PART1 Practicing with Plots that We Discussed

PART1: This part is for you to practice every plot that we discussed during the presentation, we will use the practicing data set (dig.csv) and data sets provided by R.

Note, everything after # is a comment, R commands are in blue and gray shaded, you can copy the R commands into R to run.

load and read data sets for plotting

require the datasets provided by R

```
library (" datasets ")
```

check the available data sets

```
data ()
```

Check details of a data set

```
str(iris)
```

Read in external data

we are going to use DIG NHLBT Teaching Dataset as an example, this data set will be used across the workshop

set up work directory first:

```
setwd(“where your dig.csv is ”)

dig<-read.csv("dig.csv")
```

Check details of a data set

```
str(dig)
```

basic plotting by using default graphics tools in R

take a very simple data set Pressure as an example, check the structure of the data set

```
str (pressure)
```

since this data set only have two variables, R just map the first one to x axis and the other to y axis

```
plot (pressure)
```

this is equivalent as

```
plot (pressure$temperature, pressure$pressure)
```

we are going to use this simple example to check the formatting option in default graphics tools in R

Plot Example 1

scatter plot

```
plot (pressure, type="p")
```

scatter plot with different shape, size and color

```
plot (pressure, type="p", pch = 8, cex =0.8, col="red")
```

Plot Example 2

line graph

```
plot (pressure, type="l")
```

line graph with different line type, width adn color

```
plot (pressure, type="l", lty = 3, lwd =2, col="blue")
```

add title

```
plot (pressure,main="Relation" )
```

add text (parameter from statistics analysis or some other annotation)

```
plot ( pressure )
text (150 ,200 , label =" p value = 0.05 ")
```

#plot for multiple groups

```
data(iris) # load iris data
pch.vec <- c(2 ,8 ,21)[iris$Species]
col.vec <- c(2 ,3 ,6)[iris$Species]
plot(iris$Sepal.Length , iris$Sepal.Width ,col = col.vec ,pch=pch.vec, xlab="sepal.length", ylab="sepal.width",main="iris")
legend ("topleft", pch=c(2 ,8 ,21) ,col=c(2 ,3 ,6) ,legend = unique(iris$Species), cex=0.8)
```

#formatting on size and color for title and labeling

```
plot(iris$Sepal.Length , iris$Sepal.Width ,col = col.vec ,pch=pch.vec, xlab="sepal.length", ylab="sepal.width",main="iris")
legend ("topleft", pch=c(2 ,8 ,21) ,col=c(2 ,3 ,6) ,legend = unique(iris$Species), cex=0.8)
```

Plot different Graph types by simple R plot and ggplot2 R package

Install and load R package ggplot2

ggplot2 is one of the most popular graphic packages in R, we are going to practice with it for different types of graphs.

```
install.packages ("ggplot2")
```

load the library

```
library(ggplot2)
```

Scatter Plots

#Scatter plots are frequently used to display the relationship between two continuous variables.

#simple basic plot

```
plot(iris$Sepal.Length,iris$Sepal.Width ) #check ?plot for more options for formatting the plot
```

#scatterplot in ggplot (geom_point())

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_point() # check ?geom_point for more options, following are a few examples of changing the formatting
```

change dot shape (default is #16) # please refer to the slides on shape options

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
 geom_point(shape=21)
```

change dot size (default is 2)

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
 geom_point(shape=21, size=2.5)
```

add labels and change title position

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
 geom_point()+labs(x="Length",y="Width",title="Sepal Length and Width")+
 theme(plot.title = element_text(hjust = 0.5))
```

add lines from a fitted regression model to a scatter plot

basic plot

```
bp <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width))
```

add linear regression line

```
bp + geom_point() + stat_smooth(method=lm)
```

Default will have 95% confidence interval as shown in the shaded region

99% confidence interval

```
bp + geom_point() + stat_smooth(method=lm, level=0.99)
```

No confidence interval

```
bp + geom_point() + stat_smooth(method=lm, se=FALSE)
```

change regression line color

```
bp + geom_point() + stat_smooth(method=lm, se=FALSE, colour="red")
```

Default is the loess (locally weighted polynomial) curve

```
bp + geom_point() + stat_smooth()
```

#equals

```
bp + geom_point() + stat_smooth(method=loess)
```

now what if you want to separate the data points into different groups, for example group by Species

grouping data points by a categorical variable by color and shape

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, shape=Species, colour=Species)) +
 geom_point()
```

add linear regression line

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, shape=Species, colour=Species)) +
 geom_point() +
 geom_smooth(method=lm, se=FALSE)
```

if you don't like the default color and shape, you can set different shape and color for the grouping variables, please refer to the slide on color panel selection

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, shape=Species, colour=Species)) +
 geom_point() +
 scale_colour_brewer(palette="Dark2")+
 scale_shape_manual(values=c(2,8,0))+
 labs(x="Length",y="Width",title="Sepal Length and Width")+
 theme(plot.title = element_text(hjust = 0.5))
```

Line Graphs

Line Graph are used to visualize the trend over time or other continuous variables

#basic R plot

```
plot(pressure$temperature, pressure$pressure, type="l")
```

#add points

```
points(pressure$temperature, pressure$pressure)
```

ggplot (geom_line())

```
ggplot(pressure, aes(x=temperature, y=pressure)) + geom_line()
```

add points (geom_point())

```
ggplot(pressure, aes(x=temperature, y=pressure)) + geom_line() + geom_point()
```

Bar Graphs

Bar Graphs are commonly used to display numeric values (y-axis) for different categories (x-axis)

1. Bar graph for exact value for y

#simple bar graph

```
barplot(BOD$demand, names.arg=BOD$Time, xlab="Time", ylab= "demand", main="BOD", col="blue") # check ?barplot for more formatting options
```

```
# use ggplot (geom_bar())
ggplot(BOD, aes(x=Time, y=demand)) + geom_bar(stat="identity")# check ?geom_bar for more formatting options
# Convert the x variable to a factor, so that it is treated as discrete
ggplot(BOD, aes(x=factor(Time), y=demand)) + geom_bar(stat="identity")
# change bar color and add labels
ggplot(BOD, aes(x=factor(Time), y=demand)) + geom_bar(stat="identity", fill="blue")+
 labs(x="Time", y="demand", title="BOD")+
 theme(plot.title = element_text(hjust = 0.5))
# generating bar graphs for multiple groups
ggplot(ToothGrowth,aes(x=factor(dose),y=len,fill=supp))+
 geom_bar(stat="identity",position="dodge", width=0.5)
# making a stacked bar graph, just by leaving (postion="dodge") out
ggplot(ToothGrowth,aes(x=factor(dose),y=len,fill=supp))+
 geom_bar(stat="identity", width=0.5)
# change width of the bar graph
ggplot(ToothGrowth,aes(x=factor(dose),y=len,fill=supp))+
 geom_bar(stat="identity", width=0.5)+
 scale_fill_brewer(palette="Dark2")
# 2. bar graph for counts of a categorical variable. This is used more frequently than plotting for the exact value for y
# simple R plot
barplot(table(mtcars$cyl), xlab="cyl", ylab="count", main="mtcars")
# ggplot
ggplot(mtcars, aes(x=factor(cyl))) + geom_bar(fill="blue",width=0.5)
# change the format of the bar graph and add labels
ggplot(mtcars, aes(x=factor(cyl))) +
 geom_bar(fill="blue",colour="black", width=0.5)+
 labs(x="cyl", y="count", title="mtcars")+
 theme(plot.title = element_text(hjust = 0.5))
# 3. plot for percentage of event of hospitalization . *** this will be used to answer one of the questions in our practice
ggplot(dig, aes(x= factor(CVD),  group=factor(TRTMT))) +
 geom_bar(aes(y = ..prop.., fill = factor(TRTMT)), stat="count", position="dodge")+
```

```
 geom_text(aes(label = scales::percent(..prop..), y= ..prop.. ),
           stat= "count",position=position_dodge(0.9), vjust = -.5) +
 labs(x="CVD",y = "Percent", fill="treatment") +
 scale_y_continuous(labels=scales::percent)
# 4. plot mean and error bars
ggplot(ToothGrowth, aes(factor(dose), len )) +
 stat_summary(fun.y = mean, geom = "bar", width=0.5, fill="lightgreen") +
 stat_summary(fun.data = mean_cl_normal, geom = "errorbar", width=0.2)+
 labs(x="dose", y="length")
```

Plots to summarize the data distribution

1. Histogram

#Histogram can be used to view the distribution of one-dimensional data

basic plot

```
hist(ChickWeight$weight) # check ?hist for more formatting options
```

Specify number of bins with breaks

```
hist(ChickWeight$weight, breaks=20)
```

#ggplot (geom_histogram)

```
ggplot(ChickWeight,aes(x=weight)) + geom_histogram() # check ?geom_histogram for more formatting options
```

Specify number of bins

```
ggplot(ChickWeight,aes(x=weight)) +
 geom_histogram(bins=20, fill="white", colour="black")
```

histogram of multiple groups

#make the bars NOT stacked, and make them semitransparent

```
ggplot(ChickWeight, aes(x=weight, fill=factor(Diet))) +
 geom_histogram(position="identity")
```

if you remove position="identity", the bars will stacked on top of each other, try leave it out and compare with previous graph

```
ggplot(ChickWeight, aes(x=weight, fill=factor(Diet))) +
 geom_histogram()
```

2. Density Curve

#density curvers can also be used to view the distribution of the data with smoothing (geom_density())

```
ggplot(ChickWeight, aes(x=weight, fill=Diet)) + geom_density() # check ?geom_density for more formatting options
```

3. Boxplot

#simple R plot

```
boxplot(weight~Diet, data=ChickWeight) # check ?boxplot for more formatting options
```

#ggplot (geom_boxplot())

```
ggplot(ChickWeight, aes(x=Diet, y=weight)) + geom_boxplot(fill="lightgreen") # check ?geom_boxplot for more formatting
options
```

add notches to boxplot

```
ggplot(ChickWeight, aes(x=Diet, y=weight)) + geom_boxplot(notch=TRUE)
```

label mean on boxplot

```
ggplot(ChickWeight, aes(x=Diet, y=weight)) + geom_boxplot(fill="lightgreen",notch=TRUE)+
  stat_summary(fun.y="mean",geom="point",  fill="blue", shape=21, size=3)
```

make a violin plot ot compare density estimates of different groups

```
p <- ggplot(ChickWeight, aes(x=Diet, y=weight))
p + geom_violin()
```

include the boxplot in the middle of the violin plot to view the summary of the data

```
p + geom_violin() +
  geom_boxplot(width=.1, fill="lightgreen", outlier.colour=NA) +
  stat_summary(fun.y=mean, geom="point", fill="white", shape=21, size=3)
```

Change the appearance of the above graphs for publication or presentation

polish the graph

#base plot

```
bp <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species, shape=Species))+geom_point()
bp
```

Change to black and white theme

```
bp1<-bp + theme_bw()
bp1
```

```
# remove grid lines
bp2<-bp1 + theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())
bp2
# add labels
bp3<- bp2+labs(x="Length",y="Width",title="Sepal Length and Width")
bp3
# modify title and axis
bp4<- bp3 +
 theme(axis.title.x = element_text(colour="red", size=11,face="bold"),
       axis.text.x = element_text(colour="blue"),
       axis.title.y = element_text(colour="red", size=11,face="bold", angle = 90),
       axis.text.y = element_text(colour="blue"),
       plot.title = element_text(colour="red", size=12, face="bold", hjust=0.5))
bp4
# modify legends
bp4 +
 theme(legend.background = element_rect(fill="grey85", colour="red", size=0.2),
       legend.title = element_text(colour="blue", face="bold", size=11),
       legend.text = element_text(colour="red"),
       legend.key = element_rect(colour="blue", size=0.2))
```

Output for publication or presentation

```
# output to pdf files
# Width and height are in inches for pdf
pdf("myplot.pdf", width=4, height=4)
plot(mtcars$wt, mtcars$mpg)
ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()
dev.off()
# output to pNG/TIFF files
# width and height are in pixels
png("myplot-%d.png", width=400, height=400)
plot(mtcars$wt, mtcars$mpg)
```

```
ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()

dev.off()
```

Other Useful Plots

Pie Charts

#Pie Chart are used to simply check the composition of the data groups

```
pie(table(iris$Species), col=rainbow(3)) # basic R plot, check ?pie for more options
```

Heat Map

heat maps are used frequently used to visualize the level of signal of one variable across different time or groups

scale and center data by columns and convert to matrix

```
data<-as.matrix(scale(mtcars))
```

create heatmap using R plot heatmap function

```
heatmap(data)
```

change color palette

```
palette <- colorRampPalette(c('blue','white'))(100)

heatmap(data, col=palette)

help("heatmap") # check help for more options
```

create enhanced heatmap using gplots package heatmap.2 function

```
install.packages("gplot")

library(gplots)

heatmap.2(data,col=greenred(100),trace='none')

help("heatmap.2") #check help for more options
```

Venn Diagram

#Venn Diagrams are used to check the overlap between different data sets

Note, you perform some R programming to check the overlapping between the datasets and get the numbers for the weight

```
install.packages("Vennerable")

library(Vennerable)

V <- Venn(SetNames=c('A','B','C'),Weight=c(0,10,30,5,20,2,16,1))
```

```
plot(V, doWeights=TRUE,type='circles')
```

Correlation Plot

Correlation plot are used to check the association or similarity among variables

make a correlation matrix

```
mcor<-cor(mtcars)

install.packages("corrplot")

library(corrplot)

corrplot(mcor, method="shade", shade.col=NA, tl.col="black", tl.srt=45) # check ?corrplot for more options
```

PART2. Hands on Exercise and Quiz on Medical Data Set

PART2: This part is for you to take what you have learned and try to generate plots on some practicing medical data

```
library(ggplot2)
```

we are going to use DIG NHLBT Teaching Dataset as an example, this data set will be used across the workshop

load data

```
setwd()

dig <-read.csv("dig.csv")
```

I. read the following questions and identify the graph types you are going to draw

 # Q1: Check the relationship between BMI and Systolic BP
 # Q2: plot the number of patients for different SEX group
 # Q3: use ggplot to check the distribution of age in different treatment group using three different types of plots
 # Q4: A). use ggplot to plot the percentage of death in different treatment group;
 # B). use ggplot plot the percentage of death attributed to worsening heart failure
 # Q5: take plot from Q4B, try to polish the graph by changing the background into black and white, get rid of grid lines, add labels for x, y axis and plot title, adjust font and position for labels, and adjust legends.

 # II. check the clue and refer to the example code to see whether you can modify based on this to draw your plot to the question.

Clue for Q1: Check the relationship between BMI and Systolic BP

graph type: scatter plot for relationship between two continuous variables

Example Code:

simple R plot

```
plot(iris$Sepal.Length,iris$Sepal.Width, xlab="Length", ylab="Width",main = "Sepal Length and Width" )
```

scatterplot in ggplot (geom_point())

```
ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_point()
```

Clue for Q2: plot the number of patients for different SEX group

graph type: bar graph for count of categorical variable

Example Code:

simple R plot

```
barplot(table(mtcars$cyl), xlab="cyl", ylab="count", main="mtcars")
```

ggplot

```
ggplot(mtcars, aes(x=factor(cyl))) + geom_bar()
```

Clue for Q3: use ggplot to visualize the distribution of age in different treatment group using three different types of plots in ggplot

graph type: visualize the distribution of a continuous variable by histogram, density curve and boxplot

Example Code

histogram

```
ggplot(ChickWeight, aes(x=weight, fill=factor(Diet))) + geom_histogram(position="identity", alpha=0.4)
```

density curve (geom_density)

```
ggplot(ChickWeight, aes(x=weight, fill=factor(Diet))) + geom_density()
```

boxplot

```
ggplot(ChickWeight, aes(x=factor(Diet), y=weight)) + geom_boxplot()
```

Clue for Q4: A). use ggplot to plot the percentage of death in different treatment group;

B). use ggplot to plot the percentage of death attributed to worsening heart failure.

graph type: bar graph for percentage of categorical variable.

We have done the following during the practice. Can you modify it to graph for different variables?

Example Code

```
ggplot(dig, aes(x= factor(CVD),  group=factor(TRTMT))) +
 geom_bar(aes(y = ..prop.., fill = factor(TRTMT)), stat="count", position="dodge")+
 geom_text(aes( label = scales::percent(..prop..), y= ..prop.. ), stat= "count",position=position_dodge(0.9), vjust = -.5) +
 labs(x="CVD",y = "Percent", fill="treatment") +
 scale_y_continuous(labels=scales::percent)
```

Clue for Q5: take plot from Q4B in ggplot, try to polish the graph by changing the background into black and white, get rid of grid lines, add labels for x and y axis, plot title and adjust font and position

first run

```
Q4B<-ggplot(dig, aes(x= factor(DWHF),  group=factor(TRTMT))) +
 geom_bar(aes(y = ..prop.., fill = factor(TRTMT)), stat="count", position="dodge")+
 geom_text(aes( label = scales::percent(..prop..), y= ..prop.. ), stat= "count",position=position_dodge(0.9), vjust = -.5) +
 labs(x="DWHF",y = "Percent", fill="treatment") +
 scale_y_continuous(labels=scales::percent)
```

Then change the appearance of the above graphs for publication or presentation by polishing the graph

III. Compare your code and plots with the plots generated by code in Key

Key:

Q1: Check the relationship between BMI and Systolic BP

```
plot(dig$BMI,dig$SYSBP, xlab="BMI", ylab="SYSBP", col="blue")
ggplot(dig, aes(x=BMI, y=SYSBP)) + geom_point(colour="blue", shape=21)
```

Q2: plot the number of patients for different SEX group

```
barplot(table(dig$SEX), col="lightgreen")
ggplot(dig, aes(x=factor(SEX))) + geom_bar( colour="black", fill="lightgreen", width=0.5)
```

```
# Q3 use ggplot to check the distribution of age in different treatment group using three different types of plots
```

histogram

```
ggplot(dig, aes(x=AGE, fill=factor(TRTMT))) + geom_histogram(position="identity")
```

denstity curve

```
ggplot(dig, aes(x=AGE, fill=factor(TRTMT))) + geom_density()
```

boxplot

```
ggplot(dig, aes(x=factor(TRTMT), y=AGE)) + geom_boxplot(notch=TRUE, width=0.5, colour="black", fill="lightgreen")+
 stat_summary(fun.y="mean",geom="point",  fill="white", shape=21, size=3)
```

add a violin plot

```
ggplot(dig, aes(x=factor(TRTMT), y=AGE))  + geom_violin() + geom_boxplot(notch=TRUE, width=0.2, colour="black",
fill="lightgreen")+
 stat_summary(fun.y="mean",geom="point",  fill="white", shape=21, size=3)
```

Q4.A. use ggplot to plot the percentage of death in different treatment group

```
ggplot(dig, aes(x= factor(DEATH),  group=factor(TRTMT))) +
 geom_bar(aes(y = ..prop.., fill = factor(TRTMT)), stat="count", position="dodge")+
 geom_text(aes( label = scales::percent(..prop..), y= ..prop.. ), stat= "count",position=position_dodge(0.9), vjust = -.5) +
 labs(x="DEATH",y = "Percent", fill="treatment") +
 scale_y_continuous(labels=scales::percent)
```

Q4.B. use ggplot to plot the percentage of death attributed to worsening heart failure

```
ggplot(dig, aes(x= factor(DWHF),  group=factor(TRTMT))) +
 geom_bar(aes(y = ..prop.., fill = factor(TRTMT)), stat="count", position="dodge")+
 geom_text(aes( label = scales::percent(..prop..), y= ..prop.. ), stat= "count",position=position_dodge(0.9), vjust = -.5) +
 labs(x="DWHF",y = "Percent", fill="treatment") +
 scale_y_continuous(labels=scales::percent)
```

Q5: take plot from Q4B, try to polish the graph by changing the background into black and white, get rid of grid lines, add labels for x, y axis and plot title, adjust font and position for labels, and adjust legends.

```
Q4B<-ggplot(dig, aes(x= factor(DWHF),  group=factor(TRTMT))) +
 geom_bar(aes(y = ..prop.., fill = factor(TRTMT)), stat="count", position="dodge")+
 geom_text(aes( label = scales::percent(..prop..), y= ..prop.. ), stat= "count",position=position_dodge(0.9), vjust = -.5) +
 labs(x="DWHF",y = "Percent", fill="treatment") +
 scale_y_continuous(labels=scales::percent)
Q4B+
 theme_bw()+
 theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
 labs(x="DWHF", y="Percentage", title="Percentage of DWHF in Different Treatment Group") +
```

```
theme(axis.title.x = element_text(colour="red", size=11,face="bold"),
    axis.text.x = element_text(colour="blue"),
    axis.title.y = element_text(colour="red", size=11,face="bold", angle = 90),
    axis.text.y = element_text(colour="blue"),
    plot.title = element_text(colour="red", size=12, face="bold", hjust=0.5)) +
theme(legend.background = element_rect(fill="grey85", colour="red", size=0.2),
    legend.title = element_text(colour="blue", face="bold", size=11),
    legend.text = element_text(colour="red"))
```