Andrey Arguedas Espinoza - R cheatsheet

Basic Use

Include and install packages

```
if(!require(psych)){install.packages("psych")}
```

Set directory to read files

```
setwd(this.path::here())
```

Read a file

```
data <-readLines("file.txt")
```

Sort data

```
headTail(Data)
```

Compact display data

```
str(Data)
```

Summary of data

```
summary(Data)
```

Basic Analysis

Summarize with factors

```
Summarize(FactorX ~ FactorY, Data, digits = 4)
```

Normal Histograms

```
plotNormalHistogram(X)
```

Box Plot

```
M = tapply(Data$factorX, INDEX = Data$factorY, FUN = mean)
boxplot(factorX~ factorY, data = Data)
points(M, col="red", pch="+", cex=2)
```

Basic T-Test

```
t.test(factorX~ factorY, data=Data)
```

ANOVA monofactorial

Confidence interval charts

```
Sum = groupwiseMean(factorX~ factorY, Data, conf = 0.95, digits = 3, traditional = FALSE, percentile = TRUE)

ggplot(Sum ,aes(x=factorX, y=Mean))+
geom_errorbar(aes(ymin=Percentile.lower, ymax=Percentile.upper), width=0.05, size=0.5)+
geom_point(shape=15, size=4)+
theme_bw()+
theme(axis.title = element_text(face='bold'))+
ylab("Tiempo promedio, s")
```

ANOVA monofactorial

Lineal Model

```
lm(factorX~factorY, Data)
```

Anova

```
Anova(model, type="II")
```

Anova assumptions

Residuals Histogram

```
x = residuals(model)
plotNormalHistogram(x)
```

Homosteacidity

```
plot(fitted(model), residuals(model))

plot(model)
```

Post-Hoc Analysis

Means

```
lsmeans(model, ~factorX)
```

Pairs with tukey

```
pairs(marginal, adjust="tukey")
```

CLD

```
cld(marginal, alpha=0.05, Letters = letters, adjust="tukey")
```

CLD Groups chart

```
ggplot(CLD, aes(x = factorX, y = lsmean, label=.group)) +
geom_point(shape = 15, size = 4) +
geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.2, size = 0.7) +
theme_bw() +
theme(axis.title = element_text(face="bold"), axis.text = element_text(face = "bold"), plot.caption = element_text(vjust  = 0)) +
ylab("Promedio del minimo cuadrado \n Tiempo de ejecucion") +
geom_text(nudge_x = c(0,0,0), nudge_y = c(1100, 1100, 1100), color="black")
```

ANOVA multifactorial

Interaction Charts

```
interaction.plot(x.factor= Data$factorX,
                 trace.factor= Data$factorY,
                 response = Data$factorZ,
                 fun=mean,
type="b", col=c("black", "red", "green"),
                 pch=c(19,17,15),
fixed=TRUE, leg.bty="o")
```

Lineal Model

```
lm(factorX~ factorY* factorZ, data=Data)
```

Data Transformation

Square Root

```
T.sqrt = sqrt(Data$factorX)

model = lm(T.sqrt ~ factorX * factorY * factorZ, data=Data)

Anova(model, type="II")
```

Cubic Power

```
T.cub = sign(Data$factorX) * abs(Data$factorX) ^ (1/3)

model = lm(T.cub ~ factorY * factorZ* factorW, data=Data)

Anova(model, type="II")
```

Logarithmic

```
T.log = log(Data$factorX)

model = lm(T.log ~ factorY* factorZ * factorW
           , data=Data)

Anova(model, type="II")
```

Levene Test

```
leveneTest(T.log ~ factorX * factorY * factorZ, data=Data)
```

Group and Interaction charts

```
ggplot(Sum, aes(x = factorX, y = mean, color = factorX)) +
geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.2, size = 0.7, position = pd) +
geom_point(shape=15, size=4, position=pd) +
theme_bw() +
theme(axis.title = element_text(face="bold")) +

scale_color_manual(values= c("black", "red", "green"))
ylab("Description")
```

Non parametric tests

Create ranges

```
XT = xtabs(~factorX+ factorY.f, data=Data)
XT

prop.table(XT, margin = 1)
```

Ranges histogram

```
histogram(~factorY.f | factorX, data = Data, layout = c(1, 3))
```

Kruskal Test

```
kruskal.test(factorY~ factorX, data=Data)
```

Dunn Test

```
dunnTest(factorY~ factorX, data = Data, method = "bh")
```