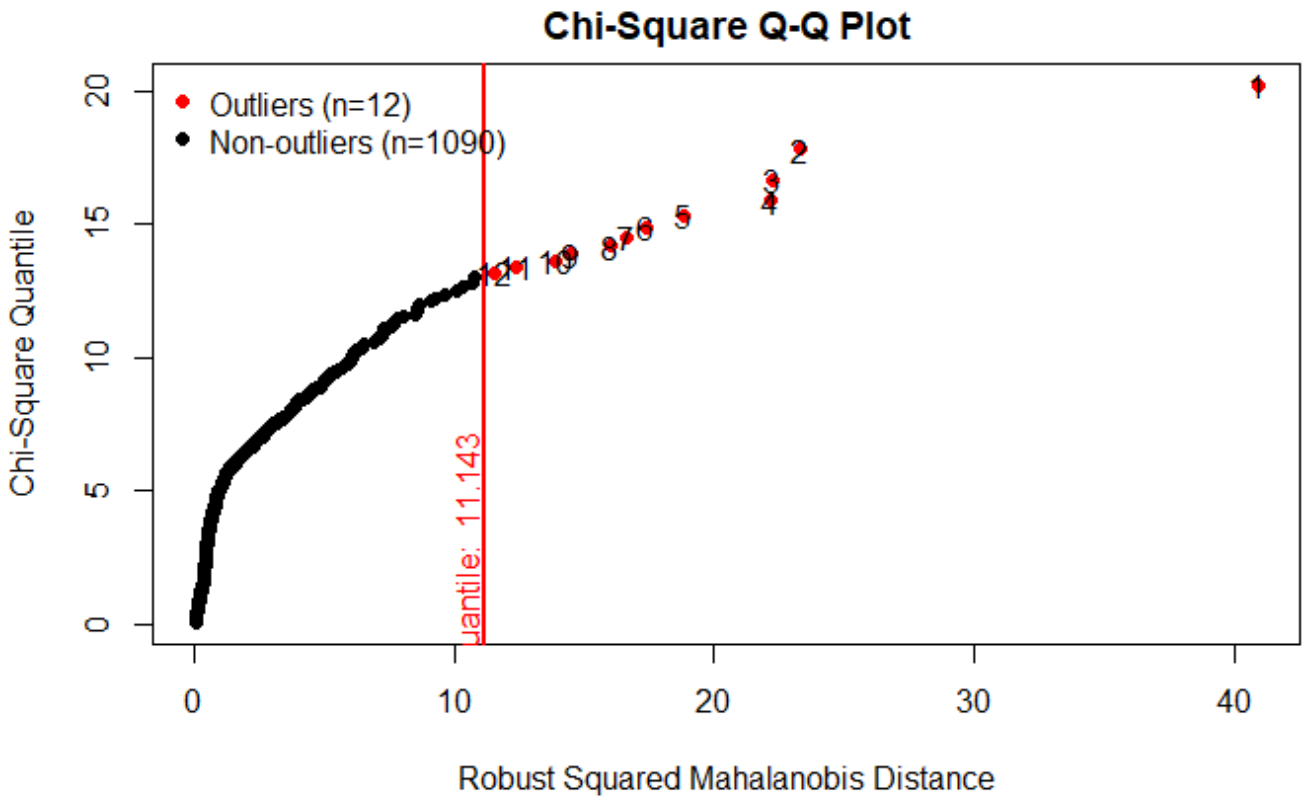
```
which( abs(zscores_meat) >3 )
```

```
[1] 1022 1023 1025
```

Hide

```
#create a subset of the meat_cons_pop data set to analyse multivariate outliers
meat_cons_pop_subset<- meat_cons_pop %>% dplyr::select("Beef and veal consumption (kg/
capita)","Pork consumption (kg/capita)","Lamb consumption (kg/capita)","Meat consumpti
on (kg/capita)")
multi_outliers <- mvn(data = meat_cons_pop_subset, multivariateOutlierMethod = "quan",
showOutliers = TRUE)
```

```
The covariance matrix of the data is singular.
There are 1102 observations (in the entire dataset of 1102 obs.) lying on the
hyperplane with equation a_1*(x_i1 - m_1) + ... + a_p*(x_ip - m_p) = 0 with (m_1, ...,
m_p) the mean of these observations and coefficients a_i from the vector a <- c(-0.5,
-0.5, -0.5, 0.5)
```



Hide

```
multi_outliers$multivariateOutliers
```

	Observation <fctr>	Mahalanobis Distance <dbl>	Outlier <chr>
1	1	40.893	TRUE
2	2	23.341	TRUE
3	3	22.270	TRUE
4	4	22.161	TRUE
5	5	18.810	TRUE
6	6	17.387	TRUE
7	7	16.662	TRUE
8	8	16.015	TRUE
9	9	14.519	TRUE
10	10	13.873	TRUE

1-10 of 12 rows Previous **1** 2 Next

Hide

```
#12 outliers were found
#These outliers should not be removed or replaced, these are good points of observatio
ns for examples: finding out which countries out consumed everyone else. Excluding or
impute these outliers can lead to some over generalisations of the data, which possibl
e lead to underfitting in future analysis.
#However I'll perform some handling techniques here just to show the possibilities.
library(outliers)
beef_veal_clean<- meat_cons_pop$`Beef and veal consumption (kg/capita)`[ - which( abs(
zscores_beef) >3 )]
lamb_clean<- meat_cons_pop$`Lamb consumption (kg/capita)`[ - which( abs(zscores_lamb)
>3 )]
pork_clean<- meat_cons_pop$`Pork consumption (kg/capita)`[ - which( abs(zscores_pork)
>3 )]
meat_clean<- meat_cons_pop$`Meat consumption (kg/capita)`[ - which( abs(zscores_meat)
>3 )]
lamb_clean
```

```
   [1] 13.699329624 12.084480410 11.920952053 11.935469129 12.322278981 12.550873524 1
0.972332261
   [8] 10.324144008  8.969861448  8.044461783  8.720105015  8.716901165  8.107499586
9.331542874
  [15]  8.550365388  8.537929538  0.764272223  0.741148210  0.752332884  0.723221435
0.713323289
  [22]  0.664067854  0.651948044  0.730094276  0.753867099  0.829785698  0.911196365
0.910415471
  [29]  0.952507759  0.973777595  0.982951162  1.066423447  1.070975244  1.024720129
1.013341834
  [36]  0.954364265  0.932547495  0.848751630  0.879485148  0.924069666  0.909873857
0.930115655
  [43]  0.885564286  0.772784737  0.811072174  0.683020401  0.571794327  0.585048734
0.504681216
  [50]  0.444474214  0.425857203  0.357453177  0.324378998  0.310996772  0.297423731
0.255047498
  [57]  0.322503797  0.367721359  0.370241212  0.266869910  0.269907059  0.272553472
0.218548879
  [64]  0.220323466  0.190034662  0.125903410  0.141513953  0.132097181  0.146290821
0.147081018
  [71]  0.157340811  0.299883723  0.239479192  0.275852555  0.255908013  0.214451912
0.193039231
  [78]  0.154387424  0.134180509  0.114270204  0.094322111  0.112569650  0.111644734
0.111577983
  [85]  0.110909045  0.109924686  0.109530422  0.108842259  0.108179501  0.089629601
0.106974771
  [92]  0.106431882  0.105923057  0.140589216  0.174971546  0.191666673  0.192642272
0.595611299
  [99]  0.636915864  0.644664886  0.668995778  0.411689439  0.401294788  0.491592867
0.512224291
 [106]  0.580432640  0.687270364  0.727653473  0.732683948  0.977859783  0.839348936
0.689909086
 [113]  0.661304067  0.671650756  0.656185765  0.582026716  0.540572705  0.515065793
0.475062061
 [120]  0.533069643  0.526332280  0.533404590  0.521997460  0.517430527 13.736864230
4.920556417
 [127]  2.663581800  7.773084727  7.559891169  2.439047007  4.583594653  3.947113811
0.888752395
 [134]  3.245971867  5.817571492  5.676287149  5.571049307  4.964521895  6.135095123
5.402724137
 [141]  5.478680865  5.336387541  5.184313119  5.190372700  4.798801671  4.486088994
4.156214067
 [148]  4.178365313  4.084843316  4.034657299  4.088872854  4.015673751  3.704675089
3.346682262
 [155]  3.543126080  3.785745711  4.063831116  4.097838870  4.105467393  4.078266752
4.091713768
 [162]  0.620546584  0.600503433  0.583741361  0.523016010  0.521671269  0.494724777
0.487098617
 [169]  0.519978664  0.510847964  0.499488832  0.513972995  0.527195841  0.503400709
0.485471053
```

```
 [176]  0.473107147  0.468799888  0.502829357  0.441804200  0.431340880  0.404473802
0.374669357
 [183]  0.379550900  0.383571245  0.408284037  0.417358254  0.415569994  0.378633236
5.070369070
 [190]  5.341982538  5.637375396  5.522880783  5.480153472  5.714705370  5.270391987
5.279382718
 [197]  5.091243575  5.136023161  4.958480952  4.951486993  4.889660067  5.255777302
5.449098215
 [204]  5.397440483  5.368050932  5.005305785  5.269790030  5.421221351  6.471094010
6.628306040
 [211]  6.936548182  7.074476432  7.077000957  7.061629510  7.077428347  2.338816320
1.886415645
 [218]  1.860565881  2.362608184  2.222376976  1.778622434  1.617024538  1.414468260
1.329264592
 [225]  1.405007574  1.387649645  1.355902715  1.301618494  1.251013872  1.177557860
1.140432395
 [232]  1.170444672  1.092868921  1.038564005  1.027963258  1.080659088  1.173898493
1.182358375
 [239]  1.170208698  1.198974907  1.207275443  1.194949059  0.656899919  0.689807659
0.729180945
 [246]  0.759185432  0.795083788  0.865546680  0.897275327  0.913690923  0.888569432
0.884821338
 [253]  0.874894955  0.904363156  0.952379784  1.024681315  1.058980068  1.008566463
1.032520789
 [260]  1.062511808  1.097755765  1.131800847  1.164494571  1.156273463  1.164692287
1.161697842
 [267]  1.164255131  1.166773044  1.191488780  0.666600418  0.662496452  0.676056046
0.648934241
 [274]  0.693573444  0.540711169  0.562563515  0.390671912  0.390774068  0.406701778
0.374265228
 [281]  0.348373681  0.344449835  0.373847499  0.375936895  0.387750192  0.414305262
0.387094780
 [288]  0.411695955  0.401160456  0.394883592  0.400568561  0.407545998  0.407639989
0.388258967
 [295]  0.389126533  0.386311507  0.922566562  0.972953309  0.958001782  0.754932800
0.743976852
 [302]  0.672408068  0.783880028  0.773568364  0.763660387  0.696092849  0.630273166
0.622726293
 [309]  0.559466523  0.553002805  0.492030290  0.594604456  0.587995894  0.687254670
0.522877060
 [316]  0.465470275  0.562747837  0.506083352  0.450618833  0.396337258  0.441271352
0.436800437
 [323]  0.434652611  0.884257776  0.924094229  1.002906978  1.070269957  1.251366951
1.287768632
 [330]  1.503520748  1.648866186  1.756217649  1.908321673  2.031900081  2.190493351
2.118428759
 [337]  2.262052214  2.334802795  2.412858360  2.538053993  2.517240952  2.571766259
2.649248936
 [344]  2.614854425  2.680799011  2.801115637  2.830702128  2.955825099  3.063364339
3.075676965
 [351]  0.302433473  0.296973903  0.316064270  0.310670837  0.329042428  0.277536569
```

```
0.296024806
 [358]  0.269153893  0.265198165  0.304920607  0.279099883  0.254014133  0.292274534
0.308958678
 [365]  0.345611154  0.240899321  0.237973708  0.215582230  0.232515741  0.210810116
0.208591694
 [372]  0.206480149  0.185880045  0.184133586  0.200710365  0.217041052  0.222138987
1.128795145
 [379]  1.149994518  1.171217919  1.249493046  1.268528809  1.245368589  1.277438797
1.281502270
 [386]  1.062091211  0.991587022  1.011494980  0.980415523  0.864803454  0.693515292
1.033330789
 [393]  1.222954681  1.462786126  1.671391302  1.586227370  1.405151254  1.344351189
1.356044590
 [400]  1.325816871  1.316370109  1.279012739  1.262748897  1.280147651  1.003778380
0.969538992
 [407]  0.922476507  0.893757762  0.852639520  0.870070723  0.831601347  0.807903244
0.849208100
 [414]  0.987623874  1.032688598  1.204719505  1.263568513  1.430609350  1.577412443
1.461730502
 [421]  1.485543732  1.427106178  1.351455118  1.374534514  1.377350308  1.343251918
1.301529569
 [428]  1.226862918  1.224608077  0.643499788  0.625665335  0.608527941  0.592435190
0.577531518
 [435]  0.615054507  0.701251894  0.636651657  0.717998872  0.981673828  0.957820647
0.978361830
 [442]  1.083455632  0.928960883  0.946258236  1.042279275  1.210935350  1.370484065
1.521518009
 [449]  1.519878706  1.553242273  1.515782617  1.513505167  1.511203363  1.541049599
1.506775137
 [456]  1.525164495  0.485960303  0.476514728  0.467420027  1.146593250  1.125347035
0.441902945
 [463]  0.542409655  0.639342833  0.628242349  0.720535255  0.708651997  0.697247457
0.686231697
 [470]  0.675499332  0.664981902  0.654658918  0.736635918  0.725389562  0.625193470
0.528020065
 [477]  0.520459069  0.513177883  0.506171360  0.582669554  0.575105942  0.567837709
0.574802612
 [484]  0.605145309  0.600930193  0.593979962  0.584423445  0.598954091  0.595778313
0.592653615
 [491]  0.587877105  0.579877828  0.571362901  0.571402881  0.579569039  0.545449831
0.543740674
 [498]  0.539077014  0.580059192  0.590801466  0.568254102  0.518208890  0.543304973
0.531197338
 [505]  0.517446471  0.500008257  0.487797519  0.475221958  0.468917138  0.464825918
0.448068231
 [512]  0.468678184  0.516363608  0.453748793  0.424456299  0.440161517  0.468514647
0.350710941
 [519]  0.328981297  0.328636942  0.385733997  0.514148539  0.575195223  0.488738654
0.458951792
 [526]  0.541210181  0.458379049  0.426239258  0.476044363  0.418851409  0.409790111
0.393810864
```

```
 [533]  0.399254534  0.401171477  0.406578843  0.405248117  0.403752259  6.282637702
6.221905691
 [540]  5.850520626  5.664415007  5.587660065  5.885170984  5.878311456  6.072679435
5.420161296
 [547]  5.826570242  5.847986477  5.849622168  5.355354851  5.420505338  5.521795405
4.901078858
 [554]  4.245347867  3.676236914  3.010454542  2.808761407  3.019778592  3.235422413
3.125241000
 [561]  3.175690306  3.248165480  3.254244742  3.199108247  1.137538780  1.098402739
1.235546017
 [568]  1.363310676  1.320424186  1.283215037  0.937821377  1.221331668  1.194990545
1.316990676
 [575]  1.292010741  1.128165353  1.247094836  1.088090509  1.199331282  1.302769570
1.271536652
 [582]  1.488621770  1.453952658  1.660294568  1.861613866  1.829876106  1.801013582
1.884254384
 [589]  1.855149528  1.826069337  1.819063548 12.477043676 14.200794223 13.098933541 1
0.609126888
 [596]  9.245657502  8.307442623  6.872139616  5.784864399  5.589442105  5.712453966
5.979646817
 [603]  5.580171368  5.868787869  6.093808145  6.485903095  6.925949716  7.243039538
7.379633702
 [610]  7.715218673  7.973754267  8.108689733  8.079744789  8.105173841  8.038476661
7.984148195
 [617]  7.890672859  0.376272793  0.458221165  0.446683275  0.391962977  0.551980495
0.620858591
 [624]  0.524575009  0.551029983  0.538024629  0.526029269  0.625392961  0.540939711
0.460002686
 [631]  0.521079440  0.545818518  0.603128730  0.658421188  0.647120205  0.636274440
0.751080211
 [638]  0.615966769  0.667081525  0.806370326  0.971172497  1.044475604  1.144655888
1.146792857
 [645]  0.962109644  0.929316340  0.952904328  0.973763707  0.995406300  1.072465782
1.095247409
 [652]  1.219584025  1.188247824  1.204519796  1.217516054  1.137026119  1.059788745
1.071959679
 [659]  1.041337304  0.890612047  0.905202409  0.956735133  0.744254965  0.723640768
0.738700048
 [666]  0.786540825  0.764720684  0.776006126  0.786336053  0.765181195  0.777123503
1.525200886
 [673]  1.522181682  1.544232498  1.714181590  1.801801469  1.884052169  2.084666776
2.183796793
 [680]  2.291318019  2.413475881  2.555887421  2.532838856  2.502310277  2.497035285
2.483466596
 [687]  2.492917673  2.481779107  2.463278463  2.455041695  2.528089430  1.192887591
1.778407564
 [694]  2.403473927  2.384998669  2.352115202  2.296623369  2.306752901  4.128191593
4.306429649
 [701]  4.491351446  4.688331269  4.902456979  3.038402070  2.964165895  2.966359244
2.962455903
 [708]  2.953479586  2.983534784  2.994907036  2.994060571  3.004232638  3.024067209
```

```
2.209498736
 [715]  2.208707181  2.212180753  2.198401667  2.183906370  2.163648052  2.152968915
2.175805696
 [722]  2.149539443  2.119360993  2.085605454  2.082779310  0.610630522  0.595568429
0.581254541
 [729]  0.567593019  0.554522827  0.542017661  0.530083227  0.518734367  0.507991241
0.497859581
 [736]  0.488289378  0.479249756  0.470760260  0.617138930  0.607368833  0.448765894
0.442495676
 [743]  0.436570737  0.430819945  0.566838651  0.559284116  0.551796788  0.544413804
0.537197902
 [750]  0.530190748  0.523386609  0.519878837  1.184752770  1.122396651  1.024834996
0.968696866
 [757]  0.951796182  0.972133801  0.992451488  1.012708861  1.273800628  1.290378346
1.273353703
 [764]  1.257073860  1.241381051  1.290644769  1.274881467  1.290878413  1.275237455
1.228962312
 [771]  1.213727813  1.198353354  1.212370073  1.254693079  1.180417373  1.221703314
1.234037901
 [778]  1.246293379  1.254116957  0.346402469  0.365095292  0.369664969  0.399764772
0.390631097
 [785]  0.369554427  0.373480681  0.365408218  0.380676574  0.383923215  0.386912606
0.368045122
 [792]  0.349998728  0.353679980  0.367767543  0.462137913  0.494573684  0.516517027
0.528142810
 [799]  0.539129332  0.530786742  0.504075723  0.505064584  0.505957866  0.498116175
0.490561723
 [806]  0.491407198  2.066867912  2.204858233  1.881499101  1.729601948  1.418295496
1.292271876
 [813]  1.142059080  0.875703300  0.854100272  0.818318815  0.824290882  0.758816406
0.798800066
 [820]  0.949463944  1.036540276  1.098663154  1.166805516  1.135655034  1.185623344
1.201979683
 [827]  1.244779029  1.282947680  1.092435341  1.158755185  1.170677646  1.171724469
5.678911771
 [834]  5.931840417  7.621492094  6.398629052  7.135870362  6.809464371  6.571200437
6.111655935
 [841]  6.211478366  6.376135559  5.637965207  5.528678784  5.312123655  5.267802842
5.582280827
 [848]  5.088908375  5.330568880  4.573934556  4.267402608  4.855708216  5.288233749
5.542091761
 [855]  5.464209183  5.584789498  5.444575036  5.286155463  5.278516523  3.945626131
4.075888321
 [862]  3.664858300  4.252622755  3.313796066  3.439925212  3.362416653  3.270922859
3.720237927
 [869]  4.057299330  3.359430055  3.290095007  3.166192086  3.193813769  3.184914261
3.356462383
 [876]  2.886489563  3.443179690  3.347974183  3.136672078  2.914394263  3.047848611
3.031266866
 [883]  3.081766309  3.100729101  3.025160011  2.987115063  0.004856906  0.005544624
0.006145457
```

```
 [890]  0.006910559  0.007696202  0.007407262  0.007729358  0.007757893  0.007395308
0.007333347
 [897]  0.007463738  0.007118944  0.007209771  0.007135395  0.008206554  0.010786647
0.010646953
 [904]  0.010966434  0.010284843  0.010222787  0.010036200  0.009775524  0.009525923
0.009285730
 [911]  0.009072806  0.008811150  0.008699658  1.070417995  1.067495176  1.063905443
1.060960810
 [918]  1.088841945  1.059415096  1.060450095  1.034527030  1.035754036  1.035550264
1.033838734
 [925]  1.006261804  1.026413604  0.974289666  0.946100917  0.961728329  0.932517701
0.903728206
 [932]  0.875580111  0.867497814  1.045773826  1.085398897  1.104086434  1.086784111
1.069750340
 [939]  1.053023390  1.066299643  0.015377634  0.030470086  0.015110495  0.014985669
0.014848288
 [946]  0.014696314  0.014534658  0.014367119  0.014199512  0.014036583  0.013876806
0.013720700
 [953]  0.013576633  0.026909455  0.040082613  0.026596352  0.039786857  0.039727171
0.039670494
 [960]  0.039584946  0.039460137  0.039306695  0.039139280  0.038980610  0.038846746
0.051653340
 [967]  0.051774964  0.600225123  0.515453954  0.741194157  0.675435387  0.540309646
0.404020241
 [974]  0.372166610  0.339858881  0.306895276  0.273230717  0.312378303  0.314988682
0.317439368
 [981]  0.300884854  0.283854364  0.285410342  0.325016164  0.345619782  0.404841448
0.387002946
 [988]  0.388349943  0.370196441  0.371534915  0.373016454  0.374683161  0.382876477 1
4.048307019
 [995] 14.292853573 12.006908934 13.546631199 11.837322417 10.424861716 10.370710599
 [ reached getOption("max.print") -- omitted 72 entries ]
```

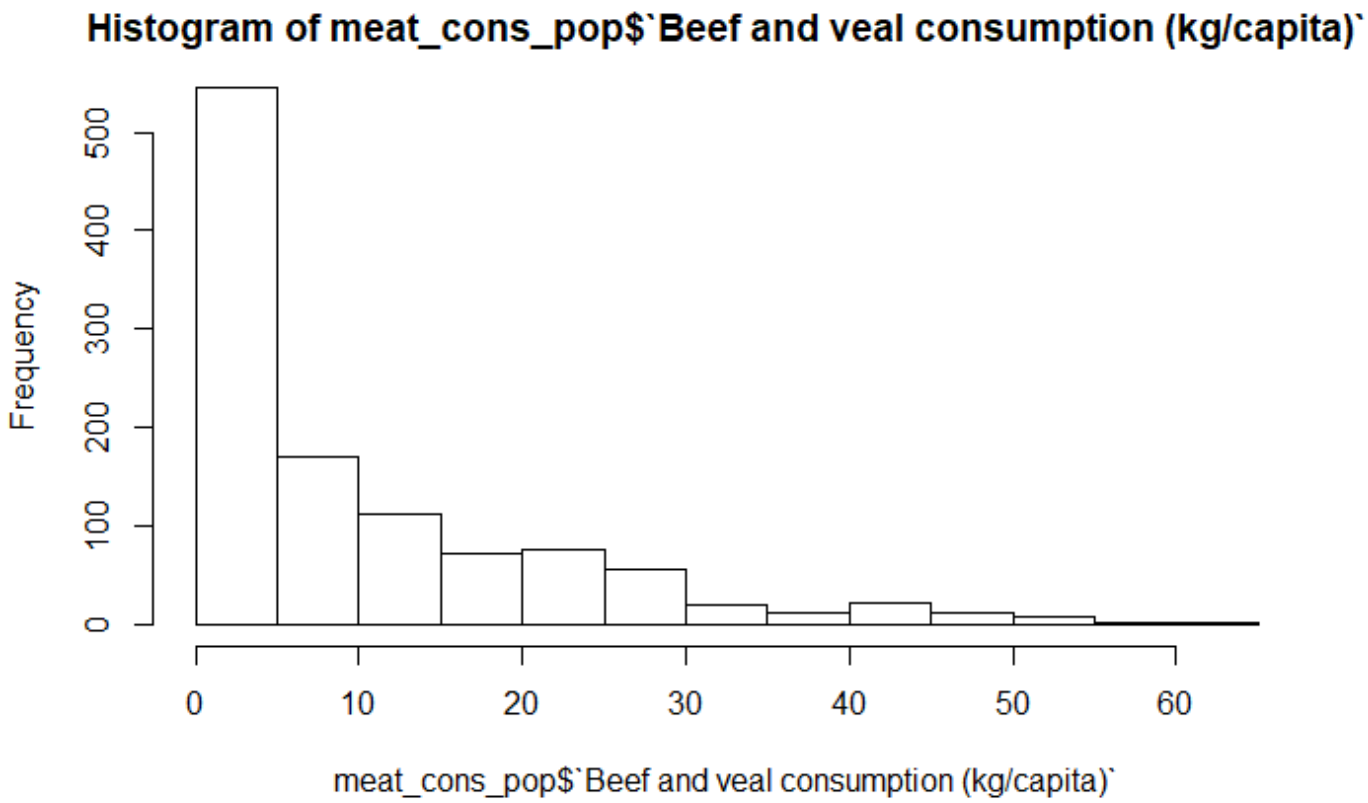
Hide

```
#
```

Transform

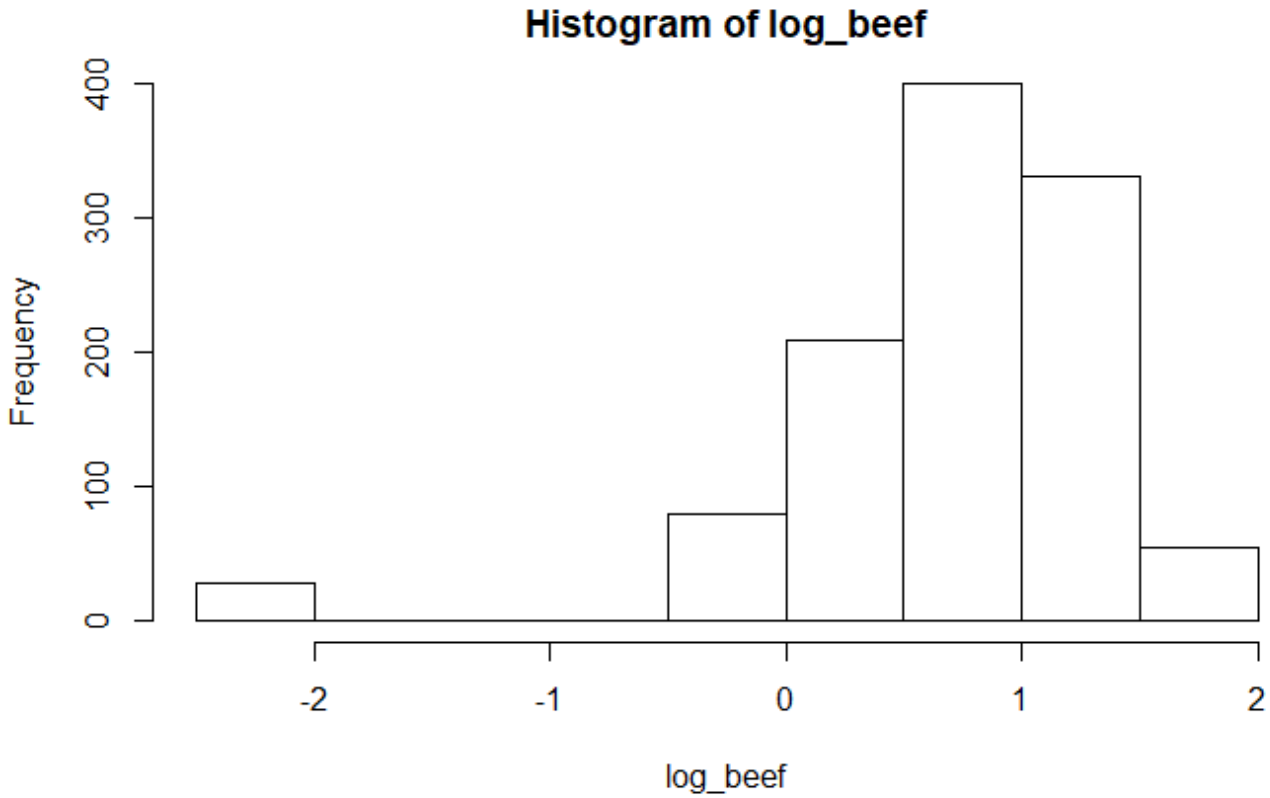
Hide

```
hist(meat_cons_pop$`Beef and veal consumption (kg/capita)`)
```



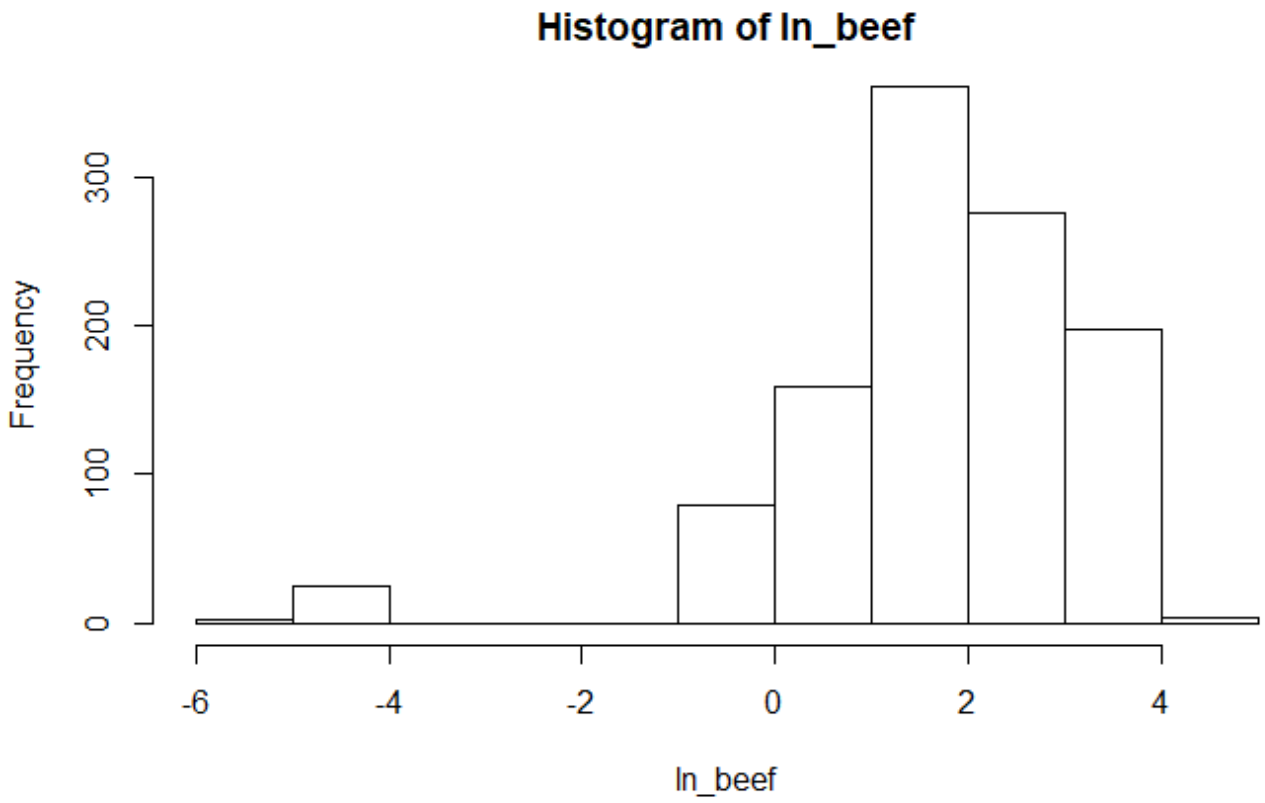
Hide

```
log_beef <- log10(meat_cons_pop$`Beef and veal consumption (kg/capita)`)
hist(log_beef)
```



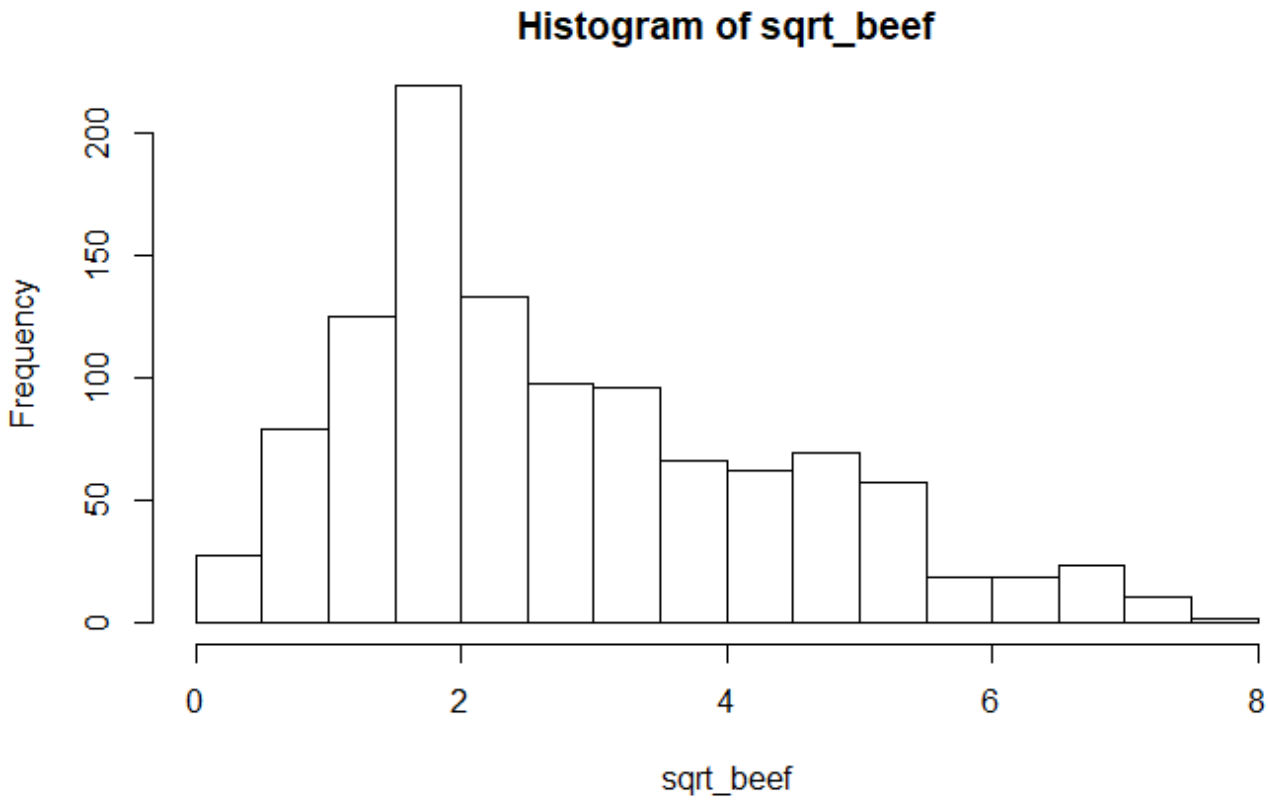
Hide

```
ln_beef<-log(meat_cons_pop$`Beef and veal consumption (kg/capita)`)
hist(ln_beef)
```



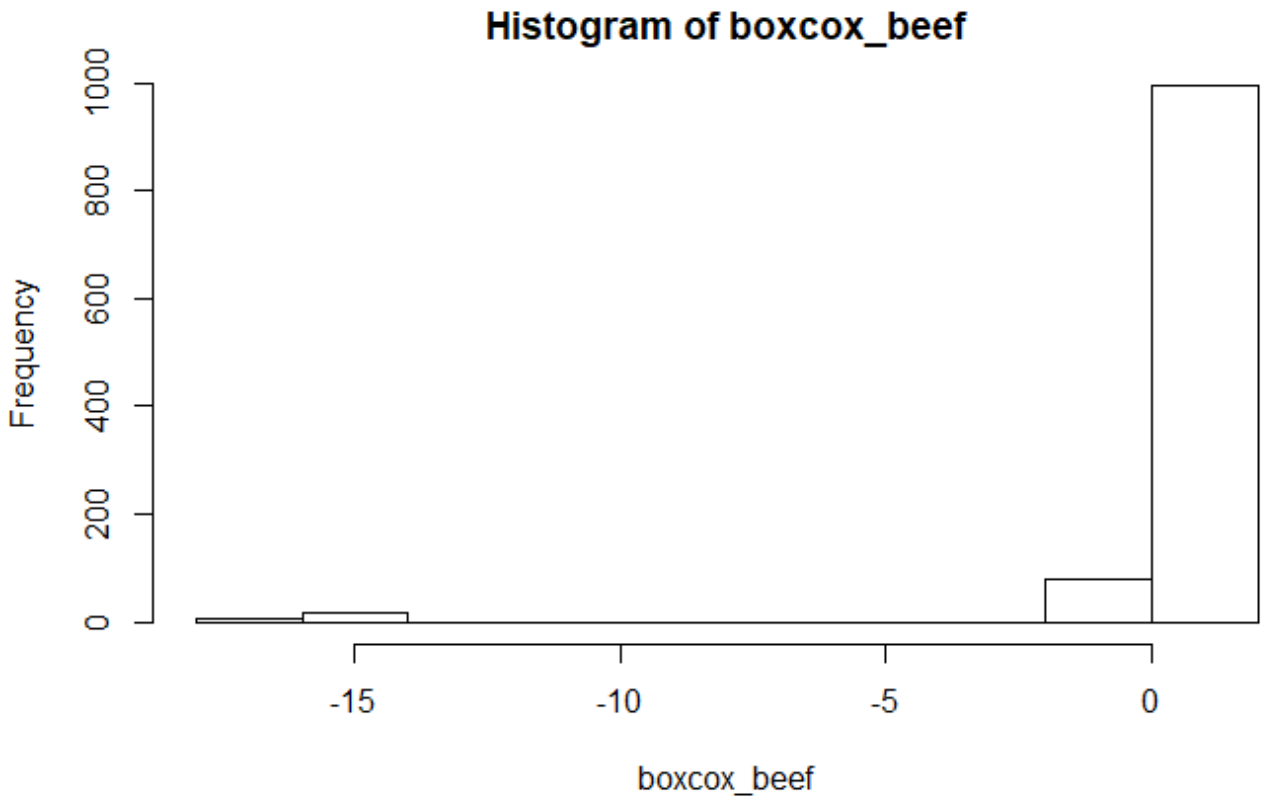
Hide

```
sqrt_beef<-sqrt(meat_cons_pop$`Beef and veal consumption (kg/capita)`)
hist(sqrt_beef)
```



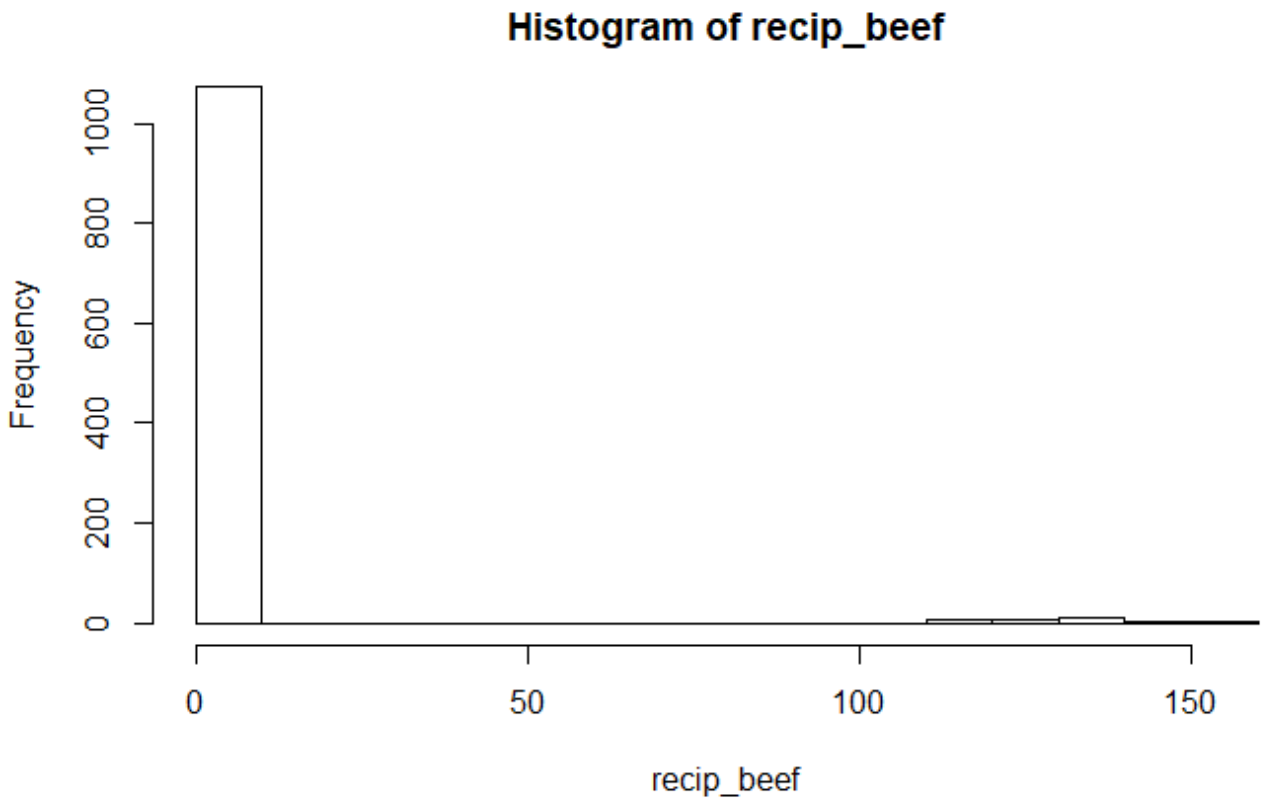
Hide

```
boxcox_beef<- BoxCox(meat_cons_pop$`Beef and veal consumption (kg/capita)`,lambda = "a
uto")
hist(boxcox_beef)
```



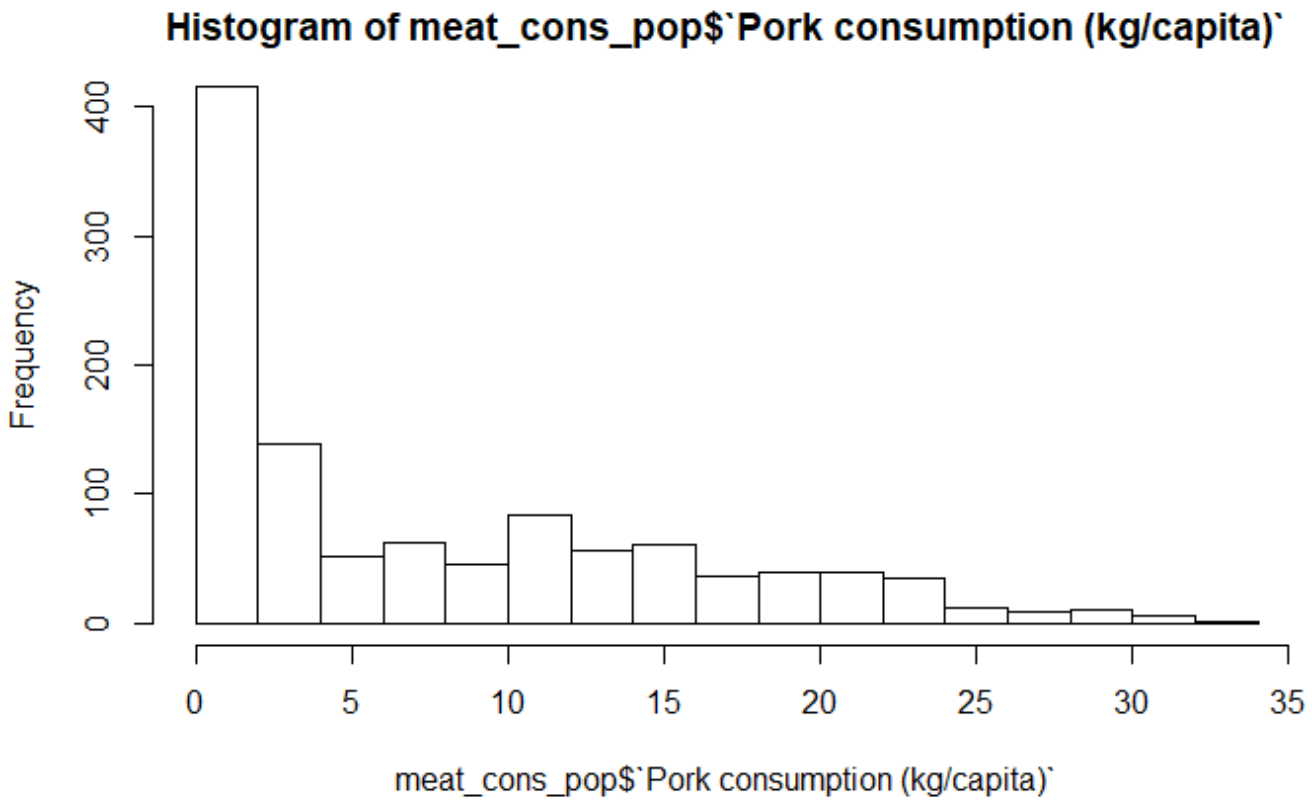
Hide

```
recip_beef <- 1/meat_cons_pop$`Beef and veal consumption (kg/capita)`
hist(recip_beef)
```



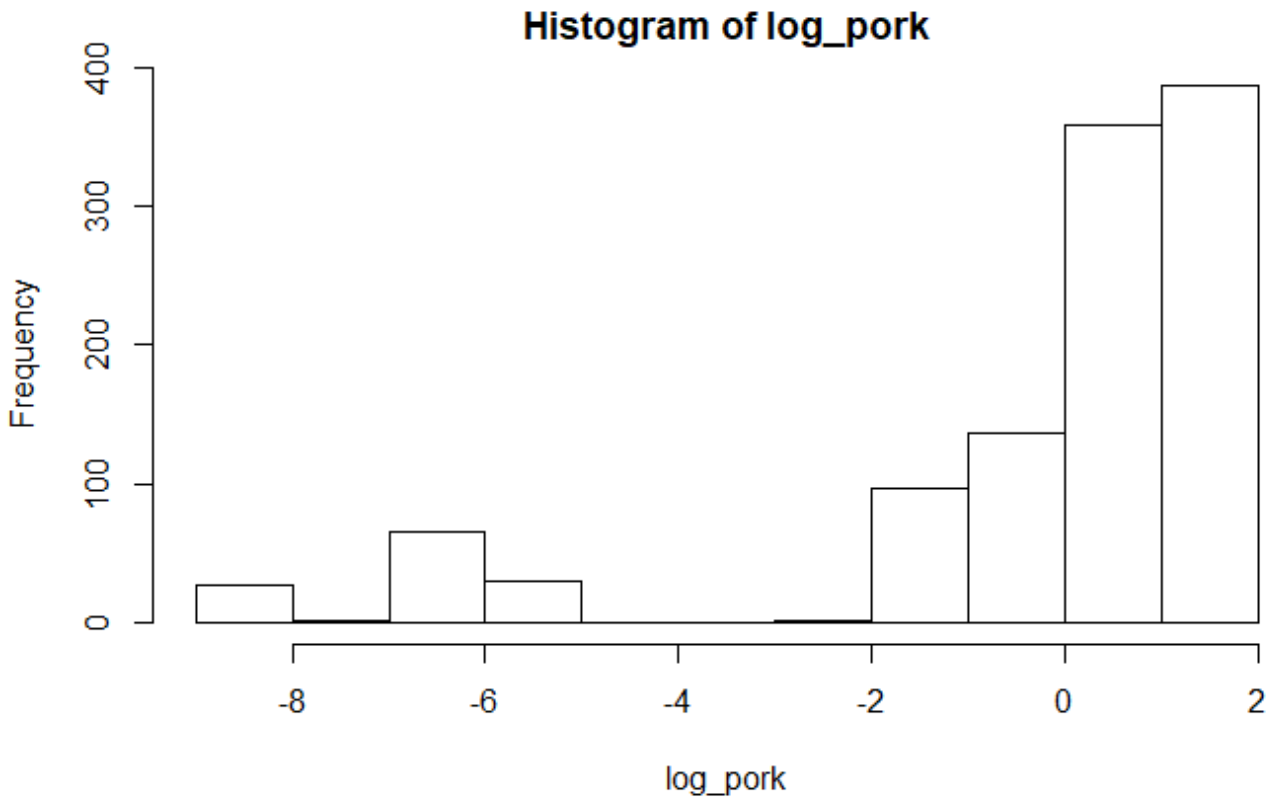
Hide

```
#square root transformation works best for this variable
hist(meat_cons_pop$`Pork consumption (kg/capita)`)
```



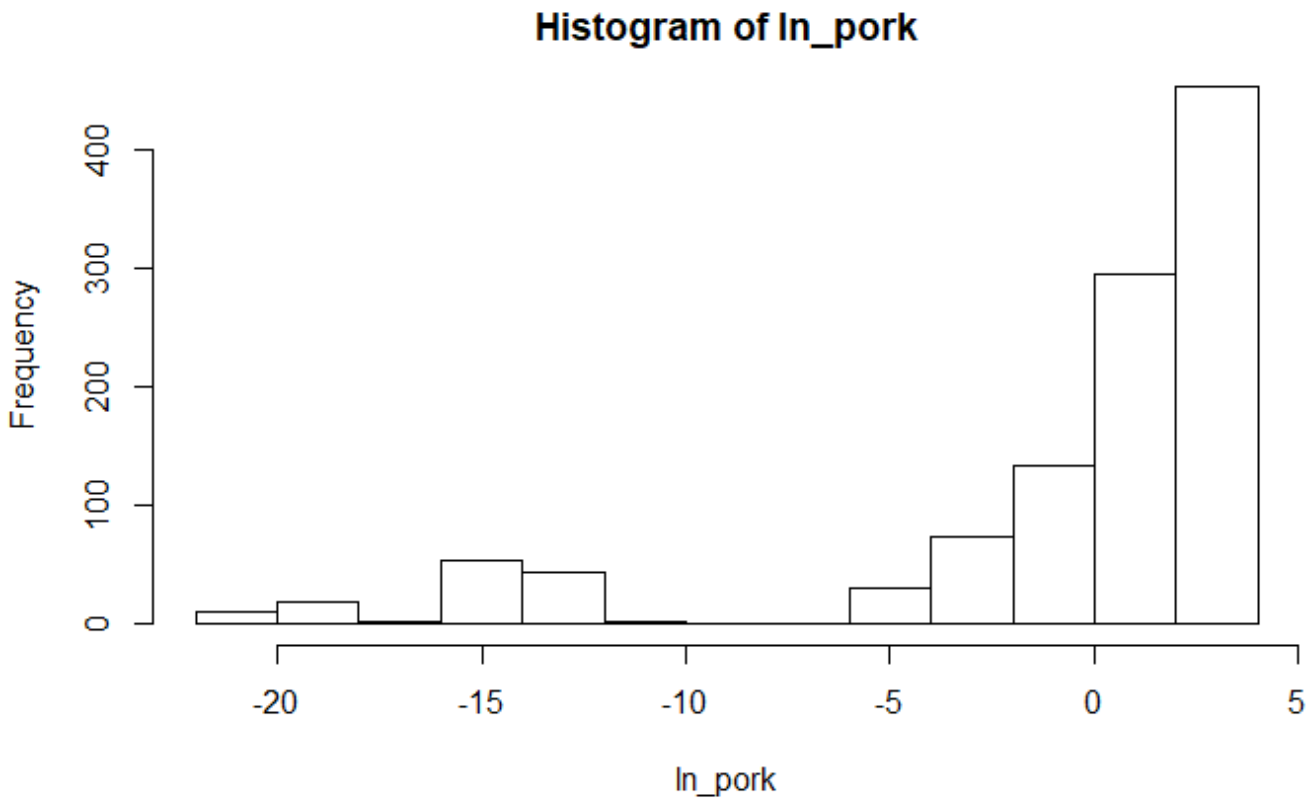
Hide

```
log_pork <- log10(meat_cons_pop$`Pork consumption (kg/capita)`)
hist(log_pork)
```



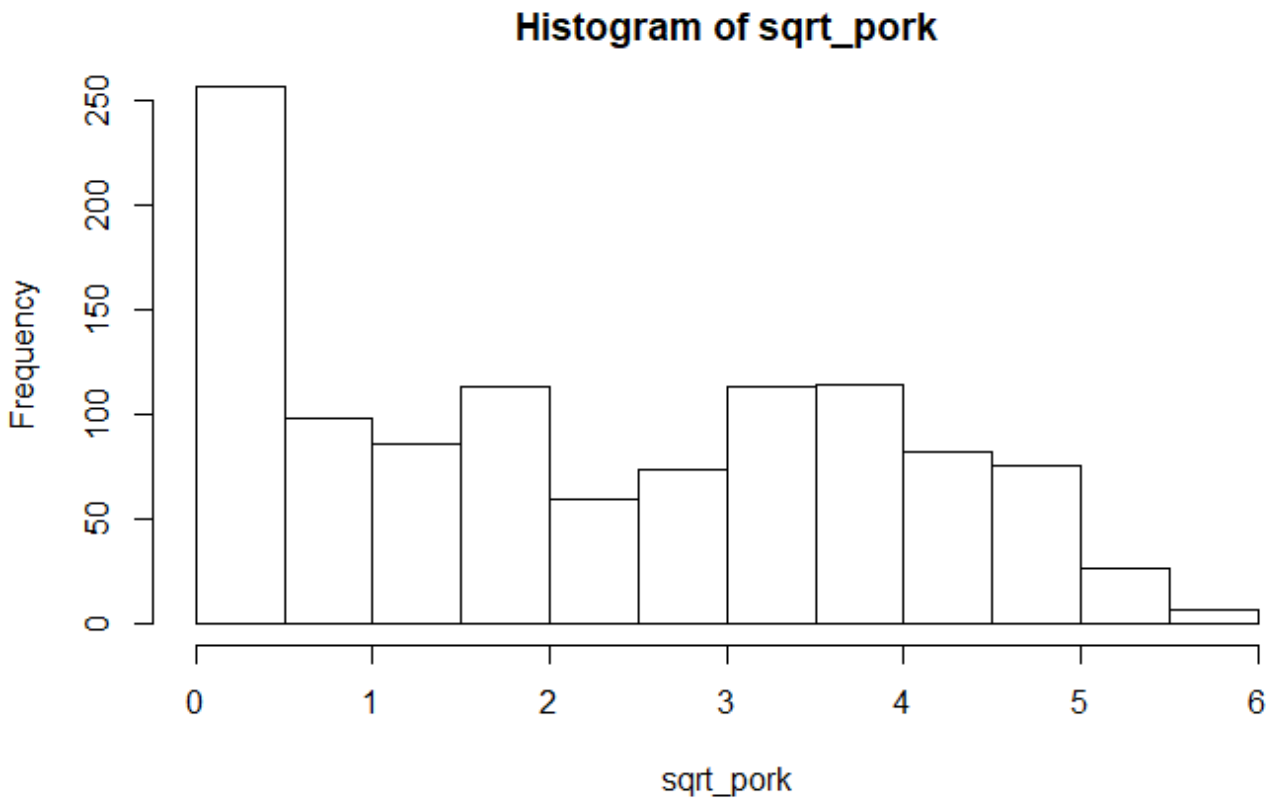
Hide

```
ln_pork<-log(meat_cons_pop$`Pork consumption (kg/capita)`)
hist(ln_pork)
```



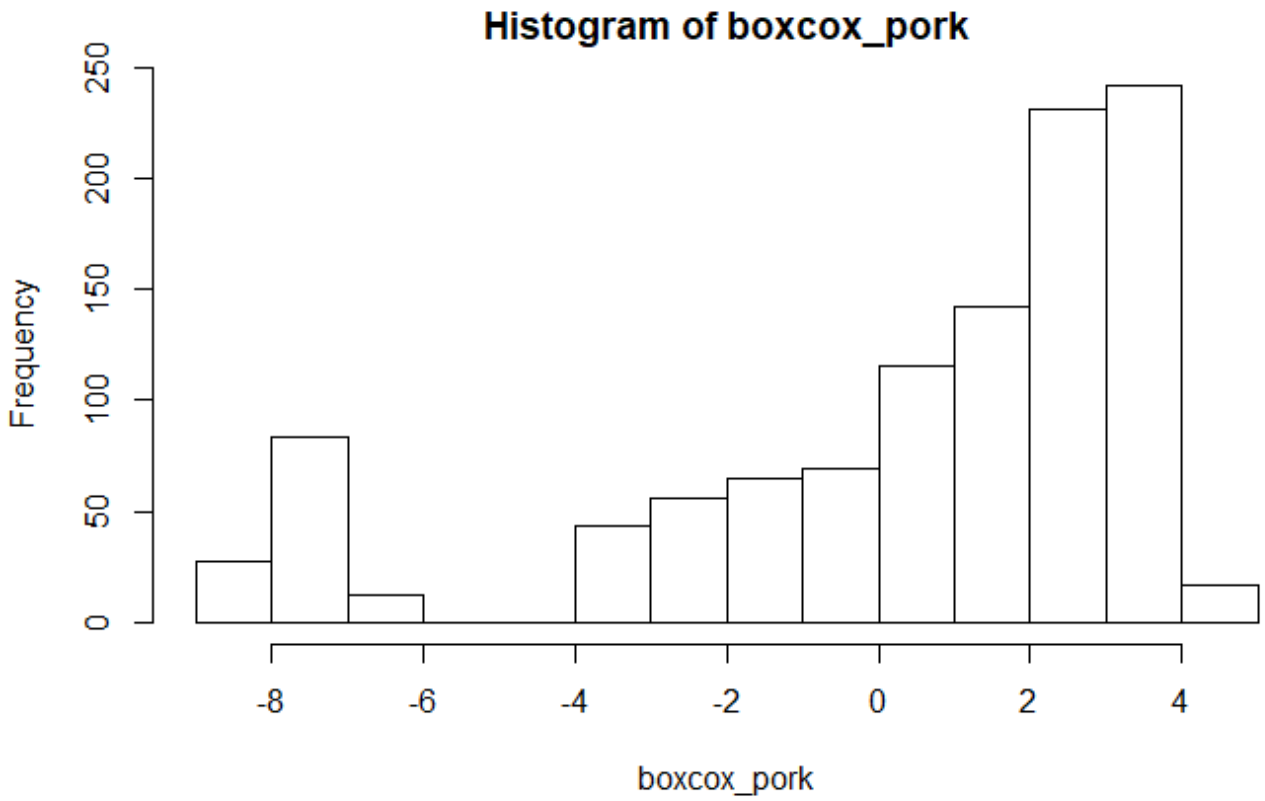
Hide

```
sqrt_pork<-sqrt(meat_cons_pop$`Pork consumption (kg/capita)`)
hist(sqrt_pork)
```



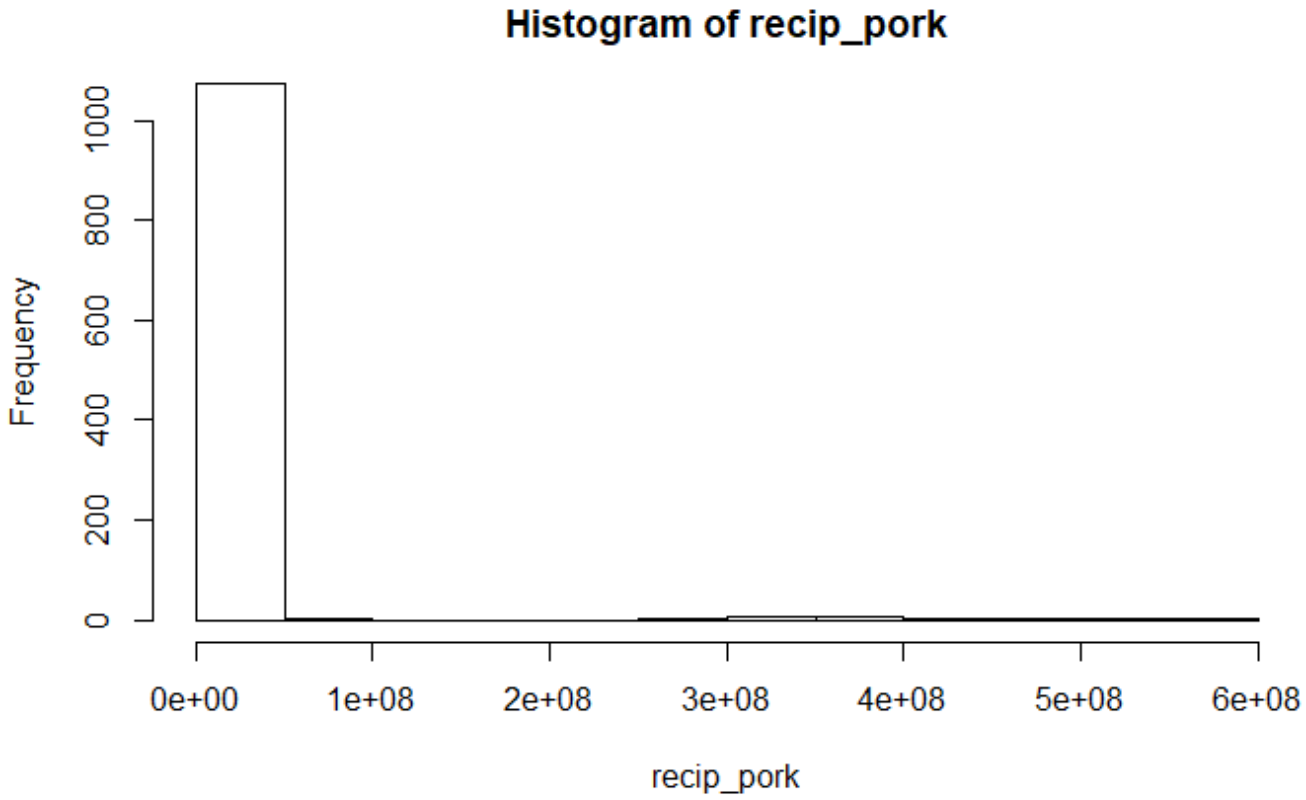
Hide

```
boxcox_pork<- BoxCox(meat_cons_pop$`Pork consumption (kg/capita)`,lambda = "auto")
hist(boxcox_pork)
```



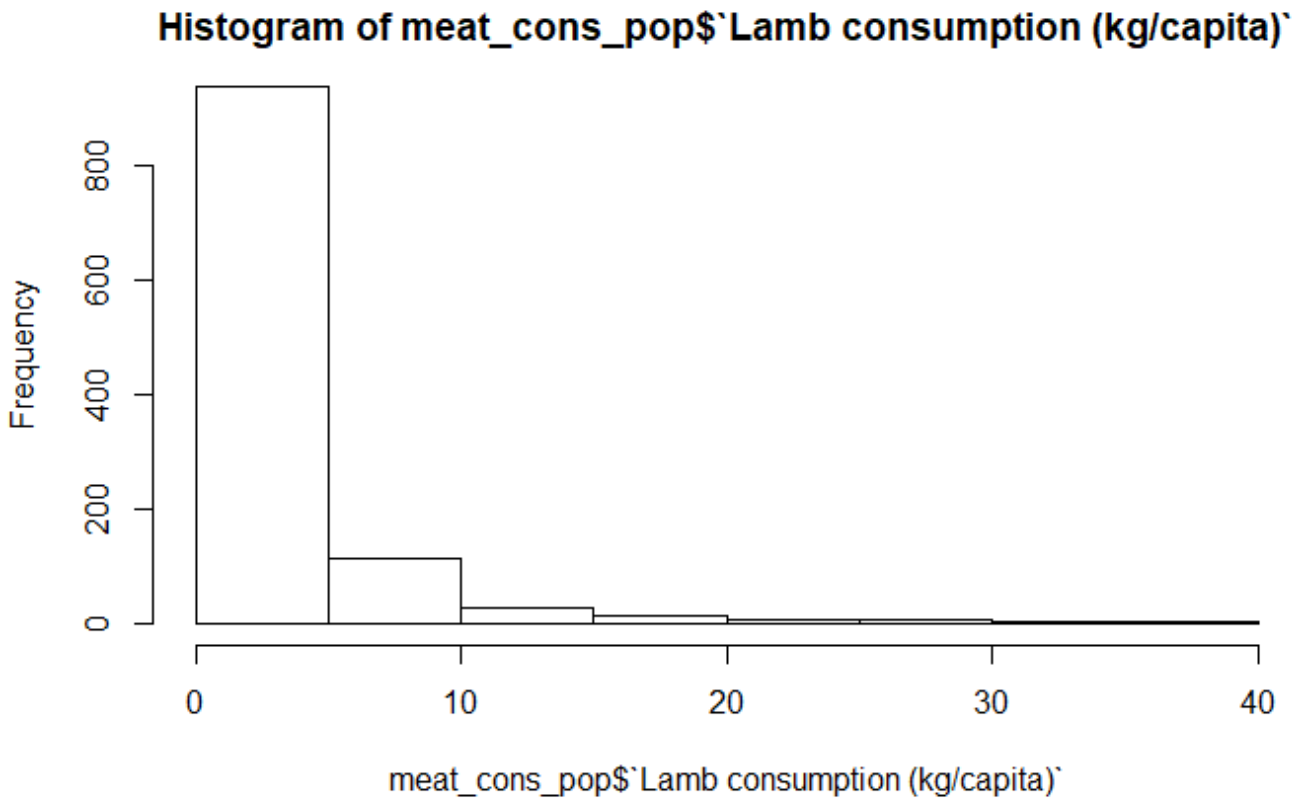
Hide

```
recip_pork <- 1/meat_cons_pop$`Pork consumption (kg/capita)`
hist(recip_pork)
```



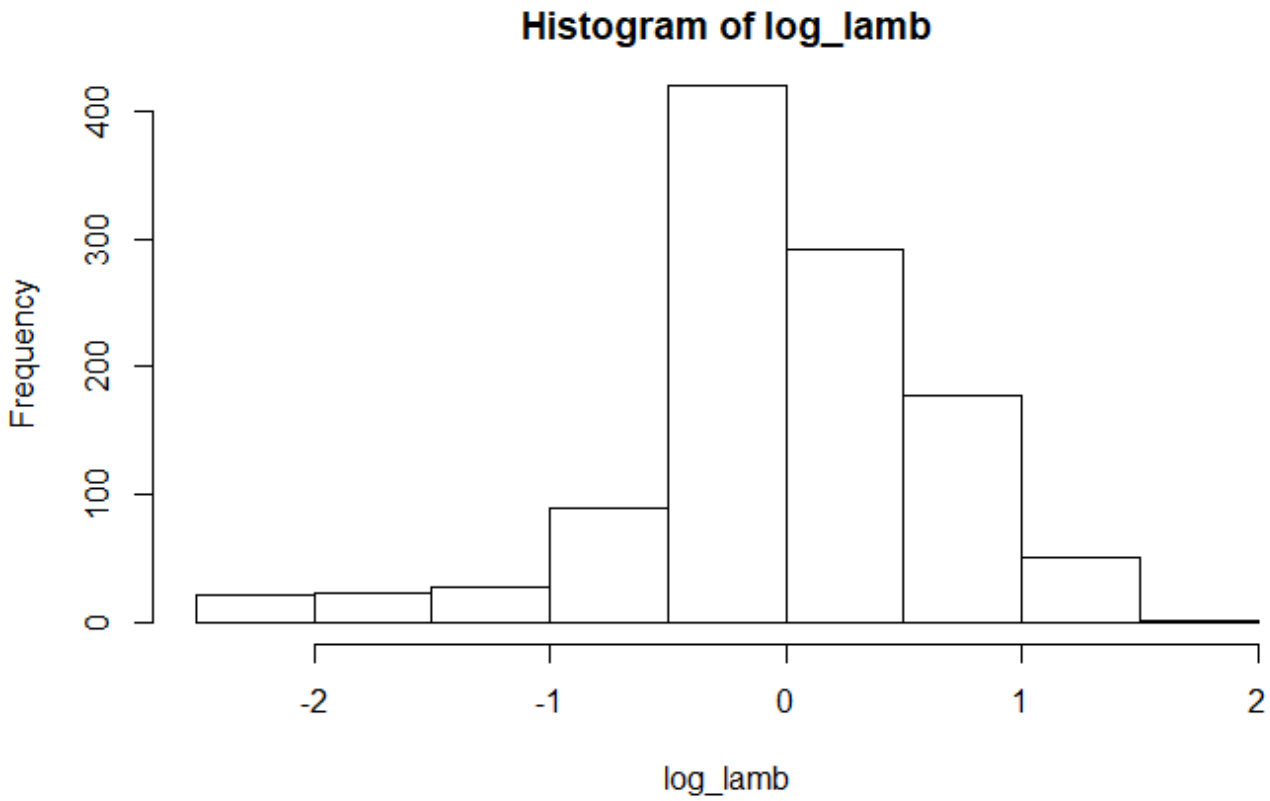
Hide

```
#square root seem to perform best
hist(meat_cons_pop$`Lamb consumption (kg/capita)`)
```



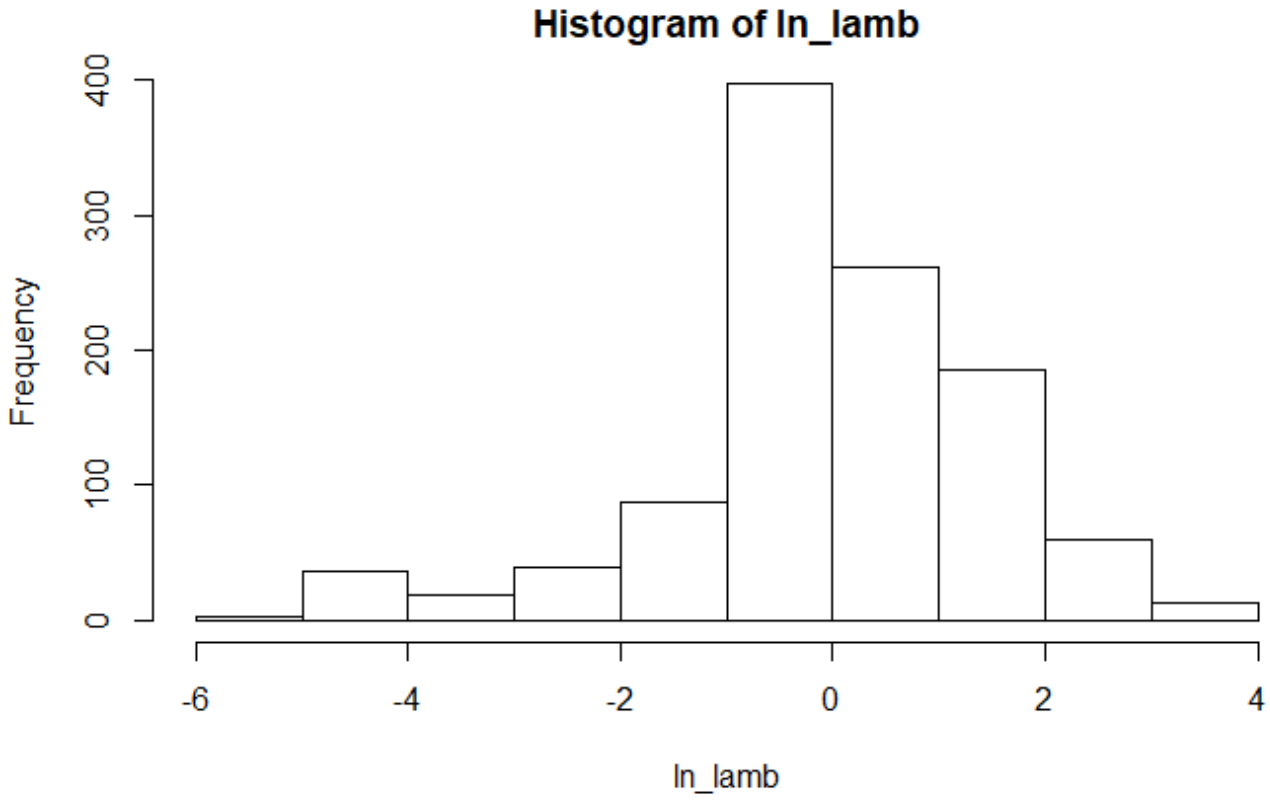
Hide

```
log_lamb <- log10(meat_cons_pop$`Lamb consumption (kg/capita)`)
hist(log_lamb)
```



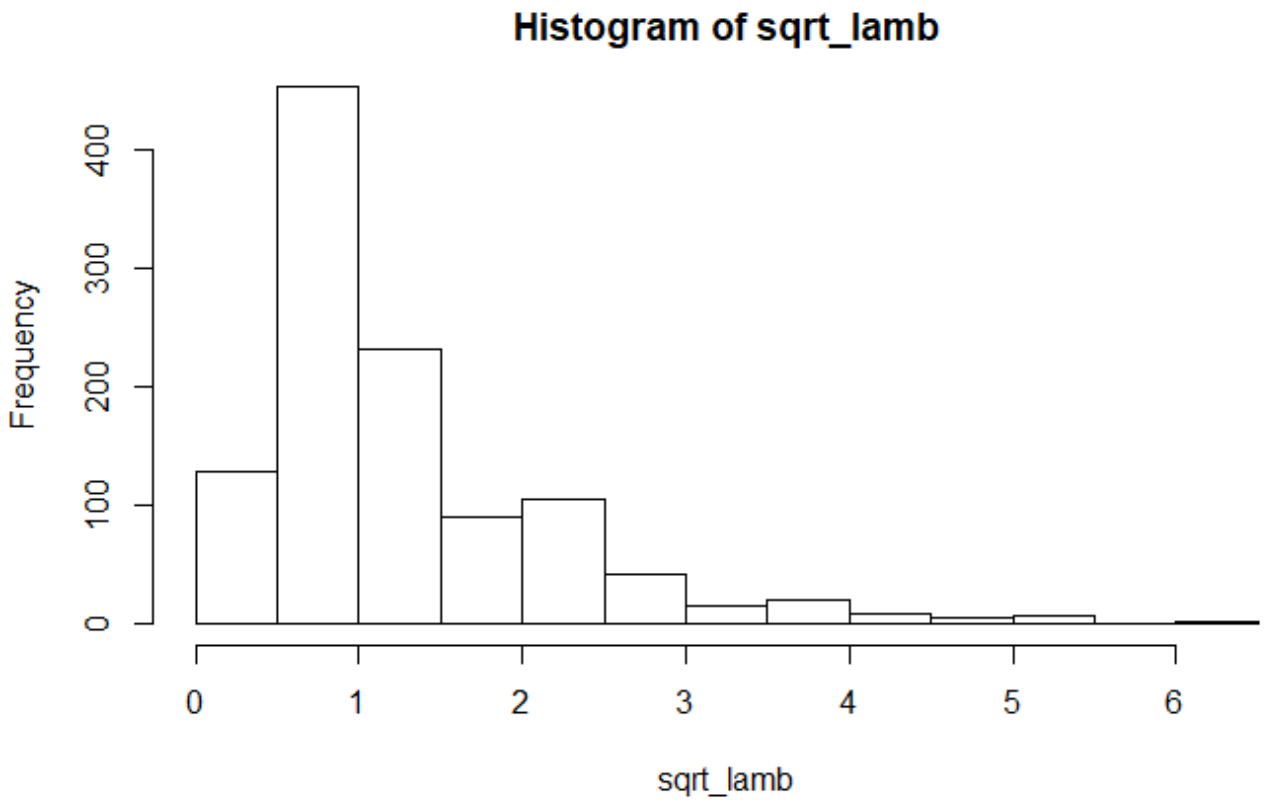
Hide

```
ln_lamb<-log(meat_cons_pop$`Lamb consumption (kg/capita)`)
hist(ln_lamb)
```



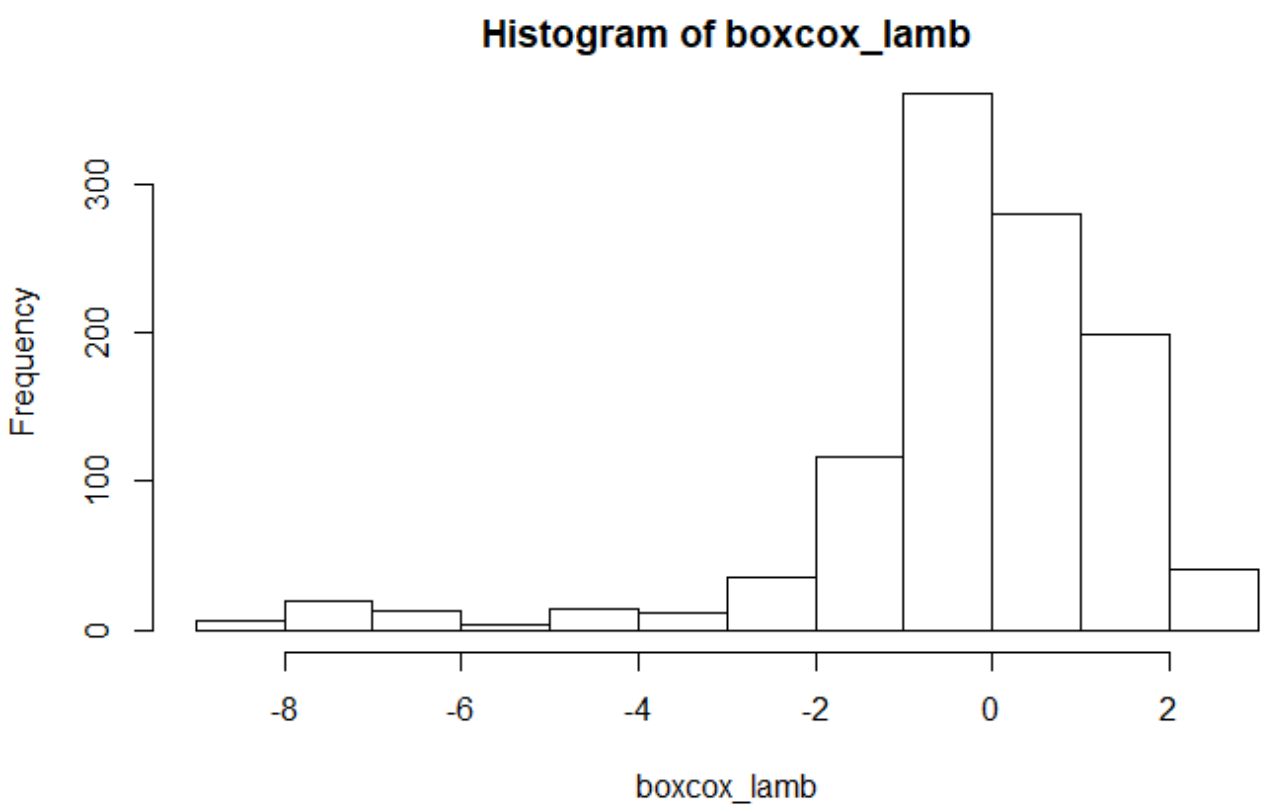
Hide

```
sqrt_lamb<-sqrt(meat_cons_pop$`Lamb consumption (kg/capita)`)
hist(sqrt_lamb)
```



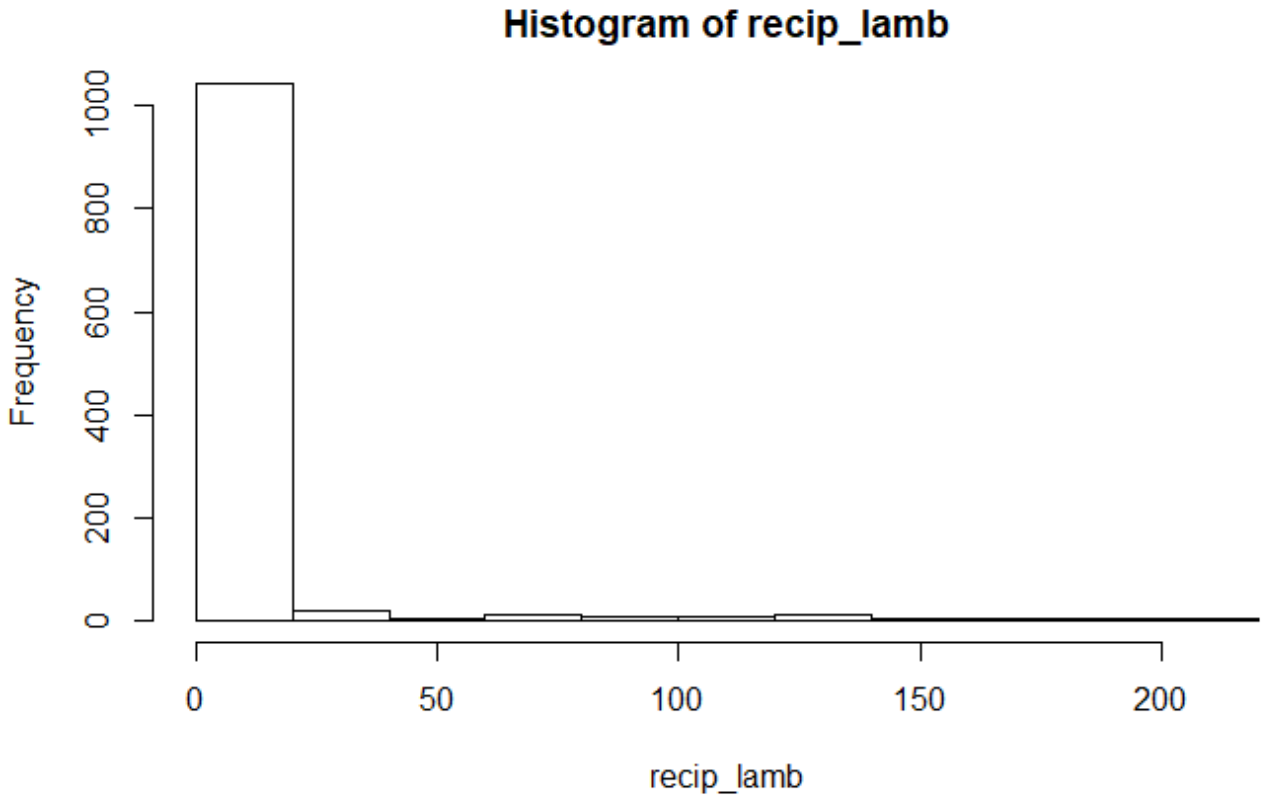
Hide

```
boxcox_lamb<- BoxCox(meat_cons_pop$`Lamb consumption (kg/capita)`,lambda = "auto")
hist(boxcox_lamb)
```



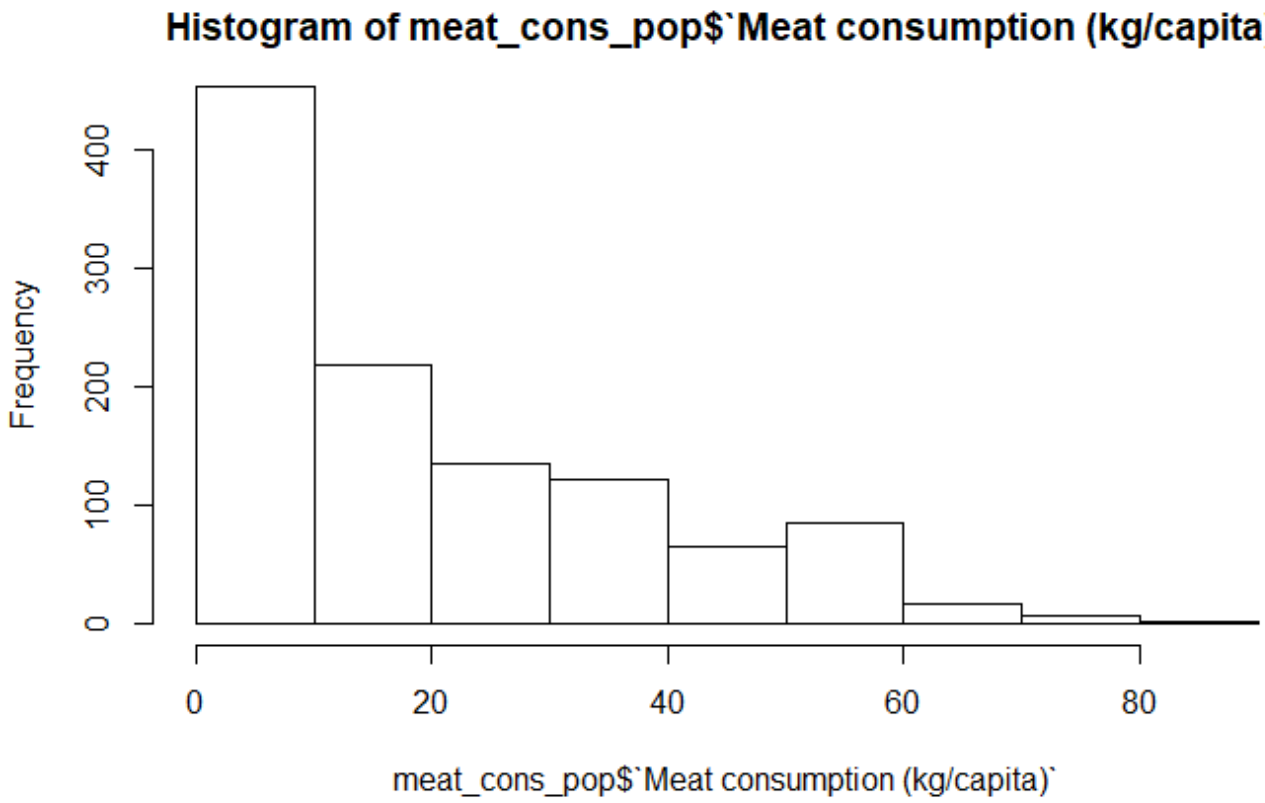
Hide

```
recip_lamb <- 1/meat_cons_pop$`Lamb consumption (kg/capita)`
hist(recip_lamb)
```



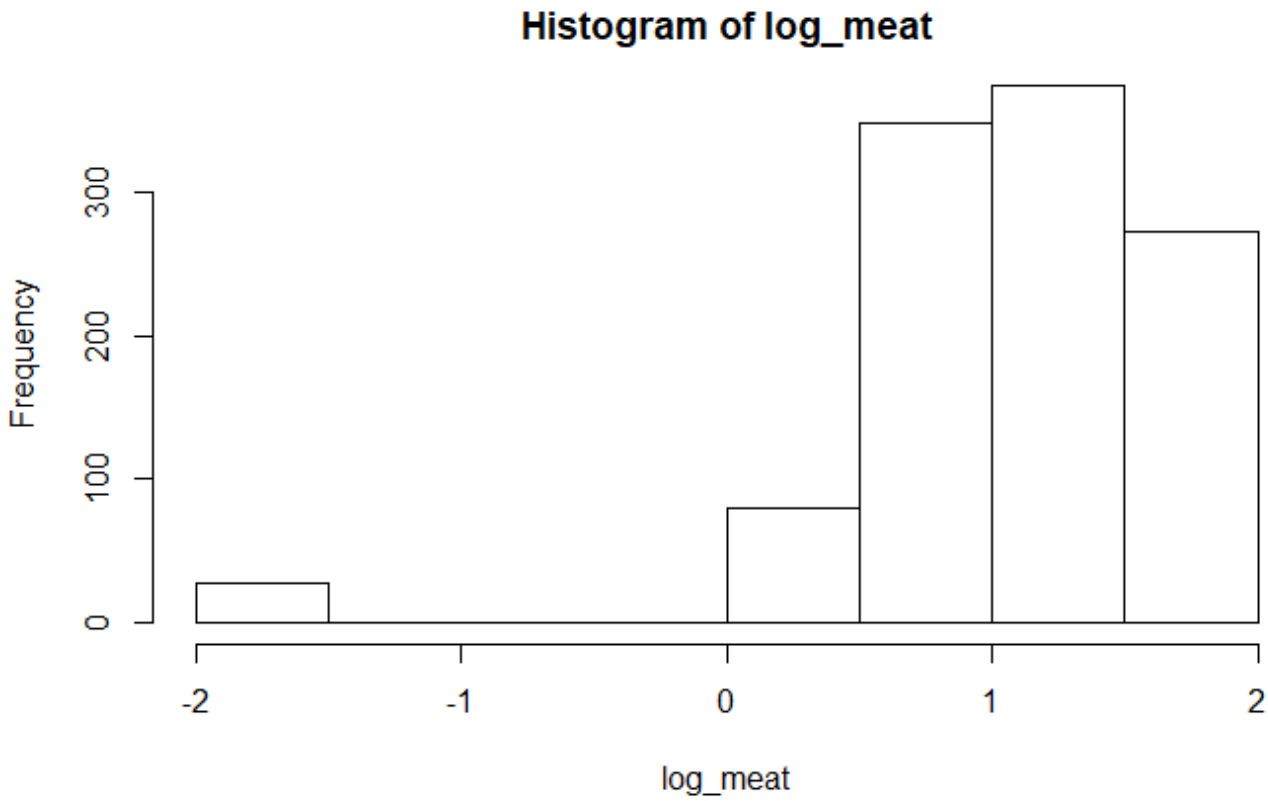
Hide

```
#log10 and ln seem to perfeorm well
hist(meat_cons_pop$`Meat consumption (kg/capita)`)
```



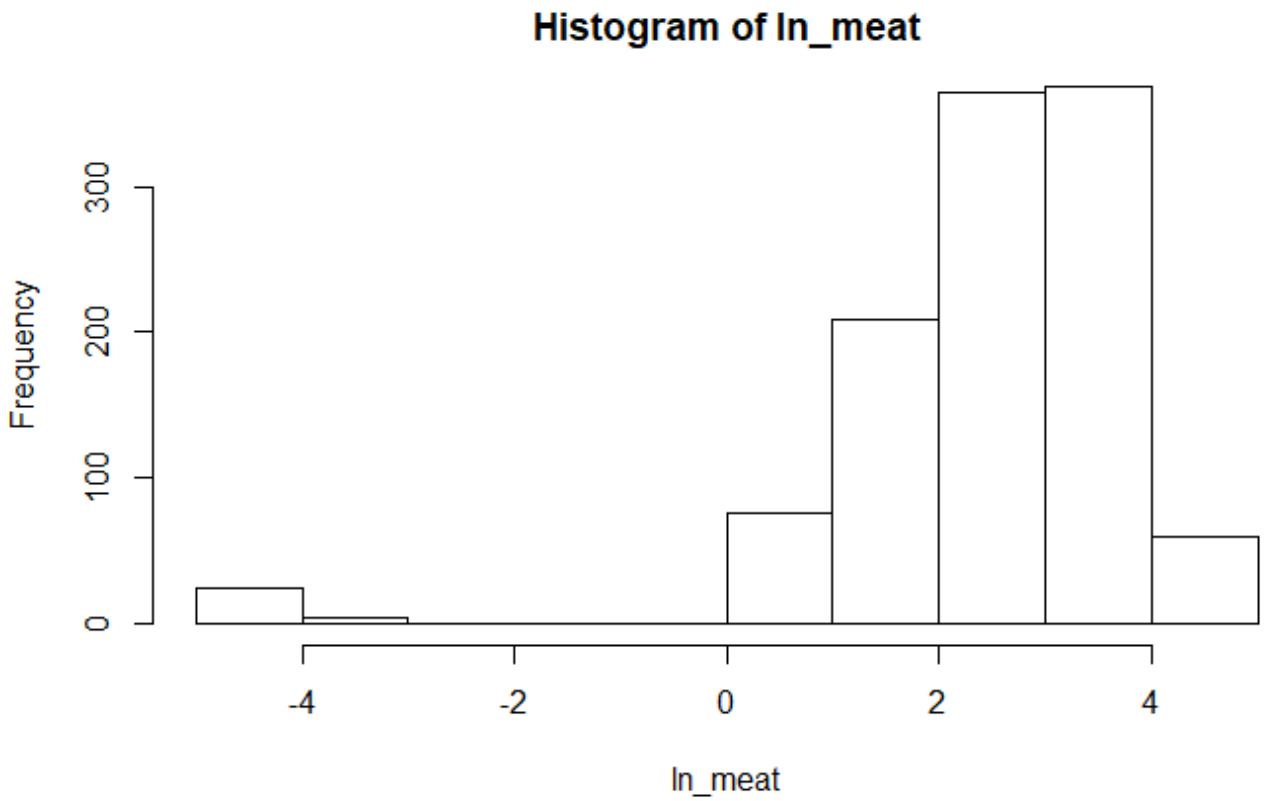
Hide

```
log_meat <- log10(meat_cons_pop$`Meat consumption (kg/capita)`)
hist(log_meat)
```



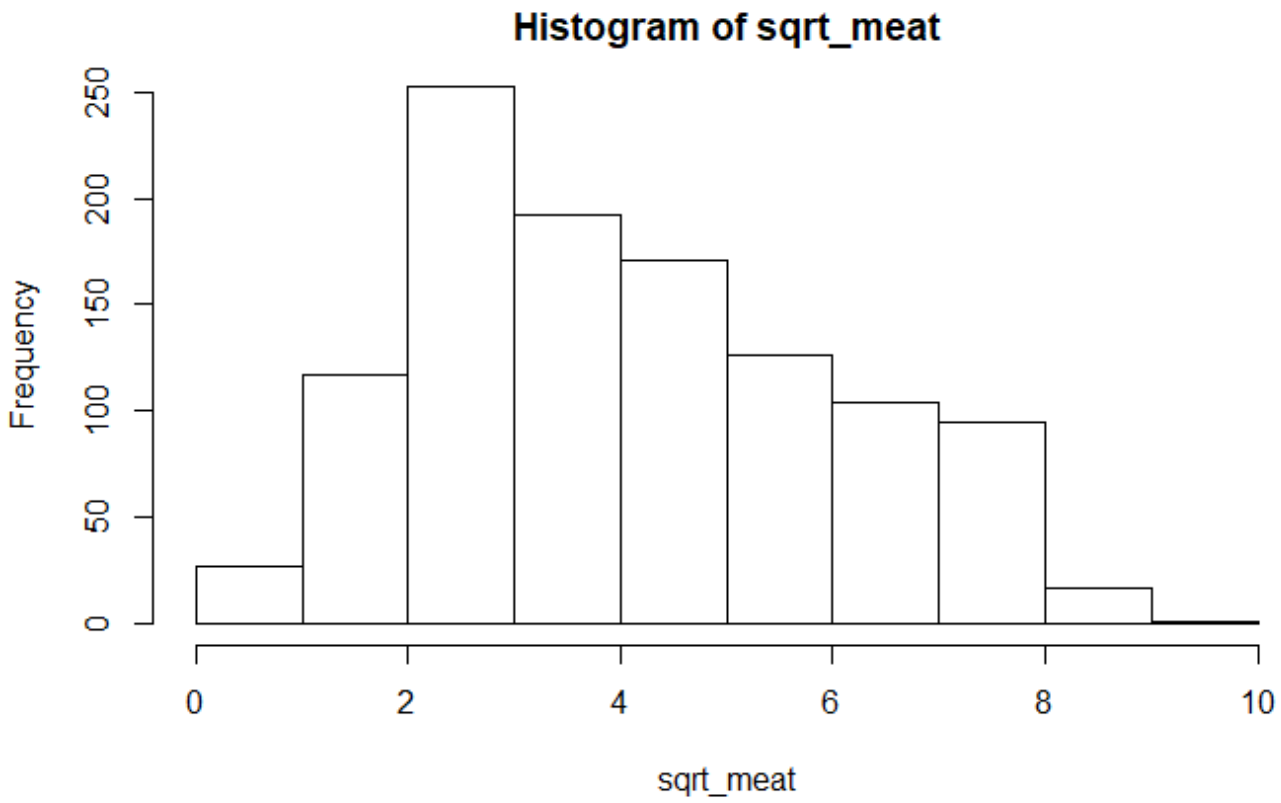
Hide

```
ln_meat<-log(meat_cons_pop$`Meat consumption (kg/capita)`)
hist(ln_meat)
```



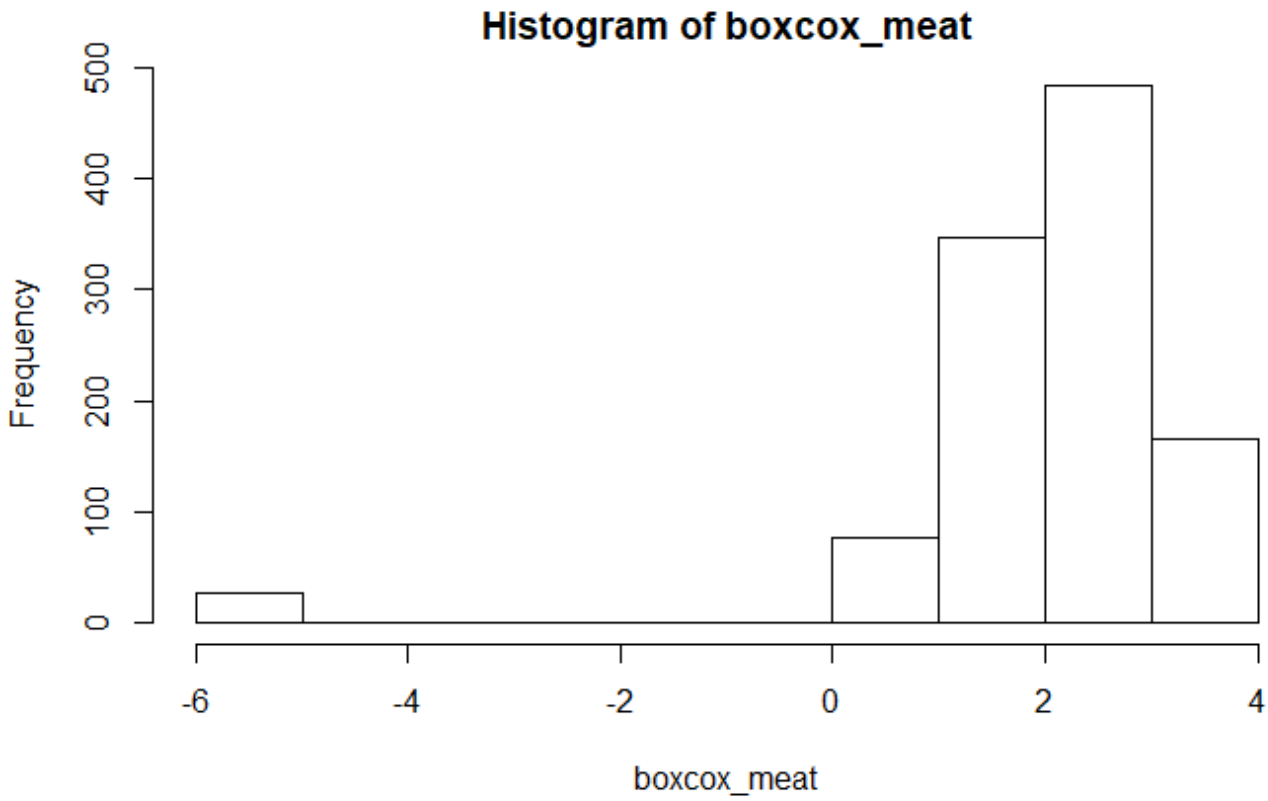
Hide

```
sqrt_meat<-sqrt(meat_cons_pop$`Meat consumption (kg/capita)`)
hist(sqrt_meat)
```



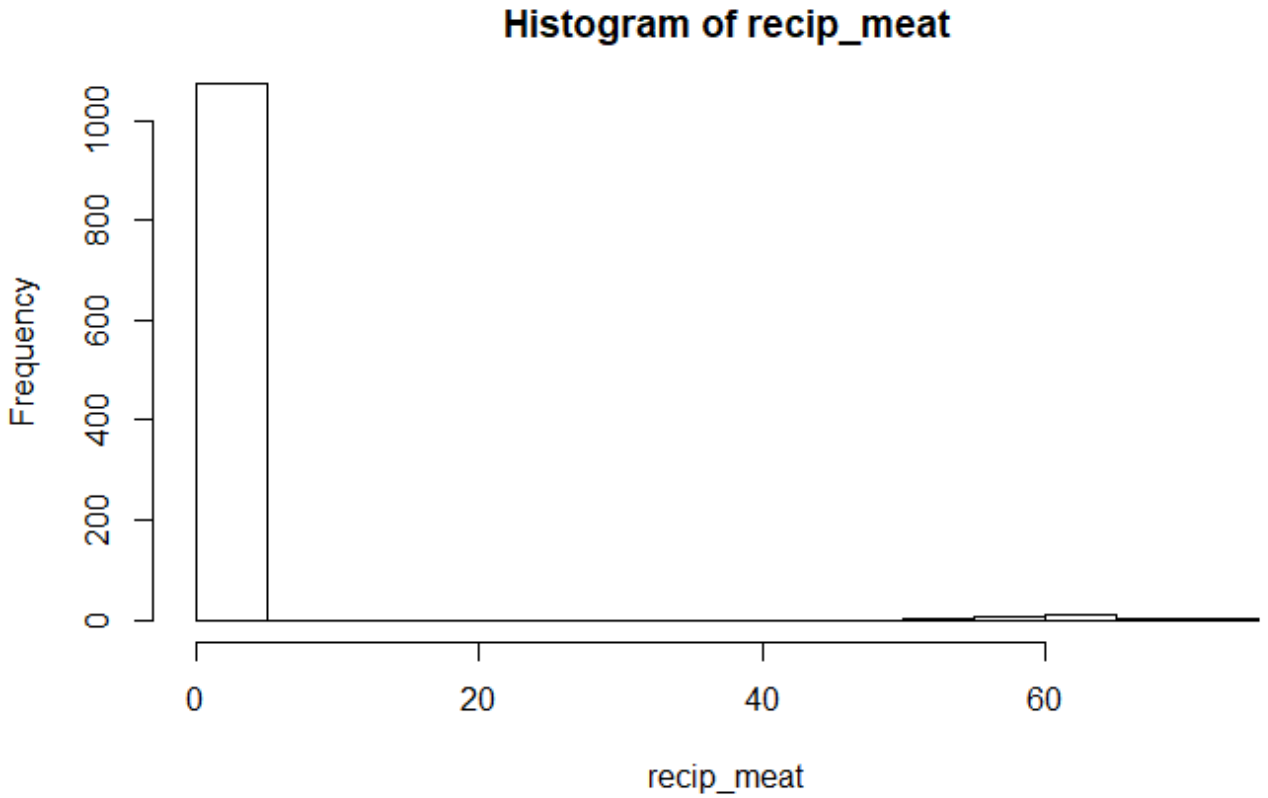
Hide

```
boxcox_meat<- BoxCox(meat_cons_pop$`Meat consumption (kg/capita)`,lambda = "auto")
hist(boxcox_meat)
```



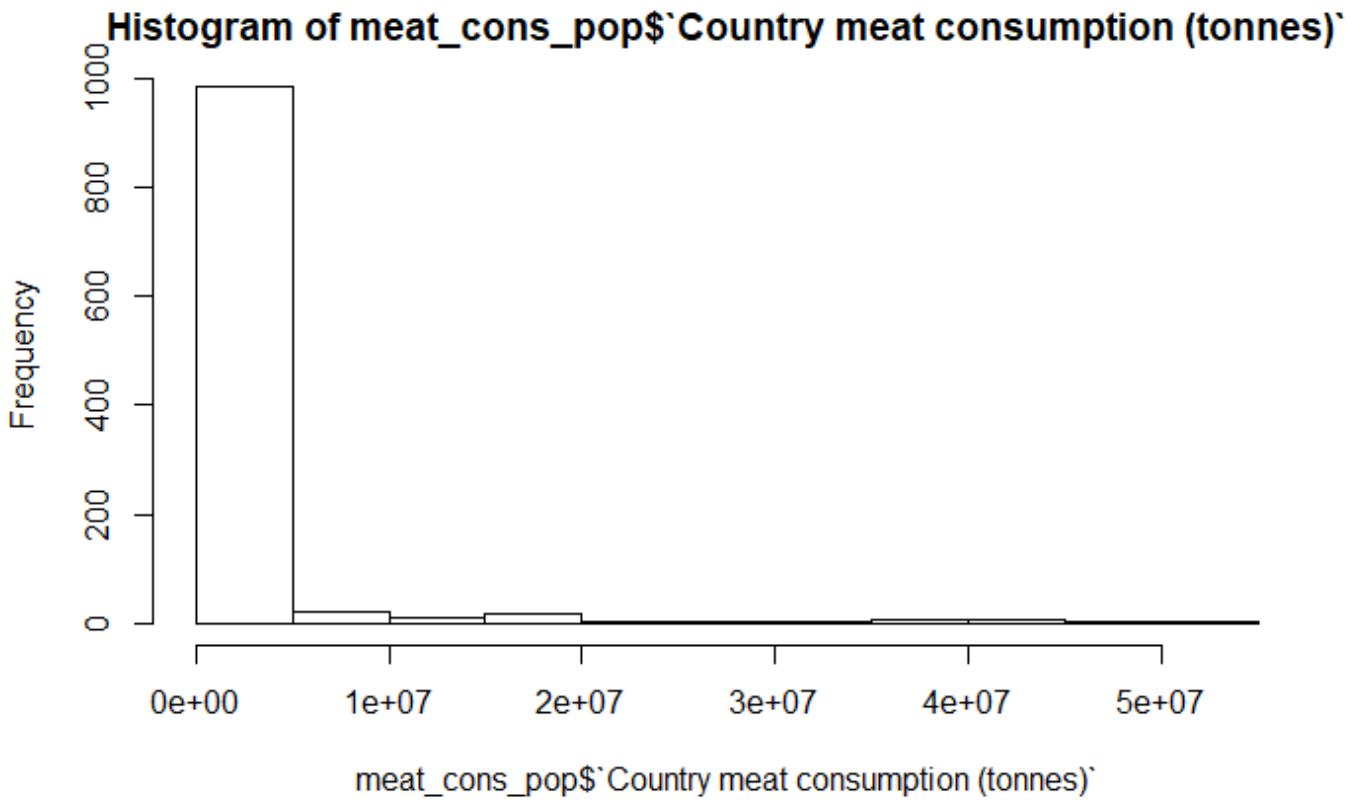
Hide

```
recip_meat <- 1/meat_cons_pop$`Meat consumption (kg/capita)`
hist(recip_meat)
```



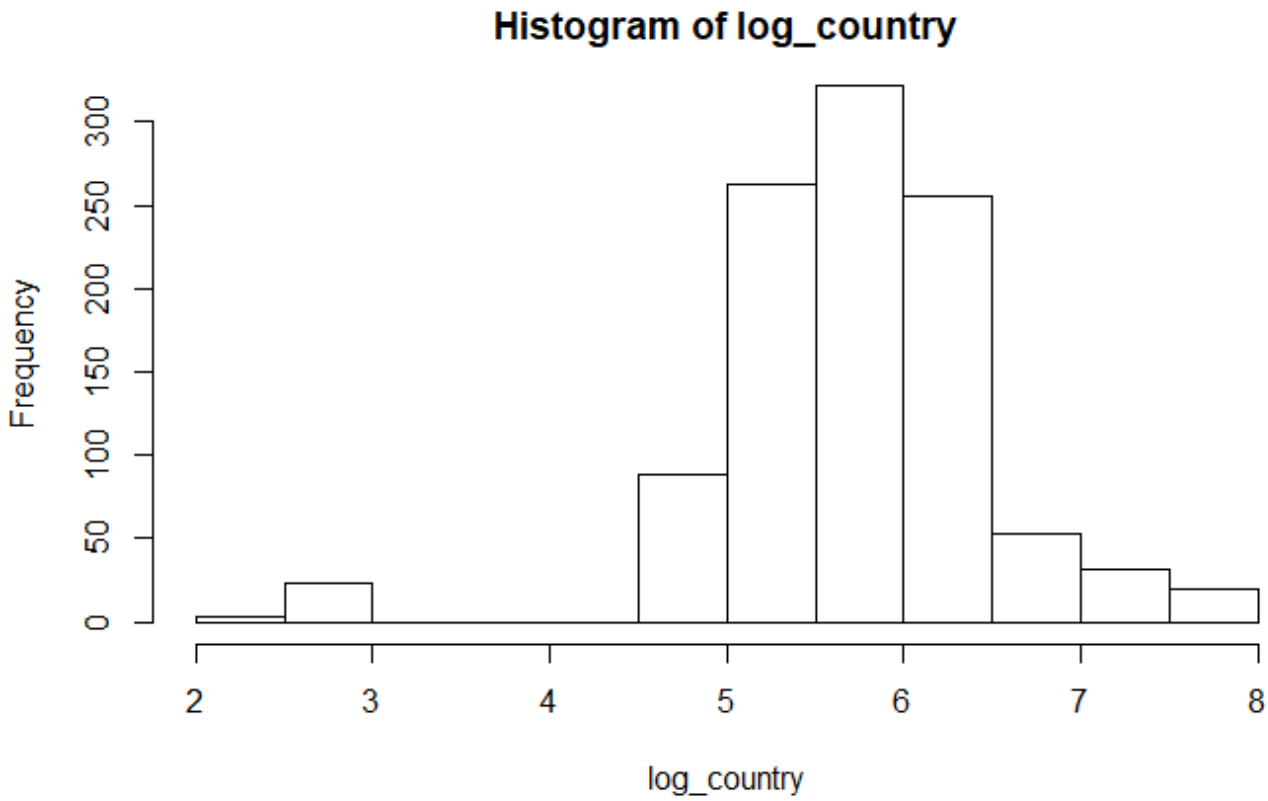
Hide

```
#square root seem to perfeom best in this case
hist(meat_cons_pop$`Country meat consumption (tonnes)`)
```



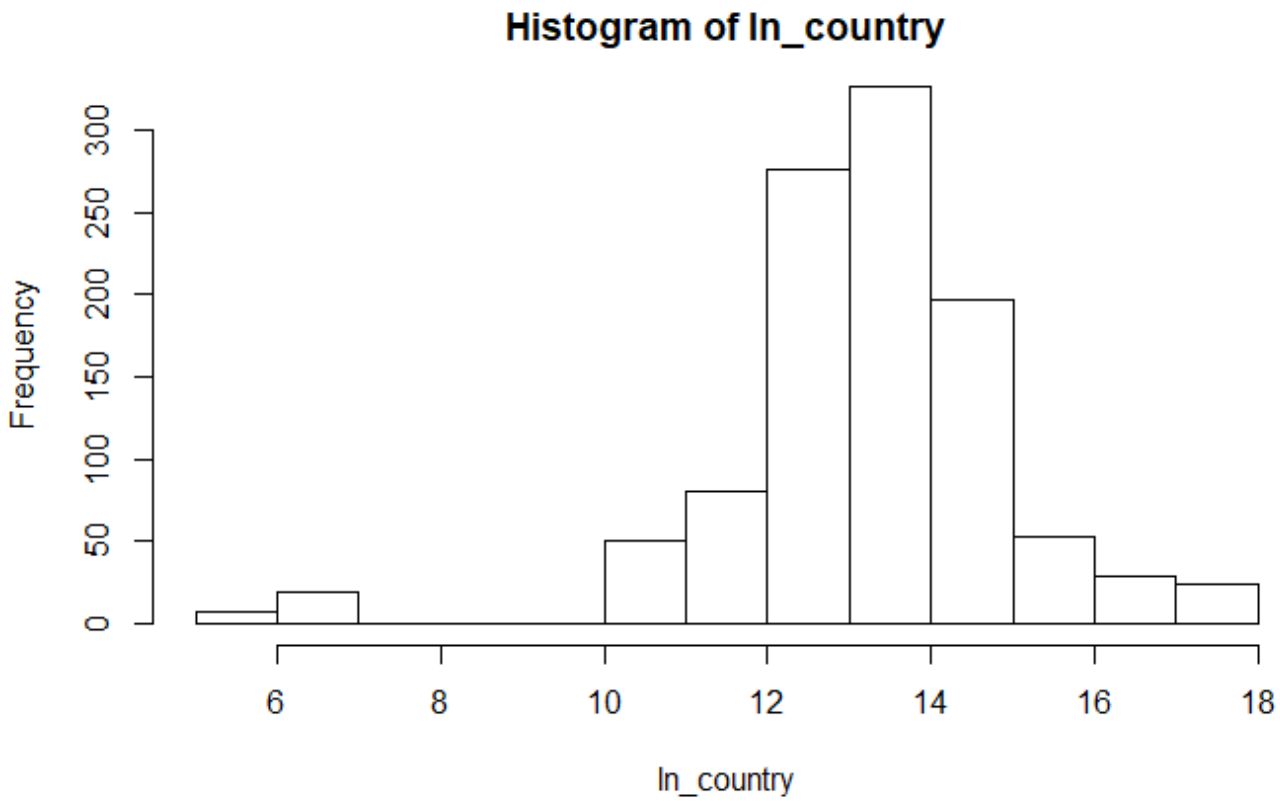
Hide

```
log_country <- log10(meat_cons_pop$`Country meat consumption (tonnes)`)
hist(log_country)
```



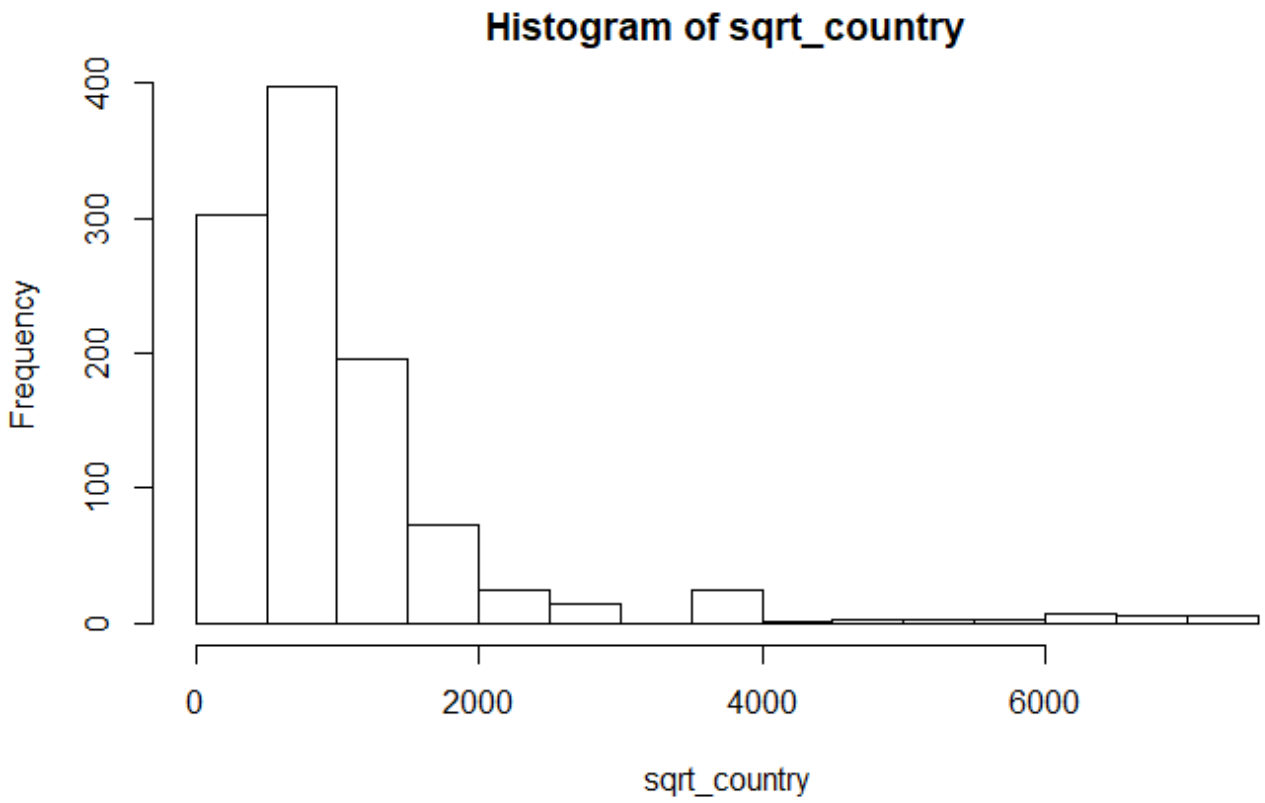
Hide

```
ln_country<-log(meat_cons_pop$`Country meat consumption (tonnes)`)
hist(ln_country)
```



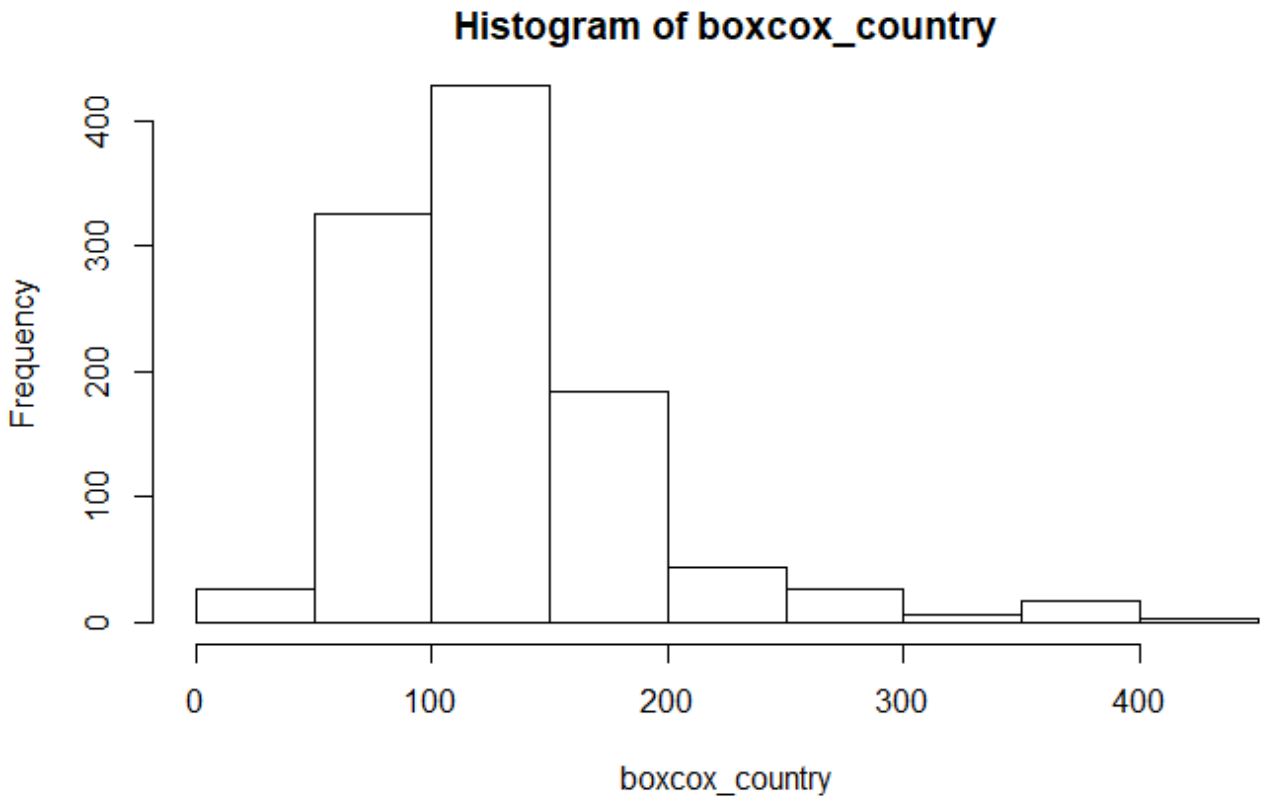
Hide

```
sqrt_country<-sqrt(meat_cons_pop$`Country meat consumption (tonnes)`)
hist(sqrt_country)
```



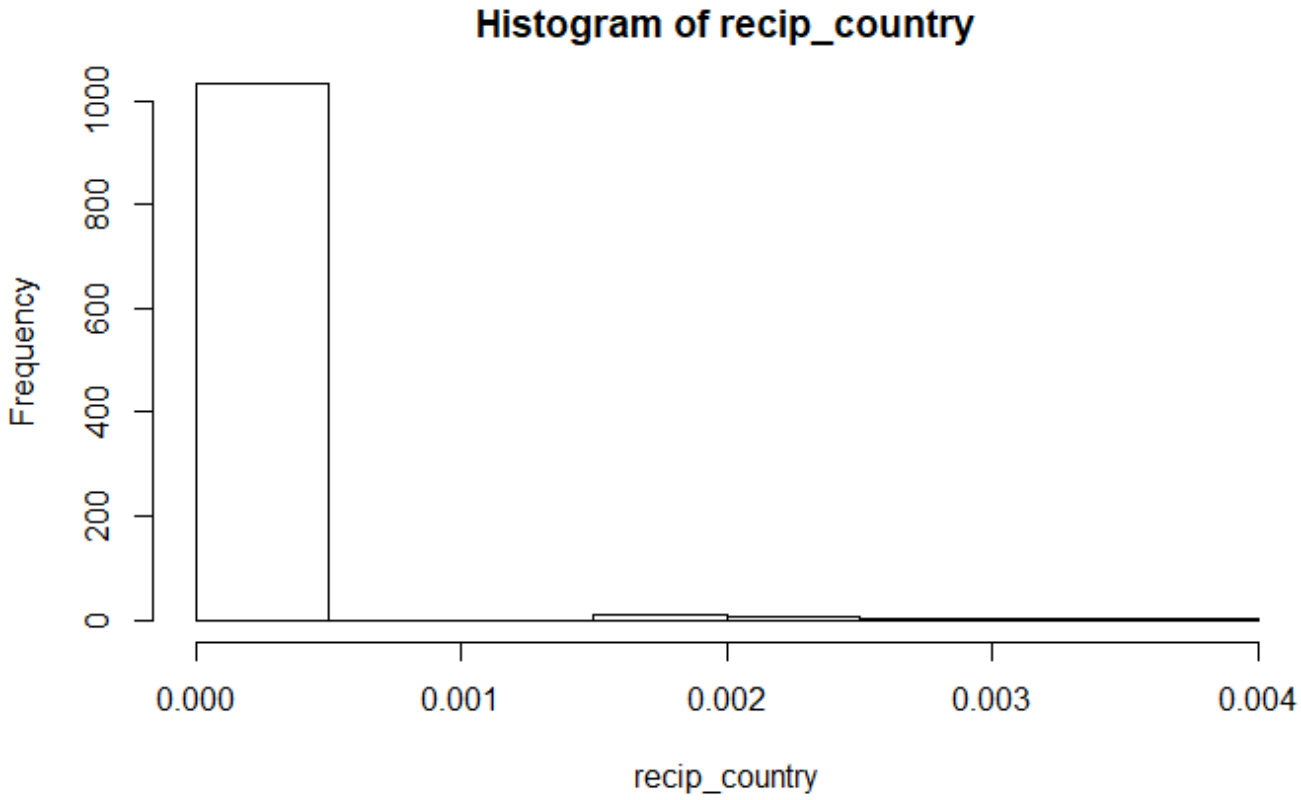
Hide

```
boxcox_country<- BoxCox(meat_cons_pop$`Country meat consumption (tonnes)`,lambda = "au
to")
hist(boxcox_country)
```



Hide

```
recip_country <- 1/meat_cons_pop$`Country meat consumption (tonnes)`
hist(recip_country)
```



Hide

```
#the log10 and ln seem to perform well here
```

Others codes

Hide

```
#Use filter() and summarise() to find out wich country eats the most meat in a certain
years
cons_by_year <- meat_cons_pop %>% arrange(meat_cons_pop$Year,meat_cons_pop$`Meat consu
mption (kg/capita)`)
cons_by_year
```

Country Name <fctr>	**Y…** <int>	**Population (people)** <dbl>	**Beef and veal consumption (kg/capit** <db
Sudan	1991	20893625	0.0086972
Bangladesh	1991	108727432	0.9224983
Mozambique	1991	13591970	0.7653144
India	1991	888054875	1.6221468
Ghana	1991	15039514	2.1405674
Nigeria	1991	97726323	1.4701468
Indonesia	1991	184615979	1.1905874
Zambia	1991	8239732	3.0975748
Haiti	1991	7243391	2.4159958
Tanzania	1991	26315013	5.3216803

1-10 of 1,102 rows | 1-4 of 9 columns Previous **1** 2 3 4 5 6 … 100 Next

Another analysi

Hide

```
#Subseting the meat_cons_pop in order to find some interested information about austra
lia. This process will use some other data sets
meat_prod_aus
```

X__1 <S3: POSIXct>	**red_meat_prod** <dbl>	**pork_prod** <dbl>
1972-07-01	18028	209456
1972-08-01	20091	216225
1972-09-01	18718	192245

X__1 <S3: POSIXct>	red_meat_prod <dbl>	pork_prod <dbl>
1972-10-01	20007	204724
1972-11-01	20629	212670
1972-12-01	17828	187932
1973-01-01	18472	206744
1973-02-01	18166	197938
1973-03-01	20582	224219
1973-04-01	17838	161131

1-10 of 549 rows Previous **1** 2 3 4 5 6 ... 55 Next

Hide

```
#step 1 calculating amount of meat producing by Australia each year by seperating the
dates
str(meat_prod_aus)
```

```
Classes tbl_df, tbl and 'data.frame':   549 obs. of  3 variables:
 $ X__1         : POSIXct, format: "1972-07-01" "1972-08-01" "1972-09-01" ...
 $ red_meat_prod: num  18028 20091 18718 20007 20629 ...
 $ pork_prod    : num  209456 216225 192245 204724 212670 ...
```

Hide

```
meat_prod_aus<- rename(meat_prod_aus, 'Date'='X__1')
str(meat_prod_aus)
```

```
Classes tbl_df, tbl and 'data.frame':   549 obs. of  3 variables:
 $ Date         : POSIXct, format: "1972-07-01" "1972-08-01" "1972-09-01" ...
 $ red_meat_prod: num  18028 20091 18718 20007 20629 ...
 $ pork_prod    : num  209456 216225 192245 204724 212670 ...
```

Hide

```
meat_prod_aus$Date<-as.character(meat_prod_aus$Date)
meat_prod_aus$Date<-as.Date(meat_prod_aus$Date)
meat_prod_aus<- meat_prod_aus %>% tidyr::separate(Date, into = c("Year", "Month","Date
"))
#calculating meat producing in each year
anual_meat_aus<- meat_prod_aus %>% dplyr::mutate(total_meat = red_meat_prod+pork_prod)
%>% dplyr::select("Year","total_meat")
total_meat_anual_aus<-anual_meat_aus %>% group_by(Year)%>% summarise(Meat_produced= su
m(total_meat))    #subseting Australian population and meat consumption from the meat_
cons_pop data set
aus_cons<- meat_cons_pop %>%dplyr::filter(meat_cons_pop$`Country Name` =="Australia")
str(aus_cons)
```

```
Classes tbl_df, tbl and 'data.frame':   27 obs. of  9 variables:
 $ Country Name                          : Factor w/ 41 levels "Algeria","Argentina",..
: 3 3 3 3 3 3 3 3 3 ...
 $ Year                                  : int  1991 1992 1993 1994 1995 1996 1997 1998
1999 2000 ...
 $ Population (people)                   : num  17284000 17495000 17667000 17855000 180
72000 ...
 $ Beef and veal consumption (kg/capita): num  27.7 26.2 26.2 25.5 25.3 ...
 $ Lamb consumption (kg/capita)          : num  19.3 17.8 17.4 18.4 15 ...
 $ Pork consumption (kg/capita)          : num  14.4 15.1 14.9 15.6 15.5 ...
 $ Meat consumption (kg/capita)          : num  61.4 59.2 58.4 59.4 55.9 ...
 $ Country meat consumption (tonnes)     : num  1060904 1034837 1031691 1060933 1009328
...
 $ Country Code                          : Factor w/ 41 levels "ARG","AUS","BGD",..: 2
2 2 2 2 2 2 2 2 2 ...
```

Hide

```
total_meat_anual_aus$Year<- as.integer(total_meat_anual_aus$Year)
a_meat<- aus_cons %>% left_join (total_meat_anual_aus, by =c("Year") ) %>%dplyr:: sele
ct("Year","Country meat consumption (tonnes)","Meat consumption (kg/capita)","Populati
on (people)","Meat_produced" )
str(meat_export)
```

```
Classes tbl_df, tbl and 'data.frame':   121 obs. of  8 variables:
 $ X__1                       : POSIXct, format: "1988-03-01" "1988-06-01" "1988-09-
01" ...
 $ Quantity ;  Beef Bone In ;   : num  3937 4204 4410 12297 20984 ...
 $ Quantity ;  Beef Bone Out ;  : num  154288 152078 136346 132532 98429 ...
 $ Quantity ;  Mutton and Lamb ;: num  52369 30867 20211 40024 38181 ...
 $ Quantity ;  Pork ;           : num  1446 1278 1297 2225 1356 ...
 $ Quantity ;  Bacon and Ham ;  : num  9 22 12 15 8 4 25 3 6 3 ...
 $ Quantity ;  Edible Offal ;   : num  13318 11786 11145 13730 10385 ...
 $ Quantity ;  Preserved Meat ; : num  3095 2789 3062 3867 2707 ...
```

Hide

```
meat_export<- rename(meat_export, 'Date'='X__1')
meat_export$Date<-as.character(meat_export$Date)
meat_export$Date<-as.Date(meat_export$Date)
meat_export<- meat_export %>% tidyr::separate(Date, into = c("Year", "Month","Date"))
anual_meat_export_aus<- meat_export %>% dplyr::mutate(total_meat_export = meat_export$
`Quantity ;  Beef Bone In ;`+meat_export$`Quantity ;  Beef Bone Out ;`+meat_export$`Qu
antity ;  Mutton and Lamb ;`+meat_export$`Quantity ;  Pork ;`+meat_export$`Quantity ;
Bacon and Ham ;`+meat_export$`Quantity ;  Edible Offal ;`+meat_export$`Quantity ;  Pre
served Meat ;`) %>% dplyr::select("Year","total_meat_export")
anual_meat_export_aus$Year<- as.integer(anual_meat_export_aus$Year)
total_meat_export_aus<-anual_meat_export_aus %>% group_by(Year)%>% summarise(Meat_expo
rt_total= sum(total_meat_export))
a_meat<- a_meat %>% left_join (total_meat_export_aus, by =c("Year") )
a_meat<- a_meat %>% dplyr::mutate(total_meat_wasted_tones= a_meat$Meat_produced-a_meat
$`Country meat consumption (tonnes)`-a_meat$Meat_export_total)
a_meat<- a_meat %>% dplyr::mutate(meat_waste_percapita=total_meat_wasted_tones*1000/ a
_meat$`Population (people)`)
mean(a_meat$`Meat consumption (kg/capita)`, na.rm = TRUE)
```

```
[1] 55.38919
```

Hide

```
mean(a_meat$total_meat_wasted_tones, na.rm= TRUE)
```

```
[1] 897115.1
```

Hide

```
mean(a_meat$meat_waste_percapita,na.rm = TRUE)
```

```
[1] 45.11226
```

Hide

```
mean(a_meat$`Country meat consumption (tonnes)`,na.rm = TRUE)
```

```
[1] 1122924
```

Hide

```
statement<- cat("From 1991 to 2017, Autralia has produced " , mean(a_meat$Meat_produce
d,na.rm = TRUE), " tones on non-poultry meat per year on average. During this time, an
average Australian has consumed " , mean(a_meat$`Meat consumption (kg/capita)`, na.rm
= TRUE) ," kilogram each year. The country has managed to export " , mean(a_meat$Meat_
export_total, na.rm = TRUE ), " tonnes each year, on average. This means Australia as
a country have wasted or lost " , mean(a_meat$total_meat_wasted_tones, na.rm= TRUE), "
tonnes of meat each year, which worked out to be "  ,mean(a_meat$meat_waste_percapita,
na.rm = TRUE), " kiligram per person, per year. This is the equivelent of $" ,mean(a_m
eat$meat_waste_percapita,na.rm = TRUE)*10 , " wasted- assuming the meat price is 10$ p
er kilogram on average" )
```

```
From 1991 to 2017, Autralia has produced  3432335  tones on non-poultry meat per year
on average. During this time, an average Australian has consumed  55.38919  kilogram e
ach year. The country has managed to export  1414909  tonnes each year, on average. Th
is means Australia as a country have wasted or lost  897115.1  tonnes of meat each yea
r, which worked out to be  45.11226  kiligram per person, per year. This is the equive
lent of $ 451.1226  wasted- assuming the meat price is 10$ per kilogram on average
```