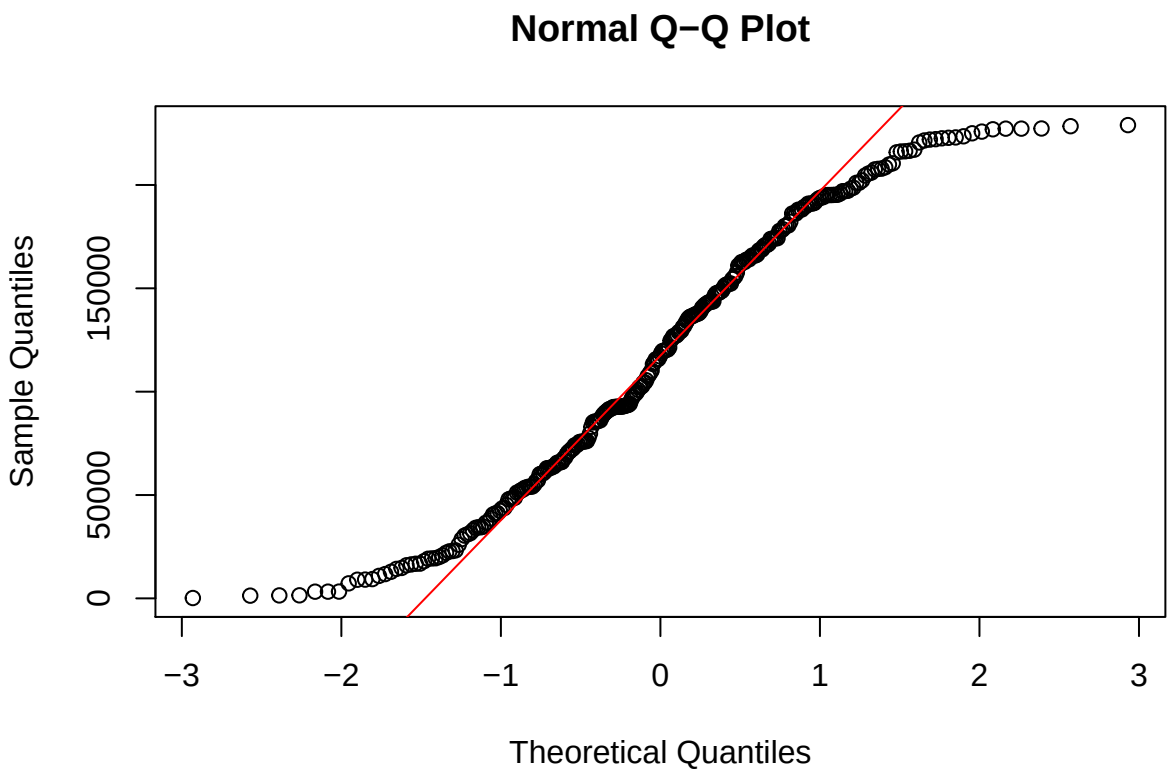
Lab2

Leomart Crisostomo

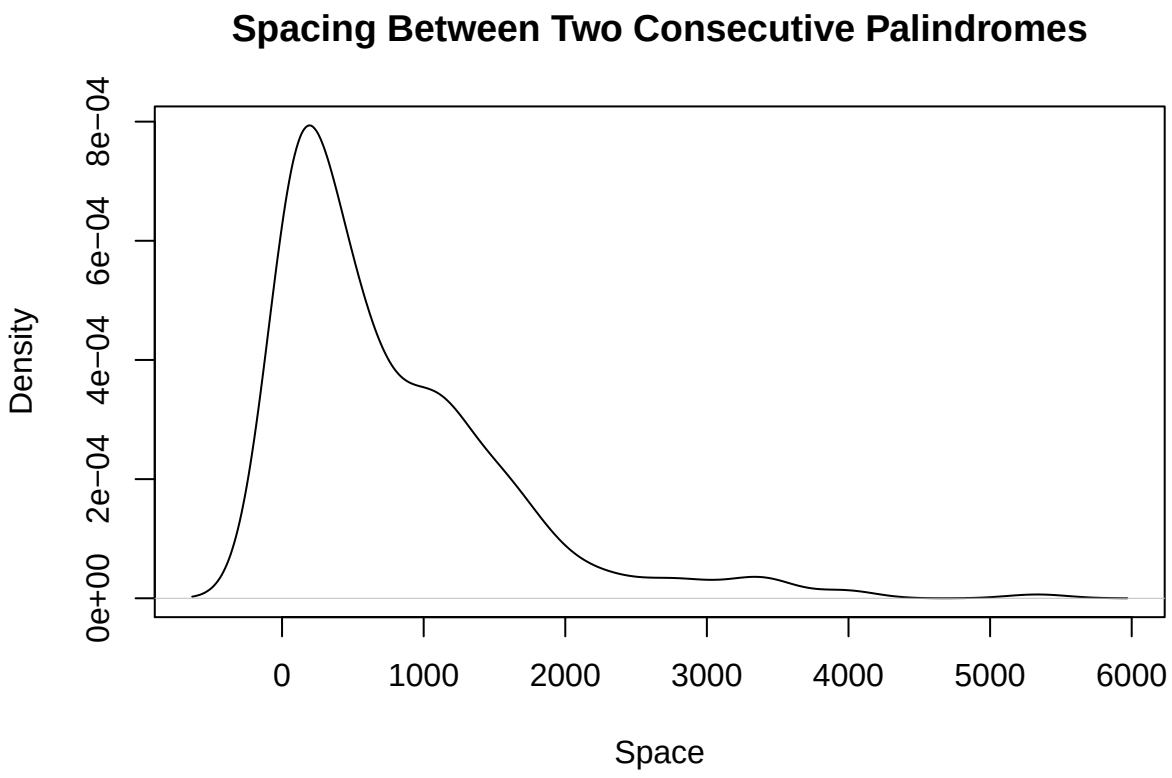
3/22/2018

```
col_name = "Location"
col_type = "integer"
hcmv <- read.csv(file = "hcmv.data",
                 sep = ",",
                 header = TRUE,
                 na.string = "?",
                 col.names = col_name,
                 colClasses = col_type
                 )
locations = hcmv$Location
qqnorm(locations)
qqline(locations, col = 'red')
```

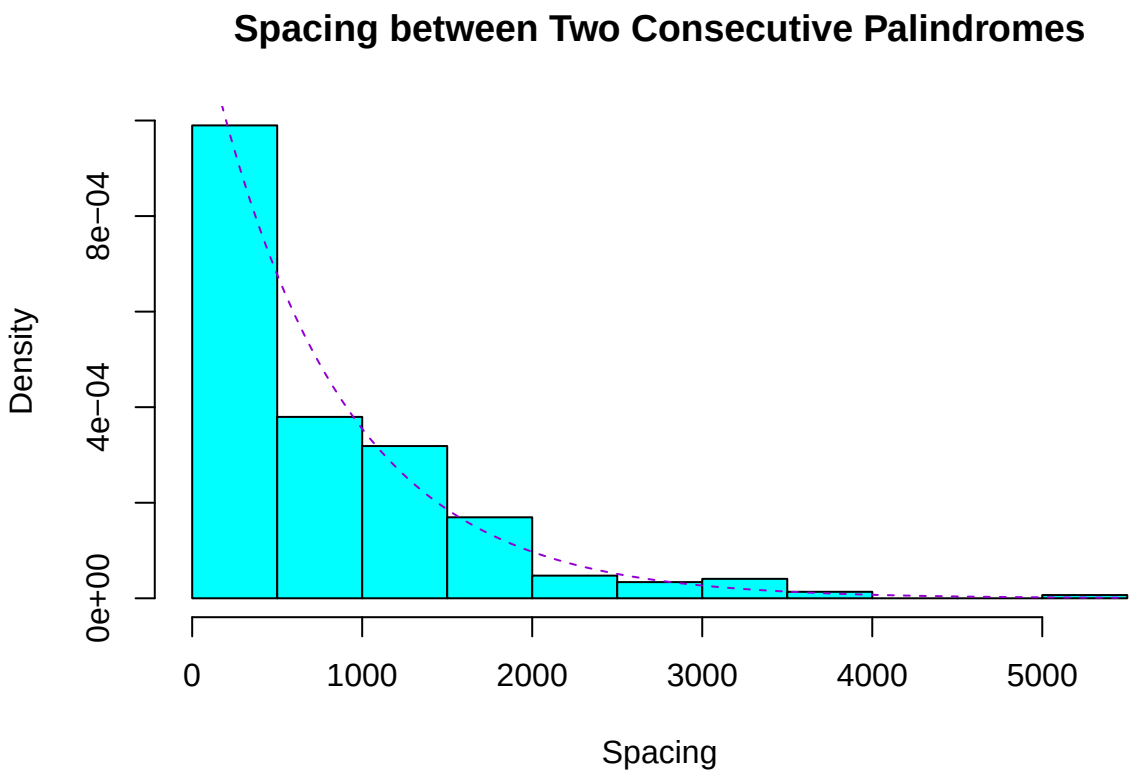



```
# Spacings of Palindromes
spacing = locations[-1] - locations[-296]

# exponential
range = locations[296] - locations[1]
rate_exp = 296/range
x = c(0:max(spacing))
plot(density(spacing), main = "Spacing Between Two Consecutive Palindromes", xlab = "Space")
```

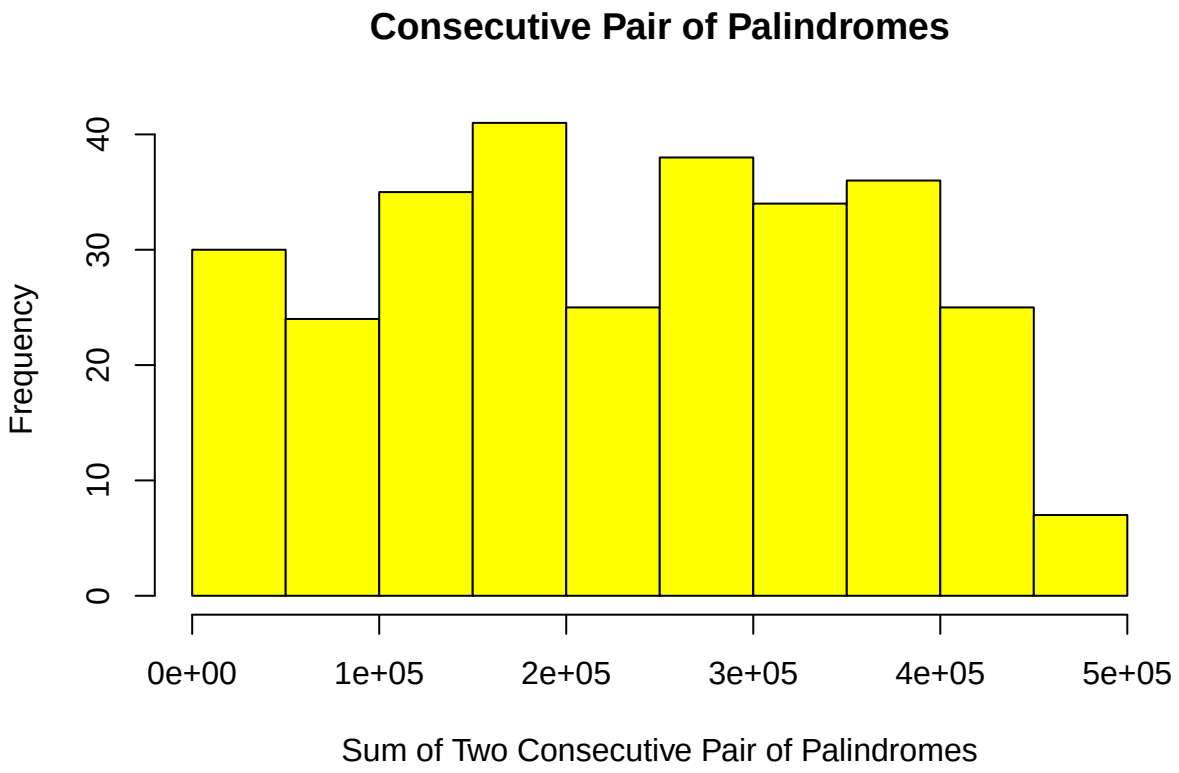



```
hist(spacing, xlab = "Spacing", main = "Spacing between Two Consecutive Palindromes", prob = TRUE, col =
curve(dexp(x,rate_exp), col = 'darkviolet', lty = 2, add = TRUE)
```

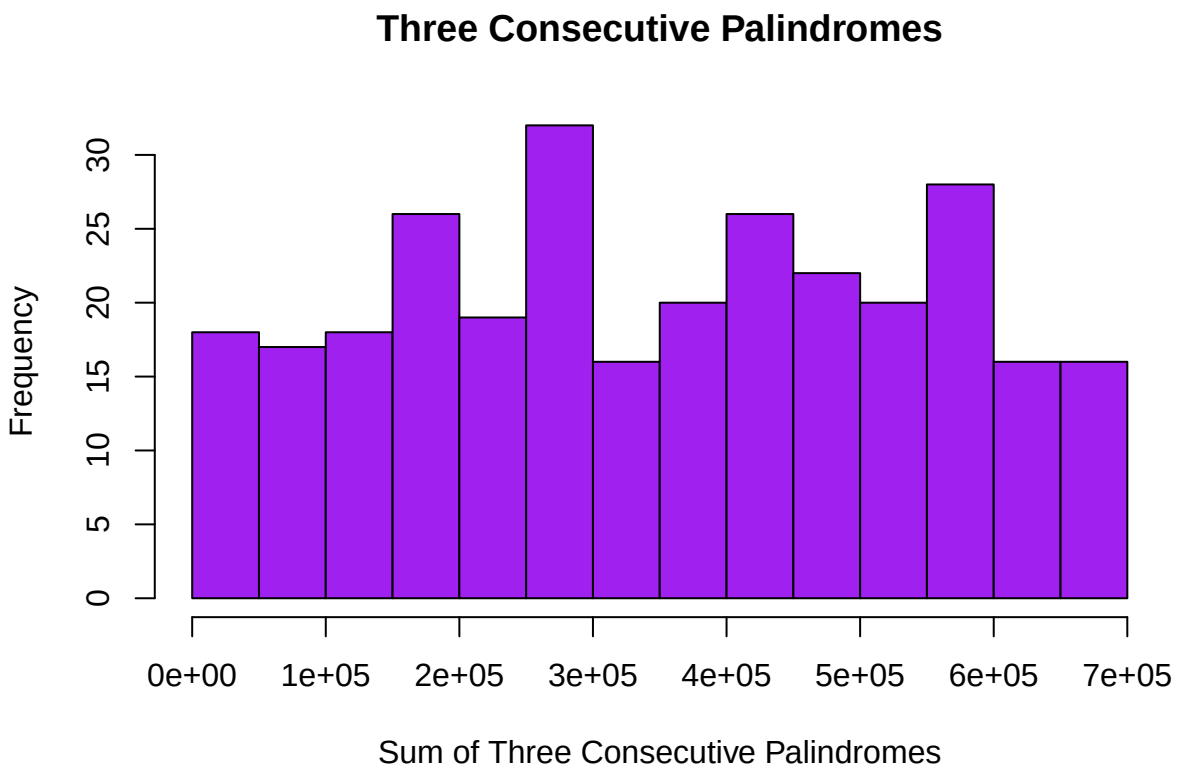



```
# Sum of Consecutive Pair
sum_consecutive_pair = locations[-1] + locations[-296]
```

```
hist(sum_consecutive_pair, main= "Consecutive Pair of Palindromes", xlab = "Sum of Two Consecutive Pair
```



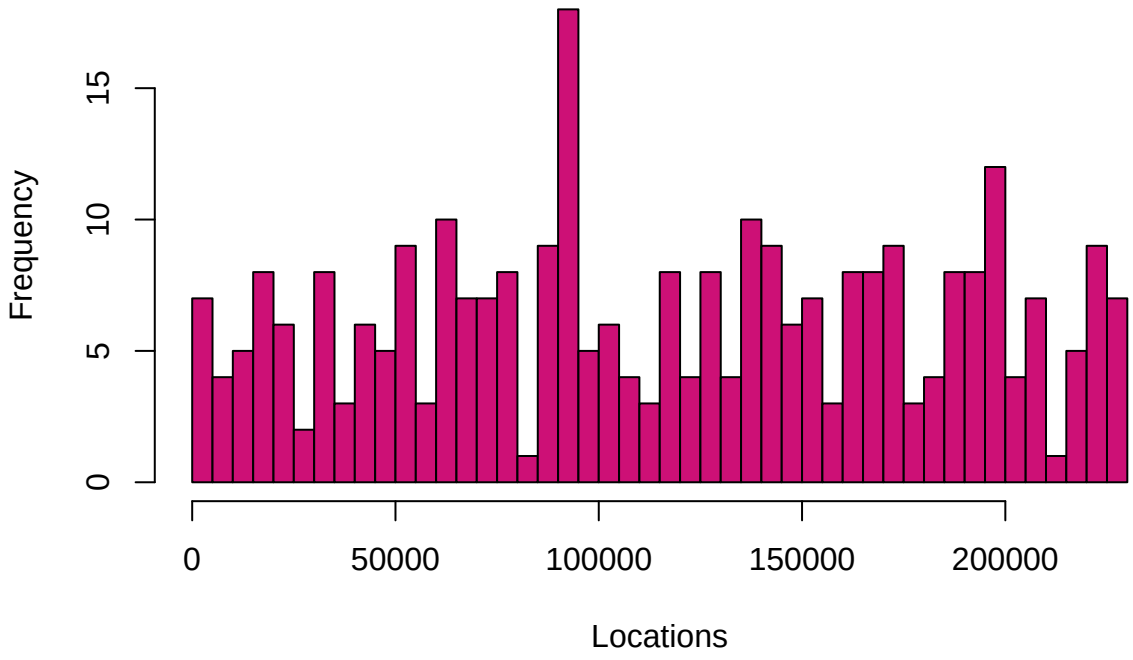

```
# Sum of Three Consecutive Palindromes
grp = 3
sum_consecutive_triplets = sapply(1:(length(locations)-grp+1),function(x){sum(locations[x:(x+grp-1)])})
hist(sum_consecutive_triplets, xlab = "Sum of Three Consecutive Palindromes", main = "Three Consecutive
```




```
# Counts of Palindromes in intervals of 5000
# Use Chi Squared Test
library(knitr)
interval = 5000
max(hcmv) / 5000
```

```
## [1] 45.7906
```

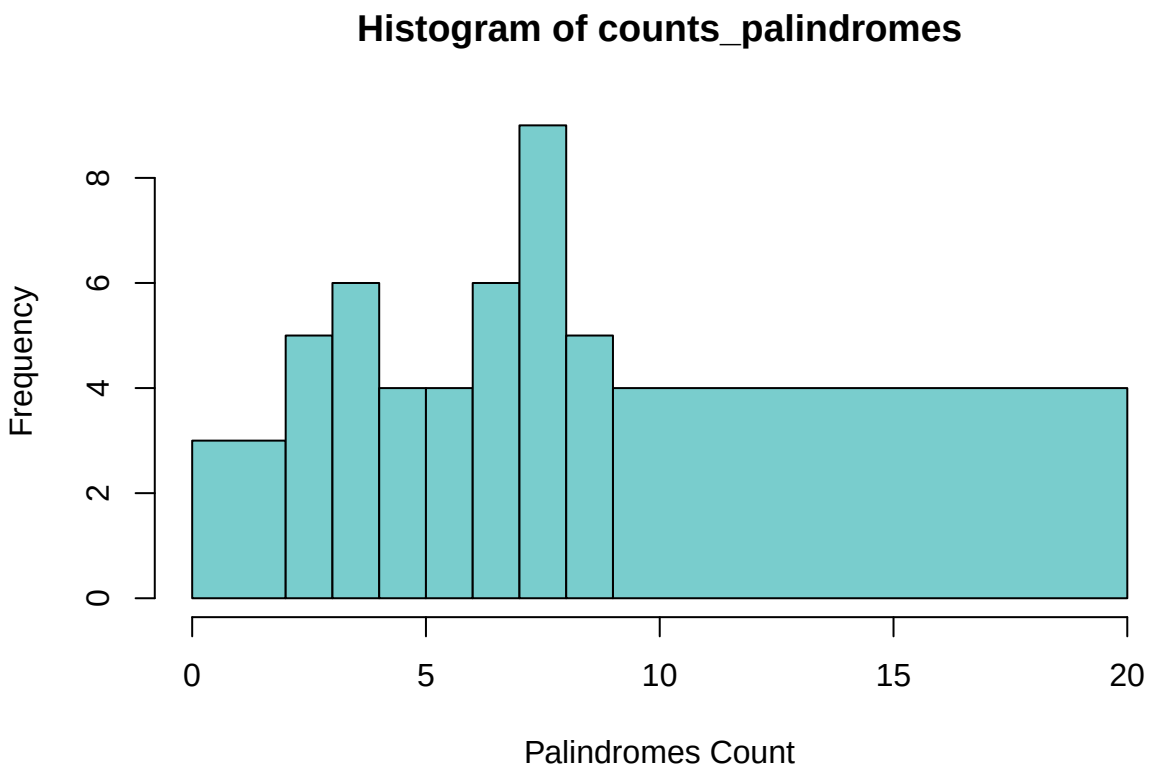
```
bins = c()
for (i in 0:46){
  bins = c(bins, i * interval)
}
counts_hist = hist(locations, breaks = bins, main = "Palindrome Counts in Intervals of 5000", xlab = "L
```




```
counts_palindromes = counts_hist$counts
palindrome_bin_count = c(0,2,3,4,5,6,7,8,9,20)

counts_palindromes_hist = hist(counts_palindromes, breaks = palindrome_bin_count , freq = TRUE, xlab =
```

```
## Warning in plot.histogram(r, freq = freq1, col = col, border = border,
## angle = angle, : the AREAS in the plot are wrong -- rather use 'freq =
## FALSE'
```




```
#hist(counts_palindromes, breaks= palindrome_bin_count , freq = TRUE)
observed_num_interval = counts_palindromes_hist$counts
num_intervals = length(counts_palindromes)
rate = length(locations)/num_intervals

expected_0_1_2 = num_intervals * exp(-rate)*(1+rate+ rate^2/2)
expected_0_1_2
```

```
## [1] 2.077114
```

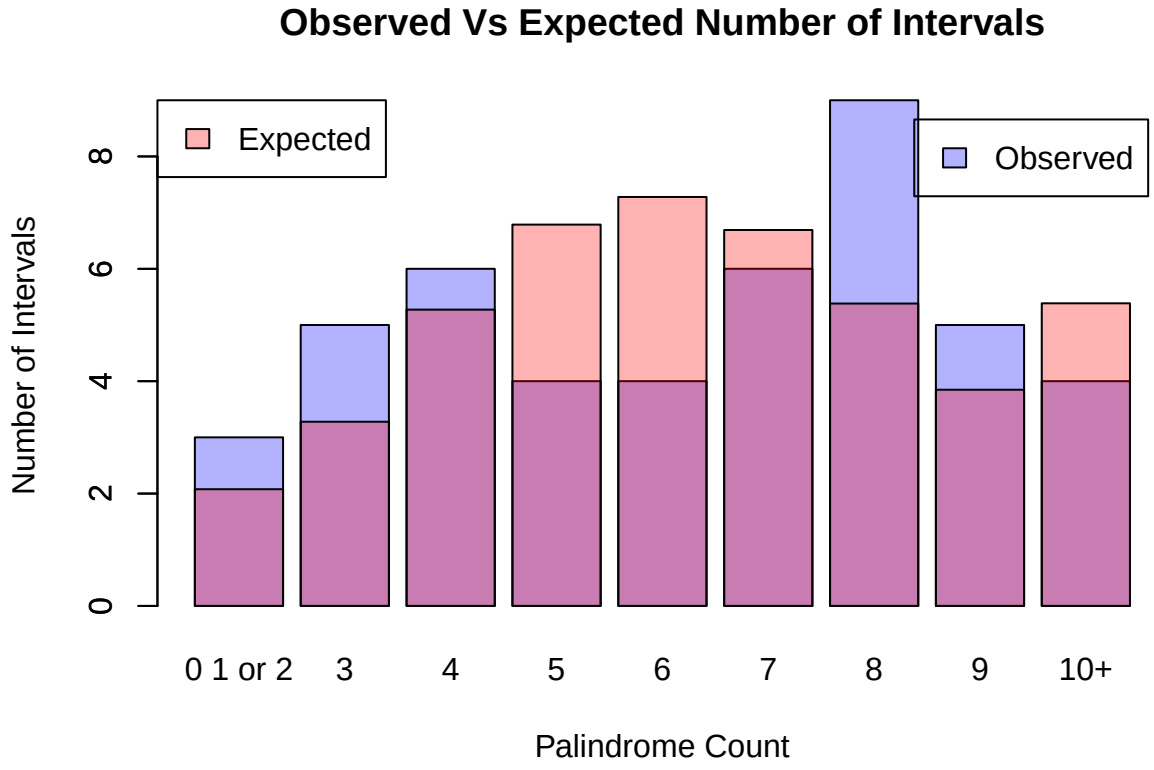
```
expected_num_interval = c(expected_0_1_2)
for (i in 3:9){
  expected_num_interval = c(expected_num_interval, num_intervals* exp(-rate) *(rate^i / factorial(i)))
}
expected_num_interval = c(expected_num_interval, num_intervals- sum(expected_num_interval))

# Table of Observed and Expected Number of Intervals
categories = c('0 1 or 2', 3, 4, 5, 6, 7, 8, 9, '10+')
column_names = c('Palindrome Count', 'Observed Number of Intervals', 'Expected Number of Intervals')
data_table = data.frame(categories, observed_num_interval ,round(expected_num_interval, 2))
kable(data_table, col.names = column_names)
```

Palindrome Count	Observed Number of Intervals	Expected Number of Intervals
0 1 or 2	3	2.08
3	5	3.28
4	6	5.27
5	4	6.79
6	4	7.28
7	6	6.69
8	9	5.38

Palindrome Count	Observed Number of Intervals	Expected Number of Intervals
9	5	3.85
10+	4	5.39

```
barplot(observed_num_interval, col = rgb(0,0,1,.3), names.arg = categories, main = "Observed Vs Expecte
barplot(expected_num_interval,col = rgb(1,0,0,.3),legend = c("Expected"), add=TRUE, args.legend = list(
```

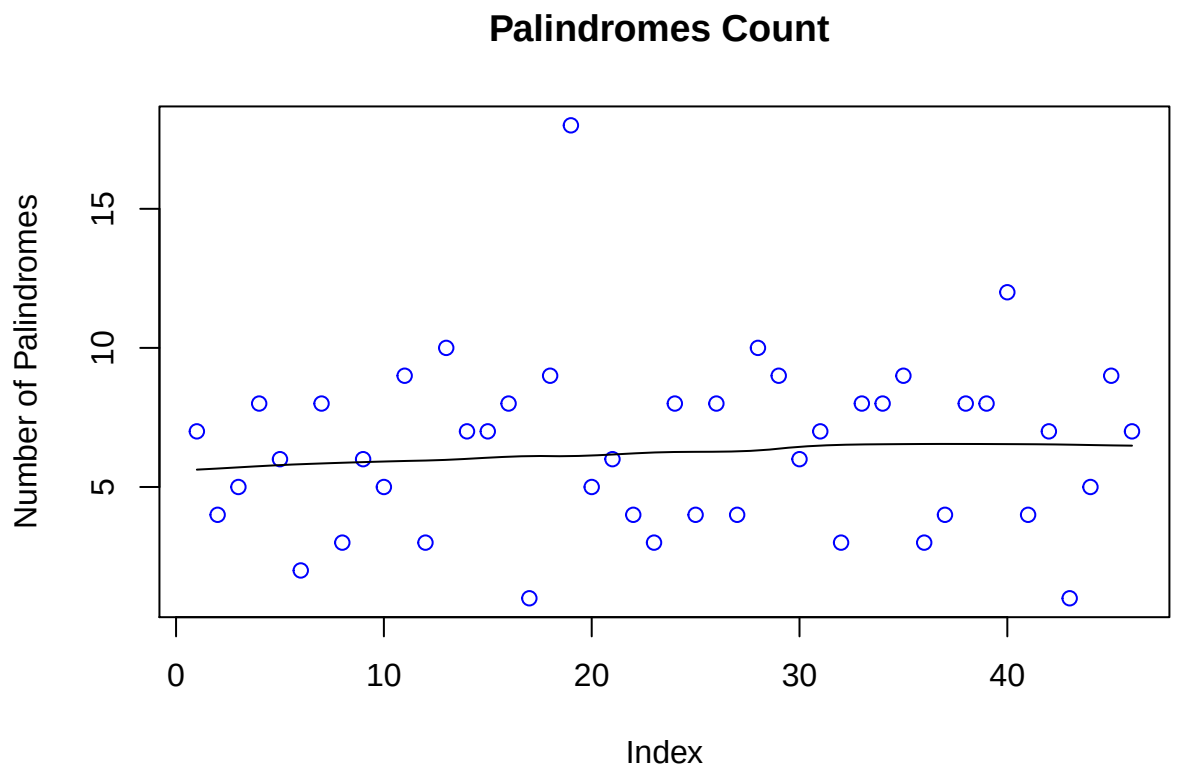



```
# Perform Chi Squared Test
test_statistic = sum((expected_num_interval - observed_num_interval)^2 / expected_num_interval)
test_statistic
```

```
## [1] 7.241427
```

```
p_value = pchisq(test_statistic, df=9, lower.tail = FALSE)
```

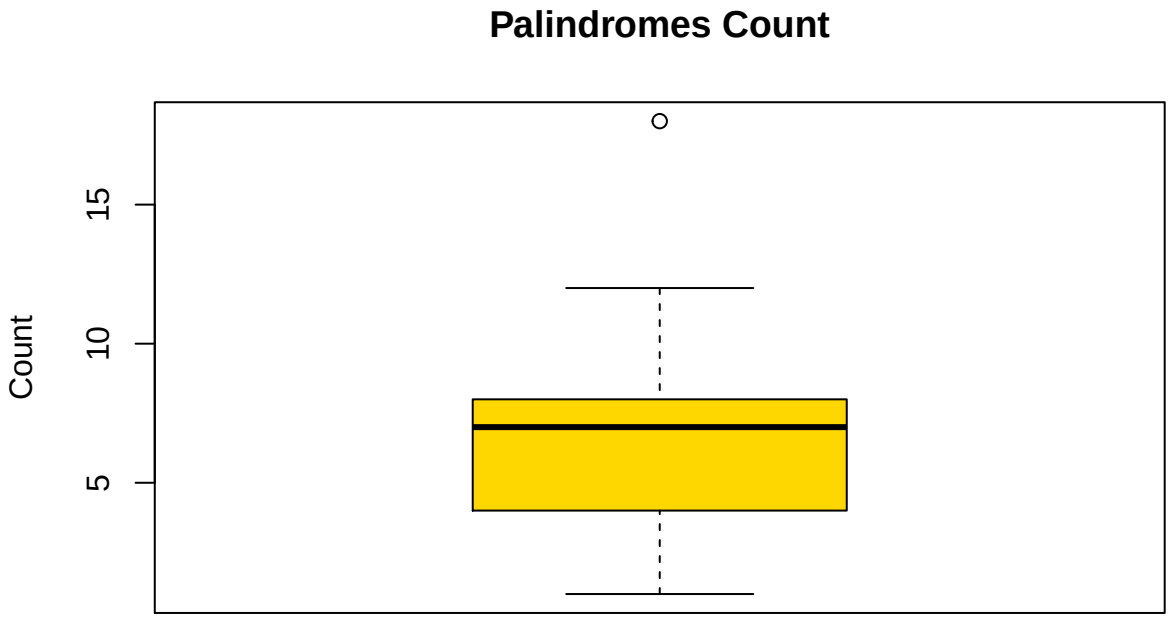
```
# Maximum Number of Hits
scatter.smooth(counts_palindromes, col = 'blue', main = "Palindromes Count ", ylab = 'Number of Palindr
```




```
summary(counts_palindromes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   7.000   6.435   8.000  18.000
```

```
boxplot(counts_palindromes, main = "Palindromes Count", col = 'gold', ylab = 'Count')
```




```
test_statistic_max = max(counts_palindromes)
p_value_max = 0
for (i in 0:test_statistic_max){
  p_value_max = p_value_max + (rate^i *exp(-rate) / factorial(i))
```

```
}
p_value_max = 1- p_value_max
p_value_max
```

```
## [1] 4.434552e-05
```

```
bins_interval = bins[1:length(bins)-1]
data_interval = data.frame(bins_interval, counts_palindromes)
kable(data_interval, col.names = c('Interval', 'Palindromes Count') )
```

Interval	Palindromes Count
0	7
5000	4
10000	5
15000	8
20000	6
25000	2
30000	8
35000	3
40000	6
45000	5
50000	9
55000	3
60000	10
65000	7
70000	7
75000	8
80000	1
85000	9
90000	18
95000	5
100000	6
105000	4
110000	3
115000	8
120000	4
125000	8
130000	4
135000	10
140000	9
145000	6
150000	7
155000	3
160000	8
165000	8
170000	9
175000	3
180000	4
185000	8
190000	8
195000	12
200000	4
205000	7
210000	1

Interval	Palindromes Count
215000	5
220000	9
225000	7

```
max_segment = c(90000, 95000)
```