Supplementary file

All listed R codes can directly be performed in the R console. Just copy and paste the contents if you want to execute it by yourself. All graph examples can be generated using internal R codes, the package “ggplot2 (H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.)” can provide better plots than one by internal commands. However, some of plots below could be easier to draw with internal commands. It is recommended using internal commands when the complex plot components are not shown as your intention.

The lines start with a special characters “#>” indicate output results corresponding the preceding commend.

Before starting statistical analysis and graphing, set the directory where the data is located as the working directory. The command is:

setwd("Your working directory here”) #e.g., setwd(“C:/temp”)

Be careful the use of slash and back-slash. R code uses the backslash (/) and computer system uses the slash (\) as a typographical mark for file system path.

# # 1. One Sample Mann-Whitney test #

#Set Working Directory

setwd("Your working directory here")

#Read CSV file

mydata = read.csv("size.csv", header = TRUE, fileEncoding = "UTF-8-BOM")

head (mydata)

#> Case size

#> 1 1 1.1

#> 2 2 1.3

#> 3 3 0.8

#> 4 4 0.8

#> 5 5 0.1

#> 6 6 1.1

# Libraries required

library(ggplot2)

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

#Descriptives

summary(mydata)

#> Case size

#> Min. : 1.0 Min. :0.100

#> 1st Qu.:11.5 1st Qu.:0.800

#> Median :22.0 Median :1.100

#> Mean :22.0 Mean :1.077

#> 3rd Qu.:32.5 3rd Qu.:1.300

#> Max. :43.0 Max. :2.100

count(mydata)

#> n

#> 1 43

#Find the outlier

outlier\_values <- boxplot.stats(mydata$size)$out

outlier\_values

#> [1] 2.1

 # Box-Whisker plot

ggplot(data=mydata, aes(y=size))+

 stat\_boxplot(geom = 'errorbar', width=0.1)+

 geom\_boxplot(width=0.3)+

 theme\_classic()+

 theme(axis.text.x = element\_blank(),

 axis.ticks = element\_blank()

 )+

 scale\_y\_continuous(name="Size (Describe the unit)")+

scale\_x\_continuous(limits = c(-0.3,0.6),name = "")+

coord\_fixed(ratio = 1/2)+ #Fix y/x aspect ratio

 #Annotations

 annotate("segment", x = 0.4, xend = 0.17

 , y=2.1, yend=2.1,

 colour = "black", size = 0.5

 , arrow=arrow(length = unit(0.2, "cm")))+

 annotate("text", x = 0.467, y = 2.1, label = "Outlier", size = 5)+

 annotate("segment", x = 0.4, xend = 0.17

 , y=1.6, yend=1.6,

 colour = "black", size = 0.5

 , arrow=arrow(length = unit(0.2, "cm")))+

 annotate("text", x = 0.4475, y = 1.6, size = 5

 , label = "Max")+

 annotate("segment", x = 0.4, xend = 0.17

 , y=1.3, yend=1.3,

 colour = "black", size = 0.5

 , arrow=arrow(length = unit(0.2, "cm")))+

 annotate("text", x = 0.44, y = 1.3, size = 5

 , label = "Q3")+

 annotate("segment", x = 0.4, xend = 0.17

 , y=1.1, yend=1.1,

 colour = "black", size = 0.5

 , arrow=arrow(length = unit(0.2, "cm")))+

 annotate("text", x = 0.4725, y = 1.1, size = 5

 , label = "Median")+

 annotate("segment", x = 0.4, xend = 0.17

 , y=0.8, yend=0.8,

 colour = "black", size = 0.5

 , arrow=arrow(length = unit(0.2, "cm")))+

 annotate("text", x = 0.44, y = 0.8, size = 5

 , label = "Q1")+

 annotate("segment", x = 0.4, xend = 0.17

 , y=0.1, yend=0.1,

 colour = "black", size = 0.5

 , arrow=arrow(length = unit(0.2, "cm")))+

 annotate("text", x = 0.445, y = 0.1, size = 5

 , label = "Max")



# # 2. Independent t-test #

#Set Working Directory

setwd("Your working directory here")

#Read CSV file

mydata = read.csv("reverse.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

)

head (mydata)

#> case rev\_type mov\_sec

#> 1 1 ache 70

#> 2 2 ache 43

#> 3 3 ache 77

#> 4 4 ache 67

#> 5 5 ache 72

#> 6 6 ache 93

#library

library(ggplot2)

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

library(psych)

#>

#> Attaching package: 'psych'

#> The following objects are masked from 'package:ggplot2':

#>

#> %+%, alpha

library(ggpubr)

#Descriptive statistics

#gmodels::CrossTable(mydata$sex, mydata$rev\_type)

describeBy(mydata, group = mydata$rev\_type) #Library "psych"

#>

#> Descriptive statistics by group

#> group: ache

#> vars n mean sd median trimmed mad min max range skew kurtosis

#> case 1 109 55.49 32.09 55 55.45 41.51 1 111 110 0.01 -1.24

#> rev\_type\* 2 109 1.00 0.00 1 1.00 0.00 1 1 0 NaN NaN

#> mov\_sec 3 109 70.15 11.24 70 69.71 10.38 43 95 52 0.20 0.00

#> se

#> case 3.07

#> rev\_type\* 0.00

#> mov\_sec 1.08

#> ------------------------------------------------------------

#> group: new

#> vars n mean sd median trimmed mad min max range skew

#> case 1 109 163.51 32.79 164 164.00 40.03 60 218 158 -0.21

#> rev\_type\* 2 109 1.00 0.00 1 1.00 0.00 1 1 0 NaN

#> mov\_sec 3 109 58.41 7.72 58 58.19 8.90 35 80 45 0.16

#> kurtosis se

#> case -0.61 3.14

#> rev\_type\* NaN 0.00

#> mov\_sec 0.01 0.74

#Normality test for each group

ache <- filter(mydata, rev\_type == "ache") %>% select(mov\_sec)

shapiro.test(ache$mov\_sec)

#>

#> Shapiro-Wilk normality test

#>

#> data: ache$mov\_sec

#> W = 0.97685, p-value = 0.05425

new <- filter(mydata, rev\_type == "new") %>% select(mov\_sec)

shapiro.test(new$mov\_sec)

#>

#> Shapiro-Wilk normality test

#>

#> data: new$mov\_sec

#> W = 0.98737, p-value = 0.3997

#Testing homogeneity of variances in k samples

bartlett.test(mydata$mov\_sec ~ mydata$rev\_type, data = mydata)

#>

#> Bartlett test of homogeneity of variances

#>

#> data: mydata$mov\_sec by mydata$rev\_type

#> Bartlett's K-squared = 14.862, df = 1, p-value = 0.0001157

#Unequal variance t-test

t.test(mydata$mov\_sec ~ mydata$rev\_type, var.equal=FALSE)

#>

#> Welch Two Sample t-test

#>

#> data: mydata$mov\_sec by mydata$rev\_type

#> t = 8.9857, df = 191.3, p-value = 2.467e-16

#> alternative hypothesis: true difference in means between group ache and group new is not equal to 0

#> 95 percent confidence interval:

#> 9.158235 14.309655

#> sample estimates:

#> mean in group ache mean in group new

#> 70.14679 58.41284

#Plotting

temp <- mydata %>% #Data for graph

 group\_by(rev\_type) %>%

 summarise(mn = mean(mov\_sec), sd = sd(mov\_sec))

temp

#> # A tibble: 2 x 3

#> rev\_type mn sd

#> <chr> <dbl> <dbl>

#> 1 ache 70.1 11.2

#> 2 new 58.4 7.72

ggplot(temp, aes(x=rev\_type, y=mn))+

 geom\_errorbar(aes(ymin=mn-sd, ymax=mn+sd), width=0.3)+

 geom\_col(width=0.5)+

 ggpubr::geom\_bracket(xmin="ache", xmax="new"

 , y.position=85

 , label = "\*"

 , tip.length = c(0.05,0.05)

 , coord.flip = FALSE)+

 theme\_classic()+

 scale\_y\_continuous(limits = c(0,95),

 , name = "Movement time (sec)"

 , expand = c(0,0)

 , breaks = seq(0,90,by = 10))+

 scale\_x\_discrete(labels = c("Anticholinergic", "New drug")

 , name = "Reversal agents")



# # 3. Paired t-test example #

# Cautions: This process does not include

# the basic assumptions validations for the test

#Set Working Directory

setwd("Your working directory here")

#Read CSV file

mydata = read.csv("cuffpressure.csv", header = TRUE, fileEncoding = "UTF-8-BOM")

head (mydata)

#> patient manual cuff25

#> 1 1 16 10

#> 2 2 60 27

#> 3 3 58 26

#> 4 4 70 31

#> 5 5 38 18

#> 6 6 72 32

t.test( mydata$manual, mydata$cuff25, paired = TRUE)

#>

#> Paired t-test

#>

#> data: mydata$manual and mydata$cuff25

#> t = 23.263, df = 99, p-value < 2.2e-16

#> alternative hypothesis: true difference in means is not equal to 0

#> 95 percent confidence interval:

#> 27.27656 32.36344

#> sample estimates:

#> mean of the differences

#> 29.82

#Library

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

library(reshape)

#>

#> Attaching package: 'reshape'

#> The following object is masked from 'package:dplyr':

#>

#> rename

library(ggpubr)

#> Loading required package: ggplot2

library(ggplot2)

mydata.melt <- melt(mydata, id="patient") #Library: reshape

mydata.melt.ordered <- mydata.melt %>%

 arrange(mydata.melt$patient

 , mydata.melt$variable

 ) #Library: dplyr

head(mydata.melt.ordered)

#> patient variable value

#> 1 1 manual 16

#> 2 1 cuff25 10

#> 3 2 manual 60

#> 4 2 cuff25 27

#> 5 3 manual 58

#> 6 3 cuff25 26

temp <- mydata.melt.ordered %>%

 group\_by(variable) %>%

 summarise(mean = mean(value)

 , sd = sd(value)

 ) #A dataset for ggplot

temp

#> # A tibble: 2 x 3

#> variable mean sd

#> <fct> <dbl> <dbl>

#> 1 manual 55.1 20.4

#> 2 cuff25 25.3 7.77

p <- ggplot(temp, aes(x=variable, y=mean, group = 1))+

 geom\_line()+

 geom\_errorbar(aes(ymin = mean - sd, ymax = mean+sd), width = 0.05)+

 geom\_point(shape=21, size = 3, fill = "white")+

 theme\_classic()+

 scale\_y\_continuous(limits = c(0,80),

 , name = "Inflation volume (ml)"

 , expand = c(0,0)

 , breaks = seq(0,80,by = 10))+

 scale\_x\_discrete(labels = c("Manual", "Cuff pressured at 25 mmHg")

 , name = "Cuff pressure manipulations")

p+geom\_bracket(xmin="manual", xmax="cuff25"

 , y.position=77

 , label = "\*"

 , label.size = 5

 , vjust = 0.7

 , tip.length = c(0.01,0.01)

) #library: ggpubr



# 4. ANOVA example #

# Cautions: This process does not include

# the basic assumptions validations for the test

#Set Working Directory

setwd("Your working directory here")

#Read CSV file

mydata = read.csv("opioid\_surgery.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

)

head (mydata)

#> case a b c

#> 1 1 565 574 741

#> 2 2 477 519 674

#> 3 3 712 668 853

#> 4 4 325 422 557

#> 5 5 517 544 704

#> 6 6 552 566 731

shapiro.test(mydata$a)

#>

#> Shapiro-Wilk normality test

#>

#> data: mydata$a

#> W = 0.98628, p-value = 0.7636

shapiro.test(mydata$b)

#>

#> Shapiro-Wilk normality test

#>

#> data: mydata$b

#> W = 0.98627, p-value = 0.7631

shapiro.test(mydata$c)

#>

#> Shapiro-Wilk normality test

#>

#> data: mydata$c

#> W = 0.9863, p-value = 0.7648

#Library

library(reshape)

mydata.melt <- melt(mydata, id="case") #library: reshape

head(mydata.melt)

#> case variable value

#> 1 1 a 565

#> 2 2 a 477

#> 3 3 a 712

#> 4 4 a 325

#> 5 5 a 517

#> 6 6 a 552

library(dplyr)

#>

#> Attaching package: 'dplyr'

#> The following object is masked from 'package:reshape':

#>

#> rename

#> The following objects are masked from 'package:stats':

#>

#> filter, lag

#> The following objects are masked from 'package:base':

#>

#> intersect, setdiff, setequal, union

temp <- mydata.melt %>%

 group\_by(variable) %>%

 summarise(mn = mean(value, na.rm = TRUE)

 , sd = sd(value, na.rm = TRUE)

 )

head(temp)

#> # A tibble: 3 x 3

#> variable mn sd

#> <fct> <dbl> <dbl>

#> 1 a 544. 158.

#> 2 b 561. 99.9

#> 3 c 724. 121.

anova.result <- aov(value ~ variable, data = mydata.melt)

summary(anova.result)

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

#> variable 2 1136553 568277 34.51 2.78e-13 \*\*\*

#> Residuals 168 2766386 16467

#> ---

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

TukeyHSD(anova.result)

#> Tukey multiple comparisons of means

#> 95% family-wise confidence level

#>

#> Fit: aov(formula = value ~ variable, data = mydata.melt)

#>

#> $variable

#> diff lwr upr p adj

#> b-a 17.22807 -39.61142 74.06756 0.7539437

#> c-a 180.91228 124.07279 237.75177 0.0000000

#> c-b 163.68421 106.84472 220.52370 0.0000000

library(ggplot2)

ggplot(temp, aes(x=variable, y=mn, group = 1))+

 geom\_errorbar(aes(ymin = mn-sd, ymax=mn+sd), width=0.05)+

 geom\_bar(stat="identity", width = 0.5)+

 theme\_classic()+

 scale\_y\_continuous(limits = c(0,950),

 , name = "Administered dose (mg)"

 , expand = c(0,0)

 , breaks = seq(0,900,by = 100))+

 scale\_x\_discrete(labels = c("Surgery A", "Surgery B", "Surgery C")

 , name = "Surgical types")+

 annotate("text", x=3.0, y=860, label = "\*", size = 7)



# # 4. RM ANOVA example #

#Leaved out the detailed statistical analysis process

#Set Working Directory

setwd("Your working directory here")

#Library

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

library(rstatix)

#>

#> Attaching package: 'rstatix'

#> The following object is masked from 'package:stats':

#>

#> filter

library(ggplot2)

library(gg.gap)

#> Warning: package 'gg.gap' was built under R version 4.1.1

#Read CSV file

mydata = read.csv("dbpmedication.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

 )

head (mydata)

#> case group dbp00 dbp05 dbp10 dbp15 dbp20 dbp25

#> 1 1 Control 85 67 55 47 50 47

#> 2 2 Control 76 70 74 82 80 73

#> 3 3 Control 55 60 50 55 55 56

#> 4 4 Control 80 65 70 58 61 60

#> 5 5 Control 105 100 96 84 79 81

#> 6 6 Control 90 91 95 85 92 85

mydata.long <- mydata %>% #Library "dplyr"

 gather(key="time", value="dbp", dbp00

 , dbp05, dbp10, dbp15, dbp20, dbp25

 ) %>%

 convert\_as\_factor(case, time) #Library: rstatix

# Data transform into long type for analysis and plotting

mydata.long %>% sample\_n\_by(group, time, size = 1)

#> # A tibble: 12 x 4

#> case group time dbp

#> <fct> <chr> <fct> <int>

#> 1 45 Control dbp00 76

#> 2 23 Control dbp05 84

#> 3 33 Control dbp10 53

#> 4 24 Control dbp15 65

#> 5 50 Control dbp20 72

#> 6 14 Control dbp25 103

#> 7 68 Treatment dbp00 75

#> 8 95 Treatment dbp05 88

#> 9 68 Treatment dbp10 59

#> 10 78 Treatment dbp15 108

#> 11 72 Treatment dbp20 64

#> 12 84 Treatment dbp25 94

# Summary statistics for plotting

sum.stat <- mydata.long %>%

 group\_by(group, time) %>%

 get\_summary\_stats(dbp, type = "mean\_sd")

sum.stat

#> # A tibble: 12 x 6

#> group time variable n mean sd

#> <chr> <fct> <chr> <dbl> <dbl> <dbl>

#> 1 Control dbp00 dbp 57 71.1 11.6

#> 2 Control dbp05 dbp 57 70.8 11.9

#> 3 Control dbp10 dbp 57 71.4 13.7

#> 4 Control dbp15 dbp 57 70.2 14.0

#> 5 Control dbp20 dbp 57 68.5 13.8

#> 6 Control dbp25 dbp 57 69.2 12.2

#> 7 Treatment dbp00 dbp 57 73.0 12.2

#> 8 Treatment dbp05 dbp 57 73.5 12.1

#> 9 Treatment dbp10 dbp 57 76.2 13.4

#> 10 Treatment dbp15 dbp 57 78.1 14.2

#> 11 Treatment dbp20 dbp 57 76.6 14.8

#> 12 Treatment dbp25 dbp 57 76.2 14.5

# Plotting

# 1st stage: plotting the whole graph

p <-

 ggplot(sum.stat, aes(x=factor(time), y=mean, group=group))+

 geom\_line(position = position\_dodge(width = 0.1)

 ,aes(linetype=group)

 )+

 geom\_errorbar(aes(x=time, ymin=mean-sd, ymax=mean+sd)

 , width = 0.05

 , position = position\_dodge(width = 0.1)

 )+

 geom\_point(aes(shape=group), size = 2,

 , position = position\_dodge(width = 0.1)

 )+

 theme(panel.background = element\_rect(fill = "white")

 , panel.grid = element\_blank()

 , axis.line = element\_blank()

 ,legend.background = element\_rect(fill = "white"

 , size = 0.5

 , linetype = "solid"

 , colour = "black"

 )

 , legend.key = element\_rect(fill = "white")

 , legend.title = element\_blank()

 , axis.title.x = element\_text(size = 12)

 )+

 ylab("Diastolic blood pressure (mmHg)")+

 scale\_x\_discrete(name = "Time points (sec)"

 ,labels = c("Initial", "5", "10", "15", "20", "25")

 )+

 annotate("text", x=4.1, y=78.5, label = "\*", size = 7)+

 annotate("text", x=5.1, y=77, label = "\*", size = 7)+

 annotate("text", x=6.1, y=76.5, label = "\*", size = 7)+

 annotate("segment", x = 6.1, xend = 6.2

 , y=69.2, yend=69.2,

 colour = "black", size = 0.5)+

 annotate("segment", x = 6.2, xend = 6.2

 , y = 69.2, yend = 76.2

 , colour = "black", size = 0.5)+

 annotate("segment", x = 6.1, xend = 6.2

 , y = 76.2, yend = 76.2

 , colour = "black", size = 0.5)+

 annotate("text", x = 6.3, y = (0.5\*(76.2+69.2)), size = 5

 , label = "†")

print(p)



# Insert gap into Y axis

gg.gap(plot = p

 , segments = c(20,50)

 , ylim = c(0,100)

 , tick\_width = c(10,10)

 , rel\_heights = c(0.25,0.0001, 0.5))

# Insert a legend

add.legend(plot = p

 , margin = c(top = 220, right = 1, bottom = 1, left = 350) )



# # 5. Categorical data analysis #

#Set Working Directory

setwd("Your working directory here")

#Read CSV file

mydata = read.csv("sorethr.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

)

head (mydata)

#> case group intubation sorethr

#> 1 1 N 1 0

#> 2 2 C 0 0

#> 3 3 N 1 0

#> 4 4 C 0 0

#> 5 5 C 1 1

#> 6 6 N 1 0

#Data prep for graph

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

tab.intu <- xtabs(~ group + intubation, mydata)

 #Basic table of successful intubation

tab.intu

#> intubation

#> group 0 1

#> C 21 32

#> N 9 44

tab.sorethr <- xtabs(~ group + sorethr, mydata)

 #Basic table of sorethroat

tab.sorethr

#> sorethr

#> group 0 1

#> C 33 20

#> N 42 11

#Make merged data

mydata.plot <- rbind( #Bind successful intubation and sorethroat by rows

 mydata %>% #From the loaded dataset

 count(group,intubation) %>% #Count by the values of group and intubation, then

 filter(intubation == 1) %>% #Filter out intubation = 0

 data.frame(total= data.frame(margin.table(tab.intu, 1))$Freq) %>%

 #Get the total number from the basic table

 mutate(measure = c("intubation", "intubation")) %>%

 #Make a varible of measuring method: intubation or sorethroat

 select(group, measure, n, total)%>% #Select and arrange the columns

 mutate(Freq = n/total) %>% #Add column of rate

 mutate(ci.up = n/total+1.96\*sqrt((n/total)\*(1-n/total)/total)) %>%

 #Add 95%CI columns

 mutate(ci.dn = n/total-1.96\*sqrt((n/total)\*(1-n/total)/total))

 , mydata %>%

 count(group,sorethr) %>%

 filter(sorethr == 1) %>%

 data.frame(total= data.frame(margin.table(tab.sorethr, 1))$Freq) %>%

 mutate(measure = c("sorethr", "sorethr")) %>%

 select(group, measure, n, total)%>%

 mutate(Freq = n/total) %>%

 mutate(ci.up = n/total+1.96\*sqrt((n/total)\*(1-(n/total))/total)) %>%

 mutate(ci.dn = n/total-1.96\*sqrt((n/total)\*(1-(n/total))/total))

)

#Check the generated data for plotting

mydata.plot

#> group measure n total Freq ci.up ci.dn

#> 1 C intubation 32 53 0.6037736 0.7354557 0.4720914

#> 2 N intubation 44 53 0.8301887 0.9312744 0.7291029

#> 3 C sorethr 20 53 0.3773585 0.5078596 0.2468574

#> 4 N sorethr 11 53 0.2075472 0.3167322 0.0983621

#Plotting

library(ggplot2)

library(ggpubr)

ggplot(mydata.plot, aes(x=measure, y=Freq, fill=group))+

 geom\_bar(stat="identity"

 , position = position\_dodge())+

 geom\_errorbar(aes(ymin=ci.dn, ymax=ci.up), width = .2

 , position = position\_dodge(.9))+

 theme\_classic()+

 scale\_y\_continuous(limits = c(0,1),

 , name = "Incidence rate (%)"

 , expand = c(0,0)

 , breaks = seq(0,1, by = 0.2)

 , labels = scales::percent\_format(accuracy = 1)

 )+

 scale\_x\_discrete(name = "Events"

 ,labels = c("Successful intubation", "Sore throat")

 )+

 geom\_signif(annotation = "\*"

 , y\_position = 0.97

 , xmin = 0.775, xmax = 1.225

 , tip\_length = c(0.25, 0.01)

 , vjust = 0.5

 )+ #Library: ggpubr

 scale\_fill\_grey(name = "Intubation technique"

 , labels = c("Conventional (N=53)", "New (N=53)")

 )+

 theme(legend.position = c(0.8, 0.9)

 , legend.background = element\_rect(color = "black")

 )



# # 6.1.1 Correlation graph #

## Statistical analytic process is not included

#Set Working Directory

setwd("Your working directory here")

#Library

library(ggplot2)

library(ggpubr)

#Read CSV file

mydata = read.csv("pretxlevel.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

 )

head (mydata)

#> Case pretx exposure blevel

#> 1 1 B 2.620 18.1

#> 2 2 B 2.875 17.8

#> 3 3 A 2.320 15.5

#> 4 4 B 3.215 17.3

#> 5 5 C 3.440 19.7

#> 6 6 B 3.460 21.0

#Plotting

ggplot(mydata

 , aes(x=exposure

 , y=blevel

 , group = pretx

 , shape = pretx

 , linetype = pretx

 )

 )+

 geom\_point(aes(group = pretx))+

 geom\_smooth(method = lm, se = FALSE

 , fullrange = FALSE

 , color = "black"

 )+

 theme\_classic()+

 scale\_y\_continuous(limits = c(0,40),

 , name = "Blood level (pg/ml)"

 , expand = c(0,0)

 , breaks = seq(0,40, by = 10)

 )+

 scale\_x\_continuous(limits = c(0, 7)

 , name = "Exposure time (hour)"

 , expand = c(0,0)

 , breaks = seq(0,7, by = 1)

 )+

 theme(legend.background = element\_rect(fill = NA, color = "black")

 ,legend.position = c(.8, .15)

 , legend.margin = margin(0,80,0,10)

 , legend.text = element\_text(size = 11)

 , legend.key.height = unit(5.5, "mm")

 )+

 labs(shape = "Pre-treatment", linetype = "Pre-treatment")+

 stat\_cor(method = "spearman"

 , cor.coef.name = "rho"

 , p.accuracy = 0.001

 , label.x = 5.3

 , label.y.npc = 0.18

 , size = 4

 )+

 annotate("text", x=3.3, y=15, label = "\*", size = 9)

#> `geom\_smooth()` using formula 'y ~ x'



# # 6.1.2 linear regression graph example #

# Regression diagnostic plots are listed in this example.

#Set Working Directory

setwd("Your working directory here")

#Read CSV file

mydata = read.csv("dmmvi.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

)

head (mydata)

#> case bglc glcnod

#> 1 1 279.1 48.5

#> 2 2 59.6 37.6

#> 3 3 25.8 35.3

#> 4 4 188.7 48.1

#> 5 5 223.2 40.7

#> 6 6 16.4 33.6

## Plotting linear regression y = ax + b

library(ggplot2)

ggplot(mydata, aes(x=bglc, y = glcnod))+

 geom\_point()+

 geom\_smooth(method = "lm"

 , se = TRUE

 , colour = "black"

 )+

 theme\_classic()+

 scale\_y\_continuous(limits = c(0,60),

 , name = "Glucose in nodule (mg/mg)"

 , expand = c(0,0)

 , breaks = seq(0,60, by = 10)

 )+

 scale\_x\_continuous(name = "Blood glucose level (mg/dL)"

 , expand = c(0,0)

 , breaks = seq(0,360, by = 20)

 )+

 annotate("text", x = 120, y = 20

 , label = "Glucose in nodule = 0.048 × Blood glucose concentration + 32.98"

 , size = 4

 , hjust = 0

 )+

 annotate("text", x = 225, y = 17

 , label = as.character(expression(paste(R^2,"= 0.603, F[1,198]=300.2, P<0.001")))

 , size = 4

 , parse = TRUE

 , hjust = 0

 )

#> `geom\_smooth()` using formula 'y ~ x'



# # 6.2 Logistic regression #

# Cautions: This process does not include

# the basic assumptions validations for the test

#Set Working Directory

setwd("Your working directory here")

mydata = read.csv("fivefactors.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

)

head (mydata)

#> case f1 f2 f3 f4 f5 test

#> 1 1 1.2 10.4 1.44 44 19.1 0

#> 2 2 2.4 9.9 4.90 38 16.4 0

#> 3 3 1.2 9.7 3.60 44 19.9 0

#> 4 4 14.4 11.8 6.72 119 35.2 1

#> 5 5 3.6 10.0 4.62 59 27.6 0

#> 6 6 1.2 10.3 4.90 74 30.2 0

#Library

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

library(tibble)

log.model <-

 glm( test ~ f1 + f2 + f3 + f4 + f5

 , data = mydata

 , family = binomial)

summary(log.model)

#>

#> Call:

#> glm(formula = test ~ f1 + f2 + f3 + f4 + f5, family = binomial,

#> data = mydata)

#>

#> Deviance Residuals:

#> Min 1Q Median 3Q Max

#> -2.4428 -0.5935 -0.3911 0.5868 2.4395

#>

#> Coefficients:

#> Estimate Std. Error z value Pr(>|z|)

#> (Intercept) -10.963079 2.135669 -5.133 2.85e-07 \*\*\*

#> f1 0.213764 0.052862 4.044 5.26e-05 \*\*\*

#> f2 0.563035 0.174759 3.222 0.00127 \*\*

#> f3 0.092671 0.157971 0.587 0.55745

#> f4 -0.002323 0.009644 -0.241 0.80969

#> f5 0.084583 0.048788 1.734 0.08297 .

#> ---

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

#>

#> (Dispersion parameter for binomial family taken to be 1)

#>

#> Null deviance: 181.53 on 149 degrees of freedom

#> Residual deviance: 130.77 on 144 degrees of freedom

#> AIC: 142.77

#>

#> Number of Fisher Scoring iterations: 5

# Data for plotting: collecting coefficients and its CIs

library(tibble)

log.plot <-

 as.data.frame(exp(cbind(OR = coef(log.model)

 , confint(log.model)

 )

 )

 )

#> Waiting for profiling to be done...

# Data for plotting: Generate a column containing variable names

log.plot <- rownames\_to\_column(log.plot, var = "factors") #Library tibble

# Data for plotting: Remove intercept

log.plot <- log.plot[!(log.plot$factors == "(Intercept)"),]

# Data for plotting: Prevent errors which come from the column names

# of numeric characters

colnames(log.plot) <- c("factors", "OR", "LL", "UL")

log.plot #Checkout the generated data

#> factors OR LL UL

#> 2 f1 1.2383300 1.1216741 1.382200

#> 3 f2 1.7559940 1.2603738 2.513343

#> 4 f3 1.0971010 0.8048928 1.501346

#> 5 f4 0.9976802 0.9785388 1.016564

#> 6 f5 1.0882634 0.9914794 1.202641

#Plotting

library(ggplot2)

ggplot(log.plot, aes(x = OR, y = factors))+

 geom\_pointrange(aes(xmin = LL, xmax = UL))+

 scale\_y\_discrete(name = "Factors"

 , limits=c("f5", "f4", "f3", "f2", "f1")

 )+

 scale\_x\_continuous(limits = c(0, 3)

 , breaks = seq(0, 2.5, by = 0.5)

 , expand = c(0,0)

 , name = "Odds ratio (95% confidence interval)"

 )+

 geom\_vline(xintercept = 1

 , linetype="dotted"

 , color = "grey"

 , size = 1.0

 )+

 theme\_classic()+

 annotate("text", x = 1.4, y = "f1"

 , label = "\*"

 , size = 6

 , hjust = 0

 )+

 annotate("text", x = 2.53, y = "f2"

 , label = "\*"

 , size = 6

 , hjust = 0

 )



# # 6.3 Survival analysis #

# Refer to the statistical round article: In K, Lee DK. Survival analysis: part II #– applied clinical data analysis. Kor J Anesth 2019;72(5):441-57. DOI: #https://doi.org/10.4097/kja.19183

# # 6.4 Dose-response curve #

#Set Working Directory

setwd("Your working directory here")

mydata = read.csv("antiobsorp.csv"

 , header = TRUE

 , fileEncoding = "UTF-8-BOM"

 )

head (mydata)

#> case dose anti obsorp

#> 1 9 2.5 A 0.570

#> 2 10 2.5 A 0.570

#> 3 11 2.5 A 0.570

#> 4 12 2.5 A 0.495

#> 5 13 5.0 A 0.615

#> 6 14 5.0 A 0.525

library(ggplot2)

library(drc)

#> Loading required package: MASS

#>

#> 'drc' has been loaded.

#> Please cite R and 'drc' if used for a publication,

#> for references type 'citation()' and 'citation('drc')'.

#>

#> Attaching package: 'drc'

#> The following objects are masked from 'package:stats':

#>

#> gaussian, getInitial

ggplot(mydata, aes(x = dose

 , y = obsorp

 , group = anti

 , linetype = anti

 )

 )+

 geom\_point(aes(shape=anti))+

 geom\_smooth(method = drm

 , method.args = list(fct=L.4())

 , se= FALSE, color = "black"

 )+

 scale\_x\_log10(breaks = c(1,2.5, 5 , 10, 20, 40, 80, 160)

 , name = "log(Dose (mg))"

 )+

 scale\_y\_continuous(limits = c(0, 0.8)

 , expand = c(0, 0)

 , name = "Absorption rate"

 )+

 theme\_classic()+

 theme(legend.background = element\_rect(fill = NA, color = "black")

 ,legend.position = c(.9, .9)

 )+

 labs(shape = "Antibiotics", linetype = "Antibiotics")+

 geom\_segment(aes(x = 2.5, y = 0.33, xend = 17.197, yend = 0.33))+

 # y value of ED50 = Lower limit + 0.5(Upper - Lower limit)

 # x value of ED50 = each estimated ED50 value

 geom\_segment(aes(x = 17.197, y = 0.33, xend = 17.197, yend = 0))+

 geom\_segment(aes(x = 7.32, y = 0.33, xend = 7.32, yend = 0), linetype = "dashed")

#> `geom\_smooth()` using formula 'y ~ x'

