S1Y Lab 3: Probability and the normal distribution

Contents

1 Welcome to Lab 3 **5**

2 Part 1 - Exploring crime data **7**

3 Part 2 - Probability **17**

4 Group Exercises **35**

Chapter 1

Welcome to Lab 3

1.0.1 Learning Outcomes

In this lab, you will investigate *probability* via simulation studies and learn how to compute probabilities from specific probability distributions. Particular focus will be given to the *normal distribution*, which forms the core of many statistical analyses. You will learn how to generate random numbers and use graphical tools to assess whether data can be assumed to be normally distributed.

The first part of this lab forms an extension of Lab 1. Part 2 is based on the H5P materials (P1-P12). Please refer to the materials to help you within this lab if needed. You will explore and visualise the data using the packages `dplyr` and `ggplot2`, and you can use the following lines of code to install the packages once on your device, then load them in each session:

```
#only needed once on your machine
install.packages("dplyr")
install.packages("ggplot2")

#needed everytime you open RStudio
library(dplyr)
library(ggplot2)
```

Chapter 2

Part 1 - Exploring crime data

Here, you will look at crime data from the British Crime Survey (2007-2008). In practice, such data might be helpful to identify groups in the population who are particularly vulnerable to crime or draw inferences for city planners to find ways of making citizens feel safer. Formal hypothesis tests for differences in proportions for different groups will be introduced later in the semester. Still, here, you will get an idea of how to formulate hypotheses in the first place.

To get started, save the data in the same folder as the `R` file you use to work on this lab (in `RStudio`), then set your working directory to that folder and load the data:

```
#set working directory to source file location
#load the RDS file:
crimedata <- readRDS("crimedata.rds")
```

Get an impression of the dataset by looking at the first 6 rows. Recall that you can do that by calling the function `head()`:

```
head(crimedata)
```

Most variables in the dataset should be easily interpretable, but here is a brief explanation for some of the less intuitive ones:

• **deprivation quintile** : Index of multiple deprivation by quintile in England (1=20% most deprived wards)

• **walkdark** : Answer to the question "How safe do you feel walking alone after dark?".

• **wburgl** : Answer to the question "How worried are you about having your home broken into?".

• **wmugged** : Answer to the question "How worried are you about being mugged and robbed?".

• **victim** : Indicates whether or not someone was a victim of crime in the last 12 months.

Take a closer look at the crime dataset, then answer the following questions…

What type of variable is age?

Hint

Here, age is recorded in full years.

- (A) Numerical, continuous
- (B) Numerical, discrete
- (C) Categorical, nominal
- (D) Categorical, ordinal

What type of variable is ethnicity?

Hint

There are different groups of ethnicities and it is not plausible to put them in a particular ranking.

- (A) Numerical, continuous
- (B) Numerical, discrete
- (C) Categorical, nominal
- (D) Categorical, ordinal

What type of variable is years_in_area?

Hint

There are different groups, and they can be ranked (for example, someone who lived in an area "10 years but less than 20 years" has lived there longer than someone who has lived there "2 years but less than 3 years").

- (A) Numerical, continuous
- (B) Numerical, discrete

- (C) Categorical, nominal
- (D) Categorical, ordinal

What type of variable is sex?

Hint

There are two sexes (female, male) and it is not plausible to put them in a particular ranking.

- (A) Numerical, continuous
- (B) Numerical, discrete
- (C) Categorical, nominal
- (D) Categorical, ordinal

What type of variable is walkdark?

Hint

There are different groups and they can be ranked. For example, someone who feels "very unsafe" walking alone after dark feels less safe than someone who feels "a bit unsafe".

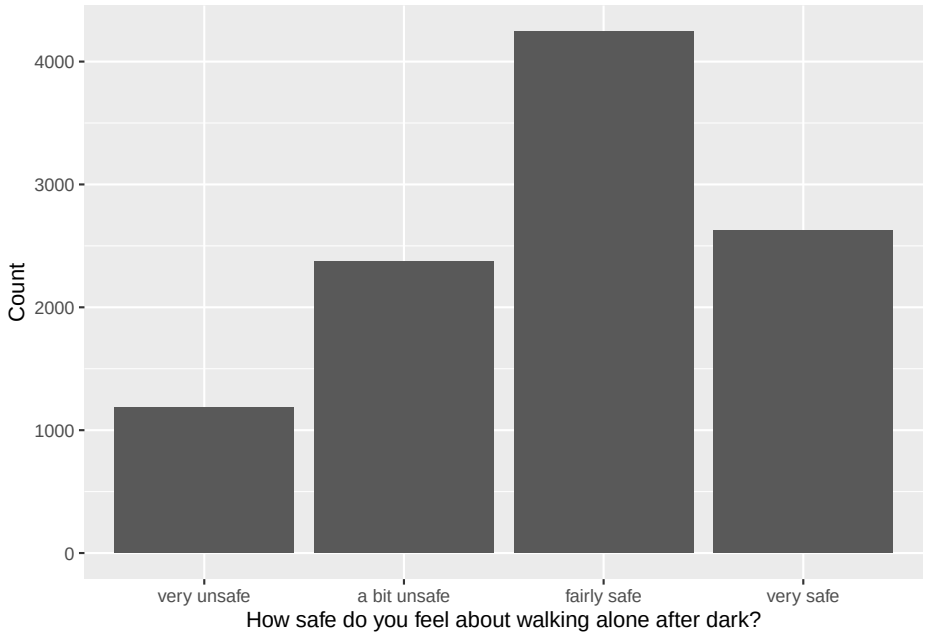
- (A) Numerical, continuous
- (B) Numerical, discrete
- (C) Categorical, nominal
- (D) Categorical, ordinal

The data contain 10,427 observations, which are responses from individuals to 11 questions. Since the dataset is fairly large, you can create plots to get an impression of the data and then compute frequencies as a best guess to the true population proportions for a first informal check of possible associations between variables.

2.0.1 Plotting the data

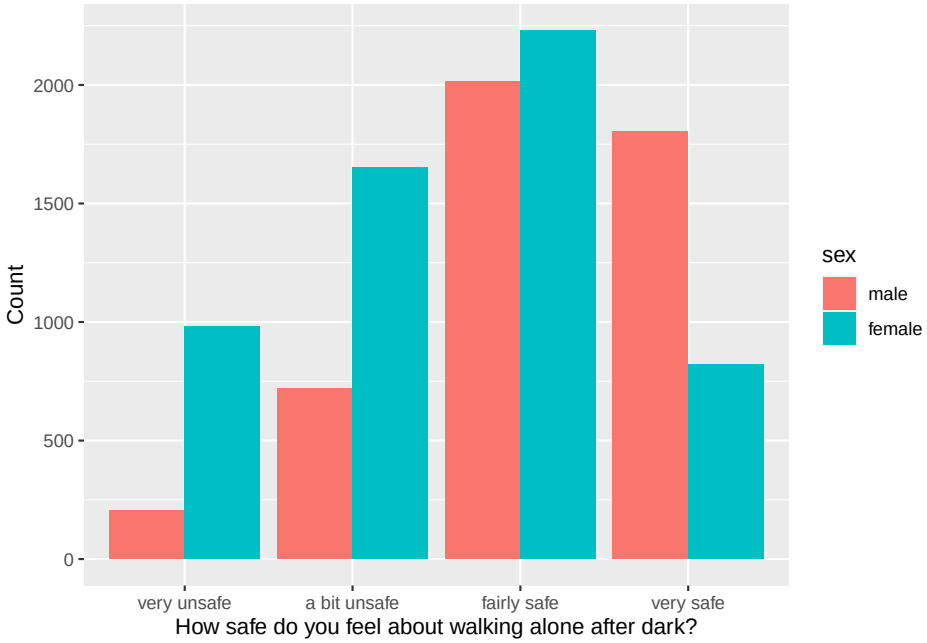
You might be interested in how safe people feel walking alone after dark. You can create a barplot for that variable by writing the code below:

```
ggplot(data=crimedata,aes(walkdark)) + geom_bar() + xlab("How safe do you feel about wa
```



A lot of people seem to feel at least fairly safe when walking alone after dark, but how does that response look like for different groups of people? Below is an example showing you how to separate the responses by sex:

```
ggplot(data=crimedata,aes(walkdark,fill=sex)) + geom_bar(position="dodge") + xlab("How
```



Note that in the code above, the function `geom_bar()` creates a barplot and the argument `position="dodge"` specifies that the bars are supposed to appear side-by-side. Choosing `position="fill"` would give you a stacked barplot instead.

How do you interpret the barplot of responses to walking alone after dark, by females and males?

- (A) Females tend to feel less safe than males walking alone after dark.
- (B) Males tend to feel less safe than females walking alone after dark.
- (C) The two sexes seem to feel similar about walking alone after dark.

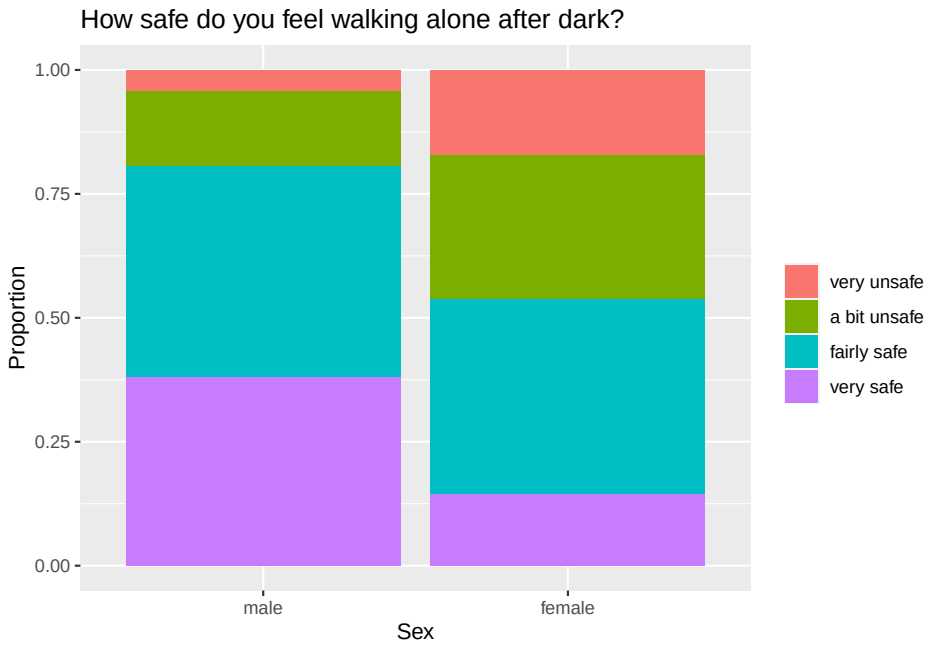
Note that the plot shows the counts for each sex side-by-side. Since the number of female and male respondents might not be the same, it would be useful to know how many of each sex responded to the survey. You can find out via the function `table()`, as presented below:

```
table(crimedata$sex)
```

```
##
##   male female
##   4743   5684
```

The survey contains quite a few more responses from females than males. Hence, the following plot might give a better insight to how the two sexes feel about walking alone after dark:

```
#the argument position="fill" gives you a stacked barplot
ggplot(data=crimedata,aes(sex,fill=walkdark)) + geom_bar(position="fill") + xlab("Sex")
```



Now, you can repeat the steps above for the variables `deprivation_quintile`, `years_in_area`, `marital_status` and `urban_rural`, then answer the questions below.

How would you describe a possible relationship between the deprivation quantiles and how safe people feel walking alone after dark?

- (A) It seems that the more deprived an area is, the more unsafe people feel about walking alone after dark.
- (B) There is no apparent relationship between deprivation and how safe people feel walking alone after dark.
- (C) It seems that the less deprived an area is, the more unsafe people feel about walking alone after dark.

Solution

```
ggplot(data=crimedata,aes(deprivation_quintile,fill=walkdark)) + geom_bar(position="fi
  xlab("Deprivation Quintile") + ylab("Proportion") +
  labs(fill='') + ggtitle("How safe do you feel walking alone after dark?") + theme(ax
```

How would you describe a possible relationship between the number

of years people have lived in an area and how safe they feel walking alone after dark?

- (A) It seems that the longer a person has lived in an area, the less safe they feel about walking alone after dark.
- (B) It seems that the longer a person has lived in an area, the more safe they feel about walking alone after dark.
- (C) There is no apparent relationship between the number of years lived in an area and how safe people feel walking alone after dark.

Solution

```
ggplot(data=crimedata,aes(years_in_area,fill=walkdark)) + geom_bar(position="fill") +
  xlab("Years in the area") + ylab("Proportion") +
  labs(fill='') + ggtitle("How safe do you feel walking alone after dark?")
```

What can you say about possible relationships between a person's marital status and how safe they feel walking alone after dark?

- (A) It seems that widowed people might feel more unsafe than the other groups.
- (B) It seems that single people feel more safe than the other groups.
- (C) It seems that married people feel more unsafe than single people.
- (D) There are no apparent differences by marital status for how safe people feel walking alone after dark.

Solution

```
ggplot(data=crimedata,aes(marital_status,fill=walkdark)) + geom_bar(position="fill") +
  xlab("Marital Status") + ylab("Proportion") +
  labs(fill='') + ggtitle("How safe do you feel walking alone after dark?")
```

How do urban and rural areas compare for how safe people feel walking alone after dark?

- (A) It seems that people in rural areas feel more safe.
- (B) It seems that people in urban areas feel more safe.
- (C) People in urban and rural areas feel similarly safe walking alone after dark.

Solution

```
ggplot(data=crimedata,aes(urban_rural,fill=walkdark)) + geom_bar(position="fill") +
  xlab("Urban/Rural") + ylab("Proportion") +
  labs(fill='') + ggtitle("How safe do you feel walking alone after dark?")
```

To compare responses by group, it might be helpful to obtain numerical summaries. For example, the code below shows you the number of males and females who live in urban and rural areas, respectively.

```
table(crimedata$sex,crimedata$urban_rural)
```

```
##
##          urban rural
##   male    3604  1139
##   female  4342  1342
```

You can compute frequencies from the table above using the following code:

```
sexurb <- table(crimedata$sex,crimedata$urban_rural)
#proportion of males living in urban areas:
#value from first row and column, divided by sum of first row
sexurb[1,1]/sum(sexurb[1,])
```

```
## [1] 0.7598566
```

```
#proportion of females living in urban areas:
sexurb[2,1]/sum(sexurb[2,])
```

```
## [1] 0.7638987
```

Alternatively, you can first split the data into different groups and then compute the frequencies for the different datasets:

```
#create a dataset for females only
fem <- crimedata %>% filter(sex=="female")
#then divide the number of females living in urban areas by the number of females in t
sum(fem$urban_rural=="urban")/nrow(fem)
```

```
## [1] 0.7638987
```

You randomly select an individual from the crime data. What is the probability that the individual is married? (Round to 4 decimal places)

Hint

Since you select the individual randomly, this probability can be computed as the number of married individuals divided by the total number of individuals.

Solution

```
#solution
marital <- table(crimedata$marital_status)
#probability of randomly selected individual being married:
marital[1]/sum(marital)
```

```
##   married
## 0.4753045
```

You randomly select a female from the crime data. What is the probability she is feeling very unsafe walking alone after dark?

Hint

You randomly select an individual and you already know that the individual is female. Hence, the probability of her feeling very unsafe is the same as the proportion of females who feel very unsafe walking alone after dark.

Solution

```
#solution
femdark <- table(crimedata$sex,crimedata$walkdark)
#probability of randomly selected female feeling very unsafe:
femdark[2,1]/sum(femdark[2,])
```

```
## [1] 0.1727657
```

Compute the proportions of widowed females and widowed males who feel very unsafe walking alone after dark. Without performing a formal hypothesis test, would you suggest that these proportions differ substantially?

- (A) Yes, the proportion of people who feel very unsafe walking alone after dark appears to be higher in widowed females than widowed males.

- (B) Yes, the proportion of people who feel very unsafe walking alone after dark appears to be higher in widowed males than widowed females.

- (C) No, the proportions are about the same.

Solution

```
#solution to the multiple choice question above
femwidowed <- crimedata %>% filter(sex=="female" & marital_status=="widowed")
#proportions of widowed females feeling very unsafe walking alone after dark
sum(femwidowed$walkdark=="very unsafe")/nrow(femwidowed)
```

```
## [1] 0.3033839
```

```
malwidowed <- crimedata %>% filter(sex=="male" & marital_status=="widowed")
#proportions of widowed females feeling very unsafe walking alone after dark
sum(malwidowed$walkdark=="very unsafe")/nrow(malwidowed)
```

```
## [1] 0.09459459
```

Hopefully, this part of the lab gave you some good ideas how to use plots to gain insights for a given dataset. If you want, you can explore the crime data further by checking possible associations between other variables in the data.

Chapter 3

Part 2 - Probability

3.0.1 The normal distribution

In the lecture, you were introduced to the normal distribution and how to use the Z-table to look up cumulative proabilities or percentiles of the standard normal distribution $N(\mu = 0, \sigma = 1)$. These tables were particularly useful before computers were invented or whenever they are unavailable to you (for example, when you are stranded on an island… or when you are sitting a closed-book exam).

In R, you can compute cumulative probabilities and percentiles directly from the normal distribution, rather than having to approximate them using the tables. You can get an overview of the different functions available by calling the help function:

```
?Normal
```

If you want to compute the cumulative probability of a variable $X \sim N(\mu = 3, \sigma = 2)$, e.g. $P(X \leq 2)$, this can be done using the following line of code:

```
pnorm(q=2,mean=3,sd=2)
```

```
## [1] 0.3085375
```

To confirm this result, you can compute the Z-score $\left(Z = \frac{X-\mu}{\sigma}\right)$ of $X = 2$ and look up the cumulative probability in the Z-table.

If you want to find the 95th percentile of the variable $X \sim N(\mu = 3, \sigma = 2)$, you can do so by typing the following:

```
qnorm(p=0.95,mean=3,sd=2)
```

```
## [1] 6.289707
```

Again, you can confirm the result by looking up the Z-score belonging to the 95th

percentile of the standard normal distribution and back-transform to obtain the corresponding value of X.

For $X \sim N(\mu = 82, \sigma = 7)$, what is $P(X < 90)$? (Round to 4 decimal places)

Hint

pnorm(q=?,mean=?,sd=?)

Solution

```
#P(X<90)=P(X 90) for continuous variables
pnorm(q=90,mean=82,sd=7)
```

```
## [1] 0.873451
```

For $X \sim N(\mu = 82, \sigma = 7)$, what is $P(X > 73)$? (Round to 4 decimal places)

Hint 1

What is the complement of $X > 73$?

Hint 2

Apply pnorm() to the complementary event, i.e. compute 1-pnorm(q=?,mean=?,sd=?)

Solution

```
#P(X>73)=1-P(X<73)=1-P(X 73)
1-pnorm(q=73,mean=82,sd=7)
```

```
## [1] 0.9007286
```

```
#or alternatively, you can specify lower.tail=FALSE to compute P(X>73) directly:
pnorm(q=73,mean=82,sd=7,lower.tail=FALSE)
```

```
## [1] 0.9007286
```

For $X \sim N(\mu = 82, \sigma = 7)$, what is $P(73 < X < 90)$? (Round to 4 decimal places)

Hint

pnorm(q=?,mean=?,sd=?) - pnorm(q=?,mean=?,sd=?)

Solution

```
#P(73<X<90)
pnorm(q=90,mean=82,sd=7)-pnorm(q=73,mean=82,sd=7)
```

```
## [1] 0.7741796
```

For $X \sim N(\mu = 12, \sigma = 3)$, what is the 95th percentile? (Round to 2 decimal places)

Hint

qnorm(p=?,mean=?,sd=?)

Solution

```
#95th percentile
qnorm(p=0.95,mean=12,sd=3)
```

```
## [1] 16.93456
```

For $X \sim N(\mu = 12, \sigma = 3)$, what is the 10th percentile? (Round to 2 decimal places)

Hint

qnorm(p=?,mean=?,sd=?)

Solution

```
#10th percentile
qnorm(p=0.1,mean=12,sd=3)
```

```
## [1] 8.155345
```

3.0.2 Discrete distributions

Similar to the normal distribution, you can use `R` to compute cumulative probabilities and percentiles for the binomial and Poisson distributions. Additionally, you can evaluate the probability mass functions of these distributions for particular values that the random variables might take on. The probability mass functions are evaluated with the functions `dbinom()` and `dpois()`. The cumulative probabilities are computed with `pbinom()` and `ppois()`, and the percentiles are computed with `qbinom()` and `qpois()`. You can call the help functions below to get additional info on how to use these functions:

```
?Binomial
?Poisson
```

Note that functions for the geometric and negative binomial distributions exist in the standard `R` package. However, in `R`, these distributions are defined slightly differently from how we covered them in the lecture. Hence, I recommend you

do not use the built-in function and instead use the lines of code provided in the exercises below.

3.0.2.1 Binomial

For $X \sim$ Bin$(n = 30, \theta = 0.1)$, what is $P(X = 4)$? (Round to 4 decimal places)

Hint

dbinom(x=?,size=?,prob=?)

Solution

```
#P(X=4)
dbinom(x=4,size=30,prob=0.1)
```

```
## [1] 0.1770659
```

For $X \sim$ Bin$(n = 10, \theta = 0.3)$, what is $P(X < 5)$? (Round to 4 decimal places)

Hint

pbinom(x=?,size=?,prob=?)

Solution

```
#P(X<5)=P(X 4) for discrete variables
pbinom(q=4,size=10,prob=0.3)
```

```
## [1] 0.8497317
```

For $X \sim$ Bin$(n = 10, \theta = 0.3)$, what is $P(X > 1)$? (Round to 4 decimal places)

Hint 1

What is the complementary event of $X > 1$?

Hint 2

Apply pbinom() to the complementary event, i.e. compute 1-pbinom(q=?,size=?,prob=?)

Solution

```
#P(X>1)=1-P(X 1)
1-pbinom(q=1,size=10,prob=0.3)
```

```
## [1] 0.8506917
```

For $X \sim$ Bin$(n = 10, \theta = 0.3)$, what is $P(1 < X < 5)$? (Round to 4 decimal places)

Hint

pbinom(q=?,size=?,prob=?)-pbinom(q=?,size=?,prob=?)

Solution

```
#P(1<X<5)
pbinom(q=4,size=10,prob=0.3)-pbinom(q=1,size=10,prob=0.3)
```

```
## [1] 0.7004233
```

For $X \sim$ Bin$(n = 40, \theta = 0.2)$, what is the 90th percentile? (As a whole number)

Hint

qbinom(p=?,size=?,prob=?)

Solution

```
qbinom(p=0.9,size=40,prob=0.2)
```

```
## [1] 11
```

3.0.2.2 Geometric

The geometric distribution is defined differently in R than in our course. Hence, I recommend you compute probabilities from the geometric distribution using the formulas in the notes rather than the built-in functions. Luckily, the PMF and CDF of the geometric distribution are quickly computed without any built-in functions.

For $X \sim$ Geom$(\theta = 0.2)$, what is $P(X = 3)$? (Round to 3 decimal places)

Hint

Recall the PMF of a geometric random variable X: $P(X = x) = (1-\theta)^{(x-1)} \times \theta$

Solution

```
0.8^2*0.2
```

```
## [1] 0.128
```

For $X \sim$ Geom$(\theta = 0.02)$, what is $P(X > 4)$? (Round to 4 decimal places)

Hint

Recall that for a geometric random variable X, you can compute $P(X > x) = (1-\theta)^x$.

Solution

```
(1-0.02)^4
```

```
## [1] 0.9223682
```

For $X \sim$ Geom$(\theta = 0.02)$, what is $P(X < 20)$? (Round to 4 decimal places)

Hint

For a geometric random variable X, it is easier to compute the probability of X being greater than some value x: $P(X > x) = (1-\theta)^x$. Now, you can compute $P(X < 20) = 1 - P(X > 19)$.

Solution

```
1-(1-0.02)^19
```

```
## [1] 0.3187674
```

For $X \sim$ Geom$(\theta = 0.02)$, what is $P(4 < X < 20)$? (Round to 4 decimal places)

Hint

$P(4 < X < 20) = P(X < 20) - P(X \leq 4) = (1 - P(X > 19)) - (1 - P(X > 4))$

Solution

```
(1-(1-0.02)^19)-(1-(1-0.02)^4)
```

```
## [1] 0.2411355
```

3.0.2.3 Negative binomial

Recall that for a negative binomial random variable $X \sim$ Neg.Bin.(θ), the PMF can be written as

$$\begin{aligned} P(X = x) &= \binom{x-1}{k-1}(1-\theta)^{(x-k)}\theta^k \\ &= \binom{x-1}{k-1}(1-\theta)^{(x-k)}\theta^{k-1} \times \theta \\ &= P(X_1 = (k-1)) \times P(X_2 = 1), \end{aligned}$$

where $X_1 \sim \text{Bin}(n = (x-1), \theta)$ and $X_2 \sim \text{Bern}(\theta)$. You can use that trick to compute probabilities from the negative binomial distribution in R.

For $X \sim$ Neg.Bin.$(\theta = 0.1)$ and $k = 4$, what is $P(X = 10)$? (Round to 4 decimal places)

Hint

$P(X = 10) = P(X_1 = 3) \times P(X_2 = 1)$, for $X_1 \sim \text{Bin}(n = (x-1), \theta)$ and $X_2 \sim \text{Bern}(\theta)$.

Alternatively, you can use the formula for the negative binomial PMF.

Solution

```
#using the hint:
dbinom(x=3,size=9,prob=0.1)*0.1
```

```
## [1] 0.004464104
```

```
#or:
choose(9,3)*(1-0.1)^6*0.1^4
```

```
## [1] 0.004464104
```

3.0.2.4 Poisson

For $X \sim$ Pois$(\lambda = 3)$, what is what is $P(X = 2)$? (Round to 3 decimal places)

Hint

dpois(x=?,lambda=?)

Solution

```
dpois(x=2,lambda=3)
```

```
## [1] 0.2240418
```

For $X \sim$ Pois$(\lambda = 10)$, what is what is $P(X < 12)$? (Round to 4 decimal places)

Hint

ppois(q=?,lambda=?)

Solution

```
#P(X<12)=P(X 11) for discrete variables
ppois(q=11,lambda=10)
```

```
## [1] 0.6967761
```

For $X \sim$ Pois$(\lambda = 10)$, what is what is $P(X > 7)$? (Round to 4 decimal places)

Hint

What is the complement of $X > 7$?

Solution

```
#P(X>7)=1-P(X 7)
1-ppois(q=7,lambda=10)
```

```
## [1] 0.7797794
```

For $X \sim$ Pois$(\lambda = 10)$, what is what is $P(7 < X < 12)$? (Round to 4 decimal places)

Hint

ppois(q=?,lambda=?) - ppois(q=?,lambda=?)

Solution

```
#P(7<X<12)
ppois(q=11,lambda=10)-ppois(q=7,lambda=10)
```

```
## [1] 0.4765555
```

For $X \sim$ Pois$(\lambda = 7)$, what is what is the 95th percentile? (As a whole number)

Hint

qpois(p=?,lambda=?)

Solution

```
qpois(p=0.95,lambda=7)
```

```
## [1] 12
```

3.0.3 Using the normal distribution to approximate a binomial distribution

In lecture 16, you saw that the binomial distribution can be approximated by a normal distribution. Specifically, if $n \times \theta \geq 10$ and $n \times (1-\theta) \geq 10$, then $X \sim \text{Bin}(n, \theta)$ can be approximated by

$$Y \sim N\left(\mu = n \times \theta, \sigma = \sqrt{n \times \theta \times (1-\theta)}\right)$$

For a binomial distribution with sample size 30 and probability of success equal to 0.2, would you suggest approximating the distribution by a normal distribution?

Hint

Check if the assumptions $n \times \theta \geq 10$ and $n \times (1-\theta) \geq 10$ hold.

- (A) Sure, why not?!
- (B) No, because we can't assume the binomial and normal distributions to be independent.
- (C) No, the expected number of successes is too small, so the normal approximation should not be used.

For $X \sim$ Bin$(n = 400, \theta = 0.1)$, fill in the blanks for the Normal approximation you would use:

Hint

Recall that X~Binomial(n, θ) can be approximated by Y~$N\left(\mu = n \times \theta, \sigma = \sqrt{n \times \theta \times (1-\theta)}\right)$.

$N(\mu =$___$, \sigma =$_$)$.

Solution

$\mu = n \times \theta = 40$, $\sigma = \sqrt{n \times \theta \times (1-\theta)} = \sqrt{36} = 6$

3.0.3.1 Graphical exploration

For the random variable $X \sim \text{Bin}(n = 600, \theta = 0.4)$, you get $n \times \theta = 240 > 10$ and $n \times (1-\theta) = 360 > 10$, so it is appropriate to approximate the distribution using a normal distribution with mean $\mu = n \times \theta = 240$ and standard deviation $\sigma = \sqrt{n \times \theta \times (1-\theta)} = \sqrt{144} = 12$. To get a visual idea for how well the normal distribution approximates the binomial, you can create a plot that overlays the cumulative densities of the two distributions. You will need a sequence of numbers at which to evaluate the respective cumulative density functions (CDFs). This can be done using the `seq()` function, as demonstrated below:

```
seq(from=0,to=10,by=1)
```

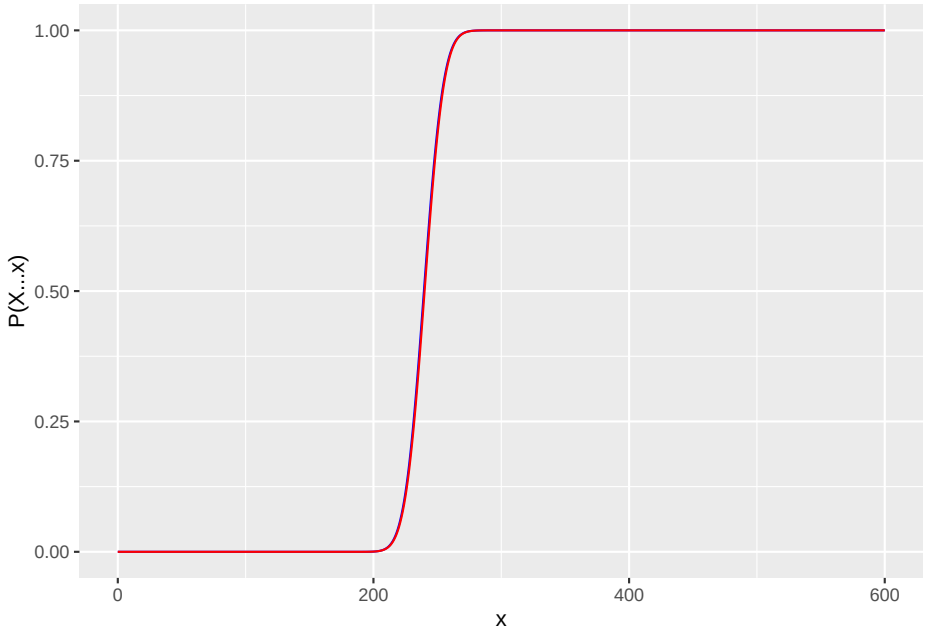
```
##  [1]  0  1  2  3  4  5  6  7  8  9 10
```

You can then compute the CDF at all possible values in the range of the binomial variable, for both the binomial and normal distributions and place them on the same plot:

```
x_vals <- seq(from=0,to=600,by=1)
bin_cdf <- pbinom(q=x_vals,size=600,prob=0.4)
norm_cdf <- pnorm(q=x_vals,mean=240,sd=12)

cdfs <- as.data.frame(cbind(x_vals,bin_cdf,norm_cdf))

ggplot(data=cdfs) + geom_line(aes(x_vals,bin_cdf),colour="blue") + geom_line(aes(x_vals
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>
```



Visually, the CDFs appear to be very close to each other. You can compute their maximum difference to get a better idea how much they differ:

```
#what's the biggest difference between the two CDFs?
max(abs(bin_cdf-norm_cdf))
```

```
## [1] 0.01772144
```

```
#for what value of x do they differ the most?
x_vals[which(abs(bin_cdf-norm_cdf)==max(abs(bin_cdf-norm_cdf)))]
```

```
## [1] 240
```

Hence, the two CDFs are furthest apart for $x = 240$, with an absolute difference of 0.0177.

Now, try to approximate the variable $X \sim \text{Bin}(n = 30, \theta = 0.2)$ with the distribution $Y \sim N\left(\mu = n \times \theta, \sigma = \sqrt{n \times \theta \times (1 - \theta)}\right)$. Compute the CDFs as in the code chunk above, and place the two CDFs on the same plot to answer the question that is to follow.

Hint

```
x_vals <- seq(from=0,to=30,by=1)
bin_cdf <- pbinom(q=x_vals,size=30,prob=0.2)
norm_cdf <- pnorm(q=x_vals,mean=6,sd=2.19089)

cdfs <- as.data.frame(cbind(x_vals,bin_cdf,norm_cdf))
```

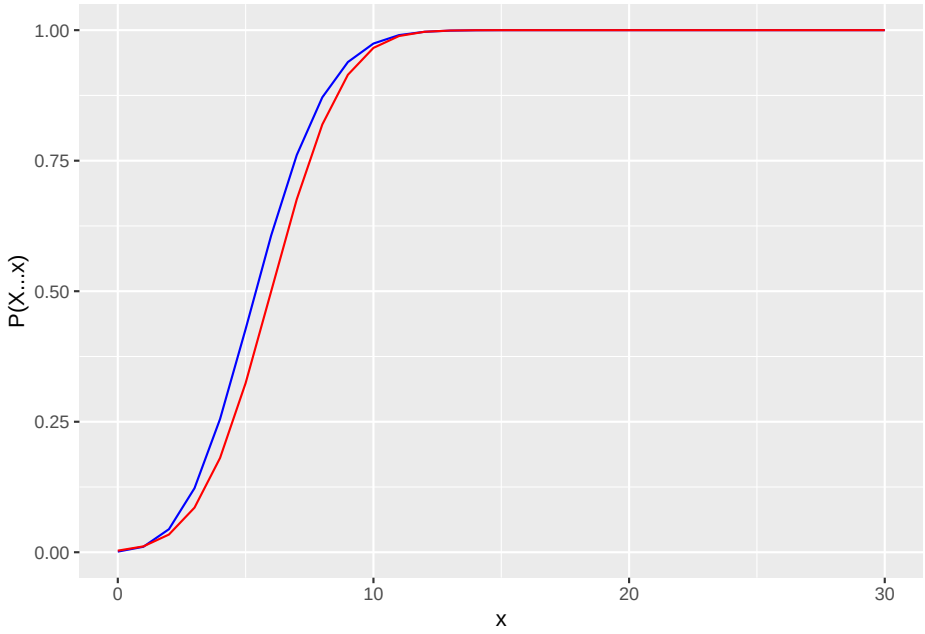
```r
#Now use these to make the appropriate plot.
```

Solution

```r
ggplot(data=cdfs) + geom_line(aes(x_vals,bin_cdf),colour="blue") + geom_line(aes(x_vals,norm_cdf)
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>
```



Which of the following best describes the normal approximation for the binomial example under consideration?

Hint

Is one of the coloured lines consistently higher than the other? If so, which one?

- (A) The binomial CDF returns higher probabilities than the normal CDF for most values of x.

- (B) The binomial CDF returns lower probabilities than the normal CDF for most values of x.

- (C) While the approximation is not very accurate, sometimes the binomial CDF returns higher probabilities than the normal CDF, and sometimes it is the other way around.

Let's see how the probability of success θ influences the dynamic of the normal approximation. Try to approximate the variable $X \sim \text{Bin}(n = 30, \theta = 0.8)$ by $Y \sim N\left(\mu = n \times \theta, \sigma = \sqrt{n \times \theta \times (1 - \theta)}\right)$ and answer the question that is to follow.

Hint

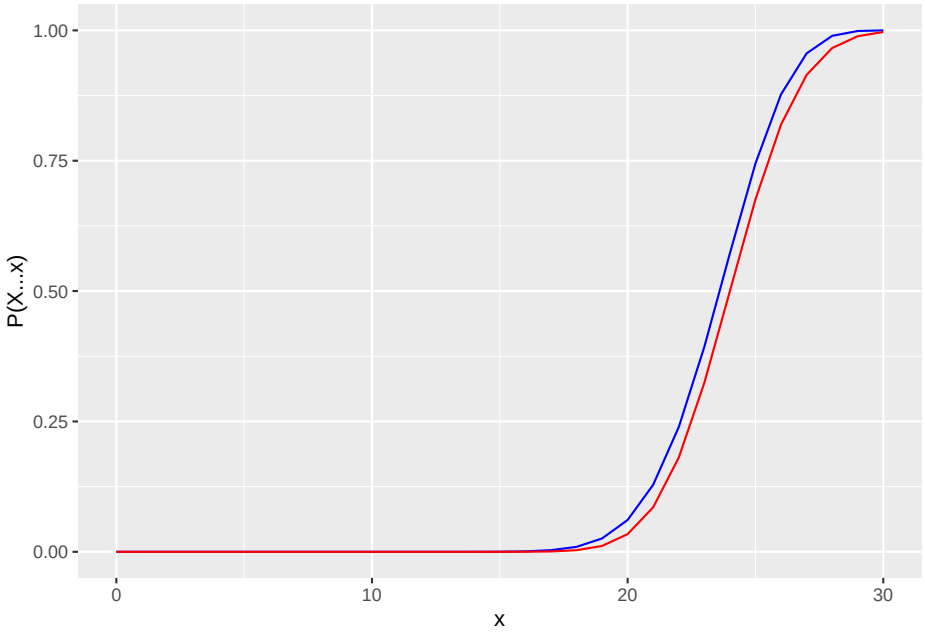
```
x_vals <- seq(from=0,to=30,by=1)
bin_cdf <- pbinom(q=x_vals,size=30,prob=0.8)
norm_cdf <- pnorm(q=x_vals,mean=24,sd=2.19089)

cdfs <- as.data.frame(cbind(x_vals,bin_cdf,norm_cdf))

#Now use these to make the appropriate plot.
```

Solution

```
ggplot(data=cdfs) + geom_line(aes(x_vals,bin_cdf),colour="blue") + geom_line(aes(x_vals,norm_cdf)
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'P(X x)' in 'mbcsToSbcs': dot substituted for <a4>
```



Now that we have increased the probability of success to $\theta = 0.8$, which of the following best describes the Normal approximation for the Binomial example under consideration?

Hint

Is one of the coloured lines consistently higher than the other? If so, which one?

- (A) The binomial CDF still returns higher probabilities than the normal CDF for most values of x.
- (B) The binomial CDF now returns lower probabilities than the normal CDF for most values of x.
- (C) While the approximation is still not very accurate, now the binomial CDF sometimes returns higher probabilities than the normal CDF, and sometimes it is the other way around.

Sometimes, when computations become increasingly expensive, it can be quite useful to approximate the binomial distribution by a normal distribution. Hopefully, this part of the lab helped you understand why the normal approximation is only appropriate when the number of observations n is sufficiently large for a given probability of success θ. Keeping this in mind, you should always check the assumptions $n \times \theta \geq 10$ and $n \times (1 - \theta) \geq 10$ before applying the normal approximation and compute the binomial probabilities directly if the assumptions do not hold.

Chapter 4

Group Exercises

In this group exercise, you will generate random samples from normal distributions. You will then pretend the samples had been handed to you, and explore the samples visually to examine whether the normality assumption seems appropriate for the sample at hand. You might find this exercise helpful in the preparation for later parts of the course, where you will be given a sample or dataset and asked to check if the normality assumption is appropriate for some variable, before performing tests that assume an underlying normal distribution.

In `R`, you can generate random values from several probability distributions. For example, you can use the functions `rbinom()` and `rpois()` to obtain random values from the binomial and Poisson distributions, respectively. Similarly, you can obtain random values from a normal distribution using the function `rnorm()`.

Note: the function `rnorm()` returns random values from the specified probability distribution, so your results of the following simulation study will vary every time you run the code. The questions for this exercise are based on a seed of 1 (setting a seed allows for reproducibility of results obtained from random functions), which is done via the command `set.seed(1)`, which you should run before each random function in this exercise to ensure you can get to the correct answers.

For example, in the code below you will see that the results differ if you do not set a seed but that repeated trials lead to the same result if you use the same seed:

```
set.seed(1)
rnorm(n=5,mean=3,sd=1)
```

```
## [1] 2.373546 3.183643 2.164371 4.595281 3.329508
```

```
#here, no seed was specified, so the result is different:
rnorm(n=5,mean=3,sd=1)
```

```
## [1] 2.179532 3.487429 3.738325 3.575781 2.694612
```

```
#using the same seed as before leads to the same result
set.seed(1)
rnorm(n=5,mean=3,sd=1)
```

```
## [1] 2.373546 3.183643 2.164371 4.595281 3.329508
```

4.0.1 Checking the normality assumption

1. Generate 30 values from a normal distribution with mean $\mu = 10$ and variance $\sigma^2 = 4$. Then, create a boxplot for your sample. Remember to set a seed of 1 before running the function.

2. Imagine you were handed the sample without being told that it was generated from a normal distribution. Based on the boxplot, would you say that the sample approximately follows a normal distribution?

- (A) The boxplot suggests it is unlikely that the majority of observations fall within one standard deviation of the mean. Hence, the normality assumption seems to be violated.

- (B) The boxplot shows that the median is closer to the third quartile than to the first. However, the boxplot looks roughly symmetric and hence, the normality assumption seems appropriate.

- (C) The minimum is further away from the median than the maximum. Hence, the sample does not seem to be symmetric and thus, it appears that the normality assumption is violated.

3. Next, create a histogram to get a better idea of the shape of the distribution. You might have to change the binwidth to get a better plot.

4. Based on the histogram, would you say that the sample approximately follows a Normal distribution?

- (A) The lower tail of the distribution is slightly heavier than the upper tail. However, the data closer to the mode of the distribution seem roughly symmetric and hence, the normality assumption isn't violated.

- (B) It doesn't look like the majority of data fall within one standard deviation of the mean. Hence, the normality assumption seems to be violated.

- (C) The sample contains many very small values which make it unlikely that the data follow a Normal distribution.

4.0.2 Heights in the Scottish population

In Scotland, the average height for females is 161.3cm while that of males is 175cm. The standard deviation is approximately 6cm for females and 7cm for males.

5. Based on this information, generate a random sample of the heights of 20 Scottish females and 20 Scottish males (remember to set a seed of 1 before you evaluate each function). Then, create a boxplot for the heights of the 40 people in the sample.

6. Imagine you were handed the sample without being told that it was generated from a Normal distribution. Based on the boxplot, would you say that the sample approximately follows a normal distribution?

- (A) It doesn't look like the majority of data fall within one standard deviation of the mean. Hence, the normality assumption seems to be violated.

- (B) The boxplot does not look symmetric, as the third quartile is further away from the median than the fist. Hence, the normality assumption seems violated.

- (C) The boxplot looks roughly symmetric and hence, the normality assumption seems appropriate.

7. As before, create a histogram to get a better idea of the shape of the distribution. You might have to change the binwidth to get a better plot.

8. Based on the histogram, would you say that the sample approximately follows a normal distribution?

- (A) The normal distriution is unimodal and symmetric. The sample is not symmetric around a single mode and hence, the normality assumption seems violated.

- (B) There are too many observations far below the median. Hence, the distribution does not look symmetric and the normality assumption seems violated.

- (C) The sample is roughly symmetric. Hence, the normality assumption seems appropriate.

Hopefully, this exercise gave you a first idea on how to check the appropriateness of assuming normality of a sample. Also, you saw that it might be helpful considering different plots when evaluating whether the normality assumption is appropriate.