MODELING CA WILDFIRES w/POISSON PROCESS

Joe Soria

Dr. Korosteleva

STAT 482

BACKGROUND

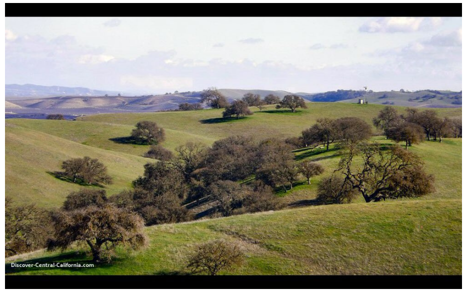
Chapparal

Joshua Tree
Woodlands

Coastal Bluffs

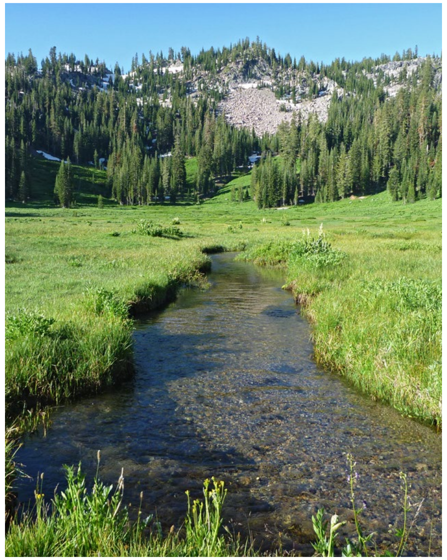
Coniferous Forests

Grasslands



Oak Savannas

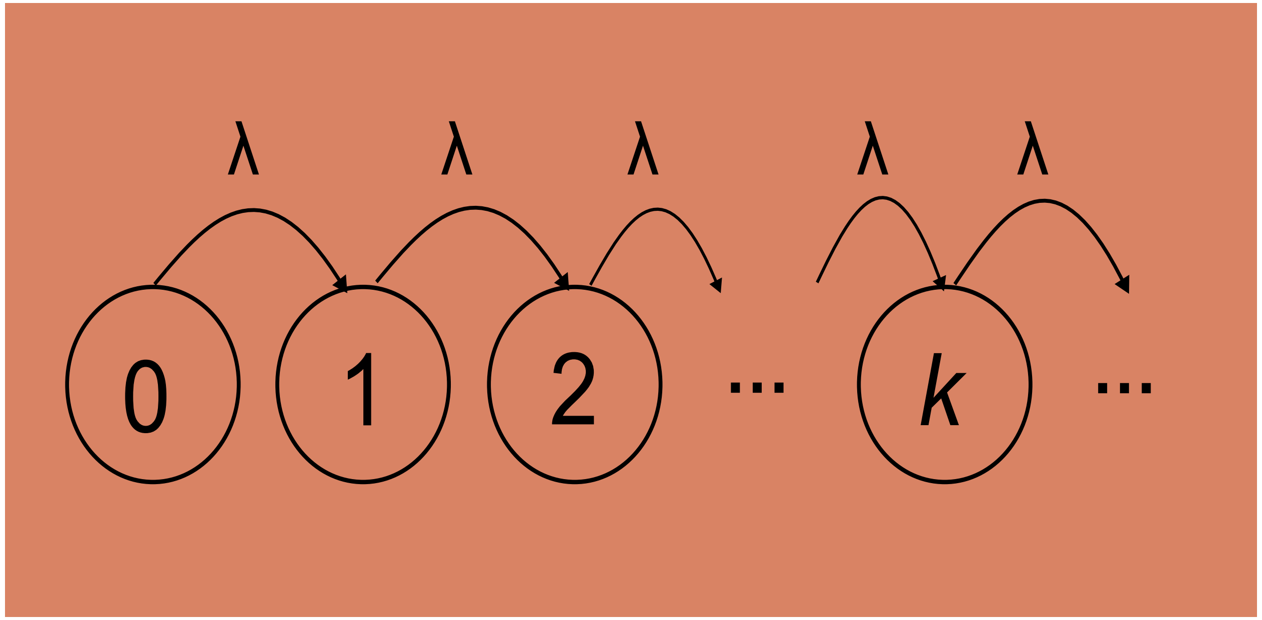
Sloughs



Wet Meadows

Wildfires

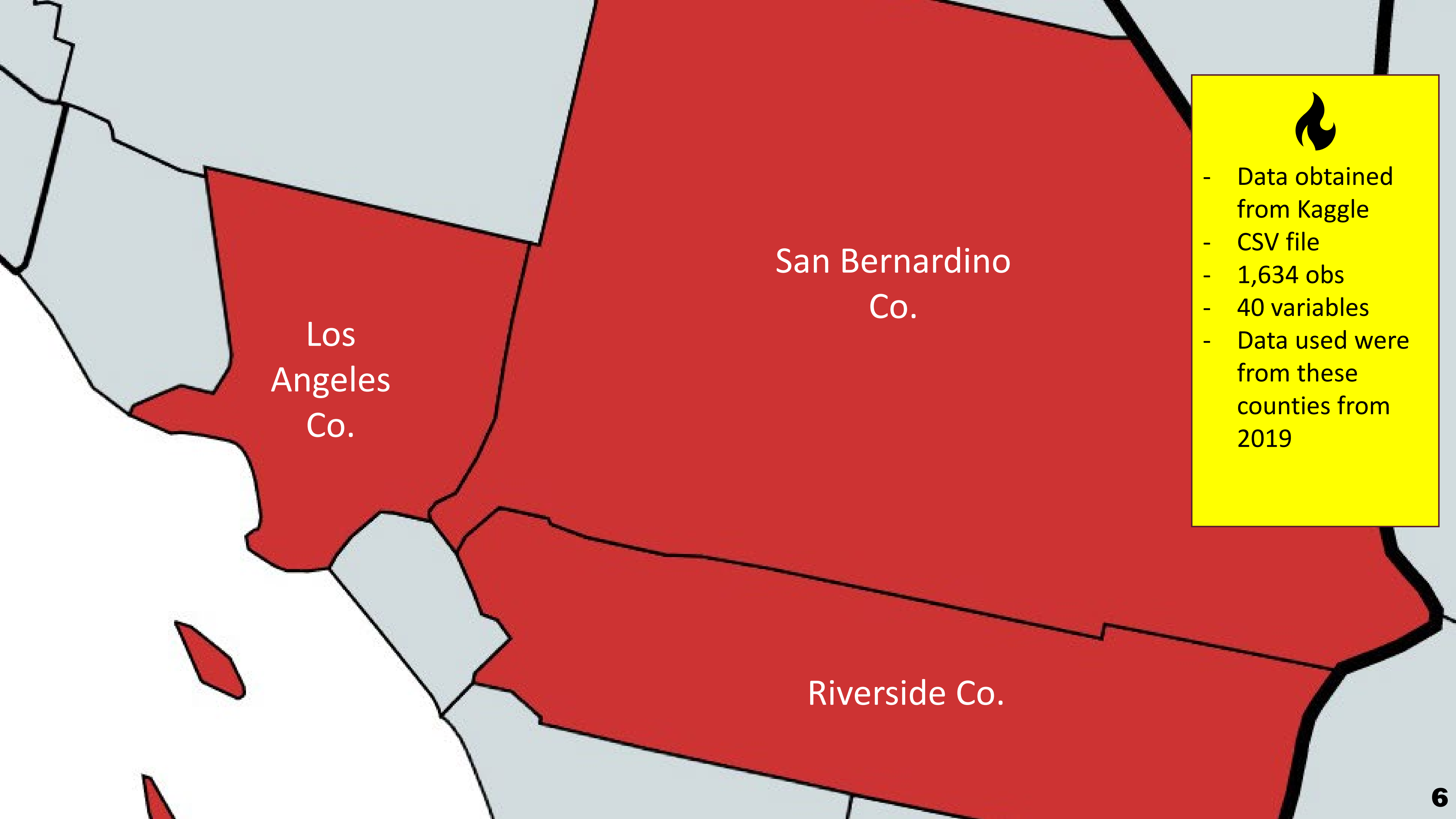
- Wildfires are a natural process in the ecosystem, especially in the Western United States
- Fires remove low-growing underbrush, cleans the floor of debris, opens it up to sunlight, and nourishes the soil
- Provides habitat making way for new generations of plants, some of which require fire to germinate (sprout)
- Fire kills diseases and insects that prey on trees and provides valuable nutrients that enrich the soil
- This is all good, however, the frequency with which wildfires are occurring now negate all these benefits



Poisson Process

- A stochastic process {N(t), t ≧ 0} is called a counting process if N(t) gives the total number of events occurring by time t
- A counting process {N(t), t ≧ 0} is called a Poisson process with rate λ, if:
  - (i) no events occur at time 0, i.e., N(0) = 0
  - (ii) it has independent increments,
  - (iii) it has stationary increments
  - (iv) P(N(t) = n) = $\frac{(\lambda t)^n}{n!} e^{-\lambda t}$

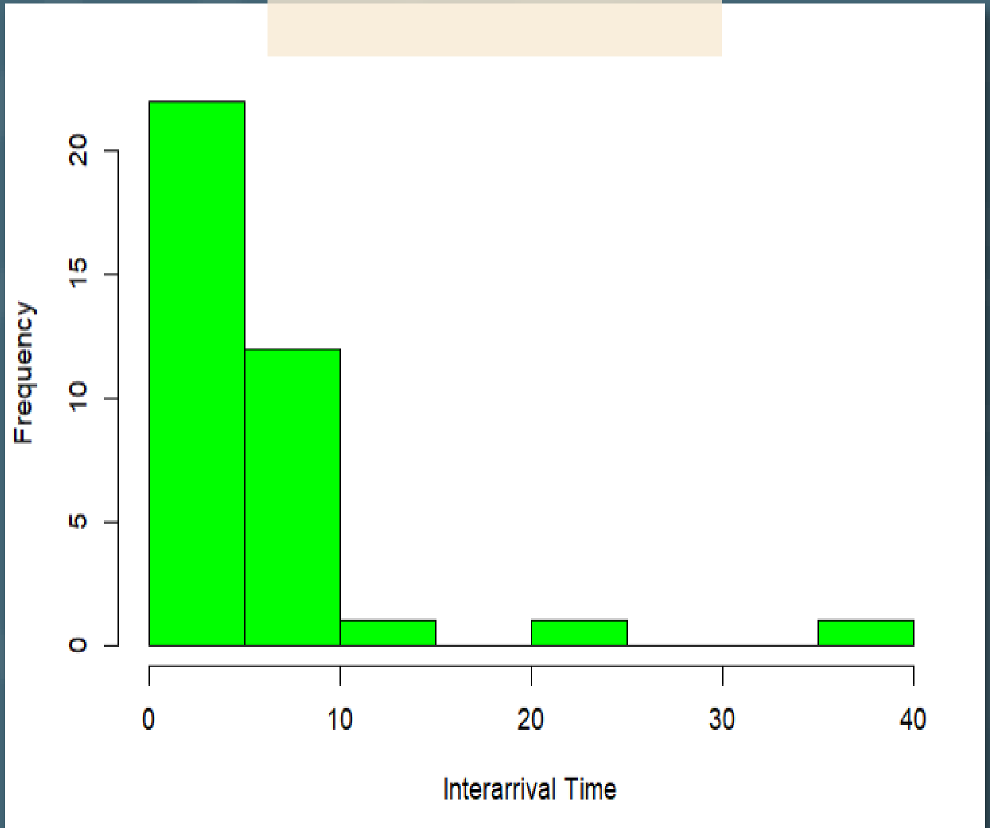
the DATA



- Data obtained from Kaggle
- CSV file
- 1,634 obs
- 40 variables
- Data used were from these counties from 2019

5	5/29/2019	San Bernardino
6	5/30/2019	Riverside
7	6/21/2019	Los Angeles
8	6/27/2019	Los Angeles
9	6/29/2019	Riverside
10	7/11/2019	Los Angeles
11	7/12/2019	Riverside
12	7/13/2019	Riverside
13	7/16/2019	Riverside
14	7/22/2019	Riverside
15	7/28/2019	Riverside
16	8/3/2019	Riverside
17	8/5/2019	San Bernardino
18	8/12/2019	Riverside
19	8/13/2019	Los Angeles
20	8/22/2019	San Bernardino
21	8/25/2019	San Bernardino
22	8/27/2019	Riverside
23	9/2/2019	Los Angeles

Excel data used in modeling after filtering and cleaning the original



HISTOGRAM

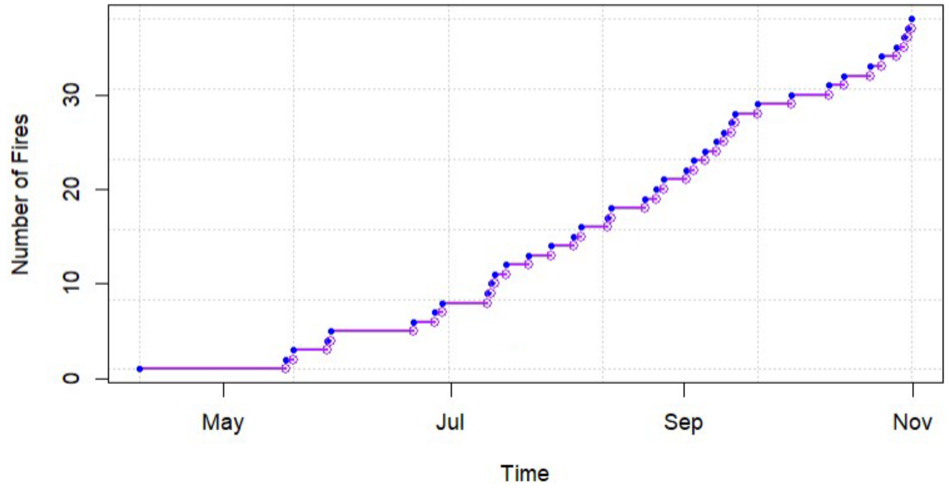
```
incidents <- read.csv(file = "…", header = TRUE, sep = ",")

# creating date-time variable
datetime<- as.POSIXct(as.Date(incidents$Date, "%m/%d/%Y"))
datetime

# computing lag
datetime.lag<- c(0,head(datetime, -1))

# computing inter arrival times (in hours) and removing 1st val
int.time<- (as.numeric(datetime)-as.numeric(datetime.lag))/(3600*24)
int<- int.time[-1]

# plotting histogram
hist(int, main="", col="green", xlab="Interarrival Time")

#binning inter arrival times
binned.int<- as.factor(ifelse(int<5,"1",
   ifelse(int>=5 & int<10,"2",ifelse(int>=10 & int<15,"3","4"))))

#computing observed frequencies
obs<- table(binned.int)
obs

#estimating mean for exponential distribution
mean.est<- mean(int)

#computing expected frequencies
exp<- c(1:4)
exp[1]<- length(int)*(1-exp(-5/mean.est))
exp[2]<- length(int)*(exp(-5/mean.est)-exp(-10/mean.est))
exp[3]<- length(int)*(exp(-10/mean.est)-exp(-15/mean.est))
exp[4]<- length(int)*exp(-15/mean.est)
round(exp,1)

#computing chi-squared statistic
print(chi.sq<- sum((obs-exp)^2/exp))

#computing p-value
print(p.value<- 1-pchisq(chi.sq, df=2))
```

```
> obs
binned.int
 1  2  3  4
22 11  2  2
> round(exp,1)
[1] 21.9  8.9  3.6  2.5
> #computing chi-squared statistic
> print(chi.sq<- sum((obs-exp)^2/exp))
[1] 1.317218
> #computing p-value
> print(p.value<- 1-pchisq(chi.sq, df=2))
[1] 0.5175706
```

Goodness-of-Fit Test

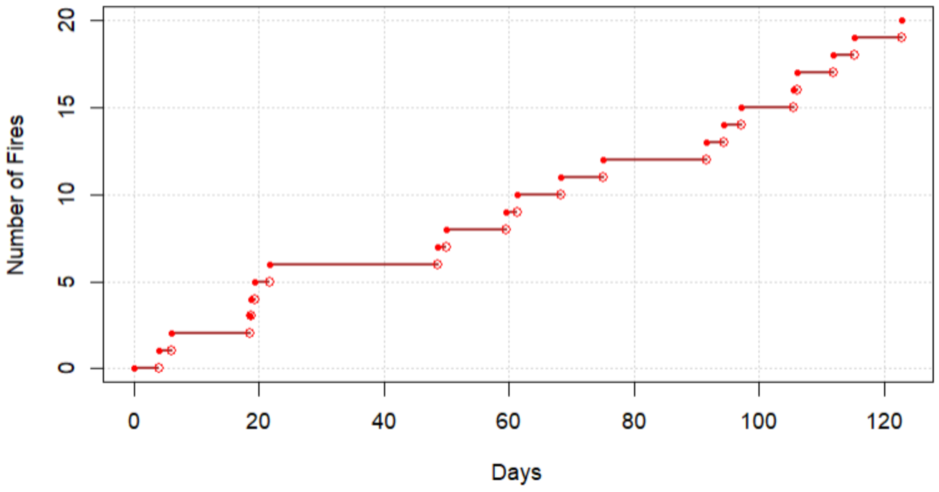
The p-value is larger than 0.05, indicating that the wildfires in the given time frame occurred according to a Poisson process.

```
incidents <- read.csv(file = "…", header = TRUE, sep = ",")

# creating date-time variable
datetime<- as.POSIXct(as.Date(incidents$Date, "%m/%d/%Y"))
datetime

# computing lag
datetime.lag<- c(0,head(datetime, -1))

# computing inter arrival times (in hours) and removing 1st val
int.time<- (as.numeric(datetime)-as.numeric(datetime.lag))/(3600*24)
int<- int.time[-1]

# plotting histogram
hist(int, main="", col="green", xlab="Interarrival Time")

#binning inter arrival times
binned.int<- as.factor(ifelse(int<5,"1",
   ifelse(int>=5 & int<10,"2",ifelse(int>=10 & int<15,"3","4"))))

#computing observed frequencies
obs<- table(binned.int)
obs

#estimating mean for exponential distribution
mean.est<- mean(int)

#computing expected frequencies
exp<- c(1:4)
exp[1]<- length(int)*(1-exp(-5/mean.est))
exp[2]<- length(int)*(exp(-5/mean.est)-exp(-10/mean.est))
exp[3]<- length(int)*(exp(-10/mean.est)-exp(-15/mean.est))
exp[4]<- length(int)*exp(-15/mean.est)
round(exp,1)

#computing chi-squared statistic
print(chi.sq<- sum((obs-exp)^2/exp))

#computing p-value
print(p.value<- 1-pchisq(chi.sq, df=2))
```



Plotting Actual Trajectory

```
fires <- c(1:38)
date<- as.POSIXct(as.Date(incidents$Date,
"%m/%d/%Y"))

#plotting stock price against date
plot(date, fires, type="n",
     xlab="Time", ylab="Number of Fires",
first.panel=grid())


segments(date[-length(date)], fires[-
length(date)], date[-1]-0.07, fires[-
length(date)],
         lwd=2, col="purple")
points(date, fires, pch=20, col="blue")
points(date[-1], fires[-length(date)],
pch=1, col="purple")
```



Plotting Simulated Trajectory

```
# creating date-time variable
datetime<- as.POSIXct(as.Date(incidents$Date, "%m/%d/%Y"))
datetime

# computing lag
datetime.lag<- c(0,head(datetime, -1))

# computing inter arrival times (in hours) and removing 1st val
int.time<- (as.numeric(datetime)-as.numeric(datetime.lag))/(3600*24)
int<- int.time[-1]

#estimating mean for exponential distribution
mean.est<- mean(int)

lambda.est <- 1 / mean.est
t<- 10
nfires <- 20

#defining states
N<- 0:nfires

#setting time as vector
time<- c()

#setting initial value for time
time[1]<- 0

#specifying seed
set.seed(483650)

#simulating trajectory
for (i in 2:(nfires+1))
  time[i]<- time[i-1]+round((-1/lambda)*log(runif(1)),2)

#plotting trajectory
plot(time, N, type="n", xlab="Days", ylab="Number of Fires", panel.first =
grid())
segments(time[-length(time)], N[-length(time)], time[-1]-0.07, N[-length(time)],
         lwd=2, col="brown")
points(time, N, pch=20, col="red")
points(time[-1], N[-length(time)], pch=1, col="red")
```

FURTHER RESULTS

```
datetime<-
as.POSIXct(as.Date(incidents$Date,
"%m/%d/%Y"))

datetime.lag<- c(0,head(datetime, -1))

int.time<- (as.numeric(datetime)-
as.numeric(datetime.lag))/(3600*24)
int<- int.time[-1]

mean.est<- mean(int)
```

```
> mean.est
[1] 5.567568
```

Wait Between Fires (days)

```
datetime<-
as.POSIXct(as.Date(incidents$Date,
"%m/%d/%Y"))

datetime.lag<- c(0,head(datetime, -1))

int.time<- (as.numeric(datetime)-
as.numeric(datetime.lag))/(3600*24)
int<- int.time[-1]

mean.est<- mean(int)

lambda.est <- 1/mean.est
```

```
> lambda.est
[1] 0.1796117
```

Estimated Lambda (fires per day)

Date	Mean	Next Fire
11/1/2019	5.56	11/6/2019

Next Fire


THANK YOU!