DSA 8020 R Session 8: CRD

Whitney

March 03, 2021

Contents

CRD . . . 1
Create the data set . . . 1
Summary statistics by treatments . . . 2
Plot the data . . . 2
ANOVA table . . . 3
Multiple Comparisons . . . 3
Model Assumptions . . . 4
Read the data into R . . . 4
Convert variable COLOR to a factor . . . 4
Model Fitting and Residuals . . . 5
Assess Equal Variance . . . 6
Plot r_{ij} versus $\hat{y}_{i\cdot}$ and treatments . . . 6
Assess Normality . . . 7
Assess Indepdence . . . 9
Fit a model with correlated AR(1) error . . . 9

CRD

Create the data set

```
r1 <- c(11.8, 10.8, 10.4, 11.5, 11.2) - 2
r2 <- c(9.2, 7.9, 8.5, 8.1, 7.5) - 1
r3 <- c(5.8, 5.6, 4.9, 6.3, 6.2) + 1
r4 <- c(2.8, 3.2, 3.4, 3.9, 2.6) + 2
times <- c(r1, r2, r3, r4)
trt <- rep(1:4, each = 5)
dat <- data.frame(y = times, trt = as.factor(trt))
```

Summary statistics by treatments

```
(means <- tapply(dat$y, dat$trt, mean))
```

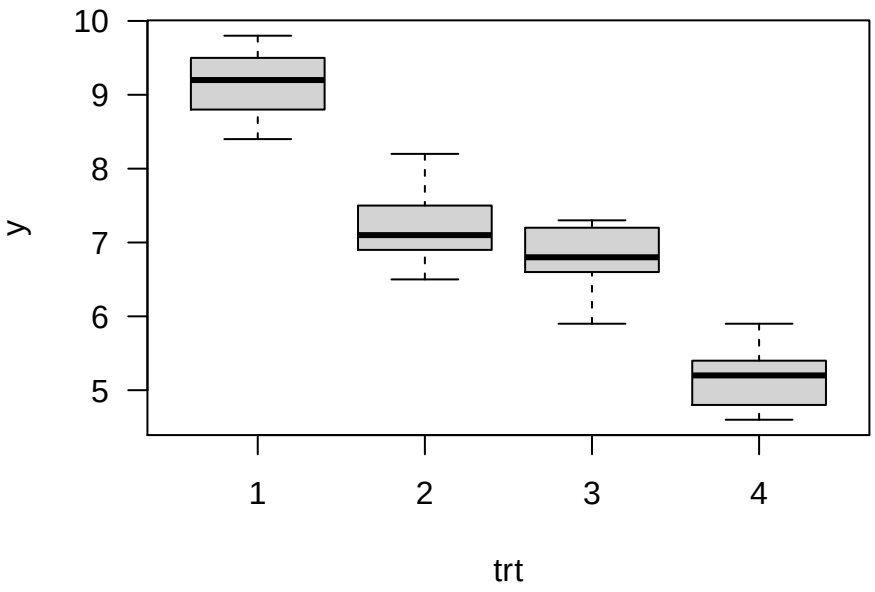
```
##    1    2    3    4
## 9.14 7.24 6.76 5.18
```

```
(vars <- tapply(dat$y, dat$trt, var))
```

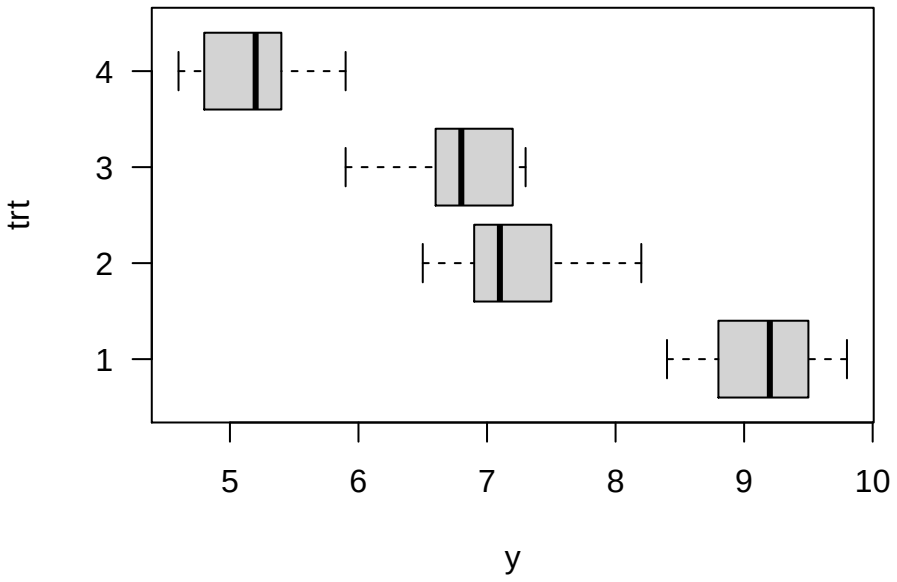
```
##     1     2     3     4
## 0.308 0.418 0.313 0.262
```

Plot the data

```
boxplot(y ~ trt, data = dat, las = 1)
```




```
boxplot(y ~ trt, data = dat, las = 1, horizontal = T)
```



ANOVA table

```
AOV <- aov(y ~ trt, data = dat)
summary(AOV)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## trt          3  39.91  13.303    40.9 9.92e-08 ***
## Residuals   16   5.20   0.325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Multiple Comparisons

```
# LSD
library(agricolae)
LSD_bon <- LSD.test(AOV ,"trt", p.adj = "bonferroni")
LSD_bon$groups
```

```
##      y groups
## 1 9.14      a
## 2 7.24      b
## 3 6.76      b
## 4 5.18      c
```

```
# HSD
HSD <- TukeyHSD(AOV, conf.level = 0.95)
HSD$trt
```

```
##      diff       lwr       upr        p adj
## 2-1 -1.90 -2.931952 -0.868048 4.024593e-04
```

```
## 3-1 -2.38 -3.411952 -1.348048 3.310735e-05
## 4-1 -3.96 -4.991952 -2.928048 4.112087e-08
## 3-2 -0.48 -1.511952  0.551952 5.577630e-01
## 4-2 -2.06 -3.091952 -1.028048 1.708962e-04
## 4-3 -1.58 -2.611952 -0.548048 2.363679e-03
```

Model Assumptions

Read the data into R

```
balloon <- read.csv("cr_assumptions.csv", header = T)
head(balloon)
```

```
##   ORDER COLOR TIME
## 1     1     1 22.0
## 2     2     3 24.6
## 3     3     1 20.3
## 4     4     4 19.8
## 5     5     3 24.3
## 6     6     2 22.2
```

```
summary(balloon)
```

```
##      ORDER           COLOR           TIME
##  Min.   : 1.00   Min.   :1.00   Min.   :14.00
##  1st Qu.: 8.75   1st Qu.:1.75   1st Qu.:17.40
##  Median :16.50   Median :2.50   Median :19.70
##  Mean   :16.50   Mean   :2.50   Mean   :20.24
##  3rd Qu.:24.25   3rd Qu.:3.25   3rd Qu.:22.60
##  Max.   :32.00   Max.   :4.00   Max.   :28.80
```

```
head(balloon, 10)
```

```
##    ORDER COLOR TIME
## 1      1     1 22.0
## 2      2     3 24.6
## 3      3     1 20.3
## 4      4     4 19.8
## 5      5     3 24.3
## 6      6     2 22.2
## 7      7     2 28.5
## 8      8     2 25.7
## 9      9     3 20.2
## 10    10     1 19.6
```

Convert variable COLOR to a factor

```r
attach(balloon)
colorf <- as.factor(COLOR)
colorf
```

```
##  [1] 1 3 1 4 3 2 2 2 3 1 2 4 4 4 3 1 2 1 4 3 1 4 4 2 2 4 2 3 3 1 1 3
## Levels: 1 2 3 4
```

Model Fitting and Residuals

```r
mod1 <- lm(TIME ~ colorf)
summary(mod1)
```

```
##
## Call:
## lm(formula = TIME ~ colorf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8750 -2.2500  0.0687  2.0531  6.2250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.337      1.162  15.778 1.83e-15 ***
## colorf2        4.237      1.644   2.578   0.0155 *
## colorf3        3.538      1.644   2.152   0.0401 *
## colorf4       -0.150      1.644  -0.091   0.9279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.287 on 28 degrees of freedom
## Multiple R-squared:  0.2967, Adjusted R-squared:  0.2214
## F-statistic: 3.938 on 3 and 28 DF,  p-value: 0.01836
```

```r
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: TIME
##           Df Sum Sq Mean Sq F value  Pr(>F)
## colorf     3 127.66  42.554  3.9379 0.01836 *
## Residuals 28 302.58  10.806
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Residuals
r <- residuals(mod1)
s <- rstandard(mod1)
var(s)
```

```
## [1] 1.032258
```

```r
t <- rstudent(mod1)
```

Assess Equal Variance

```r
# Levene's test for equal variance
library(lawstat)
levene.test(TIME, colorf, location = "mean")
```

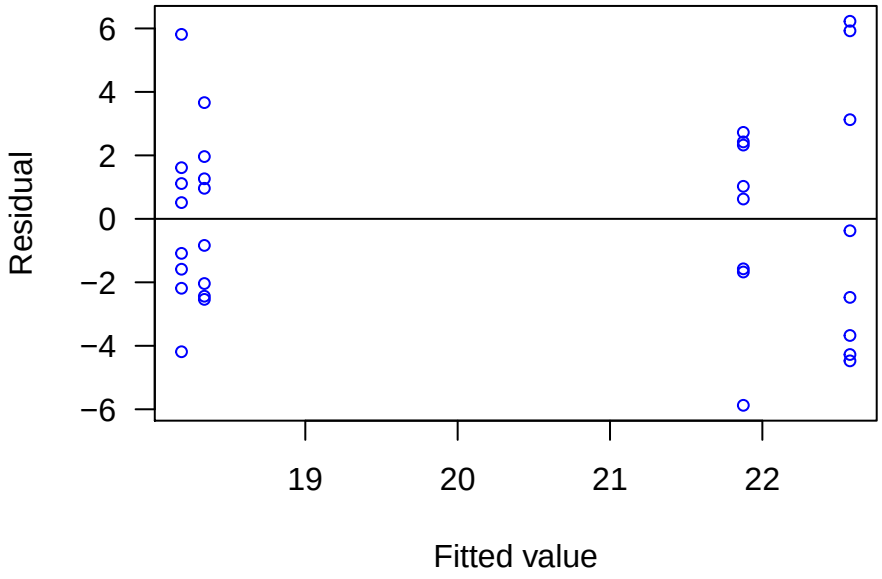
```
##
##  Classical Levene's test based on the absolute deviations from the mean
##  ( none not applied because the location is not set to median )
##
## data:  TIME
## Test Statistic = 2.1682, p-value = 0.1141
```

```r
# Brown-Forsythe test
levene.test(TIME, colorf, location = "median")
```

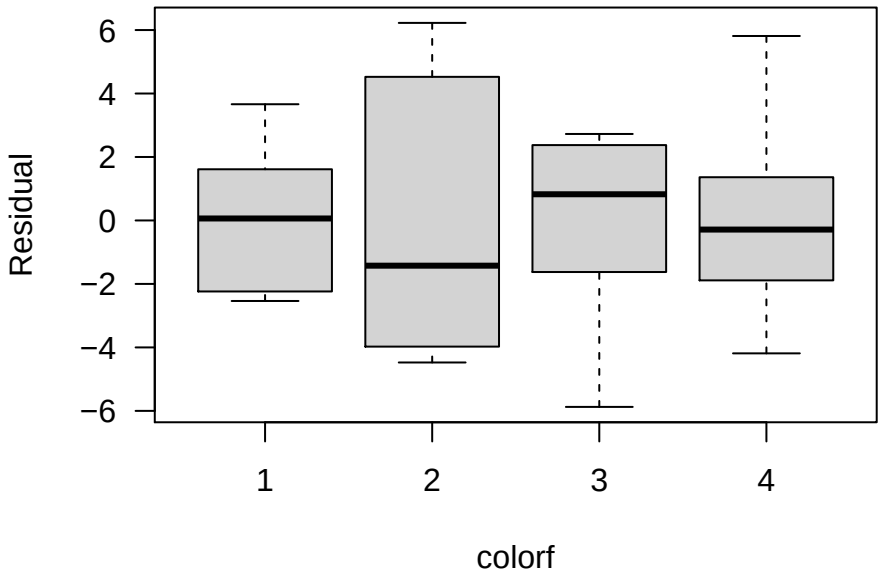
```
##
##  Modified robust Brown-Forsythe Levene-type test based on the absolute
##  deviations from the median
##
## data:  TIME
## Test Statistic = 1.3975, p-value = 0.2642
```

Plot r_{ij} versus $\hat{y}_{i\cdot}$ and treatments

```r
plot(mod1$fitted, mod1$resid, las = 1, xlab = "Fitted value", ylab = "Residual", cex = 0.75, col = "blu
abline(h = 0)
```

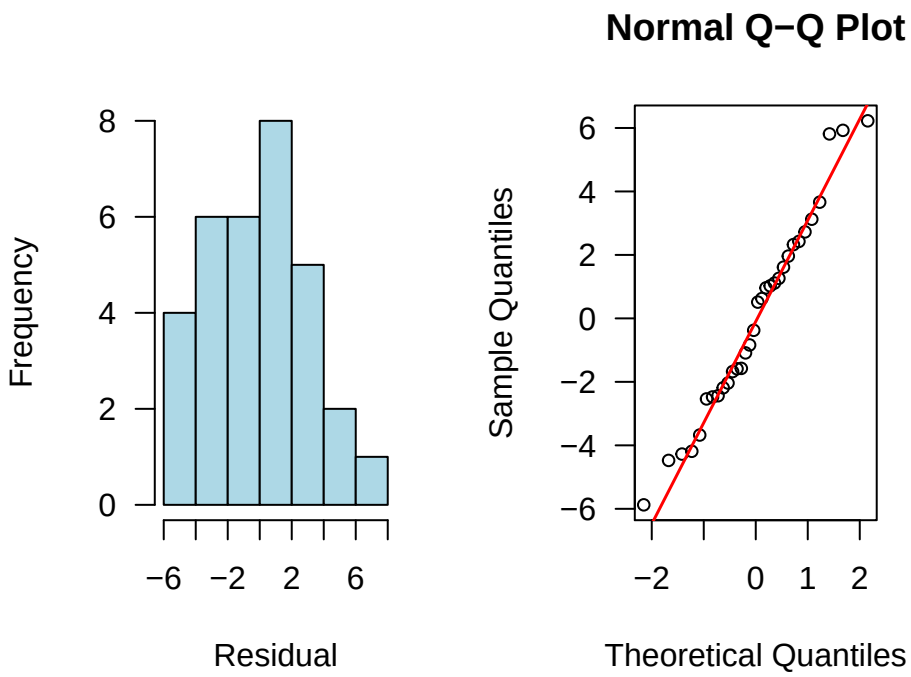


```
plot(mod1$resid ~ colorf, ylab = "Residual", las = 1)
```

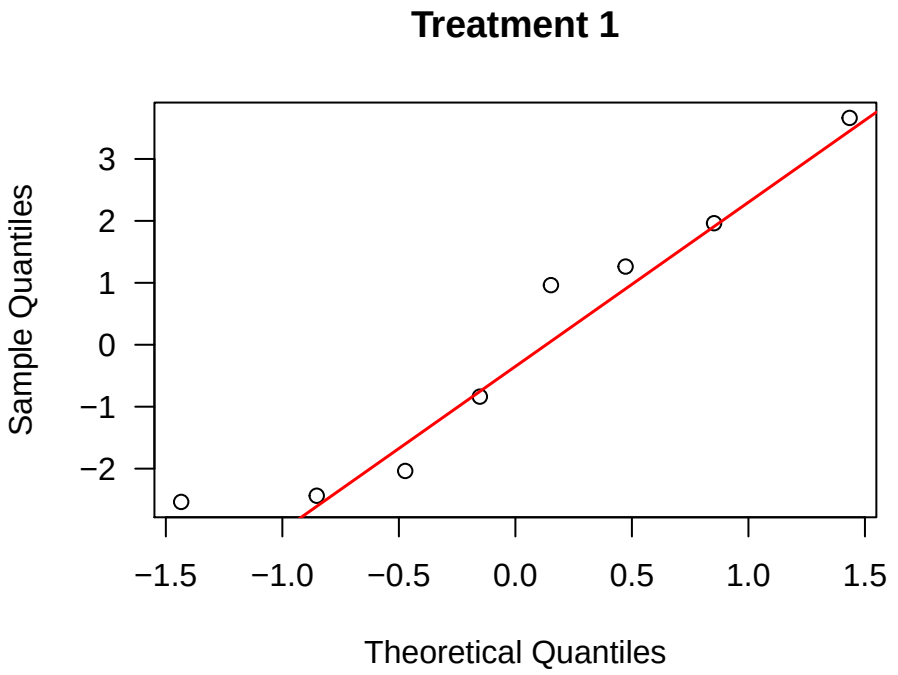


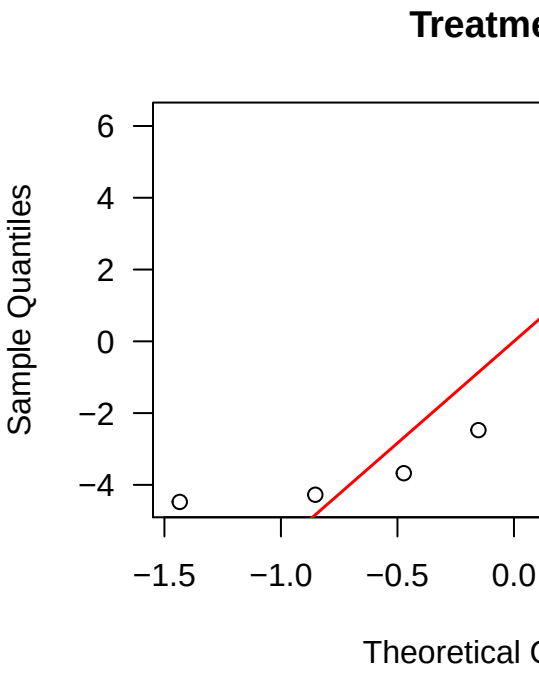
Assess Normality

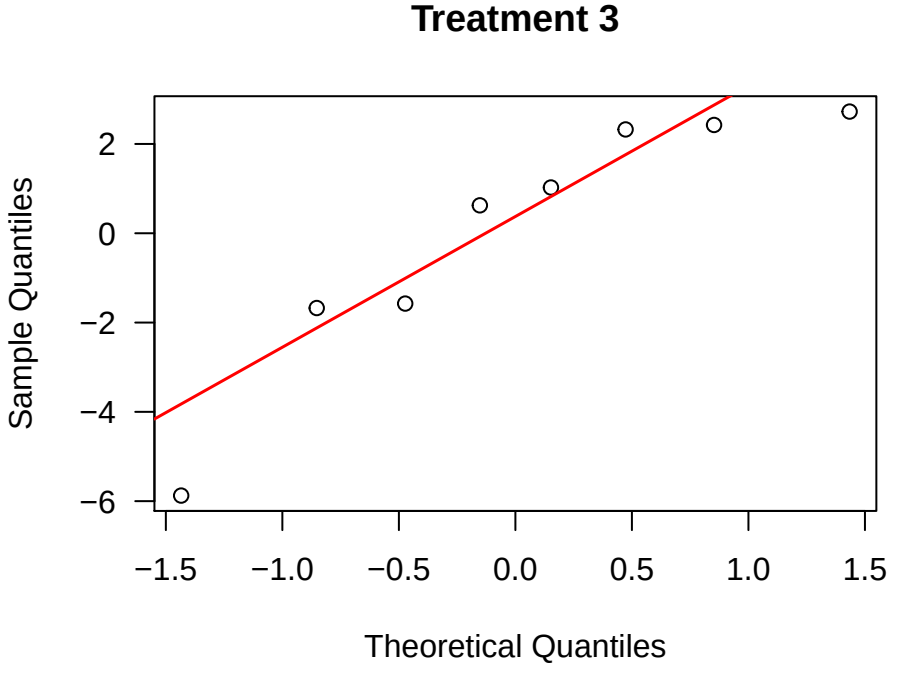
```
par(mfrow = c(1, 2), las = 1)
hist(mod1$resid, 8, main = "", xlab = "Residual", col = "lightblue")
qqnorm(mod1$resid, cex = 0.8)
qqline(mod1$resid, col = "red", lwd = 1.5)
```

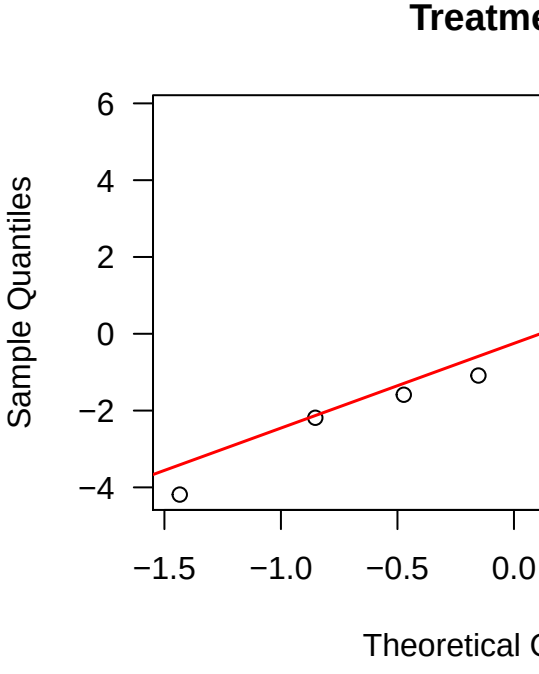


```r
new <- data.frame(colorf, mod1$resid)
trt <- paste("Treatment", 1:4)
for (i in 1:4){
  newc1 <- new[colorf == i,]
  qqnorm(newc1$mod1.resid, las = 1,
         main = trt[i])
  qqline(newc1$mod1.resid,
         col = "red", lwd = 1.5)
}
```

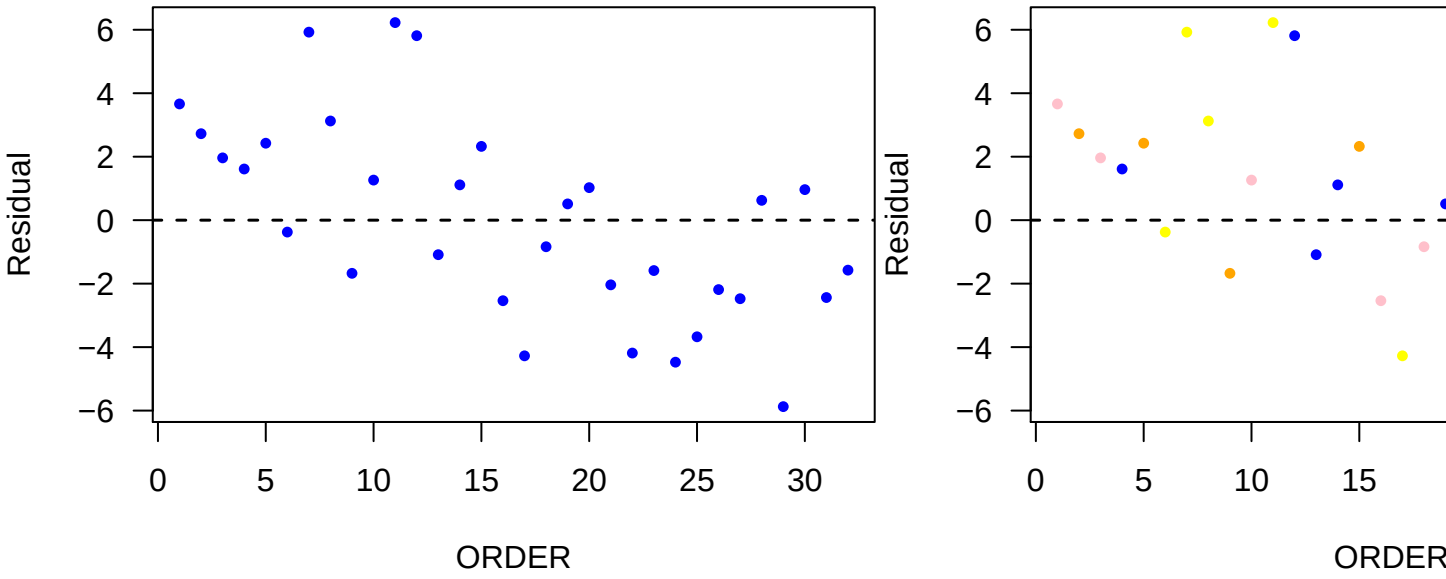








Assess Indepdence




```
# Durbin-Watson test
library(lmtest)
dwtest(TIME ~ colorf, data = balloon)
```

```
##
##  Durbin-Watson test
##
## data:  TIME ~ colorf
## DW = 1.1617, p-value = 0.006005
## alternative hypothesis: true autocorrelation is greater than 0
```

Fit a model with correlated AR(1) error

```
library(nlme)
mod2 <- gls(TIME ~ colorf, correlation = corARMA(p = 1, q = 0))
mod2
```

```
## Generalized least squares fit by REML
##   Model: TIME ~ colorf
##   Data: NULL
##   Log-restricted-likelihood: -74.42885
##
## Coefficients:
## (Intercept)     colorf2     colorf3     colorf4
##  18.5860865   3.7248742   3.4233901  -0.3578644
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
```

```
## 0.4285025
## Degrees of freedom: 32 total; 28 residual
## Residual standard error: 3.321057
```