In [1]:

```
df <- read.table('videodata.txt',head=TRUE)
head(df,10)
```

A data.frame: 10 × 15

	time	like	where	freq	busy	educ	sex	age	home	math	work	own	cdrom
	<dbl>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	2.0	3	3	2	0	1	0	19	1	0	10	1	0
2	0.0	3	3	3	0	0	0	18	1	1	0	1	1
3	0.0	3	1	3	0	0	1	19	1	0	0	1	0
4	0.5	3	3	3	0	1	0	19	1	0	0	1	0
5	0.0	3	3	4	0	1	0	19	1	1	0	0	0
6	0.0	3	2	4	0	0	1	19	0	0	12	0	0
7	0.0	4	3	4	0	0	1	20	1	1	10	1	0
8	0.0	3	3	4	0	0	0	19	1	0	13	0	0
9	2.0	3	2	1	1	1	1	19	0	0	0	0	0
10	0.0	3	3	4	0	1	1	19	1	1	0	1	0

Scenario1

In [2]:

```
sum(df$time != 0 )
```

34

In [3]:

```
nrow(df)
```

91

In [4]:

```
proportion <- mean(df$time != 0) #proportion of people who play games
proportion
```

0.373626373626374

In [5]:

```
mean(df$time)
```

1.24285714285714

Scenario 2

In [6]:

```
unique(df$freq)
```

2 · 3 · 4 · 1 · 99

In [7]:

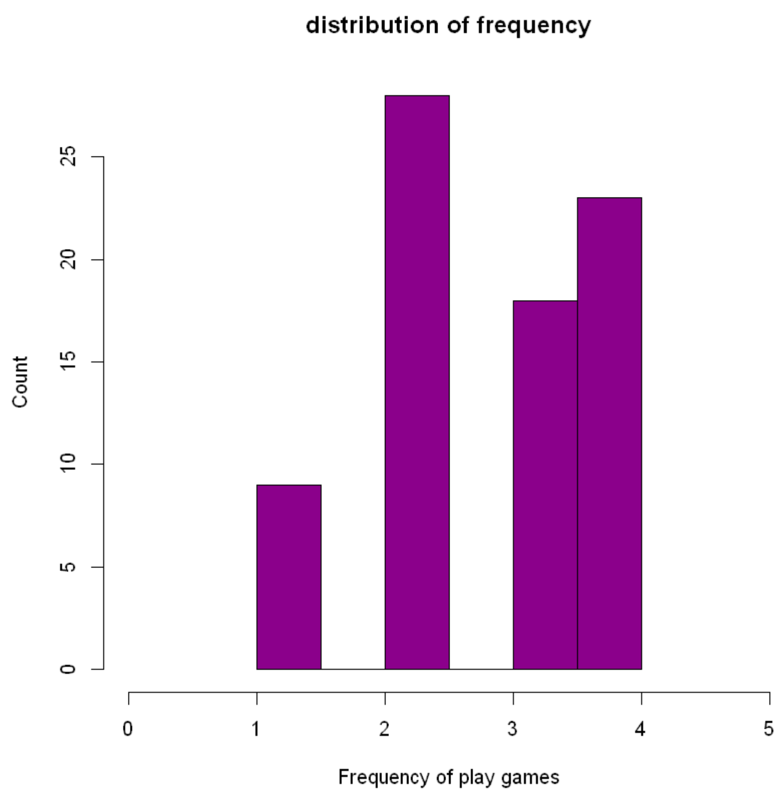
```
mean(subset(df,df$freq!=99)$freq) #average amount of frequency of play, exclude invalid
freq such as99
```

2.70512820512821

Most people play between monthly and weekly.

In [8]:

```
hist(subset(df,df$freq!=99)$freq,col = 'darkmagenta',right=F,xlim=c(0,5),xlab='Frequenc
y of play games'
     ,main='distribution of frequency',ylab='Count')
```



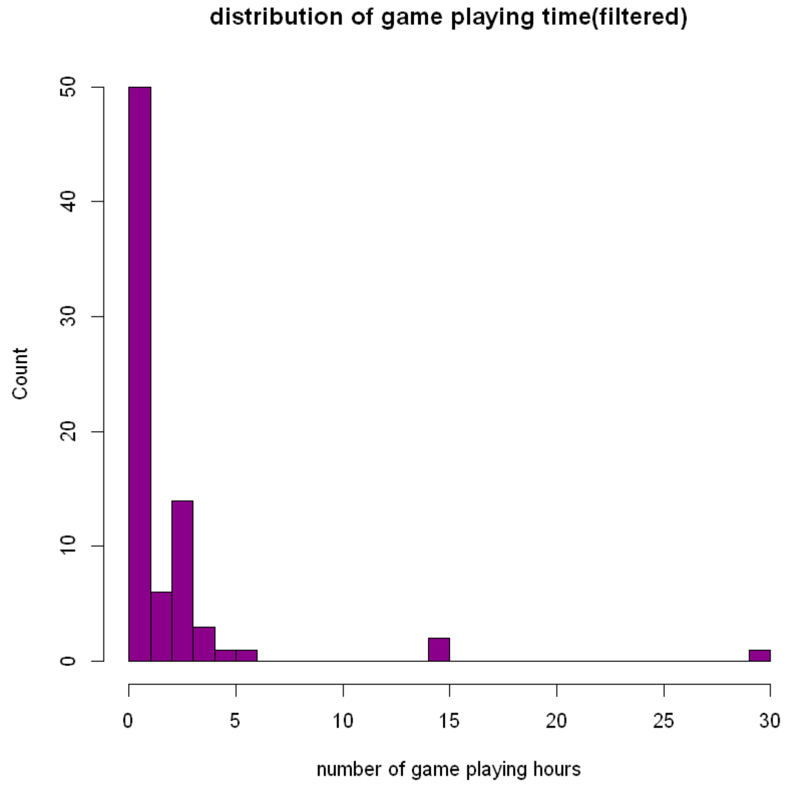
In [9]:

```
#average amount of time, exclude those who does not play game at all in coherence with
 frequency
mean(subset(df,df$time!=0)$time)
```

3.32647058823529

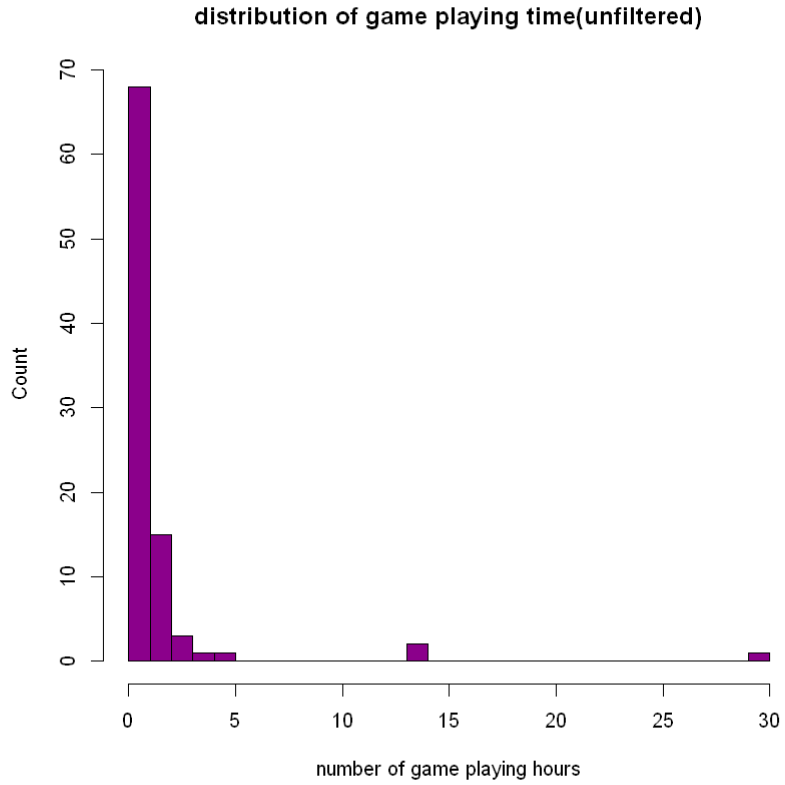
In [10]:

```
hist(subset(df,df$freq!=99)$time,col = 'darkmagenta',right=F,breaks=30,xlab='number of
 game playing hours',ylab='Count',
    main='distribution of game playing time(filtered)')
```



In [11]:

```
hist(df$time,col = 'darkmagenta',breaks=30,xlab='number of game playing hours',ylab='Co
unt',
    main='distribution of game playing time(unfiltered)')
```



Scenario3

In [12]:

```
sd <- sd(df$time)
sd
```

3.77704007736637

In [13]:

```
upper_bound <- mean(df$time)+2*sd/sqrt(nrow(df))
lower_bound <- mean(df$time)-2*sd/sqrt(nrow(df))
```

In [14]:

```
c(lower_bound,upper_bound) # construct a 95% interval of estimation
```

0.450974374698314 · 2.03473991101597

In [15]:

```
mean(df$time<lower_bound)
```

0.637362637362637

In [16]:

```
mean(df$time>upper_bound) # only 8% of the sampe is above mean in this estimation
```

0.0879120879120879

In [17]:

```
mean(df$time==0) # reason of right-skewness, a lot of 0s
```

0.626373626373626

Such an interval estimation is not so approperiate. On the right side(lower bound), about 62% people did not play the game at all. From the graph above we can observe that the sample is not normally distributed but highly right skewed. In this case, we use bootstrap to help to make an interval estimation of mean.

We proform bootstrap based on the instruction from lecture sildes:

"According to the simple random sample probability model,the distribution of the sample should look roughly similar to that of the population. We could create a new population of 314 based on the sample and use this population, which we call the the bootstrap population,to find the probability distribution of the sample average. For every unit in the sample,we make 314/91 = 3.45 units in the bootstrap population with the same time value and round off to the nearest integer."

In [18]:

```
bootobject= NULL
N = 400
for (i in 1:N) {
    bootobject[i]=mean(sample(as.vector(df$time),size=91,replace=TRUE))
}
```

In [19]:

```
boot_sd = sd(bootobject)
n=91
boot_mean = mean(bootobject)
```

In [20]:

```
boot_upper_bound <- boot_mean+2*boot_sd/sqrt(N)
boot_lower_bound <- boot_mean-2*boot_sd/sqrt(N)
```

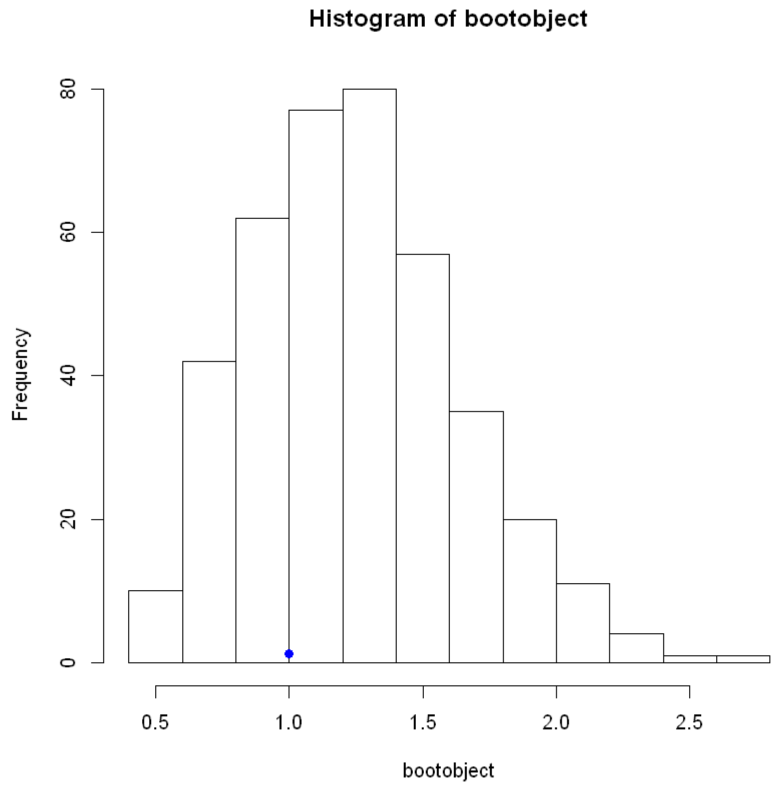
In [21]:

```
c(boot_lower_bound,boot_upper_bound) # construct a 95% confidence interval
```

1.20927120291282 · 1.28787714873553

In [22]:

```
hist(bootobject)
points(mean(df$time),col='red',pch=16)
points(mean(boot_mean),col='blue',pch=16)
```



Scenario4

In [23]:

```
video_multiple <- read.table('videoMultiple.txt',head=T)
```

In [24]:

```
colnames(video_multiple)
```

'action' · 'adv' · 'sim' · 'sport' · 'strategy' · 'relax' · 'coord' · 'challenge' · 'master' · 'bored' · 'other' · 'graphic' · 'time' · 'frust' · 'lonely' · 'rules' · 'cost' · 'boring' · 'friends' · 'point' · 'other2'

In [25]:

```
like = seq(1,12)
like = like[like!=11] #remove string column
dislike = seq(13,ncol(video_multiple)-1) # -1 to remove string column
```

In [26]:

```
cor(na.omit(video_multiple[like]))
```

A matrix: 11 × 11 of type dbl

	action	adv	sim	sport	strategy	relax	
action	1.00000000	0.30848722	0.31917000	0.25521877	0.07401957	0.34156503	0.1022
adv	0.30848722	1.00000000	0.31534245	0.11608029	0.27366311	0.23348689	-0.1394
sim	0.31917000	0.31534245	1.00000000	0.13332871	0.03263956	0.12909944	0.1903
sport	0.25521877	0.11608029	0.13332871	1.00000000	-0.02414542	0.21654640	-0.0633
strategy	0.07401957	0.27366311	0.03263956	-0.02414542	1.00000000	0.32024493	-0.1739
relax	0.34156503	0.23348689	0.12909944	0.21654640	0.32024493	1.00000000	0.1552
coord	0.10225512	-0.13940075	0.19038114	-0.06335044	-0.17399079	0.15523011	1.0000
challenge	0.06116777	0.17602414	0.09808165	0.04366064	0.37455700	0.17094086	0.0044
master	0.25765694	0.04580645	0.04637389	0.16813692	0.16831492	0.07184212	-0.0181
bored	0.03016917	0.17636891	0.12677314	0.08542123	0.25746433	0.05455447	-0.1354
graphic	0.31835356	0.19530001	0.07138003	0.26770420	-0.02920032	0.31331416	0.2417

In [27]:

```
cor(na.omit(video_multiple[dislike]))
```

A matrix: 8 × 8 of type dbl

	time	frust	lonely	rules	cost	boring	
time	1.000000000	-0.005395823	-0.10225512	-0.3020604	0.004852616	-0.11008676	-0.14
frust	-0.005395823	1.000000000	0.11729808	0.1647135	0.092865412	-0.12066529	-0.09
lonely	-0.102255123	0.117298081	1.00000000	-0.1081848	-0.068182219	-0.09613766	-0.03
rules	-0.302060418	0.164713467	-0.10818484	1.0000000	-0.167837027	0.02085590	0.11
cost	0.004852616	0.092865412	-0.06818222	-0.1678370	1.000000000	-0.29549076	0.03
boring	-0.110086758	-0.120665290	-0.09613766	0.0208559	-0.295490759	1.00000000	-0.06
friends	-0.148191715	-0.091955872	-0.03367414	0.1178360	0.030562492	-0.06717507	1.00
point	-0.097590007	-0.258023423	-0.03880753	-0.1024900	-0.182323225	0.48661135	0.05

Scenario 5

In [28]:

```
df <- subset(df,df$like!=99) # drop invalid responses
```

MALE PROPORTION VS FEMALE PROPORTION

In [29]:

```
male <- subset(df,df$sex == 1)
male_prop <- nrow(male[male$like %in% c(2,3),])/nrow(male) #48/53
female <- subset(df,df$sex == 0)
female_prop <- nrow(female[female$like %in% c(2,3),])/nrow(female) #34/38
```

Let pm (pw) be the proportion of male (female) gamers We test H0: pm = pw, H1: pm > pw Since we do not have actual vales for pm and pw, we use observed S/F There are not enough in each group, but we'll conduct two sample z-test anyway.

Assuming H0, we use a pooled estimate: p* = (48+34)/(53+38) = 82/91

In [30]:

```
z_stat <- (male_prop - female_prop) / sqrt((82/91*9/91/53) + (82/91*9/91/38))
z_stat
pnorm(z_stat)
```

2.2489739055679

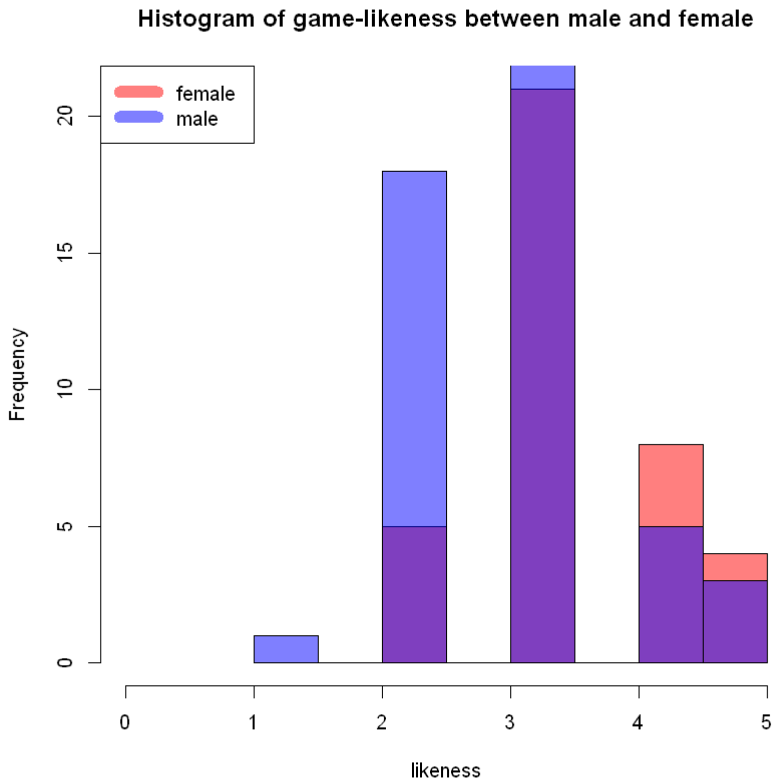
0.987742921846771

With a p-value of 0.013, we reject H0. It appears the male gamers percentage significantly higher than that of woman gamers.

Here is a diagram of distribution

In [31]:

```
hist(female$like,col=rgb(1,0,0,0.5),xlim=c(0,5),right=F,main='Histogram of game-likenes
s between male and female'
     ,xlab='likeness')
hist(male$like,add=T,col=rgb(0,0,1,0.5),right=F)
legend('topleft',c('female','male'),col=c(rgb(1,0,0,0.5),rgb(0,0,1,0.5)),lwd=10)
```



Here is the general formula for z-test.

In [43]:

```
z_test <- function(p1, p1_tot, p2, p2_tot) {
    pooled <- (p1+p2) / (p1_tot + p2_tot)
    unpooled <- ((p1_tot + p2_tot) - (p1+p2)) / (p1_tot + p2_tot)
    z <- ((p1/p1_tot) - (p2/p2_tot)) / sqrt((pooled*unpooled/p1_tot) + (pooled*unpooled
/p2_tot))
    return (z)
}
```

COMPUTER AT HOME PROPORTION VS NON

In [55]:

```
home <- subset(df,df$home == 1)
home_prop <- nrow(home[home$like %in% c(2,3),])
non_home <- subset(df,df$home == 0)
non_home_prop <- nrow(non_home[non_home$like %in% c(2,3),])
```

In [56]:

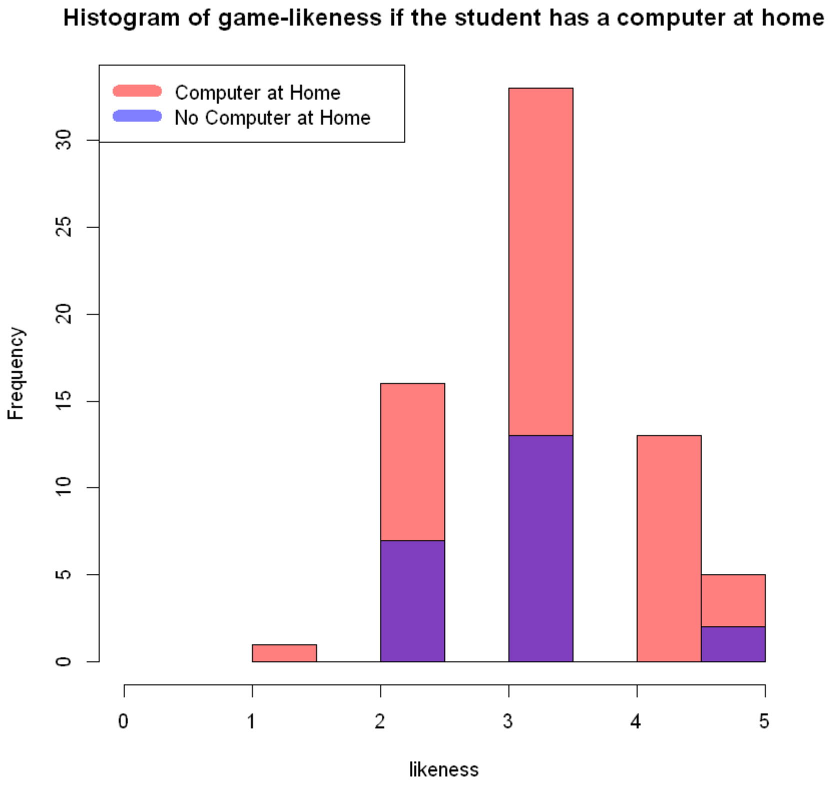
```
pnorm(z_test(home_prop, nrow(home), non_home_prop, nrow(non_home)))
```

0.0346036092821501

Diagram of distribution

In [57]:

```
hist(home$like,col=rgb(1,0,0,0.5),xlim=c(0,5),right=F,main='Histogram of game-likeness
 if the student has a computer at home'
     ,xlab='likeness')
hist(non_home$like,add=T,col=rgb(0,0,1,0.5),right=F)
legend('topleft',c('Computer at Home','No Computer at Home'),col=c(rgb(1,0,0,0.5),rgb(0
,0,1,0.5)),lwd=10)
```



OWN A PC PROPORTION VS NON

In [58]:

```
have_own <- subset(df,df$own == 1)
own_prop <- nrow(have_own[have_own$like %in% c(2,3),])
non_own <- subset(df,df$own == 0)
non_own_prop <- nrow(non_own[non_own$like %in% c(2,3),])
```

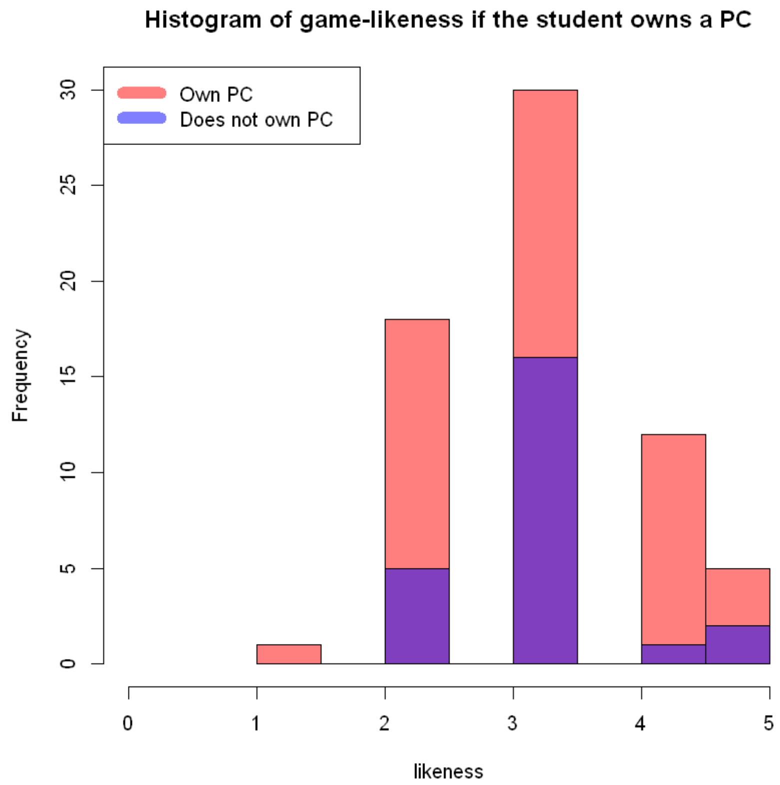
In [59]:

```
pnorm(z_test(own_prop, nrow(have_own), non_own_prop, nrow(non_own)))
```

0.0714201040072203

In [60]:

```
hist(have_own$like,col=rgb(1,0,0,0.5),xlim=c(0,5),right=F,main='Histogram of game-likeness if the student owns a PC'
     ,xlab='likeness')
hist(non_own$like,add=T,col=rgb(0,0,1,0.5),right=F)
legend('topleft',c('Own PC','Does not own PC'),col=c(rgb(1,0,0,0.5),rgb(0,0,1,0.5)),lwd=10)
```



With a p-value of 0.07 we keep H0.

WORK PROPORTION VS NON

In [61]:

```
have_work <- subset(df,df$work != 0)
work_prop <- nrow(have_work[have_work$like %in% c(2,3),])
non_work <- subset(df,df$work == 0)
non_work_prop <- nrow(non_work[non_work$like %in% c(2,3),])
```

In [62]:

```
z_test(work_prop, nrow(have_work), non_work_prop, nrow(non_work))
```

1.86132017742437

In [63]:

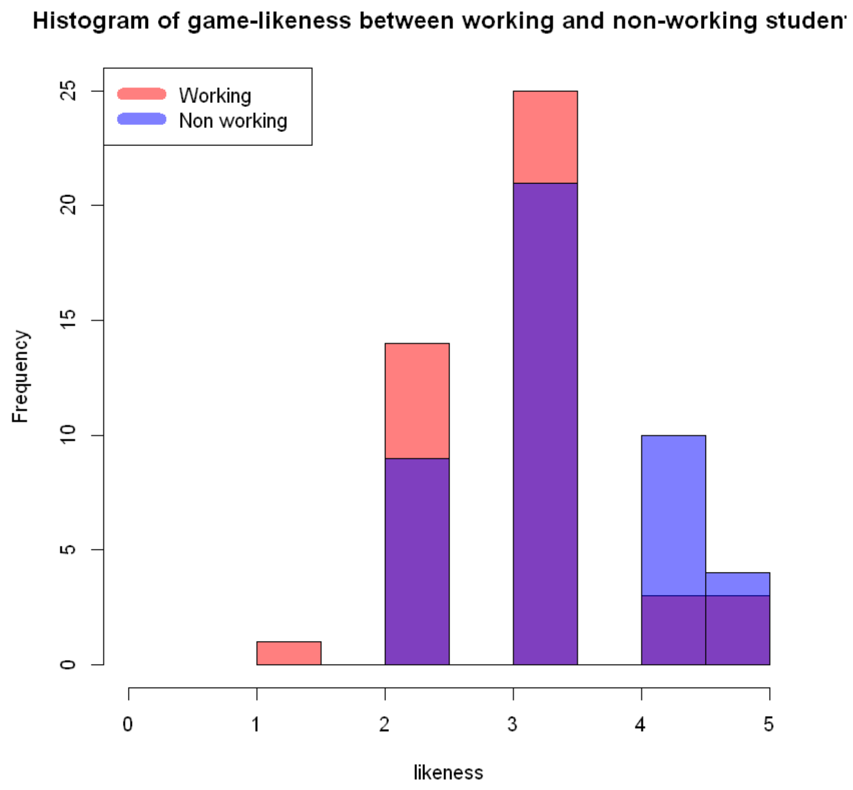
```
pnorm(z_test(work_prop, nrow(have_work), non_work_prop, nrow(non_work)))
```

0.968650512295581

With a p-value of 0.031 we reject H0.

In [64]:

```
hist(have_work$like,col=rgb(1,0,0,0.5),xlim=c(0,5),right=F,main='Histogram of game-like
ness between working and non-working students'
     ,xlab='likeness')
hist(non_work$like,add=T,col=rgb(0,0,1,0.5),right=F)
legend('topleft',c('Working','Non working'),col=c(rgb(1,0,0,0.5),rgb(0,0,1,0.5)),lwd=10
)
```



In [ ]: