

D3 projekt – Graf odhadu výskytu tropické horečky v Kanadských provinciích

Použitá datová sada: <http://www.google.org/flutrends>

Projekt byl testován v Mozilla Firefox a bohužel nefunguje v prohlížečích Chrome a Edge, kvůli metodě pro loadování dat přímo z csv souboru.

Mapa Kanady

Je potřeba najít vhodnou mapu ve formátu svg, v projektu je po drobné úpravě použita tato:

https://upload.wikimedia.org/wikipedia/commons/3/38/Canada_blank_map.svg

Import dat

Data byla původně stažena v txt formátu, soubor jsme pak přeložili jako csv. Po vložení následujícího kódu do souboru myJScode.js a spuštění se můžeme podívat do konzole, jaký formát mají naše data – je to pole objektů, kdy objekt reprezentuje záznam hodnot v daném čase.

```
d3.csv("data.csv", function(d) {  
  console.log(d[0]);  
});
```

Hlavička původního CSV souboru se použila jako Property names pro datové objekty. Použití funkce d3.csv vyžaduje, aby váš csv soubor hlavičku měl.

Všechny hodnoty jsou řetězce, to ale pravděpodobně nechceme v případě čísel. Pro korektní práci s daty tedy musíme udělat typovou konverzi. Jednoduchý způsob jak toho docílit je použití unárního operátoru plus. Pro iteraci přes všechny prvky pole použijeme forEach. Pomocí tečkové notace docílíme přístupu ke každé Property. Pokud má Property dvouslovný název, je potřeba použít následující notaci.

```
d3.csv("data.csv", function(d) {  
  d.forEach(function(data){  
    data.Canada = +data.Canada;  
    data["New Brunswick"] = +data["New Brunswick"];  
  })  
  console.log(d[0]);  
});
```

K této úpravě můžete přímo dospět také použitím parametru d3.csv funkce, který určuje schéma Properties datové sady. Uložíme si názvy Properties jako kódy provincií.

```
d3.csv("data.csv", function(d) {  
  return {  
    date : d.Date,  
    canada : +d.Canada,  
    AB : +d.Alberta,  
    BC : +d["British Columbia"],  
    MB : +d.Manitoba,
```

```

        NB : +d["New Brunswick"],
        NL : +d["Newfoundland and Labrador"],
        NS : +d["Nova Scotia"],
        ON : +d.Ontario,
        SK : +d.Saskatchewan,
        QC : +d.Quebec
    };
}, function(d){
    console.log(d[0]);
});

```

D3.csv je asynchronní metoda a cokoliv za touto metodou se provede ještě dříve, než dojde k naložování dat. Zpracování dat tedy musí probíhat přímo v bloku, kde jsme doteď data vypisovali. Tento blok se provede přesně v moment, kdy se data načtou. Abychom mohli data pohodlně zpracovávat, uložíme si odkaz na ně do proměnné. Vytvoříme si také funkci, která bude obsluhovat celou vizualizaci dat. Celý kód by měl vypadat následovně:

```

var data;

d3.csv("data.csv", function(d) {
    return {
        date : d.Date,
        canada : +d.Canada,
        AB : +d.Alberta,
        BC : +d["British Columbia"],
        MB : +d.Manitoba,
        NB : +d["New Brunswick"],
        NL : +d["Newfoundland and Labrador"],
        NS : +d["Nova Scotia"],
        ON : +d.Ontario,
        SK : +d.Saskatchewan,
        QC : +d.Quebec
    };
}, function(d){
    data = d;

    // vizualizace dat
    visualization();
});

function visualization() {}

```

Všechny následující funkce budeme psát do funkce visualization().

Dílčí úkoly vizualizace

- 1) zvýraznění vybraného státu v mapě;
- 2) po kliku na stát se zobrazí graf - barchart všech jeho hodnot v průběhu času;
- 3) v barchartu je vertikální linka označující vybraný týden, klikem do grafu se změní výběr týdne a zároveň se obarví všechny provincie v mapě podle jejich hodnot v daném týdnu;
- 4) po kliku do grafu také dojde k vypsání aktuálních vybraných hodnot – „status bar“ (textová informace o kódu vybrané provincii, vybraném časovém úseku, a také počtu nakažených ve vybrané provincii v daném čase. A když budete moc chtít, můžete doplnit výpis celého názvu provincie ☺);
- 5) zobrazení legendy.

Vytvoření základních proměnných

Abychom mohli pohodlně pracovat s odkazem na svg panel, kde máme mapu, uložíme si ji do proměnné canvas. Stejně tak vytvoříme i další proměnné – počet datových záznamů a proměnnou pro aktuálně vybraný stát.

```
var dataEntriesCount = data.length; //pocet zaznamu
var chosenState = "";

//vytvoreni promenne obsahujici odkaz na mapu
var canvas = d3.select("body").select(".map");
```

Kliknutí na provincii a její zvýraznění

Pro obsluhu kliknutí na libolnou provincii přidáme ke všem skupinám event handler pomocí metody on. Stát, na který bylo kliknuto, si uložíme do proměnné chosenState. Dále budeme potřebovat funkci, která vykoná změnu stylu vybraného státu.

```
function borderIt(obj){
    d3.selectAll("g").style("stroke-width", 0);
    d3.select(obj).style("stroke", "yellow").style("stroke-width", 4);
}

canvas.selectAll("g").on("click", function(){
    // najdi atribut ID vybraného elementu
    chosenState = d3.select(this).attr('id');
    console.log(chosenState);
    // zvýrazni vybranou zemi
    borderIt(this);
});
```

Přístup k datům

Pro přístup k celému objektu použijeme:

```
data[poziceVPoli]
```

Pro přístup k hodnotě uložené přímo pro nějaký stát použijeme:

```
data[poziceVPoli][kodStatu]
```

Vytvoření barchartu

Opět budeme potřebovat pár proměnných a také si nastavíme svg panel pro vykreslování barchartu.

```
var t = 0; //cas
var max = 0; //nejvyssi hodnota

var pom = 2.5; // poměr natažení linky
var gh = 120; // výška linky grafu
```

```
var gw = pom*dataEntriesCount; //sirka grafu
```

```
// Nastavení canvasu grafu
var graph = d3.select("body")
    .select(".graph")
    .attr("width",gw)
    .attr("height",gh);
```

Nalezení největší hodnoty v datech

Využijeme objektový přístup, potřebujeme také testovat, jestli neporovnáváme s hodnotou datumu.

```
//nalezeni nejvyssi hodnoty
data.forEach(function(object){
    for(var key in object) {
        if (key != "date") {
            if(object[key]>max){
                max = object[key];
            }
        }
    }
});
```

Provázání barchartu s klikem na stát

Pro vykreslení grafu si vytvoříme novou funkci, která nám obslouží vytvoření barchartu. Každému rect v barchartu také nastavíme třídu, abychom pak jednoduše mohli všechny vymazat když vybereme jiný stát.

```
//vykresli graf
function graphIt(){
    graph.selectAll(".gData").remove();

    // Vykreslí graf podle výběru
    for (var i = 0; i < dataEntriesCount; i++) {
        for(var key in data[i]) {
            if (key == chosenState) {
                graph.append("rect")
                    .attr("x", pom*i)
                    .attr("y", gh-Math.ceil(data[i][key]/110)-10)
                    .attr("width", 2)
                    .attr("height", Math.ceil(data[i][key]/110))
                    .attr("class", "gData")
                    .attr("i", i)
                    .attr("fill", "white");
            }
        }
    }
}
```

Do funkce, která obsluhuje kliknutí na stát pak jen přidáme příkaz

```
graphIt();
```

Obsluha kliknutí na barchart

Podobně jako v případě kliku na stát přidáme pomocí metody on event handler na kliknutí na barchart. Pozici kliknutí si uložíme do proměnné t. Pro získání pozice použijeme metodu d3.mouse(container), která vrací pozici kliku relativní vůči danému kontejneru, v našem případě celého barchartu. Souřadnice jsou ve formě pole o dvou prvcích.

```
graph.on('click', function () {  
    t = parseInt(d3.mouse(this)[0]/pom);  
  
});
```

Do svg panelu grafu přidáme objekt linky, abychom viděli, jakou pozici v grafu máme vybranou:

```
graph.append("line")  
    .attr("x1", t*pom)  
    .attr("y1", 0)  
    .attr("x2", t*pom)  
    .attr("y2", gh + 150)  
    .attr("stroke-width", 1)  
    .attr("stroke", "silver");
```

Na pozici kde bylo kliknuto vykreslíme linku jako indikátor vybraného času:

```
function line(){  
    graph.selectAll("line")  
        .attr("x1", pom*t)  
        .attr("x2", pom*t);  
}
```

A přidáme příslušný příkaz volající tuto vykreslovací funkci do funkce obsluhující kliknutí do barchartu:

```
graph.on('click', function () {  
    t = parseInt(d3.mouse(this)[0]/pom);  
    line();  
  
});
```

Obarvení států podle lineární škály (černá- červená), přidání legendy a výpisu dat

Zkuste sami ☺ Celý funkční kód najdete na následujících stránkách.

Bylo by vhodné přidat do vizualizace ještě možnost scrollování myškou s linkou indikující čas, případně tlačítka, která by umožňovala jemnější (spojitější) přechod a zkoumání v okolí vybraného času. To už si můžete zkusit v rámci domácího studia ☺

Celý kód

```
//promenna obsahujici odkaz na data
var data;

/*nacteni dat z CSV souboru a uprava properties na kody zemi. Unarni
operator plus tam je kvuli tomu, ze to potrebujeme ulozit jako číslo
(puvodne import jako string)*/
d3.csv("data.csv", function(d) {
  return {
    date : d.Date,
    canada : +d.Canada,
    AB : +d.Alberta,
    BC : +d["British Columbia"],
    MB : +d.Manitoba,
    NB : +d["New Brunswick"],
    NL : +d["Newfoundland and Labrador"],
    NS : +d["Nova Scotia"],
    ON : +d.Ontario,
    SK : +d.Saskatchewan,
    QC : +d.Quebec
  };
}, function(d){
  //ulozeni odkazu na data
  data = d;

  // vizualizace dat
  visualization();
});

/*-----
ZACATEK VIZUALIZACE
-----*/
function visualization() {

  var dataEntriesCount = data.length; //pocet zaznamu
  var chosenState = "";

  var t = 0; //cas
  var max = 0; //nejvyssi hodnota

  var showLegend = true;

  //nalezeni nejvyssi hodnoty
  data.forEach(function(object){
    for(var key in object) {
      if (key != "date") {
        if(object[key]>max){
          max = object[key];
        }
      }
    }
  }
});
```

```

//vytvoreni promenne obsahujici odkaz na mapu
var canvas = d3.select("body").select(".map");

var pom = 2.5; // poměr natažení linky
var gh = 120; // výška linky grafu
var gw = pom*dataEntriesCount; // šířka linky grafu

// Nastavení canvasu grafu a uložení do proměnné
var graph = d3.select("body")
    .select(".graph")
    .attr("width",gw)
    .attr("height",gh);

// linka urcující jaky time span máme vybrany
graph.append("line")
    .attr("x1", t*pom)
    .attr("y1", 0)
    .attr("x2", t*pom)
    .attr("y2", gh)
    .attr("stroke-width", 1)
    .attr("stroke", "silver");

/*-----
INTERAKCE
-----*/

//event handler KLIK DO GRAFU
graph.on('click', function () {
    t = parseInt(d3.mouse(this)[0]/pom);
    line();

    d3.select("#pocetNakazenych").text("Počet nakažených: " + data[t]
[chosenState] + ", týden: " + data[t]['date']);
    setColors();
});

// event handler KLIK NA MAPU
canvas.selectAll("g").on("click", function(){

showLegendFunc();
setColors();
// atribut ID vybraného elementu uložíme do chosenState
chosenState = d3.select(this).attr('id');
console.log(chosenState);
d3.selectAll("h2").remove();
d3.select("#pocetNakazenych").remove();
d3.select("#statusID").append("h2").text(chosenState);
d3.select("#statusID").append("p").text("Počet nakažených: " + data[t]
[chosenState] + ", týden: " + data[t]['date']).attr('id',
"pocetNakazenych");
// zvýrazní vybranou zemi
borderIt(this);
graphIt(); //Vykreslí graf

```

```

});
/*-----
KONEC INTERAKCE
-----*/

/*-----
FUNKCE
-----*/
//vykresleni legendy
function showLegendFunc() {
if (showLegend) {
    //gradient z cerne do cervene
    var gradient = canvas.append("defs")
        .append("linearGradient")
        .attr("id", "gradient")
        .attr("x1", "0%")
        .attr("y1", "50%")
        .attr("x2", "100%")
        .attr("y2", "50%")
        .attr("spreadMethod", "pad");
    gradient.append("stop")
        .attr("offset", "0%")
        .attr("stop-color", "black")
        .attr("stop-opacity", 1);
    gradient.append("stop")
        .attr("offset", "100%")
        .attr("stop-color", "red")
        .attr("stop-opacity", 1);

    canvas.append("rect")
        .attr("width", 200)
        .attr("height", 20)
        .attr("x", 10)
        .attr("y", 460)
        .attr("stroke", "lightgray")
        .attr("stroke-width", 2)
        .attr("fill", "url(#gradient)");
    canvas.append("text")
        .text("Legenda")
        .attr("x", 10)
        .attr("y", 450);
    canvas.append("text")
        .text("0")
        .attr("x", 5)
        .attr("y", 500);
    canvas.append("text")
        .text(max)
        .attr("x", 180)
        .attr("y", 500);

    showLegend = false;
}
}

```



```

//nastaveni barvy
function setColors() {
    canvas.selectAll("g")
        .attr("fill", function(){
            var value = data[t][d3.select(this).attr('id')];
            value = value * 255/max;
            return "rgb(" + Math.round(value) + ", 0, 0)";
        });
}

// posune čáru urcující vybraný time span v grafu
function line(){
    graph.selectAll("line")
        .attr("x1", pom*t)
        .attr("x2", pom*t);
}

//prekreslí graf - použito na zaklade kliku do mapy
function graphIt(){
    graph.selectAll(".gData").remove();
    // Vykreslí graf podle výběru
    for (var i = 0; i < dataEntriesCount; i++) {
        for(var key in data[i]) {
            if (key == chosenState) {
                graph.append("rect")
                    .attr("x", pom*i)
                    .attr("y", gh-Math.ceil(data[i][key]/110)-10)
                    .attr("width", 2)
                    .attr("height", Math.ceil(data[i][key]/110))
                    .attr("class", "gData")
                    .attr("i", i)
                    .attr("fill", function(d) {
                        var value = data[i][key];
                        value = value * 255/max;
                        return "rgb(" + Math.round(value) + ", 0, 0)";
                    });
            }
        }
    }
}

//zvyraznení vybrané země
function borderIt(obj){
    d3.selectAll("g").style("stroke-width", 0);
    d3.select(obj).style("stroke", "yellow").style("stroke-width", 4);
}

/*-----
KONEC FUNKCI
-----*/
/*-----
KONEC VIZUALIZACE
-----*/

```

