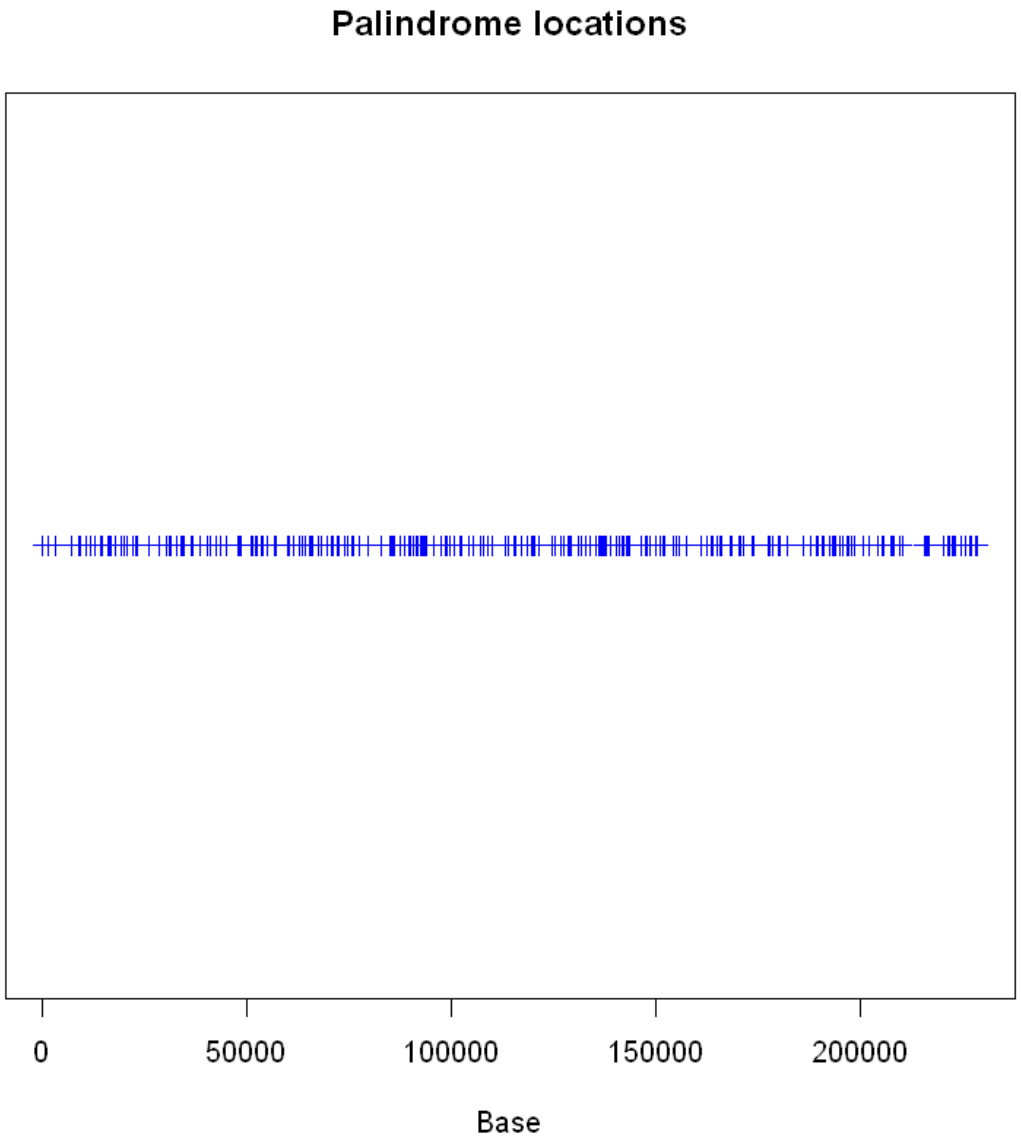
In [1]:
```
dna <- read.table("hcmv.txt", header = T)
head(dna)
```

location
177
1321
1433
1477
3248
3255

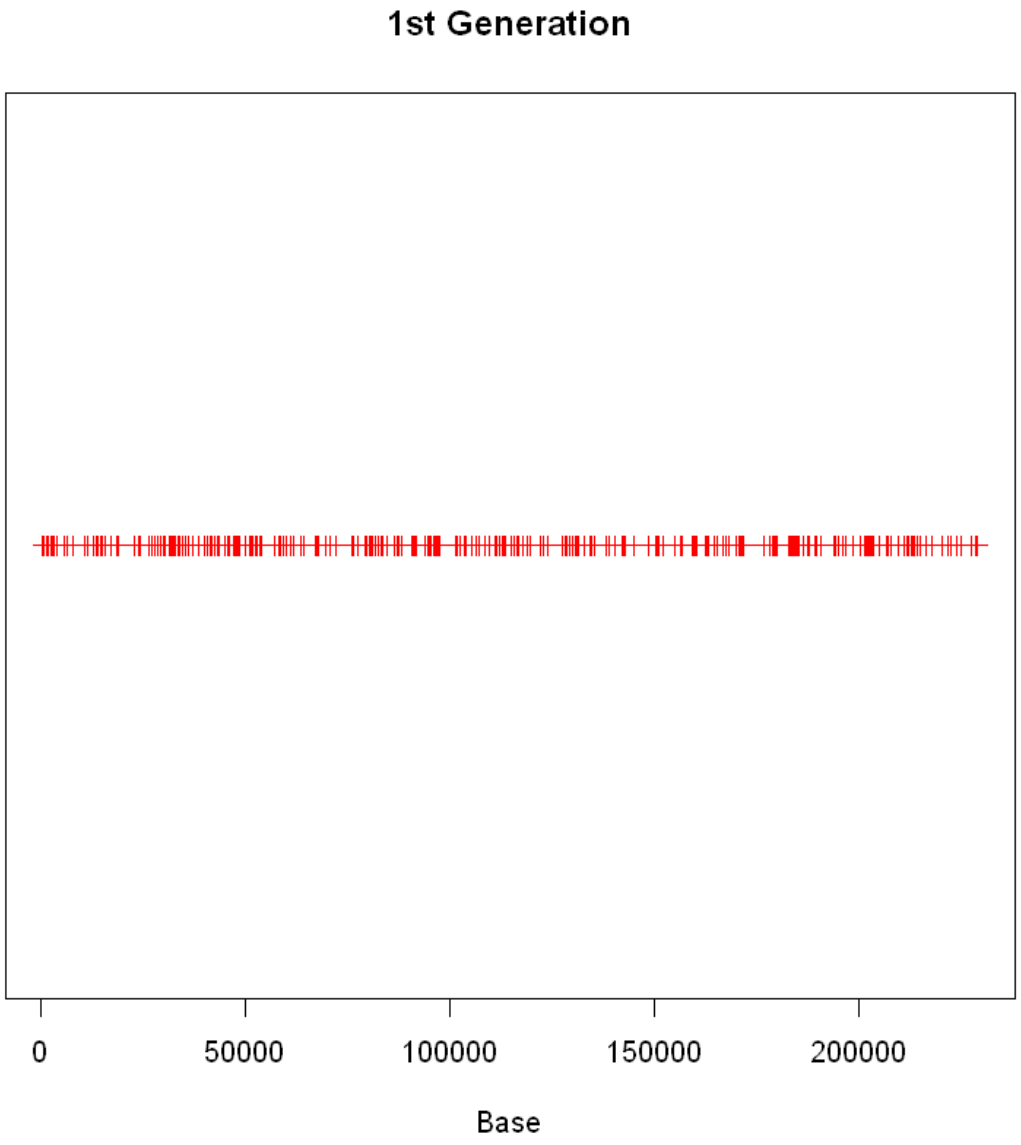
Random Scatter

In [2]:
```
Title = 'Palindrome locations'
library(ggplot2)

stripchart(dna$location,
           main = Title,
           xlab = 'Base',
           col = 'blue',
           pch = 3)
```

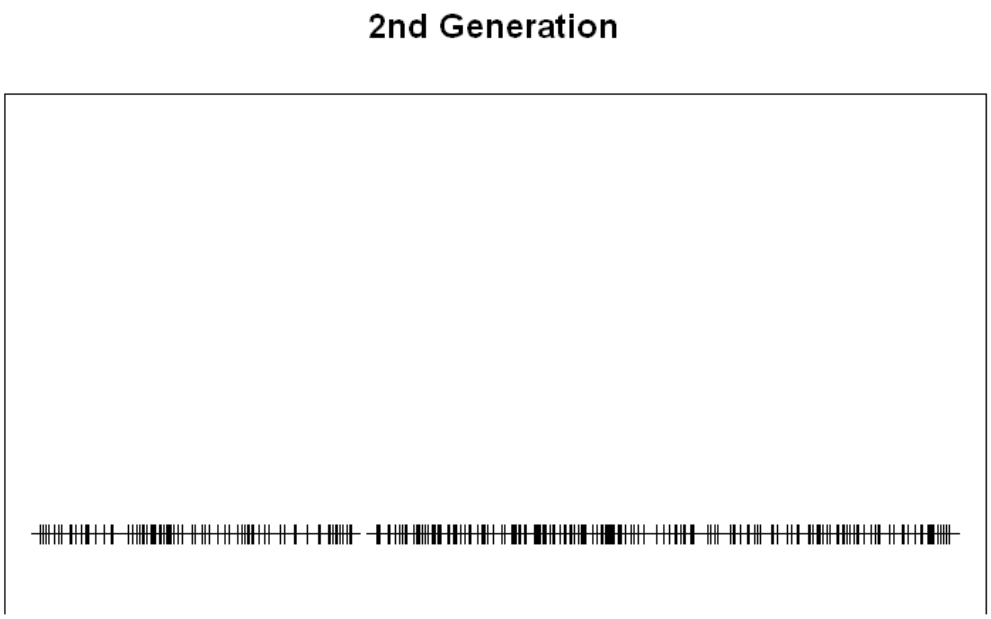
```
Warning message:
"package 'ggplot2' was built under R version 3.6.2"
```



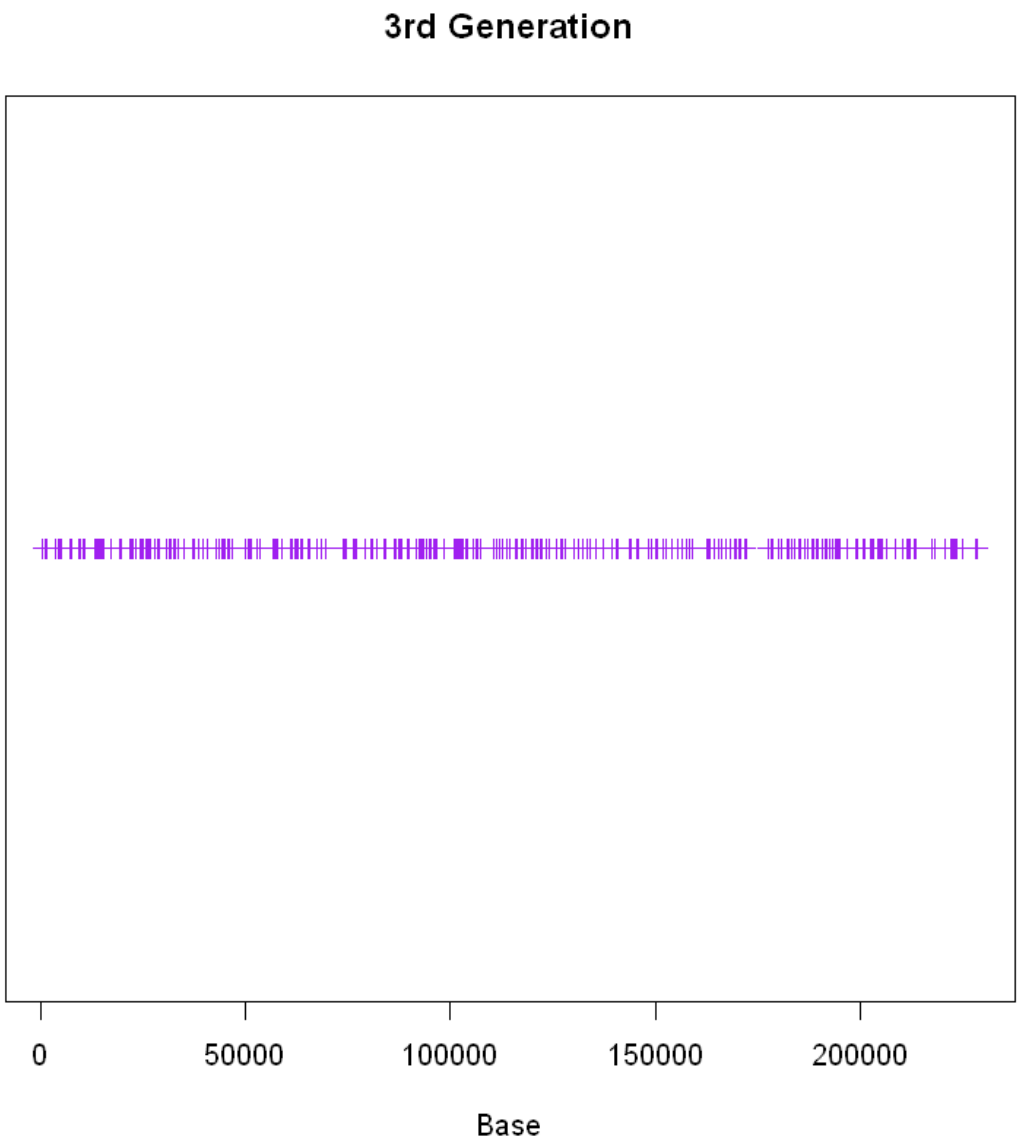
In [3]:
```
#1st Generation
one <- round(runif(296, 0, 229354))
stripchart(one,
           main = '1st Generation',
           xlab = 'Base',
           col = 'red',
           pch = 3)
```



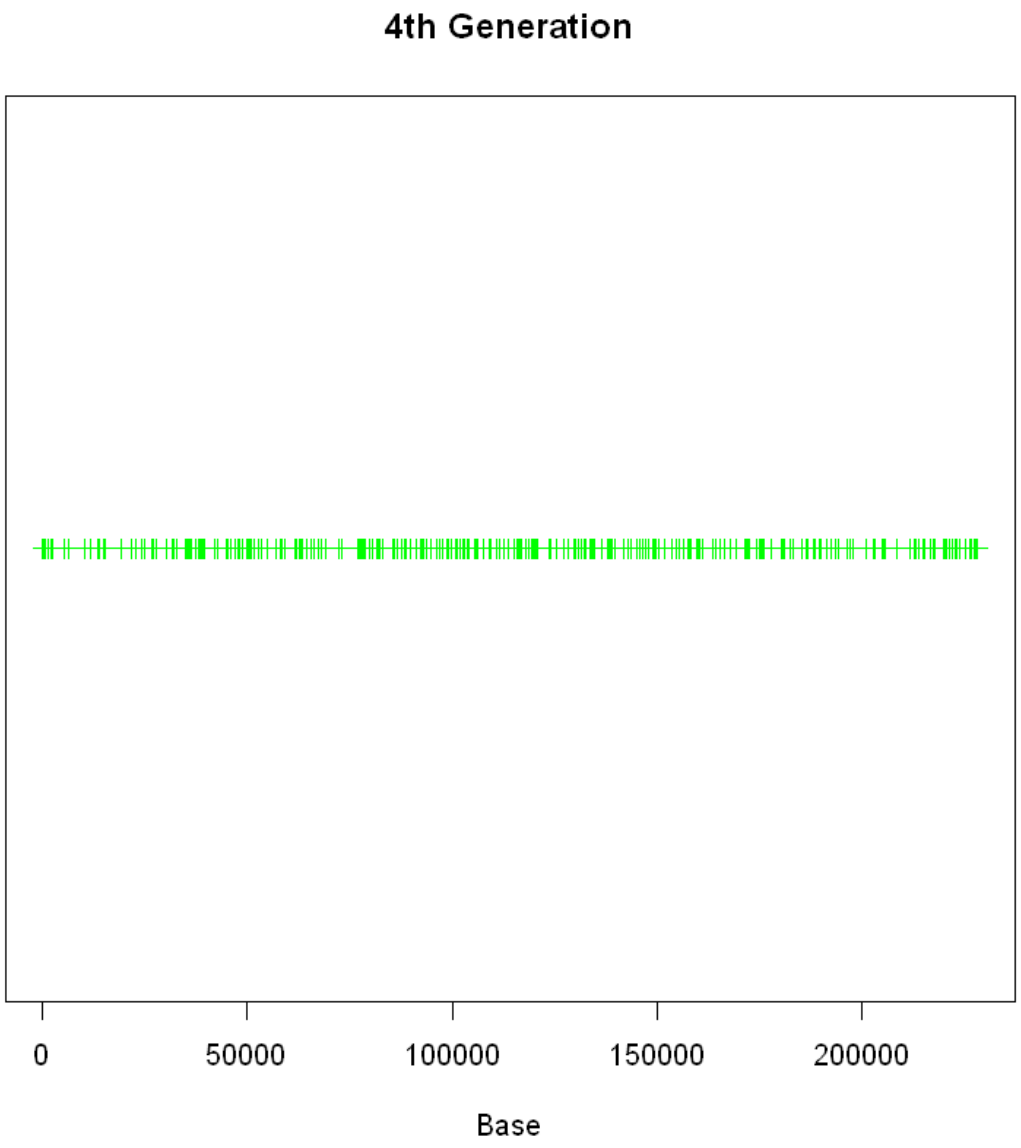
In [4]:
```
#2nd Generation
two <- round(runif(296, 0, 229354))
stripchart(two,
           main = '2nd Generation',
           xlab = 'Base',
           col = 'black',
           pch = 3)
```



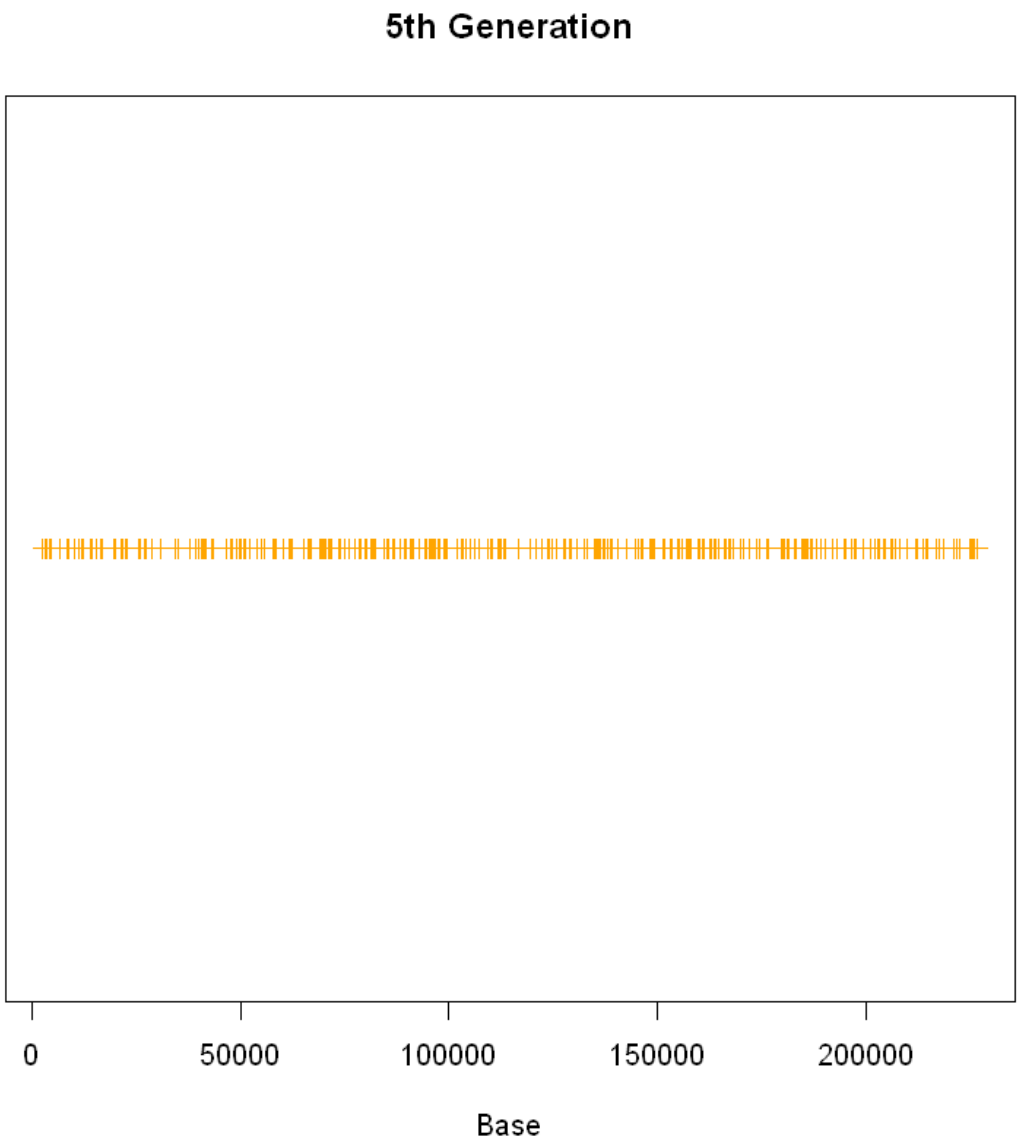
In [5]:
```
#3rd Generation
three <- round(runif(296, 0, 229354))
stripchart(three,
           main = '3rd Generation',
           xlab = 'Base',
           col = 'purple',
           pch = 3)
```



In [6]:
```
#4th Generation
four <- round(runif(296, 0, 229354))
stripchart(four,
           main = '4th Generation',
           xlab = 'Base',
           col = 'green',
           pch = 3)
```



In [7]:
```
#5th Generation
five <- round(runif(296, 0, 229354))
stripchart(five,
           main = '5th Generation',
           xlab = 'Base',
           col = 'orange',
           pch = 3)
```



Location and Spacing

I used sample 2 (two) and dna$location to compare. Feel free to change the variable and use other samples.

In [8]:
```
#Spacing between consecutive palindromes: data 1

all <- dna$location

spaces <- list()
for (i in seq(length(all) - 1)) {
    spaces[[i]] <- (all[[i + 1]] - all[[i]])
}
```

In [9]:
```
#Sum of consecutive pairs: data 1

pairs <- list()
for (i in seq(length(all) - 1)) {
    pairs[[i]] <- (all[[i]] + all[[i + 1]])
}
```

In [10]:
```
#Sum of consecutive triplets: data 1

trip <- list()
for (i in seq(length(all) - 2)) {
    trip[[i]] <- (all[[i]] + all[[i + 1]] + all[[i + 2]])
}
```

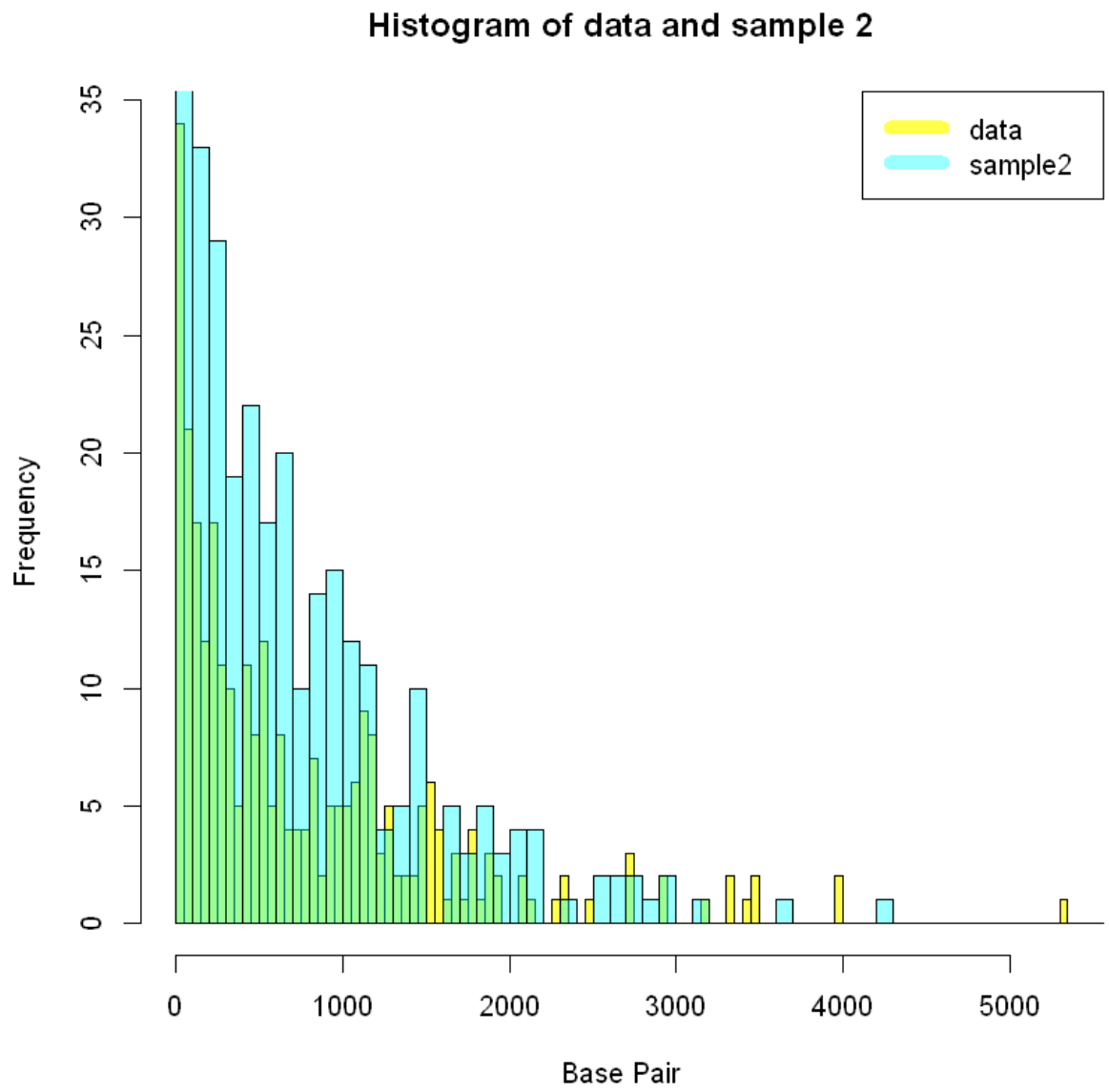
In [ ]:
```

```

In [11]:
```
#Spacing between consecutive palindromes: sample 2
two <- sort(two)

spaces2 <- list()
for (i in seq(length(two) - 1)) {
    spaces2[[i]] <- (two[[i + 1]] - two[[i]])
}
```

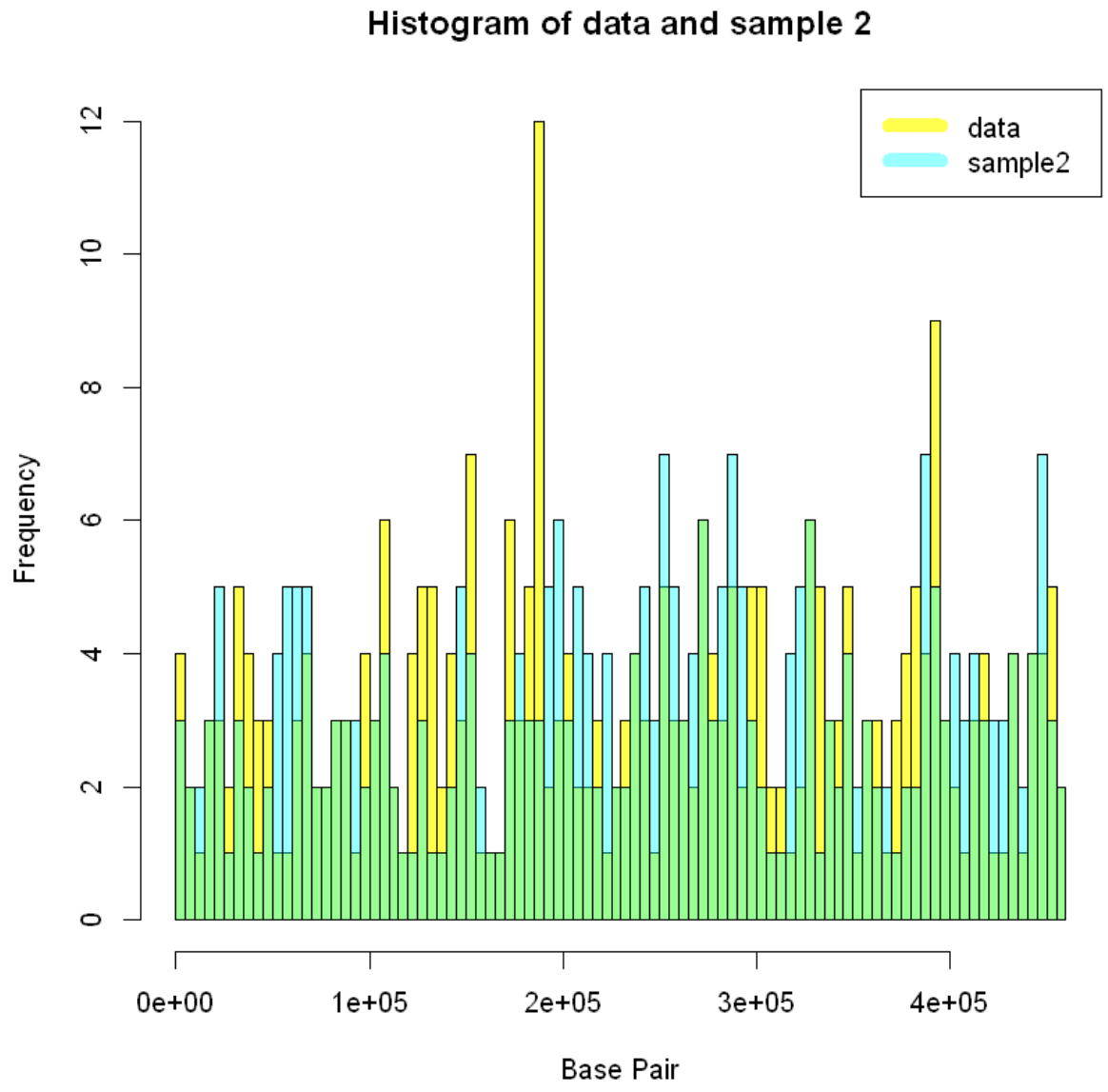
In [12]:
```
#Sum of consecutive pairs: sample 2

pairs2 <- list()
for (i in seq(length(two) - 1)) {
    pairs2[[i]] <- (two[[i]] + two[[i + 1]])
}
```

In [13]:
```
#Sum of consecutive triplets: sample 2

trip2 <- list()
for (i in seq(length(two) - 2)) {
    trip2[[i]] <- (two[[i]] + two[[i + 1]] + two[[i + 2]])
}
```

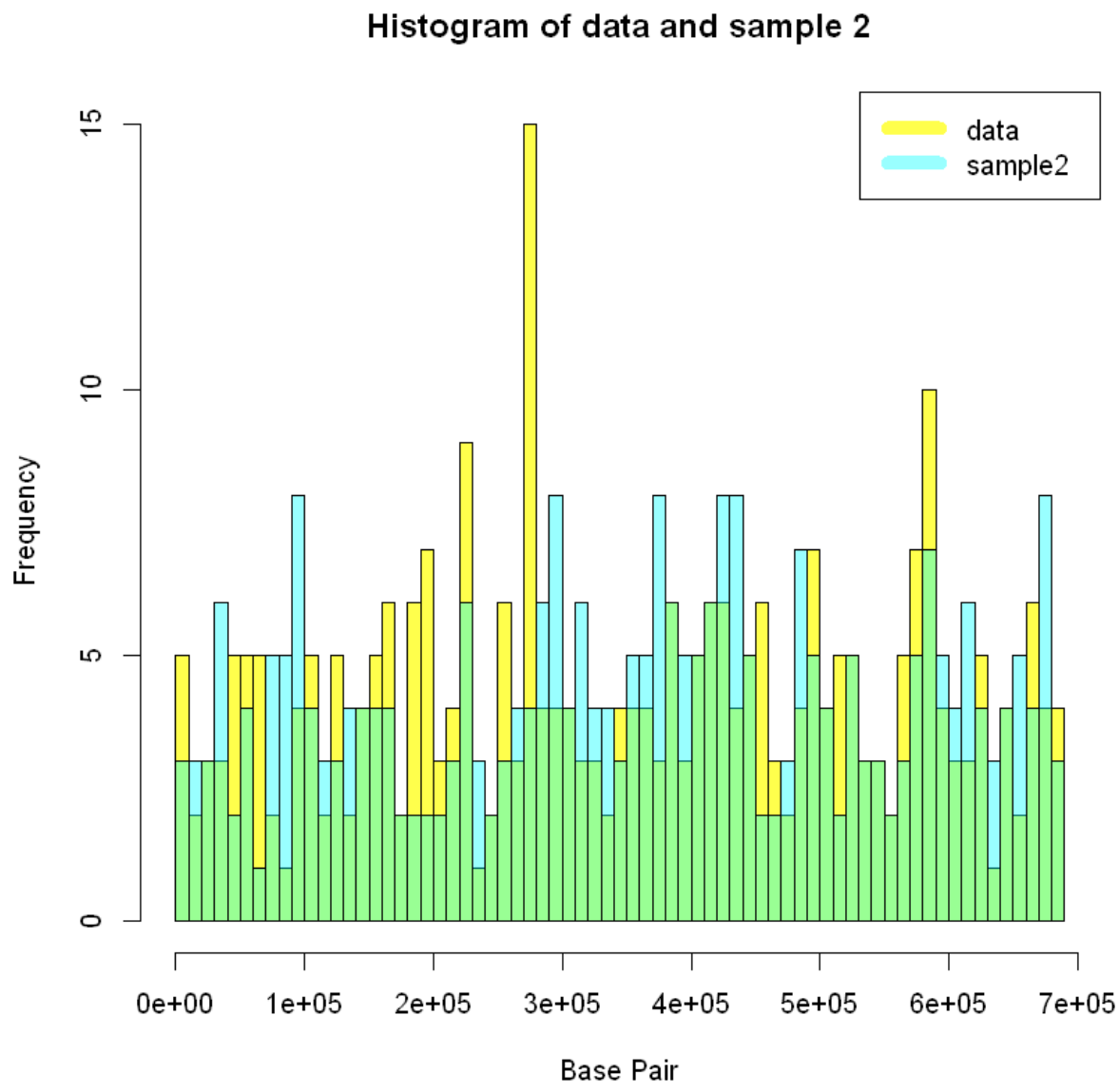
In [14]:
```
#Spacing between consecutive palindromes: sample 1 and 2 compare

hist(unlist(spaces, use.names=F), breaks=85, col=rgb(1,1,0,0.7), main="Histogram
hist(unlist(spaces2, use.names=F), breaks=85, col=rgb(0,1,1,0.4), add=T)
legend("topright", c("data", "sample2"), col=c(rgb(1,1,0,0.7), rgb(0,1,1,0.4)), l
```



In [15]:
```
#Sum of consecutive pairs: sample 1 and 2 compare

hist(unlist(pairs, use.names=F), breaks=85, col=rgb(1,1,0,0.7), main="Histogram c
hist(unlist(pairs2, use.names=F), breaks=85, col=rgb(0,1,1,0.4), add=T)
legend("topright", c("data", "sample2"), col=c(rgb(1,1,0,0.7), rgb(0,1,1,0.4)), l
```



In [16]:
```
#Sum of consecutive triplets: sample 1 and 2 compare

hist(unlist(trip, use.names=F), breaks=85, col=rgb(1,1,0,0.7), main="Histogram of
hist(unlist(trip2, use.names=F), breaks=85, col=rgb(0,1,1,0.4), add=T)
legend("topright", c("data", "sample2"), col=c(rgb(1,1,0,0.7), rgb(0,1,1,0.4)), l
```

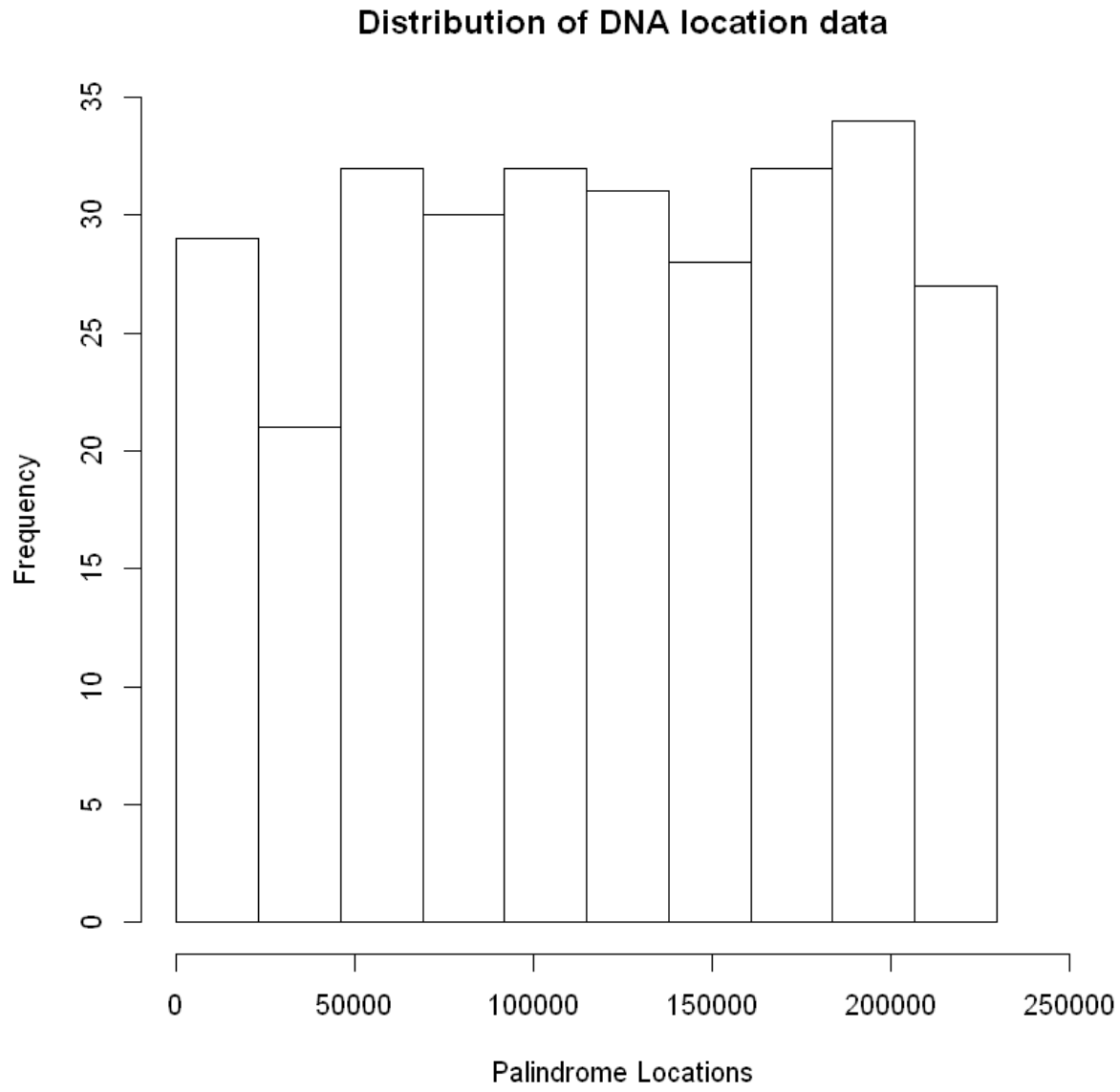


Count

To see the distribution of given data and simulation data, I use histograms to display the results

with 10 equal bins of palindrome locations constructed.

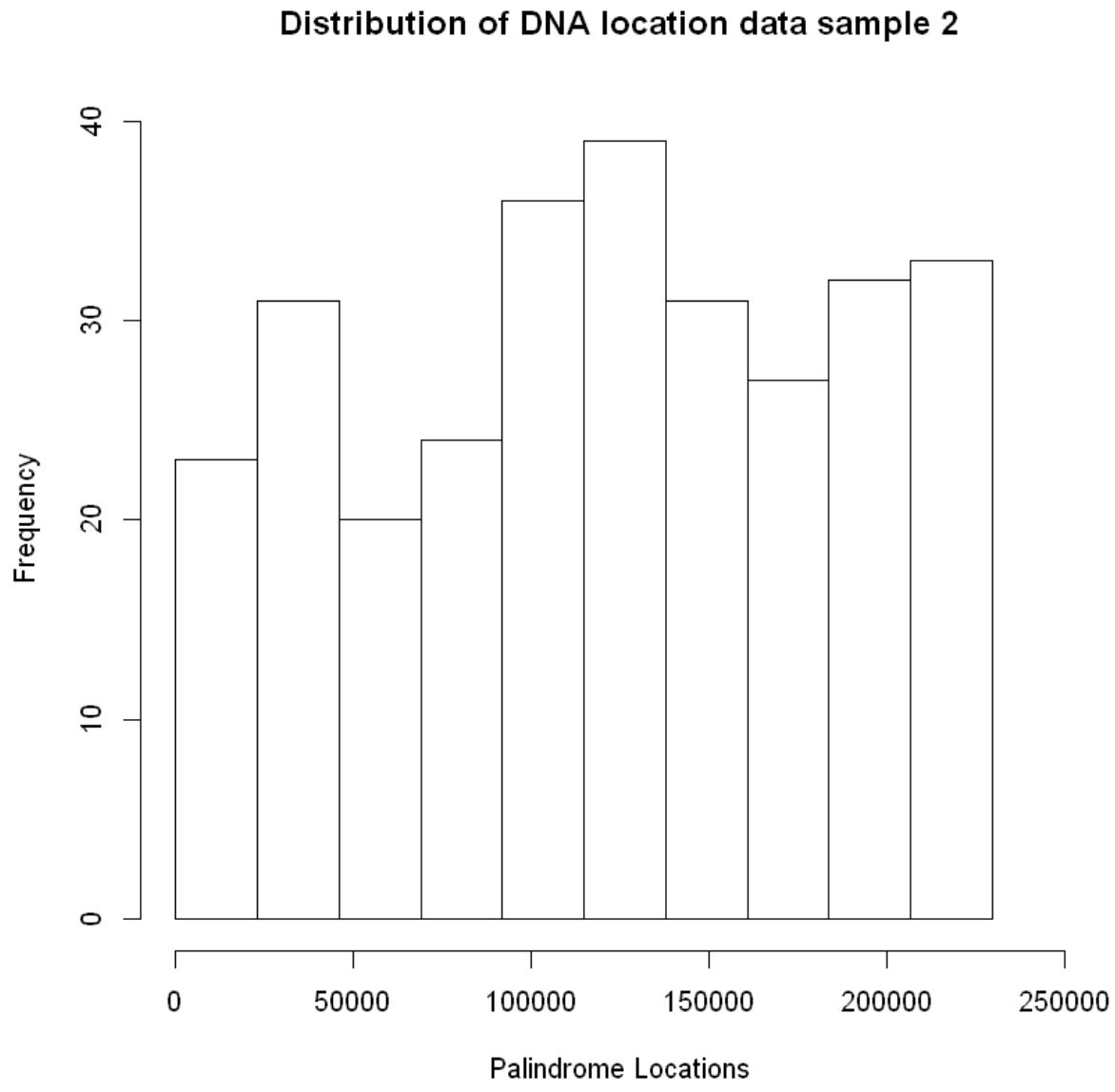
In [17]:
```
#Distribution of DNA location data by counts

hist(all, breaks = seq(0, 229354, by=229354/10),
     main = 'Distribution of DNA location data',
     xlab = "Palindrome Locations",
    xlim=c(0, 250000))
```



Since there are 296 palindromes, and I divided into 10 bins, we would expect to see 29.6 palindromes per bin. Looking at the population and the sample data, we can conclude that the expected value is close to the actual values for both.

In [18]:
```
#Group by each bin: DNA data
breaks <- seq(0, 229354, by=229354/10)
tags <- c('1', '2', '3', '4', '5', '6', '7', '8', '9', '10')
group <- cut(dna$location, breaks, include.lowest=T, right=F, labels=tags)
table_dna <- as.matrix(table(group))
colnames(table_dna) <- 'observed'
rep <- c(rep(29.6, 10))
expected = rep
table_dna <- cbind(table_dna, expected)
```

In [19]:
```
#Distribution of DNA location data sample 2 by counts

hist(two, breaks = seq(0, 229354, by=229354/10),
     main = 'Distribution of DNA location data sample 2',
     xlab = "Palindrome Locations",
    xlim=c(0, 250000),
    ylim=c(0,40))
```



As did in the lecture, we divide the CMV DNA into nonoverlapping regions of length 3000/4000/4500/6000 bases, with the CMV DNA 229354 letters long.

In [20]:
```
homog_pois_process <- function(length,df){
    interval <- 229354 %/% length #number of intervals
    x <- 294/interval # lambda of pois

    value_counts <- as.data.frame(table(cut(df$location,breaks=seq(1,229354,by=40
    res = as.data.frame(value_counts)
    res1 = as.data.frame(table(res$Freq))

    expected <- (interval*dpois((as.integer(res1$Var1)),lambda=x))
    res1$exp <- expected

    names(res1) <- c('Palindrome_Counts','Number_Observed','Number_Expected')
    return (list(x,res,res1))
}
```

In [21]:
```
homog_pois_process(3000,dna)[3]
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
1	5	6.1420509
2	2	11.8800196
3	8	15.3189726
4	10	14.8150590
5	9	11.4621772
6	8	7.3900880
7	5	4.0839960
8	4	1.9748270
9	4	0.8488292
11	1	0.3283629
14	1	0.1154769

In [22]:
```
homog_pois_process(4000,dna)[3]
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
1	5	1.6916173
2	2	4.3625919
3	8	7.5005966
4	10	9.6718219
5	9	9.9772478
6	8	8.5769324
7	5	6.3198449
8	4	4.0746368
9	4	2.3351720
11	1	1.2044571
14	1	0.5647694

In [23]:
```
homog_pois_process(4500,dna)[3]
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
1	5	0.8216669
2	2	2.4157006
3	8	4.7347732
4	10	6.9601166
5	9	8.1850971
6	8	8.0213951
7	5	6.7379719
8	4	4.9524094
9	4	3.2355741
11	1	1.9025176
14	1	1.0169821

In [24]:
```
homog_pois_process(6000,dna)[3]
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
1	5	0.1283156
2	2	0.4963788
3	8	1.2801348
4	10	2.4760503
5	9	3.8313620
6	8	4.9404405
7	5	5.4604868
8	4	5.2808655
9	4	4.5396914
11	1	3.5122876
14	1	2.4703649

Perform chi square test on each case

In [25]:
```
chi_sq_test <- function(result){
    lambda = as.numeric(result[1])
    obj = as.data.frame(result[3])
    p=c( dpois(0:9,lambda), 1-sum(dpois(0:9,lambda)))
    return (chisq.test(obj$Number_Observed,p=p))
}
```

In [26]:
```
chi_sq_test(homog_pois_process(3000,dna))
```

```
Warning message in chisq.test(obj$Number_Observed, p = p):
"Chi-squared approximation may be incorrect"


        Chi-squared test for given probabilities

data:  obj$Number_Observed
X-squared = 20.278, df = 10, p-value = 0.02673
```

In [27]:
```
chi_sq_test(homog_pois_process(4000,dna))
```

```
Warning message in chisq.test(obj$Number_Observed, p = p):
"Chi-squared approximation may be incorrect"


        Chi-squared test for given probabilities

data:  obj$Number_Observed
X-squared = 74.648, df = 10, p-value = 5.571e-12
```

In [28]:
```
chi_sq_test(homog_pois_process(4500,dna))
```

```
Warning message in chisq.test(obj$Number_Observed, p = p):
"Chi-squared approximation may be incorrect"


        Chi-squared test for given probabilities

data:  obj$Number_Observed
X-squared = 171.19, df = 10, p-value < 2.2e-16
```

In [29]:
```
chi_sq_test(homog_pois_process(6000,dna))
```

```
Warning message in chisq.test(obj$Number_Observed, p = p):
"Chi-squared approximation may be incorrect"


        Chi-squared test for given probabilities

data:  obj$Number_Observed
X-squared = 1147.2, df = 10, p-value < 2.2e-16
```

Perform same test on random samples

In [39]:
```
sp1 = data.frame('location' = one)
sp2 = data.frame('location' = two)
sp3 = data.frame('location' = three)
```

In [40]:
```
homog_pois_process(4000,sp1)[3]
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
0	1	1.6916173
1	2	4.3625919
2	4	7.5005966
3	6	9.6718219
4	11	9.9772478
5	8	8.5769324
6	8	6.3198449
7	10	4.0746368
8	2	2.3351720
9	2	1.2044571
10	3	0.5647694

In [41]:
```
homog_pois_process(6000,sp2)[3]
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
0	1	0.1283156
1	1	0.4963788
2	4	1.2801348
3	8	2.4760503
4	7	3.8313620
5	14	4.9404405
6	8	5.4604868
7	4	5.2808655
8	6	4.5396914
9	1	3.5122876
10	2	2.4703649
11	1	1.5927353

In [42]:
```
homog_pois_process(6000,sp3)[3]
chi_sq_test(homog_pois_process(6000,sp3))
```

1.

Palindrome_Counts	Number_Observed	Number_Expected
1	2	0.1283156
2	3	0.4963788
3	8	1.2801348
4	11	2.4760503
5	13	3.8313620
6	4	4.9404405
7	8	5.4604868
8	5	5.2808655
9	1	4.5396914
11	1	3.5122876
12	1	2.4703649

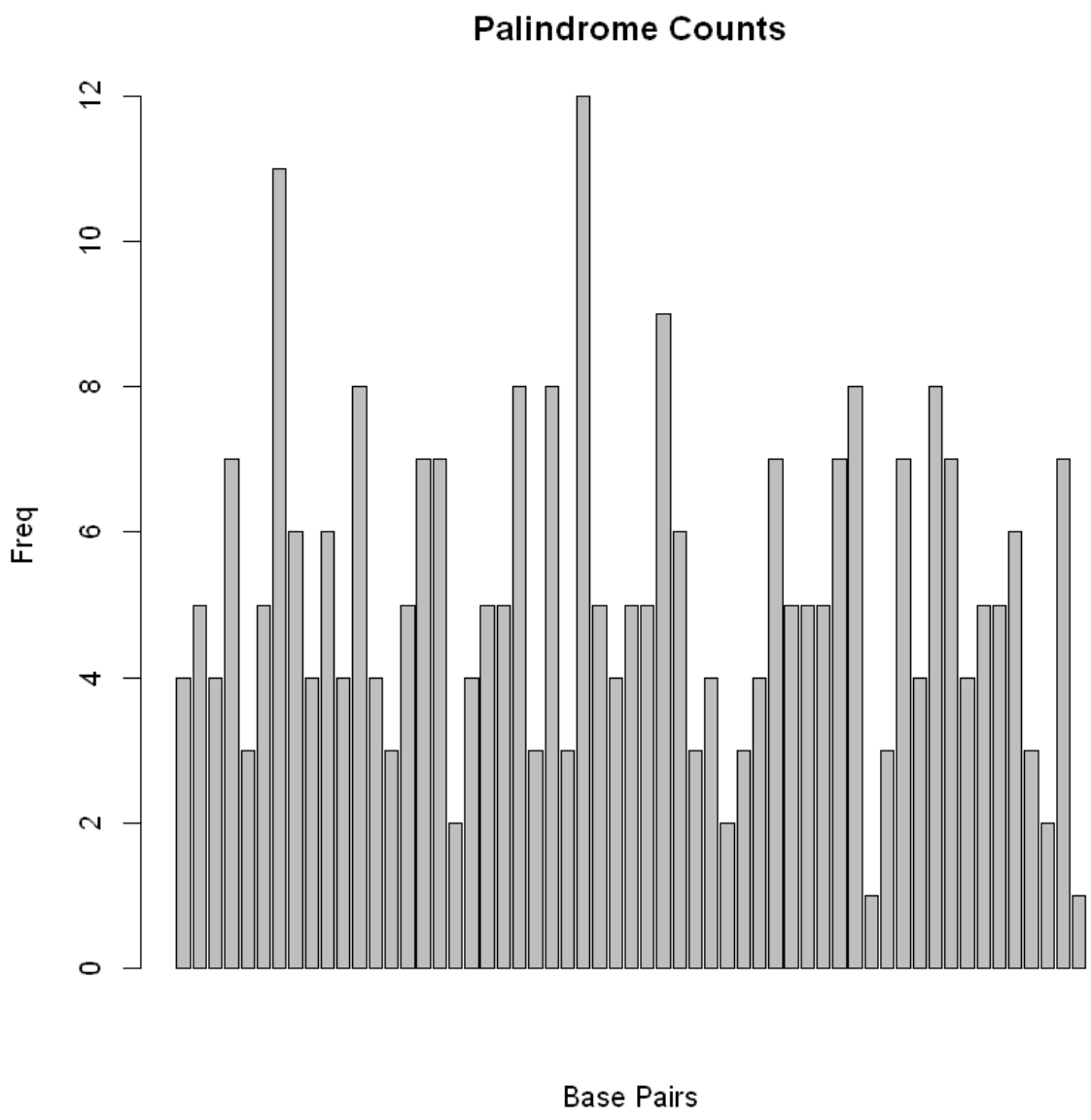
```
Warning message in chisq.test(obj$Number_Observed, p = p):
"Chi-squared approximation may be incorrect"


        Chi-squared test for given probabilities

data:  obj$Number_Observed
X-squared = 359.84, df = 10, p-value < 2.2e-16
```

Cluster

In [43]:
```
df = as.data.frame(homog_pois_process(4500,dna)[2])
df
```

(4.4e+04,4.8e+04]	2
(4.8e+04,5.2e+04]	5
(5.2e+04,5.6e+04]	8
(5.6e+04,6e+04]	2
(6e+04,6.4e+04]	9
(6.4e+04,6.8e+04]	6
(6.8e+04,7.2e+04]	4
(7.2e+04,7.6e+04]	9
(7.6e+04,8e+04]	4
(8e+04,8.4e+04]	1
(8.4e+04,8.8e+04]	7
(8.8e+04,9.2e+04]	7
(9.2e+04,9.6e+04]	14

In [49]:
```
barplot(df$Freq, ylab='Freq', main='Palindrome Counts', xlab='Base Pairs')
```

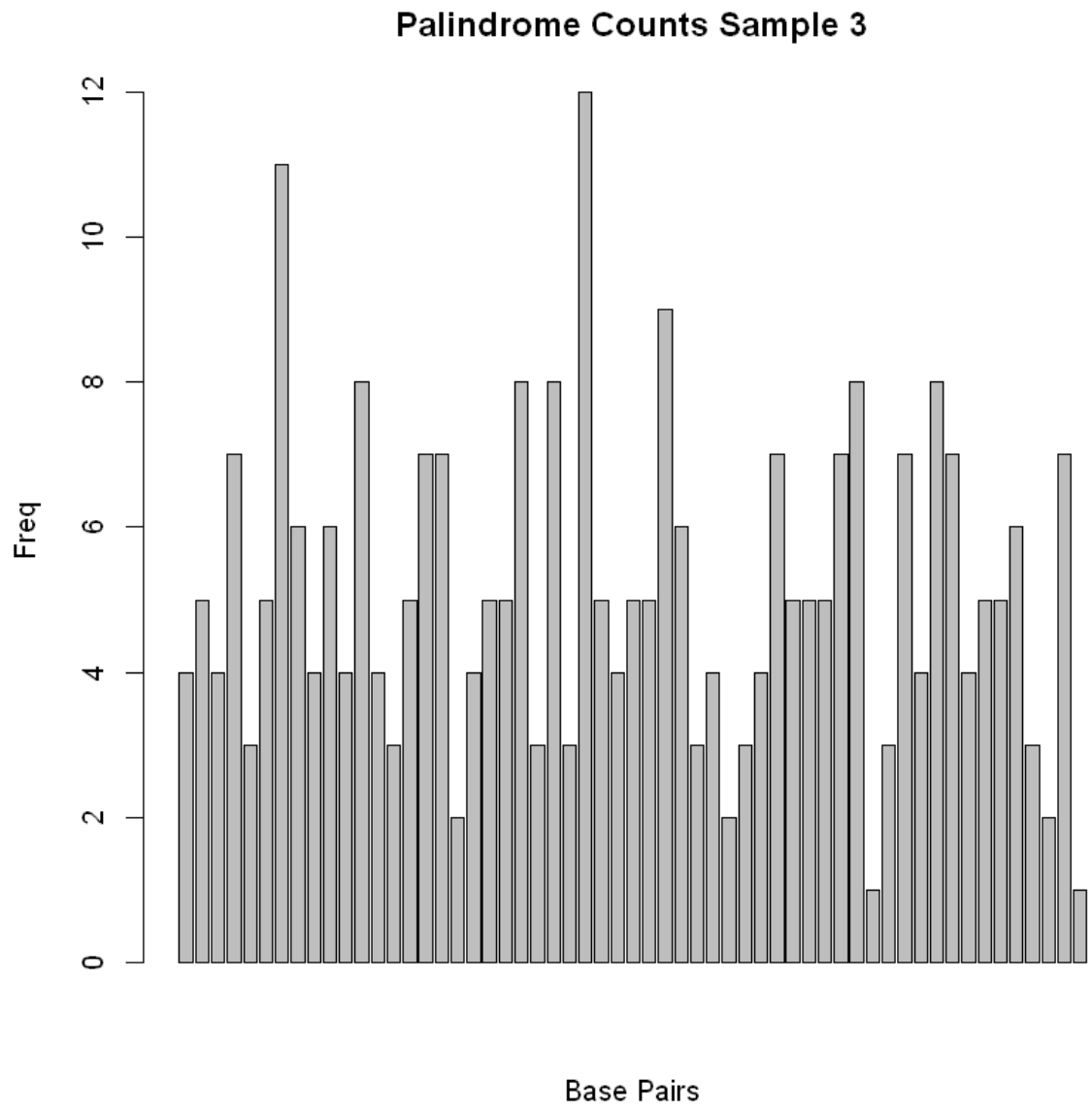


In [47]:
```
df3 = as.data.frame(homog_pois_process(6000,sp3)[2])
df3
```

Var1	Freq
(1,4e+03]	4
(4e+03,8e+03]	5
(8e+03,1.2e+04]	4
(1.2e+04,1.6e+04]	7
(1.6e+04,2e+04]	3
(2e+04,2.4e+04]	5
(2.4e+04,2.8e+04]	11
(2.8e+04,3.2e+04]	6
(3.2e+04,3.6e+04]	4
(3.6e+04,4e+04]	6
(4e+04,4.4e+04]	4
(4.4e+04,4.8e+04]	8
(4.8e+04,5.2e+04]	4
(5.2e+04,5.6e+04]	3
(5.6e+04,6e+04]	5
(6e+04,6.4e+04]	7
(6.4e+04,6.8e+04]	7
(6.8e+04,7.2e+04]	2
(7.2e+04,7.6e+04]	4
(7.6e+04,8e+04]	5
(8e+04,8.4e+04]	5
(8.4e+04,8.8e+04]	8
(8.8e+04,9.2e+04]	3
(9.2e+04,9.6e+04]	8
(9.6e+04,1e+05]	3
(1e+05,1.04e+05]	12
(1.04e+05,1.08e+05]	5
(1.08e+05,1.12e+05]	4
(1.12e+05,1.16e+05]	5
(1.16e+05,1.2e+05]	5
(1.2e+05,1.24e+05]	9
(1.24e+05,1.28e+05]	6
(1.28e+05,1.32e+05]	3
(1.32e+05,1.36e+05]	4

Var1	Freq
(1.36e+05,1.4e+05]	2
(1.4e+05,1.44e+05]	3
(1.44e+05,1.48e+05]	4
(1.48e+05,1.52e+05]	7
(1.52e+05,1.56e+05]	5
(1.56e+05,1.6e+05]	5
(1.6e+05,1.64e+05]	5
(1.64e+05,1.68e+05]	7
(1.68e+05,1.72e+05]	8
(1.72e+05,1.76e+05]	1
(1.76e+05,1.8e+05]	3
(1.8e+05,1.84e+05]	7
(1.84e+05,1.88e+05]	4
(1.88e+05,1.92e+05]	8
(1.92e+05,1.96e+05]	7
(1.96e+05,2e+05]	4
(2e+05,2.04e+05]	5
(2.04e+05,2.08e+05]	5
(2.08e+05,2.12e+05]	6
(2.12e+05,2.16e+05]	3
(2.16e+05,2.2e+05]	2
(2.2e+05,2.24e+05]	7
(2.24e+05,2.28e+05]	1

In [51]:
```
barplot(df3$Freq, ylab='Freq', main='Palindrome Counts Sample 3', xlab='Base Pair
```



In [53]:
```
as.data.frame(homog_pois_process(4000,dna)[2])
```

Var1	Freq
(1,4e+03]	7
(4e+03,8e+03]	1
(8e+03,1.2e+04]	5
(1.2e+04,1.6e+04]	3
(1.6e+04,2e+04]	8
(2e+04,2.4e+04]	6
(2.4e+04,2.8e+04]	1
(2.8e+04,3.2e+04]	4
(3.2e+04,3.6e+04]	5
(3.6e+04,4e+04]	3
(4e+04,4.4e+04]	6
(4.4e+04,4.8e+04]	2
(4.8e+04,5.2e+04]	5
(5.2e+04,5.6e+04]	8
(5.6e+04,6e+04]	2
(6e+04,6.4e+04]	9
(6.4e+04,6.8e+04]	6
(6.8e+04,7.2e+04]	4
(7.2e+04,7.6e+04]	9
(7.6e+04,8e+04]	4
(8e+04,8.4e+04]	1
(8.4e+04,8.8e+04]	7
(8.8e+04,9.2e+04]	7
(9.2e+04,9.6e+04]	14
(9.6e+04,1e+05]	4
(1e+05,1.04e+05]	4
(1.04e+05,1.08e+05]	4
(1.08e+05,1.12e+05]	3
(1.12e+05,1.16e+05]	5
(1.16e+05,1.2e+05]	5
(1.2e+05,1.24e+05]	3
(1.24e+05,1.28e+05]	6
(1.28e+05,1.32e+05]	5
(1.32e+05,1.36e+05]	3
(1.36e+05,1.4e+05]	9

Var1	Freq
(1.4e+05,1.44e+05]	9
(1.44e+05,1.48e+05]	4
(1.48e+05,1.52e+05]	5
(1.52e+05,1.56e+05]	6
(1.56e+05,1.6e+05]	1
(1.6e+05,1.64e+05]	7
(1.64e+05,1.68e+05]	6
(1.68e+05,1.72e+05]	7
(1.72e+05,1.76e+05]	5
(1.76e+05,1.8e+05]	3
(1.8e+05,1.84e+05]	4
(1.84e+05,1.88e+05]	4
(1.88e+05,1.92e+05]	8
(1.92e+05,1.96e+05]	11
(1.96e+05,2e+05]	5
(2e+05,2.04e+05]	3
(2.04e+05,2.08e+05]	6
(2.08e+05,2.12e+05]	3
(2.12e+05,2.16e+05]	1
(2.16e+05,2.2e+05]	4
(2.2e+05,2.24e+05]	8
(2.24e+05,2.28e+05]	6

In [54]:
```
as.data.frame(homog_pois_process(3000,dna)[2])
```

Var1	Freq
(1,4e+03]	7
(4e+03,8e+03]	1
(8e+03,1.2e+04]	5
(1.2e+04,1.6e+04]	3
(1.6e+04,2e+04]	8
(2e+04,2.4e+04]	6
(2.4e+04,2.8e+04]	1
(2.8e+04,3.2e+04]	4
(3.2e+04,3.6e+04]	5
(3.6e+04,4e+04]	3
(4e+04,4.4e+04]	6
(4.4e+04,4.8e+04]	2
(4.8e+04,5.2e+04]	5
(5.2e+04,5.6e+04]	8
(5.6e+04,6e+04]	2
(6e+04,6.4e+04]	9
(6.4e+04,6.8e+04]	6
(6.8e+04,7.2e+04]	4
(7.2e+04,7.6e+04]	9
(7.6e+04,8e+04]	4
(8e+04,8.4e+04]	1
(8.4e+04,8.8e+04]	7
(8.8e+04,9.2e+04]	7
(9.2e+04,9.6e+04]	14
(9.6e+04,1e+05]	4
(1e+05,1.04e+05]	4
(1.04e+05,1.08e+05]	4
(1.08e+05,1.12e+05]	3
(1.12e+05,1.16e+05]	5
(1.16e+05,1.2e+05]	5
(1.2e+05,1.24e+05]	3
(1.24e+05,1.28e+05]	6
(1.28e+05,1.32e+05]	5
(1.32e+05,1.36e+05]	3
(1.36e+05,1.4e+05]	9

Var1	Freq
(1.4e+05,1.44e+05]	9
(1.44e+05,1.48e+05]	4
(1.48e+05,1.52e+05]	5
(1.52e+05,1.56e+05]	6
(1.56e+05,1.6e+05]	1
(1.6e+05,1.64e+05]	7
(1.64e+05,1.68e+05]	6
(1.68e+05,1.72e+05]	7
(1.72e+05,1.76e+05]	5
(1.76e+05,1.8e+05]	3
(1.8e+05,1.84e+05]	4
(1.84e+05,1.88e+05]	4
(1.88e+05,1.92e+05]	8
(1.92e+05,1.96e+05]	11
(1.96e+05,2e+05]	5
(2e+05,2.04e+05]	3
(2.04e+05,2.08e+05]	6
(2.08e+05,2.12e+05]	3
(2.12e+05,2.16e+05]	1
(2.16e+05,2.2e+05]	4
(2.2e+05,2.24e+05]	8
(2.24e+05,2.28e+05]	6

In [55]:
```
as.data.frame(homog_pois_process(6000,dna)[2])
```

Var1	Freq
(1,4e+03]	7
(4e+03,8e+03]	1
(8e+03,1.2e+04]	5
(1.2e+04,1.6e+04]	3
(1.6e+04,2e+04]	8
(2e+04,2.4e+04]	6
(2.4e+04,2.8e+04]	1
(2.8e+04,3.2e+04]	4
(3.2e+04,3.6e+04]	5
(3.6e+04,4e+04]	3
(4e+04,4.4e+04]	6
(4.4e+04,4.8e+04]	2
(4.8e+04,5.2e+04]	5
(5.2e+04,5.6e+04]	8
(5.6e+04,6e+04]	2
(6e+04,6.4e+04]	9
(6.4e+04,6.8e+04]	6
(6.8e+04,7.2e+04]	4
(7.2e+04,7.6e+04]	9
(7.6e+04,8e+04]	4
(8e+04,8.4e+04]	1
(8.4e+04,8.8e+04]	7
(8.8e+04,9.2e+04]	7
(9.2e+04,9.6e+04]	14
(9.6e+04,1e+05]	4
(1e+05,1.04e+05]	4
(1.04e+05,1.08e+05]	4
(1.08e+05,1.12e+05]	3
(1.12e+05,1.16e+05]	5
(1.16e+05,1.2e+05]	5
(1.2e+05,1.24e+05]	3
(1.24e+05,1.28e+05]	6
(1.28e+05,1.32e+05]	5
(1.32e+05,1.36e+05]	3
(1.36e+05,1.4e+05]	9

Var1	Freq
(1.4e+05,1.44e+05]	9
(1.44e+05,1.48e+05]	4
(1.48e+05,1.52e+05]	5
(1.52e+05,1.56e+05]	6
(1.56e+05,1.6e+05]	1
(1.6e+05,1.64e+05]	7
(1.64e+05,1.68e+05]	6
(1.68e+05,1.72e+05]	7
(1.72e+05,1.76e+05]	5
(1.76e+05,1.8e+05]	3
(1.8e+05,1.84e+05]	4
(1.84e+05,1.88e+05]	4
(1.88e+05,1.92e+05]	8
(1.92e+05,1.96e+05]	11
(1.96e+05,2e+05]	5
(2e+05,2.04e+05]	3
(2.04e+05,2.08e+05]	6
(2.08e+05,2.12e+05]	3
(2.12e+05,2.16e+05]	1
(2.16e+05,2.2e+05]	4
(2.2e+05,2.24e+05]	8
(2.24e+05,2.28e+05]	6

In [ ]: