R Notebook

Intro / Motivation ☹

Nowadays, advertisement has become an integral part of business as such.

It is no longer a question of whether one needs to promote their business or not.

The question now is how and where to promote one's product.

There is a variety of methods and facilities for advertisement.

According to **Instamber**: ***"telegram marketing can be a productive method to promote your business, as you have millions of active Telegram users around the globe."***

Nevertheless, all the given articles' methods assume those entrepreneurs would either spend lots of time developing their channel or pay m

So we come with an alternative approach to promoting a business via Telegram.

Let us assume that we have to promote paid CS courses and materials like books, site subscriptions via Telegram at no cost. In other word
So, naturally, we would choose thematic groups for promoting our products. For example, we can take a Python programming group.

It is the right choice because the majority of members of this group are interested in CS. Everything seems to be perfect for our purpose, b
which ads are considered to be. So we need to embed the advertisement in the message and send them in a way that will not make admins

The way of embedding an advertisement into a message is neither a matter of probability theory nor statistics.

Let us assume that we come up with an idea of how to do that. Nevertheless, our plan might have drawbacks. For example, admins can be
However, that is not a big problem for us, and we can send one message a week, not risking being kicked from the group.

So, an interesting question arises, **on what day of the week and at what time should we send hidden ads to influence as many group members as possible.**

This is where the Statistics may come in handy.

```
#Here we read our data
require(chron)
```

```
## Loading required package: chron
```

```
## Warning: package 'chron' was built under R version 4.0.3
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
library(chron)
library(ggplot2)
library(fitdistrplus)
```

```
## Warning: package 'fitdistrplus' was built under R version 4.0.3
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.0.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(logspline)
```

```
## Warning: package 'logspline' was built under R version 4.0.3
```

```
process <- function(date)
{
   dt_tm <- substring(date, 1, 19)
   dt_tm <- t(as.data.frame(strsplit(dt_tm,' ')))
   row.names(dt_tm) = NULL
   dt_tm <- chron(dates=dt_tm[,1], times=dt_tm[,2], format=c('y-m-d','h:m:s'))
   return(dt_tm)
}


df_prime <- read.csv("general_df.csv", encoding = "UTF-8")
df_prime$date <- process(df_prime$date)
```

```
#Some insights of data we are using
head(df_prime) %>% select(3,4)
```

	date <chron>	from_id <chr>
1	(21-01-04 21:42:31)	PeerUser(user_id=1233962467)
2	(21-01-04 21:39:35)	PeerUser(user_id=1233962467)
3	(21-01-04 21:38:00)	PeerUser(user_id=1457110487)
4	(21-01-04 21:36:50)	PeerUser(user_id=523062282)
5	(21-01-04 21:33:06)	PeerUser(user_id=197388265)
6	(21-01-04 21:19:16)	PeerUser(user_id=1415754036)

6 rows

```
coronda_index = 244000
df <- df_prime
```

```
#Description of the data
n_msg <- nrow(df)
last <- as.Date(df[1,]$date)
first  <- as.Date(df[nrow(df),]$date)
time_range <- as.numeric(difftime(last, first, units = "days")) + 1
active_days <-length(unique(as.Date(df$date)))
avtive <- active_days / time_range * 100


sprintf("1. Total number of messages:  %d\n", n_msg) %>% cat()
```

```
## 1. Total number of messages:  769855
```

```
sprintf("2. Date of the first message: %s\n", toString(first)) %>% cat()
```

```
## 2. Date of the first message: 2018-03-16
```

```
sprintf("3. Date of the last message:  %s\n", toString(last)) %>% cat()
```

```
## 3. Date of the last message:  2021-01-04
```

```
sprintf("4. Number of days covered:    %d\n", time_range) %>% cat()
```

```
## 4. Number of days covered:    1026
```

```
sprintf("5. Activity percentage:       %d%%\n", avtive) %>% cat()
```

```
## 5. Activity percentage:      100%
```

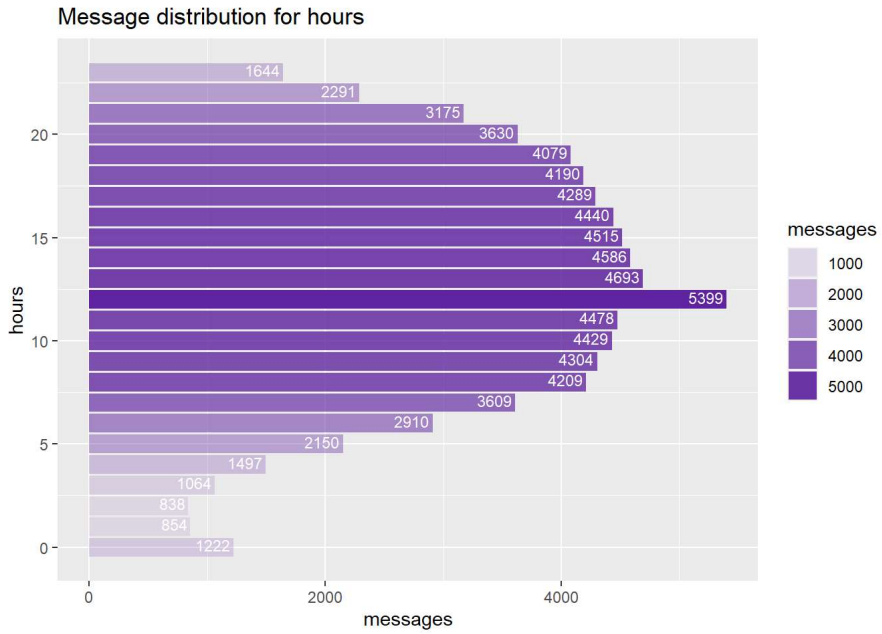
```
#Setting of important constants
hours <- c(0:23)
days <- c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
```

Firsty, lets find message distribution for hours (from 00:00 to 23:00)

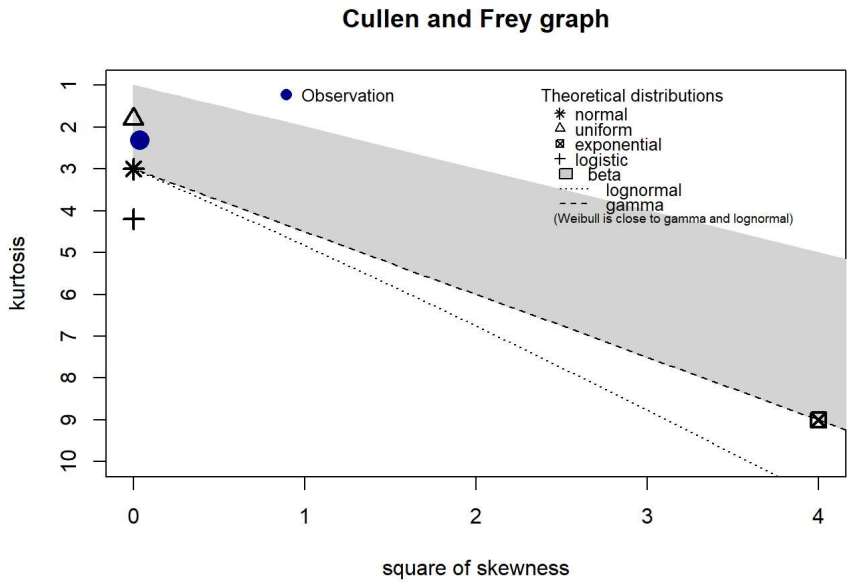
```
#Finding Message distribution for hours
messages <- c()
dt.dst_h <- c()

for (hour in hours)
{
   num <- length(unique(df[hours(df$date) == hour,]$from_id))
   dt.dst_h <- c(dt.dst_h, rep(hour, num))
   messages <- c(messages, num)
}

dt.hours = data.frame(hours = hours, messages = messages)
```

```
#Visulisation
ggplot(data=dt.hours, aes(x=hours, y=messages, alpha=messages)) +
      geom_bar(stat="identity", fill="#5f249f") +
      geom_text(aes(label=messages), vjust=0.3, hjust= 1.1, size=3, color="white", bold=TRUE, alpha=1) +
      labs(title="Message distribution for hours") +
      coord_flip()
```

```
## Warning: Ignoring unknown parameters: bold
```



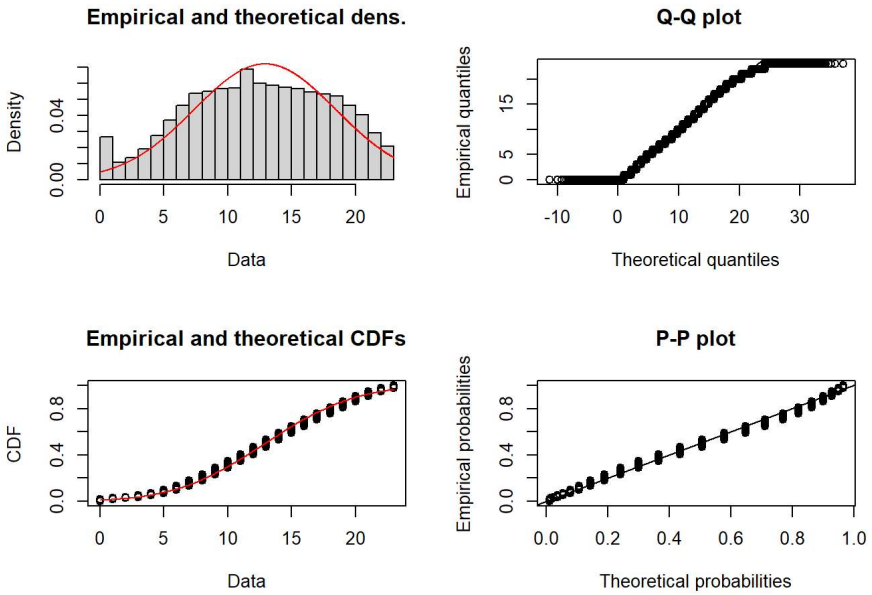

```
descdist(dt.dst_h)
```




```
## summary statistics
## ------
## min:  0   max:  23
## median:  13
## mean:  12.90819
## estimated sd:  5.55151
## estimated skewness:  -0.1921802
## estimated kurtosis:  2.300903
```

As we can say from the plot, distribution is likely to aproximately normal. And we also can say that most messages are concentrated near 12:00

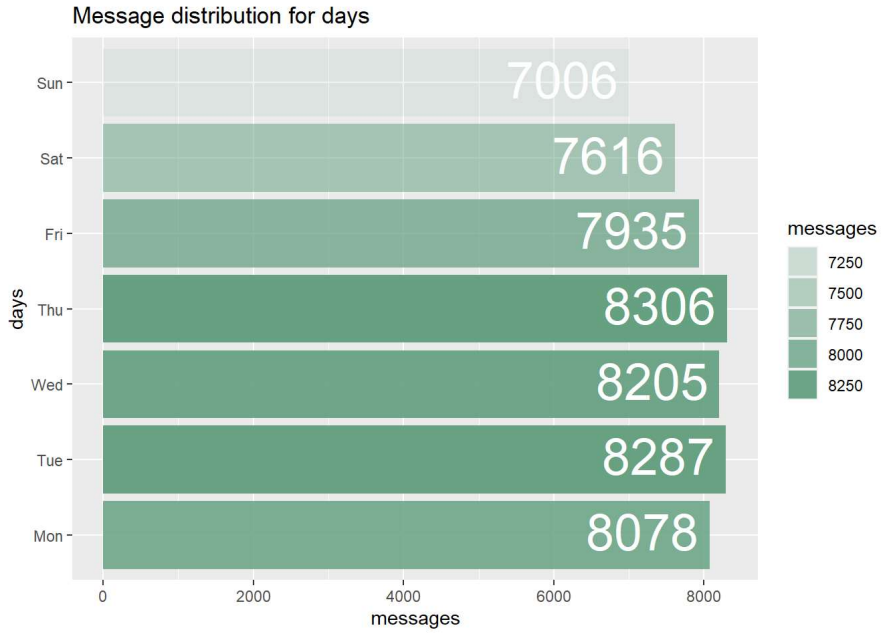
```
plot(fitdist(dt.dst_h, "norm"))
```



Secondly, lets find message distribution for days (from Monday to Sunday)

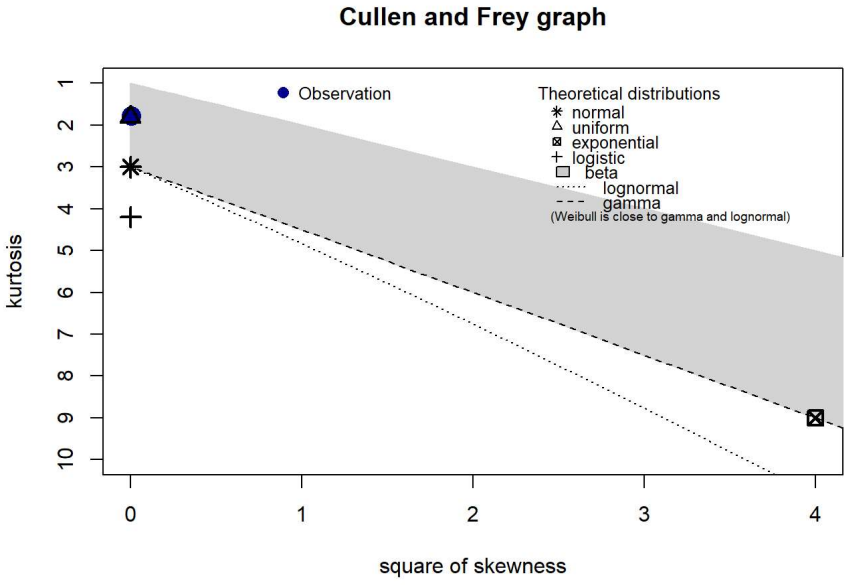
```
messages <- c()
dt.dst_d <- c()

for (day in days)
{
   num <- length(unique(df[weekdays(df$date) == day,]$from_id))
   dt.dst_d <- c(dt.dst_d, rep(which(days == day), num))
   messages <- c(messages, num)
}


dt.days = data.frame(weekday = days,  messages = messages)
```

```
ggplot(data=dt.days, aes(x=days, y=messages, alpha=messages)) +
    geom_bar(stat="identity", fill="#66a182") +
    labs(title="Message distribution for days") +
    geom_text(aes(label=messages), vjust=0.4, hjust= 1.1, size=10, color="white", alpha=1) +
    scale_x_discrete(limits=days)+
    coord_flip()
```



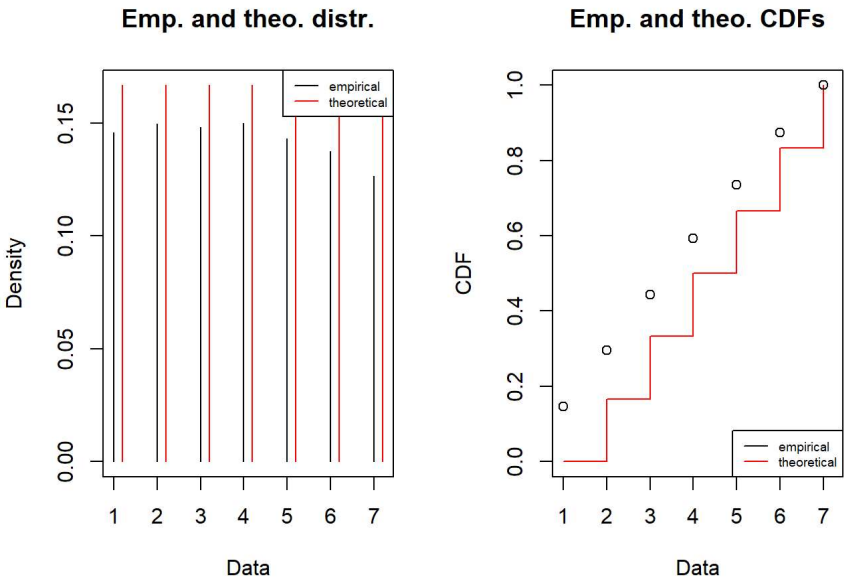
As we can say from the plot, distribution is likely to aproximately uniformal. And we also can say that most messages are concentrated near Thursday

```
descdist(dt.dst_d)
```




```
## summary statistics
## ------
## min:  1   max:  7
## median:  4
## mean:  3.912904
## estimated sd:  1.969823
## estimated skewness:  0.0511118
## estimated kurtosis:  1.785524
```

```
plot(fitdist(dt.dst_d, "unif", discrete = TRUE))
```



Now, we have two important distributions, but it is not still enough. As we want to find the best hour for the best day
It is obvious that we should calculate joint distribution. But befor doing that. Lets calculate conditional distributions for the best day and

```
best_hour <- dt.hours[which.max(dt.hours$messages),]$hours
best_day <- dt.days[which.max(dt.days$messages),]$weekday
bd_lbl <- sprintf("Message distribution for hours of all %s(s)", best_day)
bh_lbl <- sprintf("Message distribution for days at %d hour", best_hour)

sprintf("1. The most active day:  %s\n", best_day) %>% cat()
```

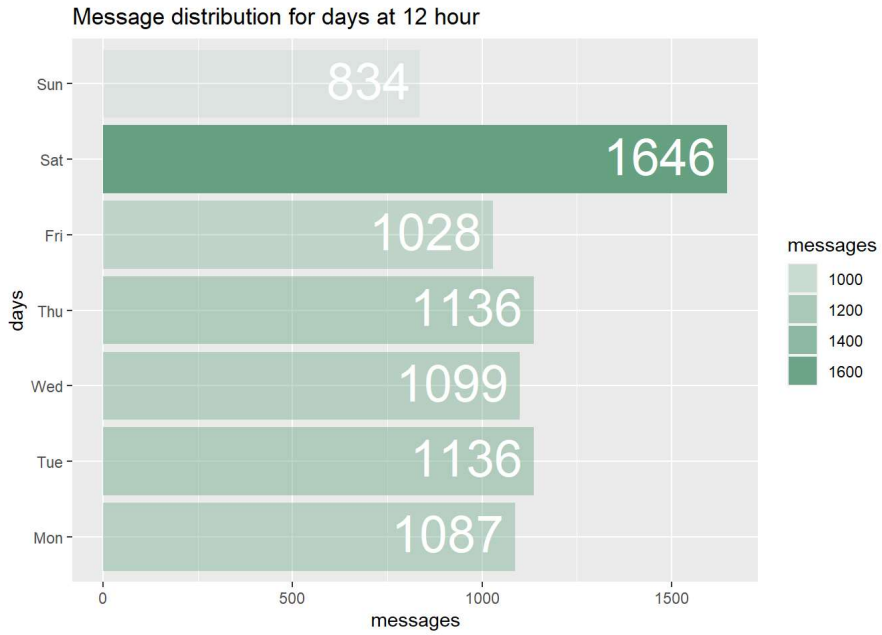
```
## 1. The most active day:  Thu
```

```
sprintf("2. The most active hour: %d\n", best_hour) %>% cat()
```

```
## 2. The most active hour: 12
```

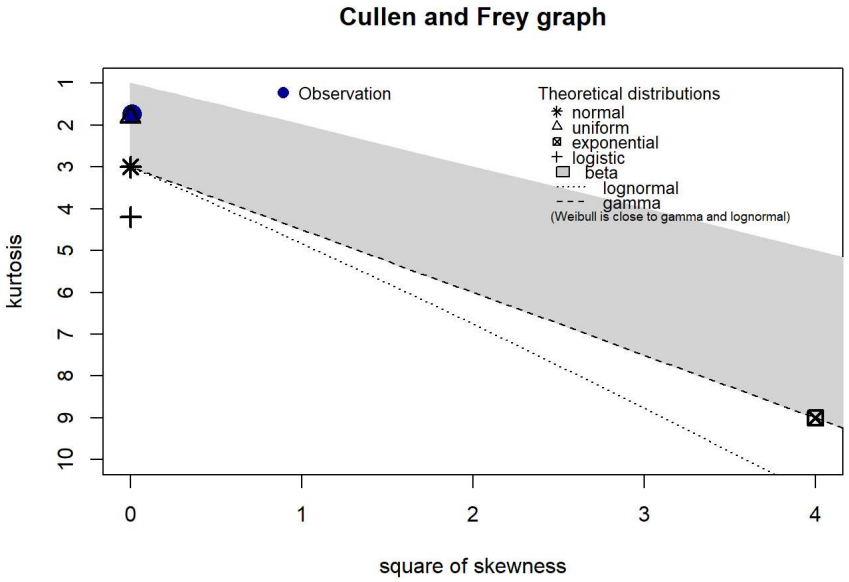
```
besth_df <- df[hours(df$date) == best_hour, ]
messages <- c()
dt.dst_dh <- c()

for (day in days)
{
   num <- length(unique(besth_df[weekdays(besth_df$date) == day,]$from_id))
   dt.dst_dh <- c(dt.dst_dh, rep(which(days == day), num))
   messages <- c(messages, num)
}


dt.best_hour = data.frame(weekday = days,  messages = messages)
```

```
ggplot(data=dt.best_hour, aes(x=days, y=messages, alpha=messages)) +
      geom_bar(stat="identity", fill="#66a182") +
      labs(title=bh_lbl) +
      geom_text(aes(label=messages), vjust=0.4, hjust= 1.1, size=10, color="white", alpha=1) +
      scale_x_discrete(limits=days)+
      coord_flip()
```



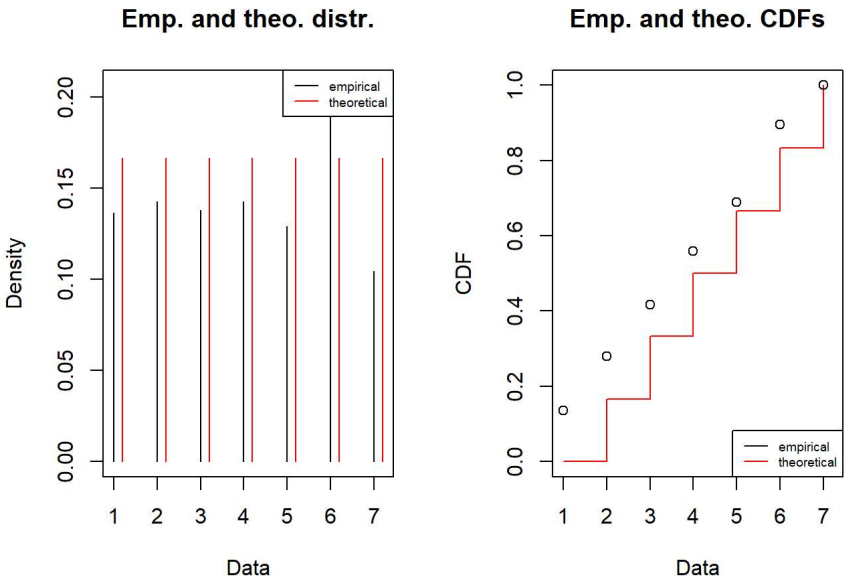
As we can say from the plot, some chances are that distribution is aproximately uniformal. Interesting thing is that the best day to choose for sending hiden advertisment at calculated best hour is different from calculated best day :)

```
descdist(dt.dst_dh)
```




```
## summary statistics
## ------
## min:  1   max:  7
## median:  4
## mean:  4.023851
## estimated sd:  1.958113
## estimated skewness:  -0.08375604
## estimated kurtosis:  1.730513
```

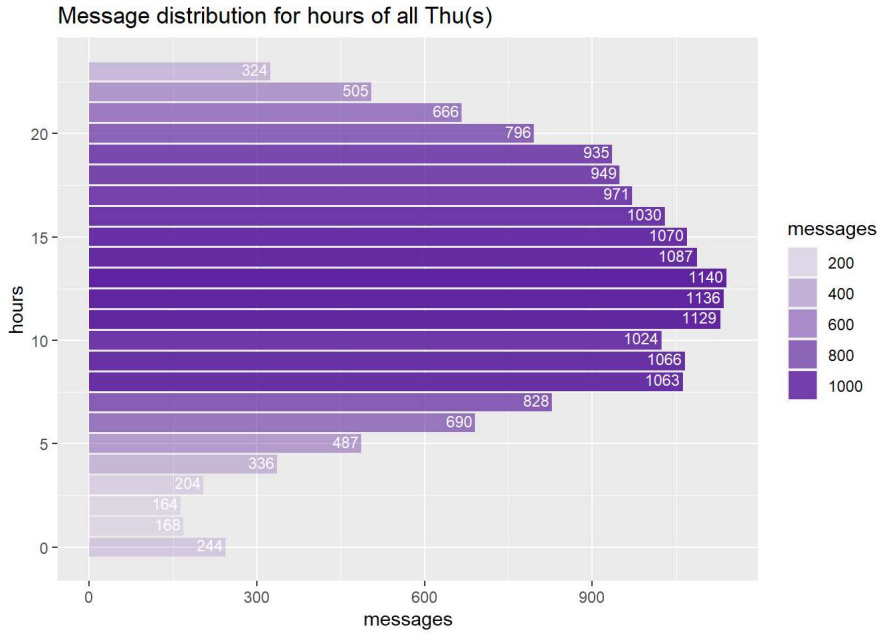
```
plot(fitdist(dt.dst_dh, "unif", discrete = TRUE))
```




```
bestd_df <- df[weekdays(df$date) == best_day,]
dt.dst_hd <- c()

messages <- c()
for (hour in hours)
{
   num <- length(unique(bestd_df[hours(bestd_df$date) == hour,]$from_id))
   dt.dst_hd <- c(dt.dst_hd, rep(hour, num))
   messages <- c(messages, num)
}


df.best_day = data.frame(hours = hours,  messages = messages)
```

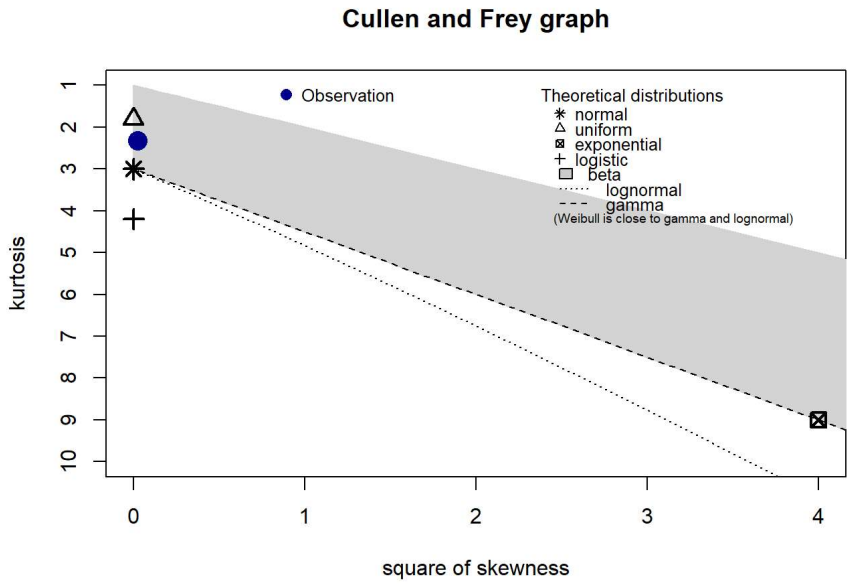
```
ggplot(data=df.best_day, aes(x=hours, y=messages, alpha=messages)) +
      geom_bar(stat="identity", fill="#5f249f") +
      labs(title=bd_lbl) +
      geom_text(aes(label=messages), vjust=0.3, hjust= 1.1, size=3, color="white", bold=TRUE, alpha=1) +
      coord_flip()
```

```
## Warning: Ignoring unknown parameters: bold
```



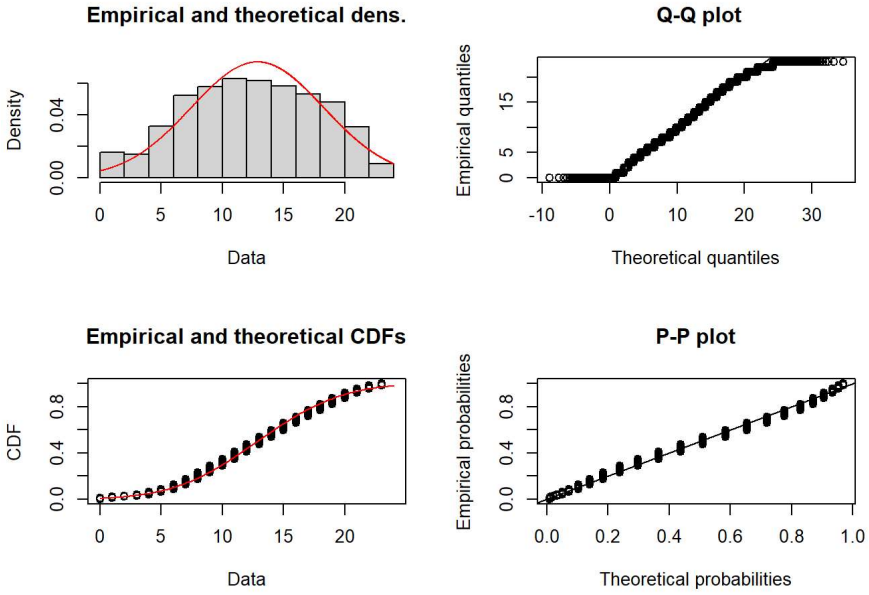
As we can say from the plot that distribution is aproximately normal. Interesting thing is that
the best hour to choose for sending hiden advertisment on calculated best day is different also from calculated best hour :)

```
descdist(dt.dst_hd)
```




```
## summary statistics
## ------
## min:  0   max:  23
## median:  13
## mean:  12.85821
## estimated sd:  5.415919
## estimated skewness:  -0.1559791
## estimated kurtosis:  2.316497
```

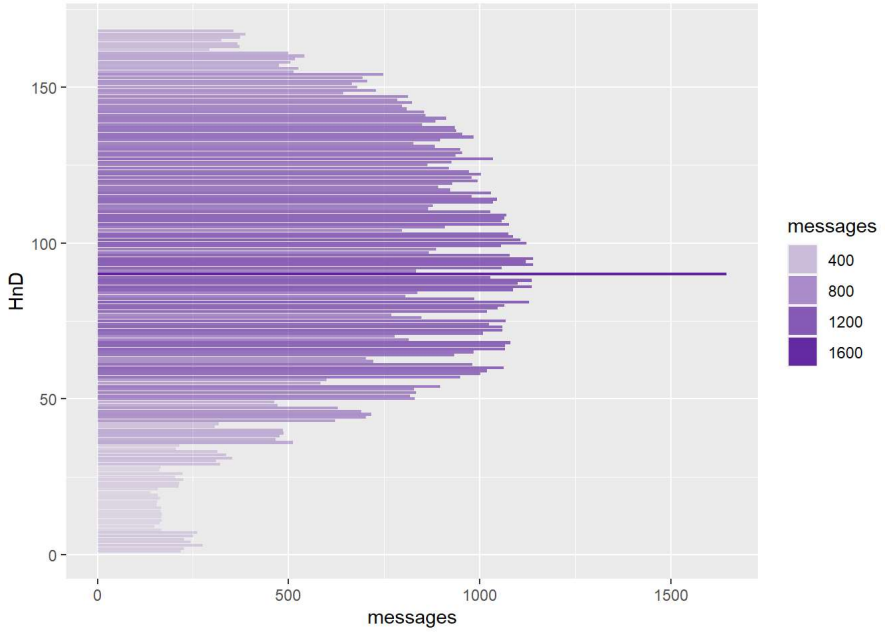
```
plot(fitdist(dt.dst_hd, "norm"))
```



Now, lest find joint distribution and find out, which of the condition distributions has given the correct result

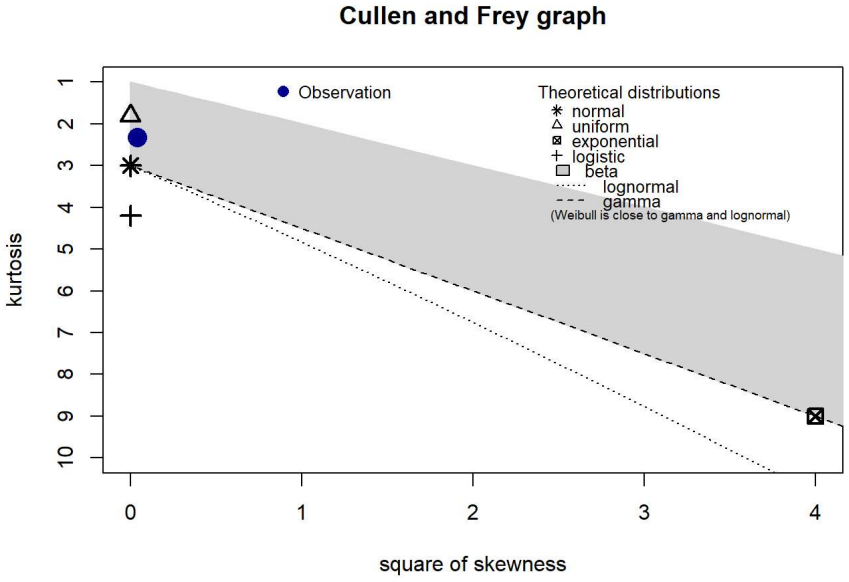
```
counter = 1
dt.dst_joint <- c()
messages <- c()

for (hour in hours)
{
   hdf <- df[hours(df$date) == hour,]
   for (day in days)
   {
     num <- length(unique(hdf[weekdays(hdf$date) == day,]$from_id))
     dt.dst_joint <- c(dt.dst_joint, rep(counter, num))
     messages <- c(messages, num)
     counter <- counter + 1
   }
}

dt.joint = data.frame(HnD = seq(1:(counter-1)),  messages = messages)
```

```
ggplot(data=dt.joint, aes(x=HnD, y=messages, alpha=messages)) +
    geom_bar(stat="identity", fill="#5f249f") +
    labs("Joint Message distribution for days and hours") +
    coord_flip()
```



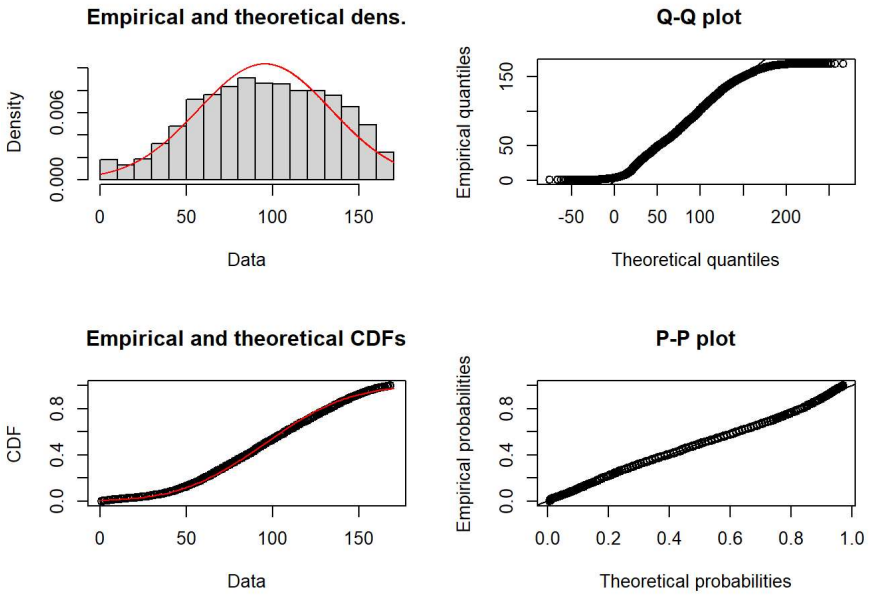
WOW, we can see that condtional distribution of messages given hour produces the same result !)

```
descdist(dt.dst_joint)
```




```
## summary statistics
## ------
## min:  1   max:  168
## median:  96
## mean:  95.28599
## estimated sd:  38.43282
## estimated skewness:  -0.1978727
## estimated kurtosis:  2.322357
```

```
plot(fitdist(dt.dst_joint, "norm"))
```

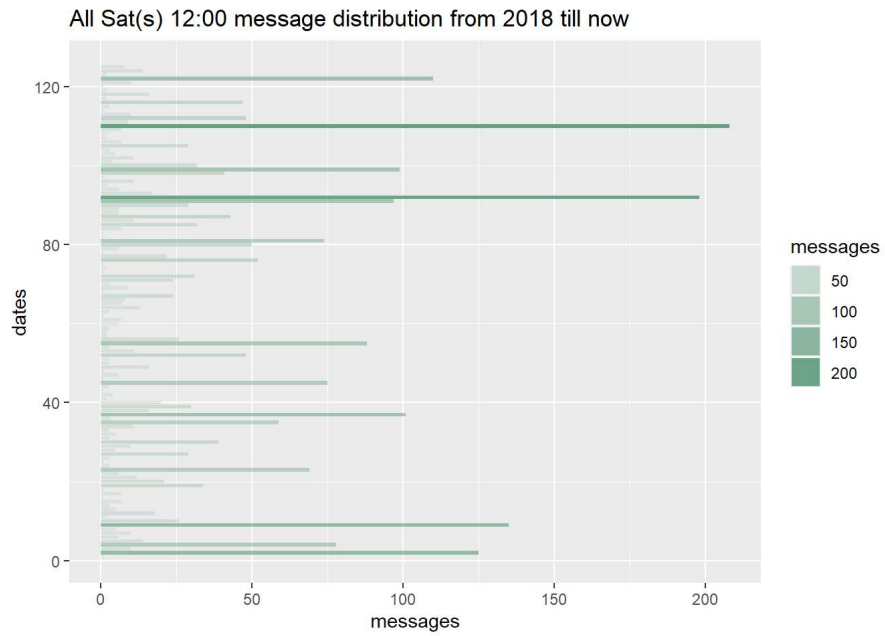


It is also interesting to find the distribution of message for all best days&hours for the whole period

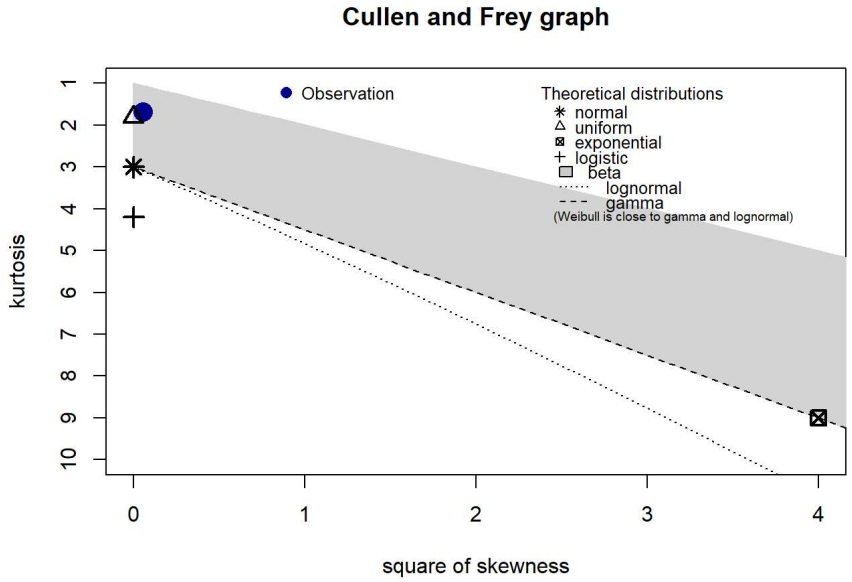
```
best_HnD <- dt.joint[which.max(dt.joint$messages),]$HnD
true_best_hour <- best_HnD %/% 7
true_best_day  <- days[best_HnD - 7*true_best_hour]

last <- df[weekdays(df$date) == true_best_day & hours(df$date) == hour,]
last$date <- as.Date(last$date)
u_dates <- unique(last$date)
```

```
counter = 1
dt.dst_best <- c()
messages <- c()


for (date in u_dates)
{
   num <- nrow(last[last$date == date, ])
   messages <- c(messages, num)
   dt.dst_best <- c(dt.dst_best, rep(counter, num))
   counter <- counter + 1
}


dt.best_dist = data.frame(dates = seq(1:(counter-1)),  messages = messages)
```

```
ggplot(data=dt.best_dist, aes(x=dates, y=messages, alpha=messages)) +
      geom_bar(stat="identity", fill="#66a182") +
      labs(title=sprintf("All %s(s) %d:00 message distribution from 2018 till now", true_best_day, true_best_hour)) +
      coord_flip()
```

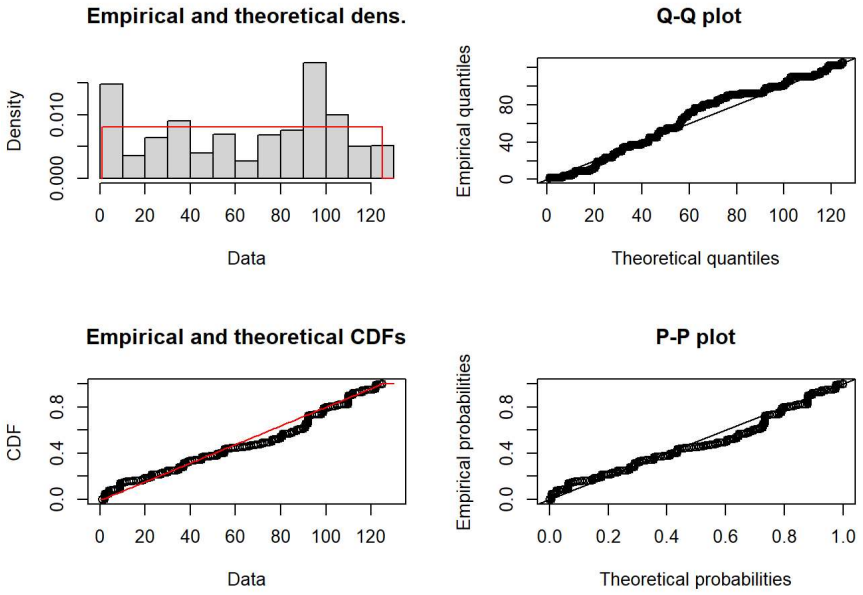


```
descdist(dt.dst_best)
```




```
## summary statistics
## ------
## min:  1   max:  125
## median:  76
## mean:  65.34508
## estimated sd:  38.49159
## estimated skewness:  -0.2359433
## estimated kurtosis:  1.68661
```

```
plot(fitdist(dt.dst_best, "unif"))
```



Hm, we can say that it tends to be uniformal. Nevertheless, it is scattered a lot

Now, we come to the interesting question. What is the reason for that high variance?
We think it is because of the COVID. So, our hypothesis would be H_0 :It is not because of COVID H_1 :It is because of COVID

To check that we would find distribution of two periods befor covid and after. The using f-test we would compare their variances.
If variance of the COVID period is differt from pevious period we would say that NULL hypothesis is rejected

```
df_covid = df[1: coronda_index,]
counter = 1
dt.dst_joint_corona <- c()
messages <- c()

for (hour in hours)
{
   hdf <- df_covid[hours(df_covid$date) == hour,]
   for (day in days)
   {
     num <- length(unique(hdf[weekdays(hdf$date) == day,]$from_id))
     dt.dst_joint_corona <- c(dt.dst_joint_corona, rep(counter, num))
     messages <- c(messages, num)
     counter <- counter + 1
   }
}

dt.joint_corona = data.frame(HnD = seq(1:(counter-1)),  messages = messages)
```

```
best_HnD_covid <- dt.joint_corona[which.max(dt.joint_corona$messages),]$HnD
true_best_hour_covid <- best_HnD_covid %/% 7
true_best_day_covid  <- days[best_HnD_covid - 7*true_best_hour_covid]

last_covid <- df_covid[weekdays(df_covid$date) == true_best_day_covid & hours(df_covid$date) == true_best_hour_covid,]
last_covid$date <- as.Date(last_covid$date)
u_dates_covid <- unique(last_covid$date)
```

```
counter = 1
dt.dst_best_covid <- c()
messages <- c()


for (date in u_dates_covid)
{
   num <- nrow(last_covid[last_covid$date == date, ])
   messages <- c(messages, num)
   dt.dst_best_covid <- c(dt.dst_best_covid, rep(counter, num))
   counter <- counter + 1
}


df.best_dist_covid = data.frame(dates = seq(1:(counter-1)),  messages = messages)
```

```
df_pre_covid = df[coronda_index: nrow(df),]
counter = 1
dt.dst_joint_pre_corona <- c()
messages <- c()

for (hour in hours)
{
   hdf <- df_pre_covid[hours(df_pre_covid$date) == hour,]
   for (day in days)
   {
     num <- length(unique(hdf[weekdays(hdf$date) == day,]$from_id))
     dt.dst_joint_pre_corona <- c(dt.dst_joint_pre_corona, rep(counter, num))
     messages <- c(messages, num)
     counter <- counter + 1
   }
}

dt.joint_pre_corona = data.frame(HnD = seq(1:(counter-1)),  messages = messages)
```

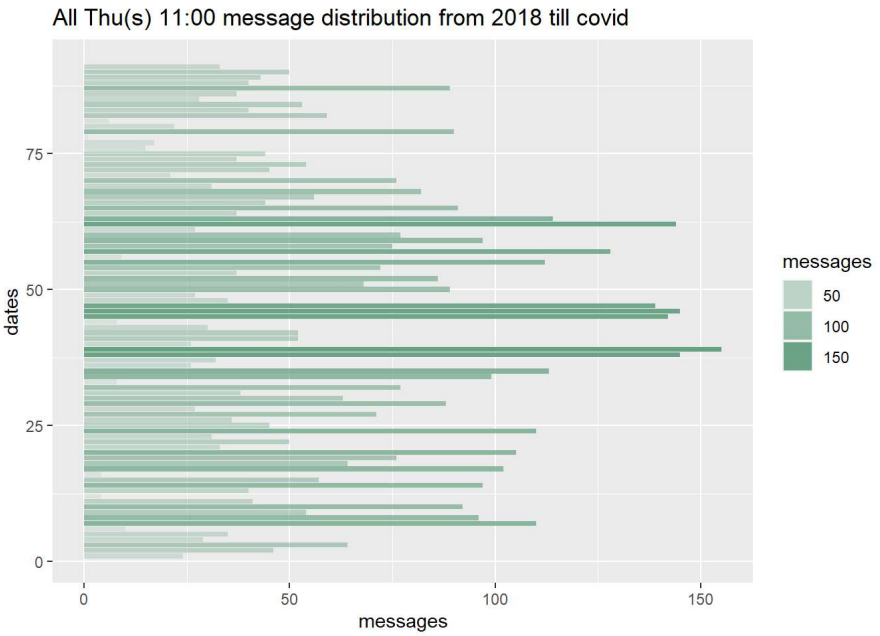
```
best_HnD_pre_covid <- dt.joint_pre_corona[which.max(dt.joint_pre_corona$messages),]$HnD
true_best_hour_pre_covid <- best_HnD_pre_covid %/% 7
true_best_day_pre_covid  <- days[best_HnD_pre_covid - 7*true_best_hour_pre_covid]

last_pre_covid <- df_pre_covid[weekdays(df_pre_covid$date) == true_best_day_pre_covid & hours(df_pre_covid$date) == true_best_hour_pre_covid,]
last_pre_covid$date <- as.Date(last_pre_covid$date)
u_dates_pre_covid <- unique(last_pre_covid$date)
```

```
counter = 1
dt.dst_best_pre_covid <- c()
messages <- c()


for (date in u_dates_pre_covid)
{
   num <- nrow(last_pre_covid[last_pre_covid$date == date, ])
   messages <- c(messages, num)
   dt.dst_best_pre_covid <- c(dt.dst_best_pre_covid, rep(counter, num))
   counter <- counter + 1
}


df.best_dist_pre_covid = data.frame(dates = seq(1:(counter-1)),  messages = messages)
```

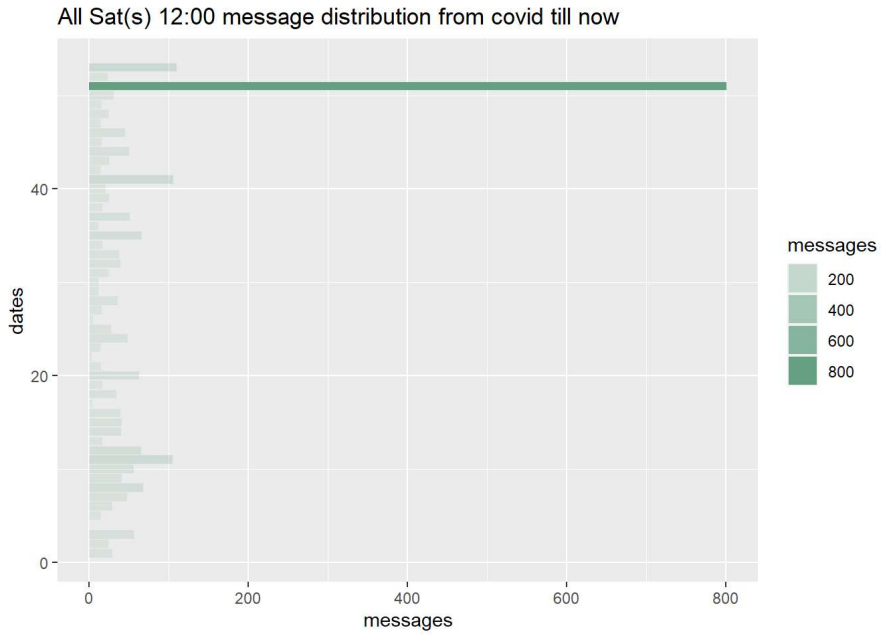
```
var.test(dt.dst_best_pre_covid, dt.dst_best_covid, alternative = "l")
```

```
##
##  F test to compare two variances
##
## data:  dt.dst_best_pre_covid and dt.dst_best_covid
## F = 1.8808, num df = 5428, denom df = 2566, p-value = 1
## alternative hypothesis: true ratio of variances is less than 1
## 95 percent confidence interval:
##  0.000000 1.987933
## sample estimates:
## ratio of variances
##           1.880769
```

```
ggplot(data=df.best_dist_pre_covid, aes(x=dates, y=messages, alpha=messages)) +
    geom_bar(stat="identity", fill="#66a182") +
    labs(title=sprintf("All %s(s) %d:00 message distribution from 2018 till covid", true_best_day_pre_covid, true_best_hour_pre_covid)) +
    coord_flip()
```




```
ggplot(data=df.best_dist_covid, aes(x=dates, y=messages, alpha=messages)) +
    geom_bar(stat="identity", fill="#66a182") +
    labs(title=sprintf("All %s(s) %d:00 message distribution from covid till now", true_best_day_covid, true_best_hour_covid)) +
    coord_flip()
```



Great! NULL hypothesis is rejected! So as we can see, during the COVID period programmers tends to spend most of their group activity on Sunday at 12:00. It may be caused by the complication of working at home. So they spend even weekend finding solutions or asking for help in this Python group.