

Relations

Commençons

Pour cet exercice, vous utiliserez des données au niveau de la circonscription de l'élection présidentielle de 2016 pour visualiser les relations entre les variables. Ces données proviennent du [MIT Election Data and Science Lab](#).

Vous devez utiliser un projet RStudio pour bien organiser vos fichiers. Les données sont dans le dossier data

- results_2016.csv

A partir de maintenant je ne vous donnerai plus de squelette, c'est à vous de créer votre propre document Rmarkdown.

Au final, la structure de votre répertoire de projet devrait ressembler à ceci :

```
your-project-name\  
  07-exercice.Rmd  
  your-project-name.Rproj  
  data\  
    results_2016.csv
```

L'exemple de cette semaine vous sera très utile pour ce devoir.

Encore une fois, vous n'avez pas besoin de rendre vos graphiques super fantaisistes, mais si vous vous sentez courageux, essayez d'ajouter un calque `labs()` ou de changer les couleurs ou de modifier les thèmes et les éléments de thème.

Vous devrez insérer vos propres morceaux de code là où c'est nécessaire. Plutôt que de les taper à la main (c'est fastidieux et vous pourriez mal compter le nombre de backticks !), utilisez le bouton "Insérer" en haut de la fenêtre d'édition, ou tapez

ctrl + alt< /kbd> + i sous Windows, ou ⌘ + ⌘ + i sous macOS.

Tache 1: Reflection

Écrivez votre réflexion pour les lectures du jour.

Tache 2: Combiner des graphiques

Faites 2 à 3 graphiques de tout ce que vous voulez à partir des données `results_2016` (histogramme, densité, boxplot, nuage de points, peu importe) et combinez-les avec **patchwork**. Regardez [la documentation](#) pour voir des façons originales de les combiner, comme avoir deux lignes dans une colonne.

Tache 3: Visualiser des regressions

Tracé des coefficients

Utilisez les données `results_2016` pour créer un modèle qui prédit le pourcentage de votes démocrates dans une circonscription en fonction de l'âge, de la race, du revenu, du loyer et de l'état (indice : la formule ressemblera à ceci :

```
percent_dem ~ median_age + percent_white + per_capita_income + loyer_médian + state )
```

Utilisez `tidy()` dans le package **broom** et `geom_pointrange()` pour créer un diagramme de coefficients pour les estimations du modèle. Vous aurez 50 lignes pour tous les états, ce qui est excessif pour un tracé comme celui-ci, vous devrez donc filtrer les lignes d'état. Vous pouvez le faire en ajoutant ceci :

```
tidy(...) %>%  
  filter(!str_detect(term, "state"))
```

La fonction `str_detect()` recherche les caractères “state” dans la colonne `term`. Le `!` le nie. Cela revient donc à dire “ne conserver que les lignes où le mot ‘état’ n’est pas dans le nom du terme”.

Vous devriez également vous débarrasser de l’ordonnée à l’origine (`filter(term != "(Intercept)")`).

Valeurs prédites

Montrez ce qui arrive à `percent_dem` lorsqu’une (ou plusieurs) des variables de votre modèle changent. Pour vous simplifier la vie, reportez-vous à la section “Valeurs prédites et effets marginaux en 2022” dans l’exemple de cette session et utilisez `predictions()` plutôt que de créer votre propre ensemble de données `newdata` à la main. Vous ferez quelque chose comme ceci (en supposant que vous manipulez `per_capita_income` ; essayez d’utiliser une variable différente lorsque vous faites l’affectation):

```
my_predictions <- predictions(  
  model_name,  
  newdata = datagrid(per_capita_income = seq(9000, 60000, by = 100),  
                     state = "Georgia"))
```

Tracez votre variable variée sur l’axe des x, les valeurs ajustées (“prédites”) sur l’axe des y, montrez la relation avec une ligne et ajoutez un ruban pour montrer l’intervalle de confiance à 95 %.

Tâche bonus 1 ! Corrélogrammes

Ceci est entièrement facultatif mais peut être amusant.

Pour des moments de plaisir supplémentaires, si vous en avez envie, créez une carte thermique de corrélogramme, soit avec `geom_tile()` , soit avec des points dimensionnés par la corrélation. Utilisez toutes les variables que vous voulez à partir de `results_2016` .

Tâche bonus 2 ! Effets marginaux

Ceci est également entièrement facultatif mais sera très utile si vous utilisez la régression pour quoi que ce soit dans votre propre travail.

Pour des moments de plaisir super bonus supplémentaires, créez un modèle plus complexe qui prédit “percent_dem” qui utilise des termes polynomiaux (par exemple, l’âge au carré) et/ou des termes d’interaction (par exemple, l’âge \times l’état). Tracez les prédictions du modèle, utilisez `marginaleffects()` pour trouver les pentes de ces prédictions à différentes valeurs, et tracez les pentes dans un graphique des effets marginaux.

Tout remettre en place

Lorsque vous avez terminé, cliquez sur le bouton “Knit” en haut de la fenêtre d’édition et créez une version HTML ou Word (ou PDF si vous avez [installé tinytex](#)) de votre document. Déposez votre travail sur Moodle.