# Pourquoi avons-nous choisi ce jeu de données :

Nos critères principaux pour ce projet étaient d’analyser un jeu de données qui ne contenait peu de textes, qui ne traitait pas de données économiques et qui traitait d’un sujet original. On entend par original, un sujet qui n’avait pas trop été analysé et donc la plupart des conclusions avec déjà été tirés. En faisant plusieurs recherches, nous sommes tombés sur le site web [www.kaggle.com](http://www.kaggle.com) qui contenait plusieurs jeux de données intéressants sur des sujets très variés. Le jeu de données sur les crashs d’avions nous a paru très intéressant. Il répondait à tous nos critères et il était riche en information sur les crashs. C’est donc pour ces raisons que nous avons choisi ce jeu données.

# Description du jeu de données:

Le jeu de données couvre 5268 crashs d’avions qui ont eu lieu entre 1908 et 2009.

Pour des raisons inconnues, mis à part un crash en 1908, notre fichier de données n’a enregistré aucun crash entre 1908 et 1912.

Le jeu de données inclut les informations suivantes concernant les crashs :

* La variable « Date » affiche la data du crash dans le jeu de données
* La variable « Time » contient l’heure à laquelle le crash a eu lieu
* La variable « Location » représente l’endroit où le crash a eu lieu
* La variable « Operator » identifie l’opérateur
* La variable « Flight.. » identifie le numéro de vol
* La variable « Route » dans le jeu de données affiche le parcours qui était supposé être parcouru lors du vol
* La variable identifié par « Type » affiche le modèle d’avion
* La variable identifié par « Registration » représente le numéro de registration de l’avion
* La variable « Cn.In » …
* La variable « Aboard » comprend le nombre de passagers à bord lors du crash
* La variable « Fatalities » affiche le nombre de morts lors du crash
* La variable « Ground » a
* « Summary » affiche des détails concernant le crash. Les données dans cette colonne sont affichées sous forme de texte.

Ces informations sont affichées à travers 13 colonnes. Plusieurs d’entre elles sont manquantes.

\*\*\*\* je dois trouver un moyen de savoir combien de variables manquantes il y a dans chaque colonnes

# Les packages :

Les packages reliés à « ggmap » sont installés au début du programme. Ces packages sont nécessaires à l’utilisation de la fonction « afficher\_map »

# Les fonctions :

La fonction appelée « trouver\_fréquence » a été créée. Cette fonction a pour but de retourner la fréquence de chaque ‘’Instance’’ d’une variable. Les arguments à passer dans la fonction sont un vecteur sans duplicates et un vecteur avec les duplicates. Une double boucle est utilisée pour calculer le nombre de fois où on retrouve une instance de variable du premier vecteur dans le deuxième vecteur. La fonction retourne le total pour instance du premier vecteur dans un vecteur appelé « Total\_occurence ».

La fonction appelée « afficher\_map » a été créée. Cette fonction a pour but de retourner un map avec les coordonées géographique de chaque valeur dans un vecteur de type «String »

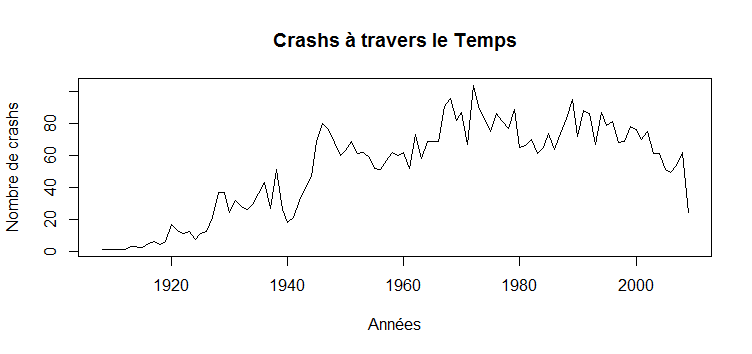
« Le nombre de crashs à travers le temps »

Pour la question #, « Le nombre de crashs à travers le temps », l’étape de préparation des données inclut les étapes suivantes :

1. Extraction de l’année (les 4 derniers caractères) de la variable « Date » à l’aide de la fonction substr et application de ce vecteur a la variable à « Annee »
2. Combination du vecteur appelé « Annee » et d’une partie du jeu de données initial a « table\_temporaire » à l’aide de la fonction cbind.
3. Appel de la fonction « trouver\_frequence » et application du resultat à la variable « x »
4. Création d’un data.frame appelé « table\_occurence\_annee » avec « x » (vecteur de fréquence des dates) et le vecteur d’années sans duplicats
5. Nomination des colonnes du data.frame
6. Ordonne « table\_occurence\_annee » en ordre croissant par date

Étape final :

Affichage du nombre de crashs par année à travers le temps à l’aide de la fonction « plot »

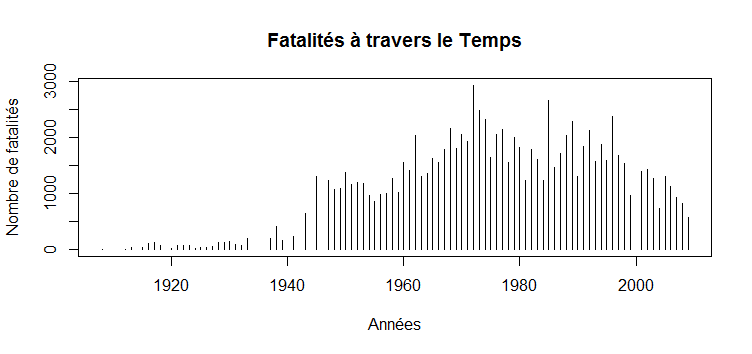


Conclusion :

« Nombre de fatalités par année à travers le temps»

Pour la question # , le nombre de fatalités à travers le temps, la préparation des données inclut les étapes suivantes :

1. Création d’un data.frame avec quelques colonnes du jeu de données initiales et les années
2. La définition du nom des colonnes du data.frame appelé table\_de\_fatalites\_par\_annee
3. Utilisation d’une boucle pour trouver la somme du nombres de fatalités pour chaque année
4. L’avant-dernière étape de préparation des données implique la création d’un data.frame avec chaque valeur de somme par années et les années
5. La dernière étape de préparation des données comprend la création d’un graphique à l’aide la fonction « plot »

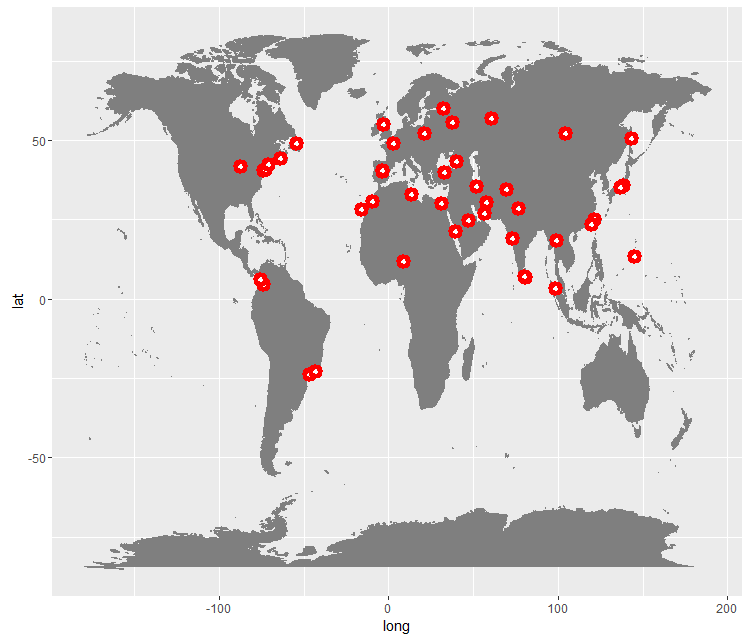


Conclusion :

Pour la question #, nombre de fatalités par location, la préparation des données inclut les étapes suivantes :

1. Utilisation d’une boucle pour trouver la somme du nombre de fatalités pour chaque location
2. L’organisation de la table en ordre décroissant par nombre de fatalités par location
3. L’extrait du top 50 des locations avec le plus grand nombre de fatalités par location
4. L’utilisation d’une boucle pour transformer en « string » chacune des instance de la variable « location » de la table top\_50\_location\_par\_fatalites
5. La dernière étape de préparation des données comprend l’utilisation de la fonction « afficher\_map » pour faire afficher sur un map le top 50 des locations avec le plus grand nombres de fatalités.

À noter que cinques valeurs qui commencent par un énoncé autre qu’un nom de ville n’affiche pas sur le map. Par exemple, les valeurs qui comment avec «Near, AtlanticOcean, off » n’ont pas retournées de coordonées géographiques.



Conclusion :

Pour la question # , le nombre de crash par « location », la préparation des données inclut les étapes suivantes :

1. L’utilisation de la fonction trouver\_fréquence pour trouver le nombre de crashs par location
2. La création d’un data.frame pour combiner la fréquence par « location » et les «  locations »
3. L’utilisation de la fonction « order » pour mettre en ordre décroissant du nombre de crash par location
4. L’extrait du top 50 des « locations » avec la plus grande fréquence de nombres crashs
5. L’étape finale comprend l’utilisation de la fonction « afficher\_map » pour faire afficher sur un map les villes avec le plus grand nombre de crashs.

À noter que la fonction afficher\_map omet d’afficher une valeur « AtlantiOcean, 110 miles West of Ireland ». Ceci est dû au fait que la valeur entrée n’est pas un nom de ville exact, donc il est difficile de retrouver les coordonnées géographiques de cette valeur.

