Statistical programming in R (part 1)

0. Summary statistics Calculate mean, median, variance, standard deviation, quantile

```
library(Stat2Data)
data(BirdNest)
mean(BirdNest$Totcare, na.rm = T)
```

```
## [1] 27.73494
```

```
median(BirdNest$Totcare, na.rm = T)
```

```
## [1] 27.5
```

```
var(BirdNest$Totcare, na.rm = T)
```

```
## [1] 23.3862
```

```
sd(BirdNest$Totcare, na.rm = T)
```

```
## [1] 4.835928
```

```
quantile(BirdNest$Totcare, probs = seq(0,
    1, 0.25), na.rm = T)
```

```
##   0%  25%  50%  75% 100%
## 19.0 23.5 27.5 31.0 37.5
```

```
IQR(BirdNest$Totcare, na.rm = T)
```

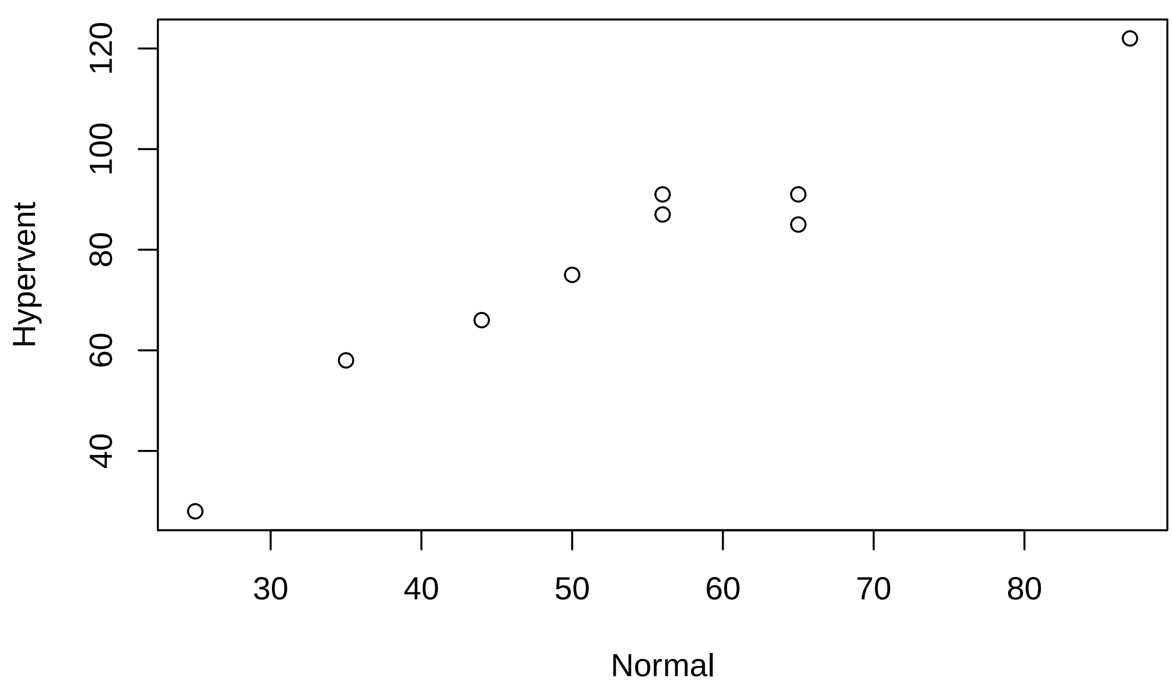
```
## [1] 7.5
```

1. Pearson Correlation Pearson correlation is a statistic that measures linear correlation between two variables, given the assumption that the sample pairs are independent and follow a bivariate normal distribution.

```
# Nine students held their
# breath, once after breathing
# normally and relaxing for one
# minute, and once after
# hyperventilating for one
# minute. The table indicates
# how long (in sec) they were
```

```r
# able to hold their breath. Is
# there an association between
# the two variables?

Normal <- c(56, 56, 65, 65, 50,
    25, 87, 44, 35)
Hypervent <- c(87, 91, 85, 91,
    75, 28, 122, 66, 58)
plot(Normal, Hypervent)
```




```r
cor(Normal, Hypervent, method = "pearson")
```

```
## [1] 0.9661943
```

```r
cor.test(Normal, Hypervent, method = "pearson")
```

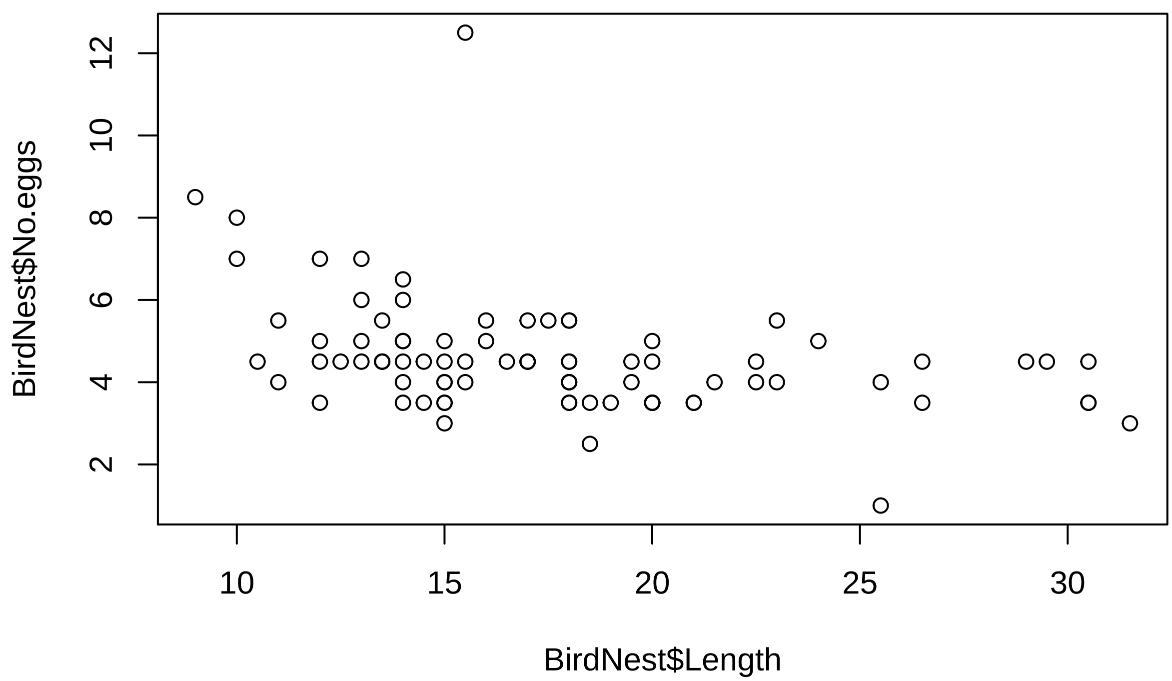
```
##
##  Pearson's product-moment correlation
##
## data:  Normal and Hypervent
## t = 9.9153, df = 7, p-value = 2.263e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8430028 0.9930835
```

```
## sample estimates:
##       cor
## 0.9661943
```

source

2. Spearman's rank correlation coefficient Spearman's rank 's correlation coefficient is a nonparametric measure of rank correlation (statistical dependence between the rankings of two variables). It assesses how well the relationship between two variables can be described using a monotonic function. (Wiki)

```r
plot(BirdNest$Length, BirdNest$No.eggs)
```




```r
cor(BirdNest$Length, BirdNest$No.eggs,
    method = "spearman")
```

```
## [1] -0.4481381
```

```r
cor.test(BirdNest$Length, BirdNest$No.eggs,
    method = "spearman", exact = T)
```

```
## Warning in cor.test.default(BirdNest$Length, BirdNest$No.eggs, method =
## "spearman", : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  BirdNest$Length and BirdNest$No.eggs
## S = 143033, p-value = 1.914e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4481381
```

- Read about Kendall Tau coefficient

3. Cramer's V Cramer's V is a measure of association between two nominal variables, returns a value between 0 and 1.

Recall that in the BirdNest data, egg color encoded as (0=plain/solid or 1=speckled/spotted) and closed encoded as 1=closed nest (pendant, spherical, cavity, crevice, burrow) or 0=open nest (saucer, cup).

```
# install.packages('rcompanion')
library(rcompanion)
cramerV(BirdNest$Closed., BirdNest$Color)
```

```
## Cramer V
##   0.0342
```

```
cramerV(BirdNest$Closed., BirdNest$Location)
```

```
## Cramer V
##   0.5842
```

```
cramerV(BirdNest$Closed., BirdNest$Nesttype)
```

```
## Cramer V
##        1
```

4. Compare the means of two groups / multiple groups Two sample t-test is a parametric test for comparing the means of two groups. The null hypothesis is the mean of two groups are equal. Alternative hypothesis is that they are not equal. Significance level is set to 0.05. The process of perform a t-test in R could be found at source.

A useful source to look up statistical analysis about their assumption, interpretation and how to perform in different software is: http://rcompanion.org/rcompanion/b_07.html You may check out more about paired or unpaired t-test, comparing the means of more than two groups using Analysis of Variance (ANOVA) or nonparametric test for comparing means of two groups (Mann-Whitney U test) etc.

- create data for t-test

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
t_test_data <- BirdNest %>% filter(Location %in%
    c("ground", "decid"))
group_by(t_test_data, Location) %>%
    summarise(count = n(), mean = mean(Totcare,
        na.rm = TRUE), sd = sd(Totcare,
        na.rm = TRUE))
```

```
## # A tibble: 2 x 4
##   Location count  mean    sd
##   <fct>    <int> <dbl> <dbl>
## 1 decid       24  29.3  3.36
## 2 ground      19  23.6  2.92
```

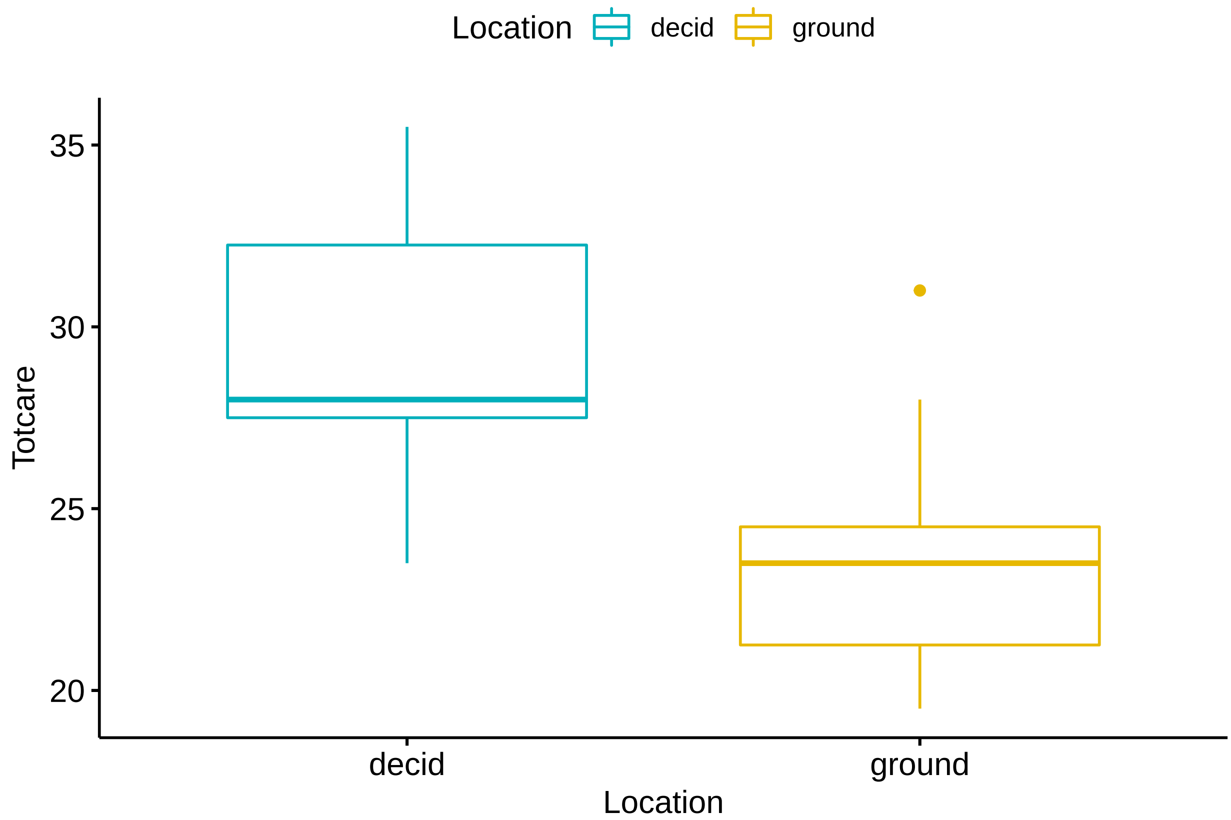
- visualize data

```r
# install.packages('ggpubr')
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```r
ggboxplot(t_test_data, x = "Location",
    y = "Totcare", color = "Location",
    palette = c("#00AFBB", "#E7B800"),
    ylab = "Totcare", xlab = "Location")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```
# Shapiro-Wilk normality test
# for decid's Totcare
with(t_test_data, shapiro.test(Totcare[Location ==
    "decid"]))  # p = 0.5101
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Totcare[Location == "decid"]
## W = 0.96224, p-value = 0.5101
```

```
# Shapiro-Wilk normality test
# for ground's Totcare
with(t_test_data, shapiro.test(Totcare[Location ==
    "ground"]))  # p = 0.1502
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Totcare[Location == "ground"]
## W = 0.92665, p-value = 0.1502
```

- check equal variances

```r
# F-test to test for
# homogeneity in variances
res.ftest <- var.test(Totcare ~
    Location, data = t_test_data)  # p = 0.5472
res.ftest
```

```
##
##  F test to compare two variances
##
## data:  Totcare by Location
## F = 1.3266, num df = 22, denom df = 18, p-value = 0.5472
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5246666 3.2186930
## sample estimates:
## ratio of variances
##           1.326626
```

- perform t-test

```r
# t_test
res <- t.test(Totcare ~ Location,
    data = t_test_data, var.equal = TRUE)
res  # p = 7.138e-07
```

```
##
##  Two Sample t-test
##
## data:  Totcare by Location
## t = 5.8726, df = 40, p-value = 7.138e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3.786512 7.760398
## sample estimates:
##  mean in group decid mean in group ground
##             29.32609             23.55263
```

```r
res$conf.int
```

```
## [1] 3.786512 7.760398
## attr(,"conf.level")
## [1] 0.95
```

- Mann-whitney test

```r
# Mann-Whitney test for
# (non-parametric)
wilcox.test(Totcare ~ Location,
    data = t_test_data, exact = F)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Totcare by Location
## W = 395.5, p-value = 7.61e-06
## alternative hypothesis: true location shift is not equal to 0
```

- ANOVA

```
# ANOVA
aov.data <- BirdNest %>% filter(Location %in%
    c("ground", "shrub", "decid"))
res.aov <- aov(Totcare ~ Location,
    data = aov.data)
summary(res.aov)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Location     2  354.6  177.31   14.82 6.82e-06 ***
## Residuals   56  669.9   11.96
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

- nonparametric Kruskal-Wallis test

```
# Kruskal-Wallis test
# (non-parametric)
kruskal.test(Totcare ~ Location,
    data = aov.data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Totcare by Location
## Kruskal-Wallis chi-squared = 22.959, df = 2, p-value = 1.034e-05
```

Nonparametric test based on resampling: permutation test
Read more about it here

5. Chi-square test Chi-square test used for testing independence by evaluating the closeness between observed and expected frequencies.
Assumption: large samples and independence of individual observation.

```
library(rcompanion)
table <- data.frame(smoker = c("Yes",
    "No", "Yes", "No"), lung_cancer = c("Cases",
    "Cases", "Control", "Control"),
    count = c(688, 21, 650, 59))

ctable <- xtabs(count ~ smoker +
    lung_cancer, data = table)
ctable
```

```
##       lung_cancer
## smoker Cases Control
##    No     21      59
##    Yes   688     650
```

```r
chisq.test(ctable, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ctable
## X-squared = 19.129, df = 1, p-value = 1.222e-05
```

6. Fisher exact test Fisher's exact test can be used for test of independence when n is small. Assumption: independence of individual observation and fixed totals. (the row and column totals are fixed, or "conditioned.") When row or column totals are unconditioned, makes this test less powerful.

```r
tea <- matrix(c(3, 1, 1, 3), ncol = 2,
    byrow = TRUE)
dimnames(tea) <- list(PouringFirst = c("Milk",
    "Tea"), GuessPouredFirst = c("Milk",
    "Tea"))
tea
```

```
##             GuessPouredFirst
## PouringFirst Milk Tea
##         Milk    3   1
##         Tea     1   3
```

```r
fisher.test(tea, alternative = "greater")  # set alternative to 'greater', 'less', 'two.sided'
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tea
## p-value = 0.2429
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  0.3135693       Inf
## sample estimates:
## odds ratio
##   6.408309
```

7. Linear regression Linear regression explain the relationship between continuous response and predictors. A linear regression has an equation of the form $Y = X\beta + \epsilon$, where X is the explanatory matrix with the first columns of all 1s (intercept) and Y is the dependent variable. The standard multiple (linear) regression equation with p predictor variables and N observations $y_i = b_0 + b_1x_{i1} + b_2x_{i2} + \ldots + b_px_{ip} + \epsilon_i$, where i=1, ..., N.
The random errors ϵ are assumed to be independently and identically normally distributed.

Example: Hourly carbon monoxide (CO) averages were recorded on summer weekdays at a measurement station in Los Angeles. The data could be downloaded from http://www.statsci.org/data/general/cofreewy.txt There are four variables, which represents for:

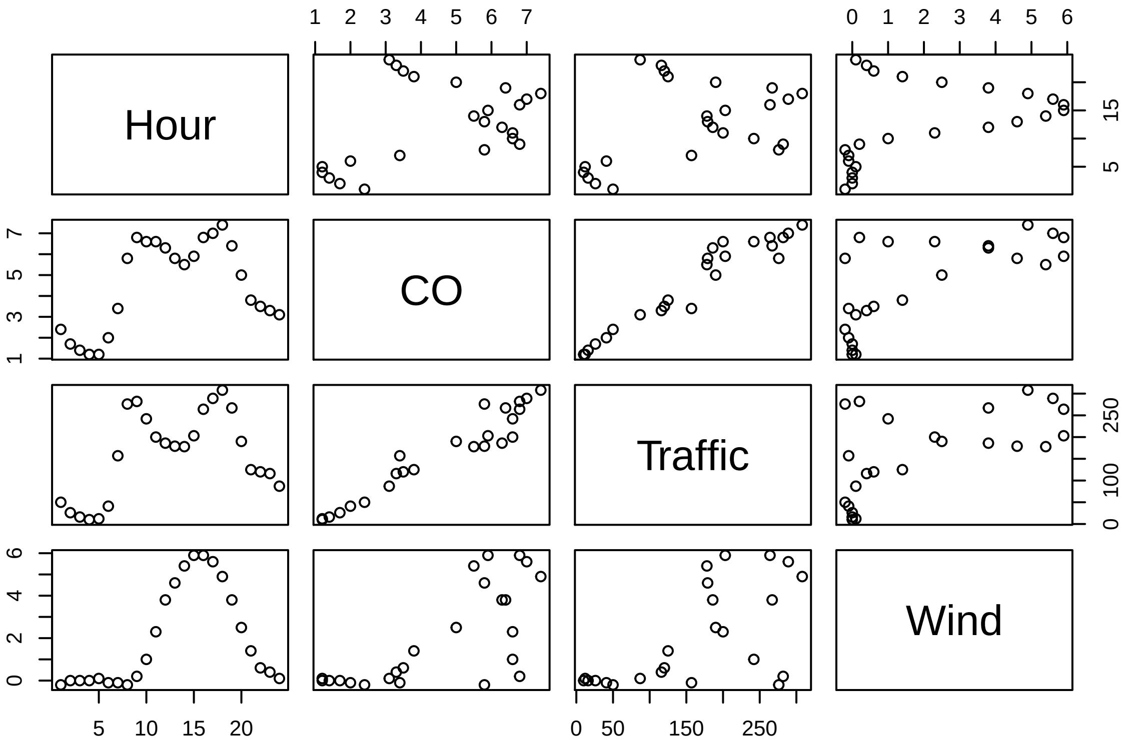
Variable	Description
Hour	hour of the day, from midnight to midnight
CO	average summer weekday CO concentration (parts per million)
Traffic	average weekday traffic density (traffic count/traffic speed)
Wind	average perpendicular wind-speed component, wind speed x cos(wind direction - 235 degrees)

Use CO as dependent variable, the other three variables as predictor to build a linear regression model.

```
lm_data <- read.table("http://www.statsci.org/data/general/cofreewy.txt",
    header = T)
head(lm_data)
```

```
##   Hour  CO Traffic Wind
## 1    1 2.4      50 -0.2
## 2    2 1.7      26  0.0
## 3    3 1.4      16  0.0
## 4    4 1.2      10  0.0
## 5    5 1.2      12  0.1
## 6    6 2.0      41 -0.1
```

```
plot(lm_data)
```



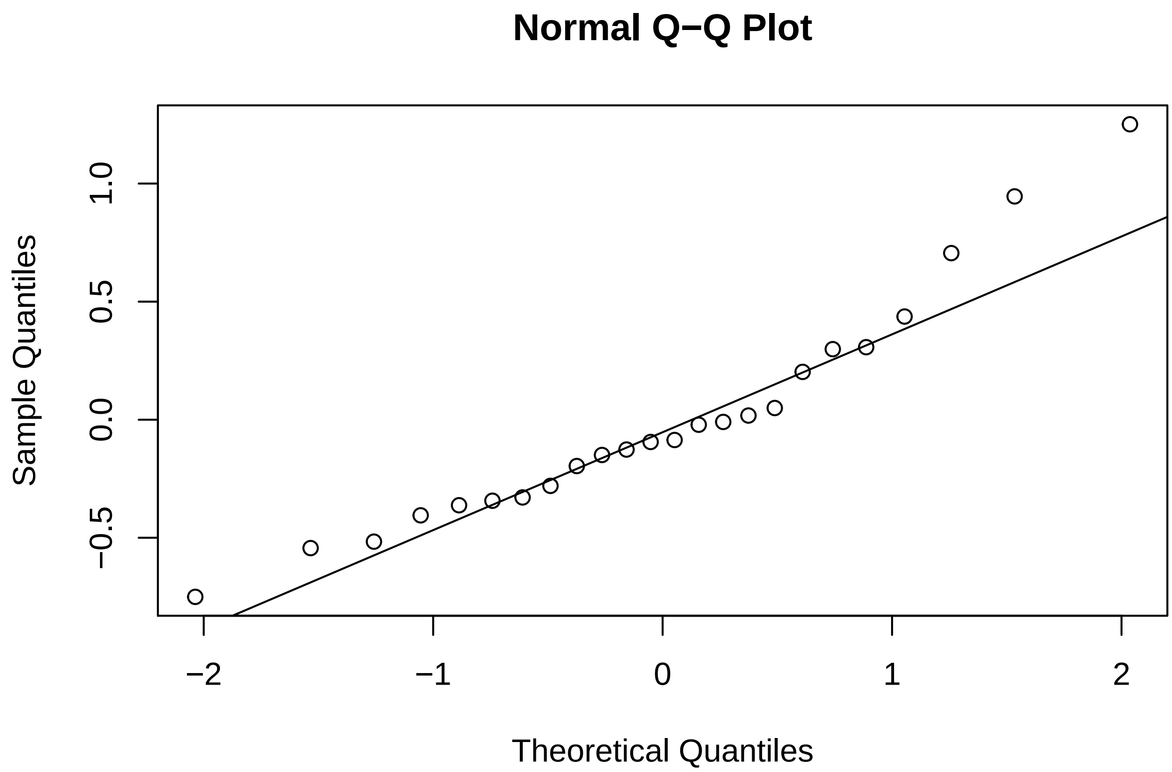
```r
lm.fit <- lm(CO ~ ., lm_data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = CO ~ ., data = lm_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75030 -0.33275 -0.09021  0.22653  1.25112
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.318967   0.242522   5.439 2.53e-05 ***
## Hour        -0.005689   0.017066  -0.333  0.74233
## Traffic      0.018402   0.001413  13.026 3.15e-11 ***
## Wind         0.179189   0.059517   3.011  0.00691 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5096 on 20 degrees of freedom
## Multiple R-squared:  0.9498, Adjusted R-squared:  0.9423
## F-statistic: 126.1 on 3 and 20 DF,  p-value: 3.682e-13
```

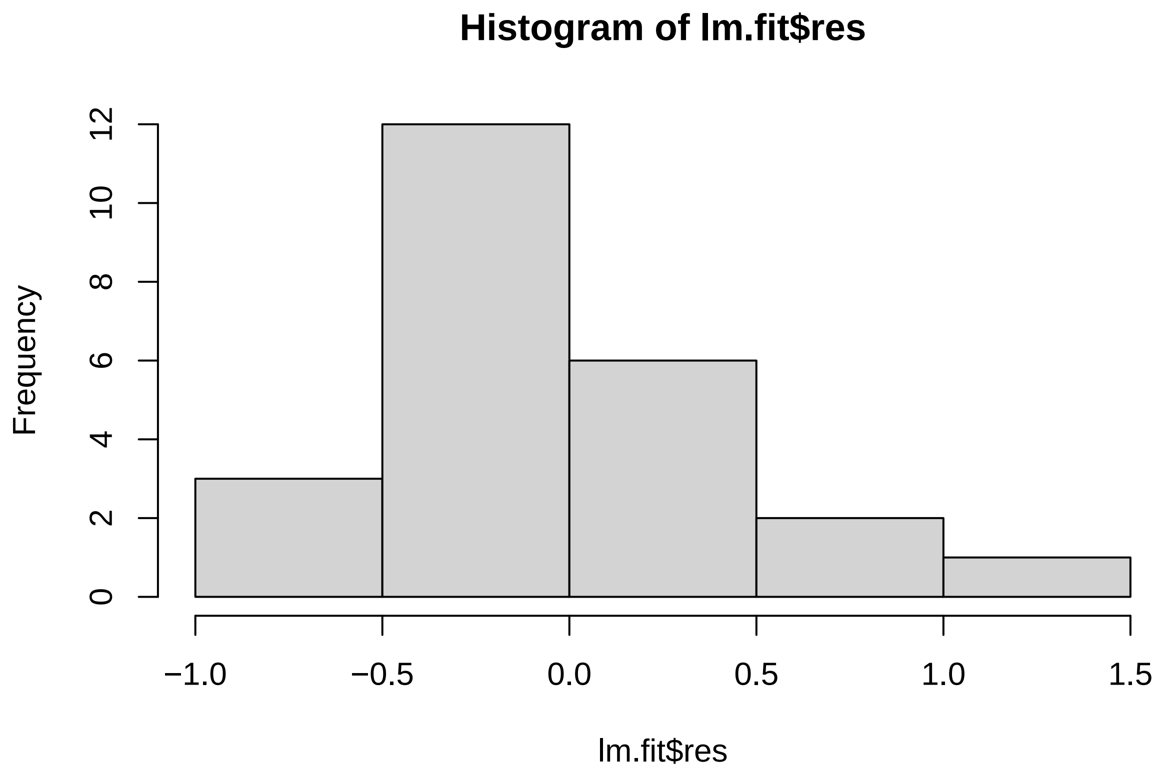
```r
shapiro.test(lm.fit$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm.fit$res
## W = 0.93027, p-value = 0.09885
```

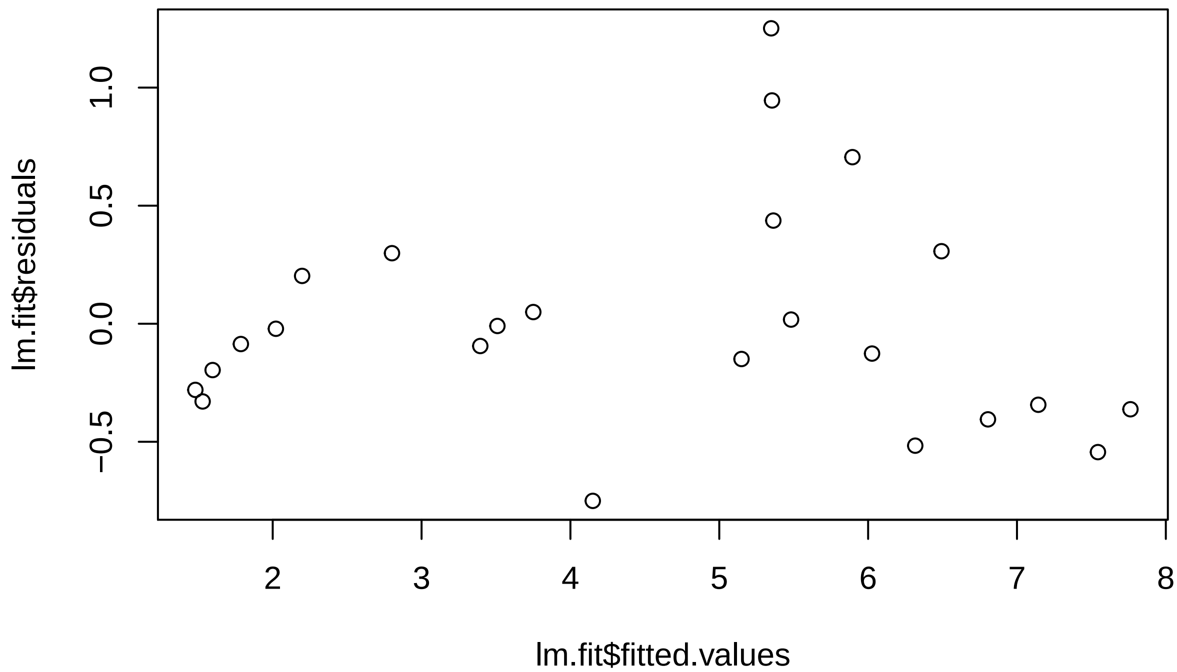
```r
qqnorm(lm.fit$res)
qqline(lm.fit$res)
```



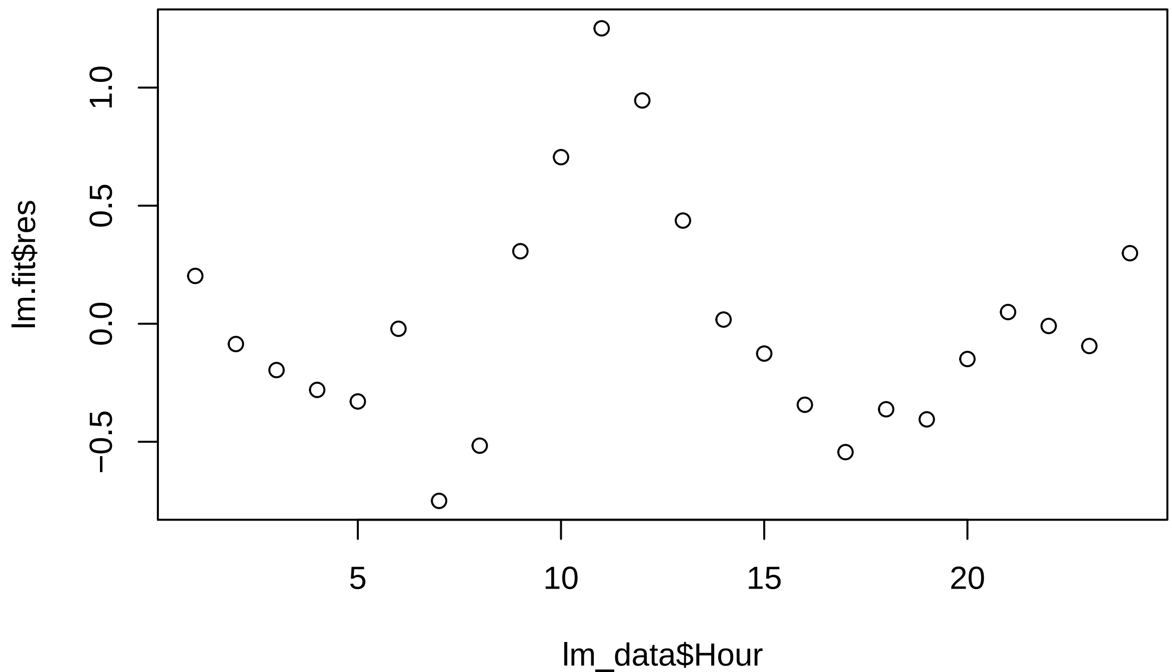

```
hist(lm.fit$res)   # residual
```




```r
plot(lm.fit$fitted.values, lm.fit$residuals)  # residual vs fitted value
```




```r
plot(lm_data$Hour, lm.fit$res)  # residuals vs time order
```



- stepwise model

```
lm.step <- step(lm.fit, direction = "backward")
```

```
## Start:  AIC=-28.73
## CO ~ Hour + Traffic + Wind
##
##           Df Sum of Sq    RSS     AIC
## - Hour     1     0.029  5.224 -30.597
## <none>                  5.195 -28.730
## - Wind     1     2.354  7.549 -21.759
## - Traffic  1    44.070 49.265  23.260
##
## Step:  AIC=-30.6
## CO ~ Traffic + Wind
##
##           Df Sum of Sq    RSS     AIC
## <none>                  5.224 -30.597
## - Wind     1     2.357  7.581 -23.659
## - Traffic  1    46.117 51.341  22.250
```

```
summary(lm.step)
```

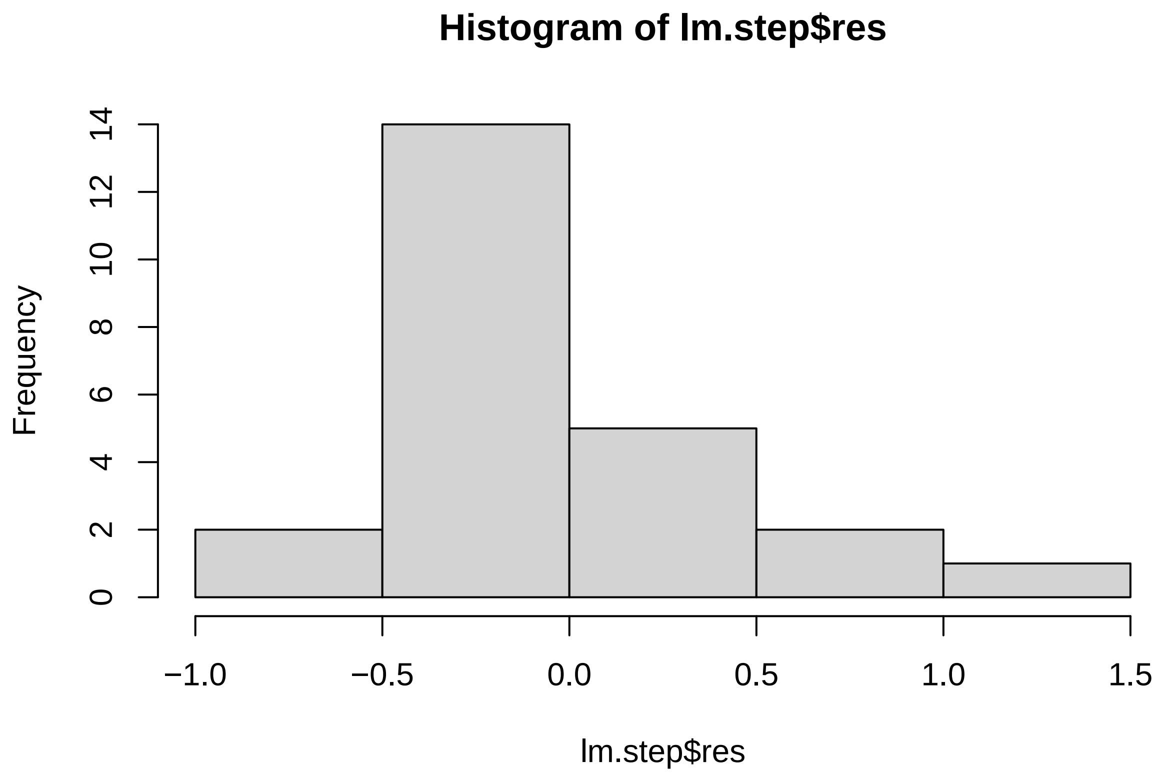
```
##
```

```
## Call:
## lm(formula = CO ~ Traffic + Wind, data = lm_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72858 -0.31710 -0.09629  0.22409  1.26554
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.274461   0.198137   6.432 2.25e-06 ***
## Traffic     0.018290   0.001343  13.616 6.85e-12 ***
## Wind        0.174747   0.056765   3.078   0.0057 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4987 on 21 degrees of freedom
## Multiple R-squared:  0.9495, Adjusted R-squared:  0.9447
## F-statistic: 197.5 on 2 and 21 DF,  p-value: 2.419e-14
```

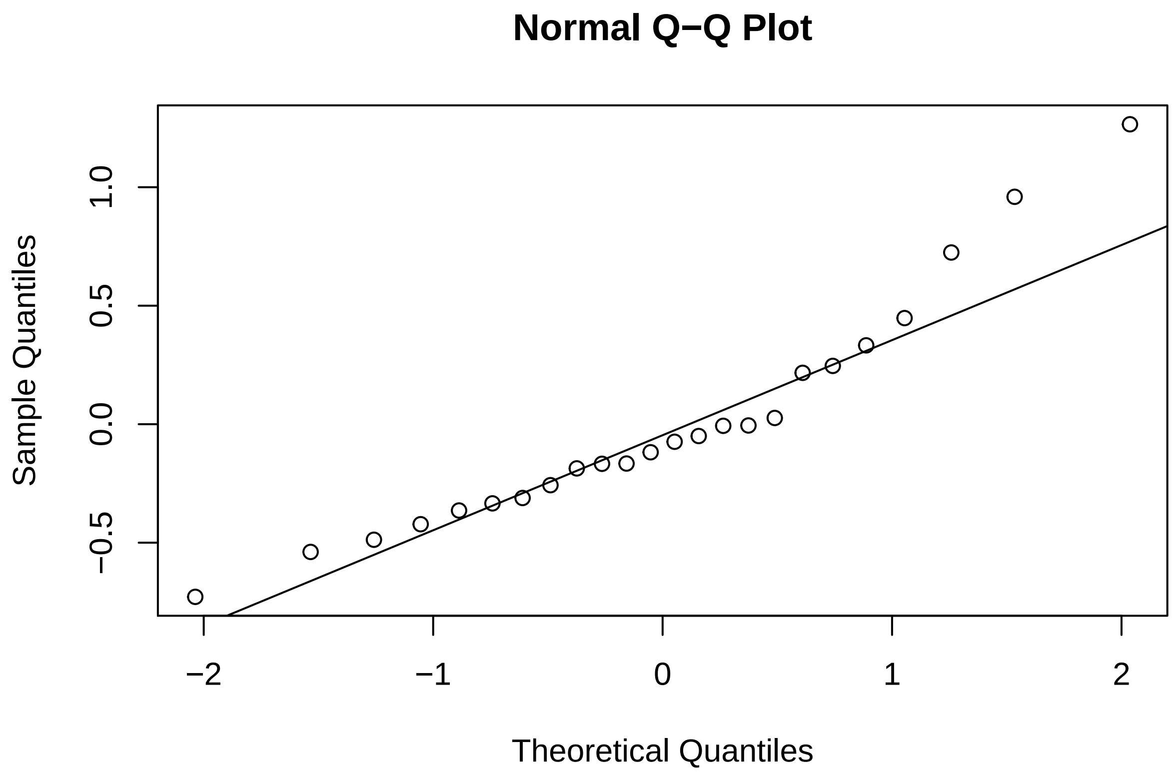
```r
shapiro.test(lm.step$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm.step$res
## W = 0.91918, p-value = 0.05601
```

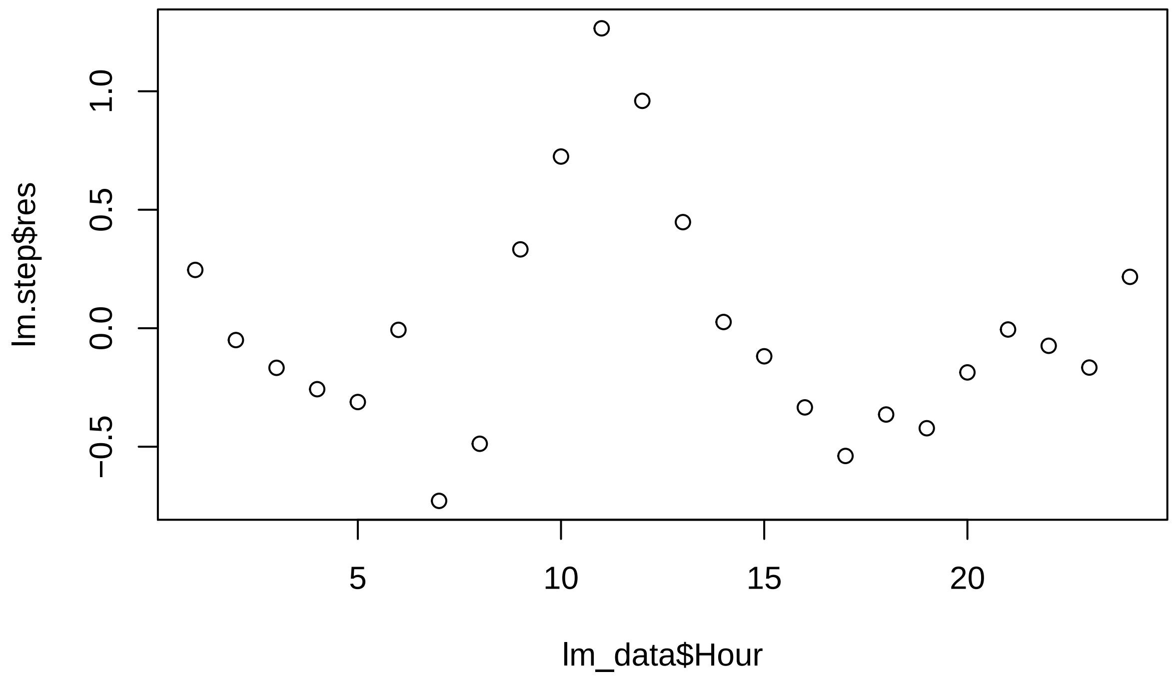
```r
hist(lm.step$res)
```




```r
qqnorm(lm.step$res)
qqline(lm.step$res)
```




```
plot(lm_data$Hour, lm.step$res)
```




```
lm.co <- step(lm(CO ~ Traffic +
    Wind + Wind^2 + sin((2 * pi)/24 *
    Hour) + cos((2 * pi)/24 * Hour) +
    sin((4 * pi)/24 * Hour) + cos((4 *
    pi)/24 * Hour), lm_data), direction = "backward")
```

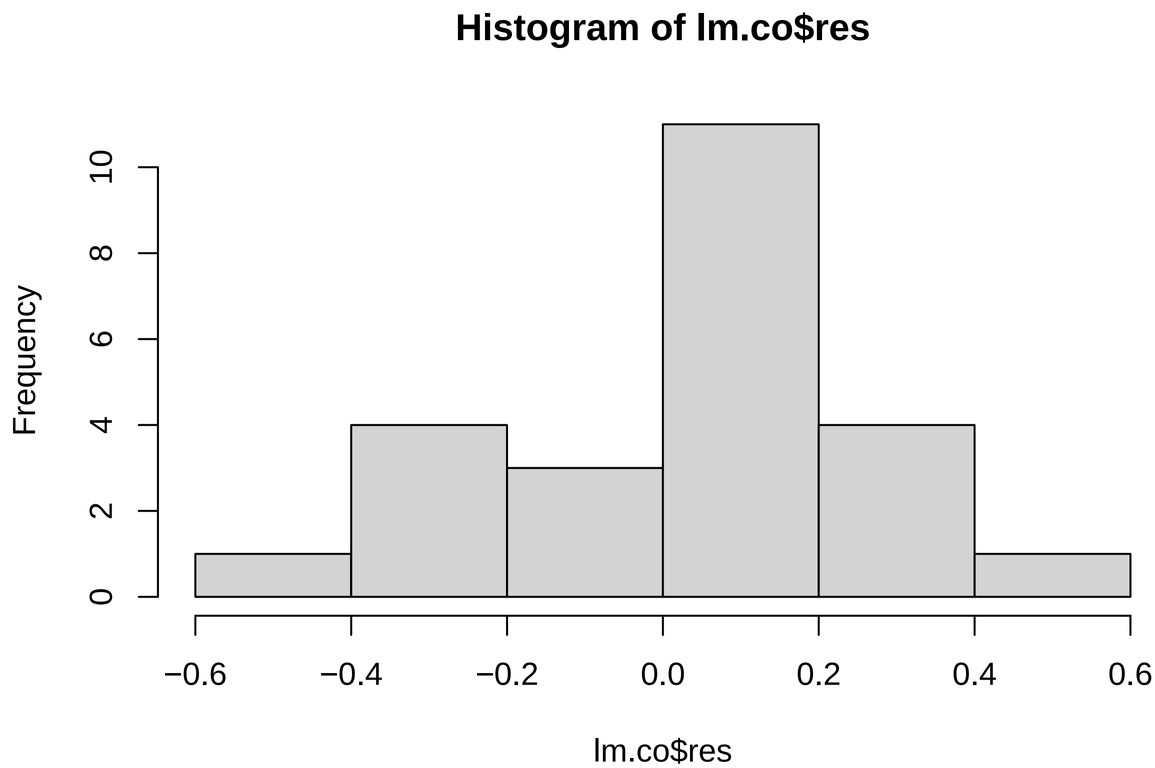
```
## Start:  AIC=-56.26
## CO ~ Traffic + Wind + Wind^2 + sin((2 * pi)/24 * Hour) + cos((2 *
##     pi)/24 * Hour) + sin((4 * pi)/24 * Hour) + cos((4 * pi)/24 *
##     Hour)
##
##                            Df Sum of Sq     RSS     AIC
## - cos((2 * pi)/24 * Hour)  1    0.0038  1.2886 -58.188
## - sin((2 * pi)/24 * Hour)  1    0.0063  1.2910 -58.142
## - Wind                     1    0.0457  1.3305 -57.421
## - sin((4 * pi)/24 * Hour)  1    0.0512  1.3360 -57.322
## <none>                                  1.2848 -56.259
## - cos((4 * pi)/24 * Hour)  1    0.6820  1.9668 -48.040
## - Traffic                  1   10.2379 11.5227  -5.610
##
## Step:  AIC=-58.19
## CO ~ Traffic + Wind + sin((2 * pi)/24 * Hour) + sin((4 * pi)/24 *
##     Hour) + cos((4 * pi)/24 * Hour)
##
##                            Df Sum of Sq     RSS     AIC
```

```
## <none>                                 1.2886 -58.188
## - sin((4 * pi)/24 * Hour)  1    0.5949  1.8835 -51.078
## - sin((2 * pi)/24 * Hour)  1    0.9582  2.2467 -46.846
## - Wind                     1    2.2561  3.5447 -35.902
## - cos((4 * pi)/24 * Hour)  1    2.8122  4.1008 -32.405
## - Traffic                  1   12.6457 13.9343  -3.049
```

```r
summary(lm.co)
```

```
##
## Call:
## lm(formula = CO ~ Traffic + Wind + sin((2 * pi)/24 * Hour) +
##     sin((4 * pi)/24 * Hour) + cos((4 * pi)/24 * Hour), data = lm_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54297 -0.15049  0.03351  0.11670  0.47671
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.098228   0.132739   8.274 1.51e-07 ***
## Traffic                  0.016435   0.001237  13.291 9.58e-11 ***
## Wind                     0.411190   0.073246   5.614 2.51e-05 ***
## sin((2 * pi)/24 * Hour)  0.539048   0.147342   3.658  0.00180 **
## sin((4 * pi)/24 * Hour) -0.437784   0.151861  -2.883  0.00991 **
## cos((4 * pi)/24 * Hour)  0.501101   0.079950   6.268 6.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2676 on 18 degrees of freedom
## Multiple R-squared:  0.9875, Adjusted R-squared:  0.9841
## F-statistic: 285.4 on 5 and 18 DF,  p-value: < 2.2e-16
```

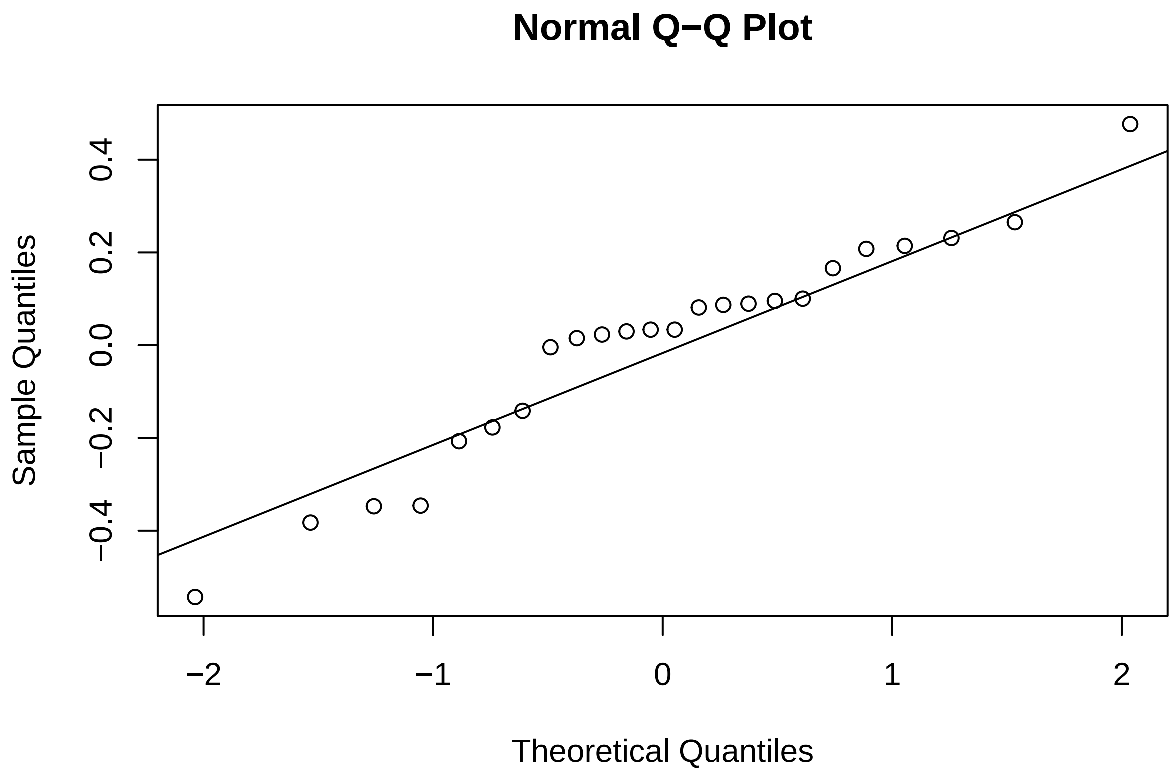
```r
hist(lm.co$res)  # residual
```



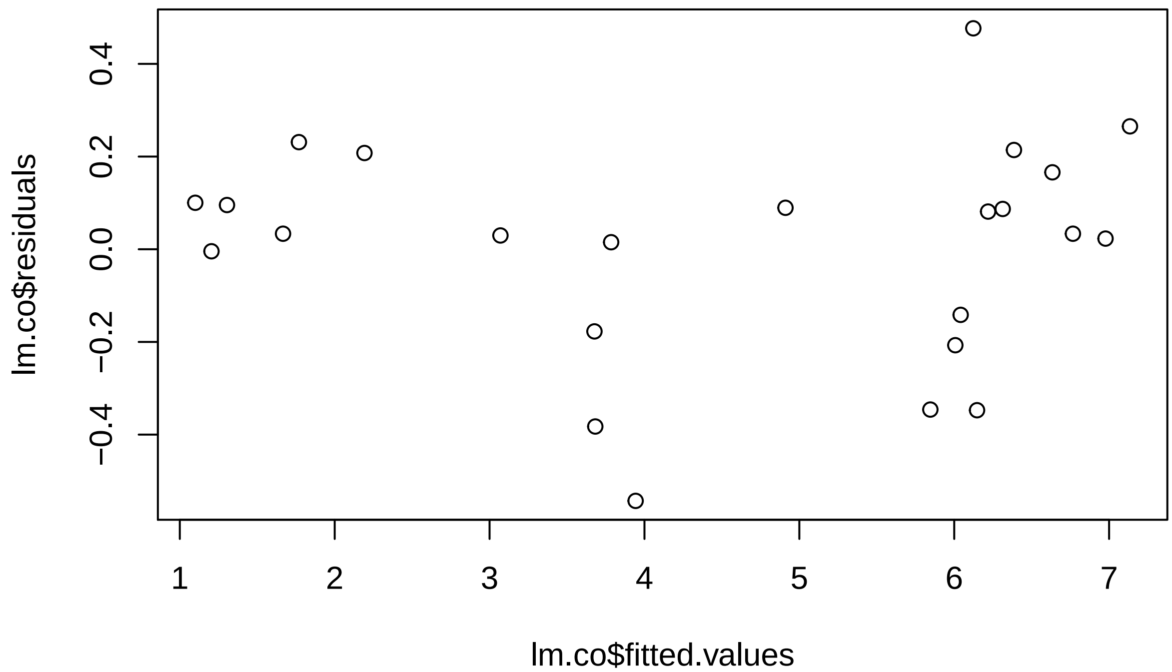

```
shapiro.test(lm.co$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm.co$res
## W = 0.94628, p-value = 0.2247
```

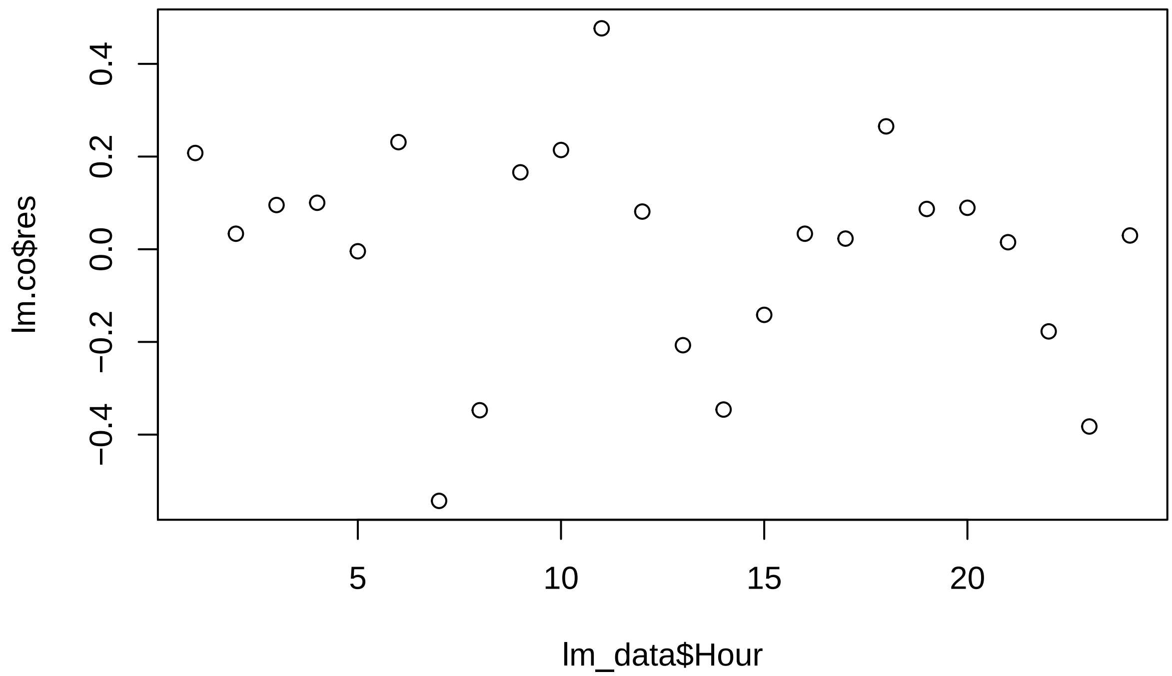
```
qqnorm(lm.co$res)
qqline(lm.co$res)
```




```r
plot(lm.co$fitted.values, lm.co$residuals)  # residual vs fitted value
```




```r
plot(lm_data$Hour, lm.co$res)  # residuals vs time order
```



If you are interested in the linear regression example, read more about this analysis from the source.

8. Format data Use “table1” package to generate summary statistics. See other options in Easily create descriptive summary statistics.

```
# install.packages('table1')
library(table1)
```

```
##
## Attaching package: 'table1'

## The following objects are masked from 'package:base':
##
##     units, units<-
```

```
table <- table1(~Totcare + factor(Nesttype) |
    Location, data = t_test_data)
table
```

```
## [1] "<table class=\"Rtable1\">\n<thead>\n<tr>\n<th class='rowlabel firstrow lastrow'></th>\n<th clas
```

Use “finalfit” package to generate summary statistics with association between dependent and independent variables.

```r
# install.packages('finalfit')
library(finalfit)
dependent = "differ.factor"

# Specify explanatory variables
# of interest
explanatory = c("age", "sex.factor",
    "extent.factor", "obstruct.factor",
    "nodes")

colon_s %>% summary_factorlist(dependent,
    explanatory, p = TRUE, na_include = TRUE)
```

```
## Note: dependent includes missing data. These are dropped.

##              label              levels        Well    Moderate        Poor
##        Age (years)           Mean (SD) 60.2 (12.8) 59.9 (11.7) 59.0 (12.8)
##                Sex              Female   51 (54.8)  314 (47.4)   73 (48.7)
##                                   Male   42 (45.2)  349 (52.6)   77 (51.3)
##   Extent of spread           Submucosa     5 (5.4)    12 (1.8)     3 (2.0)
##                                 Muscle   12 (12.9)   78 (11.8)    12 (8.0)
##                                 Serosa   76 (81.7)  542 (81.7)  127 (84.7)
##                    Adjacent structures     0 (0.0)    31 (4.7)     8 (5.3)
##        Obstruction                  No   69 (74.2)  531 (80.1)  114 (76.0)
##                                    Yes   19 (20.4)  122 (18.4)   31 (20.7)
##                              (Missing)     5 (5.4)    10 (1.5)     5 (3.3)
##              nodes           Mean (SD)   2.7 (2.2)   3.6 (3.4)   4.7 (4.4)
##        p
##    0.644
##    0.400
##
##    0.081
##
##
##
##    0.655
##
##
##   <0.001
```

Reference: Exporting tables and plots

Other sources for formatting statistical results:
apa and apaTables