

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/NCMTH207")
> mdat <- read.csv("Mirex.csv")
> str(mdat)
'data.frame':   122 obs. of  4 variables:
 $ year : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
 $ weight : num  0.41 0.45 1.04 1.09 1.24 1.25 1.3 1.34 ...
 $ mirex : num  0.16 0.19 0.19 0.1 0.13 0.19 0.28 0.16 ...
 $ species: Factor w/ 2 levels "chinook","coho": 1 1 1 2 ...
> headtail(mdat,n=2)
   year weight mirex species
1  1977   0.41  0.16  chinook
2  1977   0.45  0.19  chinook
121 1999  11.36  0.09  chinook
122 1999  11.82  0.09  chinook
```

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

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

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).

```
> coho <- filterD(mdat,species=="coho")
> cohoALT <- filterD(mdat,species!="chinook")
> just80s <- filterD(mdat,year>=1980,year<1990)
> cohochin <- filterD(mdat,species %in% c("coho","chinook"))
```

Factors (or Grouping Variables)

Force a `variable` to be considered as a factor with `factor()`

```
dfobj$fvar <- factor(dfobj$fvar)
```

See levels of a `factor variable` with `levels()`

```
levels(dfobj$fvar)
```

Change order of levels with `levels=` in `factor()`

```
dfobj$fvar <- factor(dfobj$fvar,levels=c("lev1","lev2","lev3"))
```

```
> mdat$year <- factor(mdat$year)
> levels(mdat$species)
[1] "chinook" "coho"
> mdat$species<-factor(mdat$species,levels=c("coho","chinook"))
> levels(mdat$species)
[1] "coho" "chinook"
```

Add New Variables to Data.Frame

Add `new variable` to a data.frame with `dfobj$newvar` on the left side of `<=` and an "equation" defining the new variable (which may include `variables` from `dfobj`) to the right of `<=`. Note the following:

- `sqr()` returns the square root.
- `log()` returns the NATURAL log.
- `exp()` return the exponential (anti-natural log).
- `^(x)` raises to the power of `x`.
- `sin()` returns the sine.

```
> mdat$sqrtweight <- sqrt(mdat$weight)
> mdat$logweight <- log(mdat$weight)
> mdat$weight2 <- exp(mdat$logweight)
> mdat$scubrtweight <- mdat$weight^(1/3)
> mdat$sqdweight <- mdat$weight^(2)
> mdat$invweight <- mdat$weight^(-1)
> mdat$sinweight <- sin(mdat$weight)
```

Preferred Global Option

Remove "significance stars" from your results with the code below at the beginning of your script.

```
> options(show.signif.stars=FALSE)
```

1 Fitting (Linear) Models

The five major methods from class can be fit with specific formulae in `lm()` or `glm()`. Save the result to an object.

```
oneway <- lm(qvar~fvar,data=dfobj)
twoway <- lm(qvar~fvar1*fvar2,data=dfobj)
slr <- lm(qvar~qvar,data=dfobj)
ivr <- lm(qvar~qvar*fvar,data=dfobj)
logreg <- glm(fvar~qvar,data=dfobj,family="binomial")
```

Note `qvar` is a quantitative variable and `fvar` is a factor (or grouping) variable. The response variable is to the left of `~`.

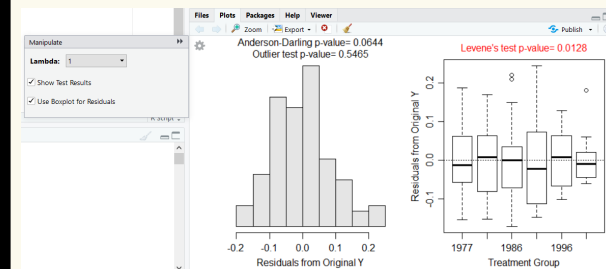
```
> oneway <- lm(mirex~year,data=mdat)
> twoway <- lm(mirex~species*year,data=mdat)
> slr <- lm(mirex~weight,data=mdat)
> ivr <- lm(mirex~weight*species,data=mdat)
> logreg <- glm(species~weight,data=mdat,family="binomial")
```

2 Checking Model Assumptions

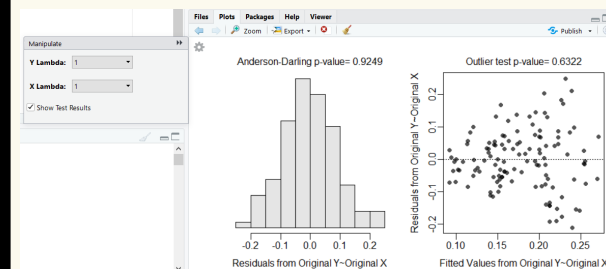
Use `transChooser()` for all plots and hypothesis tests required to check the assumptions of the saved `lm()` model object.

- Select "Show Test Results" in gear box.
- Select "Use Boxplot for Residuals" in gear box for one- and two-way ANOVAs.

> `transChooser(oneway)`



> `transChooser(ivr)`



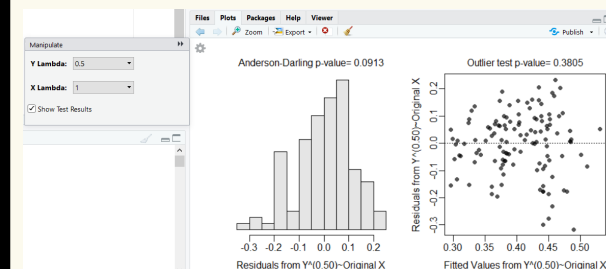
2 Transforming Variables

Use `transChooser()` (see above) to check which variable transformations lead to the model assumptions being met.

- Change "Lambda" for one- and two-way ANOVAs.
- Change "Y" first and then "X" lambda for regressions.

Transform the appropriate variables (see "Add New Variables" box) and then refit the model using the transformed variables.

> `transChooser(ivr)`



```
> mdat$tmirex <- sqrt(mdat$mirex)
> ivr2 <- lm(tmirex~weight*species,data=mdat)
```

3

ANOVA Table

An ANOVA table is constructed from a saved `lm()` or `glm()` object with `anova()`. Note that the p-value is under "Pr(>F)".

```
> anova(twoway)
Response: mirex
      Df Sum Sq Mean Sq F value    Pr(>F)
species 1 0.05040 0.050395   6.5435 0.01189
year     5 0.31221 0.062442   8.1076 1.528e-06
species:year 5 0.02077 0.004154 0.5394 0.74602
Residuals 110 0.84718 0.007702

> anova(slr)
Response: mirex
      Df Sum Sq Mean Sq F value    Pr(>F)
weight 1 0.22298 0.222980  26.556 1.019e-06
Residuals 120 1.00758 0.008396

> anova(ivr)
Response: mirex
      Df Sum Sq Mean Sq F value    Pr(>F)
weight 1 0.22298 0.222980  26.8586 9.155e-07
species 1 0.00050 0.000498  0.0600 0.80690
weight:species 1 0.02744 0.027444  3.3057 0.07158
Residuals 118 0.97964 0.008302
```

4

Coefficients (with CIs) Table

Model coefficients (estimated parameters) and confidence intervals are extracted from a saved `lm()` or `glm()` object with `coef()` and `confint()`, respectively. Column-bind (with `cbind()`) these results together and round to only useful digits (with `round()`) for a concise table of results.

```
> round(cbind(Ests=coef(oweway), confint(oweway)), 3)
      Ests  2.5 % 97.5 %
(Intercept) 0.223 0.186 0.259
year1982    -0.041 -0.093 0.010
year1986    -0.032 -0.084 0.020
year1992     0.014 -0.049 0.077
year1996    -0.061 -0.116 -0.006
year1999    -0.143 -0.198 -0.088

> round(cbind(Ests=coef(ivr), confint(ivr)), 3)
      Ests  2.5 % 97.5 %
(Intercept) 0.085 0.039 0.130
weight      0.022 0.010 0.034
specieschinook 0.051 -0.011 0.113
weight:specieschinook -0.012 -0.025 0.001
```

5

Mult. Comp. (for ANOVAs)

Tukey's or Dunnett's multiple comparison results are extracted from the saved `lm()` object with `glht()` and `mcp()` from the `multComp` package. The second argument to `glht()` is the `mcp()` function with the factor variable in the model set equal to "Tukey" or "Dunnett" depending on which procedure is being used. Save the result of `glht()` to an object.

Use `summary()` and `confint()` on the saved `glht()` object to see p-values and confidence intervals for paired differences in means. Use `glhtSig()` and `cld()` to find the significantly different groups and significance letters (Tukey's only).

Make sure to attach the `multComp` package first with `library()`.

Mult. Comp. (for ANOVAs) cont

```
> library(multcomp) ## MUST DO to use glht() and mcp()
> ## Dunnett's example with one-way ANOVA
> mclw <- glht(oweway, mcp(year="Dunnett"))
> summary(mclw)
Linear Hypotheses:
      Estimate Std. Error t value Pr(>|t|)
1982 - 1977 == 0 -0.04125   0.02610  -1.580   0.391
1986 - 1977 == 0 -0.03208   0.02610  -1.229   0.633
1992 - 1977 == 0  0.01417   0.03197   0.443   0.991
1996 - 1977 == 0 -0.06092   0.02776  -2.194   0.121
1999 - 1977 == 0 -0.14303   0.02776  -5.152  <0.001

> confint(mclw)
Linear Hypotheses:
      Estimate lwr      upr
1982 - 1977 == 0 -0.04125 -0.10810  0.02560
1986 - 1977 == 0 -0.03208 -0.09893  0.03477
1992 - 1977 == 0  0.01417 -0.06771  0.09604
1996 - 1977 == 0 -0.06092 -0.13203  0.01019
1999 - 1977 == 0 -0.14303 -0.21414 -0.07192

> glhtSig(mclw)
[1] "1999 - 1977"

> ## Tukey's example with a two-way ANOVA (and only year var)
> mc2w <- glht(oweway, mcp(year="Tukey"))
> summary(mc2w)
Linear Hypotheses:
      Estimate Std. Error t value Pr(>|t|)
1982 - 1977 == 0 -0.041250   0.026100  -1.580  0.61047
1986 - 1977 == 0 -0.032083   0.026100  -1.229  0.82002
1992 - 1977 == 0  0.014167   0.031965   0.443  0.99776
1996 - 1977 == 0 -0.060921   0.027764  -2.194  0.24655
1999 - 1977 == 0 -0.143026   0.027764  -5.152  < 0.001
1986 - 1982 == 0  0.009167   0.026100  0.351  0.99927
1992 - 1982 == 0  0.055417   0.031965  1.734  0.50991
1996 - 1982 == 0 -0.019671   0.027764  -0.709  0.98035
1999 - 1982 == 0 -0.101776   0.027764  -3.666  0.00485
1992 - 1986 == 0  0.046250   0.031965  1.447  0.69608
1996 - 1986 == 0 -0.028838   0.027764  -1.039  0.90285
1999 - 1986 == 0 -0.110943   0.027764  -3.996  0.00152
1996 - 1992 == 0 -0.075088   0.033338  -2.252  0.22055
1999 - 1992 == 0 -0.157193   0.033338  -4.715  < 0.001
1999 - 1996 == 0 -0.082105   0.029334  -2.799  0.06395

> confint(mc2w) ## CIs not shown to save space

> glhtSig(mc2w)
[1] "1999 - 1977" "1999 - 1982" "1999 - 1986" "1999 - 1992"
```

```
> cld(mc2w)
1977 1982 1986 1992 1996 1999
"b"  "b"  "b"  "b"  "ab" "a"
```

5

Mult. Comp. (for IVR)

Multiple comparisons for slopes are extracted from the saved `lm()` object with `compSlopes()`. If slopes are not different, then multiple comparisons for intercepts are extracted from the saved `lm()` object with `compIntercepts()`.

```
> compSlopes(ivr)
Multiple Slope Comparisons (using the 'holm' adjustment)
      comparison diff 95% LCI 95% UCI p.unadj p.adj
1 chinook-coho -0.01211 -0.0253 0.00108 0.07158 0.07158

Slope Information (using the 'holm' adjustment)
      level slopes 95% LCI 95% UCI p.unadj p.adj
2 chinook 0.00961 0.00389 0.01533 0.00118 0.00118
1 coho 0.02172 0.00983 0.03360 0.00044 0.00088
Warning message: Not needed with fewer than three levels.
```

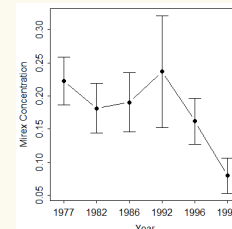
6

Fitted Plots (for ANOVAs)

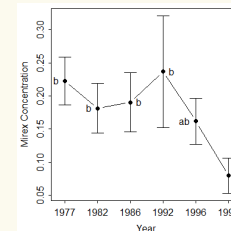
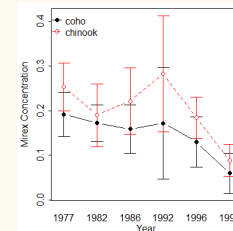
Use `fitPlot()` for a plot of group means with CIs from the saved `lm()` object. For two-way ANOVA use `change.order=TRUE` to (optionally) change which factor is on the x-axis in the interaction plot or `which=` for a main-effects plot.

Use `addSigLetters()` with the `lm()` object to add significance letters in `lets=` to the fitted plot. Use `pos=` to position the letters around the mean (2="left-of" and 4="right-of"). Must use `change.order=TRUE` and `which=` if used in `fitPlot()`.

```
> fitPlot(oweway, xlab="Year", ylab="Mirex Concentration")
> addSigLetters(oweway, lets=c("","","","","","*"),
  pos=c(1,1,1,1,1,4))
```



```
> fitPlot(twoway, xlab="Year", ylab="Mirex Concentration",
  change.order=TRUE, legend="topleft")
> fitPlot(twoway, xlab="Year", ylab="Mirex Concentration",
  which="year") # LEFT
> addSigLetters(twoway, which="year",
  lets=c("b","b","b","b","b","a"),
  pos=c(2,2,4,4,2,4)) # RIGHT
```



6

Fitted Plots (for Regressions)

Use `fitPlot()` to plot the best-fit line (SLR) or lines (IVR). Add CIs or PIs with `interval="confidence"` or `interval="prediction"`.

```
> fitPlot(ivr, xlab="Weight (g)", ylab="Mirex Concentration",
  interval="confidence", legend="topleft")
```

