solution de la feuille du 19.01.23

La bibliothèque `pandas` est l'une des bibliothèques les plus utiles (et utilisées) par les personnes qui sont confrontées à la manipulation et visualisation de données. A la fin de la seconde feuille de solution, je mettrai un certain nombre de ressources, la plupart sont en anglais, pour ceux qui souhaitent explorer un peu plus (sur votre temps personnel).

Les solutions fournies aux exercices ne sont certainement pas les seules possibles, je vous encourage à les regarder, même si celles que vous avez trouvées sont correctes (au sens font ce qui est demandé).

Exercice

créez une série indexée sur les journées de janvier et février 2020 et dont les valeurs sont positives pour la première quinzaine du mois, négatives pour la seconde quinzaine du mois
créez une seconde série avec les mêmes index, mais les valeurs seront aléatoirement prises dans 1,2,3
affichez la première série si la valeur dans la seconde série est impaire

```
import pandas as pd # bibliothèque pandas
import numpy as np # bibliothèque numpy
import random # pour aleatoire
# création index
t_index = pd.date_range("2020", end="2020-02-29", freq="D")
# valeurs aléatoire entre 0 et 1
valeurs = np.random.rand(t_index.size)
# creation série
s = pd.Series(valeurs, index=t_index)
# changement de signe
s[s.index.day>15] *= -1
# creation 2nd série avec aléatoirement 1,2 ou 3
s2 = pd.Series(random.choices([1,2,3], k=t_index.size), index=t_index)
# affichage de la première série en fonction de la seconde
print(s[s2%2==1])
```

Exercice

Dans `big_df` combien y-at-il de données telles que 'mobility' soit trottinette ? Quelle est la valeur moyenne de 'delay' pour ces valeurs ? Quelle est la valeur du premier quartile de 'delay' ? Quel est le maximum de la variable 'stop' ?

```
# combien de données de big_df telles que mobility soit trotinette
# soit avec shape[0], soit avec count(), soit avec size (pour une série)
print("mobility with shape",
      big_df[big_df.mobility=="trottinette"].shape[0])
print("mobility with count",
      big_df.mobility[big_df.mobility=="trottinette"].count())
print("mobility with size",
      big_df.mobility[big_df.mobility=="trottinette"].size)
# pour accéder aux informations statistiques on peut
# utiliser describe()
trot = big_df[big_df.mobility=="trottinette"]
print('delay', trot.delay.describe())
print('stop', trot.stop.describe())
# utiliser les fonctions adéquates
print('delay', trot.delay.quantile(.25))
print('stop', trot.stop.max())
```

Exercice

Pour la série 'couleur' de `big_df` donnez les valeurs obtenues par les méthodes `.count` et `.value_counts` ; puis appliquez les 3 remplacements (par une valeur, en utilisant `ffill`, en utilisant `bfill`) et donnez les scores que vous obtenez
Dans la table `ventes` ajoutez une colonne "facteur" qui soit le ratio entre les colonnes "Sales" et "TV". Quelles sont les statistiques pour la série "facteur" ? Quelle est la valeur du dernier quartile ?
Quelle est la commande pour supprimer ***définitivement*** la colonne "facteur" ?
Comment supprimer les données de la table `ventes` qui ont une valeur "TV" > 150, sans impacter la table `ventes` ?
On souhaite rajouter une colonne 'budget_pub' dans la table `ventes` qui contiennent 'gros' si la valeur "TV" est > 200, 'moyen' si la valeur est > 100, 'bas' sinon.

```
### comptage couleur
print(">>> avant")
print('count', big_df.couleur.count())
print('value_counts\n', big_df.couleur.value_counts())
# fillna
print(">>> fillna('noire')")
print('count', big_df.couleur.fillna('noire').count())
print('value_counts\n', big_df.couleur.fillna('noire').value_counts())
# ffill
print(">>> fillna(method='ffill')")
print('count', big_df.couleur.fillna(method='ffill').count())
print('value_counts\n',
big_df.couleur.fillna(method='ffill').value_counts())
# bfill
```

```
print(">>> fillna(method='bfill')")
print('count', big_df.couleur.fillna(method='ffill').count())
print('value_counts\n',
big_df.couleur.fillna(method='bfill').value_counts())

### ventes
ventes['facteur'] = ventes.Sales / ventes.TV
print("describe\n", ventes.facteur.describe())
print("dernier quartile", ventes.facteur.quantile(.75))
# suppression definitive
print('colonnes avant', ventes.columns)
ventes.drop('facteur', axis=1, inplace=True)
print('colonnes après', ventes.columns)
# suppression temporaire
print("après suppression\n",
      ventes.drop(ventes[ventes.TV>150].index).TV.describe())
print("toujours là ?\n", ventes.TV.describe())
# budget_pub
ventes['budget_pub'] = 'bas'
ventes.loc[ventes.TV>100, 'budget_pub'] = 'moyen'
ventes.loc[ventes.TV>200, 'budget_pub'] = 'gros'
print("répartition\n", ventes.budget_pub.value_counts())
print("TV >  200", ventes[ventes.TV>200].TV.count())
print("TV <= 100", ventes[ventes.TV<=100].TV.count())
```