

Rapport

SENTINELLE POLITIQUE

Projet réalisé par : Melvyn BERTOLONE p1804487, Christophe MARREL p1802696, Aly AMR OMAR LOUTFY p1806435

Encadrant : Gabriel Radanne

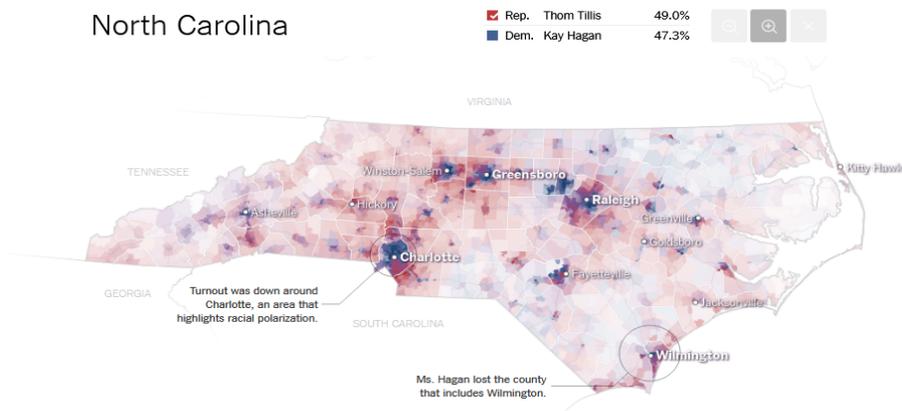
Rappel du sujet

Nous devons réaliser une visualisation de données dans le navigateur avec la bibliothèque D3.js. Le but de ce projet était de nous initier à la collecte et à la visualisation de données en utilisant D3.js. Une première phase du projet était la collecte et la mise en forme des données, la seconde était la programmation d'une visualisation interactive en javascript.

Nous avons fait le choix de faire une visualisation des élections présidentielles, notamment la visualisation d'une carte de France ainsi que d'un "diagramme" en barre pour montrer la proportionnalité des votes entre les candidats.

Inspirations

La visualisation ayant conduit à notre choix de sujet est une visualisation des élections pour le Sénat (2014) du **New York Time** 'The Most Detailed Maps You'll See From the Midterm Elections'¹.



Notre objectif est de réaliser la même chose mais cette fois avec les données des élections françaises.

Cette carte montre les différents résultats des élections Américaines dans l'État de la Caroline du Nord. Chaque couleur représente un candidat qui a été majoritaire dans une commune.

Le travail réalisé

Nous avons réalisé un site web pour la visualisation des élections présidentielles française, l'utilisation de D3 permet le traitement et l'affichage des données relatives à ces élections.



Les technologies utilisées

 **D3 (Data-Driven Document)** - d3js.org

D3.js est une bibliothèque javascript pour manipuler des documents basés sur la donnée. L'un des principaux développeurs de D3 est l'informaticien Mike Bostock. Elle permet de produire des data-visualisations dynamiques et interactives dans un navigateur web. Celle-ci est très utilisée dans le domaine professionnel.

 **React** - reactjs.org

React est une bibliothèque javascript pour la création d'interfaces utilisateur. C'est une librairie open-source, orientée Front-End et maintenue principalement par Facebook.

 **Git** - git-scm.com

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds.

La collecte des données

Pour la récupération et le traitement des données il y a deux grandes "familles" de données que l'on souhaite traiter, la première était les données relatives aux votes, la seconde était pour l'affichage de la carte (geojson).

Nos sources de données sont :

<https://www.data.gouv.fr/fr/pages/donnees-des-elections> pour les données relatives aux votes.

<https://github.com/kerneis/france-choropleth> Avec les données Geofla pour l'affichage de la carte de France.

Format des données des votes

```
{
  "Code du département": "01",
  "Libellé du département": "AIN",
  "Code de la circonscription": 1,
  "Inscrits": 79024,
  "Votants": 65819,
  "Exprimés": 61677,
  "Blancs et nuls": 4142,
  "François HOLLANDE (PS)": 27511,
  "Nicolas SARKOZY (UMP)": 34166
},
```

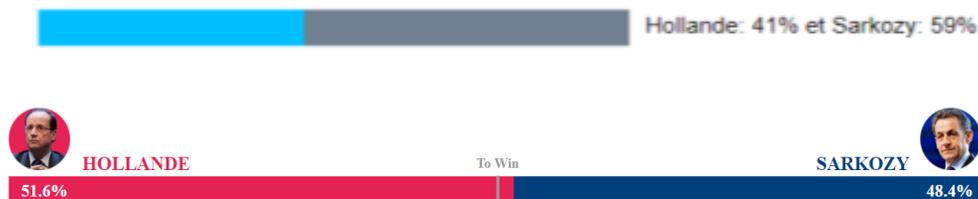
Format des données Geofla²

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "ID_GEOFLA": 1,
        "CODE_DEPT": "01",
        "NOM_DEPT": "AIN",
        "CODE_CHF": "053",
        "NOM_CHF": "BOURG-EN-BRESSE",
        "X_CHF_LIEU": 8717,
        "Y_CHF_LIEU": 65696,
        "X_CENTROID": 8814,
        "Y_CENTROID": 65582,
        "CODE_REG": "82",
        "NOM_REGION": "RHONE-ALPES"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [ 5.831226413621034, 45.938459578293212 ],
            [ 5.826316643217439, 45.903692922700827 ],
            [ 5.801109494979826, 45.845796477607138 ],
            [ 5.831226413621034, 45.938459578293212 ]
          ]
        ]
      }
    }
  ]
}
```

La légende

Toutes les cartes ont une légende associée qui permet de distinguer à quoi correspond chaque couleur. Nous avons donc décidé de faire notre légende avec le pourcentage de votes des présidents indiqué dessus.

Ci-dessous les exemples de l'avancement sur la légende inspirée du site³



On commence d'abord par créer deux rectangles, par exemple le bleu prendra une taille 100% et le gris le pourcentage du président à droite (par exemple Sarkozy dans la première capture) donc 59% dans ce cas, et le rectangle gris commence à 41% du début du rectangle bleu.

Une fois que ce concept fonctionne bien, on ajoute le bon code couleur et on le rend plus intuitif et esthétique. Au lieu d'avoir les pourcentages à droite on les place sur les rectangles, avec une barre (To Win) placée exactement au milieu.

On a pris des images en droit libre des présidents et on a placé la couleur de leur parti politique en fond.

A ce stade nous avons le nom du président, la photo, le pourcentage et les rectangles qui sont liés par des coordonnées pour que cela soit statique. Les données correspondaient pas exactement à ce que l'on voulait, nous avons donc créé un nouveau json intitulé élection qui a le format suivant :

```
{
  "Annee": "2012t2",
  "NbCandidats": "2",
  "Candidat1": {
    "Prenom": "François",
    "Nom": "HOLLANDE",
    "PartiPol": "PS",
    "CodeCouleur": "#e62154"
  },
  "Candidat2": {
    "Prenom": "Nicolas",
    "Nom": "SARKOZY",
    "PartiPol": "UMP",
    "CodeCouleur": "#00407d"
  }
},
```

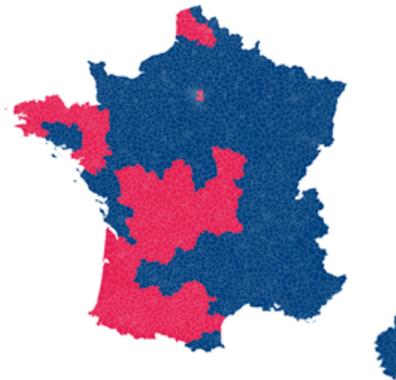
On utilise ce json pour pouvoir afficher les images, créer le nombre de bonnes cases pour les tableaux voteCand par exemple (dans ce cas 2 présidents donc 2 cases), pour l'affichage du bon code couleur.

Pour l'affichage des pourcentages de votes, nous utilisons l'année en question par exemple 2012t2.json (2012 deuxième tour).

Pour le calcul on fait une boucle forEach sur l'élément json de l'année et on fait la somme de tous les votes de chaque candidat, ensuite on divise le nombre de votes de chaque candidat par le nombre de votes total exprimés (pour avoir le bon pourcentage).

La carte de France

Nous avons également développé la visualisation de la carte de France, nous avons deux principales fonctionnalités à mettre en place, la première fut l'affichage de la carte de France (ses frontières, ses communes, etc..) au format svg.

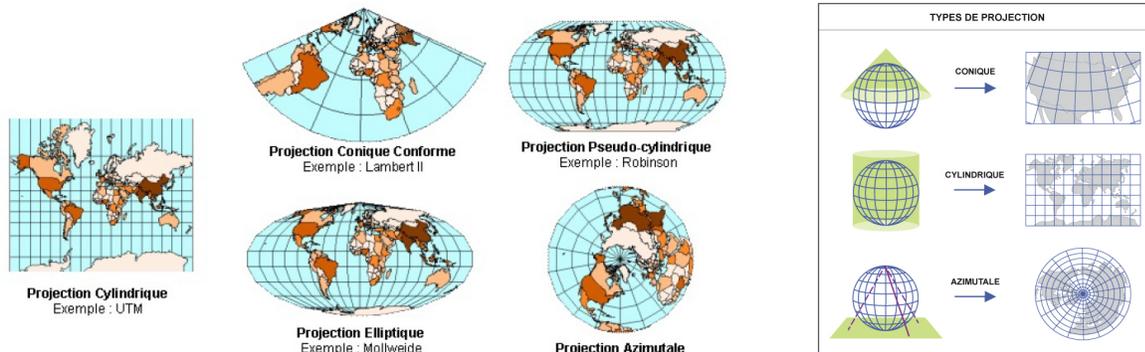


Il fallait pouvoir visualiser les frontières entre communes, départements, régions en fonction d'un formulaire envoyé par l'utilisateur (via un menu déroulant).

Pour traiter ceci, nous avons créé une classe Carte qui est une extension d'une classe Component de React. Une méthode interne (createCarte) gère la génération de la carte avec du code React.

Succinctement, pour générer une carte il faut :

- Créer une projection⁴. Une projection cartographique est le processus de transformation et représentation sur un plan des points situés sur la surface sphérique tridimensionnelle de la Terre.



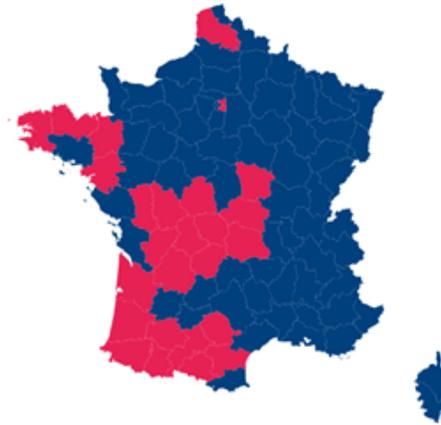
D3.js contient différentes méthodes de projections pour passer d'un format de données [longitude, latitude] à un format [x,y] : ici nous avons utilisé geoConicConformal⁵ de D3, en lui appliquant les coordonnées du centre de la France ([2.454071,46.279229]).

- Il faut ensuite générer un path, D3.js dessine avec des Path, en effet, SVG permet de créer des formes plus ou moins complexes grâce aux balises <path>, prenons un exemple : <path d="M0, 80L100, 100L200, 30L300, 50L400, 40L500, 80">, Tel que : Move(0,80), Line(100,100), Line(200,30), Line(300, 50), Line(500,80). Ce path permettra de dessiner une LineString, soit un ensemble de lignes reliées entre elles. Ici, nous utilisons la méthode geoPath de D3 en lui passant la projection créée plus haut. Pour chaque élément dans le tableau de données nous créons un path que nous ajoutons dans le conteneur <g> de la carte, le tout est contenu dans des balises <svg>.

```

▼ <svg id="svg-carte" width="1000" height="500">
  ▼ <g>
    <path id="d01"
      d="M587.0951116028284,261.65131360565465L586.8278011363603,26
      869725,261.4341944684843L586.903847648065,261.64105911728942"
      fill="#00407d"></path>
    <path id="d02"
      d="M508.8756560188227,131.05375424494423L508.42107228196284,1
      988,114.84395608617251L493.31426458728623,114.82264250074242"
      fill="#00407d"></path>
    ...
  
```

Pour finir, il fallait aussi déterminer quelle couleur afficher dans quelle régions/département (en utilisant une fonction (getCouleur), qui calcule le gagnant du département passé en paramètre et retourne la couleur de son parti politique).



Une limitation liée à la données relatives aux votes, les données ne représentait que le nombre de voix par département, il était donc impossible d'avoir une visualisation précise du vainqueur par cantons par exemple.

Ce que l'on a appris

Le fonctionnement de D3.

Pour nous lancer à l'apprentissage on a utilisé le site recommandé par le professeur chargé du projet qui est [Observablehq.com/@d3](https://observablehq.com/@d3).

Après avoir regardé un peu le code et ce que cela donne, nous avons suivi des tutoriels sur youtube pour pouvoir jouer un peu avec le code.

Le fonctionnement de React.

Cela a été une initiative personnelle, React est très utilisé dans le domaine professionnel.

Nous avons appris à passer d'un site web basique en html/css à du React, à comprendre le fonctionnement de la logique des composants.

Il y a eu aussi l'apprentissage du javascript.

Les problèmes rencontrés

D3 n'était pas une simple librairie à utiliser/comprendre donc on se sentait perdu au début. Le passage du html/css à react n'a pas été très facile aussi, avec un apprentissage d'une nouvelle technique d'agencement de site web. Il a fallu passer d'un site fonctionnel en html/css à un site react. Des problèmes sont survenus par rapport à l'affichage css qui était un peu différent à des moments, des lignes de codes qui fonctionnent sous html/css ne fonctionnent plus en React.

La grosse difficulté pendant la réalisation de la légende était les promesses. Jamais utilisé avant et pas simple à comprendre le concept. Le but dès le départ était d'avoir une fonction que l'on appelle qui permet d'obtenir la promesse au lieu de faire à chaque fois un `.then(function ..)` sur le json.

Pour régler ce problème nous avons fait le choix de créer un tableau statique recevant les données où celles-ci deviennent accessibles grâce à l'index du tableau. Une fois le tableau "fulfilled", nous pouvons assigner à des variables spécifiques les données à partir de l'index du tableau. (Par exemple une variable `Présidents` prenant les données relatives aux présidents de

l'indice 0 du tableau de promesses). Nous effectuons ensuite le traitement des données avec D3 à partir de ces variables.

La création de la carte a posé quelques problèmes, notamment le fait que nos données nous limitent à la délimitation des régions avant la loi du 16 janvier 2015. (cf. La loi n 2015-29 du 16 janvier 2015 relative à la délimitation des régions, aux élections régionales et départementales et modifiant le calendrier électoral)⁶. Il manque aussi l'affichage des DOM TOM (France d'outre-mer).

Conclusion

D'un point de vue de notre cahier de charge, ce projet est d'une réussite relative. Nous avons pu mettre en place un site avec l'option de choisir des années différentes et afficher les statistiques correspondantes.

Au niveau de la gestion du projet en équipe, nous avons réussi à bien nous répartir les tâches afin d'atteindre nos objectifs de départ, avec une ambiance de groupe très positive.

Ce projet a été très intéressant sur de nombreux points, nous avons appris de nouvelles technologies utilisables (D3, React etc...) et utilisées dans le monde professionnel. Cela nous a aussi appris ce qu'était le travail de groupe répartis sur plusieurs mois et le fait de s'organiser sur le temps long.

Nous voulions, pour finir, remercier Gabriel Radanne pour son travail de conseil et d'encadrement.

Annexe

1 - 'The Most Detailed Maps You'll See From the Midterm Elections' :

<https://www.nytimes.com/interactive/2014/11/04/upshot/senate-maps.html>

2 - Explication du format GeoJSON : <https://fr.wikipedia.org/wiki/GeoJSON>

3 - <https://theconversation.com/how-to-read-u-s-election-maps-as-votes-are-being-counted-149251>

4 - Explication sur les projections : <https://github.com/d3/d3-geo>

5 - Exemple d'utilisation de Conic Conformal : <https://observablehq.com/@d3/conic-conformal>

6 - La loi no 2015-29 du 16 janvier 2015 :

<https://www.legifrance.gouv.fr/loda/id/JORFTEXT000030109622/>