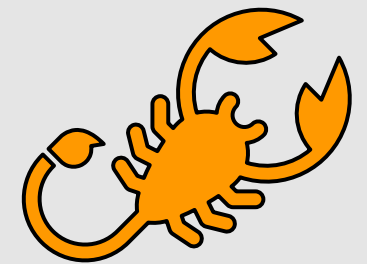


A Survival Guide to Data Analysis

BERDC Special Topics Talk 15



DaCCoTA

DAKOTA COMMUNITY COLLABORATIVE
ON TRANSLATIONAL ACTIVITY

Dr. Mark Williamson
Biostatistics, Epidemiology,
and Research Design Core

Opening

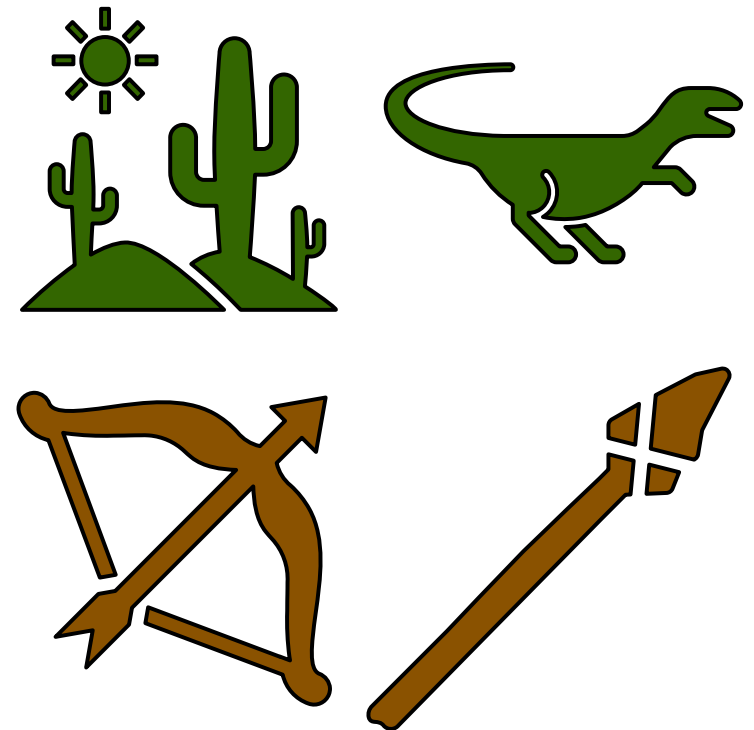
Goal: Deal with Data Analysis when backed against a wall

⚡ Set the stage for dealing with data

⚡ State three classes of problems

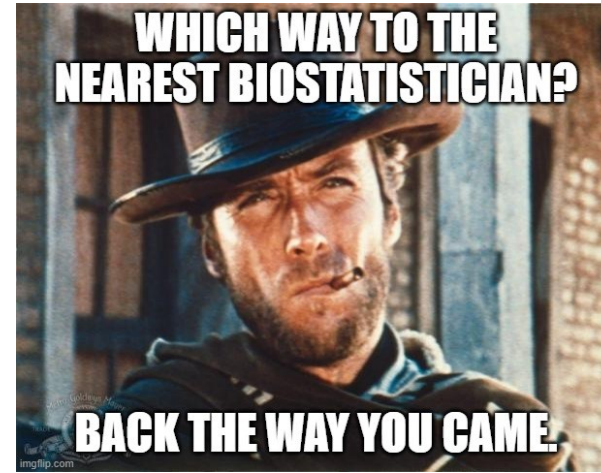
⚡ Discuss three solutions for each problem

⚡ Provide examples for each solution

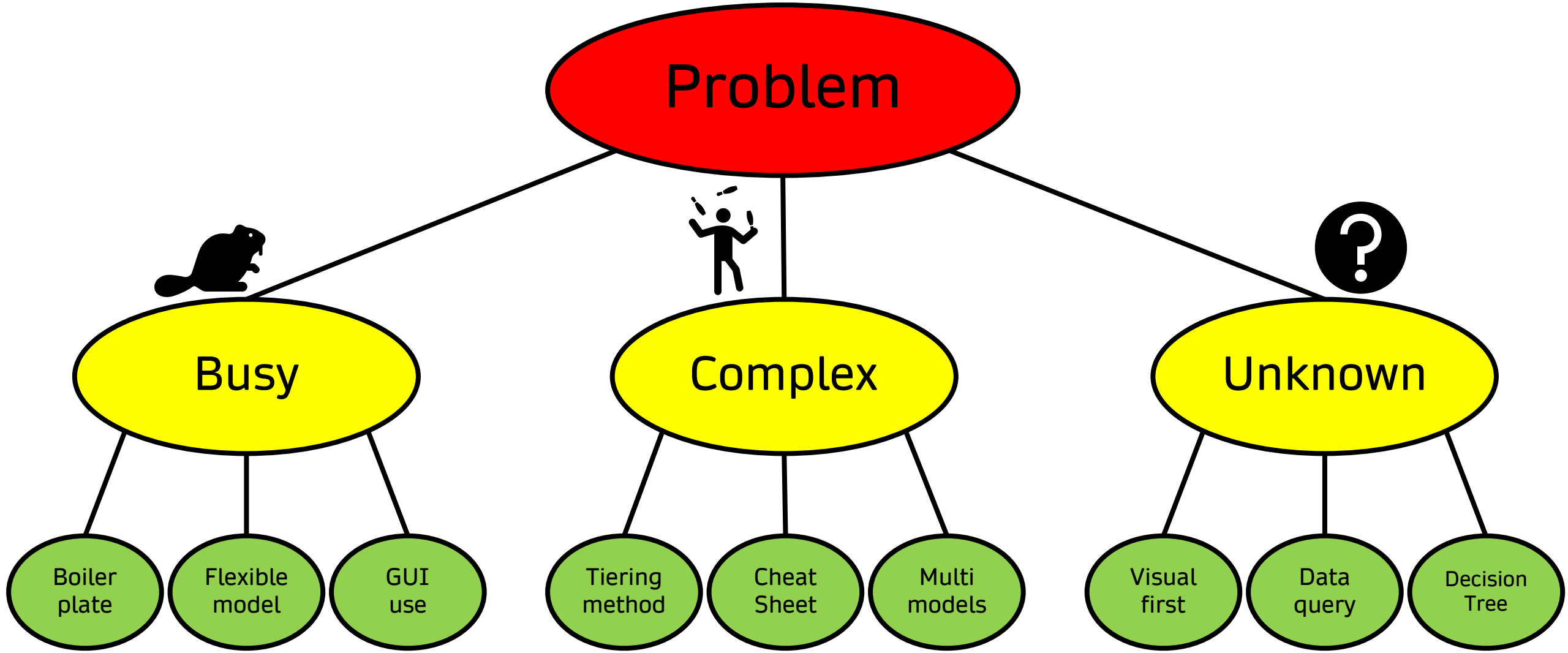


Introduction

- ⚡ Best case scenario for statistical analysis needs is to consult/collaborate with a statistician or statistically-minded researcher
- ⚡ Not always possible (funding, location, educational situation, etc.)
- ⚡ Take these tools with a grain of salt
- ⚡ Steps in this guide are not mutually exclusive



Content





Boilerplate

🔧 Create a standardized document of code that is ready for common procedures and only needs to be filled in with specifics

SAS Coding Tips | 2017

Sample Comparisons

T-tests:

```
proc ttest data=one;
  var Var2; *Numeric variable;
  class Var1; *Categorical groups for comparison;
```

Paired t-test:

```
proc ttest data=one;
  paired Var1 Var2;
```

Multiple Testing:

```
proc glimmix data=one;
  class Var1; * Categorical groups for comparison;
  model Response = Var1;
  lsmeans Var1 / adjust=TUKEY;
  lsmeans Var1 / adjust=BON;
```

Non-parametric:

```
proc npar1way data=one;
  var Var2;
  class Var1;
```

FREQ tables and Chi-square tests:

```
proc freq data = one; BY Experiment;
  weight Count;
  tables Genotype*Disease / chisq;
```

SAS Coding Tips | 2017

Correlation and Regression

Correlation between multiple variables, with matrix scatter plot:

```
proc corr data=one plots=matrix;
  var Var1 Var2 Var3 Var4;
```

Linear Regression:

```
proc reg data=one;
  model depVar = Var2;
```

Multiple Regression with REG or GLM:

```
proc reg data=one;
  model depVar = Var1;
  model depVar = Var1 Var2; *can include multiple models;
  model depVar = Var1 Var2 Var3;
```

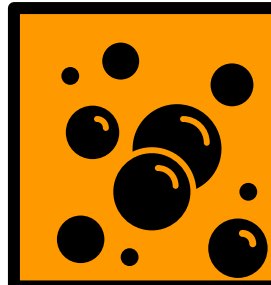
```
proc glm data=one;
  model depVar = Var1|Var2|Var3; *includes all interactions;
```

Polynomial Regression:

```
data one; set one;
  Var2=Var**2;
proc glm data=one;
  model clutch=length|length2;
```

Forward and Stepwise selection with GLMSELECT:

```
proc glmselect data=one plots=(Criteria Candidates);
  model response = var1 var2 var3 var4 var5 /
  selection=forward(select=AICC);
proc glmselect data=one plots=(Criteria Candidates);
  model response = var1 var2 var3 var4 var5 /
  selection=stepwise(select=AICC);
```



Boilerplate

🛠️ Create a standardized document of code that is ready for common procedures and only needs to be filled in with specifics

```
#Simple histogram in ggplot
ggplot(data=DATASET, aes(Y_VAR))+
  geom_histogram(bins=6, fill="blue", col="black")
```

```
#Two-sample histogram in ggplot
ggplot(data=DATASET, aes(x=Y_VAR, fill=X_CAT))+
  geom_histogram(bins=6, col="black", alpha=0.6, position='identity')+
  scale_fill_manual(values=c("red", "blue"))
```

```
#Simple boxplot in ggplot
ggplot(data=DATASET, aes(x=X_CAT, y=Y_VAR)) +
  geom_boxplot()
```

```
#Two-way boxplot in ggplot
ggplot(data=DATASET, aes(x=X_CAT1, y=Y_VAR, fill=X_CAT2)) +
  geom_boxplot()
```

```
#Simple scatter plot in ggplot
ggplot(data=DATASET, aes(x=X_NUM, y=Y_VAR)) +
  geom_point()
```

```
#Two-way scatter plot in ggplot
ggplot(data=DATASET, aes(x=X_NUM, y=Y_VAR, fill=X_CAT, color=X_CAT)) +
  geom_point()
```

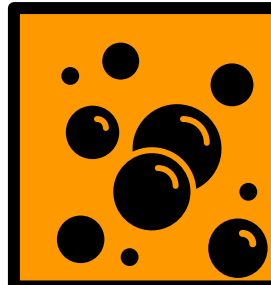
```
#Spaghetti plot in ggplot
ggplot(data=DATASET, aes(x=X_CAT, y=Y_VAR, group=ID)) +
  geom_line()
```

```
#Logistic regression plot in ggplot
ggplot(data=DATASET, aes(x=X_NUM, y=Y_BIN)) +
  geom_point() +
  stat_smooth(method="glm", se=FALSE, method.args=list(family=binomial))
```

```
#Bubble plot in ggplot
ggplot(data=DATASET, aes(x=X_NUM1, y=Y_VAR, size=X_NUM2))+
  geom_point()
```

```
#Simple bar plot in ggplot
DATASET_sum<-DATASET %>%
  group_by(X_CAT) %>%
  summarise(mean=mean(Y_VAR),
            sd = sd(Y_VAR),
            error = qt(0.975,df=n()-1)*sd/sqrt(n()),
            ul = mean + error,
            ll = mean - error)
ggplot(data=DATASET_sum, aes(x=X_CAT, y=mean, fill=X_CAT)) +
  geom_bar(stat="identity") +
  geom_errorbar(aes(ymin=ll, ymax=ul), width=0.1)
```

```
#Two-way bar plot in ggplot
DATASET_sum<-DATASET %>%
  group_by(X_CAT1, X_CAT2) %>%
  summarise(mean=mean(Y_VAR),
            sd = sd(Y_VAR),
            error = qt(0.975,df=n()-1)*sd/sqrt(n()),
            ul = mean + error,
            ll = mean - error)
ggplot(data=DATASET_sum, aes(x=X_CAT1, y=mean)) +
  geom_bar(aes(fill=X_CAT2), stat="identity",
            position=position_dodge()) +
  geom_errorbar(aes(ymin=ll, ymax=ul, group=tension), width=0.1,
            position=position_dodge(0.9))
```



Boilerplate

⚡ Create a standardized document of code that is ready for common procedures and only needs to be filled in with specifics

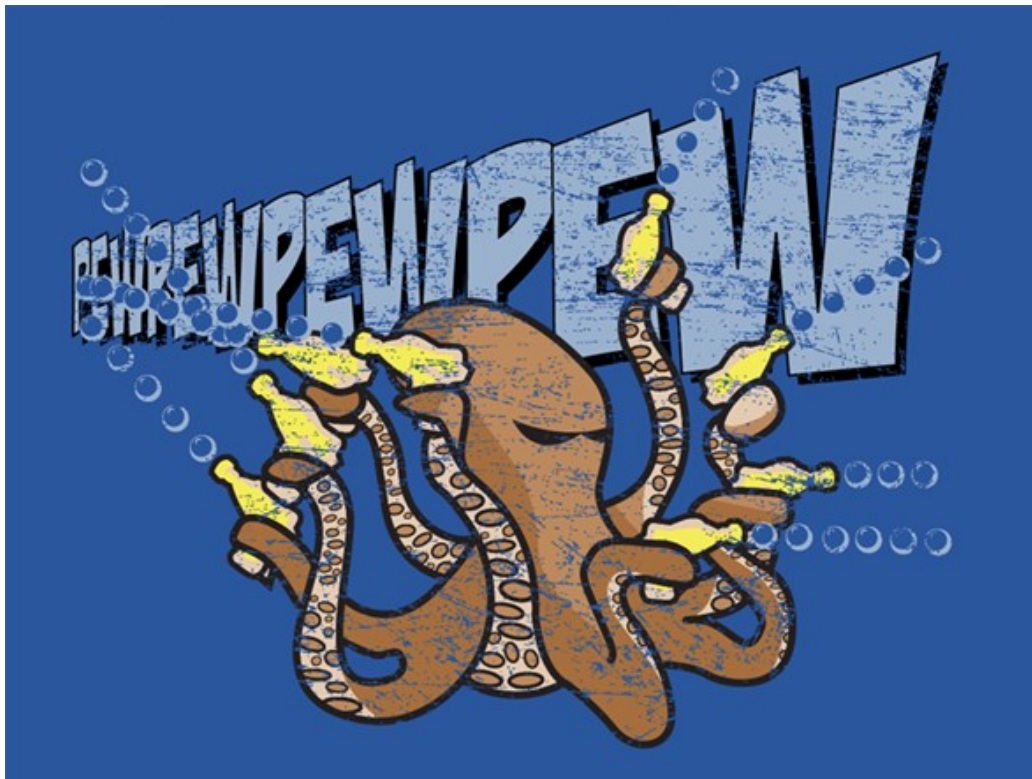
```
barchart.anova <- function(df, cat_var, num_var) {  
  cat_var <- enquo(cat_var)  
  num_var <- enquo(num_var)  
  df_sum <- df %>%  
    group_by(!!cat_var) %>%  
    summarise(mean=mean(!!num_var),  
              sd = sd(!!num_var),  
              error = qt(0.975,df=n()-1)*sd/sqrt(n()),  
              ul = mean + error,  
              ll = mean - error)  
  #print(df_sum)  
  df_plot <- ggplot(df_sum, aes(!!cat_var, mean, fill=!!cat_var))+  
    geom_bar(stat="identity", color="black")+  
    geom_hline(yintercept = 0) +  
    geom_errorbar(aes(ymin=ll, ymax=ul), width=0.1) +  
    theme_classic() + theme(legend.position="none")  
}
```

```
#Example  
barchart.anova(iris, Species, Sepal.Length)  
#basic  
df_plot  
#customized  
df_plot + labs(y="Sepal length (cm)", x="Species") +  
  scale_y_continuous(limits=c(0,7),breaks=c(0:7))+  
  geom_text(aes(label=c('*', '*', '*')), vjust=-0.8, size=6)
```



Flexible Models

- ✂ Use flexible models that can run multiple tests just by tweaking parameters, rather than having to run different code for each and every test



```
#FLEXIBLE MODELS
```

```
head(mtcars)
```

```
t.model <-lm(mpg~as.factor(vs), data=mtcars)
```

```
summary(t.model)
```

```
a.model <-lm(mpg~as.factor(carb), data=mtcars)
```

```
summary(a.model)
```

```
r.model <-lm(mpg~wt, data=mtcars)
```

```
summary(r.model)
```




Flexible Models

🔧 Use flexible models that can run multiple tests just by tweaking parameters, rather than having to run different code for each and every test

```
PROC GLIMMIX data=sashelp.Class;  
  class Sex;  
  model Height=Sex /solution dist=normal;  
  lsmeans Sex /cl;  
  ods output LSmeans=Class_lsm;  
PROC SGPLOT data=Class_lsm;  
  vbarparm category=Sex response=Estimate/limitupper=Upper  
  limitlower=Lower;
```

T-test

```
PROC GLIMMIX data=sashelp.bmimen;  
  model BMI=age/solution dist=normal;  
  output out=Bmimen_pred pred lcl ucl;  
PROC SGPLOT data=Bmimen_pred;  
  band x=age lower=lcl upper=ucl;  
  scatter x=age y=BMI;  
  series x=age y=Pred;
```

Regression

```
PROC GLIMMIX data=sashelp.bweight;  
  class MomEdLevel;  
  model Weight=MomEdLevel/solution dist=normal;  
  lsmeans MomEdLevel / cl;  
  ods output LSMeans=Bweight_lsm;  
PROC SGPLOT data=Bweight_lsm;  
  vbarparm category=MomEdLevel  
  response=Estimate/  
  limitupper=Upper limitlower=Lower;
```

ANOVA

```
PROC GLIMMIX data=multicenter;  
  class center group;  
  model SideEffect/n = group / solution;  
  random center;  
  lsmeans group / ilink cl;  
  ods output LSMeans=lsm1;  
PROC SGPLOT data=lsm1;  
  vbarparm category=group  
  response=Mu /  
  limitlower=LowerMu  
  limitupper=UpperMu;
```

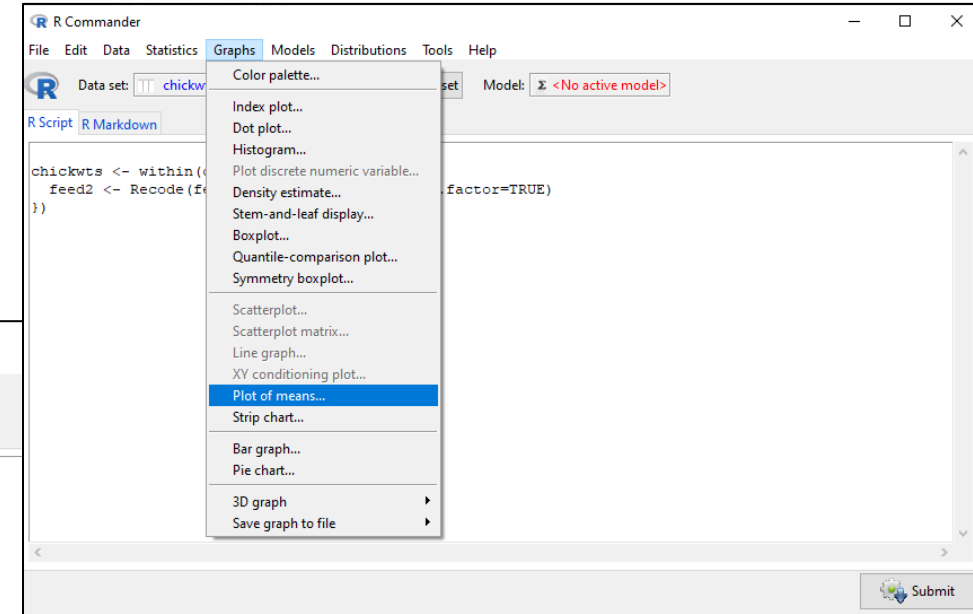
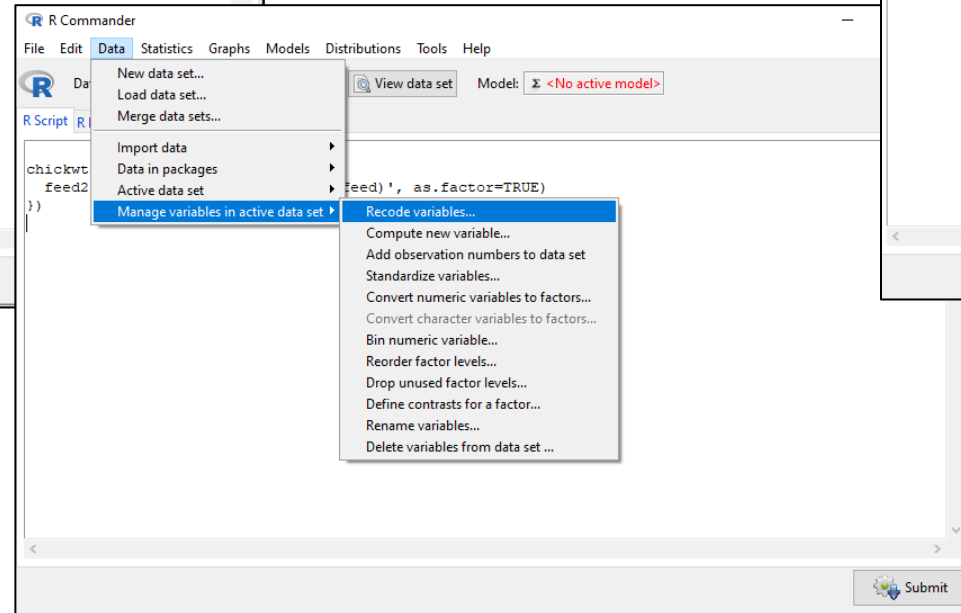
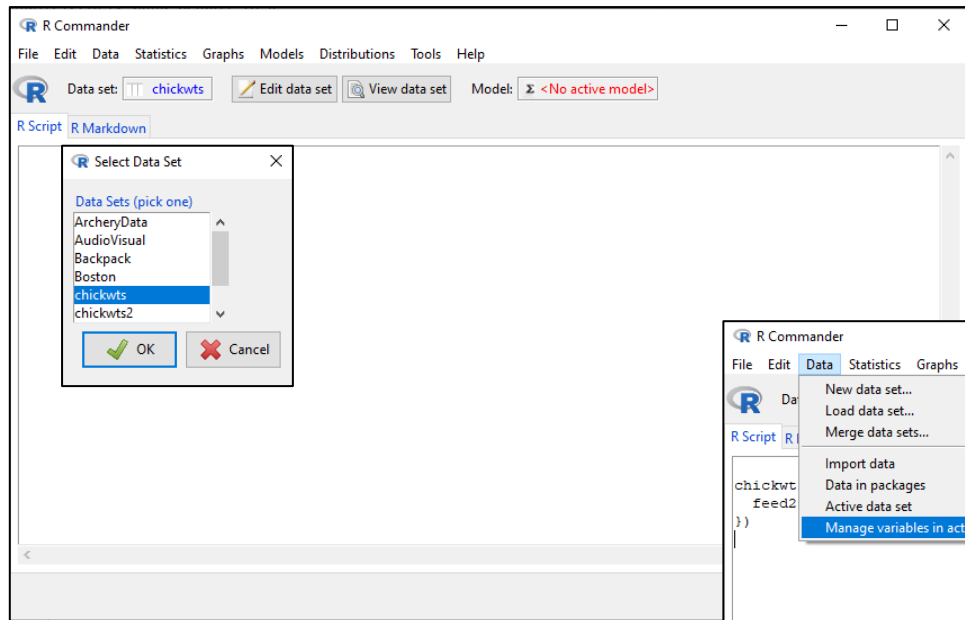
*Generalized
Linear Mixed
Model*

GUI Use



GUI Use

🔨 Use a Graphical Use Interface to rapidly select and run tests

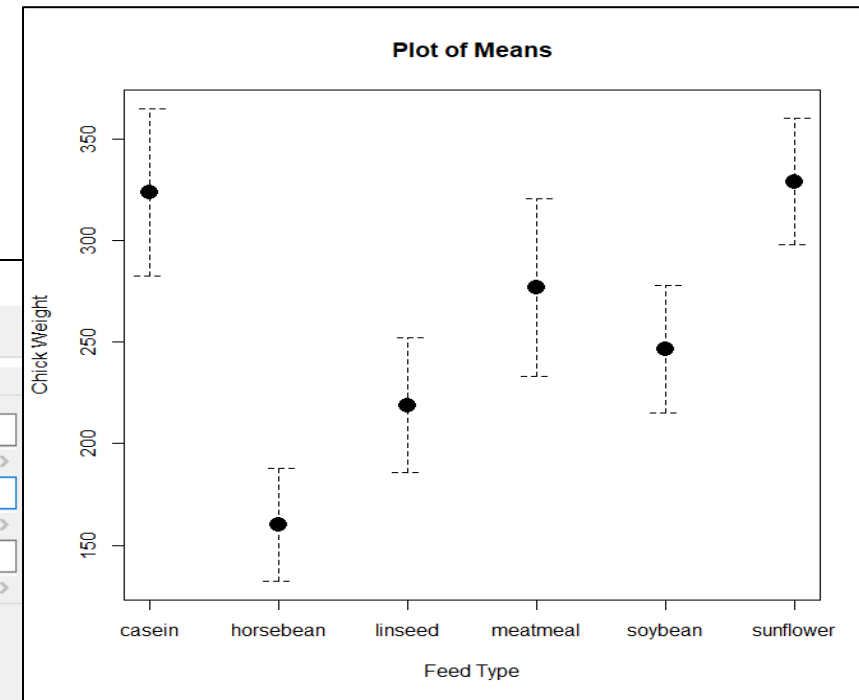
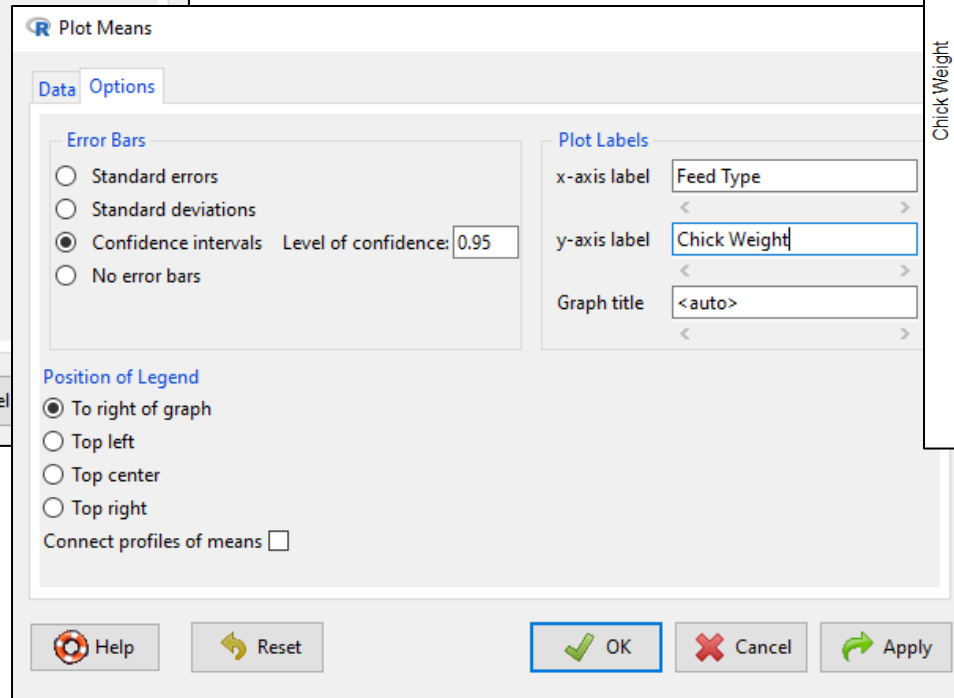
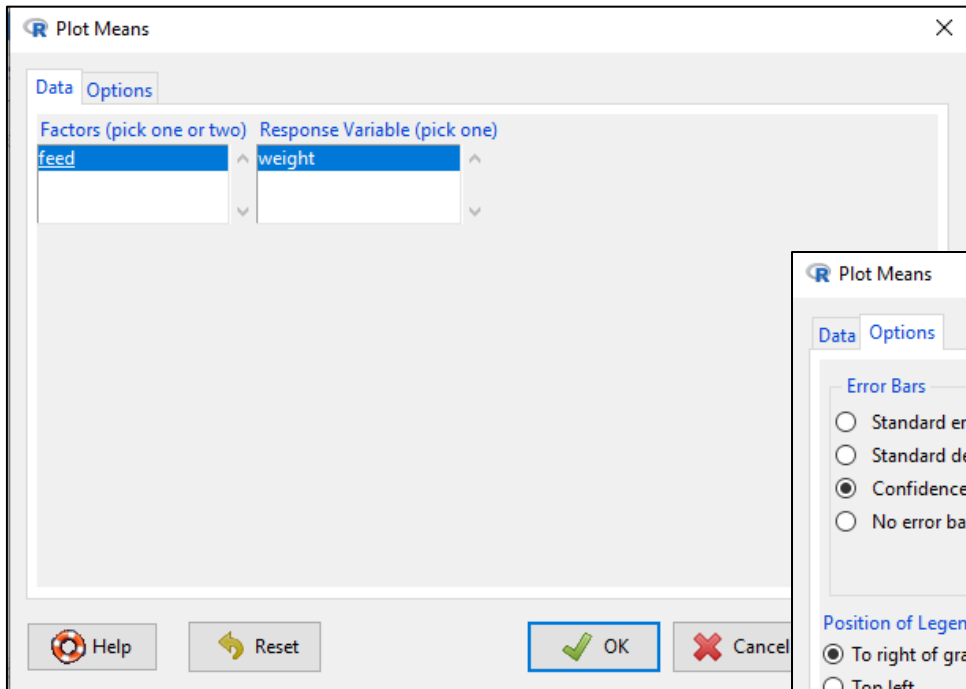


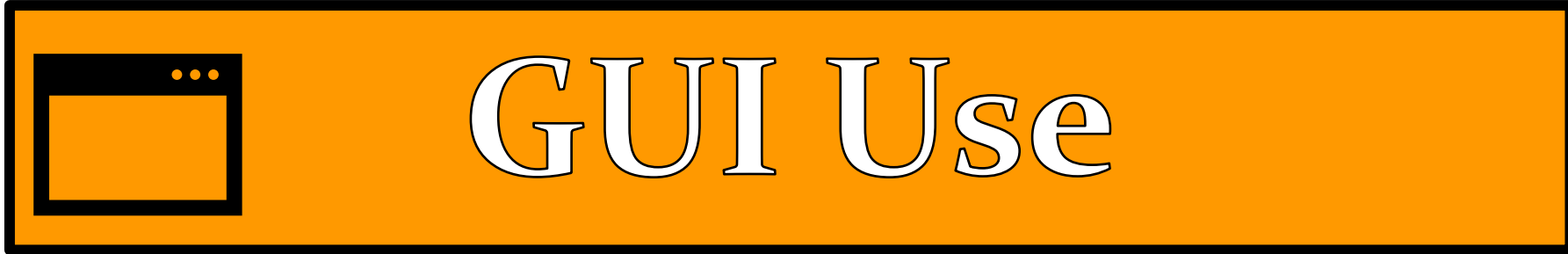
GUI Use



GUI Use

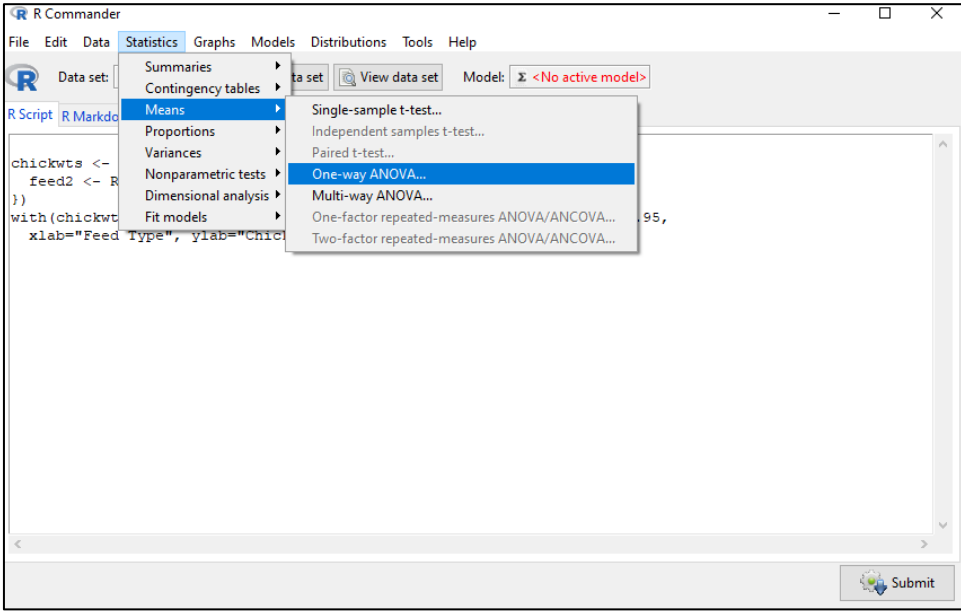
Use a Graphical User Interface to rapidly select and run tests





GUI Use

🔧 Use a Graphical Use Interface to rapidly select and run tests



```
Rcmdr> summary(AnovaModel.1)
      Df Sum Sq Mean Sq F value Pr(>F)
feed    5 231129  46226 15.37 5.94e-10 ***
Residuals 65 195556   3009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Rcmdr> with(chickwts, numSummary(weight, groups=feed, statistics=c("mean", "sd")))
      mean    sd data:n
casein 323.5833 64.43384   12
horsebean 160.2000 38.62584   10
linseed 218.7500 52.23570   12
meatmeal 276.9091 64.90062   11
soybean 246.4286 54.12907   14
sunflower 328.9167 48.83638   12
```

GUI Use



GUI Use

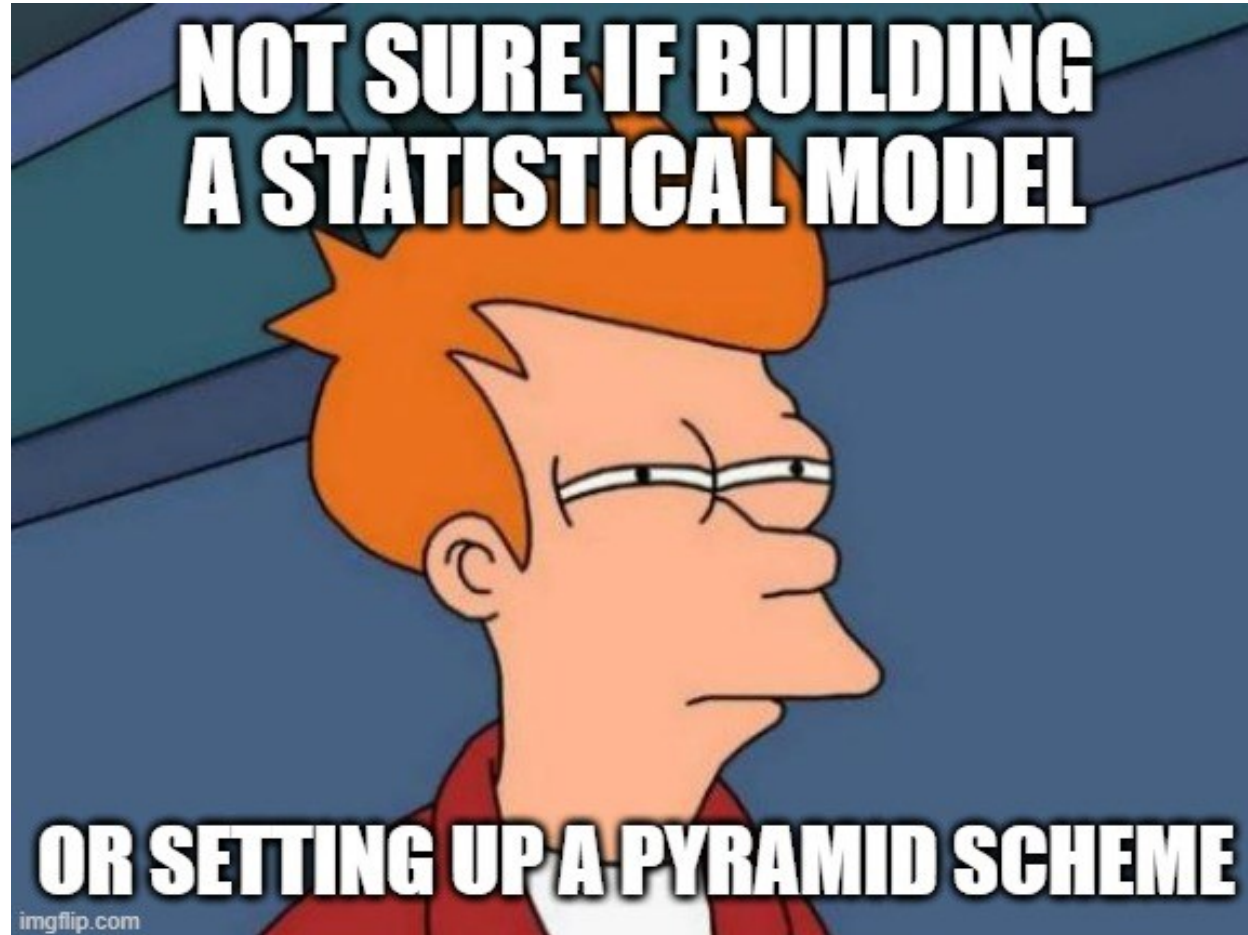
🔧 Use a Graphical Use Interface to rapidly select and run tests

The screenshot displays the IBM SPSS Statistics Data Editor interface. The main window shows a data table with 20 rows and 12 columns. The columns are: age, ed, employ, address, income, debtinc, creddebt, othdebt, default, preddef1, preddef2, and predc. The data table is currently in Data View. A dialog box titled "Logistic Regression" is open in the foreground. The "Dependent" variable is "Previously defaulted [default]". The "Covariates" list includes "age", "ed", "employ", "address", and "income". The "Method" is set to "Enter". The "Selection Variable" is empty. The dialog box has buttons for "OK", "Paste", "Reset", "Cancel", and "Help".

	age	ed	employ	address	income	debtinc	creddebt	othdebt	default	preddef1	preddef2	predc
1	41	Some college	17	12	176.00	9.30	11.36	5.01	Yes	.80839	.78864	
2	27	Did not complete high school								.19830	.12845	
3	40	Did not complete high school								.01004	.00299	
4	41	Did not complete high school								.02214	.01027	
5	24	High school degree								.78159	.73788	
6	41	High school degree								.21671	.32819	
7	39	Did not complete high school								.18596	.17926	
8	43	Did not complete high school								.01471	.01057	
9	24	Did not complete high school								.74804	.61944	
10	36	Did not complete high school								.81506	.79723	
11	27	Did not complete high school								.35031	.61051	
12	25	Did not complete high school								.23905	.21902	
13	52	Did not complete high school								.00979	.00628	
14	37	Did not complete high school								.36449	.34047	
15	48	Did not complete high school								.01187	.00771	
16	36	High school degree								.09670	.11384	
17	36	High school degree								.21205	.17502	
18	43	Did not complete high school								.00140	.00056	
19	39	Did not complete high school								.10415	.09273	
20	44	Some college	0	21	96.00	1.70	10	24	No	.00102	.08601	

Tiering Method

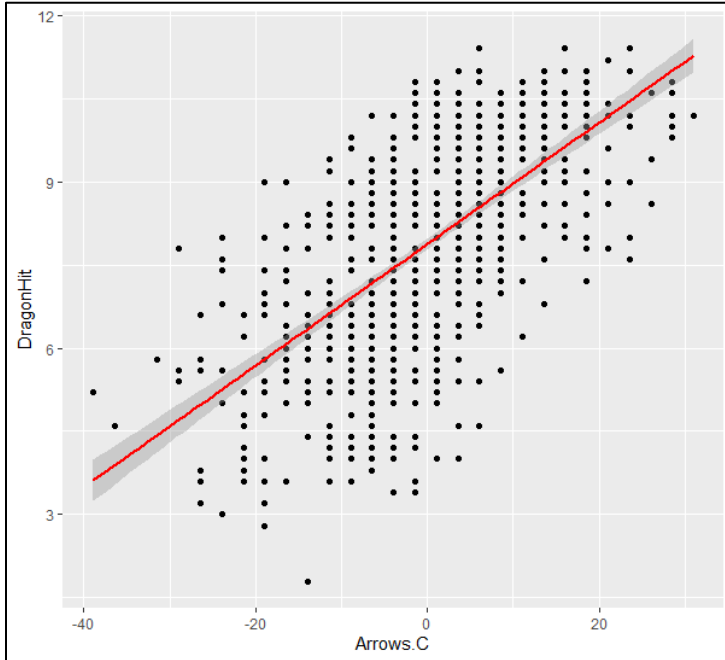
⚡ Start with simplest model design and progressively build from there



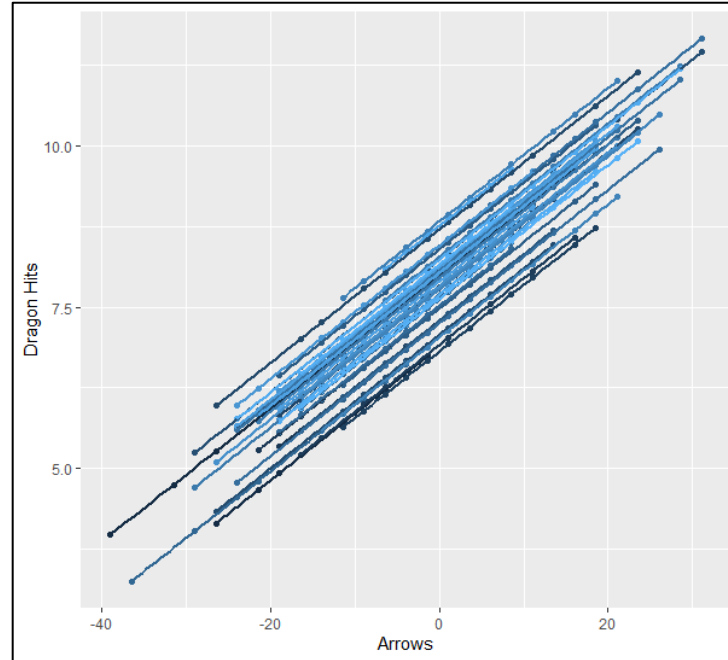
Tiering Method

🛠 Start with simplest model design and progressively build from there

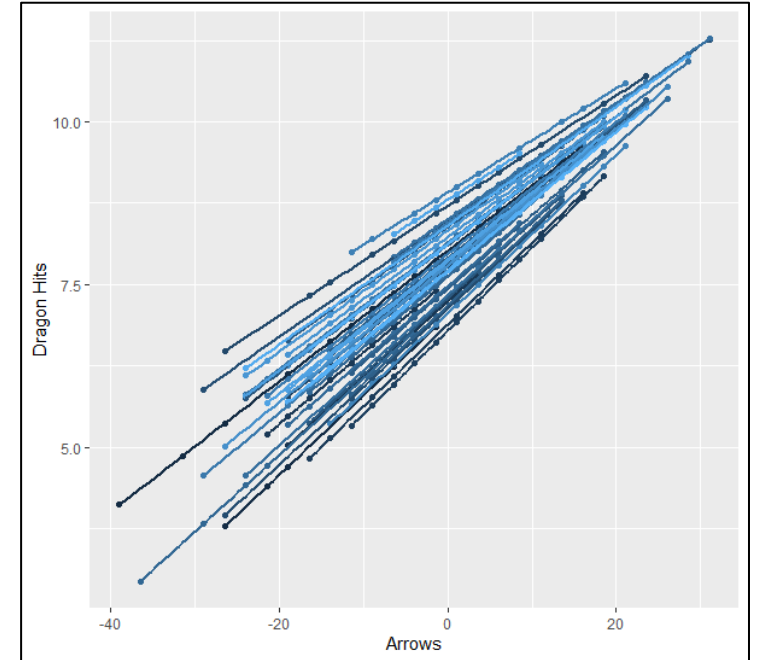
```
D3_M1 <- lm(DragonHit ~ Arrows.C,  
data=Dragon2)
```



```
D3_M2 <- lmer(DragonHit ~ Arrows.C  
+ (1 | Cathedral),  
data=Dragon2, REML=F)
```



```
D3_M3 <- lmer(DragonHit ~ Arrows.C  
+ (Arrows.C | Cathedral),  
data=Dragon2, REML=F)
```



Cheat Sheet

Deploy sheets with quickly available information for statistical methods, especially more advanced concepts

Machine Learning Modelling in R : : CHEAT SHEET

Supervised & Unsupervised Learning				Meta-Algorithm, Time Series & Model Validation			
ALGORITHM	DESCRIPTION	R PACKAGE/FUNCTION	SAMPLE CODE	ALGORITHM	DESCRIPTION	R PACKAGE/FUNCTION	SAMPLE CODE
Naive Bayes Classifier	A classification technique based on Bayes' Theorem with an assumption of independence among predictors. In both cases, the naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.	e1071::naiveBayes	naiveBayes(Class ~., data = x)	Regularization	Regularization adds a penalty on the different parameters of a model to reduce the freedom of the model. Hence, the model will be less likely to overfit the training data and will improve the generalization ability of the model.	glmnet::glmnet	library(glmnet) data <- read.csv("data.csv") fit <- glmnet(x = data[,1:10], y = data[,11]) plot(fit)
k-Nearest Neighbors	A non-parametric method used for classification and regression. In both cases, the neighbor consists of the closest training samples in the feature space. The output depends on whether a k-NN is used for classification or regression.	class::knn	knn(train, test, cl, k = 1 + 0.5 * sqrt(nrow(test)))	Boosting	Boosting is a process of iteratively adding weak classifiers to form a strong classifier. It is used to improve the predictive ability.	randomForest::randomForest	randomForest(x = data[,1:10], y = data[,11])
Linear Regression	Model the linear relationship between a scalar dependent variable and one or more independent variables for independent variables identified.	stats::lm	lm(fit ~ species, data = x)	Ensemble	Ensemble learning is a machine learning paradigm that combines the predictions of multiple individual models to improve the overall performance. It is used to reduce the variance and bias of the model.	randomForest::randomForest	randomForest(x = data[,1:10], y = data[,11])
Logistic Regression	Used to predict a binary outcome (1/0, Yes/No, True/False) given a set of independent variables.	stats::glm	glm(fit ~ family = "binomial", data = x)	Pruning	Pruning is a technique that reduces the size of decision tree by removing sections of the tree that provide little gain to the overall model. It is used to reduce the overfitting.	rpart::rpart	rpart(x = data[,1:10], y = data[,11])
Tree-Based Models	The idea is to recursively divide (split) the training dataset based on the feature that most effectively separates the data into two groups. The process is repeated until the target variable is as close as possible to a constant.	rpart::rpart	rpart(x = data[,1:10], y = data[,11])	Random Forest	An ensemble learning method for classification, regression and other tasks, that operates by constructing multiple decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression).	randomForest::randomForest	randomForest(x = data[,1:10], y = data[,11])
Artificial Neural Network	Neural networks are built from units called neurons. Neurons are arranged in layers, and each neuron is connected to other neurons in the next layer. The output of a neuron is the weighted sum of its inputs, plus a bias, and then passed through an activation function.	neuralnet::neuralnet	neuralnet(x = data[,1:10], y = data[,11])	Lead-Lag Analysis	Lead-lag analysis is a technique used to analyze time series data. It involves creating lead and lag variables from the original data to capture temporal dependencies.	stats::arima	arima(x = data[,1], order = c(1,1,1))
Support Vector Machine	A data classification method that separates data using hyperplanes.	e1071::svm	svm(x = data[,1:10], y = data[,11])	Performance Metrics	Performance metrics are used to evaluate the performance of a model. They include accuracy, precision, recall, and F1 score.	caret::confusionMatrix	confusionMatrix(fit, data[,1:10], data[,11])
Principal Component Analysis	A procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.	stats::prcomp	prcomp(x = data[,1:10])	Learning Curves	Learning curves are used to monitor the performance of a model during training. They show the training and test error rates over time.	caret::learningCurve	learningCurve(fit, data[,1:10], data[,11])
k-Mean Clustering	Aims at partitioning observations into clusters which each observation belongs to the cluster with the nearest mean.	stats::kmeans	kmeans(x = data[,1:10], centers = 5, nstart = 1, algorithm = "Lloyd")				
Hierarchical Clustering	An approach which builds a hierarchy from the bottom up and doesn't require the number of clusters to be specified beforehand.	stats::hclust	hclust(x = data[,1:10], method = "complete", members = NA)				

Standard Modelling Workflow

TRAINING

- Simple Model
- Meta Model

VALIDATION

- Simple Model
- Meta Model

PRODUCTION

- Simple Model
- Meta Model

Time Series View

TRAINING

- A simple model is trained using a first sub-sample of the original data set.
- A meta model is evaluated using a first sub-sample of the original data set.
- Simple models can be aggregated to define a meta model.

VALIDATION

- Simple models are validated using a second sub-sample of the original data set.
- Meta models are validated using a second sub-sample of the original data set.

PRODUCTION

- Validated models are tested in production.
- Depending on results, simple and meta models might go through the training and validation steps again until production results are acceptable.

Creating Survival Plots Informative and Elegant with survminer

Survival Curves

The `ggsurvplot()` function creates `ggplot2` plots from `survfit` objects.

```
library(survminer)
fit <- survfit(Surv(time, status) ~ sex, data = lung)
ggsurvplot(fit, data = lung)
```

Use the `fun` argument to set the transformation of the survival curve. E.g. "event" for cumulative events, "cumhaz" for the cumulative hazard function or "pct" for survival probability in percentage.

```
ggsurvplot(fit, data = lung, fun = "event")
ggsurvplot(fit, data = lung, fun = "cumhaz")
```

With lots of graphical parameters you have full control over look and feel of the survival plots; position and content of the legend; additional annotations like p-value, title, subtitle.

```
ggsurvplot(fit, data = lung,
  conf.int = TRUE,
  pval = TRUE,
  fun = "pct",
  risk.scale = TRUE,
  size = 1,
  linetype = "step",
  palette = c("red", "blue", "green", "yellow", "purple", "orange", "brown", "pink", "grey", "black", "white"),
  legend = "bottom",
  legend.title = "Sex",
  legend.labs = c("Male", "Female"))
```

Diagnostics of Cox Model

The function `cox.zph()` from `survival` package may be used to test the proportional hazards assumption for a Cox regression model fit. The graphical verification of this assumption may be performed with the function `ggcoxzph()` from the `survminer` package. For each covariate it produces plots with scaled Schoenfeld residuals against the time.

```
library(survival)
fit <- coxph(Surv(time, status) ~ sex + age, data = lung)
fit <- coxph(Surv(time, status) ~ sex + ph.ecog + age, data = lung)
fit
```

The function `ggcoxdiagnostics()` plots different types of residuals as a function of time: linear predictor or observation id. The type of residual is selected with `type` argument. Possible values are "martingale", "deviance", "score", "schoenfeld", "dibetas", and "scaledsch". The `ox.scale` argument defines what shall be plotted on the Ox axis. Possible values are "linear.predictions", "observation.id", "time". Logical arguments `hline` and `sline` may be used to add horizontal line or smooth line to the plot.

```
ggcoxdiagnostics(fit,
  type = "deviance",
  ox.scale = "linear.predictions")
ggcoxdiagnostics(fit,
  type = "deviance",
  ox.scale = "time")
```

Summary of Cox Model

The function `ggforest()` from the `survminer` package creates a forest plot for a Cox regression model fit. Hazard ratio estimates along with confidence intervals and p-values are plotted for each variable.

```
library(survival)
library(survminer)
lung$age <- ifelse(lung$age > 70, ">70", "<= 70")
fit <- coxph(Surv(time, status) ~ sex + ph.ecog + age, data = lung)
fit
```

```
## G11:
## coxph(formula = Surv(time, status) ~ sex+ph.ecog+age, data=lung)
## sex      coef exp(coef) se(coef)      z      p
## sex    -0.567    0.567    0.168   -3.37 0.00075
## ph.ecog  0.478    1.600    0.113   4.16 3.1e-05
## age>70  0.287    1.339    0.137   2.04 0.04173
## Likelihood ratio test=21.6 on 2 df, p=0.0001
## #= 227, number of events=164
```

```
ggforest(fit)
```

The function `ggadjustedcurves()` from the `survminer` package plots Adjusted Survival Curves for Cox Proportional Hazards Model. Adjusted Survival Curves show how a selected factor influences survival estimated from a Cox model.

Note that these curves differ from Kaplan Meier estimates since they present expected survival based on given Cox model.

```
lung$sex <- ifelse(lung$sex == 1, "Male", "Female")
library(survival)
library(survminer)
fit <- coxph(Surv(time, status) ~ sex + ph.ecog + age + strata(sex), data = lung)
ggadjustedcurves(fit, data=lung)
```

Note that it is not necessary to include the grouping factor in the Cox model. Survival curves are estimated from Cox model for each group defined by the factor independently.

```
lung$age3 <- cut(lung$age,
  c(35, 55, 65, 85))
ggadjustedcurves(fit, data=lung,
  variable="age3")
```




Cheat Sheet

🛠️ Deploy sheets with quickly available information for statistical methods, especially more advanced concepts

Examples	STATA	SPSS	Excel	SAS	R
Summary statistics	Data->Describe Data-> Summary Statistics OR summarize <i>num_var</i>	Analyze -> Descriptive Statistics -> Descriptives	=AVERAGE(<i>num_var</i>) =MEDIAN(<i>num_var</i>) =STDEV.S(<i>num_var</i>) ...	PROC UNIVARIATE; var <i>num_var</i> ;	summary(<i>num_var</i>)
Histogram	Graphics-> Histogram OR histogram <i>num_var</i>	Graphs -> Chart Builder -> Histogram	Insert (Charts)-> Histogram	PROC SGPLOT; histogram <i>num_var</i> ;	hist(<i>num_var</i>)
Boxplot	Graphics-> Box plot OR graph box <i>num_var</i> , over(<i>cat_var</i>)	Graphs -> Chart Builder -> Boxplot	Insert (Charts)-> Box and Whisker	PROC SGPLOT; vbox <i>num_var</i> / group= <i>cat_var</i> ;	plot(<i>num_var</i> ~ <i>cat_var</i>)
Bar plot	Graphics-> Bar Chart OR graph bar (mean) <i>num_var</i> , over(<i>cat_var</i>)	Graphs -> Chart Builder -> Bar	Insert (Charts)-> Column	PROC SGPLOT; vbarparm category= <i>cat_var</i> treatment= <i>num_mean</i> ;	<i>means</i> <- c(<i>mean_cat1</i> , <i>mean_cat2</i>) barplot(<i>means</i>)
Scatterplot	Graphics-> Tway graph OR tway (scatter <i>num_var1 num_var2</i>)	Graphs -> Chart Builder -> Scatter/Dot	Insert (Charts)-> Scatter	PROC SGPLOT; Scatter y= <i>num_var1</i> x= <i>num_var2</i> ;	plot(<i>num_var1</i> , <i>num_var2</i>)
T-test	Statistics -> Summaries, tables, and tests -> Classical tests of hypotheses -> t tests OR ttest <i>num_var</i> , by(<i>cat_var</i>)	Analyze -> Compare means-> Independent-Samples T Test	=TTEST(<i>num_var1</i> , <i>num_var2</i> , <i>tails</i> , <i>type</i>)	PROC TTEST; var <i>num_var</i> ; class <i>cat_var</i> ;	t.test(<i>num_var</i> ~ <i>cat_var</i>)
ANOVA	Statistics-> Linear models and related -> ANOVA/MANOVA -> One-way ANOVA OR oneway <i>num_var cat_var</i>	Analyze -> Compare means-> One-Way ANOVA	Data Analysis (add-on) -> Anova: Single Factor	PROC ANOVA; class <i>cat_var</i> ; model <i>num_var</i> = <i>cat_var</i> ;	aov(<i>num_var</i> ~ <i>cat_var</i>)
Normal linear regression model	Statistics-> Linear models and related -> Linear regression OR regress <i>num_var1 num_var2</i>	Analyze -> Regression-> Linear	Data Analysis (add-on) -> Regression	PROC REG; model <i>num_var1</i> = <i>num_var2</i> ;	lm(<i>num_var1</i> ~ <i>num_var2</i>)
Logistic regression model	Statistics-> Binary outcomes-> Logistic regression OR logit <i>binary_var num_var</i>	Analyze -> Regression-> Binary Logistic	N/A	PROC LOGISTIC; model <i>event/trial</i> = <i>num_var2</i> ;	glm(<i>binary_var</i> ~ <i>num_var</i> , family=biniomial)
Poisson regression model	Statistics -> Count outcomes-> Poisson regression OR Poisson <i>count_var num_var</i>	Analyze -> Regression-> Generalized Linear Models	N/A	PROC GLIMMIX; model <i>count_var</i> = <i>num_var</i> /dist=Poisson;	glm(<i>count_var</i> ~ <i>num_var</i> , family=Poisson)
Generalized linear mixed model	Statistics -> Multilevel mixed-effects models -> Generalized linear model OR meglm <i>var1 var2</i> <i>rand_var_eqn</i> , family(<i>distribution</i>) link(<i>link_function</i>)	Analyze-> Mixed Models-> Generalized Linear	N/A	PROC GLIMMIX; class <i>cat_var</i> ; model <i>num_var1</i> = <i>num_var2</i> <i>cat_var rand_var</i> ; random <i>rand_var</i> ;	Package lme4 Lmer(<i>num_var1</i> ~ <i>num_var2</i> + <i>cat_var</i> + (1 <i>rand_var</i>))

Multi Models

⚡ Be ready to run multiple models that could fit the data and compare best fit

```
btb_lmer1 <- lmer(bdi ~ bdi.pre + time + treatment + drug + length + (1 | subject),  
  data = BtheB_long, REML=FALSE, na.action = na.omit) #Rand Int  
  
btb_lmer2 <- lmer(bdi ~ bdi.pre + time + treatment + drug + length + (time | subject),  
  data = BtheB_long, REML=FALSE, na.action = na.omit) #Rand Int & Slope  
  
btb_gee1 <- gee(bdi ~ bdi.pre + treatment + length + drug,  
  data = BtheB_long, id = subject, family = gaussian, corstr = "independence")  
  
btb_gee2 <- gee(bdi ~ bdi.pre + treatment + length + drug,  
  data = BtheB_long, id = subject, family = gaussian, corstr = "exchangeable")  
  
anova(btb_lmer1, btb_lmer2, btb_gee1, btb_gee2)
```

```
PROC GLIMMIX data=DATASET method=RSPL;  
  class RAND1;  
  model Y_VAR=X_VAR1 | X_VAR2;  
  random RAND1;
```



Multi Models

⚡ Be ready to run multiple models that could fit the data and compare best fit

```
reg1 <- lm(Dep ~ Ind1 + Ind2 + Ind3 + Ind4)
sreg1 <- step(reg1, direction="forward")
sreg2 <- step(reg1, direction="backward")
sreg3 <- step(reg1, direction="both")
```

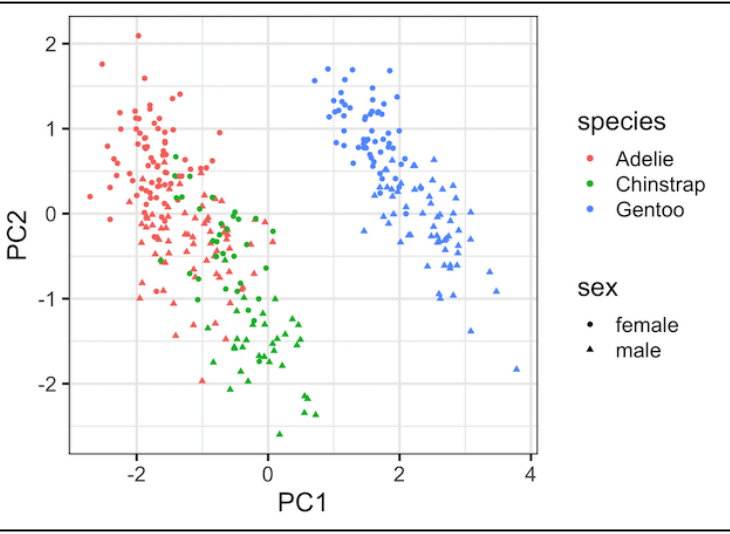
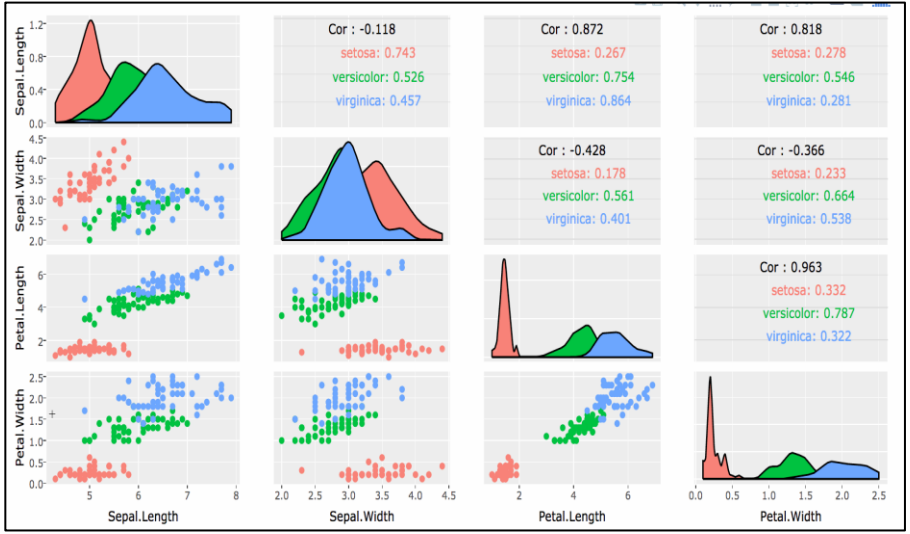
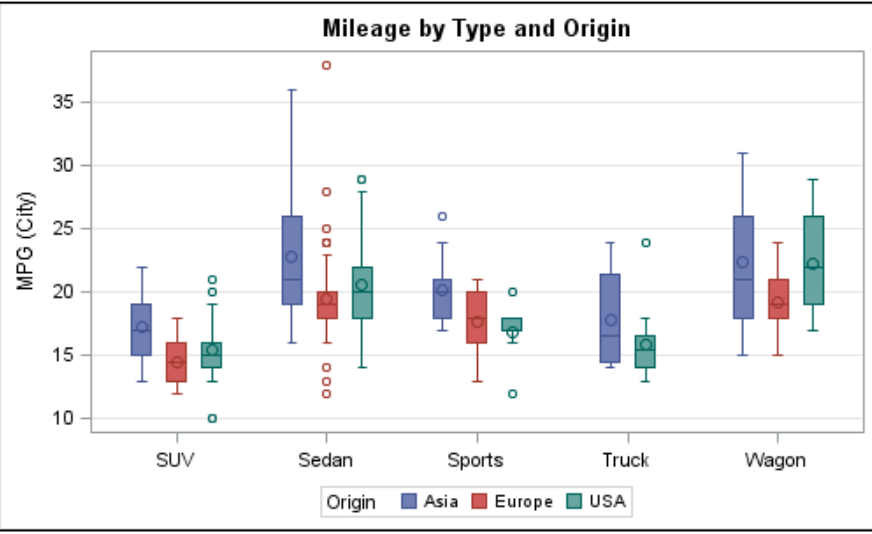
```
PROC GLMSELECT data=sashelp.baseball
plots=all;
class league division;
model logSalary = nAtBat nHits nHome nRuns
nRBI nBB yrMajor crAtBat
crHits crHome crRuns crRbi
crBB league division nOuts
nAssts nError
/ details=all stats=all;
```





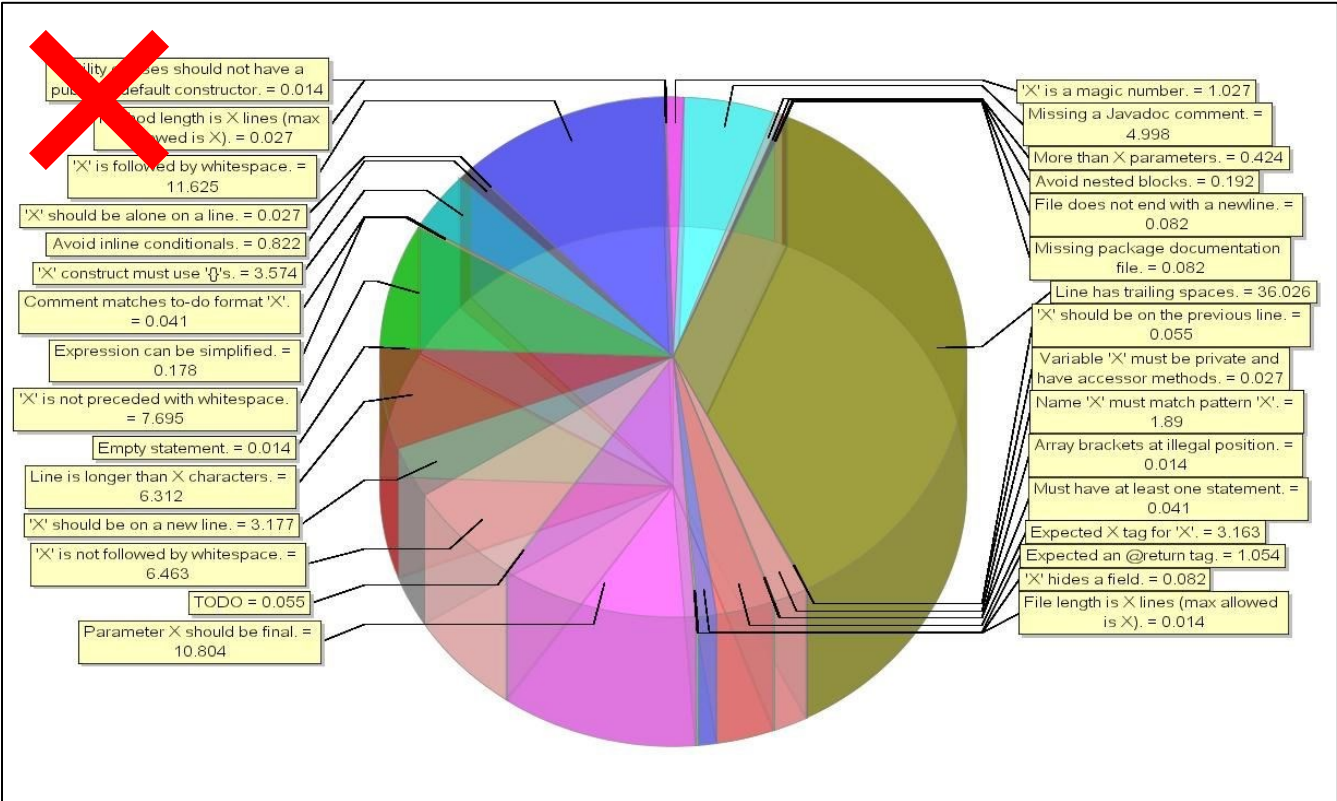
Visual First

🔨 Get to know your data through exploratory data visualization (summary tables, boxplots, scatter plots, dimensional reduction, etc.)



Visual First

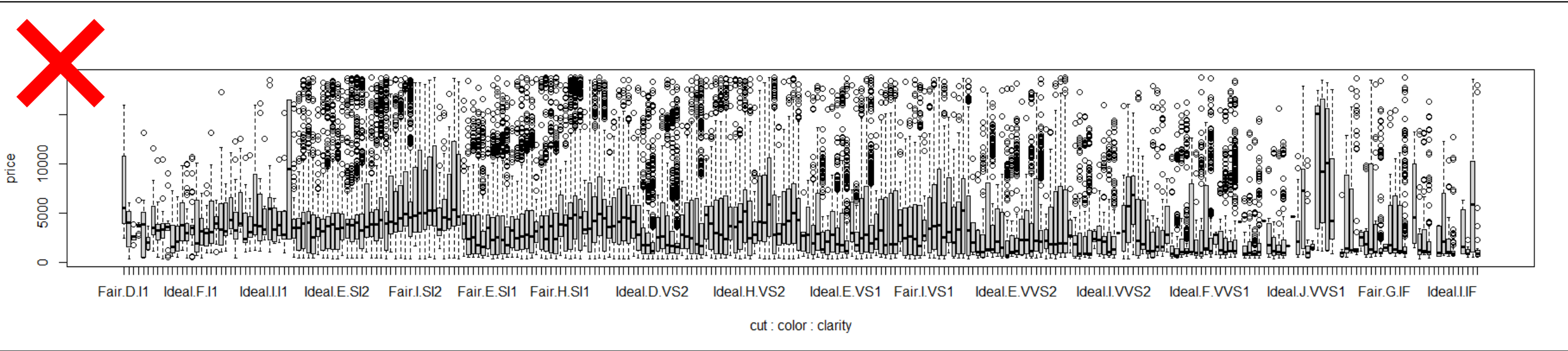
⚡ Get to know your data through exploratory data visualization (summary tables, boxplots, scatter plots, dimensional reduction, etc.)





Visual First

⚡ Get to know your data through exploratory data visualization (summary tables, boxplots, scatter plots, dimensional reduction, etc.)



? Data Query

⚡ Answer questions to get you thinking about your data to figure out structure and variable information

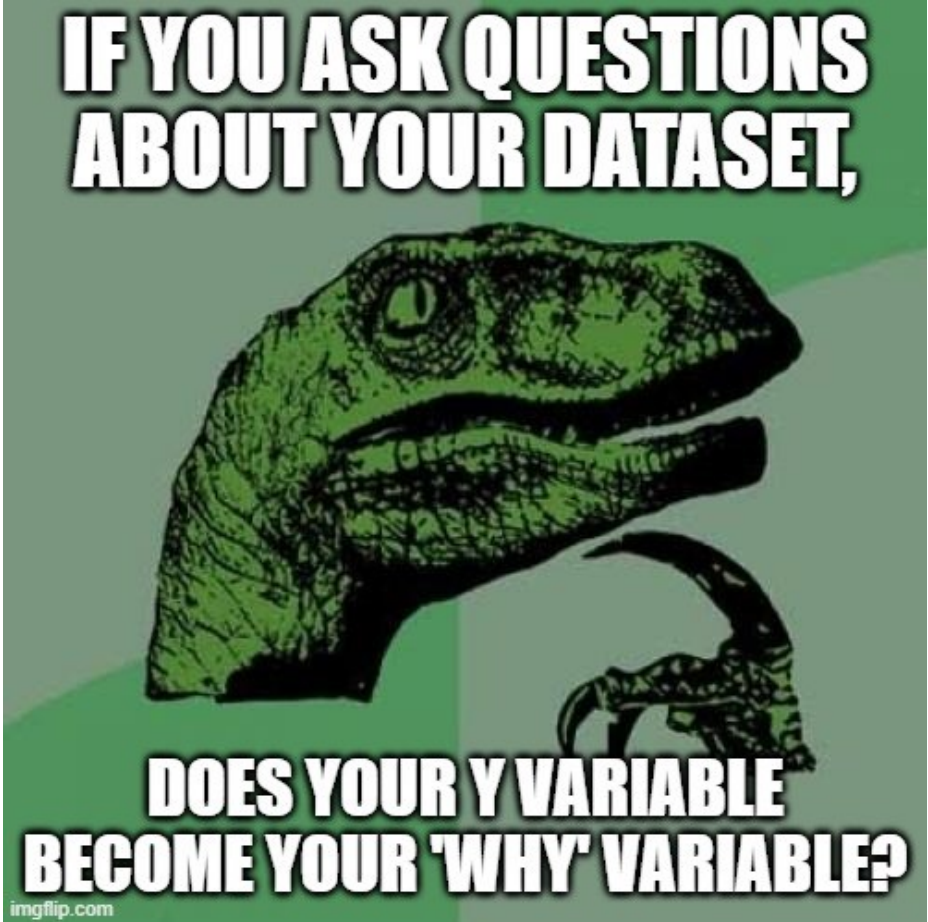
Communicating Your Data to Statisticians

BERDC Special Topics Talk 4

DaCCoTA
DAKOTA CANCER COLLABORATIVE
ON TRANSLATIONAL ACTIVITY

Dr. Mark Williamson
Biostatistics, Epidemiology,
and Research Design Core

https://und.qualtrics.com/jfe/form/SV_dcyUSPLhD4cmP5Q



? Data Query

⚡ Answer questions to get you thinking about your data to figure out structure and variable information

Communicating Your Data to Statisticians

BERDC Special Topics Talk 4

DaCCoTA
DAKOTA CANCER COLLABORATIVE
ON TRANSLATIONAL ACTIVITY

Dr. Mark Williamson
Biostatistics, Epidemiology,
and Research Design Core

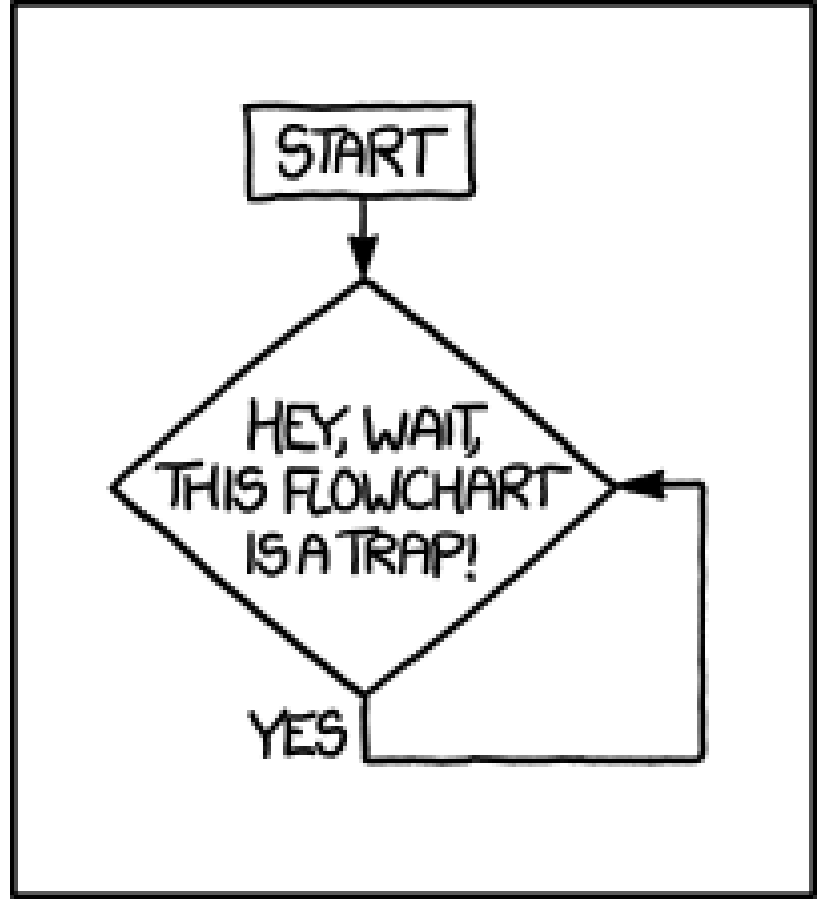
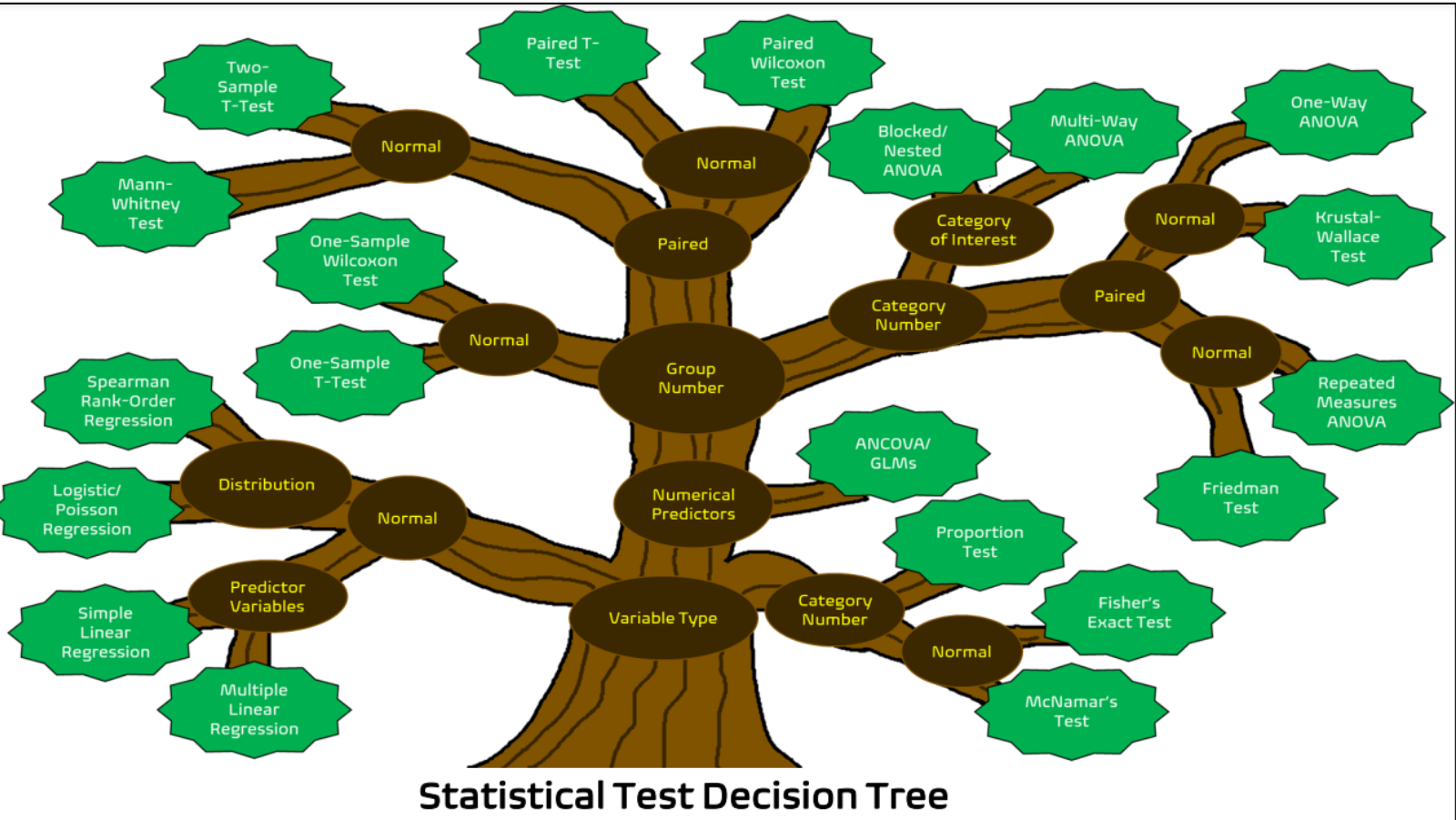
#	Question
1	What is your Y variable ?
2	How is your Y variable measured (categorical-> ordinal or nominal, numerical -> continuous or discrete)?
3	What are your X variables ?
4	How are your X variables measured (categorical-> ordinal or nominal, numerical -> continuous or discrete)?
5	Are their major considerations you need to be aware of (confounding factors, longitudinal design, missing data, etc.)?

https://und.qualtrics.com/jfe/form/SV_dcyUSPLhD4cmP5Q



Decision Tree

🔨 Walk through branching decisions to determine appropriate test to use



Decision Tree

🔨 Walk through branching decisions to determine appropriate test to use

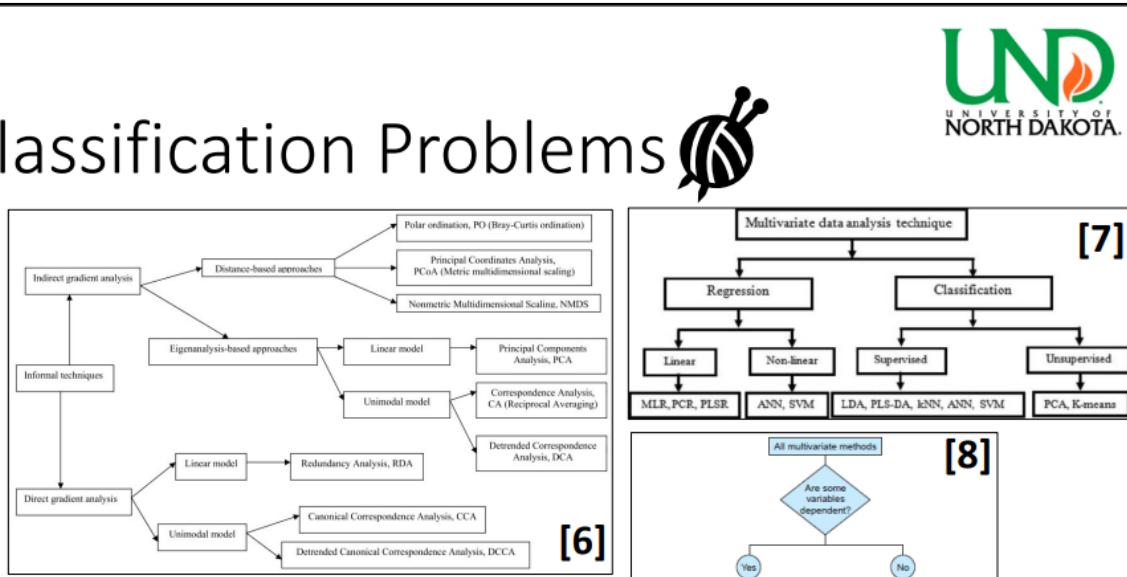
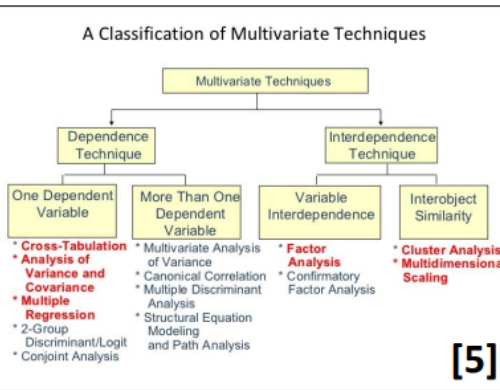
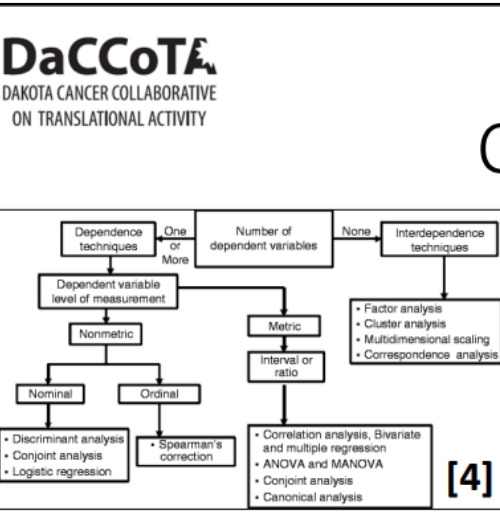
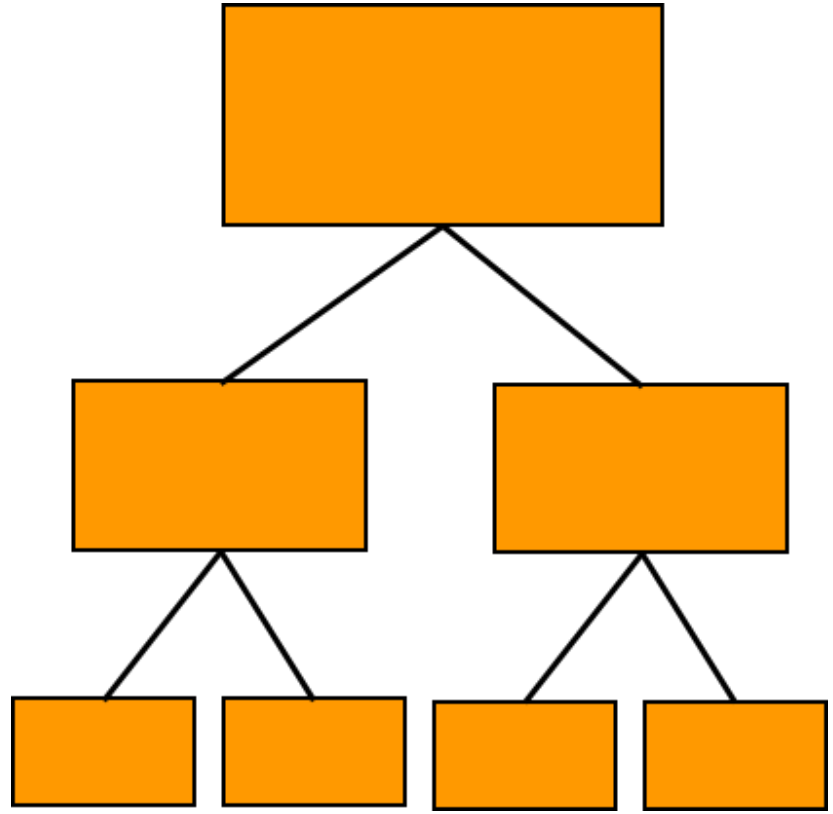
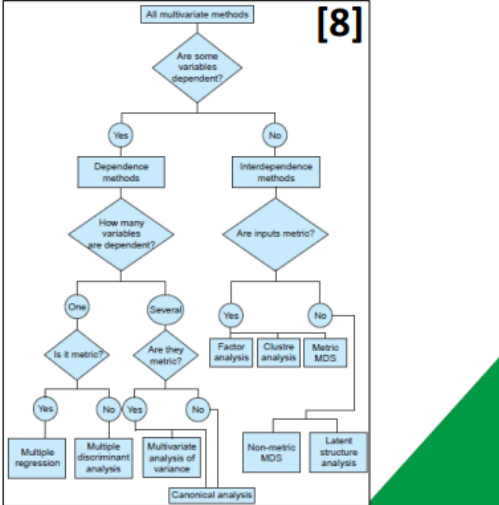
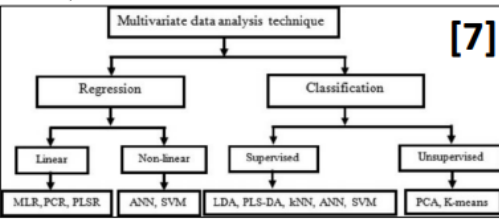


TABLE 2.1 Taxonomy of Common Multivariate Statistical Techniques

	Multiple predictors		Multiple outcomes		Multiple predictors and outcomes	
	Exploratory	Confirmatory*	Exploratory	Confirmatory*	Exploratory	Confirmatory*
Degree of association					Canonical correlation	None*
Multiple regression	Hierarchical multiple regression	Factor analysis (unconstrained factor extraction)	Factor analysis (specific factor extraction)	Multidimensional scaling (specified dimensionality)	Confirmatory factor analysis (maximization of fit indices)	Confirmatory factor analysis (nested models)
Logistic regression	Hierarchical logistic regression	Multidimensional scaling (unspecified dimensionality)	Factorial MANOVA (post hoc comparisons)	Factorial MANOVA (stepdowns and/or planned comparisons)	Factorial MANOVA (stepdowns and/or planned comparisons)	Factorial MANOVA (stepdowns and/or planned comparisons)
Group differences	ANOVA (planned comparisons)	One-way MANOVA (post hoc comparisons)	One-way MANOVA (stepdowns)	Factorial MANCOVA (post hoc comparisons)	Factorial MANCOVA (stepdowns and/or planned comparisons)	Factorial MANCOVA (stepdowns and/or planned comparisons)
ANCOVA (post hoc comparisons)	ANCOVA (planned comparisons)	One-way ANCOVA (post hoc comparisons)	One-way ANCOVA (stepdowns)	Factorial MANCOVA (post hoc comparisons)	Factorial MANCOVA (stepdowns and/or planned comparisons)	Factorial MANCOVA (stepdowns and/or planned comparisons)
Group membership	Hierarchical one-way discriminant analysis	Cluster analysis	None*	Factorial discriminant analysis	Hierarchical factorial discriminant analysis	Hierarchical factorial discriminant analysis

* All listed techniques could conceivably be "confirmed" through cross-validation analysis using multiple samples. The confirmatory procedure has a more one-sample design.
 * There are no unique confirmatory applications for these inherently exploratory statistical techniques.
 * ANOVA, analysis of variance; MANOVA, multiple analysis of variance; ANCOVA, analysis of covariance; MANCOVA, multiple analysis of covariance

[9]

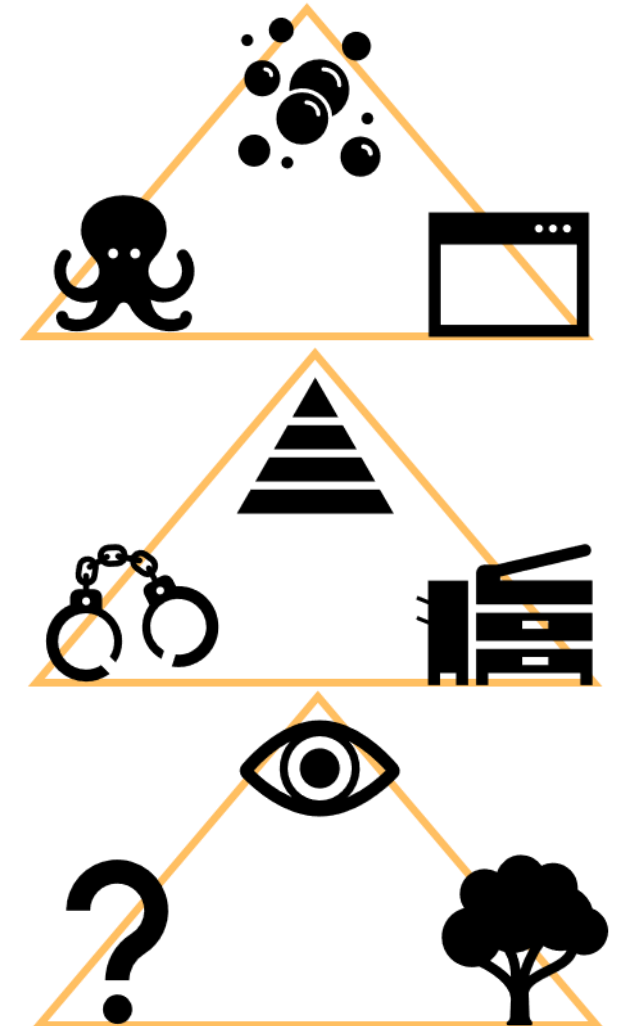


Conclusions

- ✂ Working with a biostatistician your best, but not only, option
- ✂ If you struggle with a busy schedule, analyzing complex data, or getting your head around unknown data, there are a bundle of strategies to help you survive
- ✂ Strategies are boilerplate code, flexible models, GUI usage, tiering methods, cheat sheets, multiple models, visualization first, data questionnaires, and decision trees

Please take the survey:

Survey: https://und.qualtrics.com/jfe/form/SV_9oWw3jNUcqDL8Gy



References

Resources

- [1] <https://www.youtube.com/watch?v=nkkv96teOxs>
- [2] https://med.und.edu/daccota/_files/docs/berdc_docs/mmgg_r_code.txt
- [3] <https://www.youtube.com/watch?v=YtmloAwp2rI>
- [4] <https://www.rstudio.com/resources/cheatsheets/>
- [5] https://med.und.edu/daccota/_files/docs/berdc_docs/statistical_software_toolkit_handout.docx
- [6] <http://www.ecostat.unical.it/Tarsitano/Didattica/LabStat2/Everitt.pdf>
- [7] <https://www.youtube.com/watch?v=eiivMNBZvw8>
- [8] https://med.und.edu/daccota/_files/pdfs/berdc_resource_pdfs/stat_decision_tree.pdf
- [9] <https://www.youtube.com/watch?v=kpmnFMznYt8>
- [10] <https://www.youtube.com/watch?v=YeKXdLFw0YI>

Images

- [1] <https://d3gqasl9vmjfd8.cloudfront.net/83b73cac-f64d-4f7a-b6a5-78bbc6e6c559.png>
- [2] <https://plotly-book.cpsievert.me/images/plotlyGGally.gif>
- [4] <https://blogs.sas.com/content/graphicallyspeaking/files/2013/03/VBox4.png>
- [5] <https://i.pinimg.com/originals/47/1f/29/471f291ba7a7cd81c2118fe37be00b05.jpg>
- [6] <https://xkcd.com/1195/>

Acknowledgements



The DaCCoTA is supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number U54GM128729.

For the labs that use the Biostatistics, Epidemiology, and Research Design Core in any way, including this Module, please acknowledge us for publications: *"Research reported in this publication was supported by DaCCoTA (the National Institute of General Medical Sciences of the National Institutes of Health under Award Number U54GM128729)"*

DaCCoTA
DAKOTA COMMUNITY COLLABORATIVE
ON TRANSLATIONAL ACTIVITY

