Google Analytics Sample - Analysis

Tung Anh

2025-07-16

Introduction

Setup

```
# Chargement des bibliothèques
library(bigrquery)
library(DBI)
library(dplyr)
library(tidyr)
library(readr)
library(lubridate)
library(arrow)
library(glue)
library(here)
library(ggplot2)
library(scales)

# Pour la carte intéractif
library(countrycode)
library(rnaturalearth)
library(rnaturalearthdata)
library(sf)
library(plotly)
```

```
# Définir vos couleurs personnalisées
custom_colors <- c(
  "Blue"   = "#4285F4",
  "Red"    = "#DB4437",
  "Yellow" = "#F4B400",
  "Green"  = "#0F9D58"
)
```

Data Loading

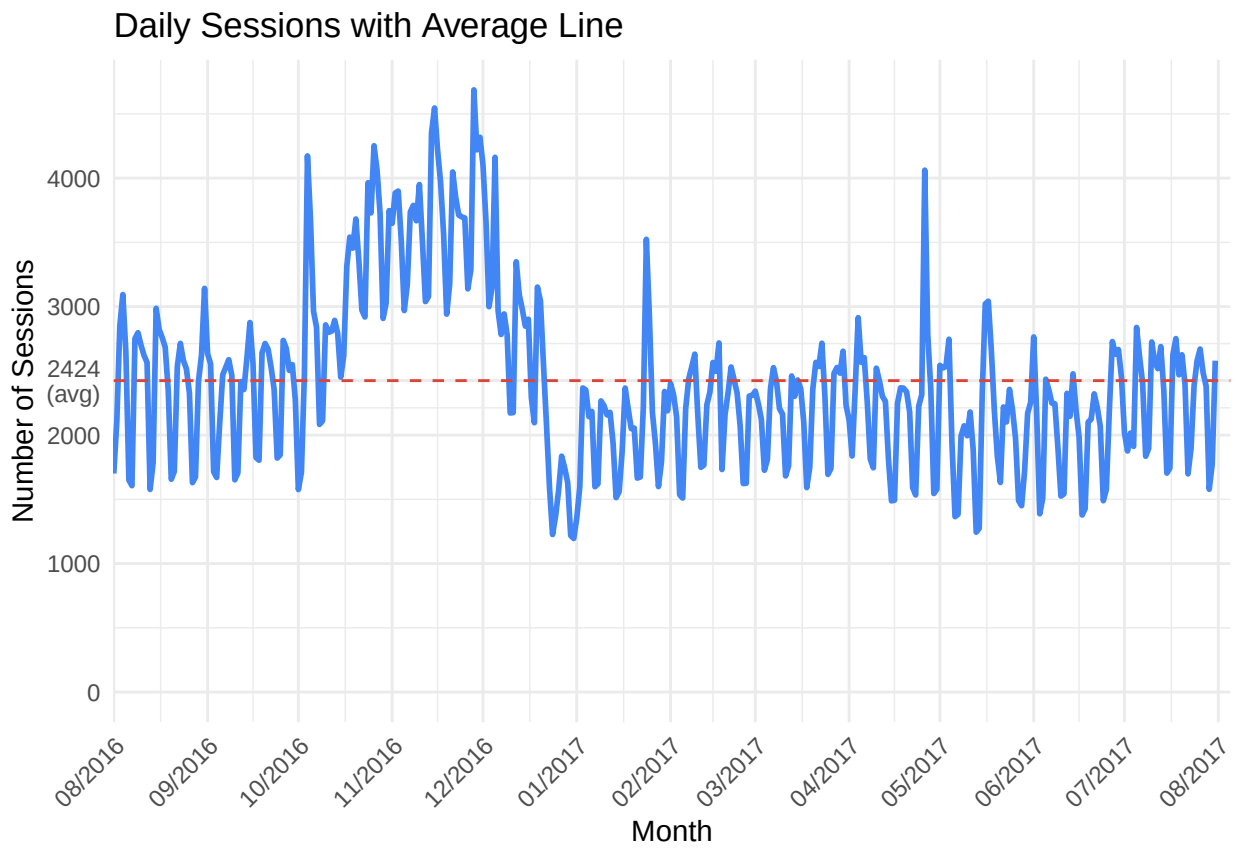
```
# Chargement le jeu de données
full_df <- open_dataset(here("data"), format = "parquet", partitioning = c("year", "month"))
```

```
df = full_df %>% collect()
```

1. Métriques principales

1a. Nombre de sessions par jour.

```
# Collection des données
sessions_per_day <- full_df %>%
  group_by(date = ymd(date)) %>%
  summarise(sessions = n_distinct(visitId)) %>%
  arrange(date) %>%
  collect()

# Moyenne et maximum des sessions
avg_sessions <- mean(sessions_per_day$sessions, na.rm = TRUE)
y_max <- max(sessions_per_day$sessions, na.rm = TRUE)

# Visualisation
ggplot(sessions_per_day, aes(x = date, y = sessions)) +
  geom_line(color = custom_colors["Blue"], linewidth = 1) +
  geom_hline(yintercept = avg_sessions, linetype = "dashed", color = custom_colors["Red"]) +  # ligne d
  scale_y_continuous(
    name = "Number of Sessions",
    limits = c(0, y_max),
    breaks = sort(unique(c(seq(0, y_max, by = 1000), round(avg_sessions, 0)))),
    labels = function(x) ifelse(x == round(avg_sessions, 0), # afficher la moyenne sur l'axe Y
                                paste0(x, "\n(avg)"),
                                x)
  ) +
  # axe X mensuel
  scale_x_date(
    date_breaks = "1 month",
    date_labels = "%m/%Y",
    expand = expansion(add = c(0, 5))
  ) +
  labs(title = "Daily Sessions with Average Line", x = "Month") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # rotation X
```

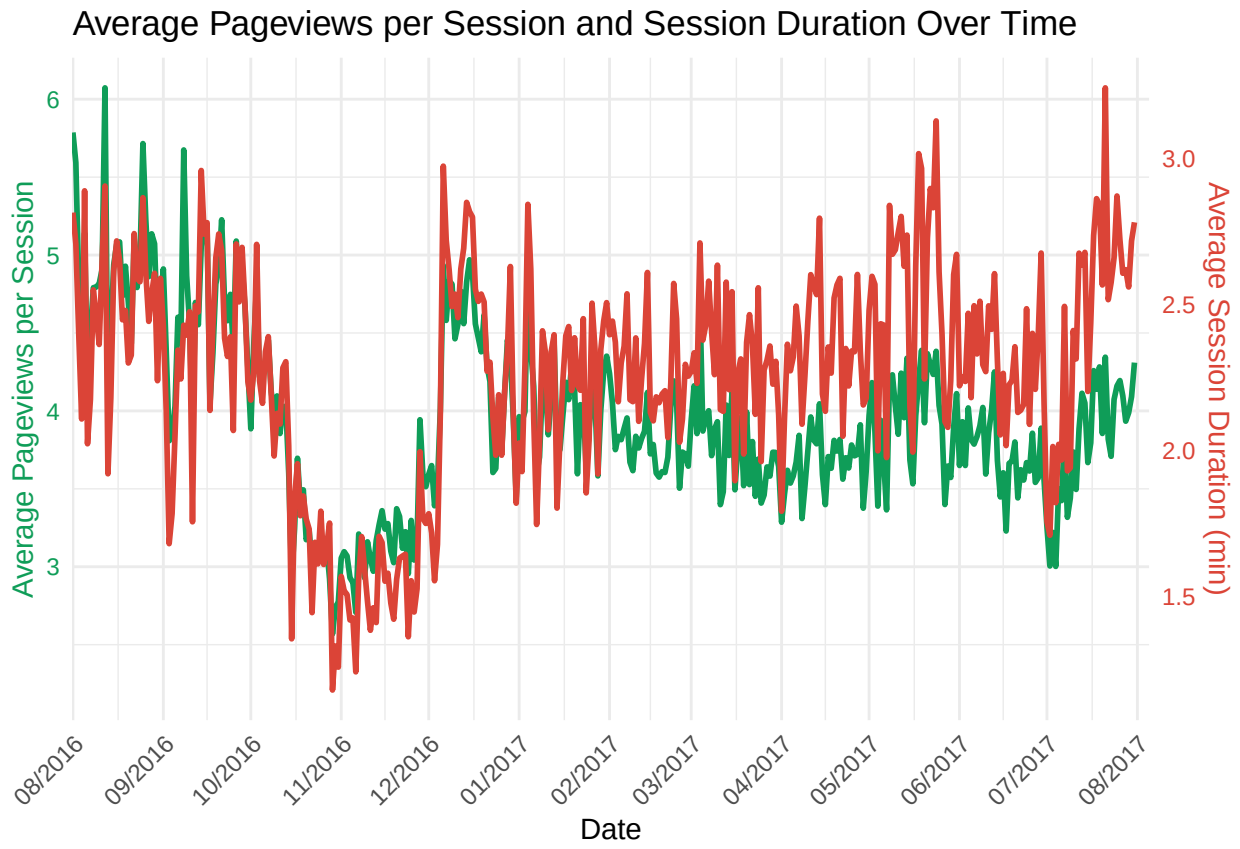


Le graphique montre une augmentation marquée du nombre de sessions quotidiennes entre octobre et décembre 2016, atteignant des niveaux bien supérieurs à la moyenne de 2 424 sessions par jour. Cette période reflète probablement une campagne marketing efficace ou une activité saisonnière. Après décembre 2016, le nombre de sessions chute et reste de manière constante en dessous de la moyenne durant les mois suivants, ce qui suggère un retour au trafic habituel ou une baisse de l'effort promotionnel. Globalement, le graphique met en évidence un pic temporaire d'engagement utilisateur suivi d'une période de trafic plus stable et réduit.

1b. Pages vues par session et durée moyenne des sessions

```
# Calculer le nombre moyen de pages vues par session
pageviews_per_session <- full_df %>%
  group_by(date = ymd(date)) %>%
  summarise(
    total_pageviews = sum(total_pageviews, na.rm = TRUE),
    sessions = n_distinct(visitId),
    avg_pageviews = total_pageviews / sessions
  ) %>%
  collect()

# 2. Calculer la durée moyenne des sessions (en minutes)
session_duration <- full_df %>%
  group_by(date = ymd(date)) %>%
  summarise(
    total_time = sum(total_timeOnsite, na.rm = TRUE),
```

```r
    sessions = n_distinct(visitId),
    avg_duration_sec = total_time / sessions
  ) %>%
  collect() %>%
  mutate(avg_duration_min = avg_duration_sec / 60)

# 3. Fusionner les jeux de données par date
combined_df <- left_join(pageviews_per_session, session_duration, by = "date")

# 4. Préparer les facteurs de mise à l'échelle pour l'axe secondaire

scale_factor <- max(combined_df$avg_pageviews, na.rm = TRUE) / max(combined_df$avg_duration_min, na.rm =

# 5. Plot avec dual y-axes
ggplot(combined_df, aes(x = date)) +
  geom_line(aes(y = avg_pageviews), color = custom_colors["Green"], size = 1) +
  geom_line(aes(y = avg_duration_min * scale_factor), color = custom_colors["Red"], size = 1) +
  scale_y_continuous(
    name = "Average Pageviews per Session",
    sec.axis = sec_axis(~ . / scale_factor, name = "Average Session Duration (min)")
  ) +
  labs(
    title = "Average Pageviews per Session and Session Duration Over Time",
    x = "Date"
  ) +
  scale_x_date(
    date_breaks = "1 month",
    date_labels = "%m/%Y",
    expand = expansion(add = c(0, 5))
  ) +
  theme_minimal() +
  theme(
    # pivoter x
    axis.text.x = element_text(angle = 45, hjust = 1),
    # coloration
    axis.title.y.left = element_text(color = custom_colors["Green"]),
    axis.text.y.left = element_text(color = custom_colors["Green"]),
    axis.title.y.right = element_text(color = custom_colors["Red"]),
    axis.text.y.right = element_text(color = custom_colors["Red"])
  )
```



Le graphique illustre l'évolution conjointe des pages vues moyennes et de la durée moyenne par sessions sur la période analysée. On observe que : - Les pages vues par session suivent globalement une tendance similaire à celle de la durée des sessions, suggérant que plus les utilisateurs restent longtemps sur le site, plus ils consultent de pages. - Une fluctuation marquée coïncident avec des décroissances dans les deux indicateurs durant les derniers mois de 2016, période déjà identifiée comme celle d'une forte activité ou campagne marketing. - Après le changement de la fin de 2016, on observe que la duration augmente par rapport au nombre de pages vues par session, ce qui indique une tendance des utilisateurs à passer plus de temps sur chaque page, probablement en lien avec la campagne menée à cette période.. Ces indicateurs conjoints permettent de mieux comprendre l'engagement des visiteurs, au-delà du simple nombre de sessions, et d'évaluer la qualité de la navigation.

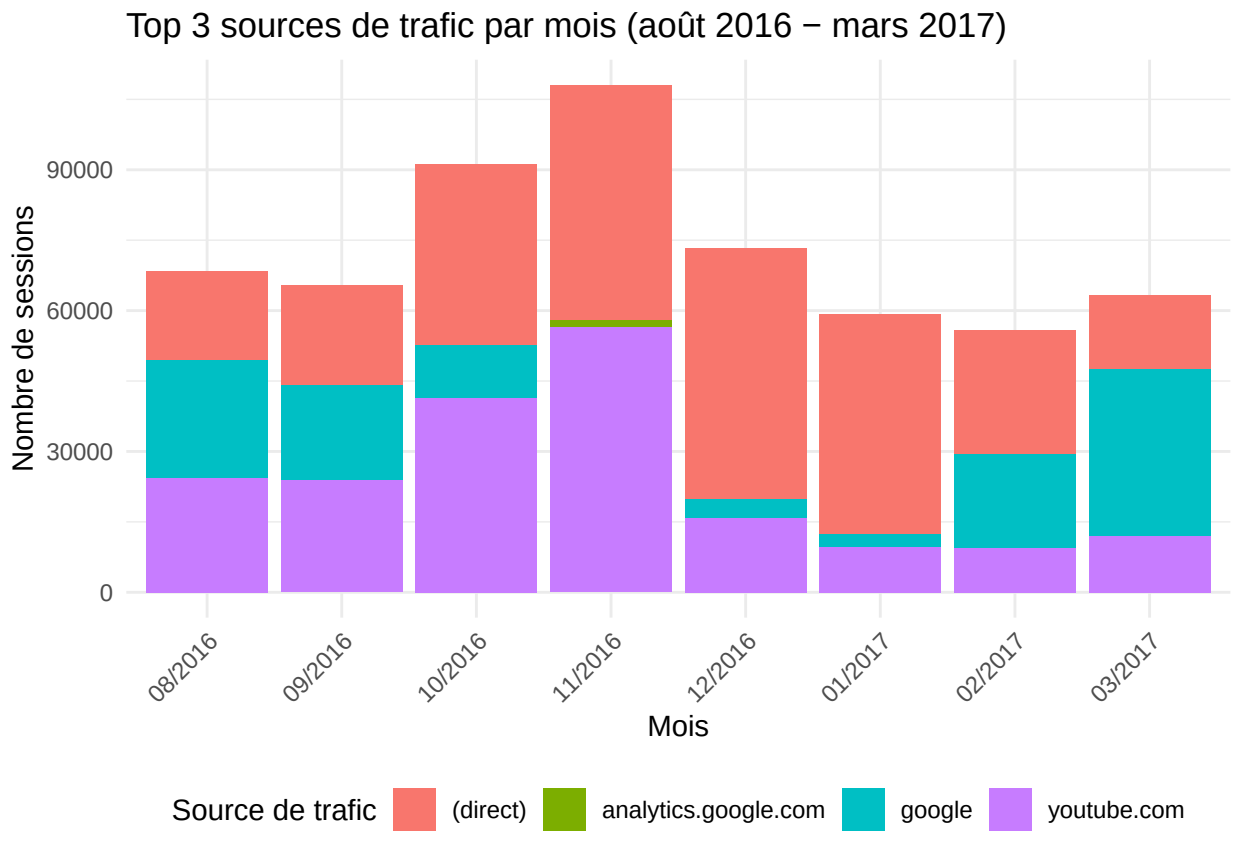
1c. Analyse de la tendance entre Octobre et Decembre 2016

La période octobre à décembre 2016 se distingue par : - Une hausse significative du nombre de sessions, - Une baisse simultanée de la durée moyenne par session, - Une réduction du nombre moyen de pages vues par session.

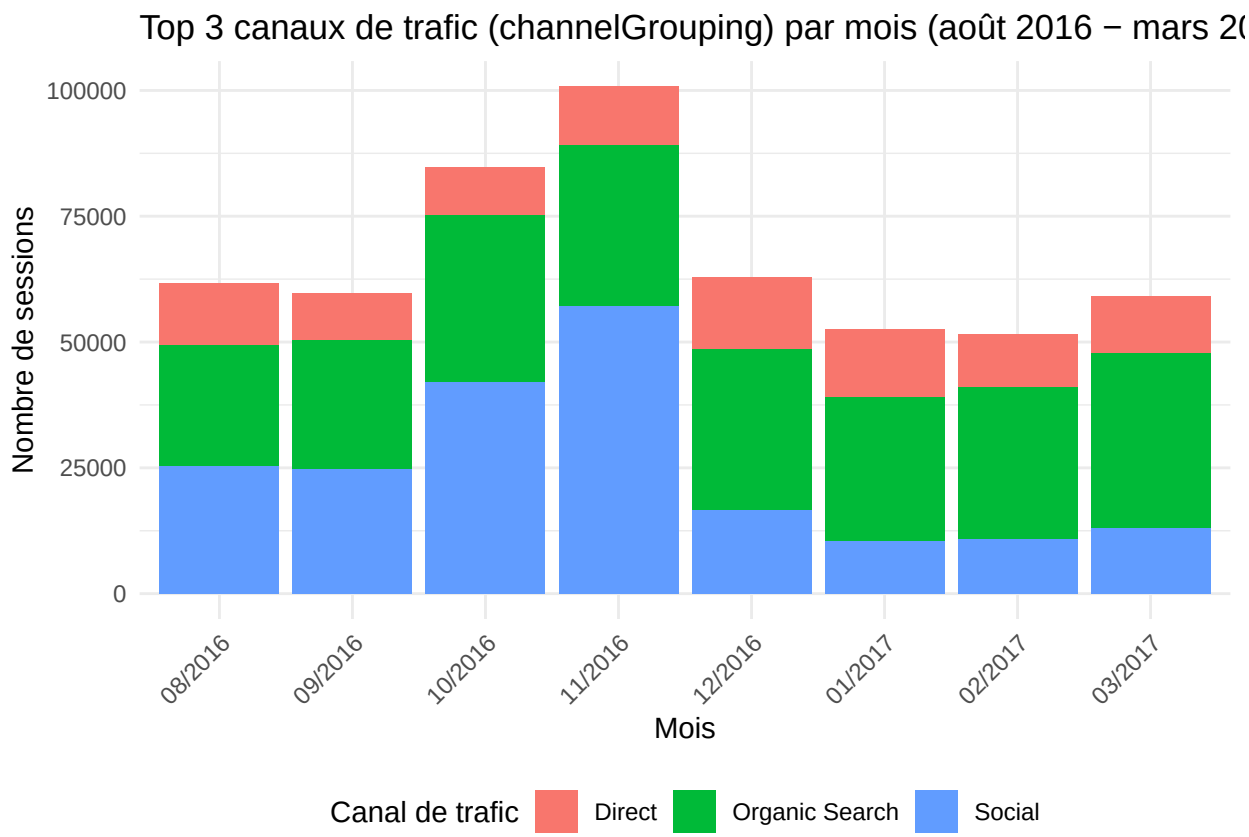
Pour mieux comprendre l'origine de cette tendance, nous allons analyser les top 3 sources de trafic, les top 3 channelGrouping, ainsi que les top 3 pays en nombre de sessions, mois par mois.

```
# Mettre date au format Date et mois au format mm/yyyy
df_monthly <- full_df %>%
  mutate(date = ymd(date)) %>%
  filter(date >= ymd("2016-08-01") & date <= ymd("2017-03-31")) %>%
  collect() %>%
  mutate(month = format(date, "%m/%Y"))
```

```r
# Calculer le top 3 des sources par mois
top_sources_monthly <- df_monthly %>%
  group_by(month, trafficSource_source) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(month, desc(count)) %>%
  group_by(month) %>%
  slice_head(n = 3) %>%
  ungroup() %>%
  mutate(month = factor(month, levels = unique(format(seq(ymd("2016-08-01"), ymd("2017-03-01"), by = "1

# Visualiser
ggplot(top_sources_monthly, aes(x = month, y = count, fill = trafficSource_source)) +
  geom_col(position = "stack") +
  labs(
    title = "Top 3 sources de trafic par mois (août 2016 - mars 2017)",
    x = "Mois",
    y = "Nombre de sessions",
    fill = "Source de trafic"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  )
```



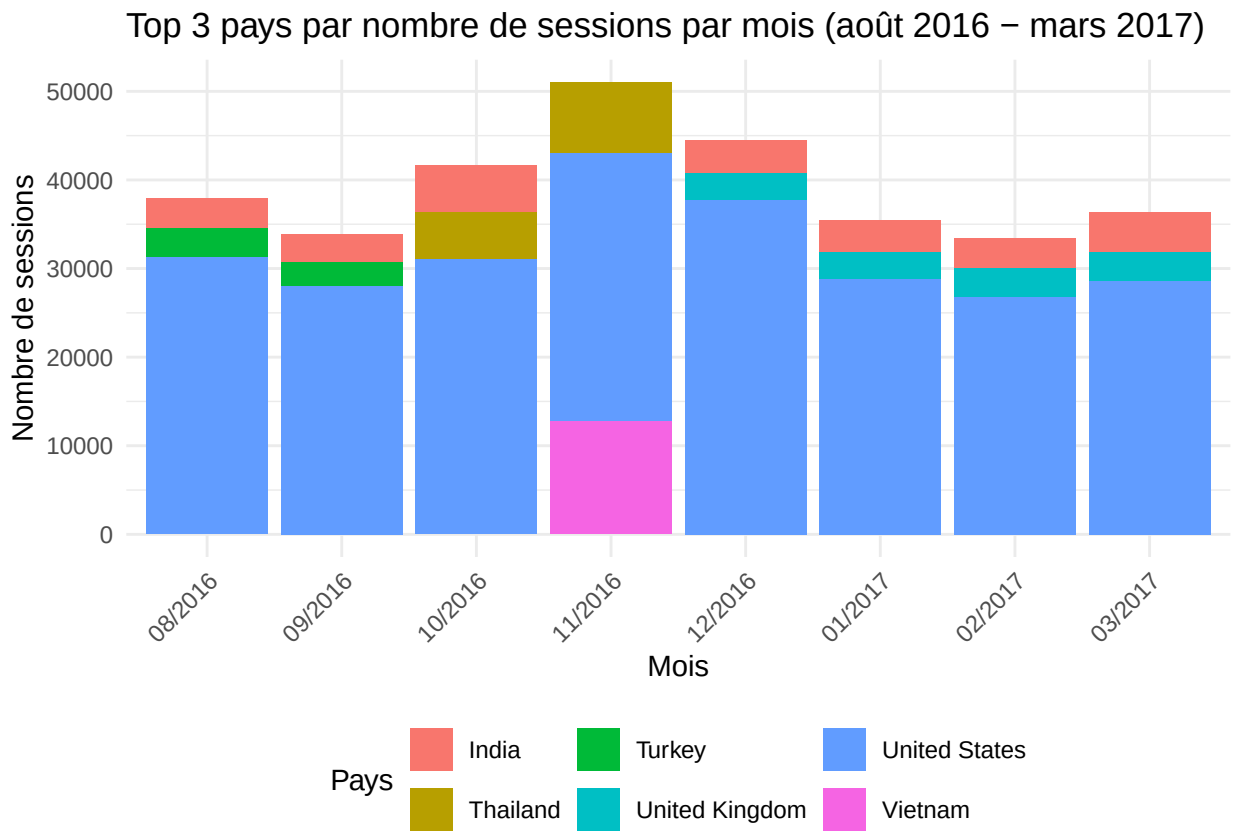
```r
# Calculer le top 3 des channelGrouping par mois
top_channels_monthly <- df_monthly %>%
  group_by(month, channelGrouping) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(month, desc(count)) %>%
  group_by(month) %>%
  slice_head(n = 3) %>%
  ungroup() %>%
  mutate(month = factor(month, levels = unique(format(seq(ymd("2016-08-01"), ymd("2017-03-01"), by = "1

# Visualisation du top 3 des canaux par mois
ggplot(top_channels_monthly, aes(x = month, y = count, fill = channelGrouping)) +
  geom_col(position = "stack") +
  labs(
    title = "Top 3 canaux de trafic (channelGrouping) par mois (août 2016 - mars 2017)",
    x = "Mois",
    y = "Nombre de sessions",
    fill = "Canal de trafic"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  )
```



L'investigation des sources de trafic et des canaux (channelGrouping) révèle les éléments suivants : - Le

trafic (direct) a doublé sur la période de 10/2016 à 02/2017. - Le trafic en provenance de google chute brutalement jusqu'à atteindre quasiment zéro, puis revient au normal a partir de 02/2017. - Le canal "Social" augmente notablement en octobre et novembre, puis décroissant à partir de 12/2016.

```r
# Calculer le top 3 des pays (geoNetwork_country) par mois
top_countries_monthly <- df_monthly %>%
  group_by(month, geoNetwork_country) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(month, desc(count)) %>%
  group_by(month) %>%
  slice_head(n = 3) %>%
  ungroup() %>%
  mutate(month = factor(month, levels = unique(format(seq(ymd("2016-08-01"), ymd("2017-03-01"), by = "1


# Ordre chronologique des mois pour le graphique
top_countries_monthly <- top_countries_monthly %>%
  mutate(month = factor(month, levels = sort(unique(month))))

# Visualisation
ggplot(top_countries_monthly, aes(x = month, y = count, fill = geoNetwork_country)) +
  geom_col(position = "stack") +
  labs(
    title = "Top 3 pays par nombre de sessions par mois (août 2016 - mars 2017)",
    x = "Mois",
    y = "Nombre de sessions",
    fill = "Pays"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  )
```



Ces éléments suggèrent que le pic de sessions observé est probablement lié à une campagne externe (emailing, réseaux sociaux ou autre), dont les liens n'ont pas été correctement balisés à l'aide de paramètres UTM. Cette absence de marquage a très certainement entraîné une mauvaise attribution du trafic, avec une hausse artificielle du canal (direct) et une baisse apparente du trafic organique en provenance de Google. En approfondissant l'analyse, on constate que cette anomalie de suivi est majoritairement associée à des connexions en provenance de la Thaïlande et du Vietnam, ce qui pourrait indiquer une diffusion ciblée ou localisée dans ces régions, sans balisage adéquat.

2. Taux de rebond

2a. Taux de rebond global et mensuel

```
bounce_summary <- full_df %>%
  mutate(
    date = ymd(date),
    total_bounces = as.numeric(as.character(total_bounces)),
    visitId = as.character(visitId)
  ) %>%
  summarise(
    total_bounces = sum(total_bounces, na.rm = TRUE),
    total_visits = n_distinct(visitId)
  ) %>%
  collect() %>%
```

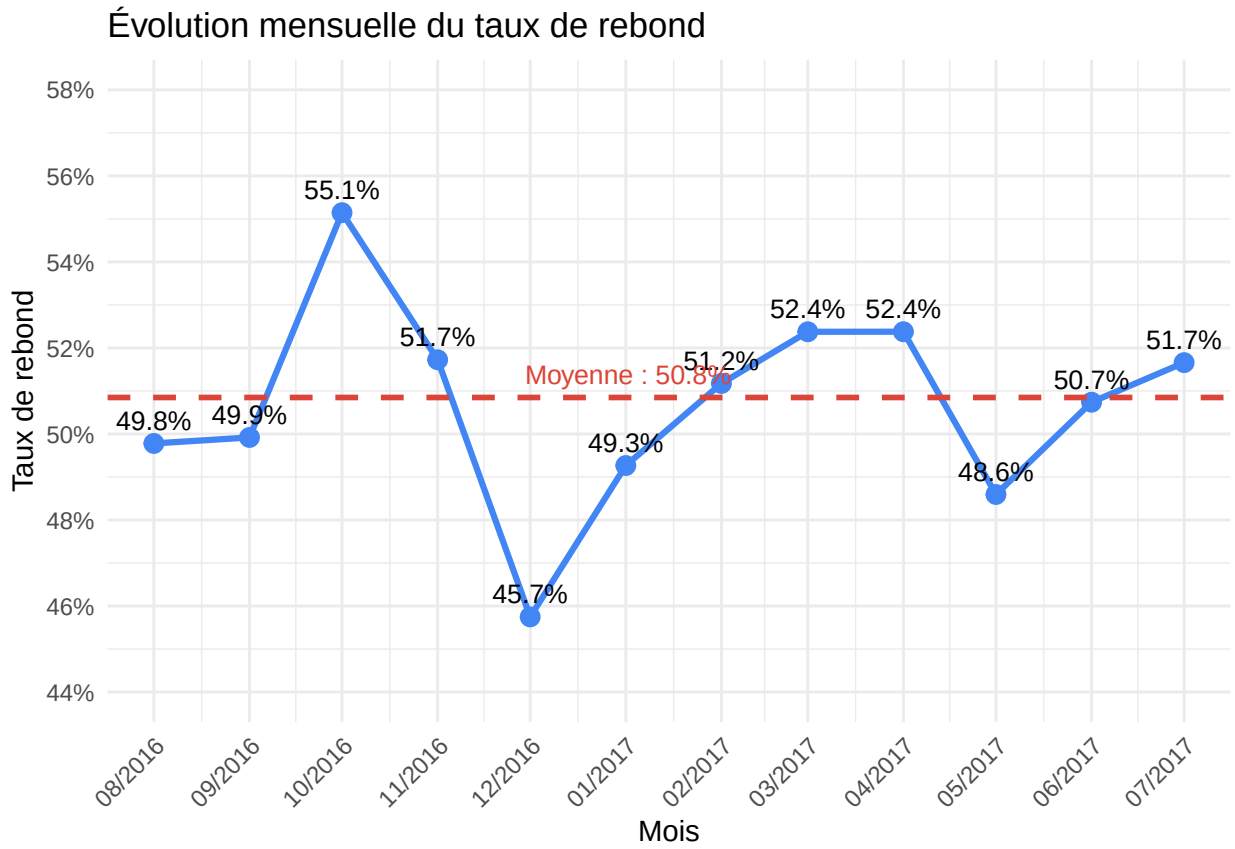
```r
  mutate(bounce_rate = total_bounces / total_visits)

avg_bounce = bounce_summary$bounce_rate  # Affiche le taux de rebond global
print(avg_bounce)
```

```
## [1] 0.5084799
```

Le taux de rebond global est d'environ 50,8%, ce qui signifie qu'un peu plus de la moitié des sessions se terminent après une seule interaction sans navigation supplémentaire sur le site.

```r
# Collection de données
bounce_monthly <- full_df %>%
  mutate(
  date = ymd(date),
  total_bounces = as.numeric(as.character(total_bounces)),
  visitId = as.character(visitId)
) %>%
  group_by(year, month) %>%
  summarise(
    total_bounces = sum(total_bounces, na.rm = TRUE),
    total_visits = n_distinct(visitId)
  ) %>%
  collect() %>%
  mutate(
    bounce_rate = total_bounces / total_visits,
    date = as.Date(paste(year, month, "01", sep = "-"))
  )

y_max <- max(bounce_monthly$bounce_rate) * 1.1
y_min <- min(bounce_monthly$bounce_rate) * 0.9

# Visualisation
ggplot(bounce_monthly, aes(x = date, y = bounce_rate)) +
  geom_line(color = custom_colors["Blue"], linewidth = 1.1) +  # ligne principale
  geom_point(color = custom_colors["Blue"], size = 3) +      # points mensuels

  # Annotations des valeurs de chaque mois
  geom_text(
    aes(label = paste0(round(bounce_rate * 100, 1), "%")),
    vjust = -0.7,
    size = 3.5,
    color = "black"
  ) +

  # Ligne de moyenne + annotation
  geom_hline(yintercept = avg_bounce,
             linetype = "dashed",
             color = custom_colors["Red"],
             size = 1) +
  annotate(
    "text",
    x = min(bounce_monthly$date),
    y = avg_bounce,
```

```r
  label = paste0("Moyenne : ", round(avg_bounce * 100, 1), "%"),
  vjust = -0.7,
  hjust = -1.8,
  color = custom_colors["Red"],
  size = 3.5
) +

# Axe Y
scale_y_continuous(
  name = "Taux de rebond",
  limits = c(0.44, 0.58),
  breaks = seq(0.44, 0.58, by = 0.02),
  labels = percent_format(accuracy = 1)
) +

# Axe X
scale_x_date(
  date_breaks = "1 month",
  date_labels = "%m/%Y",
  expand = expansion(add = c(15, 15))
) +

labs(
  title = "Évolution mensuelle du taux de rebond",
  x = "Mois"
) +
theme_minimal() +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1)
)
```



L'analyse de l'évolution mensuelle du taux de rebond révèle une rupture nette à partir d'octobre 2016, période déjà identifiée comme problématique, vraisemblablement liée à une campagne non balisée ou à une perte de suivi des sources de trafic.

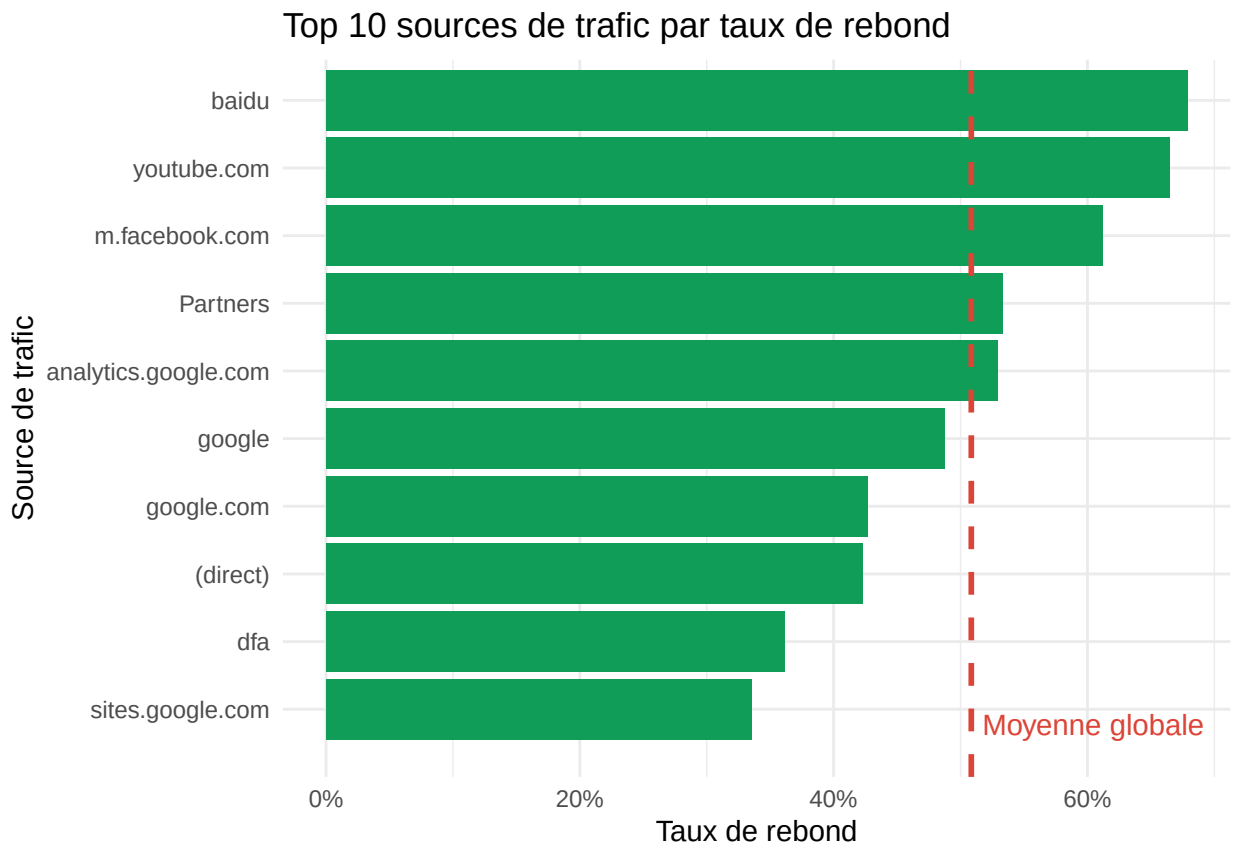
- En octobre 2016, le taux de rebond augmente brusquement pour atteindre 55,1 %, soit le niveau le plus élevé sur l'ensemble de données.
- Cette valeur reste anormalement haute jusqu'en novembre 2016, avant de chuter à 45,7 % en décembre, puis de se stabiliser à 49,3 % en janvier 2017, témoignant d'un retour progressif à un comportement plus habituel. Pour la suite de 2017, le taux de rebonds fluctue entre 48 et 52%.

Avant cette période, les taux de rebond sont relativement stables autour de 49–50%, correspondant à un comportement plus engagé des visiteurs. Cette hausse coïncide avec d'autres signaux observés sur la partie 1. Ces éléments confirment l'hypothèse d'une campagne externe non balisée (par exemple via emailing ou réseaux sociaux), générant un trafic mal attribué et souvent peu qualifié, ce qui a eu pour effet d'augmenter artificiellement le taux de rebond.

2b. Taux de rebond par source

```
# Collection de données
top_sources <- full_df %>%
  mutate(
  date = ymd(date),
  total_bounces = as.numeric(as.character(total_bounces)),
```

```r
  visitId = as.character(visitId)
) %>%
  group_by(trafficSource_source) %>%
  summarise(
    total_bounces = sum(total_bounces, na.rm = TRUE),
    total_visits = n_distinct(visitId)
  ) %>%
  collect() %>%
  mutate(bounce_rate = total_bounces / total_visits) %>%
  arrange(desc(total_visits)) %>%
  slice_head(n = 10)

# Visualisation
ggplot(top_sources, aes(x = reorder(trafficSource_source, bounce_rate), y = bounce_rate)) +
  geom_col(fill = custom_colors["Green"]) +
  geom_hline(yintercept = avg_bounce,   # ← correction ici
             linetype = "dashed",
             color = custom_colors["Red"],
             size = 1) +
  annotate(
    "text",
    x = 0,
    y = avg_bounce,
    label = paste0("Moyenne globale"),
    color = custom_colors["Red"],
    hjust = -0.05,
    vjust = -2,
    size = 4
  ) +
  coord_flip() +
  labs(
    title = "Top 10 sources de trafic par taux de rebond",
    x = "Source de trafic",
    y = "Taux de rebond"
  ) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_minimal()
```



L'analyse des **10 principales sources de trafic** met en lumière des différences notables dans la qualité du trafic, mesurée ici par le **taux de rebond** :

- **Trois sources affichent un taux de rebond anormalement élevé**, bien au-dessus de la moyenne globale (~50,8 %) :
  - `baidu` : **67,9 %** (umoteur de recherche populaire en Chine)
  - `youtube.com` : **66,5 %**
  - `m.facebook.com` : **61,2 %** (version mobile du site facebook.com)

Cela suggère un trafic peu qualifié, potentiellement dû à un manque de pertinence du contenu ou à une mauvaise correspondance entre l'intention du clic et la page d'atterrissage.

- À l'inverse, plusieurs sources issues de **l'écosystème Google** enregistrent de bons résultats, avec des taux de rebond **inférieurs ou proches de la moyenne** :
  - `google` : **48,8 %**
  - `google.com` : **42,7 %**
  - `sites.google.com` : **33,50 %**

Ces sources génèrent un trafic **plus qualifié et potentiellement mieux ciblé**.

2c. Taux de rebonds par pays.

```r
# Collection des données
# Résumer les taux de rebond par pays et calculer les codes ISO3
bounce_by_country <- full_df %>%
  mutate(
    total_bounces = as.numeric(as.character(total_bounces)),
    visitId = as.character(visitId)
  ) %>%
  collect() %>%
  group_by(geoNetwork_country) %>%
  summarise(
    total_bounces = sum(total_bounces, na.rm = TRUE),
    total_visits = n_distinct(visitId),
    .groups = "drop"
  ) %>%
  mutate(
    bounce_rate = total_bounces / total_visits,
    country_iso3 = countrycode(geoNetwork_country, origin = "country.name", destination = "iso3c")
  ) %>%
  filter(!is.na(country_iso3))

# Charger la carte du monde
world <- ne_countries(scale = "medium", returnclass = "sf")

# Fusionner les données de rebond avec la carte et préparer les tooltips
map_data <- world %>%
  left_join(bounce_by_country, by = c("iso_a3" = "country_iso3")) %>%
  mutate(
    bounce_rate_clipped = pmin(pmax(bounce_rate, 0.2), 0.8),
    tooltip_text = paste0(
      name_long, "\nTaux de rebond : ",
      ifelse(is.na(bounce_rate), "N/A", scales::percent(bounce_rate, accuracy = 0.1))
    )
  )

# Visualisation
p <- ggplot(map_data) +
  geom_sf(aes(fill = bounce_rate_clipped, text = tooltip_text), color = "white", size = 0.1) +
  scale_fill_gradientn(
    colours = c("#1a9641", "#ffffbf", "#d7191c"),
    limits = c(0.2, 0.8),
    name = "Taux de rebond",
    labels = scales::percent_format(accuracy = 1),
    na.value = "grey90"
  ) +
  labs(
    title = "Carte des taux de rebond par pays",
    subtitle = "Couleurs limitées entre 20% et 80% pour lisibilité",
    caption = "Source : Google Analytics - Données échantillon"
  ) +
  theme_minimal() +
  theme(
    legend.position = "right",
    axis.text = element_blank(),
```

```
    axis.ticks = element_blank(),
    panel.grid = element_blank(),
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 11, face = "italic")
  )

# Rendre le graphique interactif avec plotly
ggplotly(p, tooltip = "text")
```

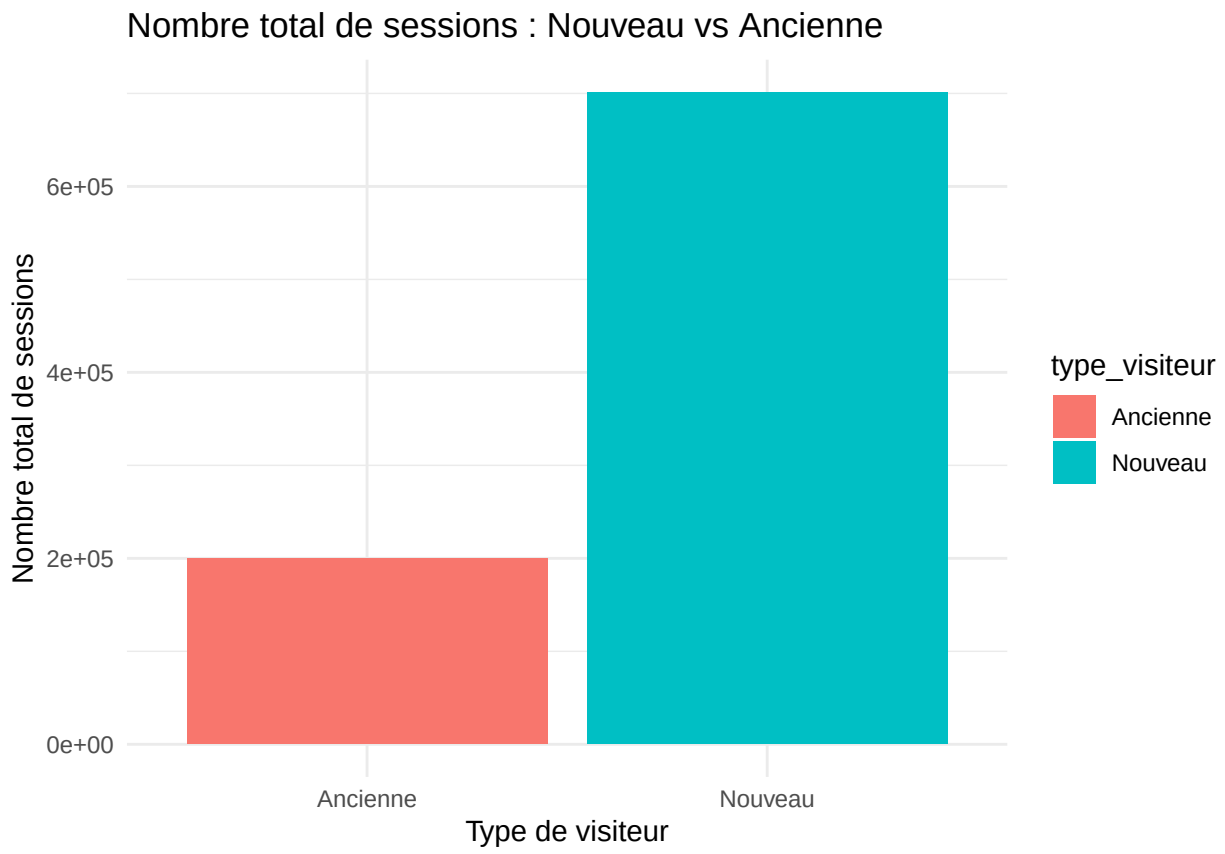
Le taux de rebond est significativement plus faible sur les marchés des États-Unis (35,4 %) et du Canada, comparé à la moyenne observée dans le reste du monde, où il dépasse souvent 60 %. Cela suggère un engagement utilisateur nettement supérieur en Amérique du Nord. Par conséquent, il est crucial de distinguer ces marchés du reste du monde lors de l'analyse des performances globales du site. Une agrégation globale masquerait des dynamiques régionales essentielles, comme les zones à fort taux de rebond (plus de 65 %) observées en Afrique, en Amérique latine et dans plusieurs pays d'Asie, ce qui pourrait orienter différemment les décisions d'optimisation UX ou marketing.

3. L'analyse des nouveaux visiteurs vs. anciennes visiteurs

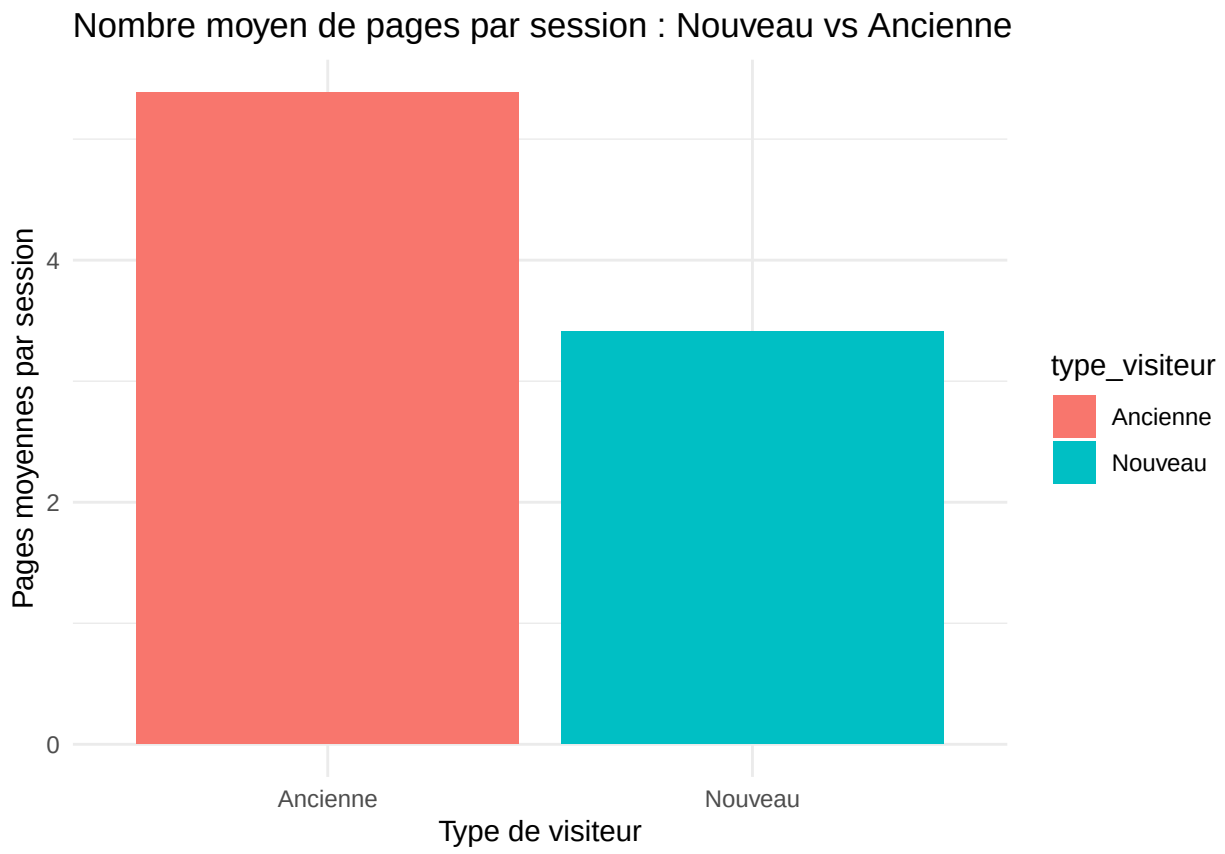
```
# Nettoyage temporaire
df$total_pageviews <- as.numeric(as.character(df$total_pageviews))
df$total_timeOnsite <- as.numeric(as.character(df$total_timeOnsite))
df$total_bounces <- as.numeric(as.character(df$total_bounces))
df$total_newVisits <- as.character(df$total_newVisits)

# Définir le type de visiteur
df$total_bounces <- ifelse(is.na(df$total_bounces), 0, df$total_bounces)

# Définir le type de visiteur
df$type_visiteur <- recode(df$total_newVisits,
                           "1" = "Nouveau",
                           .missing = "Ancienne")
```

```
# Collection des données
comparaison <- df %>%
  group_by(type_visiteur) %>%
  summarise(
    total_sessions = n(),  # count of sessions per visitor type
    total_pages_vues = sum(total_pageviews, na.rm = TRUE),
    pages_par_session_moyenne = total_pages_vues / total_sessions,  # avg pages per session
    duree_moyenne_session = mean(total_timeOnsite, na.rm = TRUE),
    taux_rebond = sum(total_bounces, na.rm = TRUE) / total_sessions,
    .groups = "drop"
  )
```

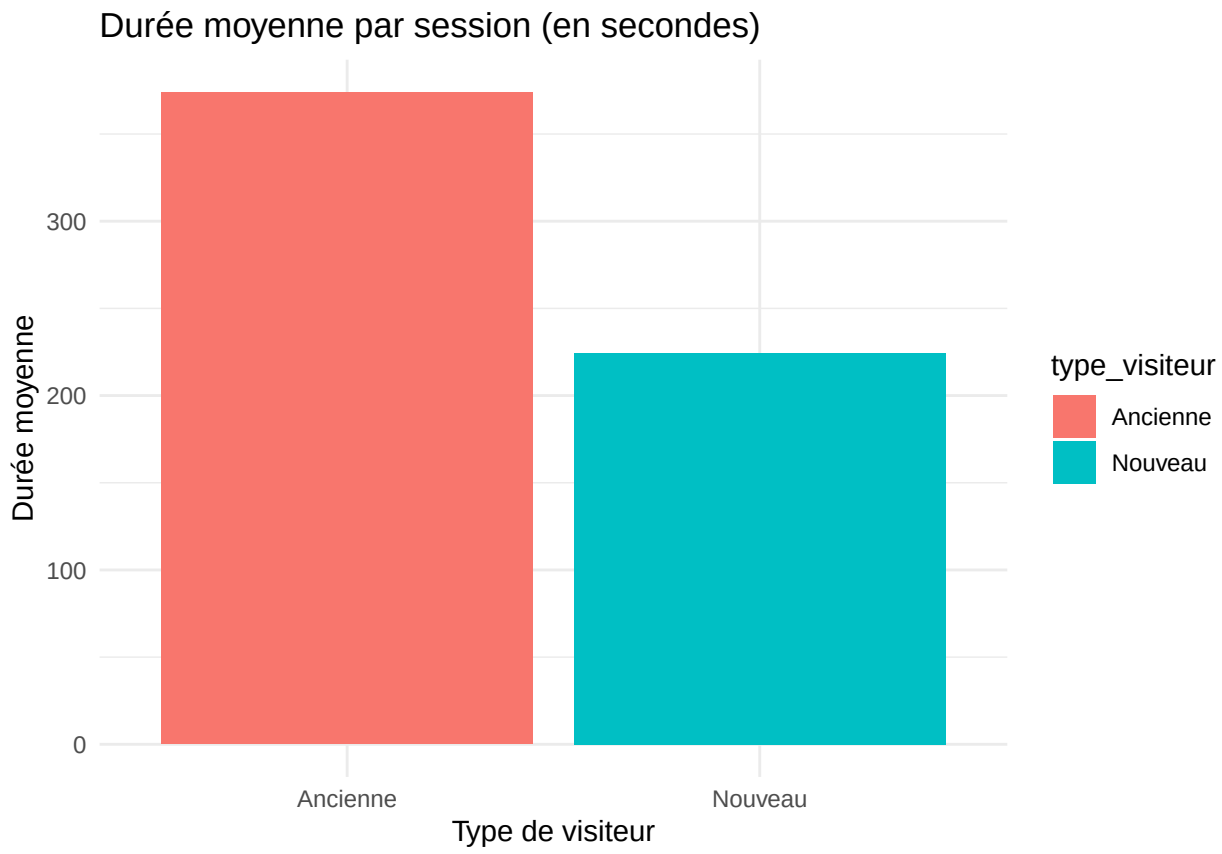
```
# Barplot du nombre total de sessions
ggplot(comparaison, aes(x = type_visiteur, y = total_sessions, fill = type_visiteur)) +
  geom_col() +
  labs(title = "Nombre total de sessions : Nouveau vs Ancienne",
       x = "Type de visiteur", y = "Nombre total de sessions") +
  theme_minimal()
```



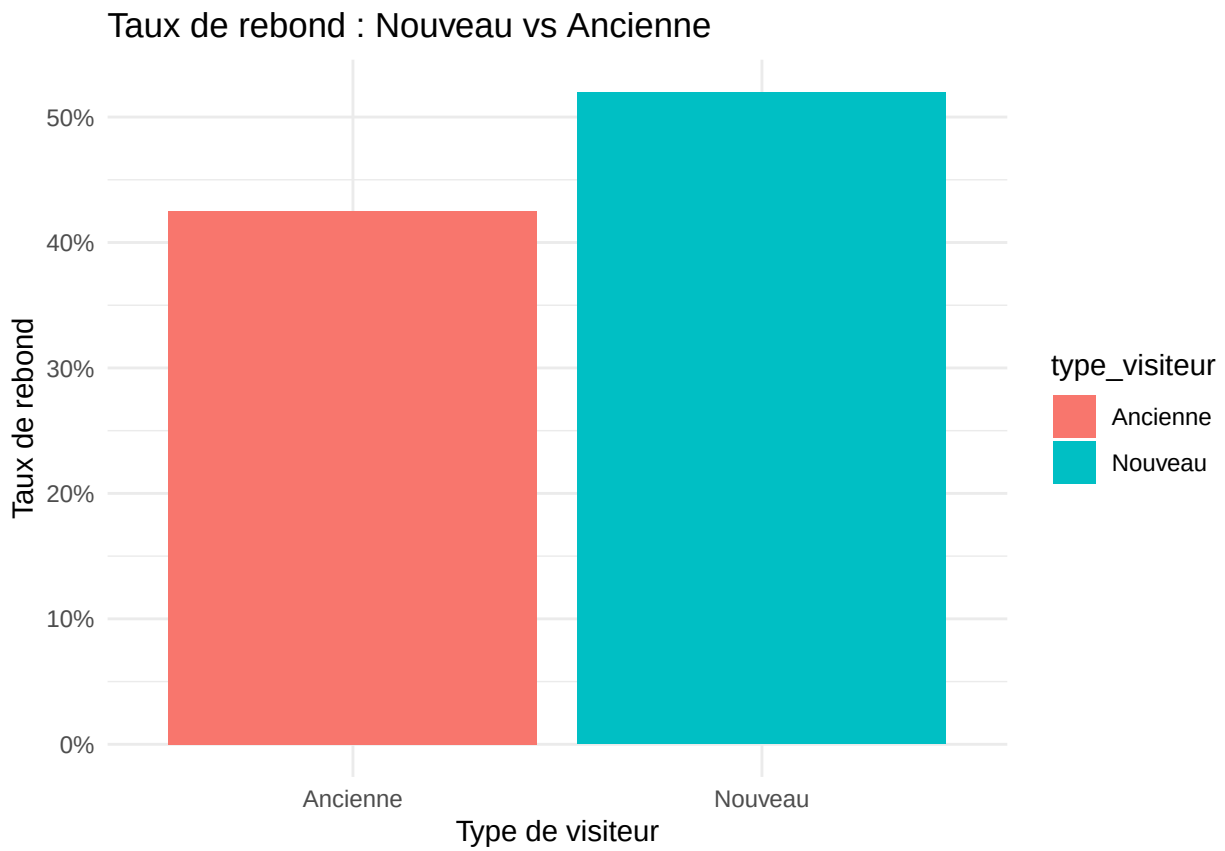

```r
# Barplot du nombre moyen de pages par session
ggplot(comparaison, aes(x = type_visiteur, y = pages_par_session_moyenne, fill = type_visiteur)) +
  geom_col() +
  labs(title = "Nombre moyen de pages par session : Nouveau vs Ancienne",
       x = "Type de visiteur", y = "Pages moyennes par session") +
  theme_minimal()
```




```r
# Barplot de la durée moyenne
ggplot(comparaison, aes(x = type_visiteur, y = duree_moyenne_session, fill = type_visiteur)) +
  geom_col() +
  labs(title = "Durée moyenne par session (en secondes)",
       x = "Type de visiteur", y = "Durée moyenne") +
  theme_minimal()
```




```r
# Barplot du taux de rebond
ggplot(comparaison, aes(x = type_visiteur, y = taux_rebond, fill = type_visiteur)) +
  geom_col() +
  labs(title = "Taux de rebond : Nouveau vs Ancienne",
       x = "Type de visiteur", y = "Taux de rebond") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_minimal()
```



Les quatre graphiques révèlent des différences intéressantes entre les visiteurs nouveaux (Nouveau) et les visiteurs récurrents (Ancienne) :

- Tout d'abord, les nouveaux visiteurs génèrent un nombre total de sessions nettement plus élevé que les visiteurs récurrents, ce qui indique qu'ils représentent une part plus importante de notre base de données.
- Cependant, le nombre de pages vues par session ainsi que la durée moyenne des sessions sont sensiblement plus élevés chez les visiteurs récurrents, ce qui suggère qu'ils s'engagent plus profondément à chaque visite.
- Enfin, le taux de rebond est plus élevé chez les nouveaux visiteurs, ce qui implique qu'une plus grande proportion d'entre eux quittent le site après avoir consulté une seule page. Pris ensemble, ces éléments suggèrent que si les nouveaux visiteurs génèrent du volume, les visiteurs récurrents apportent une qualité d'engagement supérieure grâce à des sessions plus longues et un taux de rebond plus faible. Cela souligne l'importance de mettre en place des stratégies visant à fidéliser les nouveaux visiteurs et à les convertir en utilisateurs récurrents afin d'assurer un engagement durable.