Covid19- Bayesian Inference - Tamil Nadu

by krsmanian1972@gmail.com

In [1]:

```
# Importing the Required library components

%matplotlib inline
import pymc3 as pm
from IPython.core.pylabtools import figsize
from matplotlib import pyplot as plt
import scipy.stats as stats
import numpy as np
import theano.tensor as tt
import seaborn as sns
```

Number of People Affected Per Day in Tamil Nadu

I obtain the count of the number of people affected per day with COVID-19 from https://ta.wikipedia.org/wiki/2020_தமிழ்நாட்டில்_கொரோனாவைரசுத்_தொற்று (https://ta.wikipedia.org/wiki/2020_தமிழ்நாட்டில்_கொரோனாவைரசுத்_தொற்று)

The label for this count, hereafter, will be referred to as New Cases Per Day.

The Period of Data considered for the study is from 1st Apr to 17th Apr.

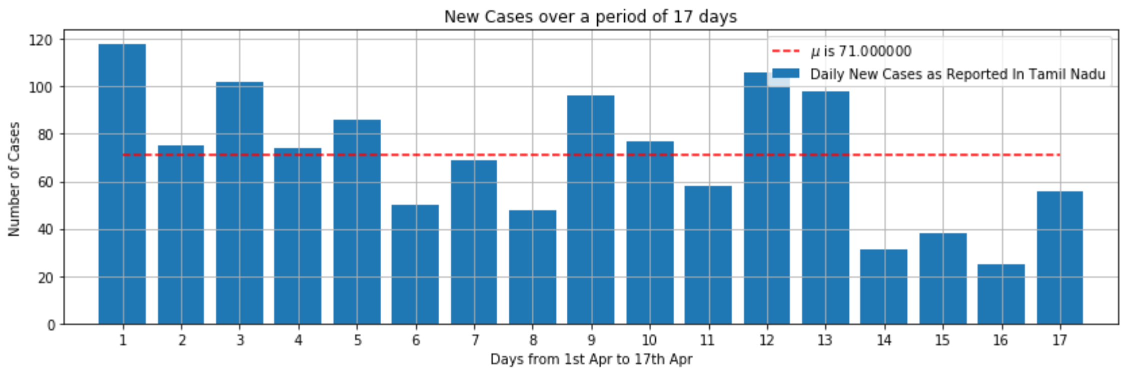
```
In [2]:
```

```python
# Representation of the Daily Cases

daily_new_cases = np.array([118,75,102,74,86,50,69,48,96,77,58
,106,98,31,38,25,56])
date_arr = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
])


total = daily_new_cases.sum()
total_days = len(daily_new_cases)
lambda0 = daily_new_cases.mean().round(0)
print("The Number of observed Days: {}".format(total_days))
print("The average count of New Cases Per Day: {}".format(lamb
da0))
print("The Total number of Reported Cases: {}".format(total))
```

```
The Number of observed Days: 17
The average count of New Cases Per Day: 71.0
The Total number of Reported Cases: 1207
```

In [3]:

```
# Visualization of the Daily Cases

figsize(14,4)
plt.bar(date_arr,daily_new_cases,label="Daily New Cases as Rep
orted In Tamil Nadu")

plt.title("New Cases over a period of {} days".format(total_da
ys))
plt.plot((1,total_days),(lambda0,lambda0),"r--",label="$\mu$ i
s %f" %lambda0)
plt.xlabel("Days from 1st Apr to 17th Apr")
plt.ylabel("Number of Cases")
plt.xlim(0,total_days+1)
plt.xticks(date_arr)

plt.legend()
plt.grid()
```



In [4]:

```
# Obtain posterior samples from two Exponential Distributions.

# Use the switch function to randomly collect lambdas before a
nd after a randomly selected date
# Random Ethana Random Sir!!!

# The idea is if there is no change then both the lambda distr
ibution will be nearly identical.
# If there is a change we will end-up in collecting two lambda
distributions

# We will get change-point (tau) also.

total_days = len(daily_new_cases)
with pm.Model() as cp_model:
    alpha = 1.0/daily_new_cases.mean()

    lambda_1 = pm.Exponential("lambda_1", alpha)
    lambda_2 = pm.Exponential("lambda_2", alpha)

    tau = pm.DiscreteUniform("tau",lower=0,upper=total_days-1)

    idx = np.arange(total_days)
    lambda_switch = pm.math.switch(tau>idx,lambda_1,lambda_2)

    obs_cp = pm.Poisson("obs_cp",lambda_switch,observed=daily_
new_cases)
```

In [5]:

```
# Do the simulation
# Watch for the selection of NUTS for the Continuous Distribut
ion (Exponential Distribution)
# Watch for the selection of Metropolis for Discrete Distribut
ion (Discrete Uniform)
with cp_model:
    trace_cp = pm.sample(10000,tune=5000)
```

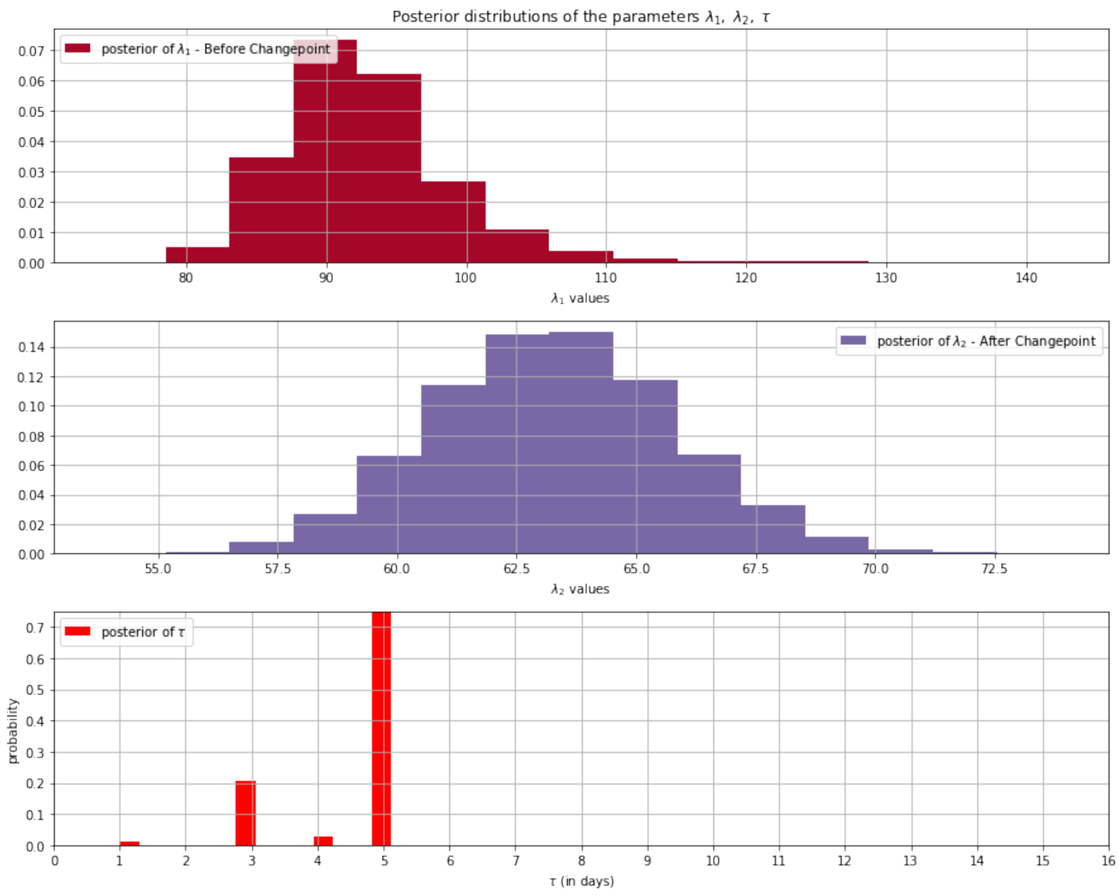
```
Multiprocess sampling (4 chains in 4 jobs)
CompoundStep
>NUTS: [lambda_2, lambda_1]
>Metropolis: [tau]
Sampling 4 chains, 0 divergences: 100%|██████████|
60000/60000 [00:09<00:00, 6115.72draws/s]
The acceptance probability does not match the targ
et. It is 0.8793015660504‌1, but should be close to
0.8. Try to increase the number of tuning steps.
The number of effective samples is smaller than 10
% for some parameters.
```

In [6]:

```
# Time to collect the samples
lambda_1_samples = trace_cp['lambda_1']
lambda_2_samples = trace_cp['lambda_2']
tau_samples = trace_cp['tau']
```

In [7]:

```
# Plot the Posterior Parameters lambda_1, lambda_2 and Tau as
Histogram of Probability Density
# The Changepoint is tau is also a probability distribution.
# Look for potential changepoints

figsize(12.5, 10)

ax=plt.subplot(311)
plt.title(r"""Posterior distributions of the parameters $\lamb
da_1,\;\lambda_2,\;\tau$""")
plt.hist(lambda_1_samples, bins=15,label="posterior of $\lambd
a_1$ - Before Changepoint", color="#A60628", density=True)
plt.legend(loc="upper left")
plt.xlabel("$\lambda_1$ values")
plt.grid()

ax = plt.subplot(312)
plt.hist(lambda_2_samples, bins=15, label="posterior of $\lamb
da_2$ - After Changepoint", color="#7A68A6", density=True)
plt.legend(loc="upper right")
plt.xlabel("$\lambda_2$ values")
plt.grid()

ax=plt.subplot(313)
w = 1.0 / tau_samples.shape[0] * np.ones_like(tau_samples)
plt.hist(tau_samples, bins=total_days, label=r"posterior of $\
tau$",color="r", weights=w )
plt.xticks(np.arange(total_days))
plt.legend(loc="upper left")
plt.ylim([0, .75])
plt.xlabel(r"$\tau$ (in days)")
plt.ylabel("probability");

plt.grid()
plt.tight_layout()
```



In [8]:

```
change_point_index = 5
old_cases = daily_new_cases[0:change_point_index]
new_cases = daily_new_cases[change_point_index:]
new_dates = date_arr[5:]
```

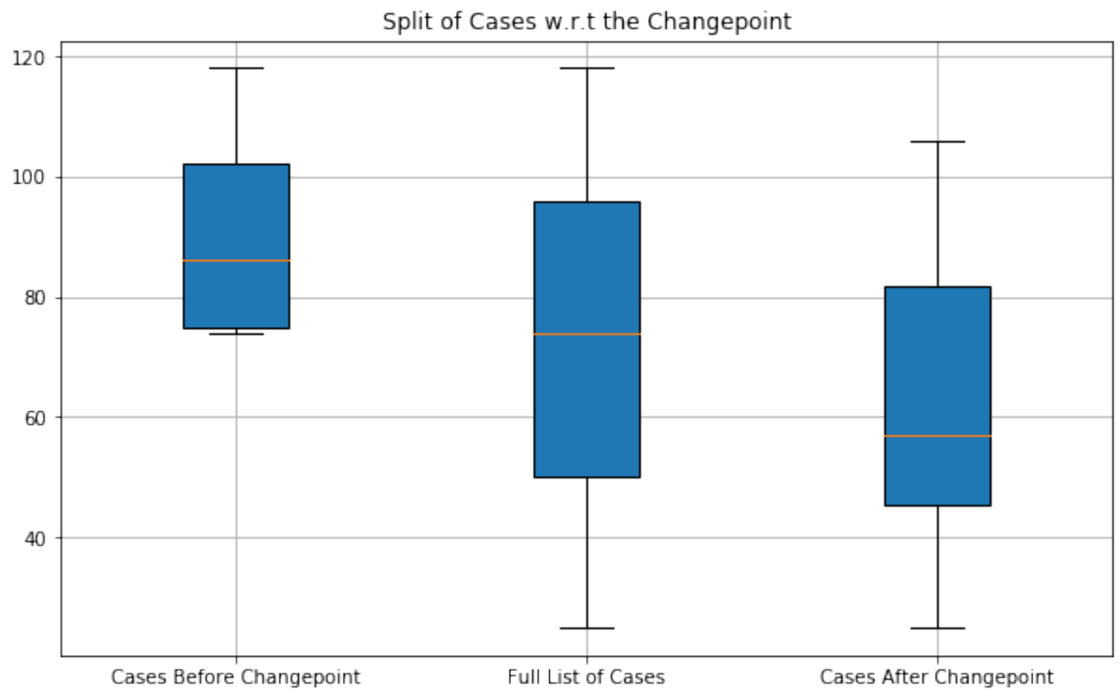
In [9]:

```
size = len(new_cases)
mean = new_cases.mean()

print("Number of New Cases {}".format(size))
print("Number of Date Array {}".format(len(new_dates)))
print("Average Number of New Cases {}".format(mean))
```

```
Number of New Cases 12
Number of Date Array 12
Average Number of New Cases 62.666666666666664
```

In [10]:

```
fig, ax = plt.subplots(figsize=(10,6))
plt.title("Split of Cases w.r.t the Changepoint")
ax.boxplot([old_cases,daily_new_cases,new_cases],patch_artist=
True);
ax.set_xticklabels(['Cases Before Changepoint', 'Full List of
Cases', 'Cases After Changepoint'])
plt.grid()
```



In [11]:

```
with pm.Model() as simple_model:
    alpha = 1.0/new_cases.mean()
    lambda_1 = pm.Exponential("lambda_1", alpha)
    obs_simple = pm.Poisson("obs_simple",lambda_1,observed=new
_cases)
```

In [12]:

```
with simple_model:
    trace_simple = pm.sample(4000,tune=1500)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [lambda_1]
Sampling 4 chains, 0 divergences: 100%|██████████|
22000/22000 [00:02<00:00, 7468.38draws/s]
```

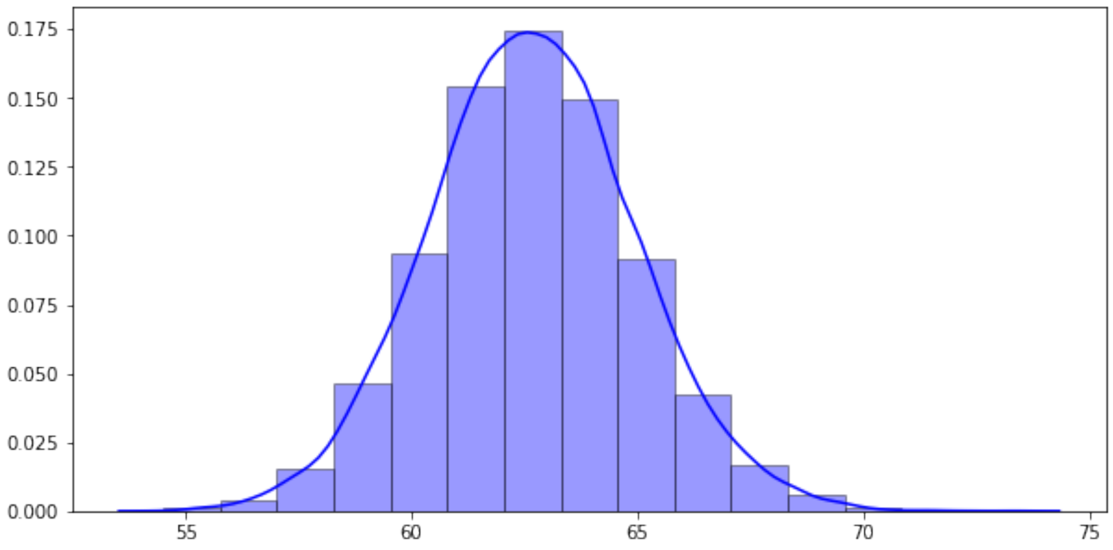
In [13]:

```
lambda_1_samples = trace_simple['lambda_1']
```

In [14]:

```
fig, ax = plt.subplots(figsize=(10,5))
sns.distplot(lambda_1_samples, hist=True, kde=True, bins=15, c
olor = 'blue',hist_kws={'edgecolor':'black'})
print("Posterior Lambdas collected from Expoential Distributio
n for the Poisson Distribution")
```

```
Posterior Lambdas collected from Expoential Distri
bution for the Poisson Distribution
```



In [15]:

```
pm.summary(trace_simple)
```

Out[15]:

	mean	sd	hpd_3%	hpd_97%	mcse_mean	mcse_sd
lambda_1	62.668	2.281	58.36	66.954	0.027	0.019

Inference

Now we have the posterior parameters for the Poisson Distributed Daily New Cases after the Switch Point.

So, we are ready to answer certain questions using the lambdas that we have collected.

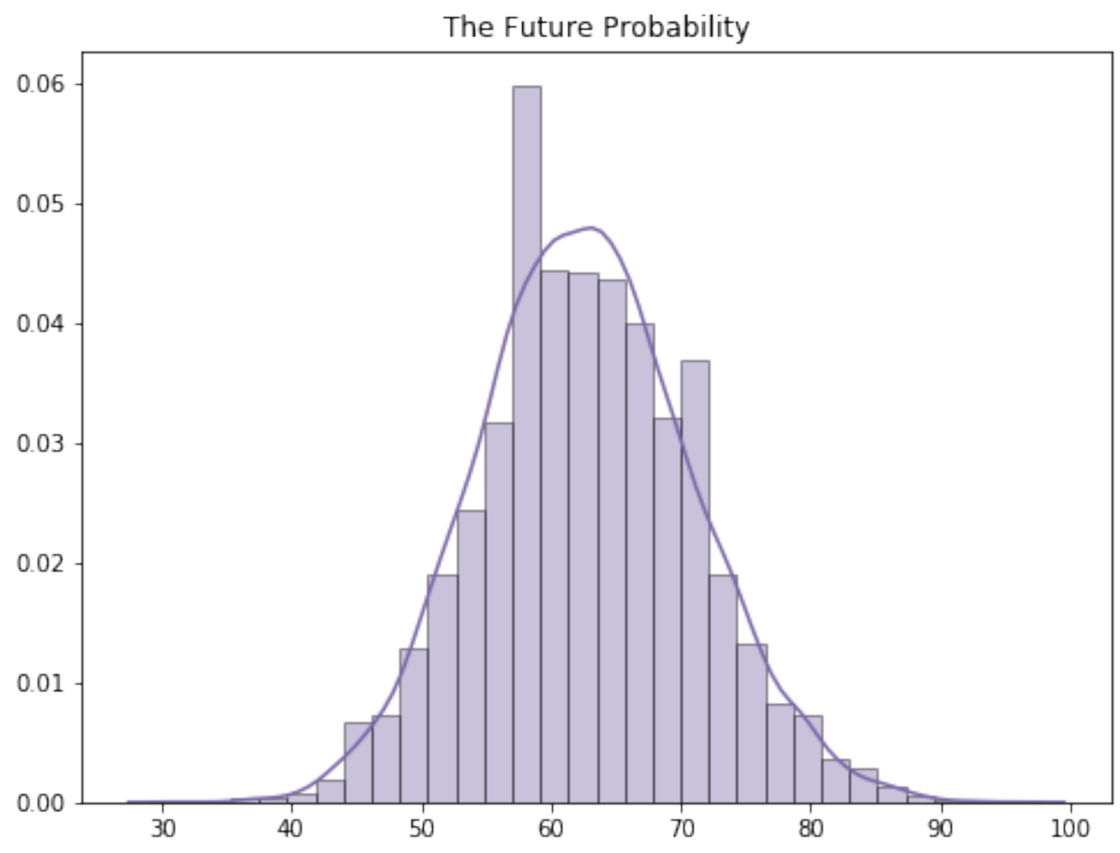
Q: What is the expected number of new cases given the current context?

The bayesian answer will be again a distribution as captured below.

In [19]:

```
future = np.random.poisson(lam=trace_simple['lambda_1'])
fig, ax = plt.subplots(figsize=(8, 6))
sns.distplot(future, color="#7A68A7", bins=30, ax=ax,hist_kws=
{'edgecolor':'black'})
plt.title("The Future Probability")
print("The Probability of daily new new cases")
```

```
The Probability of daily new new cases
```



```
In [23]:
```

```python
def how_many(future,start,end):
    count = 0
    for x in future:
        if(x>= start and x<= end):
            count+=1
    return count/len(future)


print("Probability of daily new cases between 60 and 70 : {}".
format(how_many(future,60,70)))
print("Probability of daily new cases between 60 and 80 : {}".
format(how_many(future,60,80)))
print("Probability of daily new cases between 60 and 90 : {}".
format(how_many(future,60,90)))

print("Probability of daily new cases between 71 and 80 : {}".
format(how_many(future,71,80)))
print("Probability of daily new cases between 91 and 100 : {}"
.format(how_many(future,91,100)))

less_than_60 = (future < 60).astype(int).mean()
print("Probability of daily new cases less than 60 : {}".forma
t(less_than_60))
```

```
Probability of daily new cases between 60 and 70 :
0.4721875
Probability of daily new cases between 60 and 80 :
0.6249375
Probability of daily new cases between 60 and 90 :
0.64275
Probability of daily new cases between 71 and 80 :
0.15275
Probability of daily new cases between 91 and 100
: 0.000375
Probability of daily new cases less than 60 : 0.35
6875
```

We have Good News today for Tamil Nadu

The rate of new cases being identified daily in, Tamil Nadu, is reducing. It is a very good sign.

Millions of thanks to the Police, to the Healthcare team, to our Government and every warriors involved in this combat against COVID.

```
Probability of daily new cases between 60 and 70 : 0.47
21875  Approx 47%

Probability of daily new cases between 60 and 80 : 0.62
49375

Probability of daily new cases between 60 and 90 : 0.64
275

Probability of daily new cases between 71 and 80 : 0.15
275

Probability of daily new cases between 91 and 100 : 0.0
00375

Probability of daily new cases less than 60 : 0.356875
Approx 35%
```

Special thanks to the people of Tamil Nadu.

However continue to maintain the Social Distancing because it is the only option to further reduce the rate of new cases.

எதிரதாக் காக்கும் அறிவினார்க் கில்லை அதிர வருவதோர் நோய்.

No terrifying calamity will affect the wise, who foresee and guard against evils.

Thanks

Note

For the details of my approach, please read my COVID19-Bayesian Inference-India.pdf available at https://github.com/krsmanian1972/bayes/blob/master/COVID19-Bayesian-Inference-India.pdf (https://github.com/krsmanian1972/bayes/blob/master/COVID19-Bayesian-Inference-India.pdf)

```
In [ ]:
```