

R Function Guide

Load Packages

The **NCStats** and **ggplot2** packages should ALWAYS be loaded with `library()` at the top/beginning of your new script in RStudio.

```
> library(NCStats)
> library(ggplot2)
```

Randomize Individuals

EXPERIMENT – Randomly order **N** individuals.

```
sample(N)
```

OBSERVATIONAL STUDY – Randomly select **n** from **N** individuals.

```
sample(N,n)
```

```
> sample(10) # randomly order 1 to 10
[1] 6 7 9 8 1 2 10 5 3 4
> sample(10,3) # randomly select 3 from 1 to 10
[1] 10 4 5
```

Normal Distributions

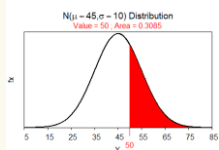
```
distrib(val,mean=mnval,sd=sdval,lower.tail=FALSE,type="q")
```

- **val** is a value of the quantitative variable (x) or an area (i.e., a percentage, but entered as a proportion)
- **mnval** is the population mean (μ)
- **sdval** is the standard deviation (σ) or error (SE)
 - For SE use (where **nval** is the sample size):

```
sd=sdval/sqrt(nval)
```

- **lower.tail=FALSE** is included for “right-of” calculations
- **type="q"** is included for reverse calculations

```
> 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") #reverse-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
> distrib(.025,type="q") #Z*, not =, alpha=.05
```

Hints:

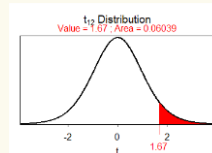
- **Green code** typed exactly as shown
- **Red code** is optional or must be replaced with context-specific name or value
- Replace **qvar** with quantitative variable name
- Replace **cvar** with categorical variable name
- Replaced **#** with numeric value

t Distributions

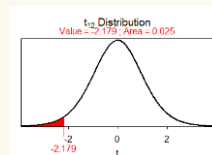
```
distrib(val,distrib="t",df=dfval,lower.tail=FALSE,type="q")
```

- **val** is a value of the t test statistic (for computing the p -value) or an area as a proportion (for computing t^* for confidence region)
- **dfval** is the degrees-of-freedom (df)
- **lower.tail=FALSE** is included for “right-of” calculations
- **type="q"** is included for reverse (confidence region) calculations

```
> distrib(1.67,distrib="t",df=12,lower.tail=FALSE) # p-value
```



```
> distrib(0.025,distrib="t",df=12,type="q") # t-star
```

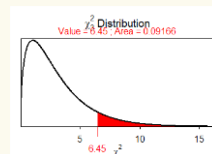


χ^2 Distributions

```
distrib(val,distrib="chisq",df=dfval,lower.tail=FALSE)
```

- **val** is a value of the χ^2 test statistic (for computing the p -value)
- **dfval** is the degrees-of-freedom (df)
- **lower.tail=FALSE** is included for ALL calculations

```
> distrib(6.45,distrib="chisq",df=3,lower.tail=FALSE) # p-value
```



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 “CSV (comma delimited)” file in your local directory/folder (a “.csv” extension will be automatically added to your filename).

DATA PROVIDED BY PROFESSOR:

1. Follow “data” link or goto the [MTH107 Resources webpage](#).
2. Save “data” link (right-click) to your local directory/folder.

LOAD THE EXTERNAL CSV FILE INTO R:

1. Open “R Assignment Template.Rmd” file.
2. Change title on line 2 (keep the quotes). Save file with new name.
3. In an R code chunk (between ````{R}``` and `````), do the following ...
4. Use **read.csv()** to load data in *filename.csv* into **dfobj**.

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

5. Observe the structure of **dfobj**.

```
str(dfobj)
```

```
> 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 the **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("value", "value", "value")	# in the list
cond, cond	# both conditions met

with **var** replaced by a variable name and **value** replaced by a number or category name (*if not a number then must be 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"))
> Sprty_n_gt30 <- filterD(dfcar,Type=="Sporty",HMPG>30)
> justWTlteq3000 <- filterD(dfcar,Weight<=3000)

> justNum17 <- dfcar[17,]
> notNum17 <- dfcar[!17,]
```

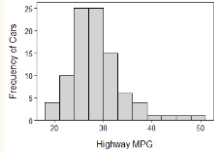
Univariate EDA - Quantitative

Summary statistics (mean, median, SD, Q1, Q3, etc.) and histogram of **qvar** quantitative variable in **dfobj** data.frame.

```
ggplot(data=dfobj,mapping=aes(x=qvar)) +
  geom_histogram(binwidth=#,boundary=0,
    color="black",fill="lightgray") +
  labs(y="Frequency of XXX",x="better qvar label") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats()
Summarize(~qvar,data=dfobj,digits=#)
```

- # in **digits**= is the desired number of decimal places
- # in **binwidth**= is the desired width of bins/bars
- **XXX** in **labs()** is a label/description of an individual

```
> ggplot(data=dfcar,mapping=aes(x=HMPG)) +
  geom_histogram(binwidth=3,boundary=0,
    color="black",fill="lightgray") +
  labs(y="Frequency of Cars",x="Highway MPG") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats()
```



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

Univariate EDA - Categorical

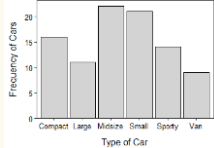
Frequency & percentage tables, bar chart of **cvar** categorical variable.

```
(freq1 <- xtabs(~cvar,data=dfobj))
percTable(freq1)
ggplot(data=dfobj,mapping=aes(x=cvar)) +
  geom_bar(color="black",fill="lightgray") +
  labs(y="Frequency of XXX",x="better cvar label") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats()
```

```
> (freq1 <- xtabs(~Type,data=dfcar))
Compact Large Midsize Small Sporty Van
16      11      22      21      14      9
```

```
> percTable(freq1)
Compact Large Midsize Small Sporty Van
17.2    11.8    23.7    22.6    15.1    9.7
```

```
> ggplot(data=dfcar,mapping=aes(x=Type)) +
  geom_bar(color="black",fill="lightgray") +
  labs(y="Frequency of Cars",x="Type of Car") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats()
```



Univariate EDA - Quant by Groups

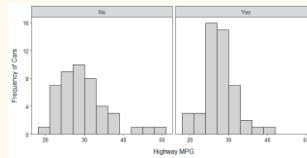
Separate histograms by "adding" (+) this to code for single histogram.

```
facet_wrap(vars(cvar))
```

Separate summary statistics of **qvar** by groups in **cvar**.

```
Summarize(qvar~cvar,data=dfobj,digits=#)
```

```
> ggplot(data=dfcar,mapping=aes(x=HMPG)) +
  geom_histogram(binwidth=3,boundary=0,
    color="black",fill="lightgray") +
  labs(y="Frequency of Cars",x="Highway MPG") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats() +
  facet_wrap(vars(Domestic))
```



```
> 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
```

Bivariate EDA - Categorical

Frequency and percentage tables for **cvarRow** and **cvarCol** variables.

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

```
> freq2 <- xtabs(~Domestic+Manual,data=dfcar)
> addmargins(freq2)
      Manual
Domestic No Yes Sum
No       6  39  45
Yes     26  22  48
Sum     32  61  93
```

```
> percTable(freq2)
      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)
      Manual
Domestic No Yes Sum
No     13.3 86.7 100.0
Yes    54.2 45.8 100.0
```

```
> percTable(freq2,margin=2)
      Manual
Domestic No Yes
No     18.8 63.9
Yes    81.2 36.1
Sum   100.0 100.0
```

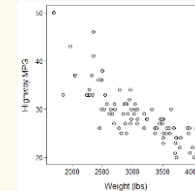
Bivariate EDA - Quantitative

Correlation coefficient (r) and scatterplot for **qvar1** and **qvar2**.

```
ggplot(data=dfobj,mapping=aes(y=qvar1,x=qvar2)) +
  geom_point(pch=21,color="black",fill="lightgray") +
  labs(x="better qvar1 label",y="better qvar2 label") +
  theme_NCStats()
```

```
corr(~qvar1+qvar2,data=dfobj,digits=3)
```

```
> ggplot(data=dfcar,mapping=aes(y=HMPG,x=Weight)) +
  geom_point(pch=21,color="black",fill="lightgray") +
  labs(y="Highway MPG",x="Weight (lbs)") +
  theme_NCStats()
```



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

Linear Regression

The coefficients for the best-fit line between the **qvarResp** response and **qvarExpl** explanatory variables.

```
(bfl <- lm(qvarResp~qvarExpl,data=dfobj))
```

The coefficient of determination (r^2) value.

```
rSquared(bfl)
```

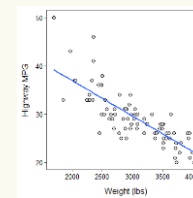
Plot best-fit line by "adding" this to code for a scatterplot.

```
geom_smooth(method="lm",se=FALSE)
```

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

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

```
> ggplot(data=dfcar,mapping=aes(y=HMPG,x=Weight)) +
  geom_point(pch=21,color="black",fill="lightgray") +
  labs(y="Highway MPG",x="Weight (lbs)") +
  theme_NCStats() +
  geom_smooth(method="lm",se=FALSE)
```



1-Sample t-Test

```
t.test(~qvar,data=dfobj,mu=mu0,alt=HA,conf.level=cnfval)
```

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

```
> t.test(~HMPG,data=dfcar,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
```

NOTE: if $n < 40$ then you may need to construct a histogram.

```
> ggplot(data=dfcar,mapping=aes(x=HMPG)) +
  geom_histogram(binwidth=3,boundary=0,
    color="black",fill="lightgray") +
  labs(y="Frequency of Cars",x="Highway MPG") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats()
```

2-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**
- **cvar** is the categorical variable that identifies the two groups
- **HA** is replaced with **"two.sided"** for not equals, **"less"** for less than, or **"greater"** for greater than alternative hypotheses (H_A)
- **cnfval** is the confidence level as a proportion (e.g., 0.95)
- **var.equal=TRUE** if the popn variances are thought to be equal

```
> levenesTest(HMPG~Manual,data=dfcar)
      Df F value Pr(>F)
group 1  7.6663 0.006818
      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
```

NOTE: if $n_1 + n_2 < 40$ then you may need to construct histograms.

```
> ggplot(data=dfcar,mapping=aes(x=HMPG)) +
  geom_histogram(binwidth=3,boundary=0,
    color="black",fill="lightgray") +
  labs(y="Frequency of Cars",x="Highway MPG") +
  scale_y_continuous(expand=expansion(mult=c(0,0.05))) +
  theme_NCStats() +
  facet_wrap(vars(Manual))
```

Chi-square Test

Two-way frequency table with **cvarResp** categorical response variable in columns and **cvarPop** populations as rows.

```
( obtbl <- xtabs(~cvarPop+cvarResp,data=dfobj) )
```

Compute chi-square test results from **obtbl**.

```
( chi <- chisq.test(obtbl,correct=FALSE) )
```

Extract expected values.

```
chi$expected
```

Compute row percentages table (i.e., percentage of individuals in each level of the response variable for each population).

```
percTable(obtbl,margin=1)
```

```
> ( 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)
      Manual
Domestic No Yes Sum
No       13.3  86.7 100.0
Yes      54.2  45.8 100.0
```

NOTE: If data were summarized, then enter frequencies (reading vertically) into a vector with **c()** and then into a table with **matrix()**, making sure to identify the number of rows in **nrow=**.

```
obtbl <- matrix(c(##, ##, ...),nrow=#)
```

Name rows and columns with **rownames()** and **colnames()**.

```
rownames(obtbl) <- c("name","name", ...)
colnames(obtbl) <- c("name","name", ...)
```

Then proceed with **obtbl** as above.

```
> freq2 <- matrix(c(6,26,39,22),nrow=2)
> rownames(freq2) <- c("No","Yes")
> colnames(freq2) <- c("No","Yes")
> freq2
      No Yes
No      6  39
Yes     26  22
```

Goodness-of-Fit Test

One-way frequency table of **cvarResp** categorical response variable

```
( obtbl <- xtabs(~cvarResp,data=dfobj) )
```

Expected proportions (or ratios or values) in **exp.p**.

```
( exp.p <- c(lvl1=#, lvl2=#, lvl3=#,...) )
```

Compute GOF test results from **obtbl** and **exp.p**.

```
( gof <- chisq.test(obtbl,p=exp.p,rescale.p=TRUE, correct=FALSE) )
```

Extract expected values.

```
gof$expected
```

Construct table of observed proportions in each level along with confidence intervals and expected proportions.

```
gofCI(gof,digits=3)
```

```
> ( freq1 <- xtabs(~Type,data=dfcar) )
      Compact Large Midsize Small Sporty Van
      16      11      22      21      14      9

> ( exp <- c(Compact=1, Large=1, Midsize=1,
  Small=1, Sporty=1, Van=1) )
      Compact Large Midsize Small Sporty Van
      1      1      1      1      1      1

> ( gof <- chisq.test(freq1,p=exp,rescale.p=TRUE,
  correct=FALSE) )
Chi-squared test for given probabilities with freq1
X-squared = 8.871, df = 5, p-value = 0.1143

> gof$expected
      Compact Large Midsize Small Sporty Van
      15.5     15.5     15.5     15.5     15.5     15.5

> gofCI(gof,digits=3)
      p.obs p.LCI p.UCI p.exp
Compact 0.172 0.109 0.261 0.167
Large   0.118 0.067 0.199 0.167
Midsize 0.237 0.162 0.332 0.167
Small   0.226 0.153 0.321 0.167
Sporty  0.151 0.092 0.237 0.167
Van     0.097 0.052 0.174 0.167
```

NOTE: If data were summarized, then enter frequencies into a named vector with **c()**.

```
( obtbl <- c(lvl1=#, lvl2=#, lvl3=#,...) )
```

Then proceed with **obtbl** as above.

```
> ( freq1 <- c(Compact=16, Large=11, Midsize=22,
  Small=21, Sporty=14, Van=9) )
      Compact Large Midsize Small Sporty Van
      16      11      22      21      14      9
```