

# Contents

<b>1</b>	<b>(Normality Test)</b>	<b>2</b>
1.1	.....	2
1.2	.....	2
1.3	.....	2
1.4	Shapiro-Wilk .....	6
<b>2</b>	<b>(Homogeneity of Variance Test)</b>	<b>6</b>
2.1	.....	6
2.2	.....	7
2.3	.....	7
2.4	.....	8
<b>3</b>	<b>(Test of Independence)</b>	<b>9</b>
3.1	.....	9
3.2	(cross tabulation) .....	9
3.3	.....	10
3.4	.....	10
3.5	.....	10
<b>4</b>	<b>One Sample T-test</b>	<b>11</b>
4.1	.....	11
4.2	One sample T-test .....	11
4.3	.....	11
4.4	.....	11
4.5	t.test .....	12
4.6	.....	13
<b>5</b>	<b>Paired samples T-test</b>	<b>14</b>
5.1	.....	14
5.2	Paired samples T-test .....	15
5.3	.....	15
5.4	.....	17
5.5	.....	18
5.6	t.test .....	18
5.7	.....	19
5.8	.....	19
5.9	.....	19
<b>6</b>	<b>Unpaired Two samples T-test</b>	<b>20</b>
6.1	.....	20
6.2	Unpaired Two samples T-test .....	20
6.3	t.test .....	21
6.4	.....	22
6.5	.....	22
6.6	.....	22
<b>7</b>	<b>(One way ANOVA)</b>	<b>23</b>
7.1	.....	23
7.2	.....	23
7.3	.....	24
7.4	One-way ANOVA test .....	26
7.5	.....	26
7.6	.....	28

<b>8</b>	<b>(Two way ANOVA)</b>	<b>29</b>
8.1	.....	29
8.2	.....	29
8.3	.....	32
8.4	Two-way ANOVA test .....	33
8.5	.....	33
<b>9</b>	<b>(MANOVA:Multi-variate Analysis Of Variance)</b>	<b>35</b>
9.1	.....	35
9.2	.....	35
9.3	MANOVA .....	37
<b>10</b>	<b>MANOVA test</b>	<b>37</b>
<b>11</b>	<b>(Regression Model)</b>	<b>38</b>
11.1	.....	38
11.2	.....	38
11.3	.....	41

## 1 (Normality Test)

### 1.1

```
my.data <- ToothGrowth
str(my.data)
```

```
## 'data.frame': 60 obs. of 3 variables:
## $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
## $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
## $ dose: num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

```
R      ToothGrowth  . 3    60    .
```

### 1.2

```

      .
      ,
      ,
      ,
      .

```

### 1.3

```

      , (n>=30)
      (C.L.T - Central Limit Theorem)

```

- , .
- , .
- ,
- .
- ggdensity

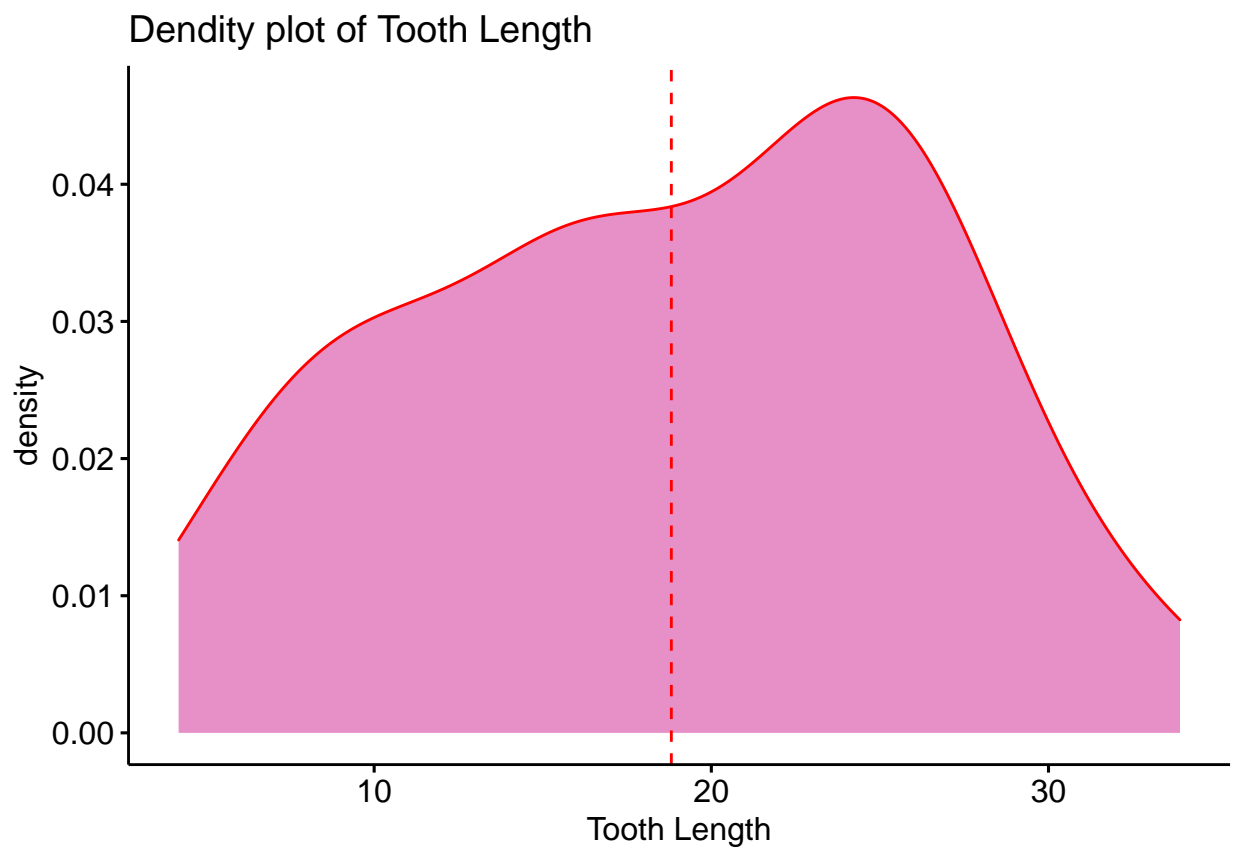
```
library(ggpubr)
```

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

```
ggdensity(my.data$len,  
  add = "mean",  
  color = "red",  
  fill = "violet red",  
  alpha = .5,  
  title = "Dendity plot of Tooth Length",  
  xlab = "Tooth Length")
```

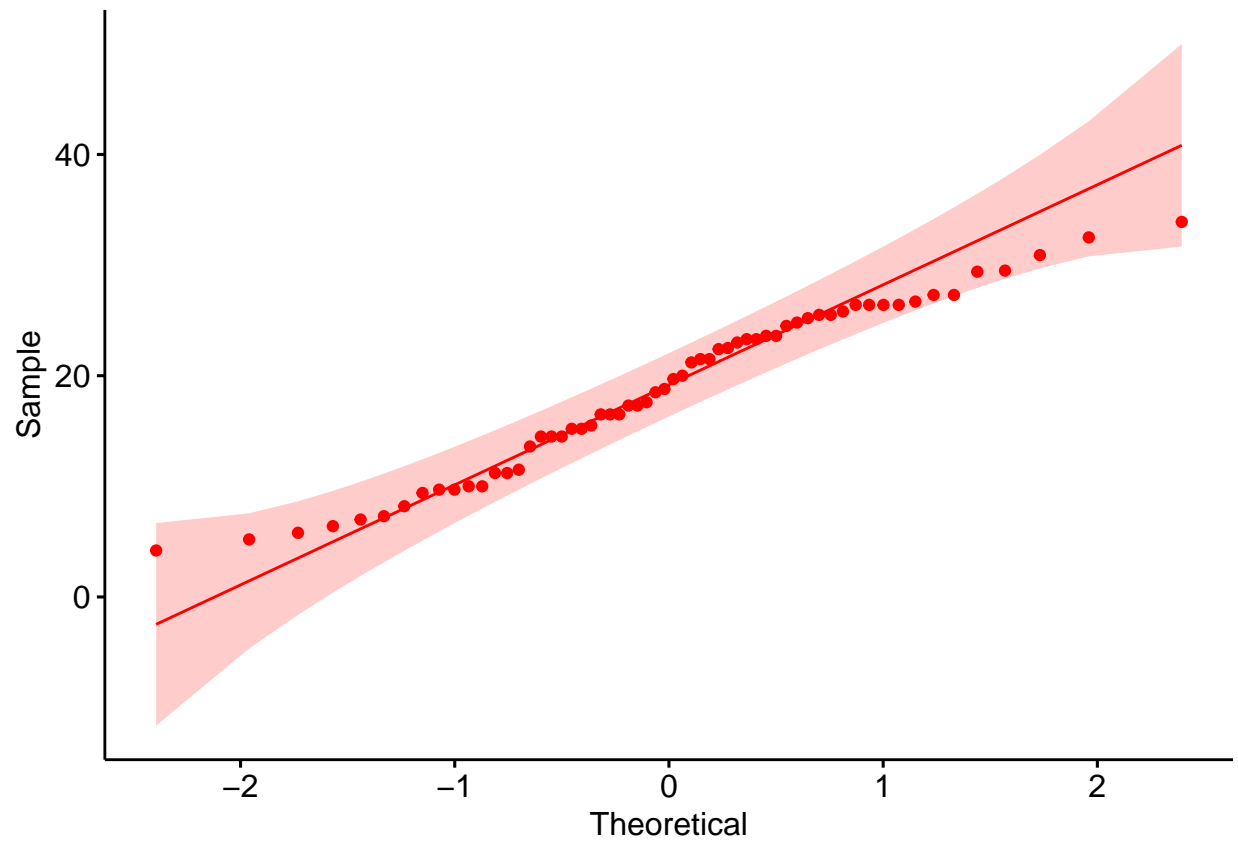
```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```

```
## Warning: geom_vline(): Ignoring `data` because `xintercept` was provided.
```



- ggqqplot

```
library(ggpubr)  
ggqqplot(my.data$len, color = "red")
```

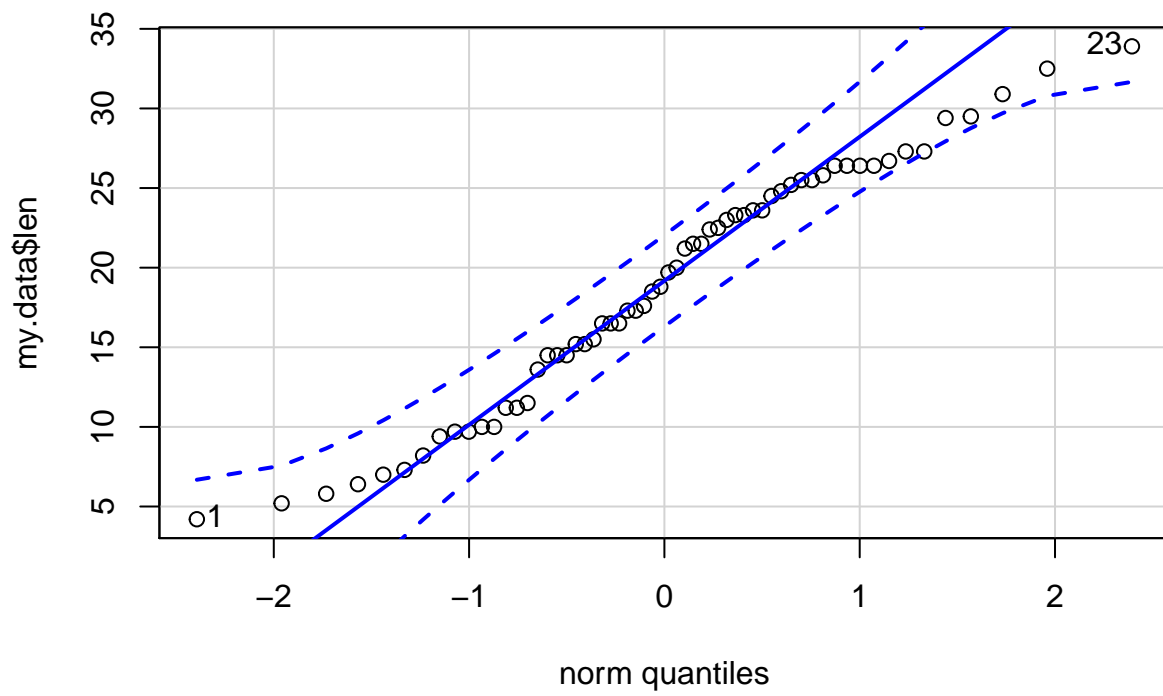


- qqPlot

```
library(car)
```

```
## Loading required package: carData
```

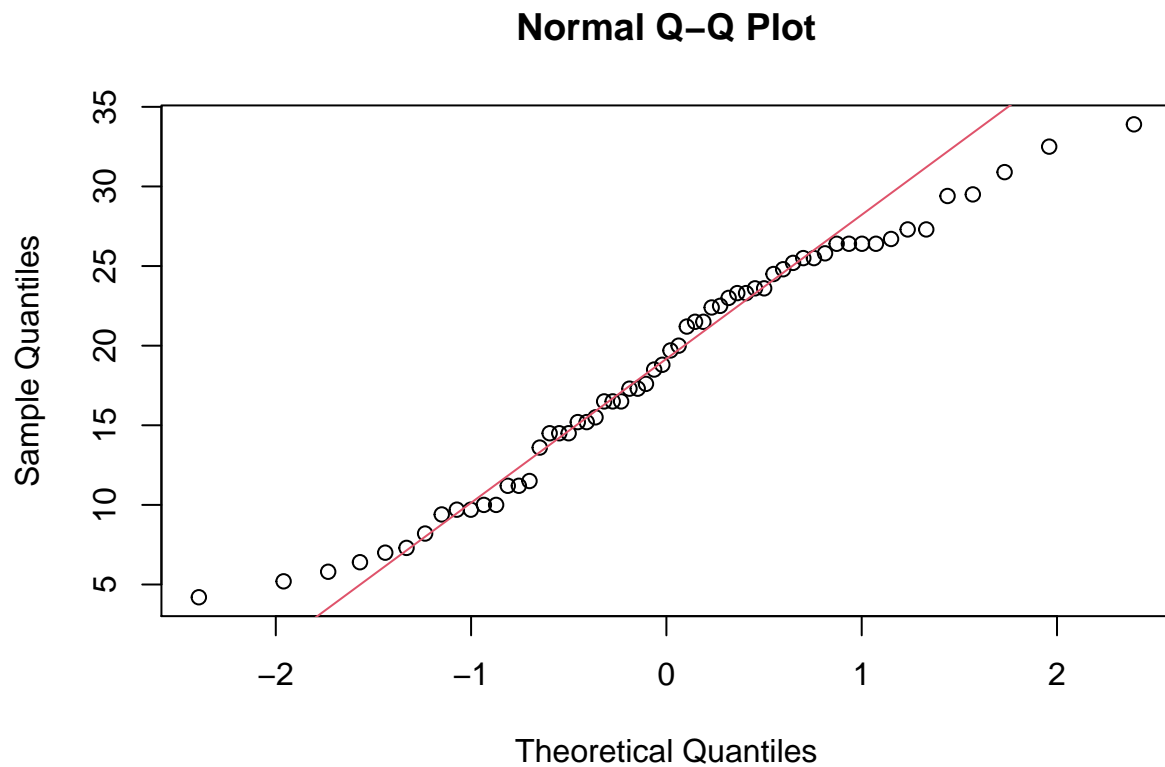
```
qqPlot(my.data$len)
```



```
## [1] 23 1
```

```
•
```

```
qqnorm(my.data$len)
qqline(my.data$len, col = 2)
```



## 1.4 Shapiro-Wilk

$H_0$  : .  $H_1$  : .

```
shapiro.test(my.data$len)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  my.data$len
## W = 0.96743, p-value = 0.1091
```

```
##
##  Shapiro-Wilk normality test
##
## data:  my.data$len
## W = 0.96743, p-value = 0.1091
```

p 0.05 . .

## 2 (Homogeneity of Variance Test)

### 2.1

```
data(ToothGrowth)
data(PlantGrowth)
str(ToothGrowth)
```

```
## 'data.frame': 60 obs. of 3 variables:
## $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
## $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
## $ dose: num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

```
str(PlantGrowth)
```

```
## 'data.frame': 30 obs. of 2 variables:
## $ weight: num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
## $ group : Factor w/ 3 levels "ctrl","trt1",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
R ToothGrowth .
```

```
3 60 .
```

## 2.2

- .
- , .
- , . , .

## 2.3

### 2.3.1 F test

.  
 $H_0$ : ( ).  
 $H_1$ : ( ).

```
res.ftest <- var.test(len ~ supp, data = ToothGrowth, alternative = "two.sided")
res.ftest
```

```
##
## F test to compare two variances
##
## data: len by supp
## F = 0.6386, num df = 29, denom df = 29, p-value = 0.2331
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3039488 1.3416857
## sample estimates:
## ratio of variances
## 0.6385951
```

```
len supp ,
p-value (0.05) 0.2331 .
```

```
 , alternative .
```

```
res.ftest$estimate
```

```
## ratio of variances
## 0.6385951
```

estimate . 63.8% .

## 2.4

### 2.4.1 Bartlett's test

- .
- $H_0 :$  ( ).
- $H_1 :$  ( ).

```
bartlett.test(weight ~ group, data = PlantGrowth)
```

```
##
## Bartlett test of homogeneity of variances
##
## data: weight by group
## Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
```

PlantGrowth group .

p-value (0.05) 0.2371 .

### 2.4.2 Levene's test

{car} .

- .
- $H_0 :$  ( ).
- $H_1 :$  ( ).

```
library(car)
```

```
## Loading required package: carData
```

```
leveneTest(weight ~ group, data = PlantGrowth)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  2  1.1192 0.3412
##      27
```

p-value (0.05) 0.3412 .

### 2.4.3 Fligner-Killeen test

- .
- , .
- $H_0 :$  ( ).
- $H_1 :$  ( ).

```
fligner.test(weight ~ group, data = PlantGrowth)
```



```
##
## Fligner-Killeen test of homogeneity of variances
##
## data: weight by group
## Fligner-Killeen:med chi-squared = 2.3499, df = 2, p-value = 0.3088

PlantGrowth      ,      .
p-value (0.05) 0.3088 .
```

### 3 (Test of Independence)

#### 3.1

```
library(ggplot2)
data(mpg)
mpg <- as.data.frame(mpg)
str(mpg)

## 'data.frame': 234 obs. of 11 variables:
## $ manufacturer: chr "audi" "audi" "audi" "audi" ...
## $ model : chr "a4" "a4" "a4" "a4" ...
## $ displ : num 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
## $ year : int 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
## $ cyl : int 4 4 4 4 6 6 6 4 4 4 ...
## $ trans : chr "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
## $ drv : chr "f" "f" "f" "f" ...
## $ cty : int 18 21 20 21 16 18 18 18 16 20 ...
## $ hwy : int 29 29 31 30 26 26 27 26 25 28 ...
## $ fl : chr "p" "p" "p" "p" ...
## $ class : chr "compact" "compact" "compact" "compact" ...

ggplot2 mpg .
11 234 .
```

#### 3.2 (cross tabulation)

```
library(prettyR)
(crosstab<-xtabs(formula = ~fl+drv, data = mpg))

##      drv
## fl  4  f  r
## c  0  1  0
## d  2  3  0
## e  6  1  1
## p 20 25  7
## r 75 76 17

fl( ) drv( ) .
, .
```

### 3.3

### 3.4

#### 3.4.1

\* , .  
\* , . ( )  
(H0): A, B ( , )  
(H1): A, B ( , ) Pearson's Chi-squared Test( )

```
chisq.test(crosstab, correct = TRUE)
```

```
## Warning in chisq.test(crosstab, correct = TRUE): Chi-squared approximation may  
## be incorrect
```

```
##  
## Pearson's Chi-squared test  
##  
## data: crosstab  
## X-squared = 6.5618, df = 8, p-value = 0.5846  
p-value (0.05) .  
, .
```

#### 3.4.2 Fisher's Exact Test( )

- 
- ,

, .  
p,r .

```
fisher.test(crosstab, alternative = 'two.sided')
```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: crosstab  
## p-value = 0.5121  
## alternative hypothesis: two.sided
```

### 3.5

#### 3.5.1 (covariance)

```

      •
      •      ,      , 0      .
++      ,      ,      ,      ,
++      ,      ,      ,      ,
++      ,      0
cov(mpg$cty,mpg$hwy)

## [1] 24.22543
mpg      cty( ) hwy( )      .
24.22      .

```

## 4 One Sample T-test

### 4.1

```

set.seed(1234)
my.data <- data.frame( name = paste0( rep("M_", 10), 1:10 ), weight = round( rnorm(40, 90, 2), 1 ) )

str(my.data)

## 'data.frame': 40 obs. of 2 variables:
## $ name : chr "M_1" "M_2" "M_3" "M_4" ...
## $ weight: num 87.6 90.6 92.2 85.3 90.9 91 88.9 88.9 88.9 88.2 ...
10      .

```

### 4.2 One sample T-test

```

)
:      500      .
:      500      .

```

### 4.3

```

(my.data)      .

```

### 4.4

```

shapiro.test(my.data$weight)

##
## Shapiro-Wilk normality test
##
## data: my.data$weight
## W = 0.96003, p-value = 0.1679

```

p-value (0.05) .

## 4.5 t.test

```
summary(my.data$weight)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      85.30  88.17   89.00   89.17   90.15   94.80
```

```
summary          10          89.17kg .
```

95kg .( )

```
(res<-t.test(my.data$weight,mu = 95))
```

```
##
##  One Sample t-test
##
## data:  my.data$weight
## t = -20.175, df = 39, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 95
## 95 percent confidence interval:
##  88.58551 89.75449
## sample estimates:
## mean of x
##      89.17
```

p-value (0.05) . . .

, 95kg .

alternative .

```
t.test(my.data$weight, mu = 95, alternative = "less")
```

```
##
##  One Sample t-test
##
## data:  my.data$weight
## t = -20.175, df = 39, p-value < 2.2e-16
## alternative hypothesis: true mean is less than 95
## 95 percent confidence interval:
##      -Inf 89.65688
## sample estimates:
## mean of x
##      89.17
```

less , 95kg .( ) .

```
t.test(my.data$weight, mu = 95, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  my.data$weight
## t = -20.175, df = 39, p-value = 1
```

```
## alternative hypothesis: true mean is greater than 95
## 95 percent confidence interval:
## 88.68312      Inf
## sample estimates:
## mean of x
##      89.17

greater ,      95kg .( )      .
```

```
res$conf.int
```

```
## [1] 88.58551 89.75449
## attr("conf.level")
## [1] 0.95

88.58551~89.75449      0.95      .
```

## 4.6

```
set.seed(1234)
my.data2 <- data.frame( name = paste0( rep("M_", 10), 1:10 ), weight = round( rnorm(70, 90, 2), 1 ) )

str(my.data2)
```

```
## 'data.frame': 70 obs. of 2 variables:
## $ name : chr "M_1" "M_2" "M_3" "M_4" ...
## $ weight: num 87.6 90.6 92.2 85.3 90.9 91 88.9 88.9 88.9 88.2 ...

shapiro.test(my.data2$weight)
```

```
##
## Shapiro-Wilk normality test
##
## data: my.data2$weight
## W = 0.94492, p-value = 0.003967

my.data      .
```

### 4.6.1 wilcox.test( )

```
wilcox.test(my.data2$weight,mu=95)

##
## Wilcoxon signed rank test with continuity correction
##
## data: my.data2$weight
## V = 1, p-value = 3.761e-13
## alternative hypothesis: true location is not equal to 95

p-value      (0.05)      .      .

,      95kg      .
```

```

t.test(my.data2$weight, mu = 95, alternative = "less")

##
## One Sample t-test
##
## data: my.data2$weight
## t = -22.665, df = 69, p-value < 2.2e-16
## alternative hypothesis: true mean is less than 95
## 95 percent confidence interval:
##      -Inf 89.85957
## sample estimates:
## mean of x
## 89.45143

t.test(my.data2$weight, mu = 95, alternative = "greater")

##
## One Sample t-test
##
## data: my.data2$weight
## t = -22.665, df = 69, p-value = 1
## alternative hypothesis: true mean is greater than 95
## 95 percent confidence interval:
## 89.04328      Inf
## sample estimates:
## mean of x
## 89.45143

```

## 5 Paired samples T-test

### 5.1

```

before <-c(200.1, 190.9, 192.7, 213, 241.4, 196.9, 172.2, 185.5, 205.2, 193.7)
after <-c(392.9, 393.2, 345.1, 393, 434, 427.9, 422, 383.9, 392.3, 352.2)

( my.data <- data.frame(group = rep(c("before", "after"), each = 10), weight = c(before, after) ) )

##      group weight
## 1 before  200.1
## 2 before  190.9
## 3 before  192.7
## 4 before  213.0
## 5 before  241.4
## 6 before  196.9
## 7 before  172.2
## 8 before  185.5
## 9 before  205.2
## 10 before 193.7
## 11 after  392.9
## 12 after  393.2
## 13 after  345.1
## 14 after  393.0

```

```
## 15 after 434.0
## 16 after 427.9
## 17 after 422.0
## 18 after 383.9
## 19 after 392.3
## 20 after 352.2

10 (before), (after) weight .
```

## 5.2 Paired samples T-test

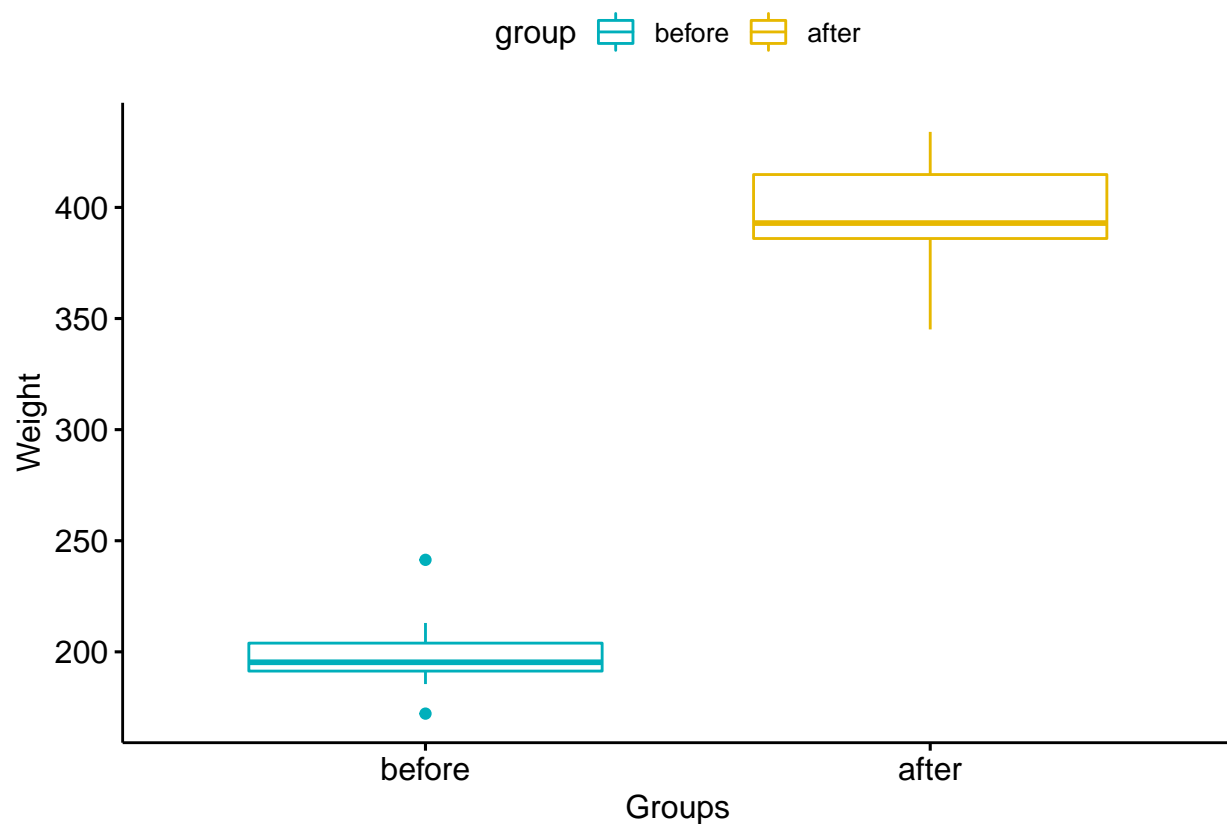
```
(1:1 or ) .
)
: 2009 2010 .
: 2009 2010 .
```

## 5.3

```
library("ggpubr")
```

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

```
ggboxplot(my.data,
  x = "group",
  y = "weight",
  color = "group",
  palette = c("#00AFBB", "#E7B800"),
  order = c("before", "after"),
  ylab = "Weight",
  xlab = "Groups")
```



before after

```
before <- subset(my.data, group == "before", weight, drop = TRUE)
```

```
after <- subset(my.data, group == "after", weight, drop = TRUE)
```

### 5.3.1

```
library(PairedData)
```

```
## Loading required package: MASS
```

```
## Loading required package: gld
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: lattice
```

```
##
```

```
## Attaching package: 'PairedData'
```

```
## The following object is masked from 'package:base':
```

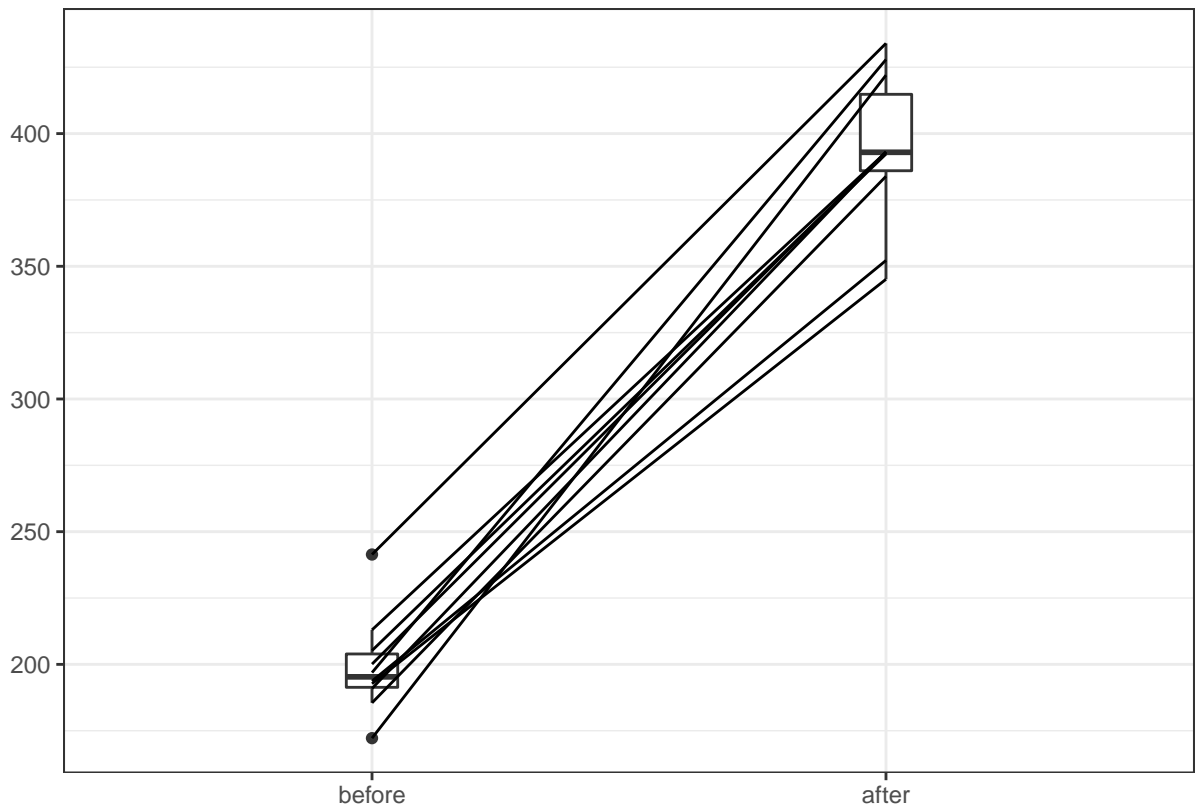
```
##
```

```
## summary
```

```
pd <- paired(before, after)
```

```
plot(pd, type = "profile") + theme_bw()
```





plot

## 5.4

```
shapiro.test(my.data$weight[my.data$group=="before"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  my.data$weight[my.data$group == "before"]
## W = 0.90938, p-value = 0.2768
```

```
shapiro.test(my.data$weight[my.data$group=="after"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  my.data$weight[my.data$group == "after"]
## W = 0.91121, p-value = 0.2894
```

p-value (0.05)

## 5.5

```
var.test(weight~group, data = my.data)
```

```
##
## F test to compare two variances
##
## data: weight by group
## F = 2.5324, num df = 9, denom df = 9, p-value = 0.1825
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.6290172 10.1955065
## sample estimates:
## ratio of variances
## 2.53242
p-value (0.05) .
```

## 5.6 t.test

```
( res <- t.test(weight ~ group, data = my.data, paired = TRUE) )
```

```
##
## Paired t-test
##
## data: weight by group
## t = 20.883, df = 9, p-value = 6.2e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 173.4219 215.5581
## sample estimates:
## mean of the differences
## 194.49
p-value (0.05) .
, before, after weight .
```

```
table(my.data$group)
```

```
t.test(weight ~ group, data = my.data, paired = TRUE, alternative = "less")
```

```
t.test(weight ~ group, data = my.data, paired = TRUE, alternative = "greater")
```

```
alternative .
```

```
table . after .
```

```
less , after weihgt before weihgt .( ) .
```

```
greater , after weihgt before weihgt .( ) .
```

## 5.7

```
res$estimate

## mean of the differences
##              194.49

res$conf.int

## [1] 173.4219 215.5581
## attr("conf.level")
## [1] 0.95

after  weihgt  before weihgt      194.49 .
before, after weight    173.4219~215.5581    0.95 .
```

## 5.8

```
my.data . ##### wilcox.test( )

wilcox.test(weight ~ group, data = my.data, paired = TRUE)

##
## Wilcoxon signed rank exact test
##
## data: weight by group
## V = 55, p-value = 0.001953
## alternative hypothesis: true location shift is not equal to 0
p-value (0.05) . .
, before, after weight .
```

## 5.9

```
table(my.data$group)

##
## after before
##      10      10

wilcox.test(weight ~ group, data = my.data, paired = TRUE, alternative = "less")

##
## Wilcoxon signed rank exact test
##
## data: weight by group
## V = 55, p-value = 1
## alternative hypothesis: true location shift is less than 0

wilcox.test(weight ~ group, data = my.data, paired = TRUE, alternative = "greater")

##
## Wilcoxon signed rank exact test
##
## data: weight by group
## V = 55, p-value = 0.0009766
```

```
## alternative hypothesis: true location shift is greater than 0
```

## 6 Unpaired Two samples T-test

### 6.1

```
9, 9
women_weight <- c(38.9, 61.2, 73.3, 21.8, 63.4, 64.6, 48.4, 48.8, 48.5)
men_weight <- c(67.8, 60, 63.4, 76, 89.4, 73.3, 67.3, 61.3, 62.4)

(my.data <- data.frame(group = rep(c("Woman", "Man"), each = 9), weight = c(women_weight, men_weight)))

##   group weight
## 1  Woman   38.9
## 2  Woman   61.2
## 3  Woman   73.3
## 4  Woman   21.8
## 5  Woman   63.4
## 6  Woman   64.6
## 7  Woman   48.4
## 8  Woman   48.8
## 9  Woman   48.5
## 10 Man    67.8
## 11 Man    60.0
## 12 Man    63.4
## 13 Man    76.0
## 14 Man    89.4
## 15 Man    73.3
## 16 Man    67.3
## 17 Man    61.3
## 18 Man    62.4
```

### 6.2 Unpaired Two samples T-test

```
)
:
:
```

#### 6.2.1

```
,
```

#### 6.2.2

```
shapiro.test(my.data$weight[my.data$group=="Woman"])
```

```
##
## Shapiro-Wilk normality test
```

```
##
## data: my.data$weight[my.data$group == "Woman"]
## W = 0.94266, p-value = 0.6101
```

### 6.2.3

```
shapiro.test(my.data$weight[my.data$group=="Man"])
```

```
##
## Shapiro-Wilk normality test
##
## data: my.data$weight[my.data$group == "Man"]
## W = 0.86425, p-value = 0.1066
p-value (0.05) .
```

### 6.2.4

```
var.test(weight~group, data = my.data)
```

```
##
## F test to compare two variances
##
## data: weight by group
## F = 0.36134, num df = 8, denom df = 8, p-value = 0.1714
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.08150656 1.60191315
## sample estimates:
## ratio of variances
## 0.3613398
p-value (0.05) .
```

## 6.3 t.test

```
( res <- t.test(weight ~ group, data = my.data, var.equal = TRUE) )
```

```
##
## Two Sample t-test
##
## data: weight by group
## t = 2.7842, df = 16, p-value = 0.01327
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 4.029759 29.748019
## sample estimates:
## mean in group Man mean in group Woman
## 68.98889 52.10000
p-value (0.05) . .
```

```

table(my.data$group)
t.test(weight ~ group, data = my.data, var.equal = TRUE, alternative = "less")
t.test(weight ~ group, data = my.data, var.equal = TRUE, alternative = "greater")
alternative
table
less , .( )
greater , .( )

```

## 6.4

```

res$estimate
## mean in group Man mean in group Woman
## 68.98889 52.10000
res$conf.int
## [1] 4.029759 29.748019
## attr("conf.level")
## [1] 0.95
68.98889kg, 52.1kg .
4.029759~29.748019 0.95 .

```

## 6.5

```
my.data
```

### 6.5.1 wilcox.test( )

```

wilcox.test(weight ~ group, data = my.data, exact = FALSE)
##
## Wilcoxon rank sum test with continuity correction
##
## data: weight by group
## W = 66, p-value = 0.02712
## alternative hypothesis: true location shift is not equal to 0
p-value (0.05)

```

## 6.6

```
table(my.data$group)
```

```
##
##   Man Woman
##     9     9

wilcox.test(weight ~ group, data = my.data, exact = FALSE, alternative = "less")

##
##   Wilcoxon rank sum test with continuity correction
##
## data:  weight by group
## W = 66, p-value = 0.9892
## alternative hypothesis: true location shift is less than 0

wilcox.test(weight ~ group, data = my.data, exact = FALSE, alternative = "greater")

##
##   Wilcoxon rank sum test with continuity correction
##
## data:  weight by group
## W = 66, p-value = 0.01356
## alternative hypothesis: true location shift is greater than 0
```

## 7 (One way ANOVA)

### 7.1

```
data(PlantGrowth)
str(PlantGrowth)

## 'data.frame':   30 obs. of  2 variables:
## $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
## $ group : Factor w/ 3 levels "ctrl","trt1",...: 1 1 1 1 1 1 1 1 1 1 ...

levels(PlantGrowth$group)

## [1] "ctrl" "trt1" "trt2"

R      PlantGrowth
2      60
group  3  (ctrl, trt1, trt2)
```

### 7.2

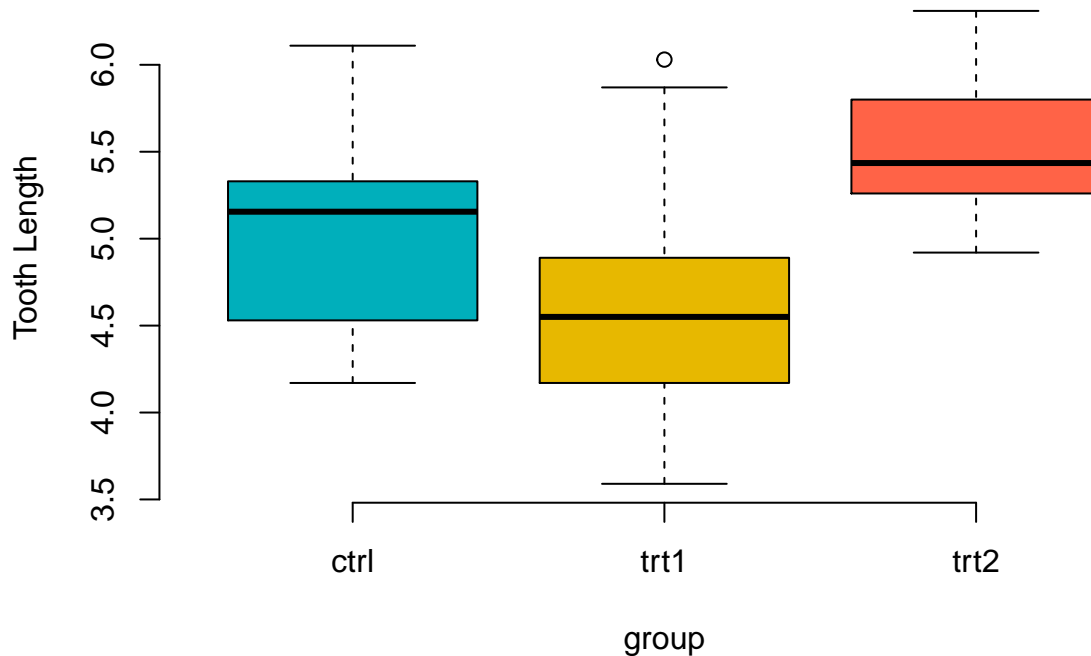
```
, T-test

3

( ) ~ (3 ) ,

boxplot(weight ~ group,
        data=PlantGrowth,
        frame = FALSE,
```

```
col = c("#00AFBB", "#E7B800", "tomato"),
ylab="Tooth Length")
```



boxplot

## 7.3

### 7.3.1

```
str(PlantGrowth)
```

```
## 'data.frame': 30 obs. of 2 variables:
## $ weight: num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
## $ group : Factor w/ 3 levels "ctrl","trt1",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
with(data = PlantGrowth, shapiro.test(weight[group=="ctrl"]))
```

```
##
## Shapiro-Wilk normality test
##
## data: weight[group == "ctrl"]
## W = 0.95668, p-value = 0.7475
```

```
with(data = PlantGrowth, shapiro.test(weight[group=="trt1"]))
```

```
##
```



```
## Shapiro-Wilk normality test
##
## data:  weight[group == "trt1"]
## W = 0.93041, p-value = 0.4519
with(data = PlantGrowth, shapiro.test(weight[group=="trt2"]))

##
## Shapiro-Wilk normality test
##
## data:  weight[group == "trt2"]
## W = 0.94101, p-value = 0.5643
      (30 ) clt
      .
      ,
3      .
```

### 7.3.2

```
bartlett.test(weight~group, data = PlantGrowth)

##
## Bartlett test of homogeneity of variances
##
## data:  weight by group
## Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
library(car)

## Loading required package: carData
leveneTest(weight~group, data = PlantGrowth)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  2  1.1192 0.3412
##      27
      .
      .    p      .
```

### 7.3.3

```
ctrl <- with(data = PlantGrowth, weight[group=="ctrl"])
trt1 <- with(data = PlantGrowth, weight[group=="trt1"])
trt2 <- with(data = PlantGrowth, weight[group=="trt2"])
cov(ctrl, trt1)

## [1] -0.2118022
cov(ctrl, trt2)

## [1] -0.1206244
```

```
cov(trt1, trt2)
```

```
## [1] -0.04887333
```

```
3          ,      0  .
```

```
,      .
```

## 7.4 One-way ANOVA test

```
:      .
```

```
:      . ('      '      .)
```

```
plant.aov <- aov(weight~group, data = PlantGrowth)
```

```
summary(plant.aov)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
```

```
## group      2  3.766  1.8832    4.846 0.0159 *
```

```
## Residuals 27 10.492  0.3886
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
p          .
```

```
, 3          .
```

## 7.5

```
3          .
```

```
,      .
```

```
.(3  )
```

### 7.5.1 TukeyHSD

```
TukeyHSD(plant.aov)
```

```
## Tukey multiple comparisons of means
```

```
## 95% family-wise confidence level
```

```
##
```

```
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
```

```
##
```

```
## $group
```

```
##           diff          lwr          upr          p adj
```

```
## trt1-ctrl -0.371 -1.0622161  0.3202161  0.3908711
```

```
## trt2-ctrl  0.494 -0.1972161  1.1852161  0.1979960
```

```
## trt2-trt1  0.865  0.1737839  1.5562161  0.0120064
```

```
TukeyHSD          .
```

```
p adj   trt2 trt1   (0.05)          .
```

```
, trt2 trt1          .
```

- diff:
- lwr, upr: 95%

```

• p adj:    p
• glht
++ glht() :
++ glht(model, lincft) + linfct : linear,
library(multcomp)

## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##      geyser
summary(glht(plant.aov, linfct = mcp(group="Tukey")))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
##
## Linear Hypotheses:
##              Estimate Std. Error t value Pr(>|t|)
## trt1 - ctrl == 0  -0.3710     0.2788  -1.331   0.391
## trt2 - ctrl == 0   0.4940     0.2788   1.772   0.198
## trt2 - trt1 == 0   0.8650     0.2788   3.103   0.012 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

TukeyHSD      .

mcp: group      Tukey      .

pairwise.t.test

## function (x, g, p.adjust.method = p.adjust.methods, pool.sd = !paired,
##      paired = FALSE, alternative = c("two.sided", "less", "greater"),
##      ...)
## {
##     if (paired & pool.sd)
##         stop("pooling of SD is incompatible with paired tests")
##     DNAME <- paste(deparse1(substitute(x)), "and", deparse1(substitute(g)))
##     g <- factor(g)
##     p.adjust.method <- match.arg(p.adjust.method)
##     alternative <- match.arg(alternative)
##     if (pool.sd) {

```

```

##      METHOD <- "t tests with pooled SD"
##      xbar <- tapply(x, g, mean, na.rm = TRUE)
##      s <- tapply(x, g, sd, na.rm = TRUE)
##      n <- tapply(!is.na(x), g, sum)
##      degf <- n - 1
##      total.degf <- sum(degf)
##      pooled.sd <- sqrt(sum(s^2 * degf)/total.degf)
##      compare.levels <- function(i, j) {
##          dif <- xbar[i] - xbar[j]
##          se.dif <- pooled.sd * sqrt(1/n[i] + 1/n[j])
##          t.val <- dif/se.dif
##          if (alternative == "two.sided")
##              2 * pt(-abs(t.val), total.degf)
##          else pt(t.val, total.degf, lower.tail = (alternative ==
##              "less"))
##      }
##  }
##  else {
##      METHOD <- if (paired)
##          "paired t tests"
##      else "t tests with non-pooled SD"
##      compare.levels <- function(i, j) {
##          xi <- x[as.integer(g) == i]
##          xj <- x[as.integer(g) == j]
##          t.test(xi, xj, paired = paired, alternative = alternative,
##              ...)$p.value
##      }
##  }
##  PVAL <- pairwise.table(compare.levels, levels(g), p.adjust.method)
##  ans <- list(method = METHOD, data.name = DNAME, p.value = PVAL,
##      p.adjust.method = p.adjust.method)
##  class(ans) <- "pairwise.htest"
##  ans
## }
## <bytecode: 0x000000001e835838>
## <environment: namespace:stats>

pairwise.t.test(PlantGrowth$weight, PlantGrowth$group, p.adjust.method = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: PlantGrowth$weight and PlantGrowth$group
##
##      ctrl  trt1
## trt1 0.194 -
## trt2 0.132 0.013
##
## P value adjustment method: BH

```

## 7.6

```
kruskal.test .
```

```
kruskal.test(weight ~ group, data = PlantGrowth)

##
## Kruskal-Wallis rank sum test
##
## data: weight by group
## Kruskal-Wallis chi-squared = 7.9882, df = 2, p-value = 0.01842
```

## 8 (Two way ANOVA)

### 8.1

```
data(ToothGrowth)
str(ToothGrowth)

## 'data.frame': 60 obs. of 3 variables:
## $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
## $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
## $ dose: num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

levels(ToothGrowth$supp)

## [1] "OJ" "VC"

R      ToothGrowth
3      60
supp( ) 2 (OJ, VC)

ToothGrowth$dose <- factor(ToothGrowth$dose, levels = c(0.5, 1, 2),
  labels = c("D0.5", "D1", "D2"))
str(ToothGrowth)

## 'data.frame': 60 obs. of 3 variables:
## $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
## $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
## $ dose: Factor w/ 3 levels "D0.5","D1","D2": 1 1 1 1 1 1 1 1 1 1 ...

dose ( ) 3 factor
```

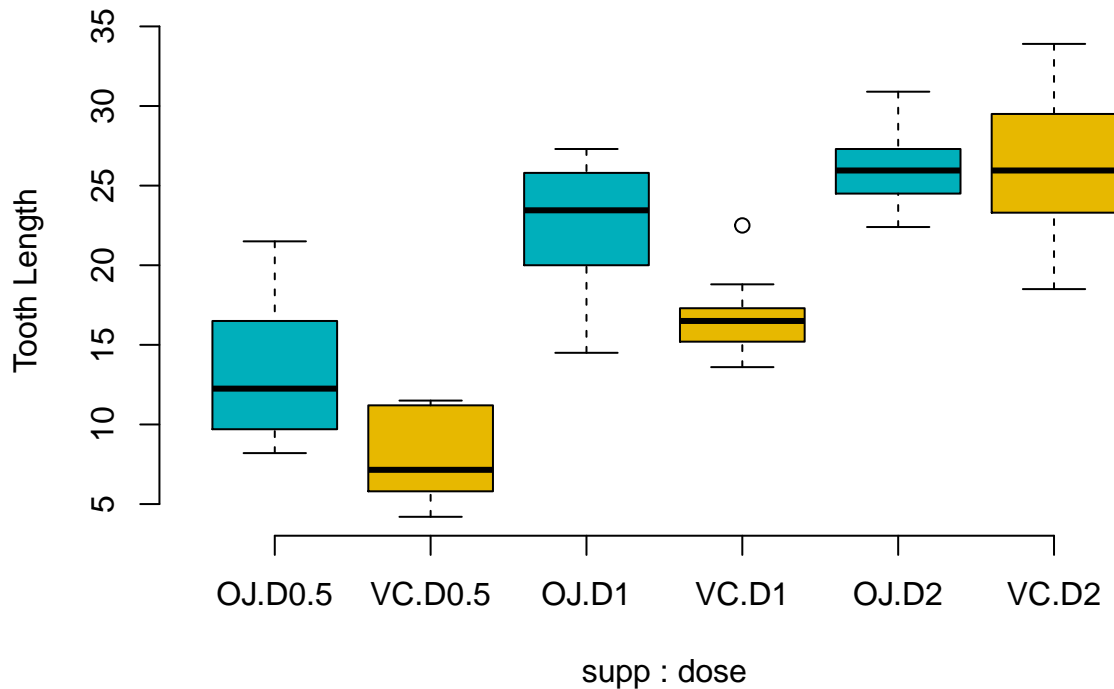
### 8.2

```
### , T-test
3
.
```

#### 8.2.1

```
( ) ~ (3 ) ,
boxplot(len ~ supp * dose,
  data=ToothGrowth,
  frame = FALSE,
```

```
col = c("#00AFBB", "#E7B800"),
ylab="Tooth Length")
```



boxplot

interaction.plot

```
## function (x.factor, trace.factor, response, fun = mean, type = c("l",
##   "p", "b", "o", "c"), legend = TRUE, trace.label = deparse1(substitute(trace.factor)),
##   fixed = FALSE, xlab = deparse1(substitute(x.factor)), ylab = ylabel,
##   ylim = range(cells, na.rm = TRUE), lty = nc:1, col = 1, pch = c(1L:9,
##     0, letters), xpd = NULL, leg.bg = par("bg"), leg.bty = "n",
##   xtick = FALSE, xaxt = par("xaxt"), axes = TRUE, ...)
## {
##   ylabel <- paste(deparse1(substitute(fun)), "of ", deparse1(substitute(response)))
##   type <- match.arg(type)
##   cells <- tapply(response, list(x.factor, trace.factor), fun)
##   nr <- nrow(cells)
##   nc <- ncol(cells)
##   xvals <- 1L:nr
##   if (is.ordered(x.factor)) {
##     wn <- getOption("warn")
##     options(warn = -1)
##     xnm <- as.numeric(levels(x.factor))
##     options(warn = wn)
##     if (!anyNA(xnm))
##       xvals <- xnm
##   }
```

```

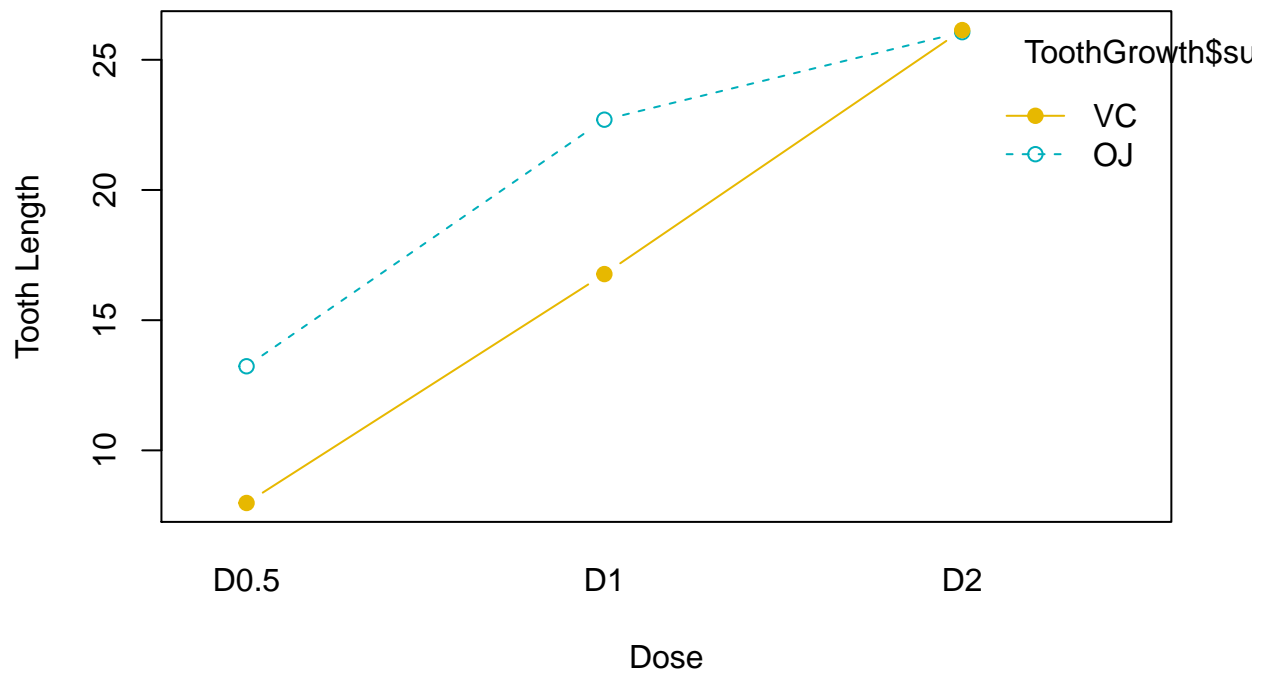
##     }
##     xlabs <- rownames(cells)
##     ylabs <- colnames(cells)
##     nch <- max(sapply(ylabs, nchar, type = "width"))
##     if (is.null(xlabs))
##       xlabs <- as.character(xvals)
##     if (is.null(ylabs))
##       ylabs <- as.character(1L:nc)
##     xlim <- range(xvals)
##     xleg <- xlim[2L] + 0.05 * diff(xlim)
##     xlim <- xlim + c(-0.2/nr, if (legend) 0.2 + 0.02 * nch else 0.2/nr) *
##       diff(xlim)
##     dev.hold()
##     on.exit(dev.flush())
##     matplot(xvals, cells, ..., type = type, xlim = xlim, ylim = ylim,
##       xlab = xlab, ylab = ylab, axes = axes, xaxt = "n", col = col,
##       lty = lty, pch = pch)
##     if (axes && xaxt != "n") {
##       axisInt <- function(x, main, sub, lwd, bg, log, asp,
##         ...) axis(1, x, ...)
##       mgp. <- par("mgp")
##       if (!xtick)
##         mgp.[2L] <- 0
##       axisInt(1, at = xvals, labels = xlabs, tick = xtick,
##         mgp = mgp., xaxt = xaxt, ...)
##     }
##     if (legend) {
##       yrng <- diff(ylim)
##       yleg <- ylim[2L] - 0.1 * yrng
##       if (!is.null(xpd) || {
##         xpd. <- par("xpd")
##         !is.na(xpd.) && !xpd. && (xpd <- TRUE)
##       }) {
##         op <- par(xpd = xpd)
##         on.exit(par(op), add = TRUE)
##       }
##       text(xleg, ylim[2L] - 0.05 * yrng, paste(" ", trace.label),
##         adj = 0)
##       if (!fixed) {
##         ord <- sort.list(cells[nr, ], decreasing = TRUE)
##         ylabs <- ylabs[ord]
##         lty <- lty[1 + (ord - 1)%length(lty)]
##         col <- col[1 + (ord - 1)%length(col)]
##         pch <- pch[ord]
##       }
##       legend(xleg, yleg, legend = ylabs, col = col, pch = if (type %in%
##         c("p", "b"))
##           pch, lty = if (type %in% c("l", "b"))
##             lty, bty = leg.bty, bg = leg.bg)
##     }
##     invisible()
## }
## <bytecode: 0x00000000138afcb8>
## <environment: namespace:stats>

```

```

interaction.plot(x.factor = ToothGrowth$dose,
trace.factor = ToothGrowth$supp,
response = ToothGrowth$len,
fun = mean,
type = "b",
legend = TRUE,
xlab = "Dose",
ylab="Tooth Length",
pch=c(1,19),
col = c("#00AFBB", "#E7B800"))

```



- x.factor : factor
- trace.factor:
- -response :

### 8.3

#### 8.3.1

(30 ) clt



### 8.3.2

```
library(car)

## Loading required package: carData
leveneTest(len ~ supp*dose, data = ToothGrowth)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  5  1.7086 0.1484
##      54
##      . p      .
```

## 8.4 Two-way ANOVA test

```
:
:
:      . ('      '      .)

tooth.aov <- aov(len ~ supp + dose, data = ToothGrowth)

summary(tooth.aov)

##      Df Sum Sq Mean Sq F value    Pr(>F)
## supp      1  205.4    205.4    14.02 0.000429 ***
## dose      2 2426.4   1213.2    82.81 < 2e-16 ***
## Residuals 56  820.4     14.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
supp(    ) dose(    ) p
,
,
, (supp, dose)
```

```
tooth.aov2 <- aov(len ~ supp*dose, data = ToothGrowth)

summary(tooth.aov2)

##      Df Sum Sq Mean Sq F value    Pr(>F)
## supp      1  205.4    205.4   15.572 0.000231 ***
## dose      2 2426.4   1213.2   92.000 < 2e-16 ***
## supp:dose  2  108.3     54.2    4.107 0.021860 *
## Residuals 54  712.1     13.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
      supp:dose(    )    0.05
.
```

## 8.5

```
,
.(3 )
```

### 8.5.1 TukeyHSD

```
TukeyHSD(tooth.aov2, which="dose")

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = len ~ supp * dose, data = ToothGrowth)
##
## $dose
##          diff          lwr          upr    p adj
## D1-D0.5  9.130   6.362488 11.897512 0.0e+00
## D2-D0.5 15.495  12.727488 18.262512 0.0e+00
## D2-D1    6.365   3.597488  9.132512 2.7e-06

TukeyHSD
      , which      .(      supp      .)
p adj
      ,
-diff:      -lwr, upr: 95%      -p adj:      p
```

### 8.5.2 glht

```
glht() :
glht(model, linct) linct : linear,
library(multcomp)

## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##      geyser
summary(glht(tooth.aov, linct = mcp(dose="Tukey")))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
## Fit: aov(formula = len ~ supp + dose, data = ToothGrowth)
##
## Linear Hypotheses:
##          Estimate Std. Error t value Pr(>|t|)
## D1 - D0.5 == 0    9.130      1.210   7.543 <1e-05 ***
```

```
## D2 - D0.5 == 0    15.495      1.210  12.802   <1e-05 ***
## D2 - D1 == 0      6.365      1.210   5.259   <1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

TukeyHSD .

### 8.5.3 pairwise.t.test

```
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose, p.adjust.method = "BH")
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data:  ToothGrowth$len and ToothGrowth$dose
##
##      D0.5      D1
## D1 1.0e-08 -
## D2 4.4e-16 1.4e-05
##
## P value adjustment method: BH
```

## 9 (MANOVA:Multi-variate Analysis Of Variance)

### 9.1

```
data(iris)
str(iris)
```

```
## 'data.frame':   150 obs. of  5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
levels(iris$Species)
```

```
## [1] "setosa" "versicolor" "virginica"
```

```
iris( ) Sepal.Length: Sepal.Width: Petal.Length: Petal.Width: Species:
```

R iris .

5 150 .

### 9.2

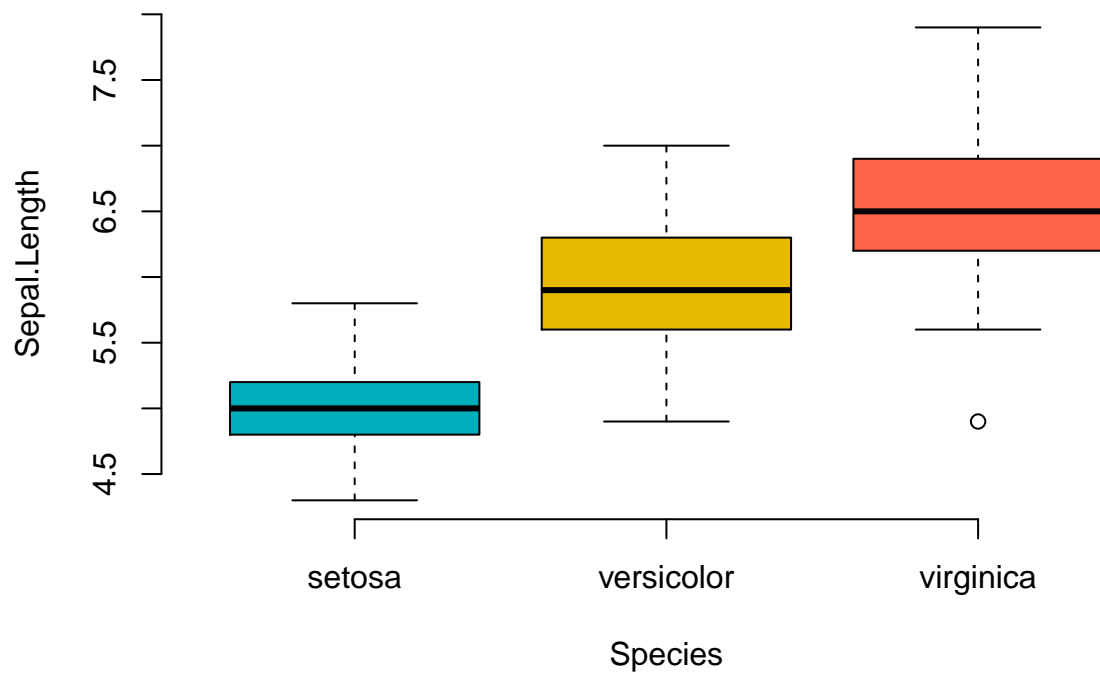
, T-test .

3 .

```

( ) ~ (3 ) ,
boxplot(Sepal.Length ~ Species,
        data=iris,
        frame = FALSE,
        col = c("#00AFBB", "#E7B800", "tomato"),
        ylab="Sepal.Length")

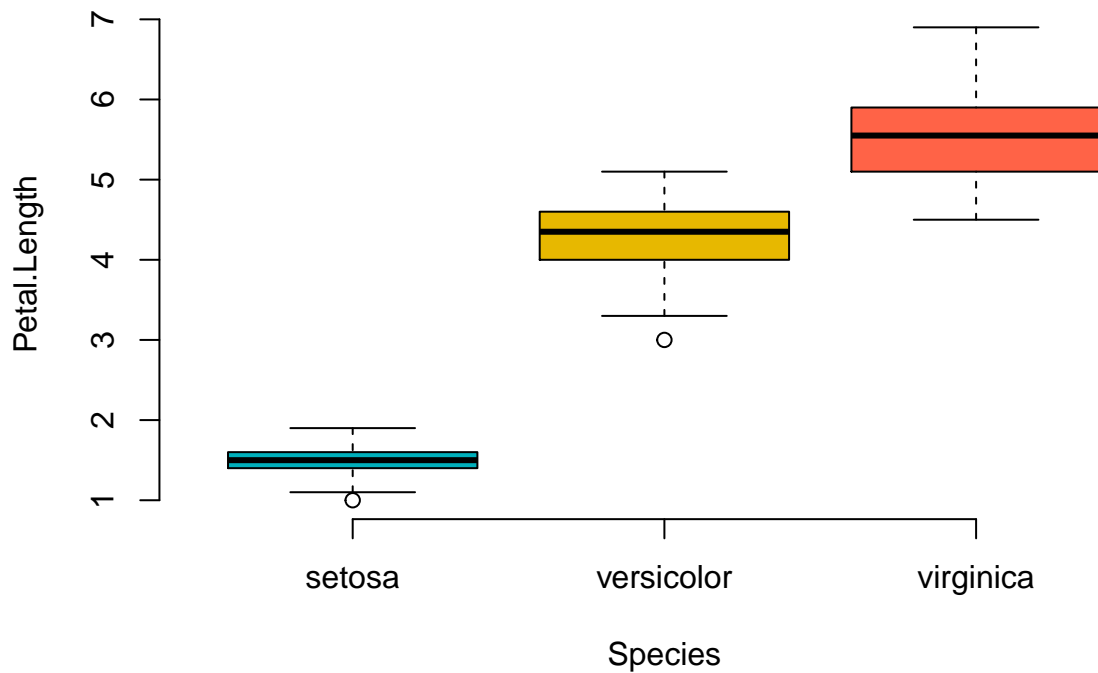
```



```

boxplot(Petal.Length ~ Species,
        data=iris,
        frame = FALSE,
        col = c("#00AFBB", "#E7B800", "tomato"),
        ylab="Petal.Length")

```



boxplot

### 9.3 MANOVA

```
:
:
: (' ' .)
```

## 10 MANOVA test

```
sepl <- irisSepal.Lengthpetl <- irisPetal.Length
```

```
res.man <- manova(cbind(Sepal.Length, Petal.Length) ~ Species, data = iris)
```

```
summary(res.man)
```

```
##              Df Pillai approx F num Df den Df    Pr(>F)
## Species      2  0.9885   71.829      4   294 < 2.2e-16 ***
## Residuals 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

p

```
,
```

```
summary.aov(res.man)
```

```
## Response Sepal.Length :
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Species      2 63.212   31.606   119.26 < 2.2e-16 ***
## Residuals    147 38.956    0.265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Response Petal.Length :
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Species      2 437.10  218.551  1180.2 < 2.2e-16 ***
## Residuals    147  27.22    0.185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary.aov      .
```

## 11 (Regression Model)

### 11.1

```
data(women)
str(women)
```

```
## 'data.frame':   15 obs. of  2 variables:
## $ height: num  58 59 60 61 62 63 64 65 66 67 ...
## $ weight: num 115 117 120 123 126 129 132 135 139 142 ...

women( ) -height: ( :in) -weight: ( :lb)

R      women      .
2      15      .
```

### 11.2

```
cor(women$height,women$weight)
```

```
## [1] 0.9954948
```

```
cor.test(women$height,women$weight)
```

```
##
## Pearson's product-moment correlation
##
## data:  women$height and women$weight
## t = 37.855, df = 13, p-value = 1.091e-14
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9860970 0.9985447
## sample estimates:
##      cor
```

```
## 0.9954948
```

```
1 .  
0.99 .
```

```
##
```

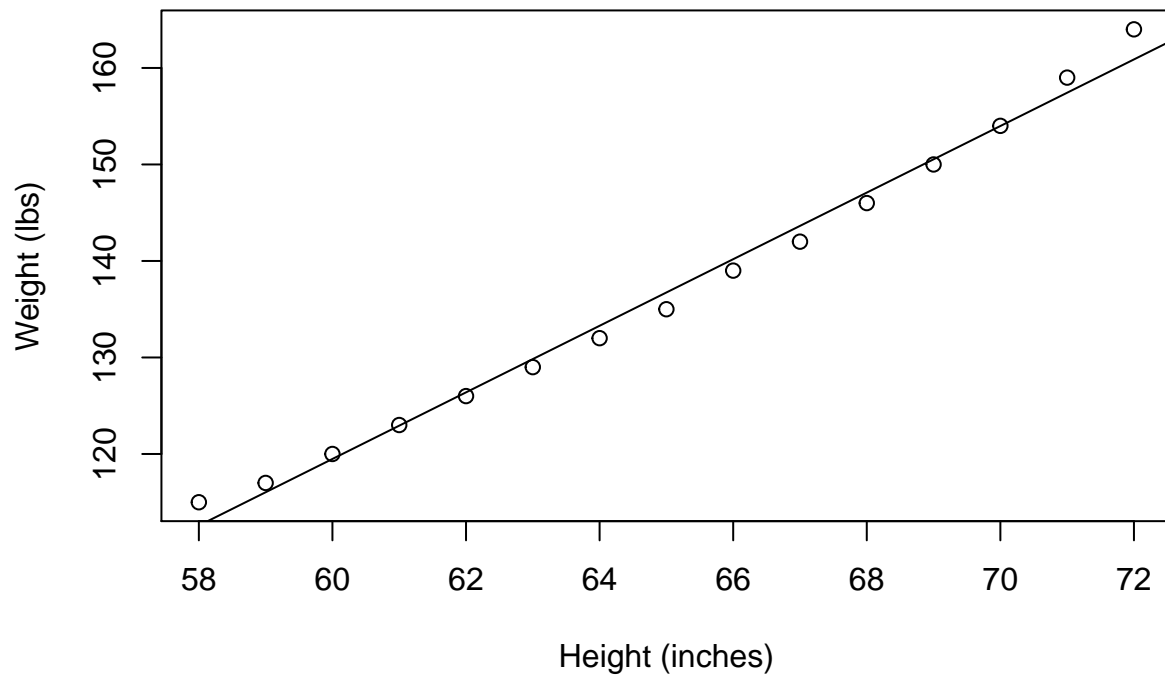
### 11.2.1

```
fit1 <- lm(weight~height, data = women)  
summary(fit1)
```

```
##  
## Call:  
## lm(formula = weight ~ height, data = women)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -1.7333 -1.1333 -0.3833  0.7417  3.1167   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept) -87.51667    5.93694  -14.74 1.71e-09 ***  
## height       3.45000     0.09114   37.85 1.09e-14 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 1.525 on 13 degrees of freedom  
## Multiple R-squared:  0.991, Adjusted R-squared:  0.9903   
## F-statistic: 1433 on 1 and 13 DF, p-value: 1.091e-14  
  
• Call:  
• Residuals( ):  
• : 0 .  
• R-squared( ): : 99.1% .(  
• ( )  
• F-statistic: :  
  , height  
R-squared 0.991  
  ,  
weight = -87.51667 + 3.45 x height
```

### 11.2.2 plot

```
plot(x = women$height, y = women$weight,  
      xlab = 'Height (inches)', ylab = 'Weight (lbs)')  
abline(fit1)
```



### 11.2.3 scatterplot

```
library(car)
```

```
## Loading required package: carData
```

```
scatterplot(
  weight ~ height,
  data = women,
  spread = FALSE,
  smoother.args = list(lty=2),
  pch = 19,
  main = 'Women Age 30 ~ 39',
  xlab = 'Height (inches)',
  ylab = 'Weight (lbs)'
)
```

```
## Warning in plot.window(...): "spread" is not a graphical parameter
```

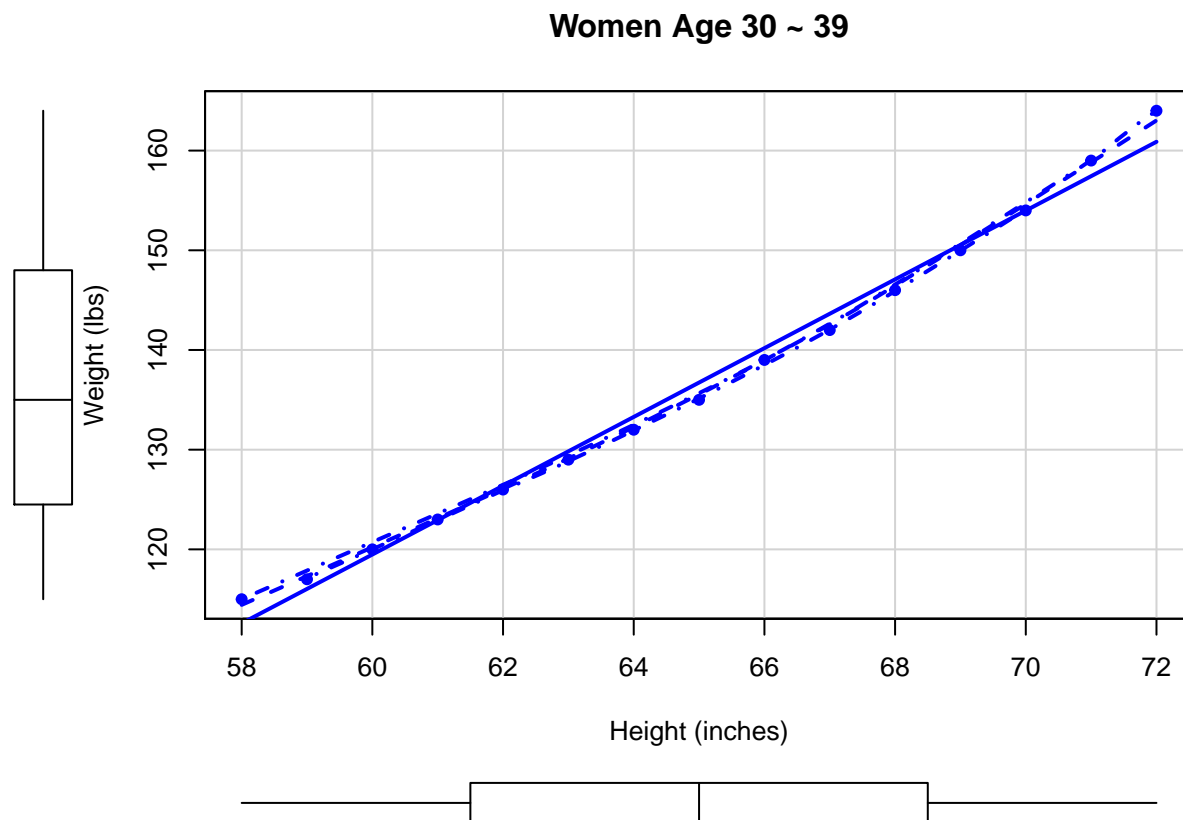
```
## Warning in plot.window(...): "smoother.args" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "spread" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "smoother.args" is not a graphical parameter
```



```
## Warning in axis(side = side, at = at, labels = labels, ...): "spread" is not a
## graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "smoother.args" is
## not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "spread" is not a
## graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "smoother.args" is
## not a graphical parameter
## Warning in box(...): "spread" is not a graphical parameter
## Warning in box(...): "smoother.args" is not a graphical parameter
## Warning in title(...): "spread" is not a graphical parameter
## Warning in title(...): "smoother.args" is not a graphical parameter
```



```
#lty ling type
#
#      .( (loess)(smoother):      .)
#      .(      .)
```

### 11.3

2 .

```
fit2 <- lm(weight~height+I(height^2), data = women)
summary(fit2)
```

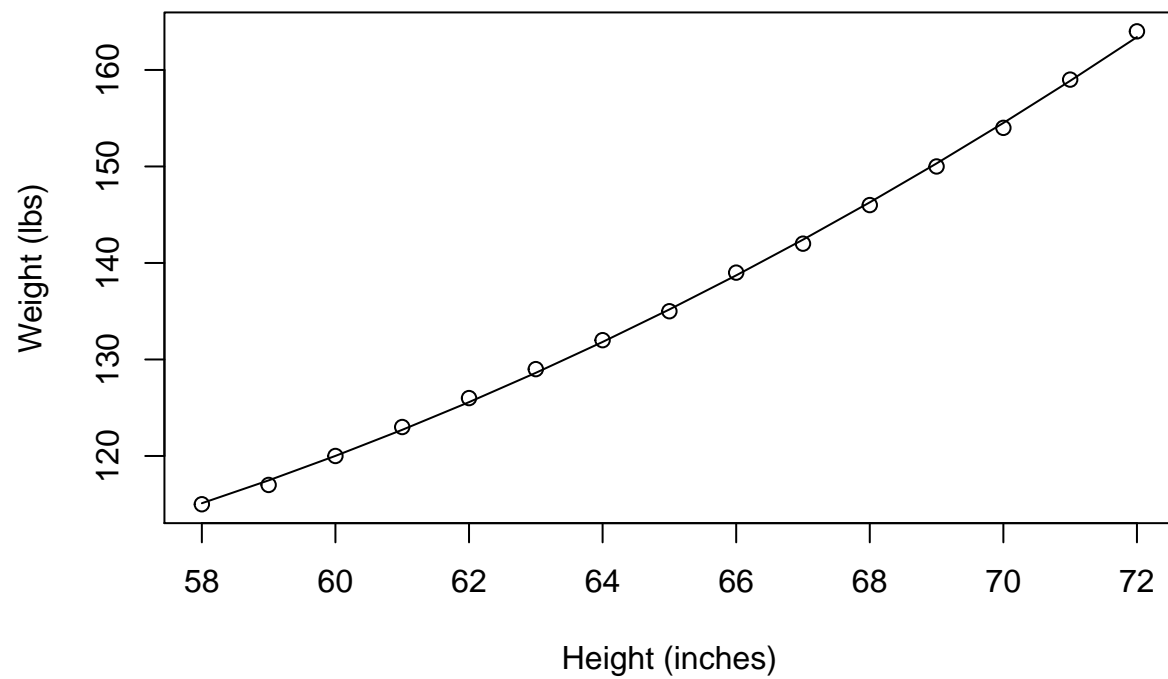
```
##
## Call:
## lm(formula = weight ~ height + I(height^2), data = women)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50941 -0.29611 -0.00941  0.28615  0.59706
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 261.87818   25.19677   10.393 2.36e-07 ***
## height      -7.34832    0.77769   -9.449 6.58e-07 ***
## I(height^2)  0.08306    0.00598   13.891 9.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3841 on 12 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 1.139e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

fit1

R-squared

**weight = 261.87818 - 7.34832 x height + 0.08306 x height<sup>2</sup>**

```
with(data = women, expr = {
  plot(x = height, y = weight, xlab = 'Height (inches)', ylab = 'Weight (lbs)')
  lines(x = height, y = fitted(fit2))
})
```



, fit1 .