Map link:
https://m24mendez.github.io/assignment2/

How would you add another class to your choropleth map?

For the present variable.
We can add another class for the function that is later used for the population variable.

```
function getColor(value) {
 return value > 50000000 ? '#54278f':
 value > 40000000 ? ‘#XXXXX:
 value > 25000000 ? '#756bb1':
 value > 10000000 ? '#9e9ac8':
 value > 5000000 ? '#cbc9e2':
 '#f2f0f7';
 }
```

For a new variable we would first add the data using a .js file in the header the same as we did with the provinces and the covid data. The new layer can be a health-related variable that tells us the risk based on the demographics of the province (if there are more old or young people).

Add this in the header:
src="https://m24mendez.github.io/assignment2/health_risk_data.js"></script>

And this in the body:

```
function getRiskColor(value) { …}

function riskStyle(feature){return { …}; }

var RiskLayer = new L.geoJson(RiskData, {style: riskStyle}).addTo(map);
```

How might you create a map where all overlay layers are toggled OFF by default? Why might this be a useful option?

We can simply not initialize the layers for the provinces and the covid data. So the user has the option to do it himself. This would allow the user to add layer by layer as they seem convenient and also it reduces the noise when first looking at the map if there is too much info or complexity.

```
var provinces = new L.geoJson(provinces, {
 style: function(feature) {
 return {
```

```
fillColor: getColor(feature.properties.POP10),
weight: 2,
opacity: 1,
color: 'gray',
fillOpacity: 0.9
};
}
});

var covid = new L.geoJson(covidData, {
pointToLayer: function(feature, ll){
return L.circleMarker(ll, {
color: '#B22222',
opacity: 1,
weight: 2,
fillColor: '#FF0000',
fillOpacity: 0.5,
radius: calcRadius(feature.properties.covid)
});
},
onEachFeature: onEachFeature
});

var basemaps = {
"Satellite": imagery
};

var overlaymaps = {
"China Provinces": provinces,
"COVID-19 Data": covid
};

L.control.layers(basemaps, overlaymaps, {collapsed: true}).addTo(map);
```