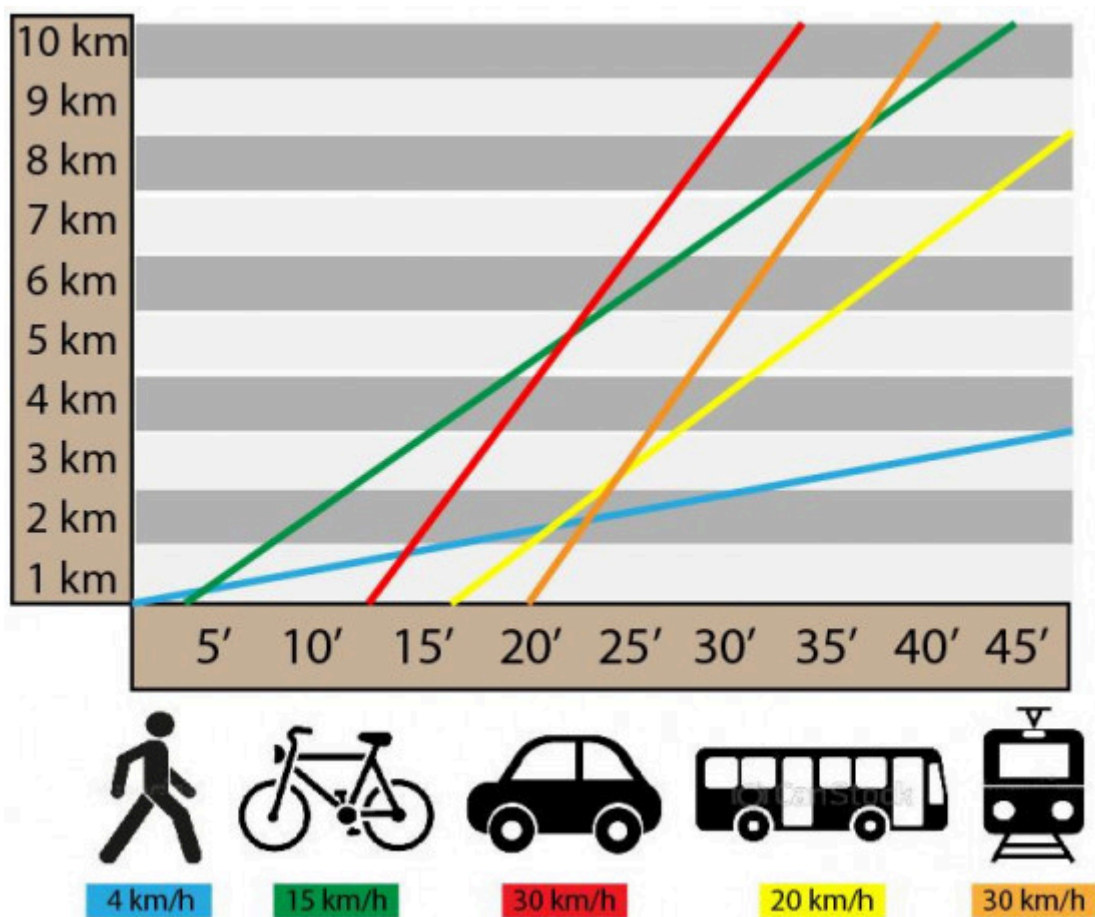


Estimation du temps de trajet domicile-travail en fonction de la distance



Pourquoi ce projet ?

Comme je suis en train d'apprendre R et que je veux intégrer un master en systèmes d'information, je me suis dit que ce serait une bonne idée de faire un projet simple mais concret. J'ai donc décidé de **modéliser le temps de trajet** en fonction de la distance, de l'heure de départ et du type de transport.

L'objectif était de voir si je pouvais prédire combien de temps ça prendrait pour aller **de chez moi à la fac (Paris 1, 5e arrondissement)** et **de chez moi au travail (Carrefour Gennevilliers)** en fonction des conditions du trajet.

Recherche des distances et temps de trajet sur Google Maps

Avant de coder, il me fallait des données. J'ai donc fait des recherches sur **Google Maps** pour estimer les temps de trajet selon le mode de transport et l'heure de départ.

1 De Val d'Argenteuil à Paris 1 (Université Panthéon-Sorbonne)



Départ : Gare de Val d'Argenteuil



Arrivée : Université Paris 1, 12 Place du Panthéon, 75005 Paris

En transport en commun :

- **Départ à 8h00** (Heure de pointe) → **50 minutes**
- **Départ à 10h00** → **40 minutes**

En voiture :

- **Départ à 8h00** → **50-60 minutes (bouchons)**
- **Départ à 10h00** → **35-40 minutes**



Observation : En voiture, le temps varie beaucoup selon l'heure. Aux heures de pointe, ce qui pose problème à cause des bouchons, donc **l'heure de départ a un vrai impact**.

2 De Val d'Argenteuil à Carrefour Gennevilliers (travail)



Départ : Gare de Val d'Argenteuil



Arrivée : Carrefour Gennevilliers, 92230

En transport en commun :

- **Départ à 13h00** → **25 minutes**
- **Départ à 15h00** → **15 minutes**

En voiture :

- **Départ à 13h00** → **20-25 minutes**

- **Départ à 15h00 → 10-15 minutes**

💡 **Observation** : Pour ce trajet, l'impact de l'heure est plus faible, sauf en cas d'accident ou de bouchon imprévu.

Mise en place du dataset dans R

Une fois que j'avais récupéré ces données, j'ai créé un **dataframe** dans R pour stocker tout ça :

```
trajets <- data.frame(
```

```
  Distance_km = c(22, 22, 9, 9, 15, 18, 12, 20),
  Temps_min = c(50, 40, 25, 15, 35, 38, 30, 45),
  Heure_depart = c(8, 10, 13, 15, 9, 11, 14, 7),
  Transport = c("Transports en commun", "Voiture", "Transports en commun", "Voiture",
    "Transports en commun", "Voiture", "Transports en commun", "Voiture")
```

Distance (km)	Temps (min)	Heure de départ	Transport	Destination
22	50	8	Transports en commun	Fac
22	40	10	Voiture	Fac
9	25	13	Transports en commun	Travail
9	15	15	Voiture	Travail
15	35	9	Transports en commun	Fac
18	38	11	Voiture	Fac
12	30	14	Transports en commun	Travail
20	45	7	Voiture	Fac

💡 Pourquoi ces colonnes ?

- **Distance_km** → J'ai mis les distances entre les lieux en km.
- **Temps_min** → C'est le temps de trajet que j'ai trouvé sur Google Maps.
- **Heure_depart** → C'est l'heure à laquelle je pars, pour voir si ça change le temps de trajet.
- **Transport** → J'ai noté si c'était en **transports en commun** ou en **voiture**, car ça impacte beaucoup le temps.

💡 **Observation** : nous pouvons constater une différence pour un même lieu dans la colonne distance cela s'explique par les différents itinéraires emprunter google maps nous donne à chaque fois le l'itinéraire le plus efficace en fonction de l'heure car le trafic varie et influe sur le temps de trajet

Visualisation des données

Après avoir créé le dataset, je voulais voir la relation entre **distance et temps de trajet**. J'ai donc utilisé ggplot2 pour tracer un graphique :

```
ggplot(trajets, aes(x = Distance_km, y = Temps_min, color = Transport)) +  
  geom_point(size = 3) +  
  geom_smooth(method = "lm", se = FALSE) +  
  labs(title = "Temps de trajet en fonction de la distance",  
        x = "Distance (km)",  
        y = "Temps (min)") +
```

Explication :

- `ggplot(trajets, aes(...))` → On utilise `ggplot()` avec le dataset `trajets`.
- `aes(x = Distance_km, y = Temps_min, color = Transport)` → On définit :
 - `x = Distance_km` (axe horizontal) → Distance du trajet en km.
 - `y = Temps_min` (axe vertical) → Temps du trajet en minutes.
 - `color = Transport` → On attribue une couleur différente selon le mode de transport ("Voiture" ou "Transports en commun").
- `geom_point()` → Ajoute les points de données sur le graphique.
- `geom_smooth(method = "lm")` → Ajoute une courbe de régression linéaire pour voir la tendance.
- `se = false`

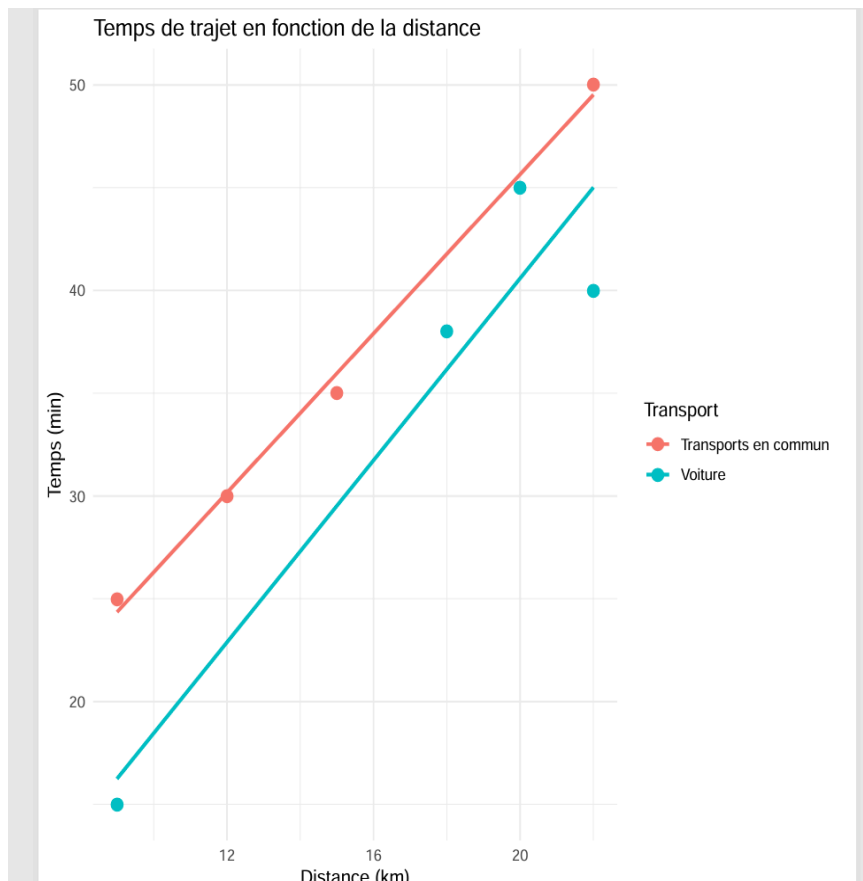
l'intervalle de confiance : C'est **une marge d'erreur** autour d'une estimation. Il nous dit :

"On est sûr à X% que la vraie valeur est dans cette fourchette."

Par exemple, si un sondage dit que **60%** des gens aiment le café avec un **intervalle de confiance de $\pm 5\%$** , ça veut dire :

: **On est sûr à 95%** que le vrai pourcentage est **entre 55% et 65%**.

représentations graphique des résultats obtenues avec le premier dataframe :



Modélisation linéaire du temps de trajet

L'étape suivante, c'était d'essayer de **prédire le temps de trajet** avec une régression linéaire.

La régression linéaire et son application à mon projet

C'est quoi une régression linéaire ?

Bon, pour faire simple, la régression linéaire, c'est une méthode mathématique qui permet de prédire une valeur en fonction d'une autre. En gros, si on a deux variables (par exemple, la distance et le temps de trajet), on cherche à voir s'il y a une relation entre elles.

On peut représenter ça par une **équation de droite** :

$$Y = aX + b$$

Où :

- **Y** est la variable qu'on veut prédire (le temps de trajet).
- **X** est la variable explicative (la distance en km).
- **a** est la pente de la droite, qui indique comment **Y varie en fonction de X**.

- **b** est l'ordonnée à l'origine, c'est-à-dire la valeur de **Y quand X = 0**.

L'idée, c'est de trouver les meilleures valeurs de **a** et **b** pour que la droite colle au mieux à nos données réelles.

Comment j'ai appliqué ça à mon projet ?

Dans mon projet, j'ai cherché à modéliser le **temps de trajet domicile-travail** en fonction de plusieurs variables :

- **La distance entre les lieux (Val d'Argenteuil → Paris 1 / Carrefour Gennevilliers)**
- **L'heure de départ (pour voir si c'est plus long aux heures de pointe)**
- **Le type de transport utilisé (RER, métro, voiture, etc.)**

Données récupérées :

J'ai utilisé **Google Maps** pour récupérer les temps de trajet moyens. Voici quelques exemples que j'ai trouvés :

- **Val d'Argenteuil → Paris 1 (Panthéon)** : 45-55 min en RER + métro
- **Val d'Argenteuil → Carrefour Gennevilliers** : 20-30 min en voiture
- **Paris 1 → Carrefour Gennevilliers** : 40-50 min en transports

Avec ces données, j'ai construit un **tableau de données** où j'ai noté la distance et le temps de trajet correspondant.

La mise en place du modèle en R

J'ai utilisé **R** pour créer un modèle de régression linéaire. Voici comment j'ai fait :

```
modele <- lm(Temps_min ~ Distance_km + Heure_depart, data = trajets)
```

Explication :

`lm(Temps_min ~ Distance_km + Heure_depart, data = trajets)` → Dit à R d'essayer de trouver une relation entre le temps de trajet (**Temps_min**), la distance (**Distance_km**) et l'heure de départ (**Heure_depart**).

`summary(modele)` → Donne les résultats du modèle (coefficients, qualité de la prédiction, etc.)

on obtient :

Call:

```
lm(formula = Temps_min ~ Distance_km + Heure_depart, data = trajets)
```

Residuals:

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

3.2933	-3.4557	1.8794	-4.8695	-1.7574	0.9263	4.9376	-0.9540
--------	---------	--------	---------	---------	--------	--------	---------

Coefficients: Estimate Std. Error t value Pr(>|t|)

(Intercept) 33.5502 17.5295 1.914 0.1138

Distance_km 1.1891 0.5028 2.365 0.0643 .

Heure_depart -1.6255 0.9424 -1.725 0.1452 ---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.982 on 5 degrees of freedom

Multiple R-squared: 0.9103, Adjusted R-squared: 0.8744

F-statistic: 25.36 on 2 and 5 DF, p-value: 0.002411

explications des résultats obtenus le

1. Résidus ("Residuals")

Les résidus sont les différences entre les valeurs réelles du temps de trajet et les valeurs prédites par le modèle.

Certains résidus sont positifs c'est à dire que le modèle sous-estime le temps réel, d'autres sont négatifs le modèle surestime. La taille de ses écarts donne une idée de la précision du modèle.

Coefficients de la régression ("Coefficients")

Chaque variable a un Estimate, qui correspond au coefficient du modèle :

Intercept (33.5502) : Quand la distance et l'heure de départ sont à 0, le temps estimé est d'environ 33.55 minutes (cela n'a pas de sens)

- Distance_km (1.1891) : Chaque kilomètre supplémentaire augmente le temps de trajet d'environ 1.19 minutes, ce qui est logique.
- Heure_depart (-1.6255) : Un coefficient négatif indique que plus l'heure de départ est élevée, plus le trajet est court. Cela pourrait s'expliquer par une diminution du trafic à certaines heures.

3. Significativité des coefficients ("Pr(>|t|)")

Ces valeurs indiquent si chaque variable a un impact significatif sur le temps de trajet.

- Distance_km (p = 0.0643) : Assez proche de 0.05, donc cette variable est presque significative (peut influencer le temps de trajet).
- Heure_depart (p = 0.1452) : Au-dessus de 0.05, donc pas significative au seuil classique (pas de preuve forte que l'heure de départ affecte le temps de trajet).
- Seule la distance semble influencer modérément le temps de trajet. L'heure de départ n'a pas d'effet significatif ici.

4 Qualités du modèle

Residual standard error: 3.982 → La moyenne des écarts entre les valeurs réelles et prédites est d'environ 4 minutes.

Prédire un trajet futur

J'ai voulu tester mon modèle en demandant **combien de temps je mettrais si je partais à 7h du matin**

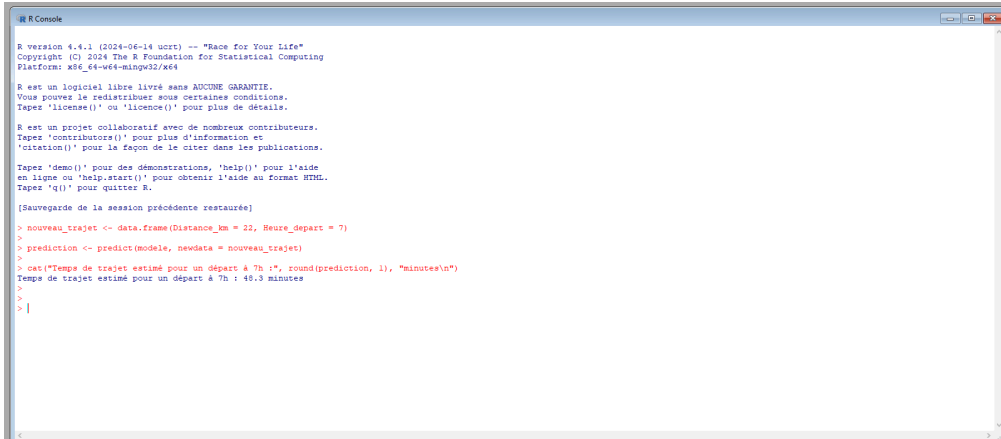
```
nouveau_trajet <- data.frame(Distance_km = 22, Heure_depart = 7)
```

```
prediction <- predict(modele, newdata = nouveau_trajet)
```

```
cat("Temps de trajet estimé pour un départ à 7h :", round(prediction, 1), "minutes\n")
```


Explication :

- J'ai créé une nouvelle donnée où je pars à 7h.
- ici on utilise le même modèle de régression créé précédemment
- `round(prediction, 1)` arrondit la valeur de la prédiction à une seule décimale.
- `prediction` stocke la valeur estimée. C'est grâce à cette ligne que le modèle "réfléchit" et te donne une réponse



```
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'licence()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contribuors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

[Sauvegarde de la session précédente restaurée]

> nouveau_trajet <- data.frame(Distance_km = 22, Heure_depart = 7)
>
> prediction <- predict(modèle, newdata = nouveau_trajet)
>
> cat("Temps de trajet estimé pour un départ à 7h :", round(prediction, 1), "minutes\n")
Temps de trajet estimé pour un départ à 7h : 45.3 minutes
>
> |
```

sources :

<https://swirlstats.com/students.html>

<https://www.youtube.com/watch?v=CCzGRyO2CTc&t=2019s>

https://www.youtube.com/watch?v=6wWI_IQeXxg

<https://www.r-bloggers.com/>

<https://larmarange.github.io/guide-R/analyses/regression-lineaire.html>

<https://larmarange.github.io/guide-R/analyses/regression-lineaire.html>
<https://larmarange.github.io/guide-R/analyses/regression-lineaire.html>