

```
> #structure of MLR
> names(mlr)
[1] "coefficients" "residuals"      "effects"         "rank"
[5] "fitted.values" "assign"          "qr"              "df.residual"
[9] "x.levels"      "call"           "terms"           "model"
> mlrs <- summary(mlr)
```

```
> names(mlrs)
[1] "call"          "terms"          "residuals"      "coefficients"
[5] "aliases"       "sigma"          "df"             "r.squared"
[9] "adj.r.squared" "fstatistic"     "cov.unscaled"
```

### 3 Adequacy Checking

#### 3.1 From the fitted model

t-tests:

➤ $H_0: \beta_0 = 0$	1.821	0.0712	
➤ $H_0: \beta_1 = 0$	7.017	1.63e-10	***
➤ $H_0: \beta_2 = 0$	5.983	2.49e-08	***
➤ $H_0: \beta_3 = 0$	0.807	0.4213	
➤ $H_0: \beta_4 = 0$	-5.223	7.83e-07	***

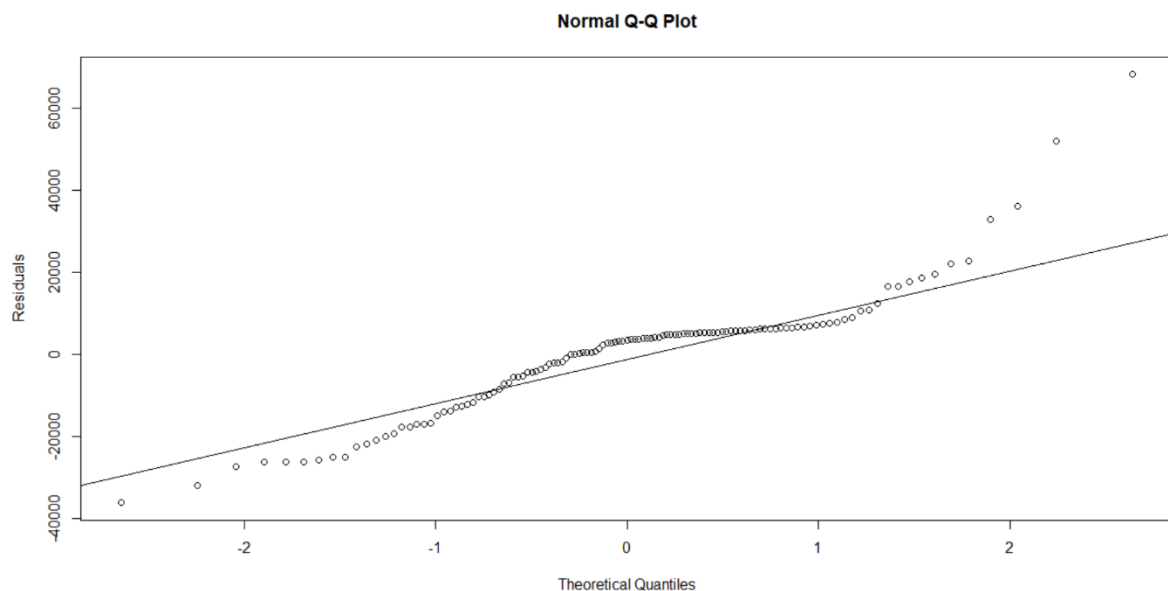
F-ratio:

F-statistic: 88.29 on 4 and 116 DF, p-value: < 2.2e-16

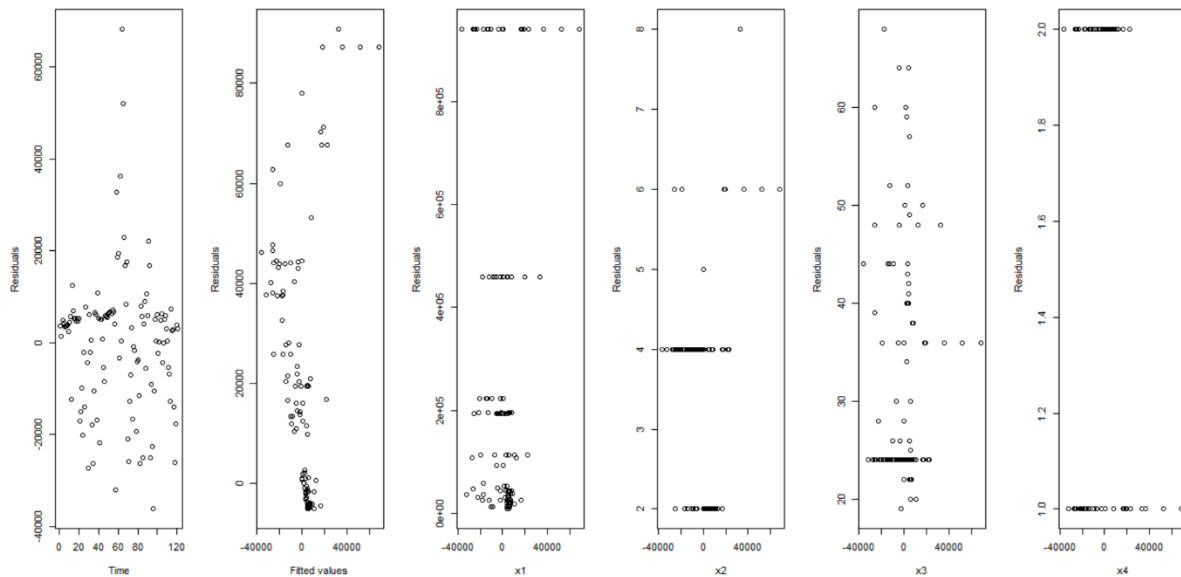
R<sup>2</sup> statistic:

Multiple R-squared: 0.7527, Adjusted R-squared: 0.7442

#### 3.2 From the viewpoint of residuals



From the normal qq-plot, the data points fall in a line in the middle but curve off at the ends. Thus, the data has more extreme values than expected if they came from a normal distribution.



From the residual plots, x1, x2, x3 and x4 displays horizontal bands of points. Hence, there are no clear patterns in the residuals that can be used as information.

### 3.3 Checking for sequential dependence

```
> dwtest(y ~ x1+x2+x3+x4, data=tm)
```

Durbin-Watson test

data:  $y \sim x1 + x2 + x3 + x4$

DW = 1.3137, p-value = 3.101e-05

alternative hypothesis: true autocorrelation is greater than 0

From the DW test, the DW value is closer to 0, thus the successive residuals are positively serially correlated.

## 4 F-test for reduced model and full model

### 4.1 Test for whether some coefficients are zeros

```
> summary(mlr1)

Call:
lm(formula = y ~ x1 + x2 + x4, data = tm)

Residuals:
    Min       1Q   Median       3Q      Max
-35593  -7883   4010   5770  68441

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.270e+04  1.146e+04   1.981   0.05 *
x1           3.356e-02  4.655e-03   7.211 5.93e-11 ***
x2           9.310e+03  1.517e+03   6.138 1.18e-08 ***
x4          -2.305e+04  4.460e+03  -5.168 9.85e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15270 on 117 degrees of freedom
Multiple R-squared:  0.7514, Adjusted R-squared:  0.745
F-statistic: 117.8 on 3 and 117 DF, p-value: < 2.2e-16
```

```
> mlr1 <- lm(y ~ x1+x2+x4,data=tm)
> anova(mlr1,mlr)
Analysis of Variance Table

Model 1: y ~ x1 + x2 + x4
Model 2: y ~ x1 + x2 + x3 + x4
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     117 2.7281e+10
2     116 2.7128e+10   1 152302593 0.6512 0.4213
```

From the results of MLR, the predictor  $x_3$  is not significant, which may be equal to zero. Thus we want to test the following hypothesis:  $H_0: \beta_3 = 0$

From the results of the ANOVA table above, we cannot reject the null hypothesis at the level of 0.1.

### 4.2 Test whether coefficients are constant

```
> mlr3 <- lm(y ~ x1+x2+offset(100.3*x3)+x4,data=tm)
> summary(mlr3)

Call:
lm(formula = y ~ x1 + x2 + offset(100.3 * x3) + x4, data = tm)

Residuals:
    Min       1Q   Median       3Q      Max
-36263  -8501   3493   6018  68317

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.118e+04  1.143e+04   1.853   0.0664 .
x1           3.303e-02  4.642e-03   7.117 9.56e-11 ***
x2           9.158e+03  1.513e+03   6.055 1.75e-08 ***
x4          -2.361e+04  4.448e+03  -5.308 5.32e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15230 on 117 degrees of freedom
Multiple R-squared:  0.7527, Adjusted R-squared:  0.7464
```

F-statistic: 118.7 on 3 and 117 DF, p-value: < 2.2e-16

```
> anova(mlr3,mlr)
Analysis of Variance Table

Model 1: y ~ x1 + x2 + offset(100.3 * x3) + x4
Model 2: y ~ x1 + x2 + x3 + x4
  Res.Df    RSS Df Sum of Sq  F Pr(>F)
1     117 2.7128e+10
2     116 2.7128e+10   1      1.831   0 0.9999
```

From the results of the MLR, we get  $\hat{\beta}_3 = 100.3$

Thus we may want  $H_0: \beta_3 = 100.3$

We cannot reject the null hypothesis at the level of 0.1.

## **5 Prediction**

```
> #prediction x1=50000 x2=3 x3=60, x4=2
> con <- (c(1,50000,3,60,2))
> lhat <- sum(con*coef(mlr))
> lhat
[1] 9106.94
> t05 <- qt(0.975,116)
> bm <- t05*mlr3$sigma*sqrt(con%*%mlr3$cov.unscaled%*%con)
> c(lhat-bm,lhat+bm)
[1] 1045.888 17167.992
> c3 <- 1
> bm <- t05*mlr3$sigma*sqrt(con%*%mlr3$cov.unscaled%*%con+c3)
> c(lhat-bm,lhat+bm)
[1] -22236.34 40450.22
> con <- data.frame(x1=50000,x2=3,x3=60,x4=2)
> predict(mlr,con,interval='confidence',level=0.95)
      fit      lwr      upr
1 9106.94 1045.888 17167.99
> predict(mlr,con,interval='prediction',level=0.95)
      fit      lwr      upr
1 9106.94 -22236.34 40450.22
```

## **6 Complete R program**

```
# Graphical display of the observed data.
aadt_raw <- read.table('c:/aadt.txt',header=FALSE)

tm <-
data.frame(y=aadt_raw$V1,x1=aadt_raw$V2,x2=aadt_raw$V3,x3=aadt_raw$V4,x4=a
adt_raw$V5)

plot(tm)


#Fit a MLR
mlr <- lm(y~x1+x2+x3+x4,data=tm)
summary(mlr)
#structure of MLR
names(mlr)
mlrs <- summary(mlr)
names(mlrs)


#Normality checking
qqnorm(residuals(mlr),ylab='Residuals')
qqline(residuals(mlr))


#Draw some plots of residuals
par(mfrow=c(1,6))
plot(residuals(mlr),ylab='Residuals',xlab='Time')
plot(residuals(mlr),fitted(mlr),ylab='Residuals',xlab='Fitted values')
plot(residuals(mlr),tm$x1,ylab='Residuals',xlab='x1')
plot(residuals(mlr),tm$x2,ylab='Residuals',xlab='x2')
plot(residuals(mlr),tm$x3,ylab='Residuals',xlab='x3')
plot(residuals(mlr),tm$x4,ylab='Residuals',xlab='x4')
par(mfrow=c(1,1))


#Durbin-Watson tests
library(lmtest)
dwtest(y ~ x1+x2+x3+x4, data=tm)


#Some F-tests
#test x3 predictor = 0
mlr1 <- lm(y ~ x1+x2+x4,data=tm)
summary(mlr1)
```

```

anova(mlr1,mlr)
#test if coefficient of x3 is constant
mlr3 <- lm(y ~ x1+x2+offset(100.3*x3)+x4,data=tm)
summary(mlr3)
anova(mlr3,mlr)

#prediction x1=50000 x2=3 x3=60, x4=2
con <- (c(1,50000,3,60,2))
lhat <- sum(con*coef(mlr))
lhat
t05 <- qt(0.975,116)
bm <- t05*mlrs$sigma*sqrt(con**mlrs$cov.unscaled**con)
c(lhat-bm,lhat+bm)
c3 <- 1
bm <- t05*mlrs$sigma*sqrt(con**mlrs$cov.unscaled**con+c3)
c(lhat-bm,lhat+bm)
con <- data.frame(x1=50000,x2=3,x3=60,x4=2)
predict(mlr,con,interval='confidence',level=0.95)
predict(mlr,con,interval='prediction',level=0.95)

```