ST2137 Cheatsheet
by Yiyang, AY22/23

4. Numerical Data Analysis

For unimodal distri, **Skewed Right** / **Positively Skewed** if peak is towards the left & the right tail is longer (e.g. income): $\frac{\sqrt{n(n-1)}}{n-2} \times \frac{m_3}{(m_2)^{3/2}}$ where $m_2 = \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2$ and $m_3 = \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^3$.
Higher (lower) **Kurtosis** values indicate a sharper (less distinct) peak: $\frac{n-1}{(n-2)(n-3)}\left[\frac{(n+1)m_4}{m_2^2} - 3(n-1)\right]$.
Graphical summaries for 1 quantitative: [1] Histogram & Density Plot, [2] Boxplot, [3] QQ Plots, plots of standardised sample quantiles against theoretical quantiles of a standard normal.
Summaries for 2 quantitative: [1] Correlation Val., [2] Scatterplot.
Summaries for quantitative & categorical: [1] Boxplots by Groups, [2] Histogram by Groups.

5. Robust Estimators

Location Estimators: [1] Arithmetic mean, [2] Trimmed mean, [3] Winsorized mean, [4] M-Estimates.
$100\alpha\%$ **Trimmed Mean** is calculated by: [1] Discard lowest $100\alpha\%$ and highest $100\ \alpha\%$. [2] Arithmetic mean of remaining. Note: [1] 2α of extreme data discarded. [2] Usually $\alpha \in [0.1, 0.2]$.
$100\alpha\%$ **Winsorized Mean** is calculated by: [1] Sort observations as $x_{(1)}, x_{(2)}, ..., x_{(n)}$. [2] Replace $[n\alpha]$ smallest observations with $x_{([n\alpha]+1)}$, and $[n\alpha]$ largest with $x_{(n-[n\alpha])}$. Here, $[a]$ denotes as the nearest integer of a. [3] Arithmetic mean of replaced.
M-Estimator w. non-const err.func ρ: $T = \arg\min_T \sum_{i=1}^n \rho(x_i - T)$.
Scale Estimators: [1] **IQR** $\text{IQR} = Q_3 - Q_1$ [2] **Median Abs Devia-n** $\text{MAD} = \text{med}_i(|x_i - \text{med}_j(x_j)|)$ [3] **Gini's Mean Diff** $G = \sum_{i<j}|x_i - x_j|/C_2^n$ .For normal, $\text{IQR} = 1.35\sigma$, $\sigma = \text{MAD} * 1.4826$, $\sqrt{\pi}G/2 = \sigma$.

6. Categorical Data Analysis

Summaries for 1 categorical: [1] Frequency Table (with category of highest frequency as **Modal Category**), [2] Bar plot.
Contingency Table - Row for explanatory var x & column for response Y (success or fail). Measures of association: [1] Sample Diff. $= p_1 - p_2$, [2] Relative risk $= p_1/p_2$, [3] Odds Ratio.
For a success prob. π, **Odds of Success** odds $= \pi/(1-\pi)$. For 2-way contingency table, **Odds Ratio** (OR), θ, & **Sample OR**, $\hat{\theta}$, are: $\theta = \frac{\pi_1/(1-\pi_1)}{\pi_2/(1-\pi_2)}$, $\hat{\theta} = \frac{p_1/(1-p_1)}{p_2/(1-p_2)} = \frac{n_{11}\times n_{22}}{n_{12}\times n_{21}}$ for n_{ij} cell counts.
The $100\%(1-\alpha)$ Confidence Interval for OR: $\exp\{\log\hat{\theta} \pm z_{\alpha/2} \times ASE(\log\hat{\theta})\}$ and $ASE(\log\hat{\theta}) = \sqrt{1/n_{11} + 1/n_{12} + 1/n_{21} + 1/n_{22}}$
Note: If x and Y independent, $\theta = 1$.
Prospective Studies sample subjects randomly from a population and randomly assign exposure variables or record exposure status. All 3 measures above are valid.
Retrospective Studies sample a group of cases and a group of controls (i.e. based on Y), and check each subject's exposure. As such, **cannot** obtain valid estimates of π_1, π_2, as we obtain $Pr(x|Y)$ but need to estiamte $Pr(Y|x)$. Can use odds ratio for test only
Dependence Test - Chi-squared Test
Assumption: All $e_{ij} \geq 5$. (**Fisher Exact Test** if small size). Null: Two var.s independent. Statistic: $\chi^2 = \sum \frac{(|o_{ij}-e_{ij}|-0.5)^2}{e_{ij}} \sim \chi_1^2$ for o_{ij}, e_{ij} observed & expected count. ExpCnt = RowTotal × ColTotal/Total.
Dependence Test - McNemar's Test
Settings: x and Y represent num. of students passing & failing a test before & after a lesson. **Dependent samples**. Null: Before & after independent. Statistic: let b, c denotes pass-then-fail & fail-then-pass: $\chi^2 = \frac{(b-c)^2}{b+c} \sim \chi_1^2$, or if small sample, $\frac{(|b-c|-1)^2}{b+c} \sim \chi_1^2$
Dependence Test - Chi-Square for General Tables
Assumption: Large samples, or $\leq$ 25% cells with expected $<$ 5. Settings: Contingency table with r rows & c cols now.
Null & Statistic Same but follows χ^2 with d.f. $(c-1)\times(r-1)$ now.
Standardised / Adjusted Residual for each cell: $r_{ij} = \frac{o_{ij}-e_{ij}}{SE(o_{ij}-e_{ij})}$, $SE = \sqrt{e_{ij}(1-p_{i+})(1-p_{+j})}$ for p_{i+} and p_{+j} marginal prob. of row i and of col j. Note: $|r_{ij}| > 2$ cell's lack of fit of H_0.
Dependence Test - Linear-by-Linear Ordinal Data Null: Two var independent. Statistic $M^2 \sim \chi_1^2$ approx. for large n.

7. Hypothesis Testing

One-Sampled t Null: $\mu = \mu_0$. Statistic, $t = \bar{X} - \mu_0/se(\bar{X}) \sim t_{n-1}$.
One-Sampled Wilcoxon Signed Rank Null: $Med = m_0$. Statistic: let $V^+ = \sum_{i=1}^n (I(x_i > m_0)$ & $<$ for V^-. Then test stat $V = \min(V^+, V^-) \sim Bin(V^+ + V^-, 0.5)$.
Two-Sample Dependent Take pair difference & use one-sampled.
Two-Sampled t Null: $\mu_x = \mu_y$. Statistic: $t = \bar{X} - \bar{Y}/se \sim t_{n_1+n_2-2}$. $se = s_p\sqrt{1/n_1 + 1/n_2}$ where $s_p^2 = \frac{(n_1-1)s_X^2 + (n_2-1)s_Y^2}{n_1+n_2-2}$.
Two-Sampled Indep - Mann-Whitney U / Wilcoxon Rank Sum Idea: Used to check if two grps of data too different, by comparing their rank sum with those uniformly distributed in a pooled grp.

8. Analysis of Variance

Definition: For Y_{ij}, j-th observation of i-th grp, **One-Way ANOVA**, $Y_{ij} = \mu + \alpha_i + e_{ij}$, $i = 1, ..., I, j = 1, ..., J$, subject to $\sum_{i=1}^I \alpha_i = 0$
$SS_W = \sum_{i=1}^I \sum_{j=1}^J (Y_{ij} - \bar{Y}_i)^2$ in-grp varia-n. $SS_B = J\sum_{i=1}^I (\bar{Y}_i - \bar{\bar{Y}})^2$ btw-grp varia-n. $SS_{TOT} = \sum_{i=1}^I \sum_{j=1}^J (Y_{ij} - \bar{\bar{Y}})^2 = SS_W + SS_B$.

Tests - Null: a_i all same. Statistic: $F = \frac{SS_B/(I-1)}{SS_W/[I(J-1)]} \sim F$.

If grp size $J_1, ..., J_I$ different, total size n, $E(SS_W) = \sigma^2\sum_{i=1}^I (J_i - 1)$, $E(SS_B) = (I-1)\sigma^2 + \sum_{i=1}^I J + i\alpha_i^2$, and F has df $I-1, n-I$.

Assumptions & Checks: [1] Random samples [2] Equal var: (1a) **Bartlett Test** sample assumed normal, (1b) **Levene Test** sample distri unknown. [34] Errors iid.: (2a) **Shapiro Wilk Test** on residual, (2b) **KS Test**, (2c) plot. [5] Additivity of treatment effects.
Kruskal-Wallis Test: Non-parametric version of ANOVA: no normal assumption, good for small sample size.
Multiple Comparisons: [1] **Bonferroni**: control k hypotheses' α at a, a/k for each, no need normal. [2] **Tukey**: For pairs, in ANOVA. [3] **Least Signif Diff**: null grp means same, in ANOVA.

9. Regression Analysis

Assumptions: [1] Linear relationship. [2] Normality & equal & const var. [3] Regressors uncorrelated.
$R^2 = \frac{SS_R}{SS_T}$, $R_a^2 = 1 - \frac{SS_{res}/(n-p)}{SST(n-1)}$. SS_R: $(\hat{y}_i - \bar{Y})^2$, SS_{res}: $(y_i - \hat{y}_i)^2$.
Overall Null: $\beta = 0$. Statistic: $F_0 = \frac{SSR/p}{SS_{Res}/(n-p-1)} \sim F$ rej large F_0.
Individual Null: $\beta_i = 0$. Statistic: $t_i = \frac{\hat{\beta}_i}{se(\hat{\beta}_i)} \sim t_{n-p-1}$.
Model Check: [1] **Outlier** if $|sr|$ large. [2] **Influential Point** if Cook' Distance $D_i i = \frac{r_i^2 h_{ii}}{p(1-h_{ii})} > 1$. [3] **Leverage** $h_i i$ of $H = X(X'X)^{-1}X'$

10. Simulation

Congruent Generator: 1) Choose $a, c, m \in Z$, & seed X_0. 2) Define $X_{n+1} = (aX_n + c) \bmod m$. Note: If we need uniform random values, $U_i = X_i/m \in [0, 1)$.
Theory of Inversion: [1] For X with CDF F, $Y = F(X) \sim U(0,1)$.
[2] For $Y \sim U(0,1)$ and X with CDF F, $F^{-1}(Y) = F$.
Inversion Method for generating from distribution F: 1) Generate $U \sim U(0,1)$. 2) Set $X = F^{-1}(U)$ assuming inverse exists. 3) Output X, following F.

R Coding

```
# Vector
numeric(n); character(n)  # vector with n 0's / ""'s
rep(a, b)  # replicate item a by b times
seq(from=a,to=b,by=c); seq(from=a,to=b,length=d);
# Matrix
matrix(v, nrows=a, ncols=b, byrow=T); rbind(...); cbind(...)
# Dataframes
df <- data.frame(m); names(df) = c(...); row.names(df)= c(...)
df[a,b:c]; df$abc; df[order(val),]; merge(df1, df2, by="id")
df[rev(order(val)),] # asc, desc

if (condition) {...} else {...} # Conditioning
while (condition) {...} # While loop
for (<variable> in <range>) {...} # For loop
read.csv(..., header=T, width=c(...)), read.table(...) # IO
# Note: Use width if each variable spans multiple lines
write.table(data, "C:/...")
cat(...); sink() # print
# Random
set.seed(999); x = rnorm(n,0,1); random std norm size n
```

4. Numerical Data Analysis

```
# descriptive stats, location
length(x); summary(x); mean(x); median(x); quantile(x)
# descriptive stats, variability
range(x); var(x); sd(x); IQR(x);  x[order(x)[1:5]] # smallest 5
# skewness
skew <- function(x){
  n<-length(x); m3<-mean((x-mean(x))^3); m2<-mean((x-mean(x))^2);
  sk=m3/m2^(3/2)*sqrt(n*(n-1))/(n-2); return(sk) }
# kurtosis
kurt = function(x) {
  n=length(X); m4=mean((x-mean(x))^4); m2=mean((x-mean(x))^2)
  kurt=(n-1)/((n-2)*(n-3))*((n+1)*m4/(m2^2)-3*(n-1))}

# Histogram w. Density Plot
hist(mark, freq=FALSE, main="Hist", xlab="mark", ylab="val",
    axes=RUE, col="grey", nclass=10) x<-seq(0,30,length.out=98)
y<-dnorm(x,mean(mark),sd(mark)); lines(x, y, col = "red")
boxplot(mark, xlab = "mark") # Boxplots
qqnorm(mark, pch = 20); qqline(mark, col = "red") # QQ plots
# For association between two
cor(v1, v2); plot(v1, v2, pch=20) # Correlation val; Scatterplot
boxplot(energy~type) # Boxplots by Group
# Others
par(mfrow=c(2,2)); ...; # Subplots
par(new=TRUE); ...; # add new plots to same graph
```

5. Robust Estimators

```
mean(x); mean(x, trim=0.2) # arithmetic & 20% trimmed
winsor <- function(x, alpha=0.2){
  n=length(x); xq=n*alpha; x=sort(x); m=x[(round(xq)+1)];
  M=x[(n-round(xq))]; x[which(x<m)]=m; x[which(x>M)]=M
  return(c(mean(x),var(x)))}; winsor(x)
library(MASS); hubers(x, k=0.84) # Or use library
median(abs(x-median(x))); mad(x); IQR(x)
```

6. Categorical Data Analysis

```
count=table(data$type); barplot(count) # freq table, barplot
# Contingency table
ct <- matrix(c(...), ncol=2, byrow=2)
dimnames(ct)<-list(rowname=c(...),colname=c(...))
test<-prop.test(ct,correct=FALSE)
RR<-(test$estimate[1])/(test$estimate[2])
odds<-test$estimate/(1- test$estimate); OR<-odds[1]/odds[2]
# Fisher Exact Test
fisher.test(ct, alternative="two.sided")
# general Chi-squared Test
chisq.test(ct)
# McNemar Test
mcnemar.test(x, correct=TRUE)
# Linear-by-linear
set=as.table(read.ftable(...)); library(coin)
lbl_test(set,scores=list(MI=c(0,1),Alcohol=c(0,0.5,1.5,4,7)))

# fre table create new column, x for gender:
ggrp=factor(gender); levels(ggrp)=c("F", "M")
ggrp; table(ggrp) # below another method for drive grp
dgrp<-ifelse(drivelic=="Y","Yes","No"); table(dgrp)
tab = table(ggrp,dgrp) # cont table
```

7. Hypothesis Testing

```
# One-sampled t-Test
t.test(weight, mu=3.3,alternative="less")
# One-sampled Sign Test
weight.non.0=(weight[weight!=3.3]); w.len=length(weight.non.0)
binom.test(sum(weight<3.3), w.len, alternative="less")
# Wilcoxon Signed Rank Test
wilcox.test(weight.non.0, mu=3.3, alternative="less")
# Equal var test: null is equal; null assume normal
var.test(x,y); bartlett.test(weight_gain~level, data=data)
# Two-sampled t-Test
t.test(x,y, mu=0, var.equal=TRUE)
# Mann Whitney U Test
wilcox.test(bf,no.bf)
```

8. ANOVA

```
anova<-aov(amount~lab, data=data); summary(anova)
tapply(amount, lab, mean) # get group mean
# Kruskal Wallis
kruskal.test(amount~lab)
# Bonferroni
pairwise.t.test(amount, lab, p.adj = "bonf")
# Tukey
TukeyHSD(anova) # default family alpha 0.05
# LSD, I(J-1)=63, alpha-0.05
MSW=sum(anova$res^2)/63; lsd<-qt(0.975,63)*sqrt(MSW*2/7)
# Model assumption checks
shapiro.test(anova$res) # Shapiro for residual normality
ks.test(resid,"pnorm",mean(resid),sd(resid)) # KS normality
bartlett.test(amount~lab, data=newdata) # equal var
```

9. Regression Analysis

```
m1<-lm(weight~height+age, data=data); summary(m1); anova(m1)
plot(weight,height, type = "n") # plot by gender, M then F
points(weight[gender=="M"], height[gender=="M"],col="red")
m1$res; rs=rstandard(m1); m1$fitted.values # r, sr, fitted
summary(m1)$r.squared; summary(m1)$sigma # r2, sigma hat
# QQ Plot of SR
qqnorm(rs,datax=TRUE,ylab="SR", xlab="Z scores",)
qqline(rs,datax=TRUE,col="red") # datax: theory-qnt Y, obs X
# SR against fitted
plot(m1$fitted.values,rs, xlab="fitted"); abline(h=0)
# Predicted
predict(m1, newdata=data.frame(height=c(65,63), age=c(40,36)),
    interval="confidence",level=0.95)
# Model Check
x=cbind(c(rep(1,n)),height); hat=x%*%solve(t(x)%*%x)%*%t(x)
lvg=diag(hat); lvg[which(lvg>2*p/n)] # Leverage & check
cooks.distance(m1) # Cook's distance
```

Python Coding

```
import pandas as pd
import scipy.stats as scst
import matplotlib.pyplot as plt
import statistics as st

# matrix
mat=np.asmatrix([[...],...]); mat.T; mat.I
np.vstack((...)); np.column_stack((...))
# Dataframe
dat={'X':[...],'Y':[...]};pd.DataFrame(dat,columns=['X','Y'])
df1=df.rename({'X':'NewX','Y':'NewY'}, axis=1)
```

4. Numerical Data Analysis

```
# Descriptive stats
df['x'].median(); df['x'].var(); df['x'].std()
df['x'].quantile(0.25); df['x'].quantile(0.75)
# Histogram w. Density Plot
l=list(np.arange(0,30,0.5))
y=scst.norm.pdf(l,loc=mean(x),scale=st.stdev(x)) # qnorm
plt.plot(l, y); plt.hist(data['x1'], density=True)
plt.title('...'); plt.xlabel('...'); plt.ylabel('...')
# Boxplot
plt.boxplot(data['x1'])
# QQ Plot
scst.probplot(x, dist="norm", plot=pylab); pylab.show()
# Scatterplot
plt.scatter(v1, v2)
# Scatterplot by Group (tut3Qn2)
groups=data.groupby("x11")
for name, grp in groups:
    plt.plot(grp["x"], grp["y"], label=name)
# Boxplots by Group
fig, ax = plt.subplots(figsize=(7,5))
bats.boxplot(column=['energy'], by='type',ax=ax,color='b')
# Others
plt.legend(); plt.show()
# correlation:
np.corrcoef(x, y)[0, 1]
```

6. Categorical Data Analysis

```
import statsmodels.api as sm
from statsmodels.stats.contingency_tables import mcnemar
# Table & Barplots
tab=pd.crosstab(index=data["type"],columns=data["count"])
plt.bar(type,counts)
# Cont table, using df or Numpy 2D array
scst.chi2_contingency(ctable, correction = True)
# Fisher Exact Test
scst.fisher_exact(ctable, alternative='two-sided')
# McNemar Test
mcnemar(ctable, exact=False, correction=True)
# General Chi-squared
scst.chi2_contingency(obs, correction=True)
# Linear-by-Linear association test
ct=sm.stats.Table(np.asarray(table)); rsc=np.asarray([0,1])
csc=np.asarray([0,0.5,1.5,4,7]) # scores for 2 rows 5 columns
ct.test_ordinal_association(row_scores=rsc, col_scores=csc)
```

7. Hypothesis Testing

```
# One-sampled t-Test
t, p = scst.ttest_1samp(weight, popmean=3.3)
# Wilcoxon Signed Rank test:
scst.wilcoxon(weight-3.3, y=None, zero_method='wilcox',
    correction=True, alternative='less')
# Equal var test
t, p = scst.bartlett(x,y)
# Two-sampled t-Test
scst.ttest_ind(x, y, axis=0, equal_var=True)
# Two-sampled ManWhitney U test / Wilcoxon Rank Sum Test:
scst.mannwhitneyu(x,y,use_continuity=True,alternative...)
# Two-sampled Paired t Test
scst.ttest_rel(after,before) #, nan_policy='propagate')
```

8. ANOVA

```
import statsmodels.stats.multicomp as mc
# ANOVA
m1=ols('amount~lab', data=newdata).fit()
anova=sm.stats.anova_lm(mod, type=2)
# Another method
anova2=scst.f_oneway(lab1, ..., lab7);print(anova2)
# Kruskal Wallis
krus=scst.kruskal(lab1, ..., lab7); print(krus)
# Bonferroni
comp=mc.MultiComparison(newdata['amount'],newdata['lab'])
res,tb1,tb2=comp.allpairtest(stats.ttest_ind,method="bonf")
print(res)
# Tukey
tukey=comp.tukeyhsd(); print(tukey.summary())
# Model check
scst.shapiro(mod.resid) # normality check
test=np.random.normal(mean(amount),np.std(amount),70)
scst.ks_2samp(amount,test) # KS test for amount, same resid
scst.bartlett(lab1, ..., lab7) # equal var assume norm
scst.levene(lab1, ..., lab7) # equal var
```

9. Regression Analysis

```
from statsmodels.formula.api import ols
scst.pearsonr(data['W'], data['H']); df.corr()
m1=ols("W~H+age",data=data).fit(); print(m1.summary())
anova1 = sm.stats.anova_lm(model, typ=1); print(anova1)
m1.bse,m1.mse_resid,np.sqrt(m1.mse_resid) # stderr MSR RSE
fitted = m1.fittedvalues;
# Model Check
model.resid # std residual
ana = model.get_influence()
SR = analysis.resid_studentized_internal
leverage = analysis.hat_matrix_diag
cooks_d, p = analysis.cooks_distance
```

Others

```
from scipy.stats import norm
x = norm.ppf(0.975)
# Random
np.random.seed(999)
np.random.uniform(0,1,6) # 6 of U(0,1), norm in QQ plot
np.random.exponential(1/5, n); np.random.weibull(4, 10)
...binomial(n=100, p=0.3, size=10) ...poisson(lam=3, size=10)
```

SAS Coding (# FOR LINEBREAK)

```
data ex_1;# input subject gender $ CA1 CA2 HW $;
datalines;  # 10 m 80 84 a  # 7 m 85 89 a #;
PROC means data=ex_1  mean var Q1 Median Q3 min max;
  var CA1 CA2; # run;
/* Read from CSV */
FILENAME REFFILE '...'; # PROC IMPORT DATAFILE=REFFILE
  # DBMS=CSV  # OUT=WORK.heat;  # GETNAMES=YES;# RUN;
PROC CONTENTS DATA=WORK.heat; RUN;
/* Read from txt */ PROC IMPORT DATAFILE=REFFILE
  # DBMS=DLM  # OUT=WORK.example1;  # DELIMITER=",";
  GETNAMES=NO;  # DATAROW=1;# RUN;
/* Export data */ PROC EXPORT data=ex_1
  outfile=_dataout  # dbms=csv replace;# run;
/* CHANGING VARIABLE NAMES */ DATA ex_1;
    set ex_1(rename=(var1=id var2=gender ...));# run;
/* To create the labels */ proc format;
    value $gen 'F'='Female' 'M'='Male';# run;
```

6. Categorical Data Analysis

```
/* McNemar Tes, agree means no correction*/
proc freq data=debate;# by gender;
tables before*after/agree; # weight count;
title "Chi-square test for the paired samples";
run;
/* Test for normality */
proc univariate data=datamark normal ;
var mark;# histogram mark /normal;
qqplot /normal (mu=est sigma=est);# run;
```

7. Hypothesis Testing

```
/* One-sampled t-Test, two versions */
/* It includes sign test and signed rank test */
PROC UNIVARIATE data=baby mu0=3.3;
var weight; run;
PROC TTEST data = baby H0=3.3; *sides = L or U;
var weight; run;
/* Two-sampled Mann-Whitney U Test */
PROC NPAR1WAY data=weightgain wilcoxon;
  class level;  # var weight_gain;  # *exact wilcoxon;
run;
/* Paired t-test*/
PROC TTEST DATA=platelet;
    PAIRED after*before;# RUN;

/* Descriptive stats by group/level */
proc means data=weightgain n nmiss mean std
    stderr median min max qrange maxdec=4;
class level; var weight_gain;# run;
/* Test for normality & produce CI on median */
proc univariate data=weightgain normal cipctldf;
class level;
var weight_gain;
histogram weight_gain /normal;
qqplot /normal (mu=est sigma=est);# run;
/* Produce boxplots */
proc sgplot data=weightgain;
title 'Boxplot of weight gain by level of protein';
vbox weight_gain /category=level;# run;
```

8. ANOVA

```
PROC ANOVA data=newdata;# class lab;
model amount=lab;# means; # run;
/* Kruskal Wallis Test */
PROC NPAR1WAY data=newdata wilcoxon dscf;
class lab;# var amount;# run;
/* Bonferroni, Tukey */
PROC ANOVA data=newdata;# class lab;
model amount=lab;
means lab / Bon cldiff alpha=0.05;# run;
means lab / tukey cldiff alpha=0.05; # run;
/* Model check, add after model amt=lab line */
means lab / hovtest=levene alpha-0.05;
means lab / hovtest=BARTLETT alpha-0.05;
/* normality plot */
PROC UNIVARIATE data=newdata normal;
var amount;# histogram amount /normal;
qqplot /normal (mu=est sigma=est);# run;
```

9. Regression Analysis

```
/* Create dummy */
data example1;# set example1;
if gender="M" then gen=1;# if gender="F" then gen=0;
run;
/* Correlation values */
proc corr data=example1 nosimple;
title "Example of a correlation matrix";
var height weight age;# run;
/* Scatterplot of height vs weight by gender */
proc sgscatter  data = example1;
   plot height * weight
   datalabel = gender group = gender;# run;
/* Multiple model, SS1 is ANOVA SSR*/
proc reg data=example1;
  model weight = height age/SS1;# run;# quit;
/* Model with interaction term, create first */
data example1;# set example1; # hg=height*gen;# run;
proc reg data=example1;
  model weight = height age gen hg;# run;# quit;
/* Normality test for SR */
proc univariate data=analysis normal;
var resid;# histogram resid /normal;
qqplot /normal (mu=est sigma=est);# run;
/* Make prediction */
/* alpha default 0.05; lclm, uclm: lower, upper boundfor CI */
/* for CI; lcl, ulc: PI */
data example1;# set example1 end=last;# output;
if last then do;
  gender = . ;  # height = 64;  # weight = .;
  age =. ;  # output;# end;# run;
proc reg data=example1 alpha = 0.01;
  model weight = height;
```

```
output out=predict(where=(weight=.)) p=predicted
  uclm=UCL_Pred lclm=LCL_Pred;# run;# quit;
/* Model check */
proc reg data=crab;
  model weight = width s1 s2;
output out=check P=yhat STUDENT=SR;# run;# quit;
proc univariate data=check normal;# var SR;
histogram SR /normal;# qqplot /normal (mu=est sigma=est);
run;
proc sgscatter data = check;# plot SR*yhat;# run;
proc sgplot data = check;# SCATTER x=yhat y=SR;
   refline 0 / axis=y lineattrs=(thickness=2 color=darkred);
run;
```

10. Simulation

```
/* Generate random uniform */
data Ugen;# call streaminit(999); /* seed 999 */
do i = 1 to 10;
 x = rand('uniform', 2, 3);  # output;
end;# keep x;# run;
proc print data=Ugen;# var x; # run;
/* Generate other special distributions*/
rand('exponential', 1/5); *rate lambda = 5;
rand('weibull',4); *shape alpha = 4;
rand('normal',mu,sigma); rand('chisq',df);
rand('binom',p,n); rand('poisson',lambda);
```