Tutorial: Weather Markov chain

Almut Veraart

Autumn 2022

Analysing a weather Markov chain in R

In this tutorial we analysis the weather Markov chain studied in the lectures in R. This tutorial has been written using R version 4.2.1.

First we load the R packages which we need for our analysis.

```
library(markovchain) #For analysing Markov chains
library(diagram) #For plotting the transition diagram
library(matrixcalc) #For computing powers of matrices
```

If you have not worked with these packages before, then you will need to install them first by running e.g.

```
#install.packages("markovchain")
#library(markovchain)
```

where you need to remove the "#" sign.

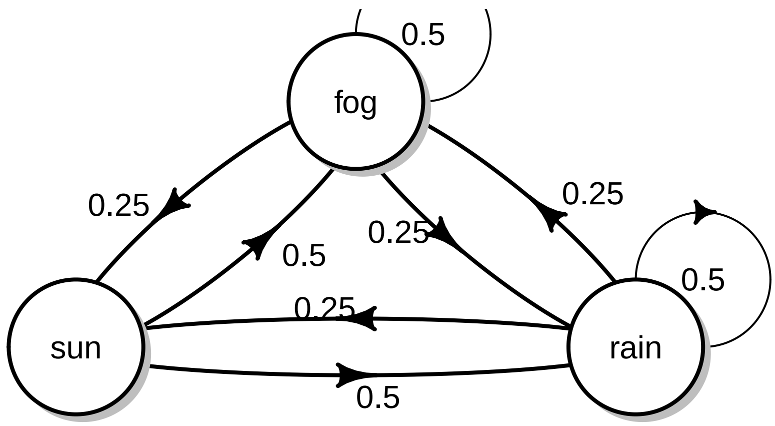
Define the Markov chain and plot the transition diagram

We can now define the weather Markov chain as follows.

```
#Define the state space (You can either use the state numbers or names)
#E <- c(1,2,3)
E <- c("rain","sun","fog")
#Define the transition matrix
P <- matrix(data = c(0.5, 0.25, 0.25,
                      0.5, 0, 0.5,
                      0.25, 0.25, 0.5),
                     byrow = TRUE, nrow = 3, dimnames = list(E,E)
                     )
WeatherMC <- new("markovchain", states = E, transitionMatrix = P, name = "Weather")
```

Let us plot the transition diagram of this Markov chain.

```
#Plot the transition diagramme
#Note that you need to transpose the transition matrix in the plotmat function
#(using the function t)
plotmat(t(P), relsize=0.8)
```



The evolution of the marginal distribution

Let us now study the evolution of this Markov chain in time. More specifically we want to compute

$$\nu^{(n)} = \nu^{(0)} P,$$

for various $n \in \mathbb{N}$.

```r
#Assign an initial distribution nu^(0)
nu0 <- c(0.5,0.5,0)

#Compute nu^(1)
nu1<-nu0 %*% P
nu1
```

```
##      rain   sun   fog
## [1,]  0.5 0.125 0.375
```

```r
#Find P2 (the two-step transition matrix)
#Note that  %*% is the matrix multiplication operator
P2 <-P %*% P
P2
```

```
##        rain    sun    fog
## rain 0.4375 0.1875 0.3750
## sun  0.3750 0.2500 0.3750
## fog  0.3750 0.1875 0.4375
```

```r
#Alternatively, you can use the package "matrixcal" for computing powers of matrices
matrix.power(P,2)
```

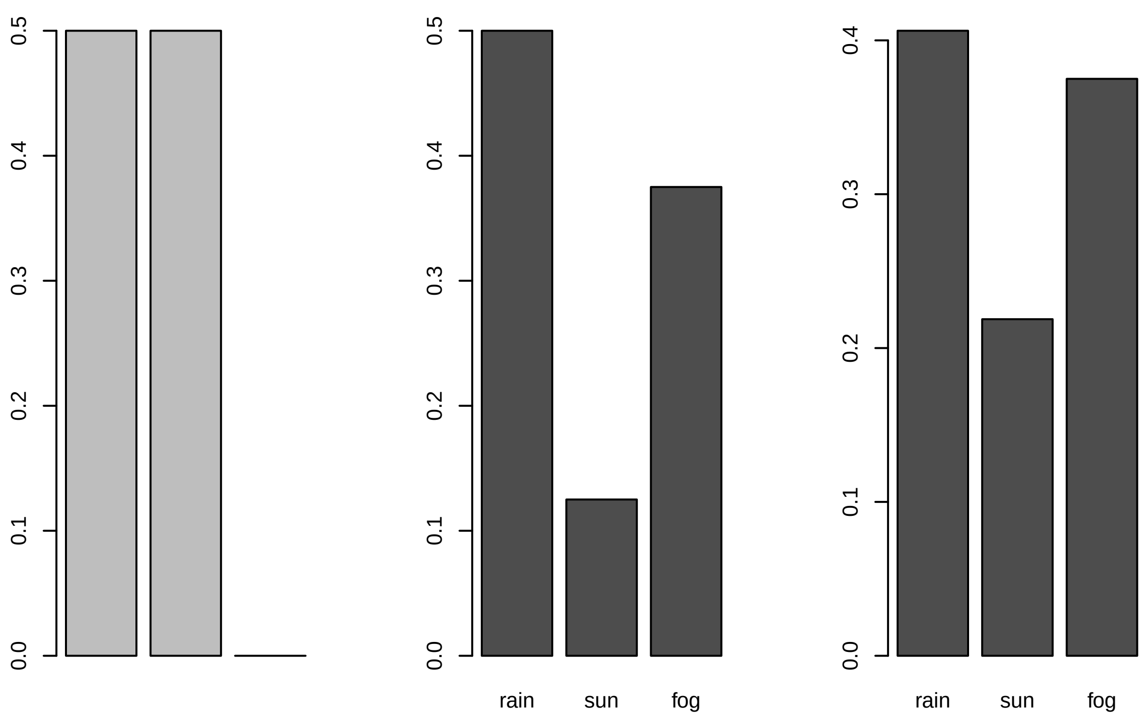
```
##        rain    sun    fog
## rain 0.4375 0.1875 0.3750
## sun  0.3750 0.2500 0.3750
## fog  0.3750 0.1875 0.4375
```

```r
#Find nu^(2):
nu2 <- nu0 %*% P2
nu2
```

```
##         rain     sun   fog
## [1,] 0.40625 0.21875 0.375
```

Let us plot the histograms of the first three marginal distributions.

```r
par(mfrow = c(1,3))
barplot(nu0)
barplot(nu1)
barplot(nu2)
```



Is there a limiting distribution?

Let us compute some more marginal distributions $\nu^{(n)}$ for $n = 1, \ldots, 20$. What do you observe?

```r
for(i in 1:20)
{
  print(nu0 %*% matrix.power(P,i))
}
```

```
##      rain   sun   fog
## [1,]  0.5 0.125 0.375
##         rain     sun   fog
## [1,] 0.40625 0.21875 0.375
##         rain       sun       fog
## [1,] 0.40625 0.1953125 0.3984375
##           rain       sun       fog
## [1,] 0.4003906 0.2011719 0.3984375
##           rain      sun       fog
## [1,] 0.4003906 0.199707 0.3999023
##           rain       sun       fog
## [1,] 0.4000244 0.2000732 0.3999023
##           rain       sun       fog
## [1,] 0.4000244 0.1999817 0.3999939
##           rain       sun       fog
## [1,] 0.4000015 0.2000046 0.3999939
##           rain       sun       fog
## [1,] 0.4000015 0.1999989 0.3999996
##           rain       sun       fog
## [1,] 0.4000001 0.2000003 0.3999996
##           rain       sun fog
## [1,] 0.4000001 0.1999999 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
##      rain sun fog
## [1,]  0.4 0.2 0.4
```

Do the n-step transition probabilities converge?

```r
for(i in 1:20)
{
  print(matrix.power(P,i))
}
```

```
##      rain  sun  fog
## rain 0.50 0.25 0.25
```

```
## sun  0.50 0.00 0.50
## fog  0.25 0.25 0.50
##        rain    sun    fog
## rain 0.4375 0.1875 0.3750
## sun  0.3750 0.2500 0.3750
## fog  0.3750 0.1875 0.4375
##          rain      sun      fog
## rain 0.406250 0.203125 0.390625
## sun  0.406250 0.187500 0.406250
## fog  0.390625 0.203125 0.406250
##           rain       sun       fog
## rain 0.4023438 0.1992188 0.3984375
## sun  0.3984375 0.2031250 0.3984375
## fog  0.3984375 0.1992188 0.4023438
##           rain       sun       fog
## rain 0.4003906 0.2001953 0.3994141
## sun  0.4003906 0.1992188 0.4003906
## fog  0.3994141 0.2001953 0.4003906
##           rain       sun       fog
## rain 0.4001465 0.1999512 0.3999023
## sun  0.3999023 0.2001953 0.3999023
## fog  0.3999023 0.1999512 0.4001465
##           rain       sun       fog
## rain 0.4000244 0.2000122 0.3999634
## sun  0.4000244 0.1999512 0.4000244
## fog  0.3999634 0.2000122 0.4000244
##           rain       sun       fog
## rain 0.4000092 0.1999969 0.3999939
## sun  0.3999939 0.2000122 0.3999939
## fog  0.3999939 0.1999969 0.4000092
##           rain       sun       fog
## rain 0.4000015 0.2000008 0.3999977
## sun  0.4000015 0.1999969 0.4000015
## fog  0.3999977 0.2000008 0.4000015
##           rain       sun       fog
## rain 0.4000006 0.1999998 0.3999996
## sun  0.3999996 0.2000008 0.3999996
## fog  0.3999996 0.1999998 0.4000006
##           rain       sun       fog
## rain 0.4000001 0.2000000 0.3999999
## sun  0.4000001 0.1999998 0.4000001
## fog  0.3999999 0.2000000 0.4000001
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
```

```
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
##      rain sun fog
## rain  0.4 0.2 0.4
## sun   0.4 0.2 0.4
## fog   0.4 0.2 0.4
```

We observe that both the marginal distributions and the transition probabilities seem to converge. This is not always the case, but seems to be the case for this particular Markov chain. We will later formally define what we mean by a *limiting distribution* and under which conditions a Markov chain has a limiting distribution.

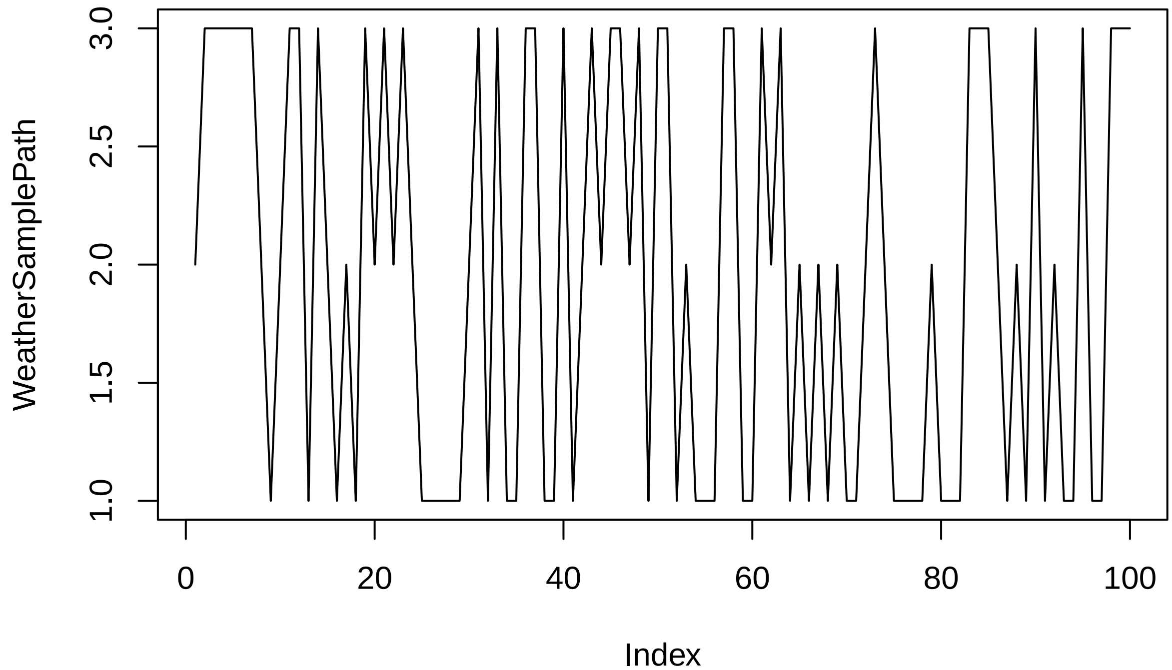
Simulating a sample path of the weather Markov chain

We can also simulate the Markov chain using the following function:

```
simMC <-function(E, nu0, P, length=100)
{
  samplepath<-integer(length)
  samplepath[1] <- sample(E,1,prob=nu0) #Draw X_0 from the initial distribution

  for(i in 2:length){
    #In each step we sample from the states in E
    #with the probability weights given by the row of P
    #corresponding to the previous state
    probvec <- P[samplepath[(i-1)],]
    samplepath[i]<-sample(E,1,prob=probvec)
  }
  samplepath
}
```

We will now use the above function to simulate one sample path of the weather Markov chain. Feel free to change the initial distribution $\nu^{(0)}$ in the code below!

```
WeatherSamplePath <- simMC(c(1,2,3), nu0, P)
plot(WeatherSamplePath, type="l")
```



Creating the plots from the lecture notes

In the following, I show you which code I used for creating the plots given in the lecture notes. Note that you can produce very nice graphics using the package ggplot2.

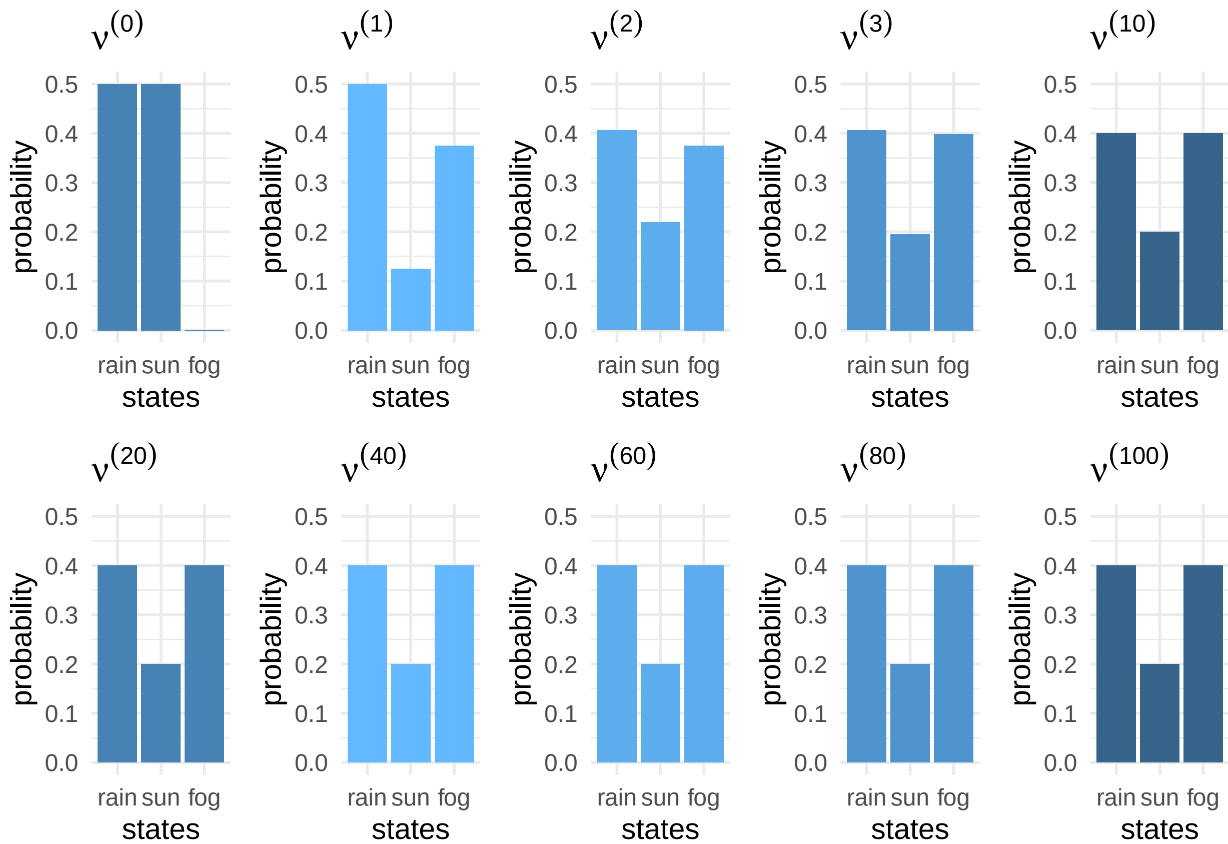
```
library(ggplot2)  #For very pretty plots
library(latex2exp) #For LaTex annotations in the graphs
library(gridExtra) #For combining several plots in one picture
```

First we plot the marginal distributions $\nu^{(n)}$:

```
######Plot of the marginal distributions nu^(n)#####################

#Compute the nu^(n), for n=3,10,20,40,60,80,100
nu3 <- nu0 %*% matrix.power(P,3)
nu10 <- nu0 %*% matrix.power(P,10)
nu20 <- nu0 %*% matrix.power(P,20)
nu40 <- nu0 %*% matrix.power(P,40)
nu60 <- nu0 %*% matrix.power(P,60)
nu80 <- nu0 %*% matrix.power(P,80)
nu100 <- nu0 %*% matrix.power(P,100)


#Create a data.frame with the relevant data
df_nu0 <- data.frame(states=factor(E, levels=E), probability=nu0)#use factor to impose the order of the
df_nu1 <- data.frame(states=factor(E, levels=E), probability=nu1[1,])
```

```r
df_nu2 <- data.frame(states=factor(E, levels=E), probability=nu2[1,])
df_nu3 <- data.frame(states=factor(E, levels=E), probability=nu3[1,])
df_nu10 <- data.frame(states=factor(E, levels=E), probability=nu10[1,])
df_nu20 <- data.frame(states=factor(E, levels=E), probability=nu20[1,])
df_nu40 <- data.frame(states=factor(E, levels=E), probability=nu40[1,])
df_nu60 <- data.frame(states=factor(E, levels=E), probability=nu60[1,])
df_nu80 <- data.frame(states=factor(E, levels=E), probability=nu80[1,])
df_nu100 <- data.frame(states=factor(E, levels=E), probability=nu100[1,])



p_nu0 <-ggplot(data=df_nu0, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(0)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue")+
  theme_minimal()


p_nu1 <-ggplot(data=df_nu1, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(1)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue1")+
  theme_minimal()


p_nu2 <-ggplot(data=df_nu2, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(2)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue2")+
  theme_minimal()


p_nu3 <-ggplot(data=df_nu3, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(3)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue3")+
  theme_minimal()


p_nu10 <-ggplot(data=df_nu10, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(10)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue4")+
  theme_minimal()


p_nu20 <-ggplot(data=df_nu20, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(20)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue")+
  theme_minimal()
```

```r
p_nu40 <-ggplot(data=df_nu40, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(40)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue1")+
  theme_minimal()


p_nu60 <-ggplot(data=df_nu60, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(60)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue2")+
  theme_minimal()


p_nu80 <-ggplot(data=df_nu80, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(80)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue3")+
  theme_minimal()


p_nu100 <-ggplot(data=df_nu100, aes(x=states, y=probability)) +
  ggtitle(TeX("$\\nu^{(100)}$"))+
  ylim(0, 0.5)+
  geom_bar(stat="identity", fill="steelblue4")+
  theme_minimal()



grid.arrange(p_nu0, p_nu1, p_nu2, p_nu3, p_nu10, p_nu20, p_nu40, p_nu60, p_nu80, p_nu100,  ncol=5)
```



Next, we simulate three sample paths of the weather Markov chain:

```
set.seed(1) #Fix the seed to get re-producible results!
len <- 50
WeatherSamplePath1 <- simMC(c(1,2,3), nu0, P, len)
WeatherSamplePath2 <- simMC(c(1,2,3), nu0, P, len)
WeatherSamplePath3 <- simMC(c(1,2,3), nu0, P, len)

df_WeatherPath1 <- data.frame(n=(1:len), states=WeatherSamplePath1)
df_WeatherPath2 <- data.frame(n=(1:len), states=WeatherSamplePath2)
df_WeatherPath3 <- data.frame(n=(1:len), states=WeatherSamplePath3)
p_path1 <-ggplot(data=df_WeatherPath1, aes(x=n, y=states,group=1)) +
  ggtitle(TeX("$X_n(\\omega_1)$"))+
  geom_line( color="steelblue", linetype="dotted") +
  geom_point()+
  theme_minimal()+
  scale_y_continuous(breaks=c(1,2,3))+
  theme(panel.grid.minor = element_blank())
p_path2 <-ggplot(data=df_WeatherPath2, aes(x=n, y=states,group=1)) +
  ggtitle(TeX("$X_n(\\omega_2)$"))+
  geom_line( color="steelblue", linetype="dotted") +
  geom_point()+
  theme_minimal()+
  scale_y_continuous(breaks=c(1,2,3))+
  theme(panel.grid.minor = element_blank())
p_path3 <-ggplot(data=df_WeatherPath3, aes(x=n, y=states,group=1)) +
  ggtitle(TeX("$X_n(\\omega_3)$"))+
```

```
  geom_line( color="steelblue", linetype="dotted") +
  geom_point()+
  theme_minimal()+
  scale_y_continuous(breaks=c(1,2,3))+
  theme(panel.grid.minor = element_blank())

######Plot of three sample paths of the weather chain #####################
grid.arrange(p_path1, p_path2, p_path3,  ncol=1)
```