

TDU.

Introduction à R pour les enquêtes de santé publique

Cahier d'exercices pour le Jour 1



Public Health
Agency of Canada

Agence de la santé
publique du Canada

Canada

Contenu

Exercice pratique.....	2
Instructions	2
Introduction	3
Organisation de votre espace de travail	4
Fonctions superstar	26

EXERCICE PRATIQUE

Instructions

Durée : 3 heures et 10 minutes

Les apprenants et apprenantes ont un scénario, des questions et des tâches avec le code associé pour effectuer chaque tâche. Nous recommandons aux :

Utilisateurs et utilisatrices novices (maîtres d'équipage): Utilisez ce cahier d'exercices ainsi que le ou les scripts R fournis par le biais de GitHub. En vous servant de ce cahier d'exercices comme guide, exécutez le code que nous vous avons fourni étape par étape pour comprendre ce à quoi chaque partie du code sert et ce que font les différentes fonctions. À ce stade-ci, ne vous en faites pas trop à propos d'écrire ou de corriger le code.

Utilisateurs et utilisatrices de niveau débutant/intermédiaire (seconds capitaines et secondes capitaines): Le code R est fourni sous forme de capture d'écran dans ce cahier d'exercices. Vous devriez avoir une assez bonne compréhension du codage pour avoir une idée générale du code en le lisant (avec la documentation d'aide, quelques recherches Google au besoin et les commentaires se trouvant dans les scripts fournis par l'entremise de GitHub). Notre but est de vous faire écrire le code à partir du guide à mesure que vous avancez dans le scénario. Référez-vous aux scripts fournis dans GitHub si vous rencontrez des problèmes.

Utilisateurs et utilisatrices avancés (capitaines au long cours): Nous vous encourageons à essayer d'écrire votre propre code où vous voulez ainsi qu'à le contraster avec le code utilisé pour l'exercice et à aider vos pairs si des questions surviennent. Référez-vous aux scripts fournis dans GitHub si vous rencontrez des problèmes.

Avancez autant que vous le pouvez cet exercice en deux heures et dix minutes (maximum). Ne vous en faites pas si vous avez besoin de plus de temps. La courbe d'apprentissage pour R est abrupte et les apprenants ainsi que les apprenantes bénéficieront encore plus du temps consacré à la pratique. Nous planifions un webinaire de suivi d'une durée d'une heure après le cours pour faire le point sur les exercices (date à venir). Avant cela, si vous avez besoin

d'aide avec le matériel de cours, contactez les personnes responsables de l'animation de votre cours par le biais de Slack ou par courriel.

Introduction

Nous sommes au début du mois de décembre 2020. Vous êtes dans le bureau de l'agente de santé provinciale au ministère de la Santé de la Colombie-Britannique. Vos collègues en santé publique et vous-mêmes en avez plein les bras avec les interventions en lien avec la pandémie de COVID-19 et le développement d'une compréhension de la présence de la COVID-19 dans la province. L'un des éléments demandés par l'agente de santé provinciale pour assurer sa connaissance de la situation est une analyse régulièrement mise à jour de la situation de COVID-19 dans les autres provinces canadiennes. Il s'agit d'une tâche supplémentaire qui s'ajoute à votre charge de travail déjà importante.

1. Envisageriez-vous d'automatiser cette analyse? Quels facteurs utiliseriez-vous pour prendre cette décision?
2. Quelles informations incluriez-vous dans cette mise à jour?
3. Où trouveriez-vous les données nécessaires à l'exécution de cette analyse? Indice : pensez à des sources qui sont déjà accessibles au public.

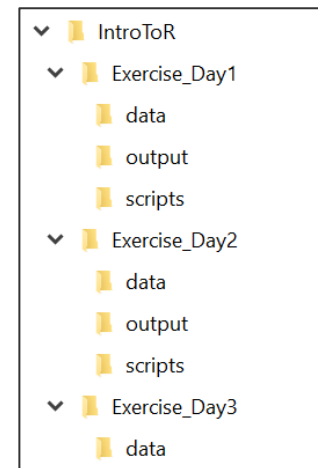
Après avoir évalué vos options, vous décidez qu'il est logique pour vous d'automatiser cette tâche. Il a également été convenu que votre partie de cette mise à jour comprendra une annexe de figures : des courbes épidémiques pour les cas et les décès pour chaque province. Vous avez également décidé d'utiliser la liste de lignes pour la COVID-19, source ouverte du Gouvernement du Canada, ainsi que les estimations de population de Statistique Canada. Il est votre devoir de produire les figures et de les partager avec l'équipe chargée de la compilation des informations pour l'agente de santé provinciale.

Organisation de votre espace de travail

Avant de planifier ou d'exécuter une analyse dans R (ou tout autre langage logiciel), une étape essentielle est de mettre en place un espace de travail efficace. Dans ce cours, nous vous demanderons de suivre l'exemple de structure de dossiers que nous avons créé afin de minimiser les conflits dans le code lors de l'accès aux fichiers de données ou de la recherche de scripts. Toutefois, n'hésitez pas à expérimenter ce qui convient le mieux à votre façon de coder.


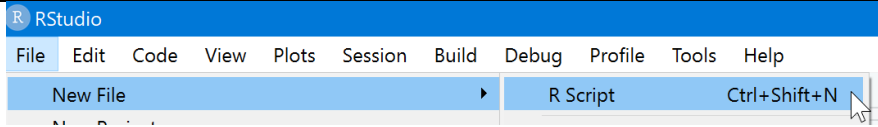
D'ici la fin des trois jours de ce cours, vos dossiers devraient ressembler à ceux sur l'image de droite. Notez que nous n'avons pas indiqué de façon explicite où vous devriez garder votre dossier principal (« **IntroToR** »). La raison est que nous utiliserons des fichiers *.Rproj* (R Project) et le paquet *here()* pour faciliter les chemins de fichiers pertinents (*relevant*) plutôt que les chemins de fichiers absolus (*absolute*). Il y a des avantages et des inconvénients à chacune de ces approches, nous en discuterons d'ailleurs dans le cours, mais lorsque vous partagez du code R avec des utilisateurs et utilisatrices dont la configuration d'ordinateur est très différente de la vôtre, la configuration de votre code avec de tels chemins de fichiers vous permettra de gagner beaucoup de temps.

Commencez par organiser votre espace de travail. Créez une structure de fichiers sur votre ordinateur, laquelle organisera tous les fichiers pour chaque exercice (p. ex. *scripts*, *data*, *output*).



1. Créez un nouveau dossier sur votre ordinateur intitulé *IntroToR*.
2. À l'intérieur du dossier *IntroToR*, créez un sous-dossier pour le Jour 1 du cours appelé *Exercise_Day1*.
3. À l'intérieur du dossier *Exercise_Day1*, créez les nouveaux dossiers suivants : *output*, *data* et *scripts*.
4. Déplacez le contenu du dossier *data* pour le Jour 1 de GitHub vers le nouveau dossier *IntroToR/Exercise_Day1*. (Si vous le désirez, vous pouvez également copier les fichiers se trouvant dans le dossier *scripts* dans GitHub vers votre dossier *scripts*. Toutefois, nous vous encourageons à commencer par écrire ces scripts vous-même, puis à garder ceux déjà faits en guise de référence).
5. Une fois que votre structure de dossiers est terminée, ouvrez RStudio et rendez-vous à *File*, puis *New Project*.
6. À la prochaine fenêtre, sélectionnez *Existing Directory*, puis cliquez sur *Browse* pour naviguer dans votre dossier **IntroToR**. Une fois que vous avez sélectionné votre dossier *IntroToR*, cliquez sur *Create Project*. Un nouveau fichier, *IntroToR.Rproj*, devrait apparaître dans le répertoire principal de votre espace de travail.

Maintenant que vous avez organisé vos dossiers de projet, vous pouvez commencer à écrire les scripts à utiliser dans votre analyse.

Tâche	Code
Après avoir ouvert RStudio, assurez-vous que vous travaillez dans votre nouvel environnement R-Project en allant dans <i>File</i> , puis Open Project , avant de vous rendre au fichier IntroToR.Rproj et de l'ouvrir OU d'ouvrir/de sélectionner votre fichier IntroToR.Rproj en cliquant sur l'icône de projet de R  dans le coin supérieur droit de l'interface RStudio, puis de défiler le menu déroulant qui apparaîtra.	
Ouvrez RStudio et commencez un nouveau script R en cliquant sur <i>File</i> , <i>New File</i> , puis R Script ou en utilisant le raccourci clavier Ctrl+Shift+N. Sauvegardez ce script dans votre dossier <i>Exercise_Day1/scripts</i> sous le nom de <i>01_define_paths.R</i> .	
Installez le paquet <i>here</i> .	<pre>install.packages("here")</pre>
Écrivez un énoncé pour créer un objet qui dirigera R vers l'emplacement de votre dossier où vos données brutes sont enregistrées. Exécutez l'énoncé. Expliquez dans vos propres mots ce que fait la ligne du code à droite.	<pre>data_folder <- here::here("Exercise_Day1", "data")</pre>
Écrivez un énoncé pour créer un objet qui dirigera R vers le dossier dans lequel toutes les figures ou les coupures de données que vous créez seront enregistrées. Exécutez l'énoncé.	<pre>output_folder <- here::here("Exercise_Day1", "output")</pre>
Écrivez un énoncé pour créer un objet qui dirigera R vers le dossier où vous enregistrerez tous vos scripts R associés à ce projet. Exécutez l'énoncé.	<pre>scripts_folder <- here::here("Exercise_Day1", "scripts")</pre>

Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les modifications et d'autres notes pertinentes.	
Sauvegardez votre script sous le nom de <i>01_define_paths.R</i> .	

4. Chargez les paquets (*packages*) dont vous aurez besoin pour ce projet :

Tâche	Code	Library – informations pour référence
Commencez un nouveau script R en sélectionnant <i>File > New File > R Script</i> ou en utilisant le raccourci clavier Ctrl+Shift+N. Sauvegardez ce script sous le nom de <i>02_load_libraries.R</i> dans votre dossier de scripts.		
(Facultatif) Installez les paquets qui ne sont pas déjà installés dans R Studio sur votre ordinateur. Veuillez noter que cela ne doit être fait qu'une fois. De plus, si vous avez déjà installé ces paquets pendant l'apprentissage précurs, vous n'avez pas à le faire maintenant.	<pre>install.packages("here") install.packages("tidyverse") install.packages("lubridate") install.packages("scales") install.packages("padr") install.packages("writexl") install.packages("fs") install.packages("RColorBrewer") install.packages("ggrepel") install.packages("ggpubr") install.packages("zoo") install.packages("viridis")</pre>	<p>Here : https://here.r-lib.org/</p> <p>readr : https://readr.tidyverse.org/</p> <p>readxl : https://readxl.tidyverse.org/</p> <p>tidyverse : https://www.tidyverse.org/</p> <p>scales : https://scales.r-lib.org/</p>

Chargez les paquets installés dont vous aurez besoin pour votre projet chaque fois que vous les utiliserez. Les paquets téléchargés seront cochés dans la fenêtre située dans le coin inférieur droit sous l'onglet <i>Packages</i> .	<pre>library(here) library(tidyverse) library(lubridate) library(scales) library(padr) library(writexl) library(fs) library(RColorBrewer) library(ggrepel) library(ggpubr) library(zoo) library(viridis)</pre>	<p>padr : https://www.rdocumentation.org/packages/padr/versions/0.5.3</p> <p>writexl : https://cran.r-project.org/web/packages/writexl/index.html</p> <p>fs : https://fs.r-lib.org/</p> <p>RColorBrewer : https://www.rdocumentation.org/packages/RColorBrewer/versions/1.1-2/topics/RColorBrewer</p> <p>zoo : https://www.rdocumentation.org/packages/zoo/versions/1.8-8</p> <p>viridis : https://www.rdocumentation.org/packages/viridis/versions/0.3.4</p>
Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les modifications et d'autres notes pertinentes		
Sauvegardez votre script sous le nom de <i>02_load_libraries.R</i> .		

5. Vous avez accéder à une version simplifiée de la liste de lignes pour la COVID-19, source ouverte de l'ASPC et l'avez sauvegardée dans votre dossier de projet. De façon similaire, vous avez identifié une source de données du dénominateur que vous souhaitez utiliser. Comme vous n'avez pas besoin de travailler avec cet ensemble de données avant de le charger dans R, vous avez décidé de le lire directement à partir du site web de Statistique Canada.

Tâche	Code
Commencez un nouveau script R en sélectionnant <i>File > New File > R Script</i> ou en utilisant le raccourci clavier Ctrl+Shift+N, puis sauvegardez-le dans votre dossier de scripts sous le nom de <i>03_import_data.R</i> .	
Expliquez dans vos propres mots ce que fait la ligne du code à droite. Lisez le fichier .csv de la liste de lignes pour la COVID-19 dans R Studio.	<pre>phac_raw <- read_csv(here("Exercise_Day1", "data", "covid19.csv"))</pre>
Créez un nouveau vecteur de caractères X et attribuez-lui l'emplacement web du fichier .csv contenant le dénominateur de Statistique Canada en utilisant le lien ci-dessous à la place de <i>WEB ADDRESS</i> dans le code à droite : https://www150.statcan.gc.ca/t1/tbl1/en/dtl/downloadDbLoadingData.action?pid=1710000901&latestN=0&startDate=20200101&endDate=20200101&csvLocale=en&selectedMembers=%5B%5B2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%2C12%2C14%2C15%5D%5D&checkedLevels=	<pre>X <- "WEB ADDRESS" canada_pop_denom_raw <- read_csv(url(X))</pre>

<p>Que se passe-t-il lorsque vous cliquez sur l'adresse web ci-dessus?</p> <p>Expliquez dans vos propres mots ce que fait la deuxième ligne du code à droite. Lire le fichier .csv des données de population dans R Studio.</p>	
Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les modifications et d'autres notes pertinentes.	
Sauvegardez votre script de nouveau sous le nom de <i>03_import_data.R</i> .	

6. Maintenant que les données sont chargées dans R Studio, vous devez les nettoyer et les traiter. Tout d'abord, nettoyons les données du dénominateur (population) :

Tâche	Code
Commencez un nouveau script R en sélectionnant <i>File > New File > R Script</i> ou en utilisant le raccourci clavier Ctrl+Shift+N, puis sauvegardez-le dans votre dossier de	

scripts sous le nom de <i>04_1_clean_population_denominator.R</i> .	
Créez un vecteur composé du nom et de l'abréviation des provinces. Quel est le type de données stockées dans ce vecteur? Quelle est sa longueur? Qu'est-ce que la commande suivante indique-t-elle : <code>provinces_short_names[5]</code> ?	<pre>provinces_short_names <- c("Alberta" = "AB", "British Columbia" = "BC", "Manitoba" = "MB", "New Brunswick" = "NB", "Newfoundland and Labrador" = "NL", "Nova Scotia" = "NS", "Northwest Territories" = "NWT", "Ontario" = "ON", "Prince Edward Island" = "PEI", "Quebec" = "QC", "Saskatchewan" = "SK", "Yukon" = "YK", "Nunavut" = "NU")</pre>
Créez un nouvel objet en manipulant un objet existant. Utilisez les colonnes <i>GEO</i> et <i>VALUE</i> de <i>Canada_pop_denom_raw</i> pour créer un nouvel ensemble de données appelé <i>pop_denom_canada_clean</i> . Renommez les variables pour que la variable <i>province</i> = <i>GEO</i> et que la variable <i>PopTotal</i> = <i>Value</i> . Par la suite, à l'aide de la fonction <i>mutate</i> , remplacez le nom complet des provinces par leur abréviation, que vous avez créée précédemment.	<pre>pop_denom_canada_clean <- canada_pop_denom_raw %>% select(province = GEO, PopTotal = VALUE) %>% mutate(province = str_replace_all(province, provinces_short_names))</pre>
Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les modifications et d'autres notes pertinentes.	
Sauvegardez votre script sous le nom de <i>04_1_clean_population_denominator.R</i> .	

7. Puis les données du nominateur (liste de lignes) :

Tâche	Code
Commencez un nouveau script R en sélectionnant <i>File > New File > R Script</i> ou en utilisant le raccourci clavier Ctrl+Shift+N, puis sauvegardez-le sous le nom de <i>04_2_clean_phac_case_data.R</i> .	
Créez une nouvelle trame de données en fonction des données brutes de la liste de lignes. Cette trame de données contiendra les variables suivantes : <i>prname</i> , <i>date</i> , <i>numconf</i> , <i>numdeaths</i> , <i>numtested</i> . Toutefois, <i>prname</i> deviendra <i>location</i> , <i>numconf</i> deviendra <i>total_cases</i> , <i>numdeaths</i> deviendra <i>total_deaths</i> et <i>numtested</i> deviendra <i>total_tests</i> .	<pre>phac_clean <- phac_raw %>% select(location = prname, date, total_cases = numconf, total_deaths = numdeaths, total_tests = numtested)</pre>
Recodez la variable de caractère contenant les informations de date en une variable de date, puis remplacez le nom des provinces par leur abréviation (comme	<pre>phac_clean <- phac_clean %>% mutate(date = dmy(date), location = str_replace_all(location, provinces_short_names))</pre>

dans l'étape précédente). (Indice : utilisez les fonctions <i>mutate</i> , <i>dmy</i> et <i>str_replace_all</i>).	
Filtrez les données pour 1) ne conserver que les dossiers datant du 1er mars 2020 ou après et 2) exclure les voyageurs et voyageuses rapatriés. Indice : utilisez les opérateurs « >= », « & », « ! », et « %in% » afin de créer un énoncé logique.	<pre>phac_clean <- phac_clean %>% filter(date >= ymd("2020-03-01") & !location %in% c("Repatriated travellers"))</pre>
Triez la trame de données par emplacement et par date. (Indice : utilisez la fonction <i>arrange</i>).	<pre>phac_clean <- phac_clean %>% arrange(location, date)</pre>
Fusionnez les données du numérateur et du dénominateur par province. (Indice : utilisez la fonction <i>left-join</i>).	<pre>phac_clean <- phac_clean %>% left_join(pop_denom_canada_clean, by = c("location" = "province"))</pre>
Les provinces n'ont pas toutes une ligne pour chaque date. Pour ajouter une ligne pour chaque date et chaque province, vous devez d'abord regrouper vos données par	<pre>phac_clean <- phac_clean %>% group_by(location)</pre>

province. (Indice : utilisez la fonction <i>group_by</i>).	
Maintenant que vos données sont regroupées par province, vous pouvez trier les données par date. (Indice : utilisez la fonction <i>arrange</i>).	<pre>phac_clean <- phac_clean %>% arrange(date)</pre>
Procédez au remplissage automatique des dates manquantes à partir du 1er mars 2020. Pouvez-vous expliquer précisément ce que la fonction <i>pad</i> a fait aux données?	<pre>phac_clean <- phac_clean %>% pad(start_val = ymd("2020-03-01"), interval = "day")</pre>
Des valeurs sont manquantes dans les autres colonnes pour les jours nouvellement saisis. Répétez les valeurs de comptage (c.-à-d. pour <i>total_cases</i> , <i>total_deaths</i> , <i>total_tests</i> et <i>PopTotal</i>) pour les jours nouvellement saisis. (Indice : utilisez la fonction <i>fill</i>). Pouvez-vous expliquer	<pre>phac_clean <- phac_clean %>% fill(total_cases, total_deaths, total_tests, PopTotal)</pre>

précisément ce que la fonction <i>fill</i> a fait à l'ensemble de données?	
<p>Calculez :</p> <ol style="list-style-type: none"> 1) La semaine épidémiologique 2) Les nouveaux cas 3) La moyenne mobile de sept jours de nouveaux cas 4) Le taux de nouveaux cas par jour 5) La moyenne mobile de sept jours du taux de nouveaux cas 6) Les nouveaux décès 7) La moyenne mobile de sept jours des nouveaux décès 8) Le taux de nouveaux décès par jour 9) La moyenne mobile de sept jours du taux des nouveaux décès <p>(Indice : utilisez les fonctions <i>mutate</i>, <i>epiweek</i>, <i>lag</i>, <i>round</i>, et <i>rollmean</i>).</p>	<pre>phac_clean <- phac_clean %>% mutate(epi_week = epiweek(date), total_cases_per_million = round(total_cases/PopTotal*1000000, digits = 2), new_cases = total_cases - lag(total_cases, default = first(total_cases)), new_cases_smoothed = round(rollmean(new_cases, 7, fill = NA, align = "right"), digits = 2), new_cases_per_million = round(new_cases/PopTotal*1000000, digits = 2), new_cases_smoothed_per_million = round(rollmean(new_cases_per_million, 7, fill = NA, align = "right"), digits = 2), total_deaths_per_million = round(total_deaths/PopTotal*1000000, digits = 2), new_deaths = total_deaths - lag(total_deaths, default = first(total_deaths)), new_deaths_smoothed = round(rollmean(new_deaths, 7, fill = NA, align = "right"), digits = 2), new_deaths_per_million = round(new_deaths/PopTotal*1000000, digits = 2), new_deaths_smoothed_per_million = round(rollmean(new_deaths_per_million, 7, fill = NA, align = "right"), digits = 2)))</pre>
Dissociez les données.	<pre>phac_clean <- ungroup(phac_clean)</pre>

<p>Transposez les données : décrivez ce que la <i>fonction</i> <i>pivot_longer()</i> a fait aux données.</p> <p>Pourquoi voudrions-nous transposer les données ainsi?</p>	<pre>phac_clean <- phac_clean %>% pivot_longer(3:17, names_to = "metric", values_to = "value")</pre>
<p>Créez trois nouvelles variables :</p> <ol style="list-style-type: none"> 1) <i>Cases_deaths</i> pour indiquer si une valeur numérique fait partie des cas ou des décès 2) <i>Count_rate</i> pour indiquer si une valeur numérique est un compte ou un taux 3) <i>Raw_smoothed</i> pour indiquer si une valeur numérique est brute ou ajustée 	<pre>phac_clean <- phac_clean %>% mutate(cases_deaths = case_when(str_detect(metric, "cases") ~ "cases", str_detect(metric, "deaths") ~ "deaths", TRUE ~ "other"), count_rate = case_when(str_detect(metric, "million thousand") ~ "rate", str_detect(metric, "population positive positivity") ~ NA_character_, TRUE ~ "count"), raw_smoothed = case_when(str_detect(metric, "smoothed") ~ "smoothed", TRUE ~ "raw"))</pre>
<p>Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les</p>	

modifications et d'autres notes pertinentes.	
Sauvegardez votre script de nouveau sous le nom de <i>04_2_clean_phac_case_data.R</i> .	

8. Maintenant que les données sont préparées, créons une courbe épidémique des nouveaux cas en Ontario par jour et incluons une moyenne mobile de sept jours comme première figure :

Tâche	Code
Commencez un nouveau script R en sélectionnant <i>File > New File > R Script</i> ou en utilisant le raccourci clavier Ctrl+Shift+N, puis sauvegardez-le dans votre dossier de scripts sous le nom de <i>05_1_plot_epi_curve.R</i> .	
Créez une nouvelle trame de données contenant seulement les données de l'Ontario à représenter graphiquement et incluez les variables suivantes :	<pre>ONnew_daily_cases_data <- phac_clean %>% select(date, location, metric, value)</pre>

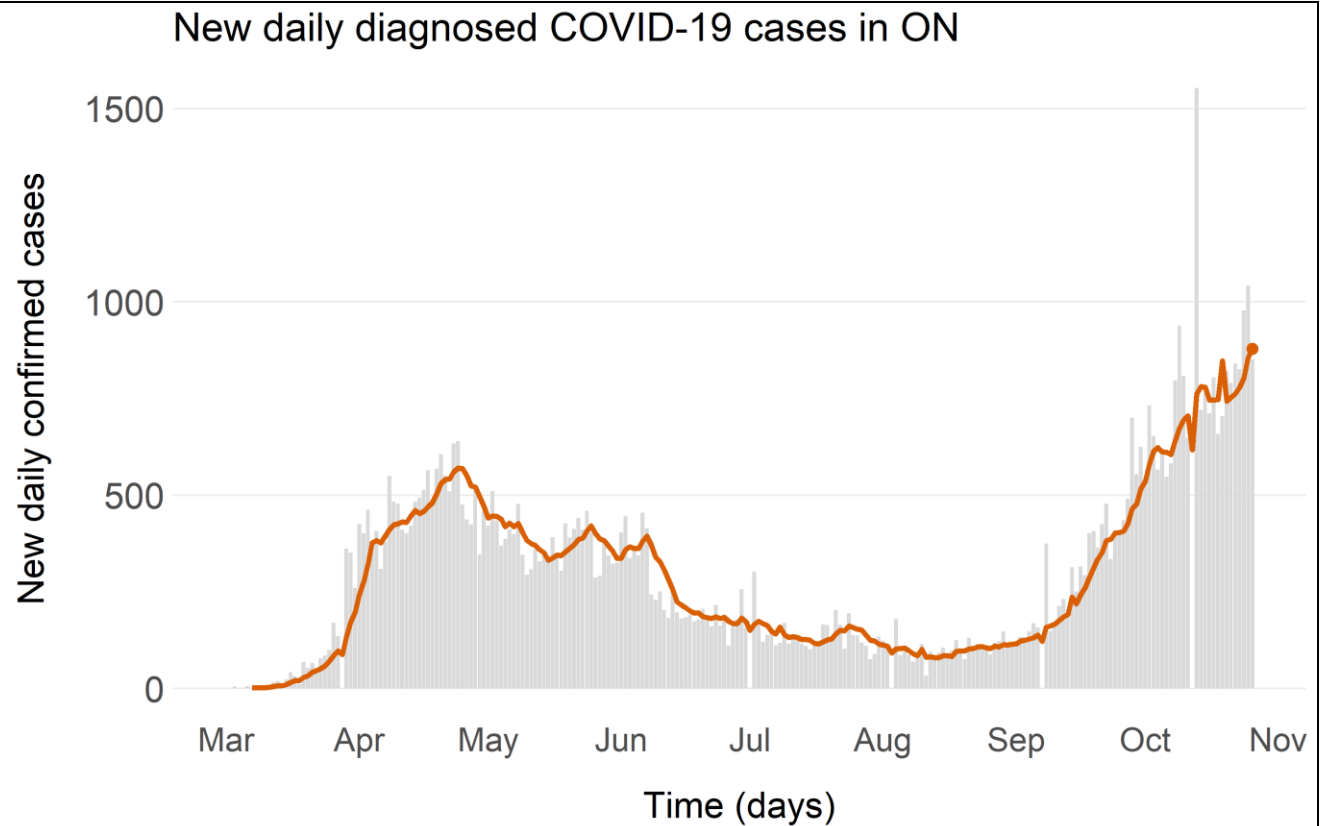
<p><i>date</i>, <i>location</i>, <i>metric</i> et <i>value</i>. (Indice : utilisez la fonction <i>select</i>).</p>	
<p>Filtrer les données en fonction de l'emplacement et des paramètres souhaités pour le graphique (c.-à-d. <i>new_cases</i> et <i>new_cases_smoothed</i>).</p>	<pre>ONnew_daily_cases_data <- ONnew_daily_cases_data %>% filter(location %in% c("ON") & metric %in% c("new_cases", "new_cases_smoothed"))</pre>
<p>Remodelez les données à tracer : Qu'est-ce que la fonction <i>pivot_wider()</i> fait aux données?</p>	<pre>ONnew_daily_cases_data <- ONnew_daily_cases_data %>% pivot_wider(names_from = "metric", values_from = "value")</pre>
<p>Créez un nouvel objet pour la représentation graphique des données. Quel est le but des fonctions <i>geom_bar()</i> et <i>geom_line()</i>?</p> <p>Conseil pratique : Si vous souhaitez tracer votre graphique et voir ce que fait le code, laissez simplement le dernier « + » et incluez la ligne suivante à la fin : <i>ONnew_daily_cases_plot</i>.</p>	<pre>ONnew_daily_cases_plot <- ggplot(data=ONnew_daily_cases_data, aes(x = date, y = new_cases)) + geom_bar(stat = "identity", fill = "#d9d9d9") + geom_line(aes(x = date, y = new_cases_smoothed), color = "#d95f02", size = 1.5) +</pre>

Cela vous permet de voir de quelle façon chaque section du code change votre graphique.	
Quel est le but de la fonction <code>geom_point()</code> ?	<pre>geom_point(data = phac_clean %>% select(date, location, metric, value) %>% filter(location %in% c("ON") & metric %in% c("new_cases_smoothed")) %>% pivot_wider(names_from = "metric", values_from = "value") %>% filter(date == max(date)), aes(x = date, y = new_cases_smoothed), size = 3, color = "#d95f02") +</pre>
Définissez le thème, dans ce cas-ci <i>minimal</i> (c.-à-d. aucune annotation d'arrière-plan).	<pre>theme_minimal() +</pre>
Modifiez le thème en précisant la taille ainsi que l'emplacement du texte et en retirant les lignes d'arrière-plan.	<pre>theme(axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10), axis.text.y = element_text(size = 10), plot.title = element_text(size = 14), axis.title.x = element_text(size = 12, margin = margin(t = 20, r = 0, b = 0, l = 0)), axis.title.y = element_text(size = 12, margin = margin(t = 0, r = 20, b = 0, l = 0)), plot.caption = element_text(size = 12, margin = margin(t = 20, r = 0, b = 0, l = 0)), legend.position = "none", panel.grid.minor = element_blank(), panel.grid.minor.x = element_blank(), panel.grid.major.x = element_blank()) +</pre>
Définissez l'échelle de la figure à un mois et ajoutez les étiquettes d'abréviation des mois.	<pre>scale_x_date(date_breaks = "1 months", date_labels = "%b") +</pre>

<p>Définissez les titres. Titre du graphique : <i>New daily diagnosed COVID-19 cases in ON</i> Axe des X : <i>Time (days)</i> Axe des Y : <i>New daily confirmed cases</i></p>	<pre>ggtitle("New daily diagnosed COVID-19 cases in ON") + xlab("Time (days)") + ylab("New daily confirmed cases")</pre>
Tracez la figure.	ONnew_daily_cases_plot
<p>Sauvegardez la figure en tant qu'image (fichier .png) à votre dossier <i>output</i> en utilisant <i>ggsave</i>. Nommez-la <i>ONnew_daily_cases_plot</i> et ajoutez automatiquement la date d'aujourd'hui au nom.</p>	<pre>ggsave(paste0(output_folder, "/ONnew_daily_cases_plot_", today, ".png"), plot = ONnew_daily_cases_plot, width = 11, height = 7, units = "in")</pre>
<p>Oh non! Il y a eu une erreur! Pouvez-vous passer en revue votre code ainsi que l'erreur et identifier le problème?</p>	<p><i>Indice : Nous n'avons pas dit à R quel était l'objet d'aujourd'hui. Envisagez de regarder la fonction Sys.Date dans la documentation d'aide et de l'entrer dans la console. Quelles informations vous donne-t-elle? Essayez de créer un objet nommé today et de l'attribuer à sys.date puis d'exécuter à nouveau votre code ggsave :</i></p> <pre>ggsave(paste0(output_folder, "/ONnew_daily_cases_plot_", today, ".png"), plot = ONnew_daily_cases_plot, width = 11, height = 7, units = "in")</pre>

Vous avez créé votre première figure pour ce projet! Hourra! C'est maintenant l'heure du dessert.

Remarque : Votre courbe épidémique aura l'air légèrement différente de celle-ci étant donné que les données sont tirées directement de la liste de lignes pour la COVID-19, source ouverte du Gouvernement du Canada, et qu'elle est mise à jour régulièrement.

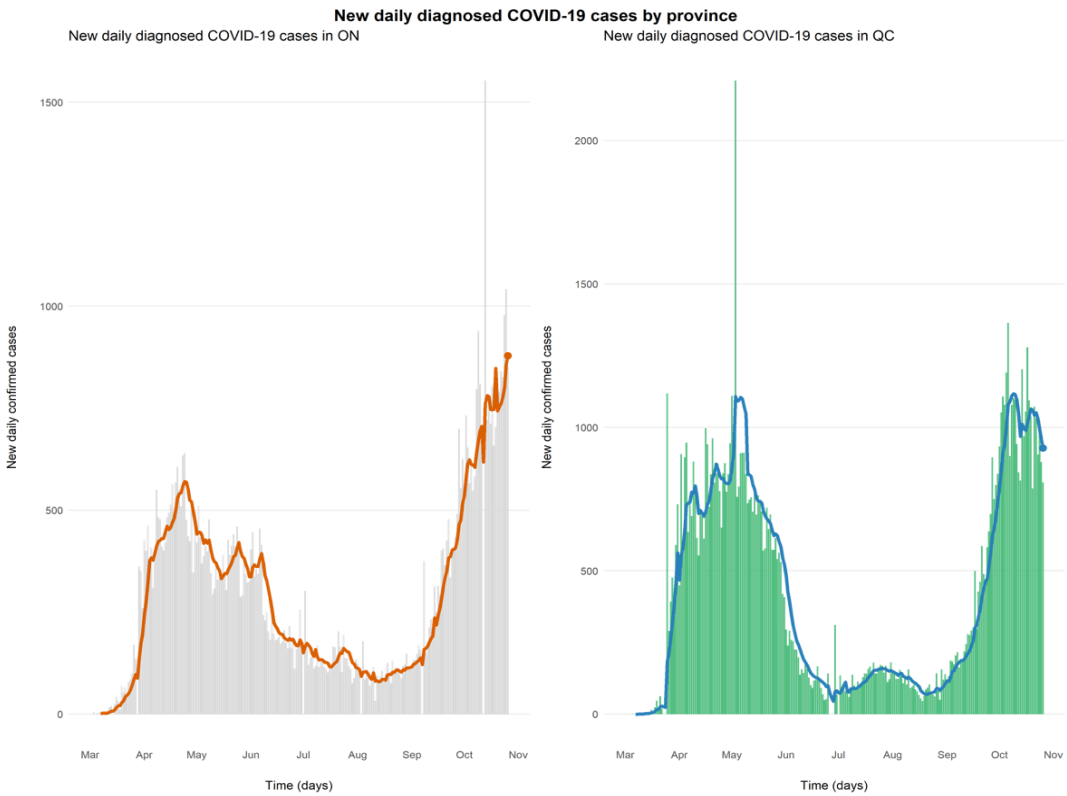


Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les modifications et d'autres notes pertinentes.

Sauvegardez votre script de nouveau sous le nom de <code>05_1_plot_epi_curve</code> .	
---------------------------------------------------------------------------------------	--

9. Réviser votre code de l'étape 9. Pouvez-vous identifier tous les endroits où les territoires sont précisés? Réutilisez votre code pour recréer une figure similaire pour le Québec. Amusez-vous avec les éléments esthétiques (c.-à-d. changez la couleur de vos barres, la taille de vos axes, la police des étiquettes, etc.). Vous pouvez trouver différents codes hexadécimaux ici : <https://htmlcolorcodes.com/> ou tapez simplement en anglais la couleur de votre choix (liste de couleurs disponibles : <http://sape.inf.usi.ch/quick-reference/ggplot2/colour>).
10. Génial! Vous avez un code qui fonctionne, et vous retournez maintenant en arrière pour passer en revue et réviser votre code. Afin de créer une figure pour chaque province, vous devrez faire beaucoup de copier-coller, de révision, et repasser plusieurs pages de code. Comment pourriez-vous gérer cela?
11. En prime! Lorsque vous avez passé en revue et révisé votre code, vous avez décidé de modifier vos figures afin qu'elles puissent être affichées ensemble dans une même figure à plusieurs panneaux. Réviser votre code de l'étape 9. Quels changements envisagez-vous d'apporter?

Tâche	Code
Créez une figure à plusieurs panneaux. Modifiez le code à droite pour inclure seulement vos figures	<pre> ALLnew_daily_cases_plot <- ggarrange(ONnew_daily_cases_plot, QCnew_daily_cases_plot, hjust=2.5, ncol = 2, nrow = 1) #To display your results: </pre>

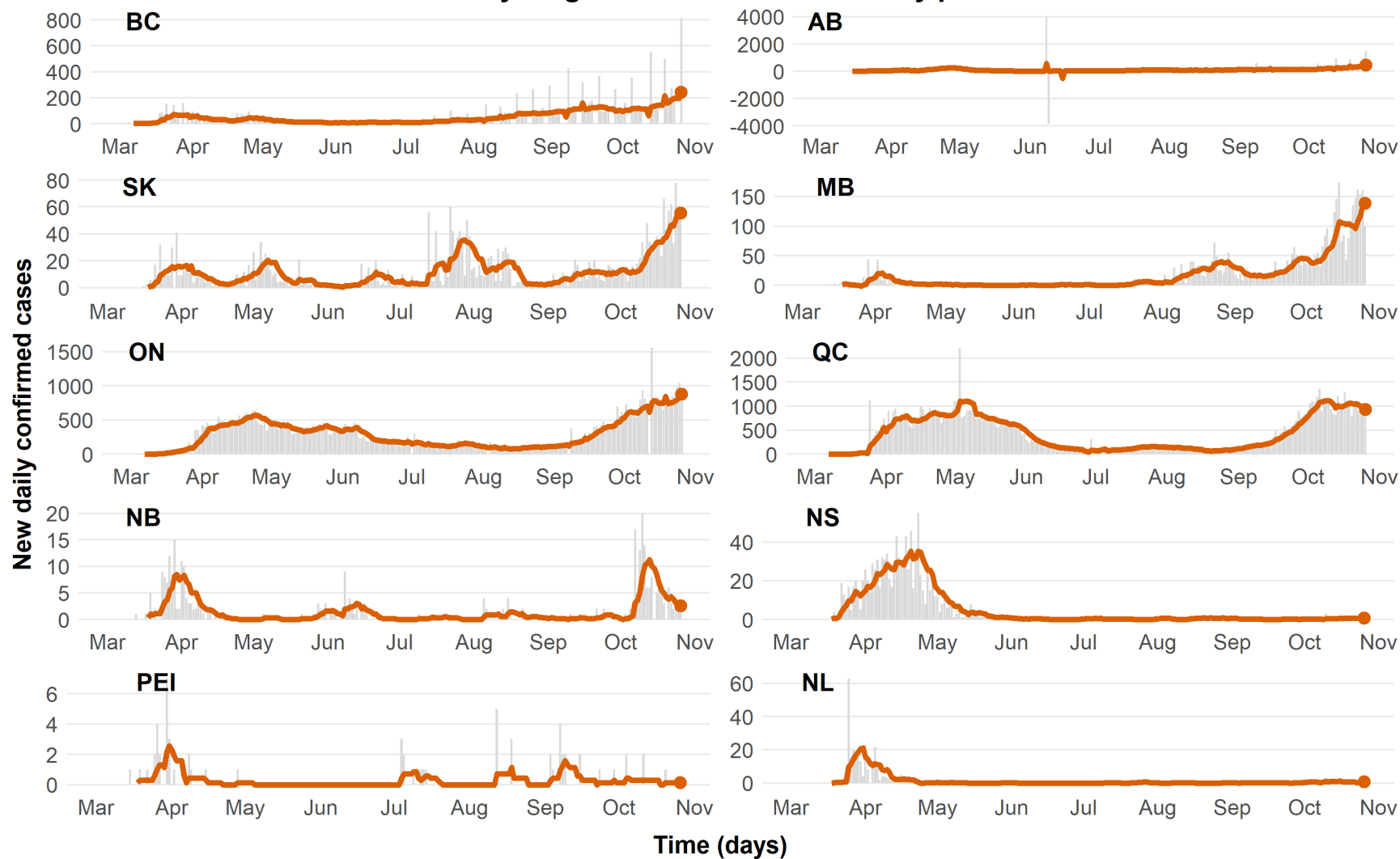
de l'Ontario et du Québec.	
<p>Consultez votre tracé à plusieurs panneaux.</p> <p>Que remarquez-vous? Y a-t-il quelque chose que vous retourneriez changer à la courbe épidémique originale de l'ON et du QC afin de la rendre plus visuellement attrayante dans un tracé à panneaux multiples?</p> <p>Conseil pratique : Regardez le code pour votre tracé de courbe épidémique original. Laisseriez-vous les titres ainsi?</p>	<p><code>ALLnew_daily_cases_plot</code></p>  <p>New daily diagnosed COVID-19 cases by province</p> <p>New daily diagnosed COVID-19 cases in ON</p> <p>New daily diagnosed COVID-19 cases in QC</p> <p>New daily confirmed cases</p> <p>Time (days)</p>
Retravaillez vos courbes épidémiques ON et QC jusqu'à ce	Facultatif!

<p>qu'elles soient bien formatées en un tracé à plusieurs panneaux.</p> <p>Indice : essayez de modifier les options dans la fonction <i>theme</i>.</p>	
<p>Exportez la figure avec annotations (c.-à-d. le titre et les étiquettes des axes).</p>	<pre>ggsave(paste0(output_folder, "/ALLnew_daily_cases_plot_", today, ".png"), plot = annotate_figure(ALLnew_daily_cases_plot , top = text_grob("New daily diagnosed COVID-19 cases by province", face = "bold", size = 16), bottom = text_grob("Time (days)", face = "bold", size = 14), left = text_grob("New daily confirmed cases", rot = 90, face = "bold", size = 14)), width = 11, height = 7, units = "in")</pre>
<p>Assurez-vous d'inclure un titre dans votre script avec des détails concernant le but, l'auteur, la date, les modifications et d'autres notes pertinentes.</p>	
<p>Sauvegardez votre 05_1_ plot_epi_curve.R.</p>	

12. En prime! Vous avez réutilisé votre code pour créer un graphique à plusieurs panneaux pour toutes les provinces (voyez l'image sur la page suivante. Note : ce code n'est pas inclus dans votre dossier de scripts - mais vous pouvez essayer de

recréer la figure par vous-même!). En regardant de plus près vos figures, vous remarquez quelque chose d'étrange. Que se passe-t-il en Alberta les 8 et 9 juin 2020? Passez en revue les données de ces dates dans la liste de lignes ainsi que toutes les données créées dans le cadre du traitement et du nettoyage, et tenez compte de la méthodologie utilisée pour obtenir les données d'énumération des nouveaux cas quotidiens. À quel moment cette erreur s'est-elle introduite? Choisiriez-vous de laisser la figure telle qu'elle est? Pourquoi ou pourquoi pas? Si ce n'est pas le cas, comment proposeriez-vous de régler ce problème?

New daily diagnosed COVID-19 cases by province



FONCTIONS SUPERSTAR

En guise de référence, nous vous avons fourni une liste des principales fonctions utilisées dans le cadre de cet exercice. Les fonctions en gras sont les plus grandes vedettes de la journée.

Fonctions superstar	
here()	left_join()
install.packages()	dmy()
library()	str_replace_all()
read_csv()	Sys.Date()
paste0()	ggplot()
url()	geom_bar()
c()	geom_line()
cselect()	geom_point()
mutate()	theme()
filter()	element_text()
group_by()	element_blank()
ungroup()	scale_x_date()
arrange()	xlab()
pad()	ylab()
fill()	ggtitle()
pivot_longer()	aes()
pivot_wider()	ggsave()
str_detect()	ggarrange()
epiweek()	annotate_figure()