Get and Load Data

ENTER RAW DATA:

1. In Excel, enter variables in columns with variable names in the first row, each individual's data in rows below that (do not use spaces or special characters).
2. Save as "Comma Separated Values (*.CSV)" file in your local directory/folder.

DATA PROVIDED BY PROFESSOR:

1. Goto the MTH107 Resources webpage.
2. Save "data" link (right-click) to your local directory/folder.

LOAD THE EXTERNAL CSV FILE INTO R:

1. Start script and save it in the same folder with the CSV file.
2. Select the Session, Set Working Directory, To Source File Location menus.
3. Copy resulting setwd() code to your script.
4. Use read.csv() to load the data into the dfobj object.

```
dfobj <- read.csv("filename.csv")
```

5. Observe the structure of the dfobj object.

```
str(dfobj)
```

```
> library(NCStats)
> setwd("C:/aaaWork/Web/GitHub/NCMTH107")
> dfcar <- read.csv("93cars.csv")
> str(dfcar)
'data.frame':    93 obs. of  26 variables:
  $ Type     : Factor w/ 6 levels "Compact","Large": 4 3 3 ...
  $ HMPG     : int  31 25 26 26 30 31 28 25 27 25...
  $ Manual   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 ...
  $ Weight   : int  2705 3560 3375 3405 3640 2880 3470 ...
  $ Domestic: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 ...
```

Filter Individuals

Individuals may be selected from the dfobj data.frame and put in a the new newdf data.frame according to a condition with

```
newdf <- filterD(dfobj,condition)
```

where condition may be as follows

```
var == value          # equal to
var != value          # not equal to
var > value           # greater than
var >= value          # greater than or equal
var %in% c("val","val","val")   # in the list
cond, cond            # both conditions met
```

with var replaced by a variable name and value replaced by a number or category level (*if value is not a number then it must be put in quotes*).

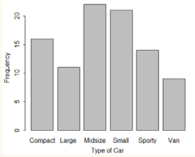
```
> justSporty <- filterD(dfcar,Type=="Sporty")
> noDomestic <- filterD(dfcar,Domestic!="Yes")
> justHMPGgt30 <- filterD(dfcar,HMPG>30)
> Sp_or_Sm <- filterD(dfcar,Type %in% c("Sporty","Small"))
> Spry_n_gt30 <- filterD(dfcar,Type=="Sporty",HMPG>30)
> justWTlteq3000 <- filterD(dfcar,Weight<=3000)
> justNum17 <- dfcar[17,]
> notNum17 <- dfcar[-17,]
```

Univariate EDA

CATEGORICAL – Frequency table, percentage table, and bar chart for the cvar variable.

```
( freq1 <- xtabs(~cvar,data=dfobj) )
percTable(freq1,digits=1)
barplot(freq1,xlab="better cvar label", ylab="Frequency")
```

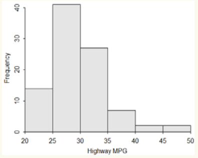
```
> ( freq1 <- xtabs(~Type,data=dfcar) )
 Compact   Large Midsize   Small  Sporty     Van
      16      11      22      21      14       9
> percTable(freq1,digits=1)
 Compact   Large Midsize   Small  Sporty     Van     Sum
    17.2    11.8    23.7    22.6    15.1     9.7   100.1
> barplot(freq1,xlab="Type of Car",ylab="Frequency")
```



QUANTITATIVE – Summary statistics (mean, median, SD, IQR, etc.) and a histogram for the qvar variable.

```
hist(~qvar,data=dfobj,xlab="better qvar label")
Summarize(~qvar,data=dfobj,digits=#)
```

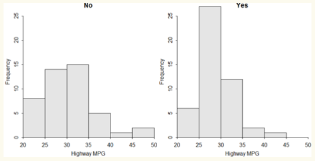
```
> Summarize(~HMPG,data=dfcar,digits=1)
     n    mean      sd     min      Q1  median      Q3     max
  93.0    29.1     5.3    20.0    26.0    28.0    31.0    50.0
> hist(~HMPG,data=dfcar,xlab="Highway MPG")
```



QUANTITATIVE BY GROUP – Summary statistics and histograms for the qvar separated by groups in the cvar.

```
hist(qvar~cvar,data=dfobj,xlab="better qvar label")
Summarize(qvar~cvar,data=dfobj,digits=#)
```

```
> Summarize(HMPG~Domestic,data=dfcar,digits=1)
  Domestic  n mean  sd min Q1 median Q3 max
1       No 45 30.1 6.2  21 25     30 33  50
2      Yes 48 28.1 4.2  20 26     28 30  41
> hist(HMPG~Domestic,data=dfcar,xlab="Highway MPG")
```



Bivariate EDA

CATEGORICAL – Frequency and percentage tables for the cvarRow and cvarCol variables.

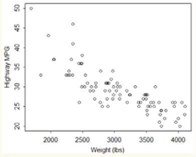
```
( freq2 <- xtabs(~cvarRow+cvarCol, data=dfobj) )
percTable(freq2,digits=#1)                # total/table %
percTable(freq2,digits=1,margin=1)      # row %
percTable(freq2,digits=1,margin=2)      # column %
```

```
> ( freq2 <- xtabs(~Domestic+Manual,data=dfcar) )
           Manual
  Domestic No Yes
       No   6  39
       Yes 26  22
> percTable(freq2,digits=1)
           Manual
  Domestic    No   Yes   Sum
       No    6.5  41.9  48.4
       Yes  28.0  23.7  51.7
       Sum  34.5  65.6 100.1
> percTable(freq2,margin=1,digits=1)
           Manual
  Domestic    No    Yes   Sum
       No   13.3   86.7 100.0
       Yes  54.2   45.8 100.0
> percTable(freq2,margin=2,digits=1)
           Manual
  Domestic    No    Yes
       No   18.8   63.9
       Yes  81.2   36.1
       Sum 100.0  100.0
```

QUANTITATIVE – Correlation (r) and scatterplot for the qvarY and qvarX variables.

```
plot(qvarY~qvarX,data=dfobj, ylab="better yvar label",
     xlab="better xvar label")
corr(~qvarY+qvarX,data=dfobj,digits=3)
```

```
> plot(HMPG~Weight,data=dfcar,ylab="Highway MPG")
       xlab="Weight (lbs)")
```



```
> corr(HMPG~Weight,data=dfcar,digits=3)
  [1] -0.811
```

QUANTITATIVE (ALL PAIRS) – Correlation (r) and scatterplot for all paris of quantitative variables.

```
pairs(~qvar1+qvar2+qvar3,data=dfobj, pch=21,bg="gray70")
corr(~qvar1+qvar2+qvar3,data=dfobj,digits=3,
          use="pairwise.complet.obs")
```

```
> plot(~HMPG+Weight+Cyl,data=dfcar,pch=21,bg="gray70")
> corr(HMPG+Weight+Cyl,data=dfcar,digits=3,
        use="pairwise.complete.obs")
```

Normal Distributions

distrib(val,mean=mnval,sd=sdval,lower.tail=FALSE, type="q")

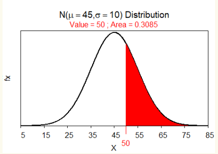
where

- **val** is a value of the quantitative variable (x) or an area (i.e., a percentage provided as a proportion).
- **mnval** is the population mean (μ)
- **sdval** is the standard deviation (σ) or error (SE)
- **type="q"** is included for reverse calculations
- **lower.tail=FALSE** is included for "right-of" calculations

For SE use (where **nval**=sample size):

sd=sdval/sqrt(nval)

```
> distrib(50,mean=45,sd=10,lower.tail=FALSE)   #forward-right
```



```
> distrib(50,mean=45,sd=10)                      #forward-left
> distrib(0.05,mean=45,sd=10,type="q")           #rev-left
> distrib(0.2,mean=45,sd=10,type="q",lower.tail=FALSE) #rev-rgt
> distrib(50,mean=45,sd=10/sqrt(30))             #using SE
> distrib(0.95,mean=45,sd=10/sqrt(30),
          type="q",lower.tail=FALSE)             #using SE
```

Linear Regression

The best-fit line between the **rspvar** response and **expvar** explanatory variables.

(bfl <- lm(rspvar~expvar,data=dfobj))

A visual of the best-fit line.

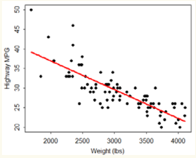
fitPlot(bfl,ylab="rspvar lbl",xlab="expvar lbl")

The r^2 value.

rSquared(bfl)

```
> ( bfl <- lm(HMPG~Weight,data=dfcar) )
  Coefficients:
  (Intercept)       Weight
    51.601365    -0.007327

> fitPlot(bfl,ylab="Highway MPG",xlab="Weight (lbs)")
```



```
> rSquared(bfl)
  [1] 0.6571665
```

Quantitative Hypothesis Tests

ONE SAMPLE Z-TEST AND T-TEST:

z.test(dfobj$qvar,mu=mu0,alt=HA, conf.level=cnfval,sd=sdval)
t.test(dfobj$qvar,mu=mu0,alt=HA, conf.level=confval)

- **qvar** is the quantitative. response variable in **dfobj**
- **mu0** is the population mean in H_0
- **HA** is replaced with **"two.sided"** for a not equals, **"less"** for a less than, or **"greater"** for a greater than H_A
- **cnfval** is the confidence level (e.g., 0.95)
- **sdval** is the popn. standard deviation (σ)

```
> z.test(dfcar$HMPG,mu=26,alt="greater",conf.level=0.95,sd=6)
  z= 4.9601, n= 93, Std. Dev= 6.000, Std. Dev of the sample
  mean = 0.622, p-value = 3.523e-07
  alternative hypothesis: true mean is greater than 26
  95 percent confidence interval:
   28.06264       Inf
  sample estimates:
  mean of dfcar$HMPG
            29.08602

> t.test(dfcar$HMPG,mu=26,alt="two.sided",conf.level=0.99)
  t = 5.5818, df = 92, p-value = 2.387e-07
  alternative hypothesis: true mean is not equal to 26
  99 percent confidence interval:
   27.63178 30.54026
  sample estimates:
  mean of x
   29.08602
```

Quantitative Hypothesis Tests

TWO SAMPLE T-TEST:

levenesTest(qvar~cvar,data=dfobj)
t.test(qvar~cvar,data=dfobj,alt=HA , conf.level=cnfval,
var.equal=TRUE)

- **qvar** is the quantitative response variable in **dfobj**
- **mu0** is the population mean in H_0
- **HA** is replaced with **"two.sided"** for a not equals, **"less"** for a less than, or **"greater"** for a greater than H_A
- **cnfval** is the confidence level (e.g., 0.95)
- **cvar** is a categorical variable in **dfobj that identifies the groups**
- **var.equal=TRUE** if the popn variances are thought to be equal

```
> levenesTest(HMPG~Domestic,data=dfcar)
        Df F value  Pr(>F)
  group  1  5.3595 0.02286
        91

> t.test(HMPG~Manual,data=dfcar,alt="less",conf.level=0.99,
         var.equal=TRUE)
  t = -4.2183, df = 91, p-value = 2.904e-05
  alt. hypothesis: true difference in means is less than 0
  99 percent confidence interval:
        -Inf -1.980103
  sample estimates:
   mean in group No mean in group Yes
           26.12500          30.63934
```

Categorical Hypothesis Tests

(TWO SAMPLE) CHI-SQUARE TEST:

Chi-square for two-way frequency in **obstbl** (with the **rspvar** response variable in columns and the populations in **popvar** as rows).

(obstbl <- xtabs(~popvar+rspvar,data=dfobj))
(chi <- chisq.test(obstbl,correct=FALSE))

Follow-up Analyses:

- Extract the expected values.

chi$expected

- Percentages of individuals in each level of the response variable for each population.

percTable(obstbl,digits=1,margin=1)   # row percent table

```
> ( freq2 <- xtabs(~Domestic+Manual,data=dfcar) )
          Manual
  Domestic No Yes
       No   6  39
       Yes 26  22

> ( chi <- chisq.test(freq2,correct=FALSE) )
  Pearson's Chi-squared test with freq2
  X-squared = 17.1588, df = 1, p-value = 3.438e-05

> chi$expected
          Manual
  Domestic       No      Yes
       No  15.48387 29.51613
       Yes 16.51613 31.48387

> percTable(freq2,margin=1,digits=1)
          Manual
  Domestic    No   Yes   Sum
       No   13.3  86.7 100.0
       Yes  54.2  45.8 100.0
```

Categorical Hypothesis Tests

(ONE SAMPLE) GOODNESS-OF-FIT TEST:

Goodness-of-fit for one-way frequency in **obstbl** and expected values (or ratios) in **exp.p**.

(obstbl <- c(lvl1=##,lvl2=##,lvl3=##))   # if summarized data
(obstbl <- xtabs(~popvar+rspvar,data=dfobj))   # if raw data
(exp.p <- c(lvl1=##,lvl2=##,lvl3=##))
(gof <- chisq.test(obstbl,p=exp.p,rescale.p=TRUE,
correct=FALSE))

Follow-up Analyses:

- Extract the expected values.

gof$expected

- Percentages of individuals in each level of the response variable.

percTable(obstbl,digits=1)