

Author : Abhisek Mohanty
UFID : 34931899

NOTE : I have had to trim a few of the R Code due to space constraint. Important parts are in **BOLD BLUE** Color. Important Comments start with **##**.

Q1. R Code (Trimmed)

```
psadata=read.table("http://www.stat.ufl.edu/~athienit/STA6166/assignment3_1.txt",  
  col.names=c("ID","Y","Volume","Weight","Age","BPH","SVI","CP","GS"))
```

```
volume_trans=psadata$Volume  
y=psadata$Y
```

```
regm=lm(y ~ volume_trans + Weight + Age + BPH + CP + volume_trans:SVI + Weight:SVI + Age:SVI + BPH:SVI +  
CP:SVI + volume_trans:GS + Weight:GS + Age:GS + BPH:GS + CP:GS, data=psadata)
```

```
source("http://www.stat.ufl.edu/~athienit/check.R")
```

```
check(regm,tests=TRUE) ## Assumptions Fail. Looks far from Normal. Initial Plot attached after the code. Figure 1.1
```

```
## Use powerTrans on volume. Some of the variables throw an error, probably because of log 0  
## Volume transformations almost fixes the failed assumptions  
## Transformation Value comes to be around 0.1. Taking 0 (the log transformation) as its close to 0.1
```

```
library(car)  
bc2=powerTransform(y~volume_trans)  
summary(bc2)  
yT=bcPower(y,0)
```

```
psadata.model <- lm(yT ~ volume_trans + Weight + Age + BPH + CP + volume_trans:SVI + Weight:SVI + Age:SVI +  
BPH:SVI + CP:SVI + volume_trans:GS + Weight:GS + Age:GS + BPH:GS + CP:GS, data=psadata)
```

```
## volume_trans will be equal to log(Volume)
```

```
summary(psadata.model)  
anova(psadata.model)  
summary(psadata.model)$adj.r.squared  
AIC(psadata.model)
```

```
updatedpsamodel <- update(psadata.model, . ~ . -volume_trans:SVI)  
summary(updatedpsamodel)  
anova(updatedpsamodel)  
summary(updatedpsamodel)$adj.r.squared  
AIC(updatedpsamodel)
```

```
## Removed the other update steps due to space constraint. We update the updatedpsamodel hereafter.
```

```
#####Testing later#####
```

```
reg2=lm(yT~volume_trans + Age + BPH + Weight:SVI, data=psadata) ## Predictors in the Final Model  
summary(reg2)
```

```
check(reg2,tests=TRUE) ## Assumptions hold true after the transformation for the final model. Figure 1.2
```

```
##### Final Model #####
```

```
#
```

```
## log y = 1.274287 + 0.082334*log(Volume) + 0.004201*Age + 0.093149*BPH + 0.012624*Weight*SVI
```

```
## implies  $y = \exp(1)^{(1.274287 + 0.082334 \cdot \log(\text{Volume}) + 0.004201 \cdot \text{Age} + 0.093149 \cdot \text{BPH} + 0.012624 \cdot \text{Weight} \cdot \text{SVI})$ 
#
#####
```

```
predict.lm(reg2,se.fit=TRUE,newdata=data.frame(volume_trans = log(4.2633), Age = 68, BPH = 1.3500, Weight = 22.783, SVI = 0),interval="prediction",level=0.90) ## 90% CI.
```

Output below. These are the lower and upper bounds for log Y.

fit	lwr	upr
1.805075	0.4444069	3.165743

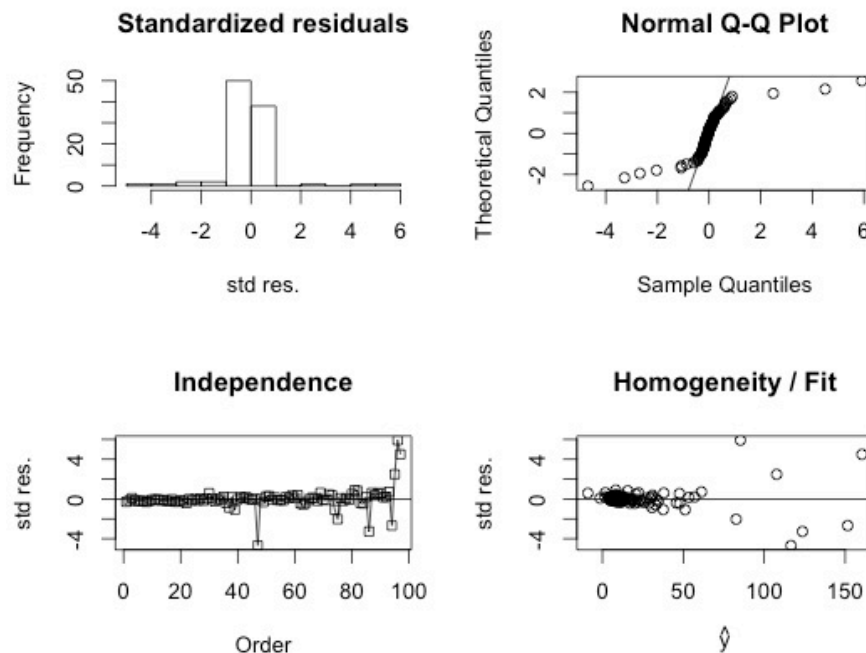


Figure 1.1

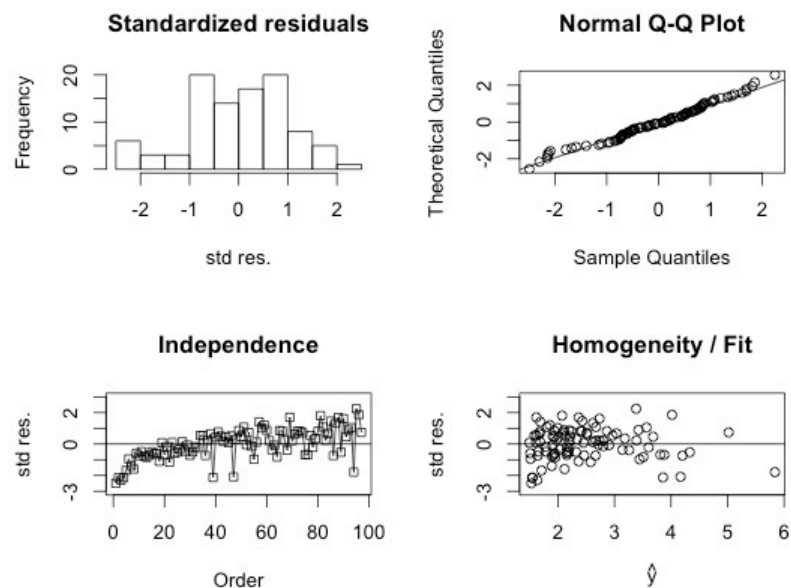


Figure 1.2

Q2. R Code :

```
aromatics_data = data.frame(rate=c(1.06, 0.79, 0.82, 0.89, 1.05, 0.95, 0.65, 1.15, 1.12),
                             solvent="Aromatics")
chloroalkanes_data = data.frame(rate=c(1.58, 1.12, 1.45, 0.91, 0.57, 0.83, 1.16, 0.43),
                                 solvent="Chloroalkanes")
ester_data = data.frame(rate=c(0.29, 0.43, 0.06, 0.06, 0.51, 0.09, 0.44, 0.10, 0.17, 0.55, 0.53, 0.17, 0.61, 0.34,
                                0.60),solvent="Esters")

allData <- rbind(aromatics_data,chloroalkanes_data, ester_data)

means = tapply(allData$rate,allData$solvent,mean);means
sdevs = tapply(allData$rate,allData$solvent,sd)

# Dot plot
stripchart(rate~solvent, data=allData, method="stack", vertical=TRUE,
            pch=1, cex=1.5, xlab="solvent", ylab="rate", main="Dotplots")
title(sub="pre-analysis plot", adj=0, cex=5/6)
mtext("Example")
points(c(1,2,3),tapply(allData$rate,allData$solvent,mean),col=2,pch=8)
abline(h=mean(allData$rate),col=3)
legend(3,250, c("Observations", "Trt Mean", "Grand Mean"), col = c(1,2,3), text.col= "black",
       lty=c(0,0,1),pch=c(1,8,NA),bg='gray90')

attach(allData)

m1=aov(rate~solvent)
summary(m1)

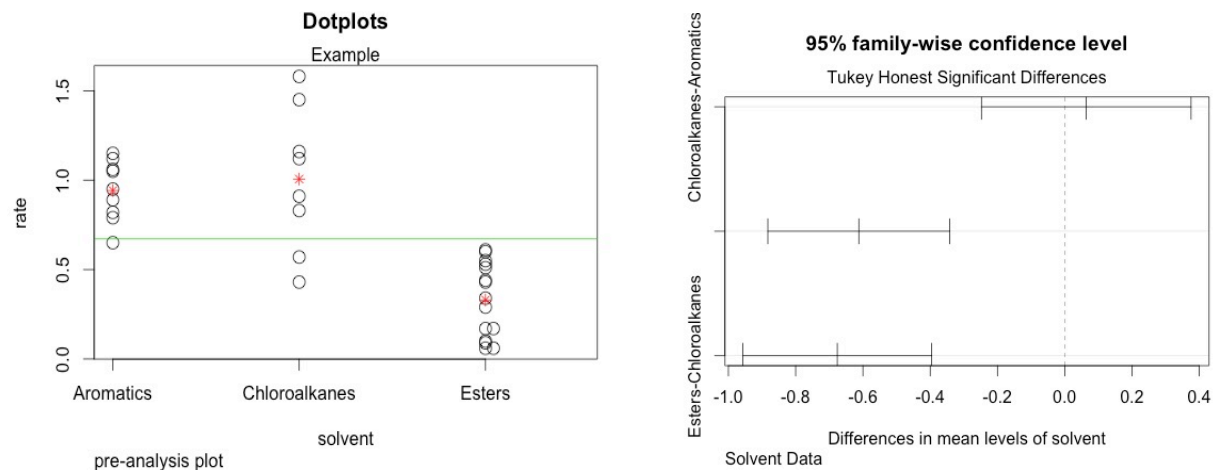
# Almost all conditions hold true
source("http://www.stat.ufl.edu/~athienit/check.R")
check(m1)

#### Obtain Tukey's Comparisons among levels of treatment
alloy.Tukey=TukeyHSD(m1, "solvent")
print(alloy.Tukey)
plot(alloy.Tukey, sub="Solvent Data", adj=0)
mtext("Tukey Honest Significant Differences",side=3,line=0.5)
```

Outputs :

	diff	lwr	upr	p
Chloroalkanes-Aromatics	0.06402778	-0.2475781	0.3756337	0.868314
Esters-Aromatics	-0.61222222	-0.8826095	-0.341835	0.0000143
Esters-Chloroalkanes	-0.67625	-0.9570006	-0.3954994	0.0000054

Note that 0 is in the Confidence Interval of only one of the pairs – Chloroalkanes-Aromatics which means there is not much variation between their means. There is difference between the following pairs 1. Esters-Aromatics and 2. Esters-Chloroalkanes and the difference values are shaded in the above table. Plots are added below.



Q3.

Create data frame

```
machines_data=data.frame(time=c(42.5, 39.3, 39.6, 39.9, 42.9, 43.6, 39.8, 40.1, 40.5, 42.3, 42.5, 43.1,40.2, 40.5,
41.3, 43.4, 44.9, 45.1, 41.3, 42.2, 43.5, 44.2, 45.9, 42.3),
machine=factor(rep(c("1","2","3","4"),each=6)), operator=factor(c("1","2","3","4","5","6")))
```

```
fmachine=factor(machines_data$machine, levels=1:4)
```

```
levels(fmachine)=c("Machine1", "Machine2", "Machine3", "Machine4")
```

```
machines=data.frame(time=machines_data$time, fmachine, operator=factor(machines_data$operator))
```

```
attach(machines)
```

```
machines
```

create easy to view table

```
table=xtabs(time~operator+fmachine);table
round(addmargins(table,c(1,2),FUN=mean),2)
```

Fit the ANOVA for the RBD with subject and interacting agent as independent variables

```
machines.rbd=aov(time~fmachine+operator)
anova(machines.rbd)
```

```
machines.Tukey=TukeyHSD(machines.rbd,"fmachine",conf.level=0.90)
```

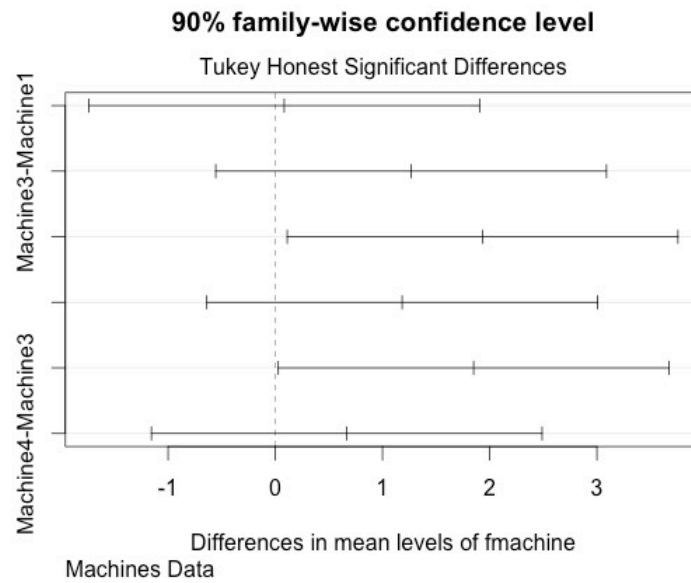
```
print(machines.Tukey)
```

```
#windows(width=5,height=5,pointsize=10)
```

```
plot(machines.Tukey, sub="Machines Data", adj=0)
```

```
mtext("Tukey Honest Significant Differences",side=3,line=0.5)
```

	diff	lwr	upr	p
Machine2-Machine1	0.08333333	-1.73885771	1.905524	0.9994381
Machine3-Machine1	1.26666667	-0.55552438	3.088858	0.3386299
Machine4-Machine1	1.93333333	0.11114229	3.755524	0.0760099
Machine3-Machine2	1.18333333	-0.63885771	3.005524	0.3947312
Machine4-Machine2	1.85	0.02780895	3.672191	0.0934325
Machine4-Machine3	0.66666667	-1.15552438	2.488858	0.7968901



Using $\alpha = 0.10$, we see that 0 is in the confidence interval of 4 of the pairs, which seems to suggest that they might have similar performance rate at some point. The differences in the other two pairs is highlighted in the table above.

	Df	Sum Sq	Mean Sq	F Value	Pr(>F)	
fmachine	3	15.925	5.3082	3.3388	0.047904	*
operator	5	42.087	8.4174	5.2944	0.005328	**
Residuals	15	23.848	1.5899			

The p-value for the operator row suggests that it has a significant effect on the performance rate analysis.