Mandha_663_Exercise_10

Harinath Reddy

2022-11-14

```
library(IDDA)
data(state.long)
View(state.long)
```

(a)

```
library(dplyr)
```

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

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

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

```
library(fable)
```

```
## Loading required package: fabletools
```

```
library(tsibble)
```

```
##
## Attaching package: 'tsibble'
```

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

```
library(ggplot2)
library(feasts)
```

```
state.ts <- as_tsibble(state.long, key = State)
```

```
## Using `DATE` as index variable.
```

```
state.ts <- as_tsibble(state.long, key = State) %>%
group_by(State) %>%
mutate(Infected = Infected/1000) %>%
mutate(YDA_Infected = lag(Infected, order_by = DATE)) %>%
mutate(YDA_Death = lag(Death, order_by = DATE)) %>%
mutate(Y.Infected = Infected - YDA_Infected) %>%
mutate(Y.Death = Death - YDA_Death) %>%
mutate(cum_infected = cumsum(Infected))%>%
mutate(cum_death = cumsum(Death)) %>%
```

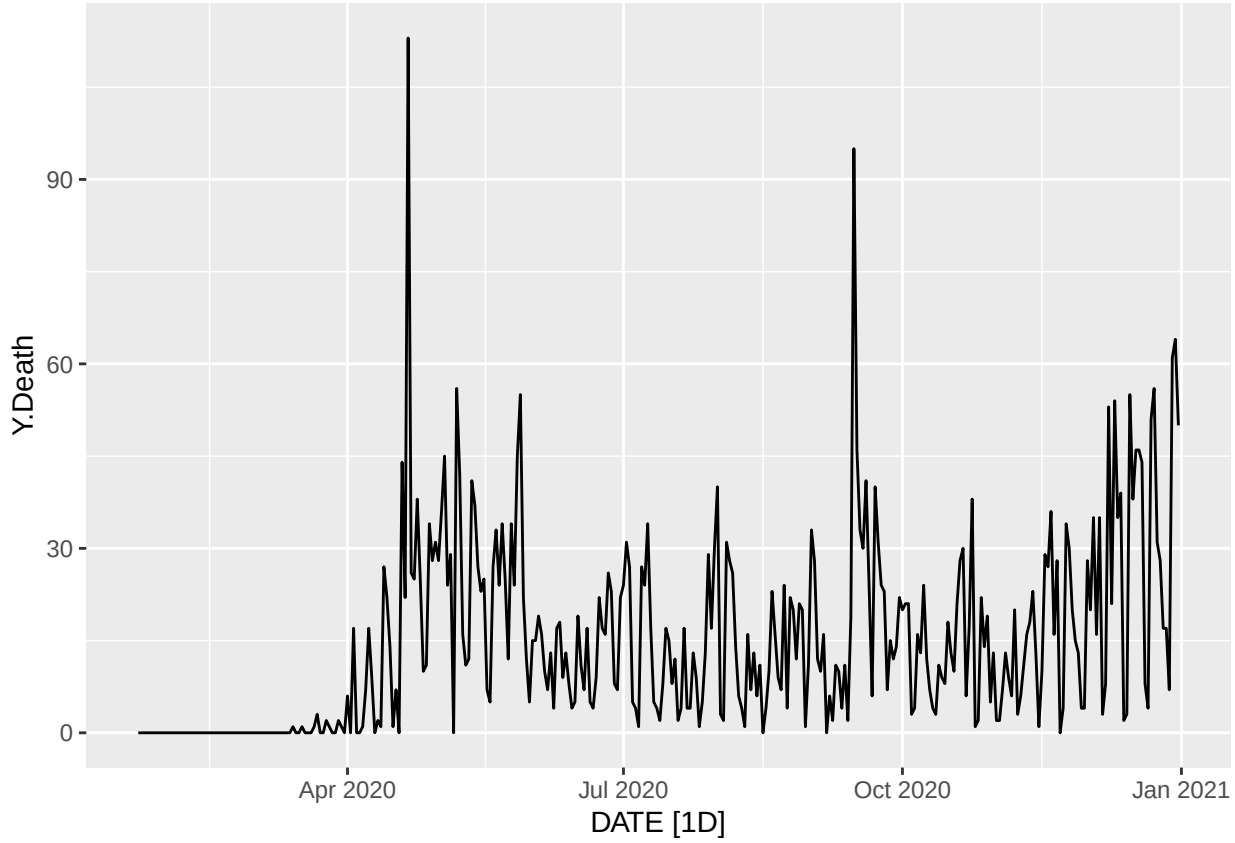
```
dplyr::filter(!is.na(Y.Infected)) %>%
dplyr::filter(!is.na(Y.Death)) %>%
dplyr::select(-c(YDA_Infected, YDA_Death))%>%
filter(State=="Virginia")
```

```
## Using `DATE` as index variable.
```

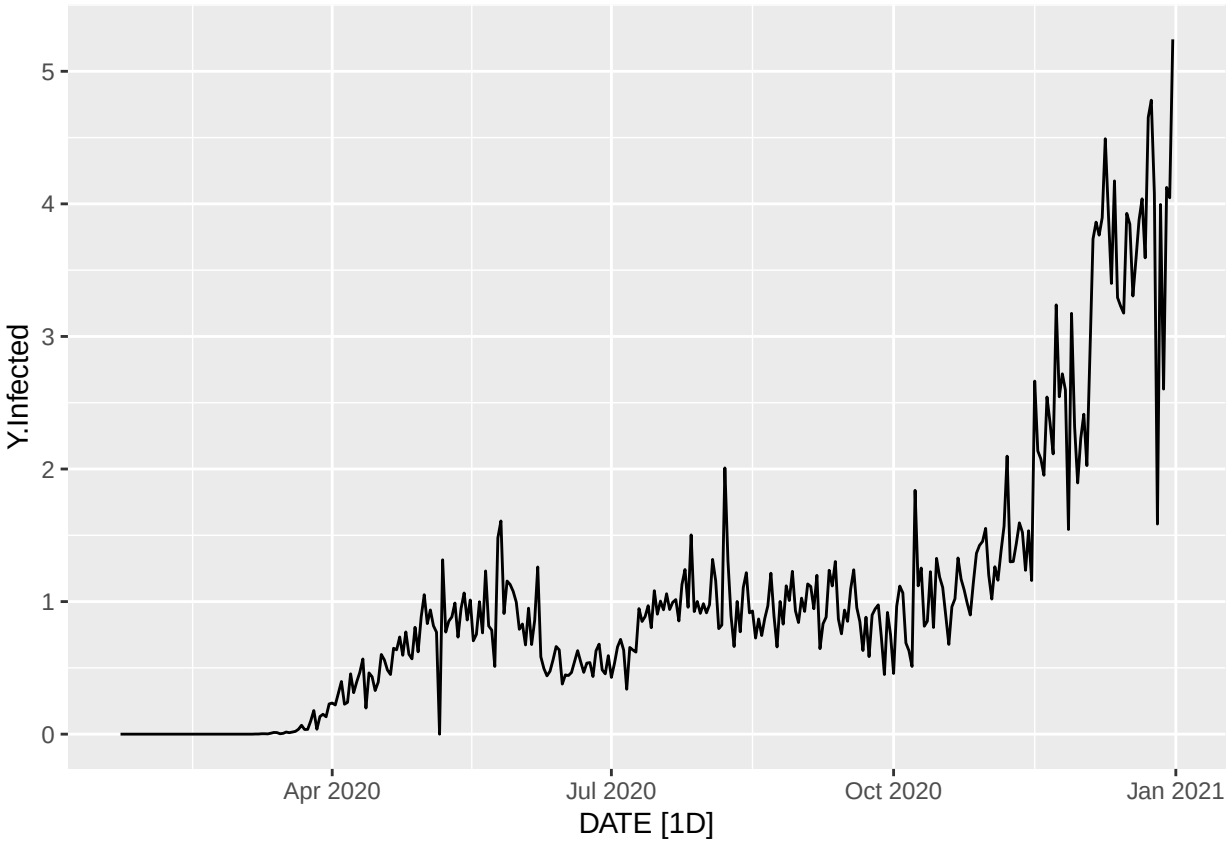
```
Virginia.ts <- state.ts %>%
dplyr::filter(State == "Virginia") %>%
dplyr::select(Infected, Death, cum_infected, cum_death, Y.Death, Y.Infected)
```

```
## Adding missing grouping variables: `State`
```

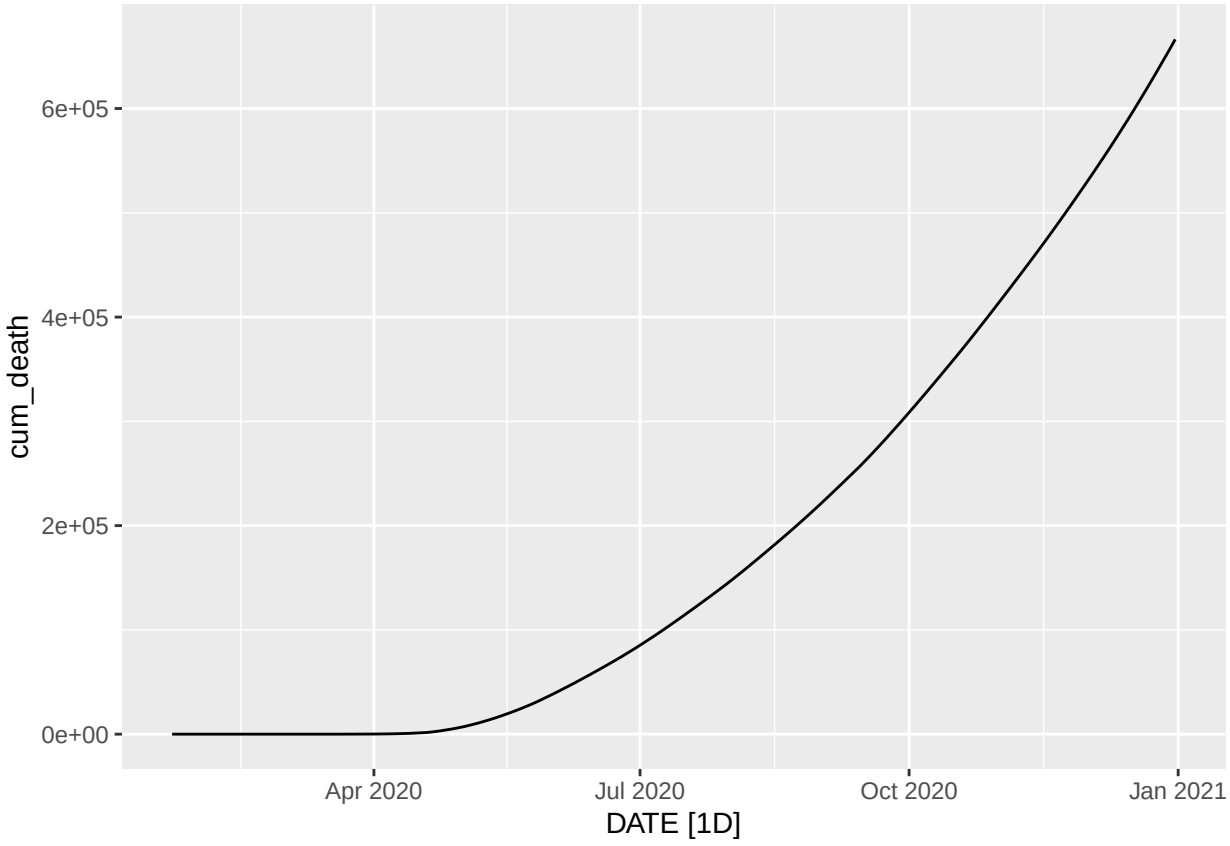
```
ts.death<- Virginia.ts%>% autoplot(Y.Death)
ts.infected<-Virginia.ts%>% autoplot(Y.Infected)
ts.cumdeath<-Virginia.ts%>% autoplot(cum_death)
ts.cuminfected<-Virginia.ts%>% autoplot(cum_infected)
ts.death
```



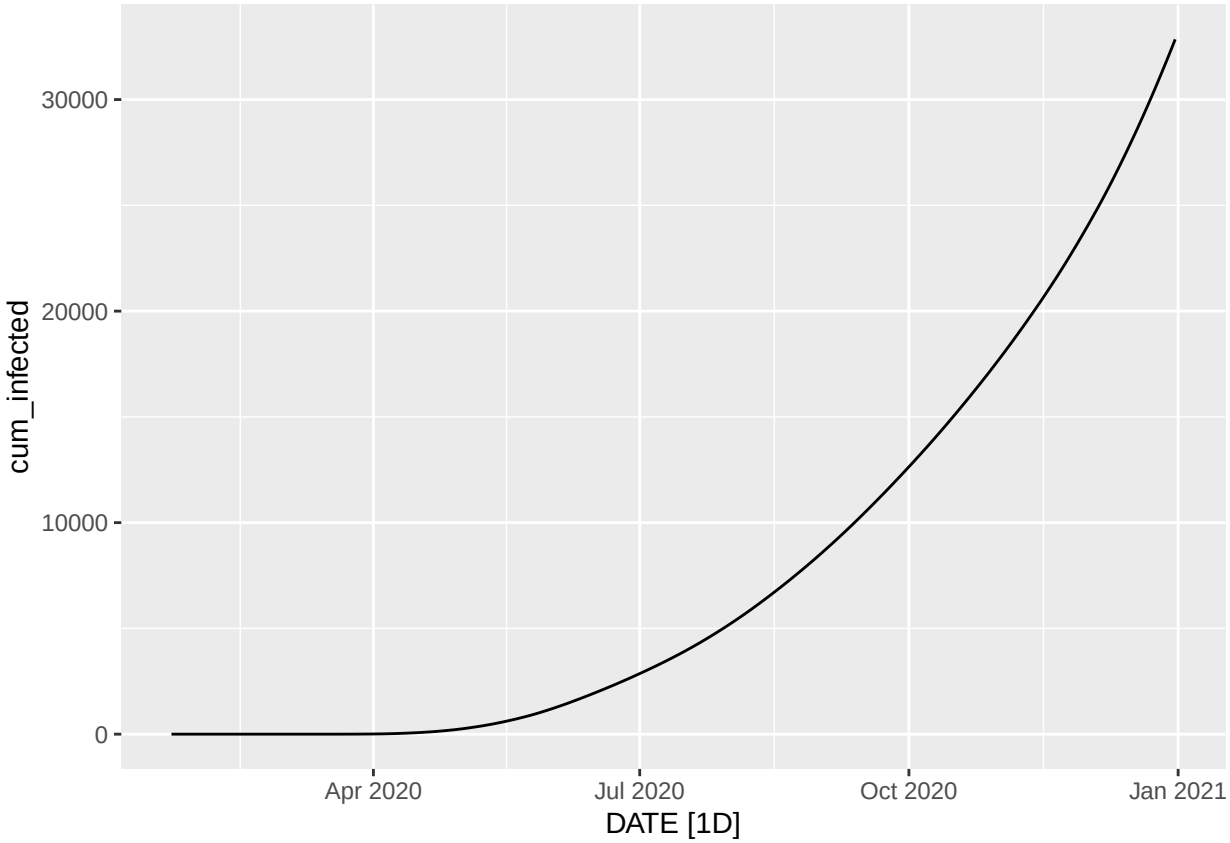

```
ts.infected
```




```
ts.cumdeath
```

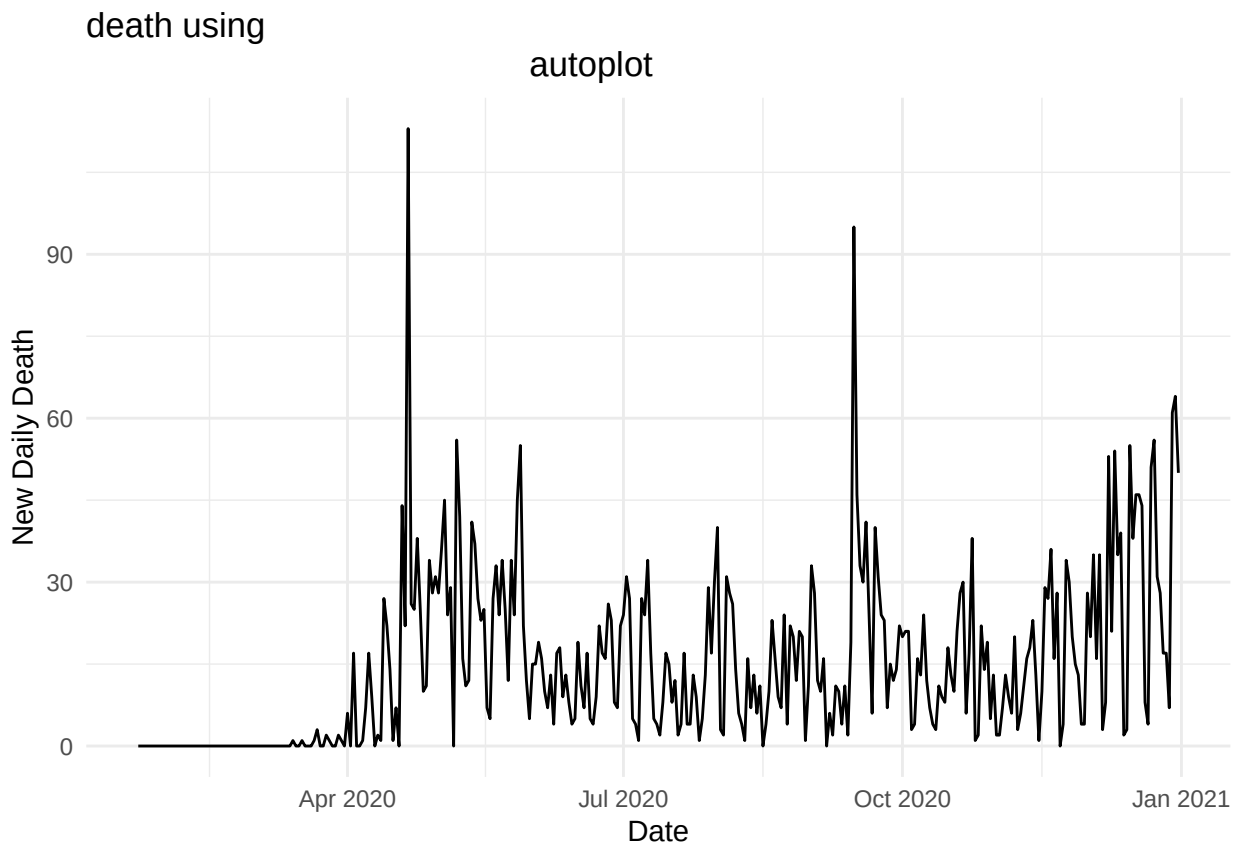



```
ts.cuminfected
```



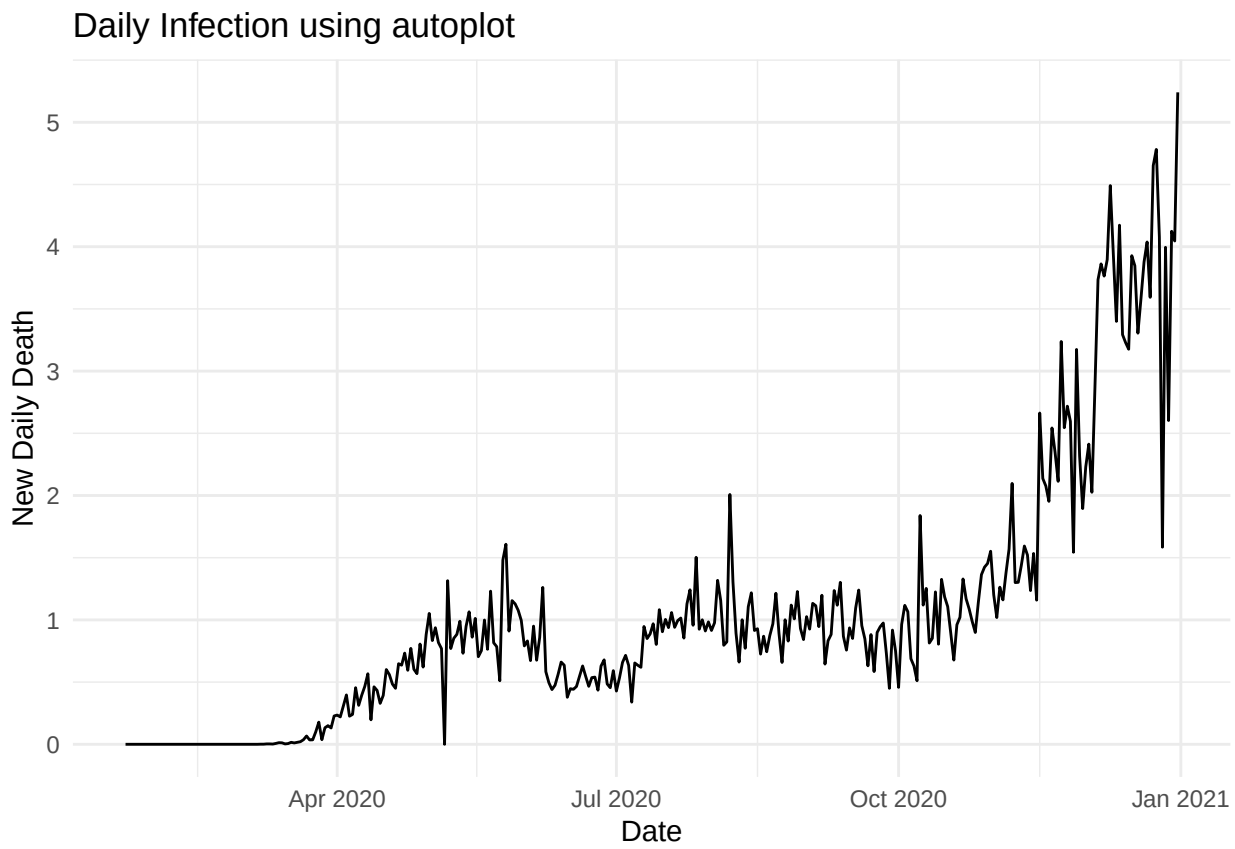
#b

```
### autoplot for daily death
autoplot_daily_death <- state.ts %>%
  autoplot(Y.Death) + theme_minimal() + labs(title="death using
                                             autoplot",x="Date",
                                             y="New Daily Death")

autoplot_daily_death
```



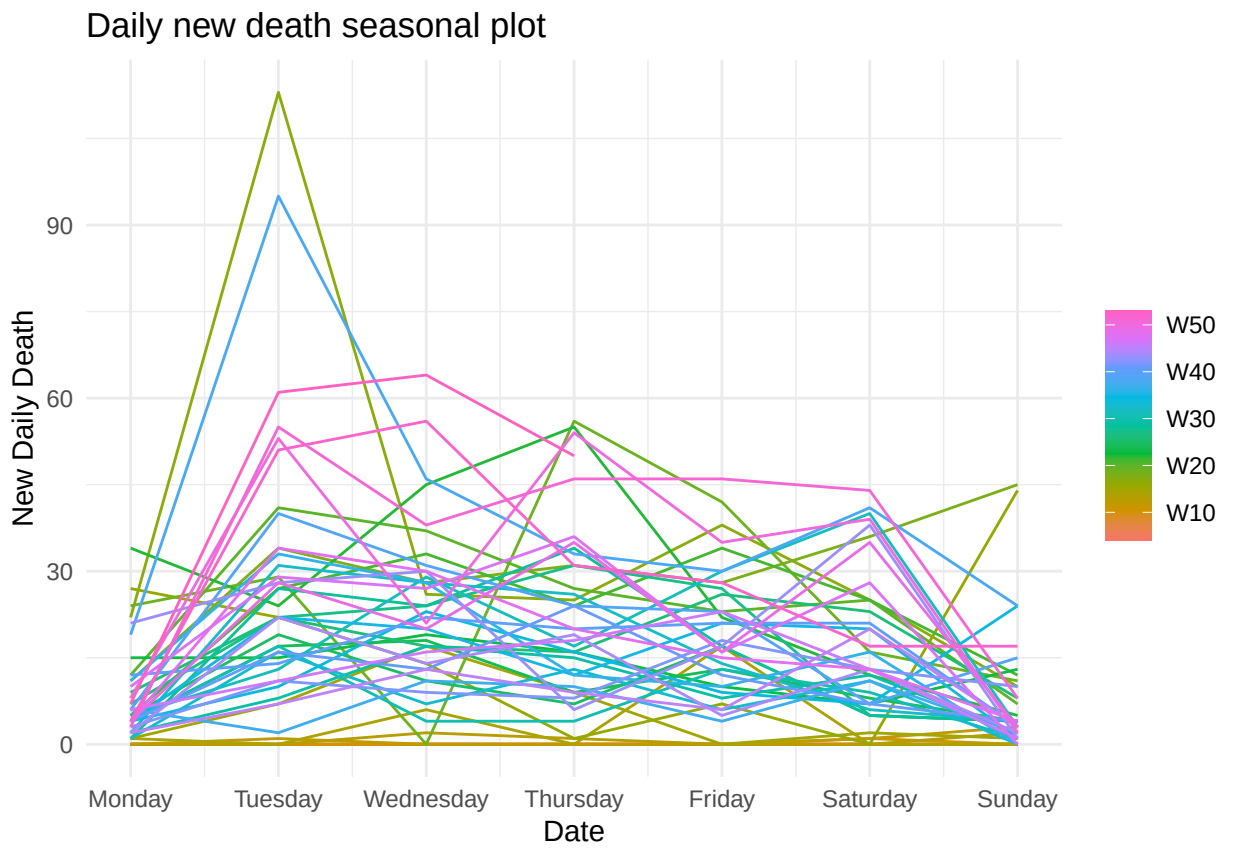
To describe above plot breifly, deaths recorded high during summer, and it's kind of low between June and September. It's raised significantly in October nad gradually increasing thereafter till January.

```
### autoplot for daily new infection
autoplot_daily_death <- state.ts %>%
  autoplot(Y.Infected) + theme_minimal() + labs(title="Daily Infection using autoplot",x="Date", y="New

autoplot_daily_death
```



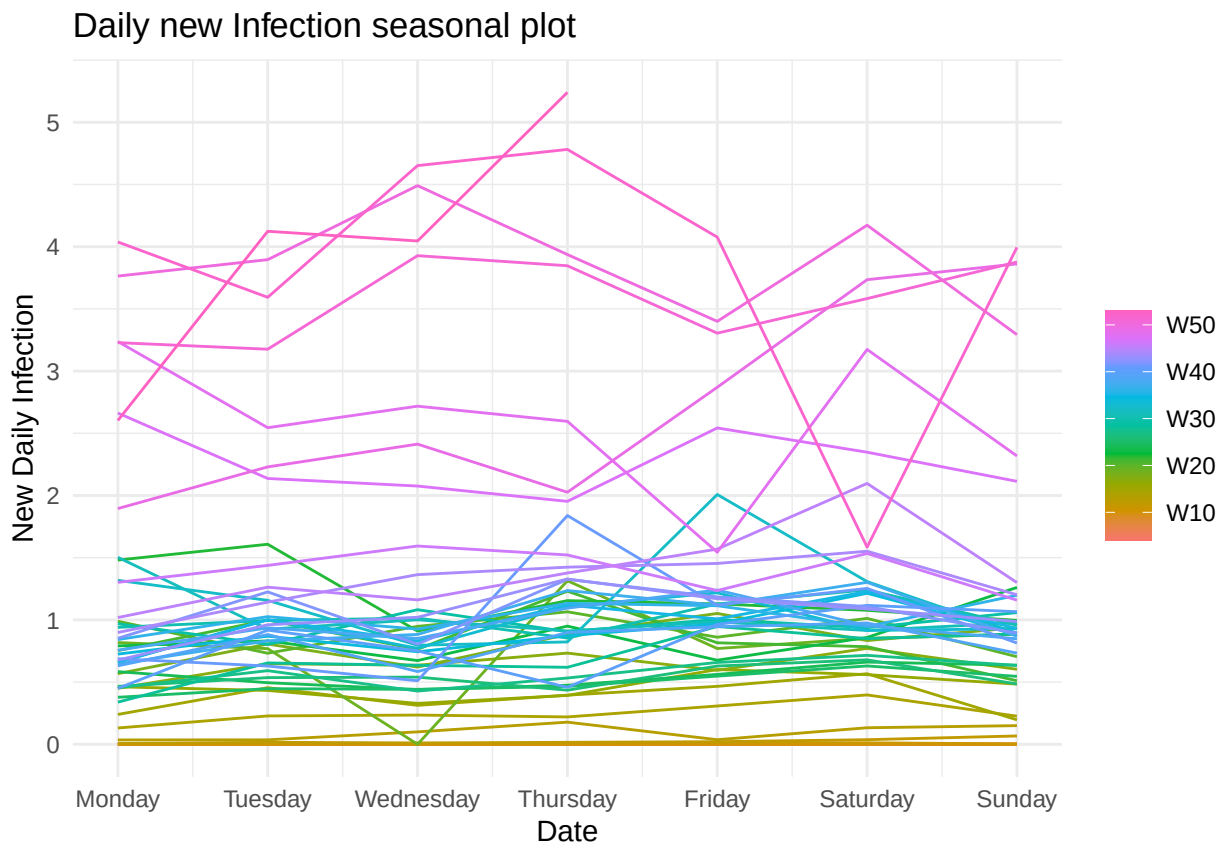
The timeseries plot of infections shows us a gradually increasing trend from April throogh January with slight drops in July and October

```
###seasonal plot for daily new Death
ggseason.death <- state.ts %>% gg_season(Y.Death, period = "week") +
  theme_minimal() + labs(title="Daily new death seasonal plot",
                         x="Date", y="New Daily Death")
ggseason.death
```

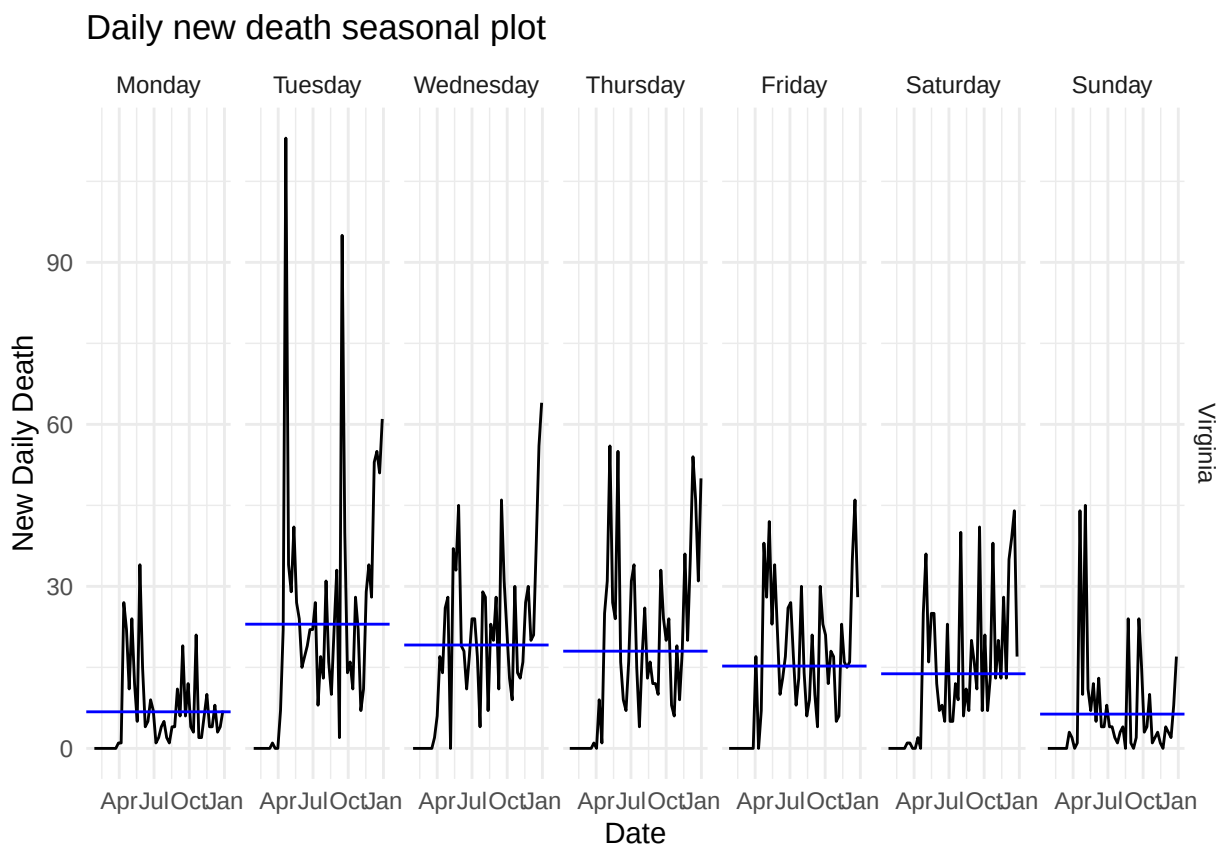


The seasonal plot of death is kind of Hard to interpret, i would say the deaths are recorded high during the start of the week and reducing till end of the week, which is measured across 50 weeks starting from April Through January

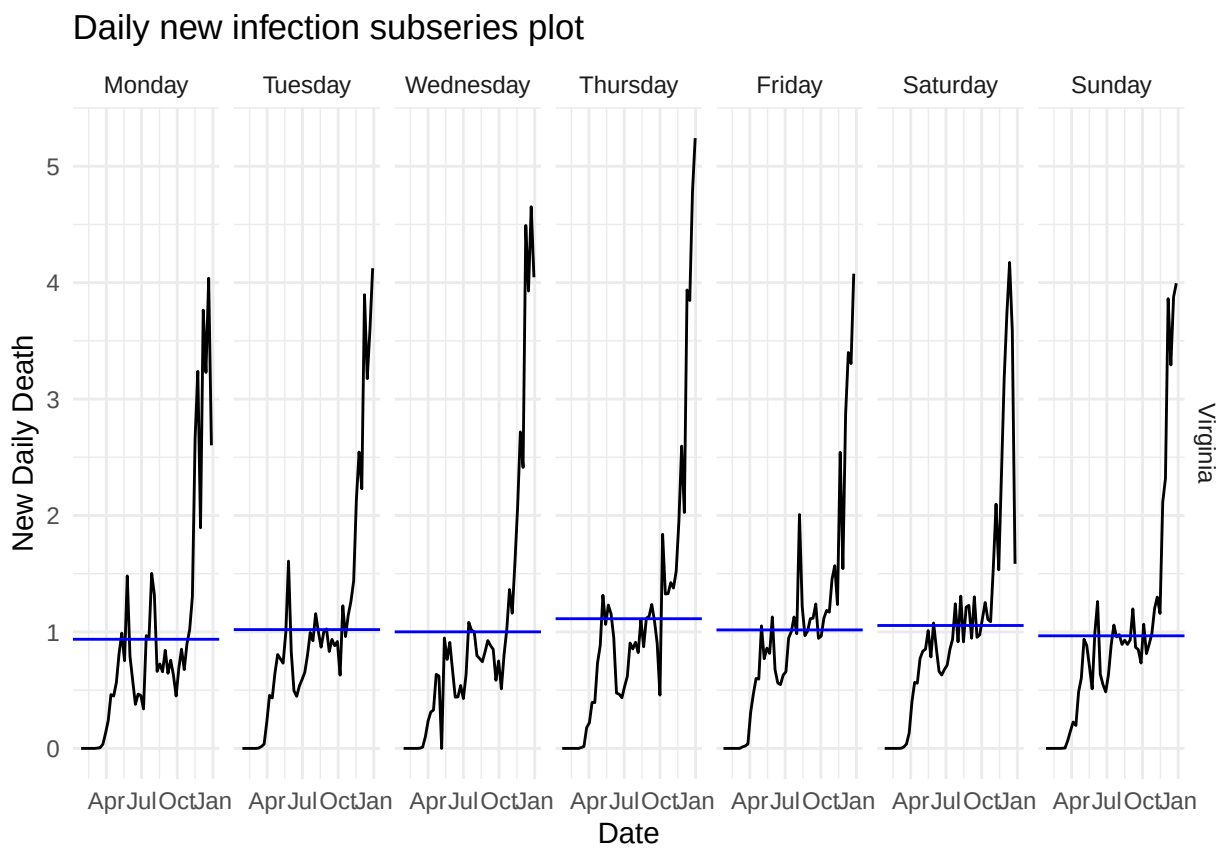
```
ggseason.infection <- state.ts %>% gg_season(Y.Infected, period = "week") +
  theme_minimal() + labs(title="Daily new Infection seasonal plot",
                         x="Date", y="New Daily Infection")
ggseason.infection
```



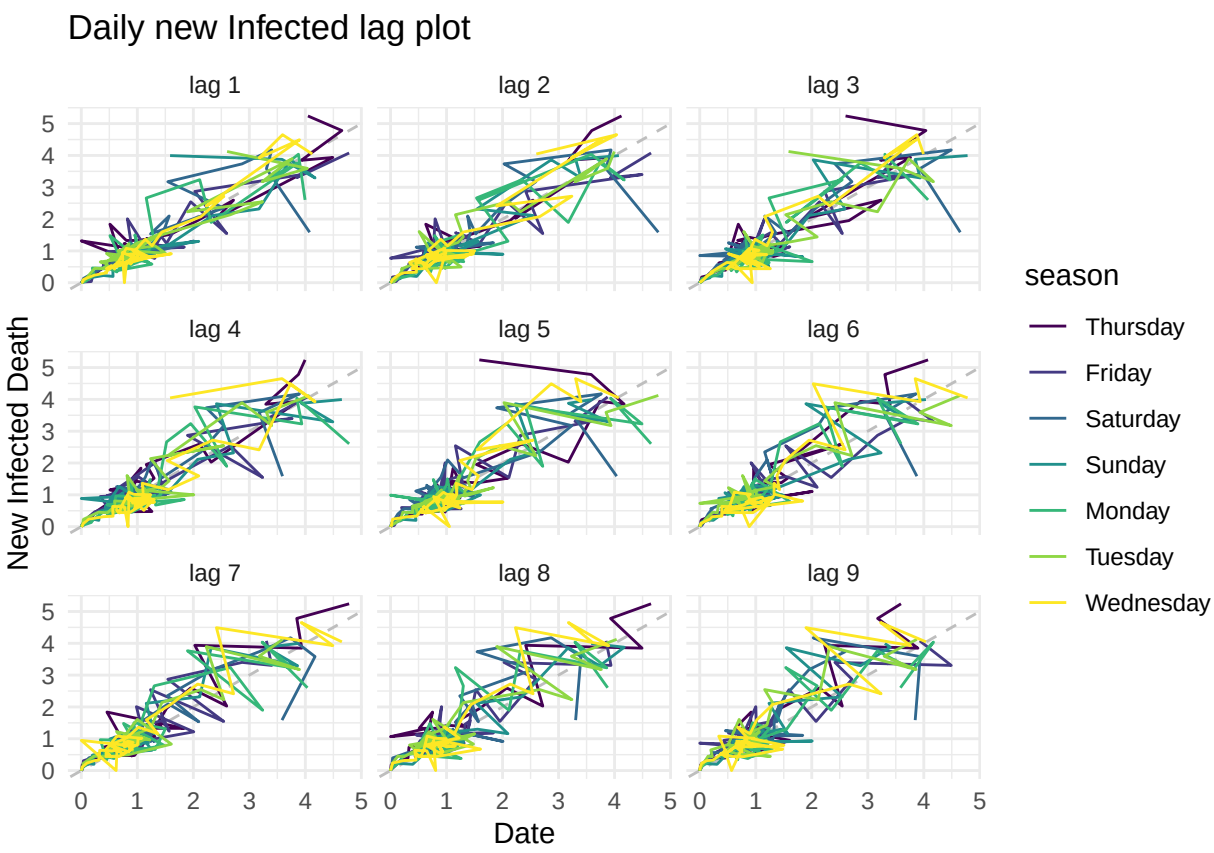

```
ggsubseries.death <- state.ts %>% gg_subseries(Y.Death, period = "week") +
  theme_minimal() + labs(title="Daily new death seasonal plot",
                         x="Date", y="New Daily Death")
ggsubseries.death
```




```
ggsubseries.Infected <- state.ts %>% gg_subseries(Y.Infected, period = "week") +
  theme_minimal() + labs(title="Daily new infection subseries plot",
                         x="Date", y="New Daily Death")
ggsubseries.Infected
```




```
gglag.Infected <- state.ts %>% gg_lag(Y.Infected, period = "week") +
  theme_minimal() + labs(title="Daily new Infected lag plot",
                         x="Date", y="New Infected Death")
gglag.Infected
```




```
gglag.death <- state.ts %>% gg_lag(Y.Death, period = "week") +
  theme_minimal() + labs(title="Daily new death lag plot",
                         x="Date", y="New Daily Death")
gglag.death
```


Daily new death lag plot
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
New Daily Death
Date
0
30
60
90
season
Thursday
Friday
Saturday
Sunday
Monday
Tuesday
Wednesday