VII. 시각화 컨트롤

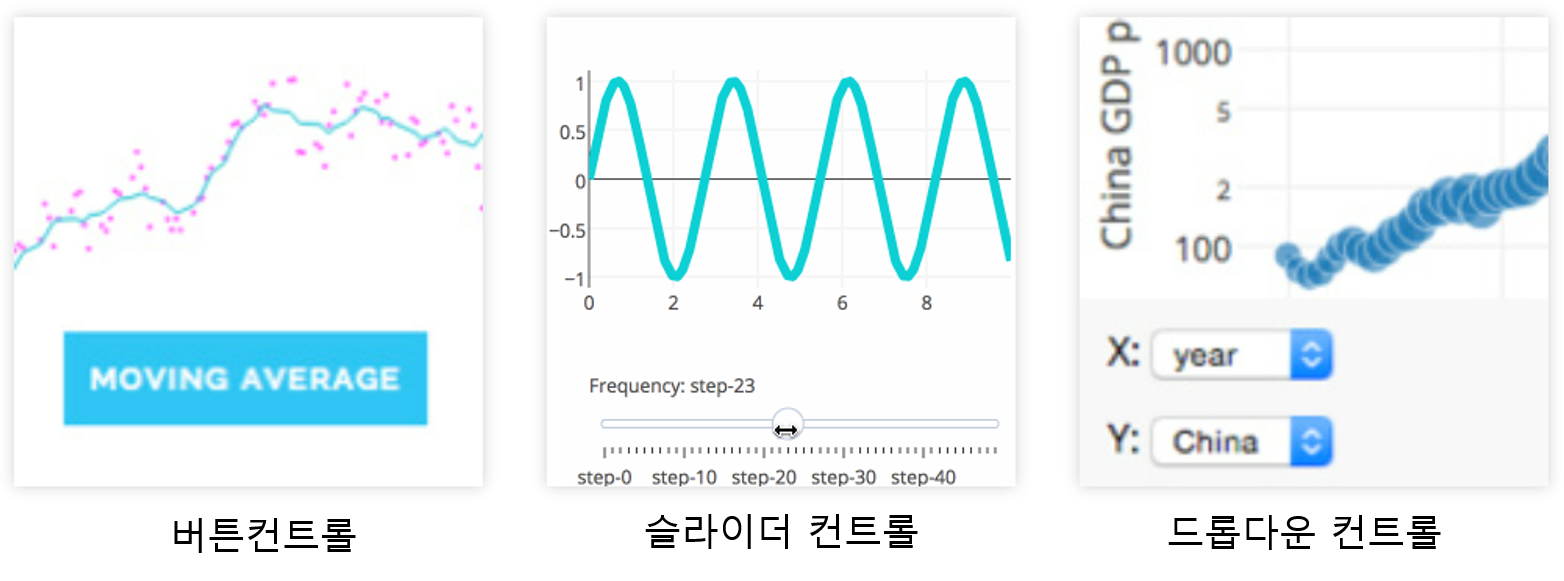
plotly 시각화는 동적 시각화로써 웹을 통해 배포될때 가장 효과가 크다. 웹을 사용한다는 것은 웹에서 제공하는 많은 기능을 사용할 수 있다는 장점이 있다. plotly 시각화를 만들때 설명했던 것처럼 호버를 사용한 사용자 반응형 기능이나 다음 장에서 설명할 범례나 마우스를 사용한 사용자 반응형 기능을 기본적으로 제공하지만 버튼, 드롭다운 버튼, 슬라이더의 세가지 컨트롤을 추가적으로 제공한다. 이 기능을 사용하면 시각화 내에서 데이터를 필터링 하거나 원하는 데이터만 추출하는 등 아주 기초적인 대시보드의 역할을 할 수 있다.

plotly의 컨트롤 사용을 알아보기 위해 2010년부터 2022년까지의 전국 17개 시도별 대학 신입생 충원율 자료를 사용하겠다.

df\_충원율\_botton <- read\_excel('D:/R/git/datavisualization/plotly/RnPy/chap6/고등 주요 01-시도별 신입생 충원율(2010-2022)\_220825y.xlsx',   
 sheet = 'Sheet1',  
 skip = 7,   
 col\_names = FALSE,   
 col\_types = c(rep('text', 2), rep('numeric', 12)))  
  
df\_충원율\_botton <- df\_충원율\_botton |> dplyr::select(1, 2, 5)  
  
colnames(df\_충원율\_botton) <- c('연도', '지역', '신입생충원율')  
  
df\_충원율\_botton <- df\_충원율\_botton |> pivot\_wider(names\_from = '연도', values\_from = '신입생충원율')  
  
df\_충원율\_botton <- as.data.frame(df\_충원율\_botton)

import pandas as pd  
import plotly.graph\_objects as go  
  
df\_충원율\_control = pd.read\_excel("D:/R/git/datavisualization/plotly/RnPy/chap6/고등 주요 01-시도별 신입생 충원율(2010-2022)\_220825y.xlsx",   
 sheet\_name = 'Sheet1',  
 skiprows=(6),   
 header = 0)  
df\_충원율\_control = df\_충원율\_control.iloc[:, [0, 1, 4]]  
df\_충원율\_control.columns = ('연도', '지역', '신입생충원율')  
df\_충원율\_control = df\_충원율\_control.pivot(index = '지역', columns = '연도', values = '신입생충원율').reset\_index()

plotly에서 제공하는 시각화 컨트롤은 데이터와 레이아웃을 정해진 방법대로 설정을 변경하는 버튼 컨트롤, 몇가지 선택사항중에 하나를 선택하는 드롭다운 버튼 컨트롤, 마우스 드래그를 통해 값을 설정하는 슬라이더의 세 가지 컨트롤을 사용할 수 있다. 이 중 버튼 컨트롤과 드롭다운 버튼 컨트롤은 같은 속성들을 사용하는데 버튼 타입을 설정하는 속성을 통해 버튼을 사용할지 드롭다운을 설정할지를 결정하게 된다. 하지만 슬라이더 컨트롤은 버튼 컨트롤과 속성이 매우 다르다.

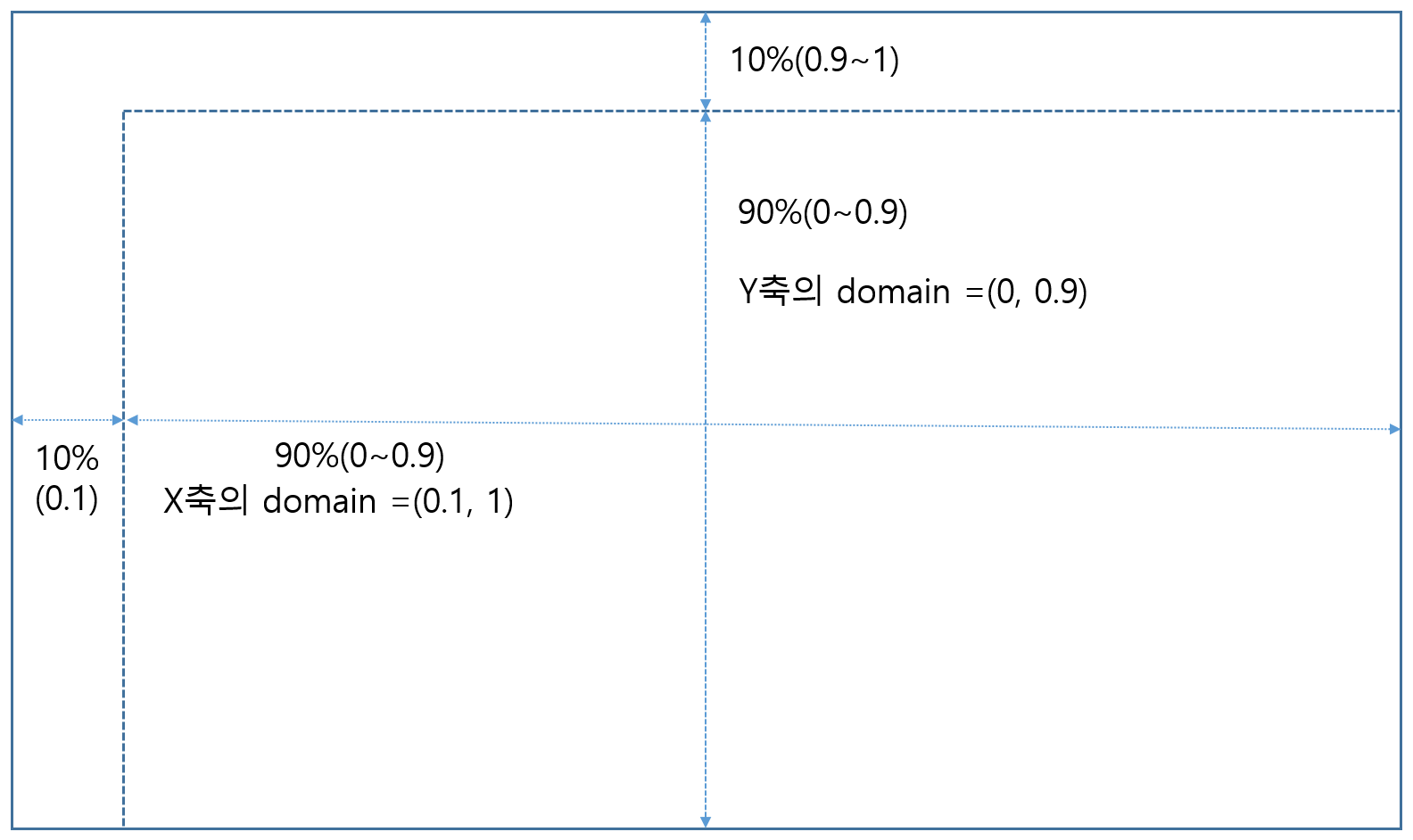


위의 메쏘드들은 ‘layout’ 속성의 하위 속성인 ‘updatemenus’ 속성으로 설정이 가능하다. 의 하위 속성인 에서 해당 버튼의 메쏘드를 설정한다.

# plotly 위치 설정

컨트롤을 생성하기 전에 가장 먼저 결정해야 하는 것은 어떤 컨트롤을 만들지에 대한 결정일 것이다. 해당 컨트이 수행해야 할 기능에 따라 컨트롤의 종류가 달라질 것이다. 컨트롤의 종류가 결정되면 다음번 고민은 이 컨트롤을 어디에 배치할지를 결정해야한다. plotly 시각화에서 컨트롤을 추가하는 것은 전체 플롯의 범위에서 plotly 그래프와의 공간을 공유하기 때문에 plotly 그래프의 크기를 어느 정도 크기로 어디에 배치할 것이고 컨트롤들을 어디에 배치 할지에 대한 결정이 이루어져야 한다. 결정이 되었다면 먼저 plotly 시각화의 크기를 ‘domain’ 속성을 사용하여 설정하여야 한다.

plotly의 크기와 위치 설정을 위해 사용하는 ‘domain’ 속성은 ‘layout’의 ’xaxis’와 ’yaxis’의 하위 속성이다. 이 ’domain’ 속성은 전체 플로팅 영역에서 X축과 Y축이 그려지는 좌측 하단의 시작 위치와 우측 상단의 끝 위치에 비례한 0과 1사이의 수치값을 통해 설정한다. 다음의 그림과 같이 전체 플로팅 영역의 오른쪽 아래에 위치시키고 왼쪽과 상단에 10%씩의 공간을 남기기 위해서는 ’xaxis’의 ’domain’은 시작 위치를 0.1, 끝 위치를 1로 설정하고 ’yaxis’의 ’domain’은 시작 위치를 0, 끝 위치를 0.9로 설정하여야 한다.



# 버튼 컨트롤

버튼 컨트롤은 plotly로 만들어진 객체의 data 속성이나 layout 속성의 속성값을 변경하여 plotly 시각화에 변경을 가하는 컨트롤을 말한다. 다른 컨트롤에서도 동일하지만 버튼 컨트롤로 어떤 속성을 갱신하느냐에 따라 해당 버튼의 속성을 설정하여야 하는데 이를 버튼 메쏘드(Method)라고 하며 메쏘드에는 다음의 네 가지가 있다.

* “update”

데이터 속성과 레이아웃 속성이 모두 갱신되는 메쏘드

* “restyle”

데이터 속성만 갱신되는 메쏘드

* “relayout”

레이아웃 속성만 갱신되는 메쏘드

* “animate”

애니메이션의 시작과 중단을 설정하는 메쏘드

해당 버튼이 어떤 속성을 갱신하는지에 따라 위와 같이 네가지 메쏘드 중에 하나를 설정하여야 하는데 만약 이 메쏘드의 설정이 적절하지 않으면 버튼은 작동하지 않는다. 예를 들어 ‘update’ 메쏘드로 속성을 설정하였는데 데이터 속성이나 레이아웃 속성 중 하나만 갱신하도록 설정되면 해당 버튼은 작동하지 않는다. 또 data 속성을 갱신하면서 ‘relayout’ 메쏘드로 속성을 설정하거나 layout 속성을 갱신하면서 ‘restyle’ 메쏘드로 속성을 설정하면 작동하지 않는다.

버튼 컨트롤은 ‘layout’ 속성의 ‘updatemenus’ 속성을 통해서 설정이 가능한데 다음은 버튼에서 주로 사용하는 하위 속성이다. 이 ‘updatemenus’ 속성에는 여러개의 버튼 컨트롤을 정의할 수 있기 때문에 ’updatemenus’는 각각의 버튼 설정들의 리스트로 구성하여야 한다.



### restyle 버튼 생성

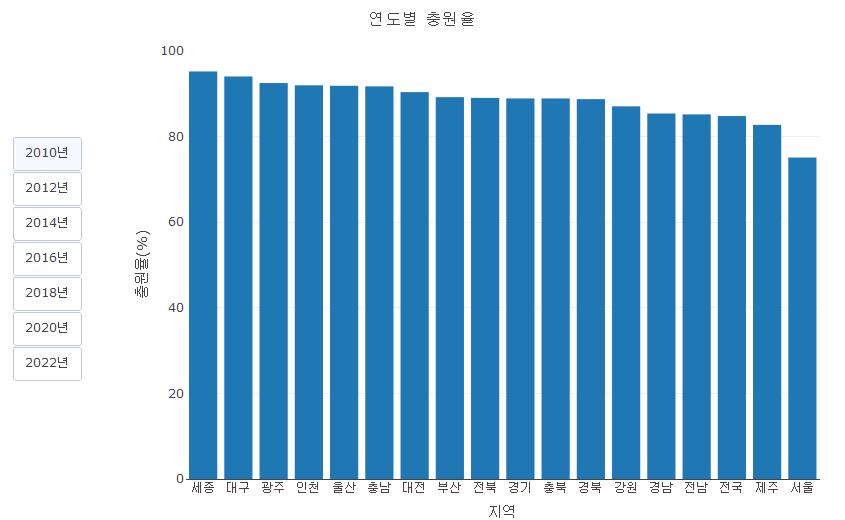
restyle 버튼은 해당 버튼을 클릭함으로써 data 속성이 갱신되는 버튼을 말한다. 앞 서 설명한 바와 같이 ’updatemenus’의 속성으로는 각각의 버튼들의 속성들이 정의된 딕셔너리(R은 리스트)의 리스트로 설정해야 한다.

restyle 버튼이 클릭되면 갱신되어야 하는 data 속성들은 ‘args’ 속성으로 설정하는데 여러개의 data 속성이 갱신될 수 있기 때문에 ‘args’ 속성에는 속성이름과 속성값으로 구성된 딕셔너리(R은 리스트)의 리스트로 설정되어야 한다. 또 해당 버튼의 메쏘드를 설정하기 위한 ‘type’ 속성도 중요한데 ‘type’ 속성은 기본값이 “restyle”으로 설정된다. 버튼이 위치하는 ‘x’, ’y’의 값도 설정이 필요하고 버튼에 표시될 라벨의 설정도 기본적으로 필요하다.

다음은 각 시도별 대학 신입생 충원율의 bar 트레이스를 그리는데 restyle 버튼으로 연도를 설정하는 R과 python 코드이다. 먼저 bar 트레이스를 생성하고 ‘layout’ 속성의 ‘xaxis’, ‘yaxis’의 ’domain’을 설정하여 bar 트레이스의 위치를 설정하였다. 이후 ’updatemenus’ 속성으로 버튼을 설정하였는데 이 버튼은 총 7개의 세부 버튼으로 구성된 버튼 그룹이다. 이는 각각의 버튼으로 구성할 수도 있는데 이럴 경우 각각의 ‘method’, ‘type’과 ’x’, ‘y’ 위치 등을 각각 설정해야 하기 때문에 유사한 위치와 기능의 버튼들은 하나의 ‘button’ 설정안에 묶어주는 것이 좋다. 또 이렇게 설정하면 드롭다운 버튼으로의 전환도 쉬워진다.

* R

fig <- df\_충원율\_botton |>  
 plot\_ly() |>  
 add\_trace(type = 'bar', x = ~지역,   
 y = ~`2022`  
 )  
  
fig <- fig %>% layout(  
 title = "연도별 충원율",  
 xaxis = list(domain = c(0.1, 1), categoryorder = "total descending"),  
 yaxis = list(title = "충원율(%)"),  
 updatemenus = list(  
 list(  
 type = "buttons",  
 y = 0.8,  
 buttons = list(  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2010`)),  
 label = "2010년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2012`)),  
 label = "2012년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2014`)),  
 label = "2014년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2016`)),  
 label = "2016년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2018`)),  
 label = "2018년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2020`)),  
 label = "2020년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2022`)),  
 label = "2022년")))  
 ),  
 margin = margins\_R)  
  
fig



* python

fig = go.Figure()  
  
fig.add\_trace(go.Bar(  
 x = df\_충원율\_control['지역'],   
 y = df\_충원율\_control[2022]  
))  
  
fig.update\_layout(title = dict(text = "연도별 충원율", x = 0.5),  
 xaxis = dict(domain = (0.1, 1)),  
 yaxis = dict(title = "충원율(%)"),  
 updatemenus =[   
 dict(  
 type = "buttons",  
 y = 0.8,  
 buttons = [  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2010]])],  
 label = "2010년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2012]])],  
 label = "2012년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2014]])],  
 label = "2014년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2016]])],  
 label = "2016년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2018]])],  
 label = "2018년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2020]])],  
 label = "2020년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2022]])],  
 label = "2022년")]  
 )]  
)  
  
fig.show()

# 드롭다운 컨트롤

* R

fig <- df\_충원율\_botton |>  
 plot\_ly() |>  
 add\_trace(type = 'bar', x = ~지역,   
 y = ~`2022`  
 )  
  
fig <- fig %>% layout(  
 title = "연도별 충원율",  
 xaxis = list(domain = c(0.1, 1)),  
 yaxis = list(title = "충원율(%)"),  
 updatemenus = list(  
 list(  
# type = "buttons",  
 y = 0.8,  
 buttons = list(  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2010`)),  
 label = "2010년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2012`)),  
 label = "2012년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2014`)),  
 label = "2014년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2016`)),  
 label = "2016년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2018`)),  
 label = "2018년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2020`)),  
 label = "2020년"),  
 list(method = "restyle",  
 args = list("y", list(df\_충원율\_botton$`2022`)),  
 label = "2022년")))  
 ),   
 margin = margins\_R)  
  
fig

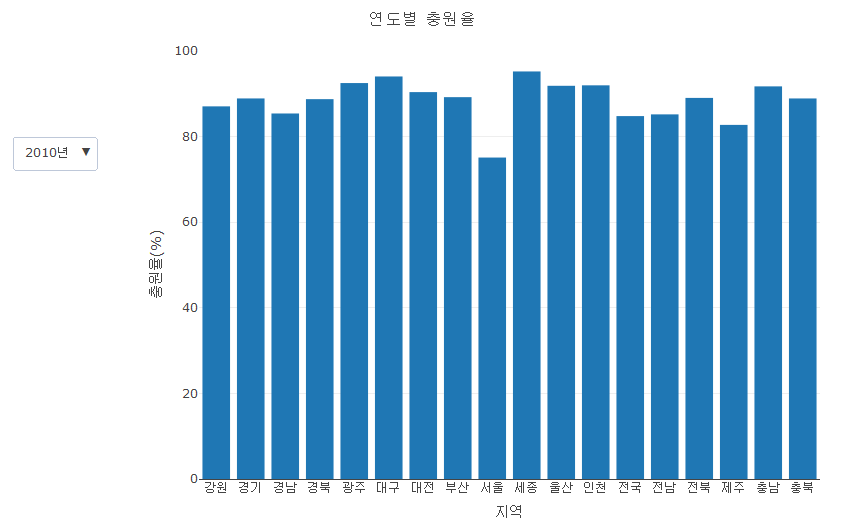
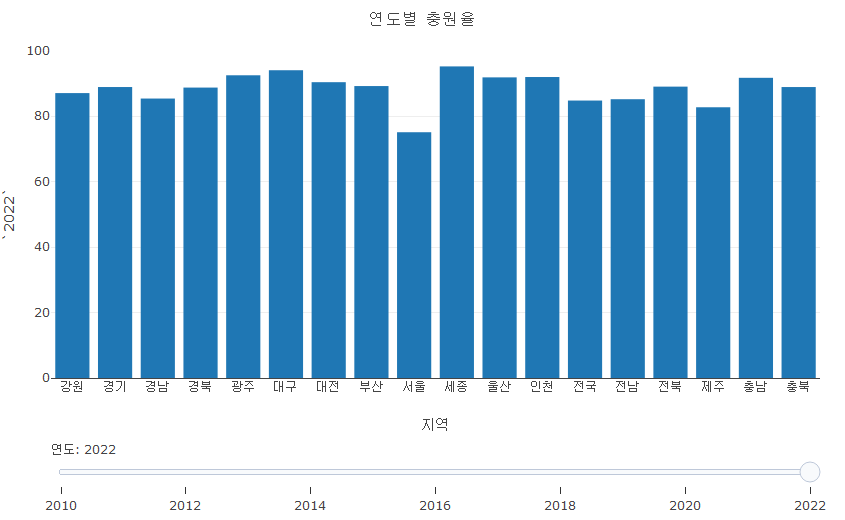


fig = go.Figure()  
  
fig.add\_trace(go.Bar(  
 x = df\_충원율\_control['지역'],   
 y = df\_충원율\_control[2022]  
))  
  
fig.update\_layout(title = dict(text = "연도별 충원율", x = 0.5),  
 xaxis = dict(domain = (0.1, 1)),  
 yaxis = dict(title = "충원율(%)"),  
 updatemenus =[   
 dict(  
# type = "buttons",  
 y = 0.8,  
 buttons = [  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2010]])],  
 label = "2010년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2012]])],  
 label = "2012년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2014]])],  
 label = "2014년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2016]])],  
 label = "2016년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2018]])],  
 label = "2018년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2020]])],  
 label = "2020년"),  
 dict(method = "restyle",  
 args = [dict(y = [df\_충원율\_control[2022]])],  
 label = "2022년")]  
 )]  
)  
  
fig.show()

# 슬라이더 컨트롤

* R

fig <- df\_충원율\_botton |>  
 plot\_ly() |>  
 add\_trace(type = 'bar', x = ~지역,   
 y = ~`2022`  
 )  
  
steps <- list(  
 list(args = list("y", list(df\_충원율\_botton$`2010`)),   
 label = "2010",   
 method = "restyle",   
 value = "1"  
 ),  
 list(args = list("y", list(df\_충원율\_botton$`2012`)),   
 label = "2012",   
 method = "restyle",   
 value = "2"  
 ),  
 list(args = list("y", list(df\_충원율\_botton$`2014`)),   
 label = "2014",   
 method = "restyle",   
 value = "3"  
 ),  
 list(args = list("y", list(df\_충원율\_botton$`2016`)),   
 label = "2016",   
 method = "restyle",   
 value = "4"  
 ),  
 list(args = list("y", list(df\_충원율\_botton$`2018`)),   
 label = "2018",   
 method = "restyle",   
 value = "5"  
 ),  
 list(args = list("y", list(df\_충원율\_botton$`2020`)),   
 label = "2020",   
 method = "restyle",   
 value = "6"  
 ),  
 list(args = list("y", list(df\_충원율\_botton$`2022`)),   
 label = "2022",   
 method = "restyle",   
 value = "7"  
 )  
)  
  
  
fig <- fig %>% layout(  
 title = "연도별 충원율",  
 sliders = list(  
 list(  
 active = 6,   
 currentvalue = list(prefix = "연도: "),   
 pad = list(t = 60),   
 steps = steps)),   
 margin = margins\_R)  
  
fig



* python

fig = go.Figure()  
  
fig.add\_trace(go.Bar(  
 x = df\_충원율\_control['지역'],   
 y = df\_충원율\_control[2022]  
))  
  
steps = [  
 dict(args = [dict(y = [df\_충원율\_control[2010]])],   
 label = "2010",   
 method = "restyle",   
 value = "1"  
 ),  
 dict(args = [dict(y = [df\_충원율\_control[2012]])],   
 label = "2012",   
 method = "restyle",   
 value = "2"  
 ),  
 dict(args = [dict(y = [df\_충원율\_control[2014]])],   
 label = "2014",   
 method = "restyle",   
 value = "3"  
 ),  
 dict(args = [dict(y = [df\_충원율\_control[2016]])],   
 label = "2016",   
 method = "restyle",   
 value = "4"  
 ),  
 dict(args = [dict(y = [df\_충원율\_control[2018]])],   
 label = "2018",   
 method = "restyle",   
 value = "5"  
 ),  
 dict(args = [dict(y = [df\_충원율\_control[2020]])],   
 label = "2020",   
 method = "restyle",   
 value = "6"  
 ),  
 dict(args = [dict(y = [df\_충원율\_control[2022]])],   
 label = "2022",   
 method = "restyle",   
 value = "7"  
 )]  
  
  
fig.update\_layout(title = dict(text = "연도별 충원율", x = 0.5),  
 xaxis = dict(domain = (0.1, 1)),  
 yaxis = dict(title = "충원율(%)"),  
 sliders = [dict(  
 dict(  
 active = 6,   
 currentvalue = dict(prefix = "연도: "),   
 pad = dict(t = 60),   
 steps = steps))]  
)  
  
fig.show()