Workshop 1

Introduction to R in statistical analysis

Lulu Shang

Department of Biostatistics
MD Anderson Cancer Center

Introduction to Bioinformatics (GS011143)

Basic R Operations and Concept

Arithmetic

```
2 + 3 # add

## [1] 5

7^8 # 7 to the 8th power

## [1] 5764801

4 * 5 / 6 # multiply and divide

## [1] 3.333333

sqrt(2)

## [1] 1.414214
```

Basic R Operations and Concepts

We can change the number of digits displayed with options:

```
options(digits = 16)
sqrt(2)

## [1] 1.414213562373095

log2(1:4)

## [1] 0.000000000000000 1.000000000000000 1.584962500721156 2

options(digits = 7) # back to default
0/0; 0/100; 100/0

## [1] NaN
## [1] 0
## [1] Inf
```

Assignment, Object names, and Data types

```
x <- 7*41/pi # don't see the calculated value
y <- 7*41/pi # don't see the calculated value
x # take a look!

## [1] 91.35494

y

## [1] 91.35494

x==y

## [1] TRUE
```

Assignment, Object names, and Data types

For assignment, both ¡- and = work. It is recommended that use ¡- for assignment of values to variables. = is used to set options in functions.

```
typeof(x);typeof(pi);typeof("HH")

## [1] "double"
## [1] "double"
## [1] "character"

sqrt(-1) # isn't defined

## Warning in sqrt(-1):  NaNs produced

## [1] NaN

sqrt(as.complex(-1)) # is defined

## [1] 0+1i
```

Vectors

```
(x1 <- LETTERS)

##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N
## [20] "T" "U" "V" "W" "X" "Y" "Z"

(x2 <- 1:10)

##  [1]  1  2  3  4  5  6  7  8  9 10

(x3 <- rep(1,10))

##  [1] 1 1 1 1 1 1 1 1 1 1

(x4 <- rep(x2,2))

##  [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8
```

Indexing Vectors

```
x <- LETTERS
x[3:10]

## [1] "C" "D" "E" "F" "G" "H" "I" "J"

x[-(3:10)]

##  [1] "A" "B" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V

x[x=="H"] <- "HH" # find "H" and replace "HH"
```

Functions and Expressions

```
x <- 1:10
y <- c(x,rep(NA,2)) # add missing values, NAs
mean(x)

## [1] 5.5

mean(y) # produce missing value

## [1] NA

mean(y,na.rm=T) # remove NAs, then mean

## [1] 5.5

min(x); min(y,na.rm=T)

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

Getting Help

There are many ways to getting help when you are using R. The simplest way is type ?name to see the help file for the function.

```
?summary
?rep
```

Displaying Continuous Data

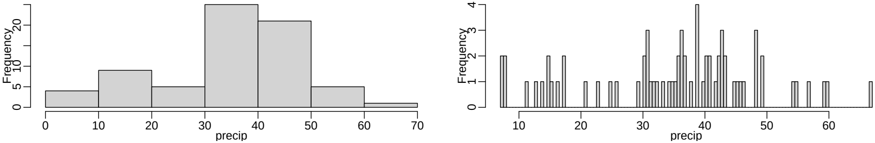
Download the average amount of rainfall data in inches for each of 70 United States (and Puerto Rico) cities from R.

```
data(precip)
?precip
summary(precip)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   29.38   36.60   34.89   42.77   67.00
```

Displaying Continuous Data

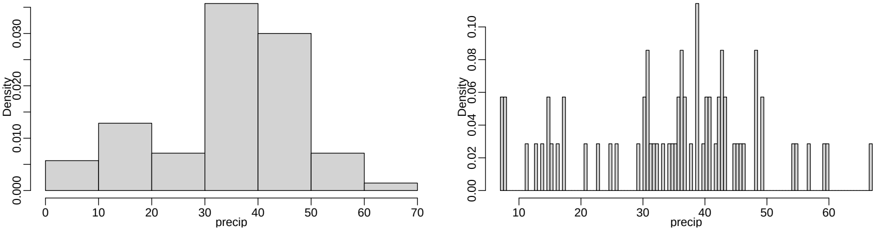
For continuous data, histogram is a standard way for displaying the empirical distribution. The histograms using hist function can be displayed in two ways: (1) frequency (by default) and (2) density/relative frequency (by setting freq=F). Check how much the histograms look different by the bin size.

```
par(mar = c(2.1, 3.5, 1, 0.3), mgp = c(1.2, 0.5, 0),
mfrow=c(1,2)) # to display two histograms side-by-side
hist(precip,main="");hist(precip,breaks=100,main="") # use ma
```



Displaying Continuous Data

```
par(mar = c(2.1, 3.5, 1, 0.3), mgp = c(1.2, 0.5, 0),
mfrow=c(1,2))
hist(precip,main="",freq=F);hist(precip,breaks=100,main="",
freq=F)
```



Displaying Discrete Data

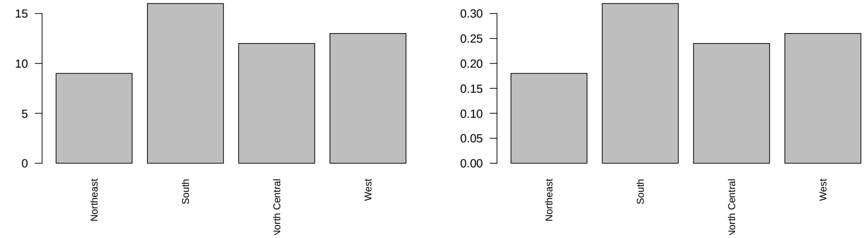
One of the best way to summarize qualitative/discrete data is with a table of the data values. We use data sets related to the 50 states of the United States of America. We can display table with frequencies or proportions.

```
ta <- table(state.region)
ta

## state.region
##     Northeast         South North Central           West
##             9            16            12             13

ta/sum(ta) # proportions/relative frequencies

## state.region
##     Northeast         South North Central           West
##          0.18          0.32          0.24           0.26

prop.table(ta) # same thing
```

Displaying Discrete Data

A bar graph is the analogue of a histogram for categorical data.

```
par(mfrow=c(1,2),mar=c(5,4.5,1,1))
# to display two plots side-by-side and control margins
barplot(table(state.region), cex.names = 0.8,las=2)
# check options!
barplot(prop.table(table(state.region)), cex.names =0.8,las=2
```



```
# check options!
```

Other Data Types in R

A logical value is either TRUE or FALSE (note that equivalently you can use TRUE=T=1, FALSE=F=0). Here is the example of a logical vector:

```
x <- 1:10
y1 <- x<5
y1

##  [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE

!y1 # you can swap

##  [1] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE

y2 <- x!=10
y2

##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

Other Data Types in R

When you perform statistical analyses, missing values denoted by NA in R should be handled carefully.

```
x <- c(3, 7, NA, 4, 7)
y <- c(5, NA, 1, 2, 2)
is.na(x)

## [1] FALSE FALSE  TRUE FALSE FALSE

is.na(y)

## [1] FALSE  TRUE FALSE FALSE FALSE

which(is.na(x)) # indices for missing values

## [1] 3

which(is.na(y))

## [1] 2
```

Normal Distribution

A continuous random variable, X takes values an inteval of real numbers. The random variable X that follows Normal distribution with mean μ and variance σ^2, $N(\mu, \sigma^2)$ has the density function:

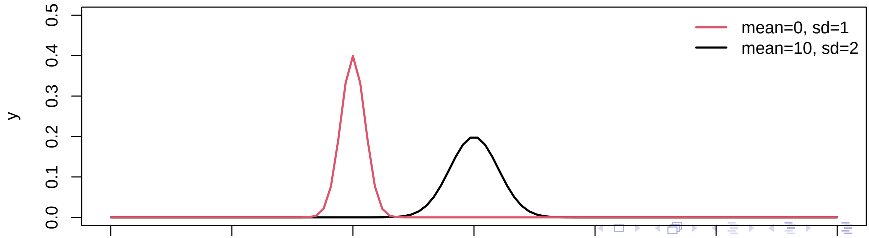
$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Let's draw the normal density with μ and σ (called standard normal) using dnorm function with options mean for μ and sd for σ.

```
x = seq(-20,40,length=100)
y = dnorm(x,mean=10,sd=2) # compute density for each x
```

Normal Distribution

```r
x = seq(-20,40,length=100)
y = dnorm(x,mean=10,sd=2) # compute density for each x
plot(x,y,type="l",ylim=c(0,0.5),lwd=2)
x = seq(-20,40,length=100)
y = dnorm(x,mean=0,sd=1) # standard normal
lines(x,y,lwd=2,col=2)
legend("topright",legend=c("mean=0, sd=1","mean=10, sd=2"),col
```



Normal Distribution

The (cumulative) distribution function is defined as the probability that X will take a value less than or equal to q, $P(X \leq q) = p$. In other words, the left tail area up to q under the density curve.

```
pnorm(q=-1.96,mean=0,sd=1) # 2.5% left tail prob.

## [1] 0.0249979

1-pnorm(q=1.96,mean=0,sd=1) # 2.5% right tail prob.

## [1] 0.0249979

pnorm(q=0,mean=0,sd=1)

## [1] 0.5

pnorm(q=0,mean=0,sd=2)

## [1] 0.5
```

Normal Distribution

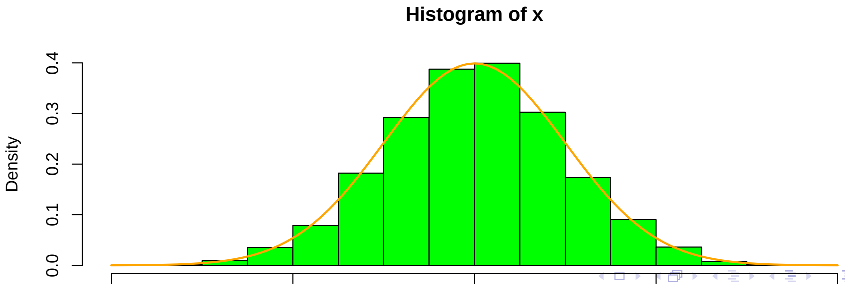
The quantile function is defined as the value (q) of X such that the probability of the variable being less than that value (q) equals the given probability (p).

```
qnorm(p=0.025,mean=0,sd=1) # 2.5% left tail quantile

## [1] -1.959964

qnorm(p=1-0.025,mean=0,sd=1) # 2.5% right tail quantile

## [1] 1.959964

pnorm(q=qnorm(p=0.025,mean=0,sd=1),mean=0,sd=1)

## [1] 0.025
```

Normal Distribution

We can generate random numbers from normal distributions using rnorm function.

```
x = rnorm(n=10000)
# 10000 values are generated from standard normal
hist(x,freq=F,col="green")
# Histogram with relative frequencies
curve(dnorm,add=T,col="orange",lwd=2)
```



Standard Normal distribution and Z-score

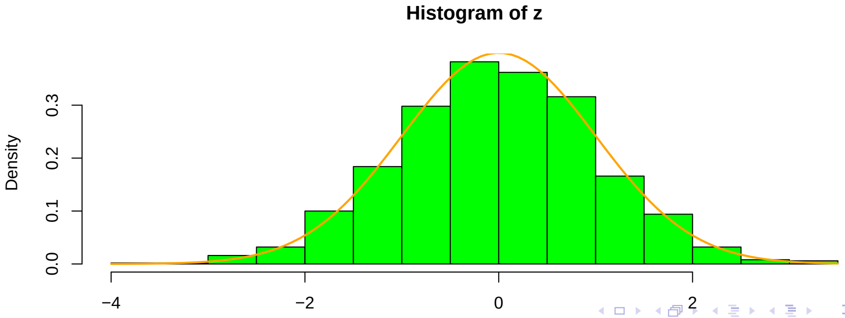
Let X follows $N(\mu, \sigma^2)$. If $\mu = 0$ and $\sigma = 1$, then the distribution is called standard normal distribution. Random variables that follows any normal distributions can be transformed into random variable that follows standard normal distribution. Let's assume that Z follows standard normal distribution, $N(\mu = 0, \sigma^2 = 1)$. We can transform X into Z as follows:

$$Z = X - \mu.$$

Using the formula, we can transform normally distributed values to those that follows standard normal if we know the μ and σ. Equivalently, we can say $X = \mu + Z\sigma$.

Standard Normal distribution and Z-score

```
x <- rnorm(1000,mean=10,sd=2)
# random numbers with mean 10 and sd 2.
z <- (x - 10)/2 # Z-scores
hist(z,freq=F,col="green")
# Histogram with relative frequencies
curve(dnorm,add=T,col="orange",lwd=2)
```



Standard Normal distribution and Z-score

Thanks to this property, we can use standard normal distribution to find distribution/quantiles for any normal distributions.

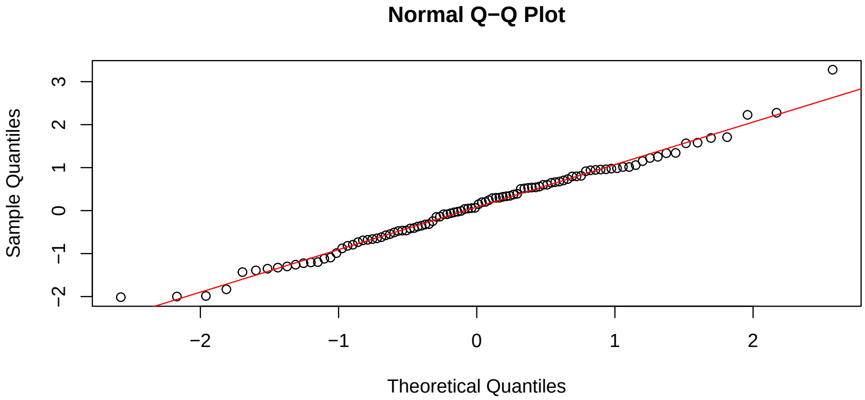
```
pnorm(q=-0.96,mean=1,sd=5)

## [1] 0.3475291

pnorm(q = (-0.96-1)/5,mean=0,sd=1)

## [1] 0.3475291

z = qnorm(p=0.025,mean=0,sd=1)
z*5 +2 # This is the 2.5% quantile for N(2,5^2)

## [1] -7.79982

qnorm(p=0.025,mean=2,sd=5)

## [1] -7.79982
```

Q-Q plot

- Q-Q (Quantile-Quantile) plot is designed to compare two probability distributions by plotting their quantiles against each other.
- Many statistical models are developed under normality assumption and this is useful for checking the assumption. Q-Q plot for normality check is called normal Q-Q plot: it is simply a scatterplot for quantiles from data versus normal distribution (theoretical).
- Note that the dots in the scatterplot should be follow the diagonal line if the empirical distribution is normal of any mean and standard deviation.

Q-Q plot

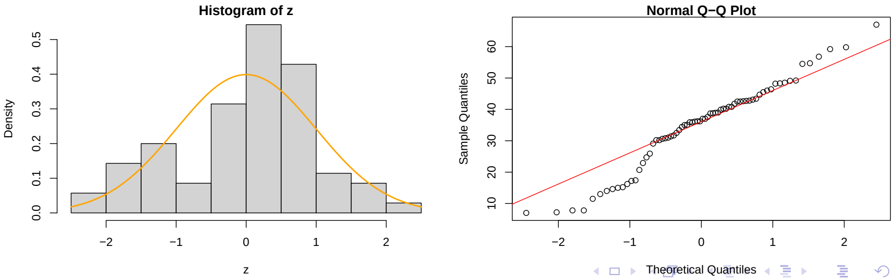
```r
z <- rnorm(100) # normal random numbers
qqnorm(z)
qqline(z,col="red")
```



Q-Q plot

Let's check real data. You see deviations at the tail from normal distribution.

```r
par(mfrow=c(1,2),mar=c(4.5,4.5,1,1))
z = (precip - mean(precip)) /sd(precip)
# compute z-scores for precip data
hist(z,freq=F)
curve(dnorm,add=T,col="orange",lwd=2)
qqnorm(precip);qqline(precip,col="red")
```



Sampling Distributions

- The probability distribution with the population is called the population distribution, and the probability distribution associated with statistic for samples from the population is called its sampling distribution.
- Our goal is to learn about population from statistic and its sampling distribution; this is statistical inference!

Distribution of Sample Means

In our previous examples, we drew random samples from a normal distribution with mean μ and standard deviation (sd) σ. This is “drawing a sample of size 500 from normal distribution with mean 10 and sd 2” if you use R command “rnorm(n=500,mean=10, sd=2)”. The sample mean, $\frac{1}{n}\sum_{i=1}^{n} X_i$, for n independent random samples $X_1, \cdots X_n$ from $N(\mu, \sigma^2)$ is a point estimator for the population mean μ.

```
x1 = rnorm(n=500,mean=10,sd=2) # Trial 1
mean(x1) # The sample mean

## [1] 9.981264

x2 = rnorm(n=500,mean=10,sd=2) # Trial 2
mean(x2) # The sample mean

## [1] 9.836566
```

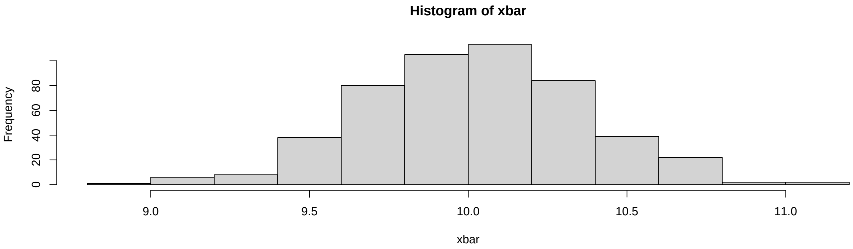
Distribution of Sample Means

Whenever sampling from the same normal distribution, we have different means. The next question to ask is “what happens if we do this repeatedly”. Let’s repeat!

```r
mu = 10 # set the population mean
sigma = 2 # set the population sd
n = 30 # each time we draw a sample of size 30
xbar = rep(0,500) # Initialize the vector of sample means. We
for (i in 1:500) {
    xbar[i] <- mean(rnorm(n=n,mean=mu,sd=sigma))
}
```

Distribution of Sample Means

```r
mu = 10 # set the population mean
sigma = 2 # set the population sd
n = 30 # each time we draw a sample of size 30
xbar = rep(0,500) # Initialize the vector of sample means. We
for (i in 1:500) {xbar[i] <- mean(rnorm(n=n,mean=mu,sd=sigma))
hist(xbar,freq=T)
```



Distribution of Sample Means

- Check if the Histogram looks like a normal distribution. Play with code by increasing the samples size n. You will see the sample means are getting close to the population mean μ as n increases. Why? check the sample standard deviations.
- In reality, the distribution of female adult heights are normal? We do not know. The central limit theorem (CLT) provides the most important theoretical basis for statistical inference. In formal way, the CLT is as follows:
- Let $X_1, \cdots, X_n$ be a random sample of size n from a population distribution with mean μ and finite standard deviation σ. Then the sampling distribution of $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ approaches to normal distribution with mean μ and standard deviation $\sigma/\sqrt{n}$ as $n \to \infty$.

Distribution of Sample Means

Notice that the shape of the underlying population distribution is not mentioned in CLT. The result is true for any population that is well-behaved enough to have a finite standard deviations. So now we can go back to Z-scores. We can construct Z-score based on the CLT:

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

that approximately follows standard normal distribution, $N(0,1)$. This is called Z-statistic.

Confidence Interval

- Confidence level refers to the percentage of all possible samples that can be expected to include true population parameter. Suppose we sample all possible samples from the same populations and compute confidence intervals for each sample. A 95% confidence level implies that 95% of the confidence intervals would include the true population parameter. 5% won't contain the parameter.
- Now we want to know how good is our mean estimator. We know that the Z statistic follows standard normal distribution and want to have a big confidence level, say $95\% = 100(1-\alpha)\%$. We find the lower and upper quantiles that has left and right tail proabilities of $0.025 (= \alpha/2)$, in other words,

$$\Pr\left(-z_{\alpha/2} < \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} < z_{\alpha/2}\right) = 1 - \alpha.$$

Confidence Interval

After some algebra, this is equivalent to

$$\Pr\left(\bar{X} - z_{\alpha/2}\sigma/\sqrt{n} < \mu < \bar{X} + z_{\alpha/2}\sigma/\sqrt{n}\right) = 1 - \alpha.$$

Because normal distribution is symmetric around the mean, $\Pr(Z > z_{\alpha/2}) = \Pr(Z < -z_{\alpha/2})$. Let's see the plant growth example.

```
dim(PlantGrowth) # n=30

## [1] 30  2

n = nrow(PlantGrowth)
xbar = mean(PlantGrowth[,"weight"]) # sample mean
sigma = sd(PlantGrowth[,"weight"]) # sample standard deviatio
z = qnorm(1-0.025)
lower = xbar - z * sigma/sqrt(n); upper = xbar + z * sigma/sq
paste0("(", lower, ",", upper, ")")

## [1] "(4.82208633235243,5.32391366764757)"
```

Confidence Interval

- The 95% CI for the population mean μ is $[4.82, 5.32]$. We are 95% confident that the interval covers μ.
- As you see from this example, you used the sample standard deviation for σ because we do not know σ, under the assumption that the sample sd is a good point estimate of σ.
- However, for small samples (say, $n < 30$), sample sd is not a good estimator. In that case, $t-$distribution with a degree of freedom $n-1$ is used for inference. The $100(1-\alpha)\%$ CI is re-written as

$$\bar{X} \pm t_{\alpha/2}(df = n-1)\frac{S}{\sqrt{n}},$$

where S is the sample standard deviation.

Confidence Interval

Let's compute the 95% CI for the plant growth data.

```
t = qt(p=1-0.025,df=n-1)
lower = xbar - t * sigma/sqrt(n)
upper = xbar + t * sigma/sqrt(n)
lower

## [1] 4.811171

upper

## [1] 5.334829
```

Now we have slightly wider 95% CI, $[4.81, 5.33]$, which reflects the uncertainty esimating σ.