# Description du besoin

Afin d’améliorer la compétitivité des transports en commun, on voudrait visualiser la qualité des services fournis par la Société de transport de Montréal (STM) sur le territoire de l’ile de Montréal. Pour ce faire, nous allons créer une [carte thermique](https://fr.wikipedia.org/wiki/Carte_thermique) (« heatmap ») dont les couleurs indiqueront un score de « qualité du service » selon le lieu et l’heure. Par chance, [la STM rend disponible gratuitement sur son site web les données au format GTFS de l’ensemble de son réseau](https://www.stm.info/fr/a-propos/developpeurs) (arrêts, horaires, tarifs, etc.).

Pour simplifier la représentation de la carte thermique, nous allons subdiviser l’ile de Montréal en blocs de 100m par 100m (approximativement). De plus, puisque la qualité du service dépend de l’heure de la journée, nous allons subdiviser une journée de 24 heures en tranches de 15 minutes. Idéalement, ces paramètres devraient être modifiables dynamiquement dans le code du logiciel final, mais nous allons les fixer ici pour simplifier le développement du prototype.

Le calcul pour mesurer la qualité de l’offre de transport en commun au bloc dont le centre est aux coordonnées GPS [*x,y*] (en degrés) et à l’heure *t* (en minutes) est alors le suivant :

Calculer le nombre total de passages d’autobus différents[[1]](#footnote-2) pour tous les arrêts qui se trouvent dans le bloc centré en [*x,y*], au cours de la période allant de l’heure *t* à (*t + 15 minutes*).

Additionner le nombre total de passages de métro pour toute station qui se trouve dans le « super-bloc » centré en [*x,y*] (ie. tous les blocs entourant le bloc central de 100m x 100m, en plus de lui-même).

Pour générer la carte thermique à une heure *t*, il faudra donc effectuer ce calcul pour chaque bloc de 10000m2 sur l’ile de Montréal. Par la suite, nous voudrions voir cette carte changer dans le temps sur une période de 24 heures (par tranches de 15 minutes).

Le résultat final souhaité est une carte thermique animée dans le temps (par exemple, 15 minutes **→** 0.15 seconde) qui révèle visuellement la qualité du service de transport en commun sur l’ile de Montréal pour une journée donnée. Il n’est pas nécessaire que cette carte soit interactive (mettre en pause, « zoomer », etc.), mais ce serait un plus. Vous pouvez vous contenter de produire un simple fichier *.gif* ou *.png* (animé). Minimalement, une série d’images statiques correspondant à la carte générée pour chaque tranche de 15 minutes est également acceptable dans le cadre du prototype.

## Questions à poser au client :

* S’il manque des informations dans la description du besoin, inscrivez-les ici et j’y répondrai

# Technologies candidates

Ci-dessous, on veut lister les technologies qui pourraient convenir au besoin précédent, organisées selon leur catégorie.  
Vous pouvez indiquer « aucun » s’il n’est pas nécessaire de préciser le choix d’une technologie pour une catégorie donnée.

Pour chaque catégorie, vous devez remplir le tableau suivant en cherchant et documentant quelles technologies seraient des candidates potentielles afin de répondre au besoin du client.  
*Conseil : Mieux vaut proposer plus de technologies que moins, on fera le choix des technologies les plus adaptées plus tard, une fois toutes les options sur la table.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom de la technologie** | **Description sommaire** | **Avantages** | **Limitations ou incertitudes** | **Références et exemples** | **Analyse (votre jugement)** |
| **Le nom de la technologie** | Un court résumé de ce que ça fait (dans le cadre du besoin ci-haut). | Quelles sont les forces de cette technologie?   * Facilité d’apprentissage * Bonne documentation * Très populaire, établie depuis longtemps * Etc… | Quelles sont les faiblesses de cette technologie?   * Faible performance * Coût ou licence * Courbe d’apprentissage * Peu d’exemples connus * Etc… | Listez les sites, articles ou vidéos qui vous ont aidés à comprendre la technologie. Des exemples de projets similaires sont aussi très utiles. | Écrivez une ou deux phrases qui indiquent si vous pensez que cette technologie serait adéquate ou non au besoin, et pourquoi. |
| **Ce tableau n’est qu’un exemple, n’inscrivez rien ici, utilisez plutôt les tableaux des catégories spécifiques dans les prochaines pages :**  **Plateformes de déploiement (matériel, logiciels, services cloud, etc.) 2**  **Sources de données (bases de données, API web ou locale, etc.) 2**  **Langages, librairies et frameworks 2**  **Environnement de développement 2**  **Autres technologies 3** | | | | | |

## Plateformes de déploiement (matériel, logiciels, services cloud, etc.)

Quelle sorte de matériel ou quelles alternatives (services web, virtualisation, etc.) sont nécessaires pour répondre au besoin?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom de la technologie** | **Description sommaire** | **Avantages** | **Limitations ou incertitudes** | **Références et exemples** | **Analyse (votre jugement)** |
|  |  |  |  |  |  |

## Sources de données (bases de données, API web ou locale, etc.)

Si l’application aura besoin de conserver ou d’obtenir des données d’une DB ou d’une API, quelles technologies ou services pourraient les fournir?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom de la technologie** | **Description sommaire** | **Avantages** | **Limitations ou incertitudes** | **Références et exemples** | **Analyse (votre jugement)** |
|  |  |  |  |  |  |

## Langages, librairies et *frameworks*

Quels langages de programmation et quelles librairies ou *frameworks* de ces langages pourraient répondre au besoin?  
*Conseil : Le choix des librairies ou frameworks est souvent plus important que le choix du langage lui-même, donc mieux vaut commencer par là.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom de la technologie** | **Description sommaire** | **Avantages** | **Limitations ou incertitudes** | **Références et exemples** | **Analyse (votre jugement)** |
| **Leaflet** | Leaflet est une bibliothèque JavaScript de cartographie open source largement utilisée et appréciée pour sa simplicité, sa flexibilité et sa légèreté. | * Facile à utiliser pour la création de cartes interactives. * Supporte de nombreuses sources de données géospatiales. * Grande communauté et documentation riche. * Gratuit et open source. * Prise en charge multiplateforme. | * Peut être limité pour des cartes très complexes. * Les animations peuvent être limitées. * Moins adapté aux applications 3D. | leafletjs.com/examples/quick-start | Excellent choix pour 2D. |
| **OpenLayers** | Bibliothèque JavaScript open source qui permet la création de cartes interactives et de visualisations de données géospatiales. Elle offre un large éventail de fonctionnalités pour la gestion des cartes et des données géospatiales. | * Riche en fonctionnalités et hautement personnalisable. * Prise en charge de nombreuses sources de données géospatiales. * Active et en constante évolution. | * Courbe d'apprentissage abrupte pour les débutants. * La personnalisation avancée peut nécessiter une expertise. | openlayers.org/en/latest/examples | Excellente alternative à Leaflet pour les projets nécessitant une personnalisation approfondie et une gestion complexe des données géospatiales. |
| **Mapbox GL JS** | Bibliothèque JavaScript de cartographie développée par Mapbox. Elle permet de créer des cartes interactives hautement personnalisables | * Facilité d'utilisation pour la création de cartes interactives élégantes. * Intégration transparente avec les services de cartographie de Mapbox. * Prise en charge de la visualisation 3D. | * Certaines fonctionnalités avancées peuvent nécessiter un abonnement payant à Mapbox. * Moins de flexibilité par rapport à OpenLayers pour les sources de données personnalisées. | docs.mapbox.com/help/tutorials/?product=Mapbox+GL+JS | idéal lorsque vous avez besoin de cartes interactives modernes et esthétiques |
| **Cesium** | Bibliothèque JavaScript de cartographie 3D qui se concentre sur la visualisation géospatiale en trois dimensions | * Prise en charge de la visualisation 3D et de la cartographie terrestre, aérienne et spatiale. * Grande flexibilité pour la visualisation de données géospatiales complexes. * Conçu pour des applications nécessitant une dimension verticale. | * Spécialisé dans la visualisation 3D, ce qui peut ne pas convenir à tous les types de projets cartographiques. * Courbe d'apprentissage pour la création de visualisations 3D. | sandcastle.cesium.com | La meilleure option lorsque vous avez besoin de la visualisation 3D |
| **D3.js** | D3 (ou D3.js) est une bibliothèque JavaScript gratuite et open source permettant de visualiser des données. Son approche bas niveau construite sur les standards du Web offre une flexibilité inégalée dans la création dynamique. | * Excellente pour la création de visualisations personnalisées. * Permet des animations complexes. * Grande flexibilité. | * Courbe d'apprentissage abrupte. * Peut être surdimensionné pour ce projet. | https://observablehq.com/@d3/gallery | La meilleure option |
| **Highcharts** | Bibliothèque JavaScript de visualisation de données qui se concentre sur la création de graphiques interactifs et de tableaux de bord. Elle offre une variété de graphiques prêts à l'emploi et une API conviviale. | * Facilité d'utilisation avec une API simple. * Large gamme de types de graphiques préconstruits. * Documentation détaillée et support de la communauté. * Bonne prise en charge des graphiques en temps réel. | * Licence payante pour certains cas d'utilisation commerciale. * Moins flexible pour des visualisations personnalisées. | https://www.highcharts.com/demo | Excellente option si on est besoin de créer rapidement des graphiques interactifs et des tableaux de bord sans vous soucier de la complexité de D3.js |
| **Chart.js** | Bibliothèque JavaScript de création de graphiques simples et réactifs. Elle se concentre sur les graphiques de base tels que les diagrammes à barres, les diagrammes circulaires et les graphiques en ligne. | * Facilité d'utilisation avec une syntaxe simple. * Idéal pour les graphiques de base. * Léger et rapide à mettre en place. * Licence open source (MIT). | * Moins adapté aux visualisations de données complexes. * Moins de fonctionnalités avancées par rapport à D3.js. | https://www.chartjs.org/docs/latest/samples/information.html | Excellent choix lorsque vous avez besoin de créer rapidement des graphiques de base sans la complexité de D3.js. Il est idéal pour les projets nécessitant des graphiques simples et réactifs, mais il peut être limité pour des visualisations de données plus avancées. |
| **Vega-Lite** | Bibliothèque de visualisation de données déclarative qui simplifie la création de visualisations complexes en utilisant une syntaxe JSON. Elle repose sur Vega, qui est un système de création de visualisations de données encore plus puissant. | * Création de visualisations déclarative avec une syntaxe JSON. * Prise en charge de nombreuses visualisations de données avancées. * Personnalisable et extensible. * Licence open source (BSD). | * Courbe d'apprentissage pour comprendre la syntaxe JSON. * Peut nécessiter des connaissances en JSON et en manipulation de données. | https://vega.github.io/vega-lite/examples/ | Alternative solide à D3.js lorsque on est besoin de créer des visualisations de données avancées tout en évitant la complexité de la création manuelle de graphiques avec D3.js. Cependant, il nécessite une compréhension de la syntaxe JSON, ce qui peut être une courbe d'apprentissage pour certains utilisateurs. |
| **PostGIS** | PostGIS est un logiciel open source qui ajoute la prise en charge des objets géographiques à la base de données objet-relationnelle PostgreSQL. PostGIS suit la spécification Simple Features for SQL de l'Open Geospatial Consortium. Techniquement, PostGIS a été implémenté en tant que logiciel externe PostgreSQL. | * Gère efficacement les données géospatiales. * Possibilité de requêtes spatiales complexes. * Intégration avec PostgreSQL. * Licence open source. * Communauté active. | * Nécessite des compétences en gestion de bases de données. * Surdimensionné pour un prototype simple. * L'installation et la configuration de PostGIS peuvent être plus complexes que celles d'une base de données non spatiale. | https://postgis.net/documentation/getting\_started/ | PostGIS est une solution de base de données géospatiale exceptionnelle pour les projets qui nécessitent des fonctionnalités géospatiales avancées, des performances élevées et une intégration transparente avec PostgreSQL. C'est particulièrement adapté aux applications cartographiques, SIG (Systèmes d'Information Géographique), et à la gestion de données géospatiales complexes. Cependant, son utilisation peut nécessiter une expertise pour tirer pleinement parti de ses capacités. Pour les projets nécessitant une base de données géospatiale robuste, PostGIS reste un choix solide et fiable. |
| **MySQL Spatial** | MySQL Spatial est une extension du système de gestion de base de données MySQL qui permet de stocker, gérer et interroger des données géospatiales. Elle prend en charge les données vectorielles et propose des fonctionnalités de requêtage spatial. | * Intégration avec MySQL, une base de données largement utilisée. * Bonne performance pour des volumes de données moyens. * Prise en charge du langage SQL standard. | * Moins de fonctionnalités géospatiales avancées par rapport à PostGIS. * Moins adapté aux applications nécessitant des performances très élevées. | https://dev.mysql.com/doc/refman/8.0/en/spatial-types.html | MySQL Spatial est une option solide si vous utilisez déjà MySQL et que vous avez besoin de fonctionnalités géospatiales de base. Cependant, pour des projets nécessitant des capacités géospatiales avancées, PostGIS offre plus de fonctionnalités. |
| **Oracle Spatial and Graph** | Oracle Spatial and Graph est une extension de la base de données Oracle qui permet de gérer des données géospatiales et graphiques. Elle offre des capacités avancées pour la gestion de données géospatiales et de réseau. | * Prise en charge de bases de données de grande taille et de performances élevées. * Fonctionnalités avancées pour les requêtes spatiales. * Intégration avec d'autres produits Oracle. | * Coût élevé de licence pour les produits Oracle. * Courbe d'apprentissage pour les utilisateurs non familiers avec Oracle. | https://www.oracle.com/database/technologies/bigdata-spatialandgraph.html | Oracle Spatial and Graph est une solution robuste pour les entreprises qui utilisent déjà Oracle comme base de données principale et qui ont besoin de fonctionnalités géospatiales avancées. Cependant, le coût élevé de licence peut être un obstacle pour de nombreuses organisations. |
| **SpatiaLite** | SpatiaLite est une extension géospatiale pour SQLite, un système de gestion de base de données léger. Il offre des fonctionnalités de gestion de données géospatiales en utilisant la base de données SQLite. | * Facilité d'intégration avec des applications légères. * Base de données légère et portable. * Prise en charge des fonctionnalités géospatiales de base. | * Moins adapté aux applications nécessitant des performances élevées. * Moins de fonctionnalités avancées par rapport à PostGIS. | https://www.gaia-gis.it/fossil/libspatialite/index | SpatiaLite est idéal pour les projets qui nécessitent des fonctionnalités géospatiales de base dans un environnement léger et portable. Il est particulièrement utile pour les applications embarquées ou les projets nécessitant une petite empreinte mémoire. Cependant, il peut ne pas être adapté aux applications avec de gros volumes de données ou nécessitant des performances élevées. |
| **Apache Kafka** | Apache Kafka est une plate-forme de streaming de données open source qui est largement utilisée pour la gestion des flux de données en temps réel. Elle est conçue pour la diffusion de données à grande échelle et la mise en œuvre de l'architecture des microservices. | * Gère efficacement les flux de données en temps réel. * Évolutif pour des volumes de données importants. * Peut être utilisé pour la gestion de données d'autobus et de métro. * Kafka s'intègre facilement avec d'autres technologies telles que Apache Spark, Apache Flink, et Hadoop. | * Complexité de configuration. * Surdimensionné pour un prototype simple. * Peut utiliser des ressources importantes en termes de stockage. | https://kafka.apache.org/documentation.html#gettingStarted | Kafka est La meilleure option en matière de streaming de données à grande échelle. |
| **RabbitMQ** | Apache Pulsar est une plate-forme de streaming de données open source qui offre une haute extensibilité et une faible latence. Elle prend en charge la persistance et la réplication des données. | * Haute extensibilité et faible latence. * Gestion des données en temps réel. * Bonne intégration avec des outils Apache tels que Flink et Spark. | * Peut avoir une courbe d'apprentissage pour les utilisateurs novices. * Moins de support et de documentation par rapport à Kafka. | https://www.rabbitmq.com/getstarted.html |  |
| **NATS Streaming** | NATS Streaming est une extension de NATS, un système de messagerie léger. Il offre des fonctionnalités de streaming de données pour le traitement en temps réel. | * Extrêmement léger et rapide. * Facilité d'utilisation et de configuration. * Idéal pour les cas d'utilisation nécessitant une faible latence. | * Moins de fonctionnalités avancées par rapport à Kafka. * Moins adapté aux cas d'utilisation nécessitant une grande persistance des données. | https://nats.io/ |  |
| **Plotly** |  | * Facilité d'utilisation : * Interactivité : Plotly permet de créer des graphiques interactifs avec des fonctionnalités telles que le zoom. * Support multi-langages : Python, R et JavaScript. * Intégration avec des frameworks web : frameworks tels que Dash. | * Licence et coûts * Courbe d'apprentissage pour les fonctionnalités avancées. * Dépendance à une connexion Internet | https://plotly.com/ | Plotly est une excellente bibliothèque pour la création de visualisations interactives et sa facilité d'utilisation en fait un choix populaire, en particulier pour les débutants en visualisation de données. Cependant, son modèle de licence peut entraîner des coûts |
| **Matplotlib** | Matplotlib est une bibliothèque Python de création de graphiques statiques et interactifs. Elle est bien établie dans la communauté Python et offre de nombreuses fonctionnalités de personnalisation. | * Gratuit et open source. * Excellente personnalisation des graphiques. * Prend en charge Python. | * Moins d'interactivité par rapport à Plotly. * Courbe d'apprentissage pour les utilisateurs débutants. | https://matplotlib.org/stable/gallery/index.html |  |
| **Tableau** | Tableau est une suite de logiciels de visualisation de données et de business intelligence. Elle permet de créer des tableaux de bord interactifs et de publier des visualisations. | * Facilité d'utilisation pour la création de tableaux de bord interactifs. * Prise en charge de la connexion à diverses sources de données. * Convient aux utilisateurs non techniques. | * Solution payante avec des coûts de licence. * Moins de flexibilité par rapport à Plotly pour la personnalisation des visualisations. | https://www.tableau.com/ |  |
| **React** | React est une bibliothèque JavaScript open source développée par Facebook, qui est largement utilisée pour la création d'interfaces utilisateur interactives et réactives. Elle est particulièrement adaptée au développement d'applications web à grande échelle. | * Réutilisation des composants. * Virtual DOM. * Une vaste communauté de développeurs. * Adoption par de grandes entreprises. | * Courbe d'apprentissage. * Les développeurs doivent suivre les mises à jour et les nouvelles pratiques pour rester à jour. * Composants non contrôlés | https://react.dev/learn | React est un choix solide pour le développement d'interfaces utilisateur réactives et interactives, en particulier pour les grandes applications. |
| **Angular** | Angular est un framework de développement web open source développé par Google. | * Structure complète pour le développement web. * Prise en charge de la création d'applications monopages (SPA). * Forte intégration avec TypeScript. | * Courbe d'apprentissage raide pour les débutants. * Plus lourd que React en termes de taille du bundle. | https://angular.io/start |  |
| **Vue.js** | Vue.js est un framework de développement web progressif et évolutif. Il est apprécié pour sa simplicité et sa flexibilité, ce qui le rend idéal pour les petites et grandes applications. | * Facilité d'apprentissage et de prise en main. * Bonne performance avec un faible coût d'entrée. * S'adapte bien à des projets de toute taille. | * Moins d'outils et de bibliothèques tierces par rapport à React. * Communauté plus petite que celle de React. | https://vuejs.org/guide/quick-start.html |  |
| **Ember.js** | Ember.js est un framework de développement web open source qui met l'accent sur la convention plutôt que la configuration. Il offre une structure solide pour les applications web. | * Structure robuste et bien définie pour les applications web. * Conventions strictes facilitant le développement en équipe. * Forte intégration de l'outillage pour une expérience de développement fluide. | * Courbe d'apprentissage plus raide en raison de conventions strictes. * Moins de flexibilité que certains autres frameworks. | https://guides.emberjs.com/release/getting-started/quick-start/ |  |
| **Python** | Python est un langage de programmation de haut niveau, interprété et polyvalent, largement utilisé dans le développement de logiciels, la science des données, l'automatisation, la web development, et bien plus encore. | * Lisibilité et simplicité de la syntaxe. * Vaste écosystème de bibliothèques et de frameworks. * Multiplateforme. * Grande communauté et support actif. | * Performance relativement plus faible. * Gère automatiquement la gestion de la mémoire. * Compatibilité entre les versions. | https://docs.python.org/3/tutorial/index.html | Python est un langage de programmation polyvalent très populaire, adapté à un large éventail de domaines |
| **Java** | Java est un langage de programmation polyvalent et orienté objet qui est utilisé pour développer des applications web, mobiles, d'entreprise et embarquées. | * Haute performance grâce à la compilation en bytecode. * Forte présence dans le développement d'applications d'entreprise. * Grande communauté et de nombreuses bibliothèques. | * Syntaxe plus verbeuse que Python. * Courbe d'apprentissage plus raide pour les débutants. | https://www.java.com/en/ |  |
| **JavaScript** | JavaScript est principalement utilisé pour le développement web front-end et back-end. Il est également utilisé pour le développement d'applications mobiles et de bureau. | * Large adoption dans le développement web. * Utilisé pour le développement front-end avec HTML et CSS. * Écosystème Node.js pour le développement back-end. | * Asynchrone par nature, ce qui peut entraîner une complexité dans le code. * Typage dynamique peut causer des erreurs difficiles à repérer. | https://www.javascript.com/ |  |
| **Ruby** | Ruby est un langage de programmation interprété, principalement utilisé pour le développement web grâce au framework Ruby on Rails. | * Syntaxe élégante et concise. * Ruby on Rails facilite le développement web rapide. * Grande convivialité pour les débutants. | * Performance inférieure à celle de langages compilés. * Moins d'applications dans des domaines autres que le développement web. | https://www.ruby-lang.org/en/ |  |

## Environnement de développement

Pour développer le logiciel répondant au besoin, quels outils ou processus de développement seraient utiles ou nécessaires?  
Certaines plateformes nécessitent parfois l’utilisation de logiciels spécifiques pour le développement d’applications; Si c’est le cas, il faut les identifier ici.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom de la technologie** | **Description sommaire** | **Avantages** | **Limitations ou incertitudes** | **Références et exemples** | **Analyse (votre jugement)** |
|  |  |  |  |  |  |

## Autres technologies

Y a-t-il d’autres catégories de technologies qui sont nécessaires ou utiles pour répondre au besoin?  
Par exemple, des logiciels ou applications existantes qui couvrent déjà la majorité des fonctionnalités répondant au besoin.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom de la technologie** | **Description sommaire** | **Avantages** | **Limitations ou incertitudes** | **Références et exemples** | **Analyse (votre jugement)** |
|  |  |  |  |  |  |

# Choix et justification des technologies

Indiquez ci-dessous quelles technologies vous avez décidé d’utiliser pour réaliser le projet décrit en haut.  
*Note : Il est possible que vous changiez d’avis lors de la mise à l’essai, et c’est tout à fait correct; Les choix fait ici sont seulement provisoires.*

* (Quelles technologies utiliser et pourquoi? – en bref, résumez ici le contenu de la colonne « analyse »)

# Recommandations:

Pour ce projet, une approche basée sur Python et JavaScript semble être appropriée. Voici notre recommandation :

- Utiliser Python pour la manipulation des données GTFS, le calcul de la qualité de service, et la création d'images statiques de la carte thermique.

- Utiliser React.js pour créer une interface utilisateur permettant de visualiser la carte thermique animée.

- Utiliser Leaflet ou D3.js pour la représentation graphique de la carte thermique.

- Utiliser Plotly pour les graphiques temporels.

# Proposition de prototype

Prototype 1 : Utilisation de Python avec Pandas et Ploty

- Charge les données GTFS de la STM avec Pandas.

- Effectue le calcul de la qualité du service pour chaque bloc de 100m x 100m à chaque intervalle de 15 minutes sur une journée.

- Crée des images statiques représentant la carte thermique pour chaque intervalle de temps.

- Crée une interface utilisateur simple pour afficher ces images statiques dans une séquence animée ou permet de sélectionner un intervalle spécifique.

- Utilise Plotly pour afficher des graphiques temporels pour une évaluation plus détaillée de la qualité de service.

Prototype 2 : Utilisation de Python avec Pandas et Matplotlib

Extraction de données GTFS : Utilisez Python avec la bibliothèque pandas pour extraire et manipuler les données GTFS.

Calcul de la qualité du service : Implémentez l'algorithme de calcul de la qualité du service en Python.

Création de la carte thermique : Utilisez Matplotlib pour générer une carte thermique statique sous forme d'image.

Automatisation de la mise à jour temporelle : Écrivez un script Python pour mettre à jour la carte toutes les 15 minutes.

Options de visualisation : Matplotlib peut fournir des options de zoom et d'interaction de base pour explorer la carte.

Prototype 3 : Utilisation de JavaScript avec Leaflet et D3.js

Extraction de données GTFS : Utilisez JavaScript pour extraire et traiter les données GTFS.

Calcul de la qualité du service : Implémentez l'algorithme en JavaScript.

Création de la carte thermique : Utilisez Leaflet pour créer une carte interactive.

Automatisation de la mise à jour temporelle : Écrivez du code JavaScript pour mettre à jour la carte toutes les 15 minutes.

Options de visualisation : Leaflet offre de nombreuses fonctionnalités d'interaction pour zoomer et explorer la carte, tandis que D3.js peut être utilisé pour des visualisations plus avancées si nécessaire.

Prototype 4 : Utilisation de Tableau

Extraction de données GTFS : Utilisez les capacités de connexion de Tableau pour extraire et préparer les données GTFS.

Calcul de la qualité du service : Utilisez les fonctionnalités de calcul de Tableau pour mettre en œuvre l'algorithme de qualité du service.

Création de la carte thermique : Tableau permet de créer des visualisations interactives, y compris des cartes thermiques.

Automatisation de la mise à jour temporelle : Utilisez les fonctionnalités de planification de Tableau pour mettre à jour la carte toutes les 15 minutes.

Options de visualisation : Tableau offre une interface utilisateur conviviale pour explorer les données et la carte thermique.

1. Si plus d’un arrêt de la même ligne d’autobus (et allant dans la même direction) se trouvent dans un même bloc, il ne faut pas les compter comme deux passages différents d’autobus, puisque c’est le même autobus qui fera les deux arrêts. Cependant, avec des blocs de 100x100m2, cette situation devrait être très rare et vous pouvez l’ignorer dans votre prototype. [↑](#footnote-ref-2)