

```

# ===== Script for data analysis =====
# Create a reproducible Script that enables the analysis of the AttrakDiff / UEQ
Methodology

# Content:
# 1. Reading the Excel/CSV Data
# 2. Treating Excel/CSV Data
# 3. Making the Tables and the Graphics
# 4. Introduce the results in Rmarkdown table.

# Load the packages ----
library(tidyverse) # Data Science Tools
library(readxl) # Read a Excel File

# Reading the Data ----
## Méthodologie Attrakdiff ----
Attrakdiff <- read_csv("data/Data Experimentaion - Attrakdiff.csv")

## Méthodologie UEQ ----
UEQ <- read_csv("data/Data Experimentaion - UEQ.csv")

## function `glimpse()` : Fournit un résumé des jeux de données ----
glimpse(Attrakdiff)
glimpse(UEQ)

## function `names()` : Savoir les noms de colonnes -----
names(Attrakdiff)
names(UEQ)

## function `View(iris)` : Affiche les données dans un tableur (attention au V majuscule)
----
View(Attrakdiff)
View(UEQ)

## function `filter()` : Permet d'extraire des observations selon une condition logique -
---
Attrakdiff_donnes <- filter(Attrakdiff, Experimentation == "Innoflow")
UEQ_donnes <- filter(UEQ, Experimentation == "Itonics")

## function `select()` : Selectionn des colonnes selon leur nom ou leur fonction ----
Attrakdiff_donnes <- select(Attrakdiff_donnes, Group:Status, QP1:ATT7)
UEQ_donnes <- select(UEQ_donnes, Group:Status, EFF1 : ATT6)

## function `%>%` : Passe l'objet se trouvant à gauche comme premier argument de la
fonction se trouvant à droite. (Alt + CMD/Ctrl + M) -----
### Version 1 pour Attrakdiff
Attrakdiff_donnes <- Attrakdiff %>% filter(Experimentation == "Innoflow")
Attrakdiff_donnes_final <- Attrakdiff_donnes %>% select(Group:Status, QP1:ATT7)

### Version 2 Attrakdiff
Attrakdiff_donnes_final <-
  Attrakdiff %>%
  filter(Experimentation == "Innoflow") %>%
  select(Group:Status, QP1:ATT7)

### Version UEQ
UEQ_donnes_final <-
  UEQ %>%

```

```

filter(Experimentation == "Itonics") %>%
select(Group:Status, EFF1 : ATT6)

# Motivation Principal ----
Attrakdiff_donnes_final %>%
  ggplot() +
  aes(x = Sex, fill = Sex) +
  geom_bar(stat = "count") +
  labs(x = "",
       y = "Quantité de Participants ",
       title = "QQuestionnaire Introduction à. la Recherche",
       subtitle = paste("Quantité de Participants:" , Attrakdiff_donnes_final %>%
nrow()),
       caption = paste("Dernière mise à jour: ", Sys.time() %>% format( '%d/%m/%Y'))
  ) +
  theme_minimal(base_size = 12, base_family = "Palatino")

ggsave("figures/Participants.jpg", dpi = "print", width=5, height = 5)

# Data Analysis of the Methodologies ----
# Step 1 : Transformer les donnees dans une version longue ----
## function `gather()` : Fusionne des colonnes en lignes.

## AttrakDiff Step 1 ----
Attrakdiff.step.1 <- Attrakdiff_donnes_final %>% gather(QP1:ATT7, key = "Variable",
value = "Value")

## EUQ Step 1 ----
UEQ.step.1 <- UEQ_donnes_final %>% gather(EFF1 : ATT6, key = "Variable", value =
"Value")

# Step 2: Changer l'echelle des reponses entre le range de (-3, +3) ----
## function `mutate()` : Calcule et ajoute une ou plusieurs nouvelles variables
## function `case_when()` : Faire quelque chose en fonction d'une premise logique.

## AttrakDiff Step 2 ----

## Changing the scale of the answers (Page 16 du PDF)
## See: https://carinelallemand.files.wordpress.com/2015/09/version-franc3a7aise-attrakdiff\_lallemand\_2015.pdf

Attrakdiff.step.2 <-
  Attrakdiff.step.1 %>%
  mutate(Valeur_ajuste =
    case_when(
      Value == 7 ~ 3,
      Value == 6 ~ 2,
      Value == 5 ~ 1,
      Value == 4 ~ 0,
      Value == 3 ~ -1,
      Value == 2 ~ -2,
      Value == 1 ~ -3,
      TRUE ~ Value
    )
  )

## UEQ Step 2 ----
UEQ.step.2 <-
  UEQ.step.1 %>%
  mutate(Valeur_ajuste=
    case_when(
      Value == 7 ~ 3,

```

```

        Value == 6 ~ 2,
        Value == 5 ~ 1,
        Value == 4 ~ 0,
        Value == 3 ~ -1,
        Value == 2 ~ -2,
        Value == 1 ~ -3,
        TRUE ~ Value
    )
)

# Step 3: Inverser certain valeurs en fonction de la Méthodplogie ----
## AttrakDiff Step 3 ----
### Approach 1 : en faisant pour chaque Variable ----
Attrakdiff.step.3 <-
  Attrakdiff.step.2 %>%
  mutate(Valeur_inverse= case_when(
    Variable == "QP1" ~ Valeur_ajuste*(-1),
    Variable == "QP2" ~ Valeur_ajuste*(-1),
    Variable == "QP3" ~ Valeur_ajuste*(-1),
    Variable == "QP5" ~ Valeur_ajuste*(-1),
    Variable == "ATT1" ~ Valeur_ajuste*(-1),
    Variable == "ATT3" ~ Valeur_ajuste*(-1),
    Variable == "ATT5" ~ Valeur_ajuste*(-1),
    Variable == "ATT7" ~ Valeur_ajuste*(-1),
    Variable == "QHS1" ~ Valeur_ajuste*(-1),
    Variable == "QHS3" ~ Valeur_ajuste*(-1),
    Variable == "QHS4" ~ Valeur_ajuste*(-1),
    Variable == "QHS7" ~ Valeur_ajuste*(-1),
    Variable == "QHS1" ~ Valeur_ajuste*(-1),
    Variable == "QHS3" ~ Valeur_ajuste*(-1),
    Variable == "QHS4" ~ Valeur_ajuste*(-1),
    Variable == "QHS7" ~ Valeur_ajuste*(-1),
    Variable == "QHS7" ~ Valeur_ajuste*(-1),
    TRUE ~ Valeur_ajuste
  )
)

### Approach 2 : Definir une vecteur avec les colonnesà inverser ----
toInvert <- c("QP1", "QP2", "QP3", "QP5",
              "ATT1", "ATT3", "ATT5", "ATT7",
              "QHS1", "QHS3", "QHS4", "QHS7",
              "QHI2", "QHI3", "QHI6")

Attrakdiff.step.3 <-
  Attrakdiff.step.2 %>%
  mutate(Valeur_inverse =
    case_when(
      Variable %in% toInvert ~ Valeur_ajuste*(-1),
      TRUE ~ Valeur_ajuste
    )
  )

rm(toInvert)

## UEQ Step 3 ----
### Approach 1 : en faisant pour chaque Variable ----
UEQ.step.3 <-
  UEQ.step.2 %>%
  mutate(Valeur_inverse= case_when(
    Variable == "EFF1" ~ Valeur_ajuste*(-1),
    Variable == "EFF4" ~ Valeur_ajuste*(-1),
    Variable == "PERS2" ~ Valeur_ajuste*(-1),
    Variable == "PERS4" ~ Valeur_ajuste*(-1),
    Variable == "DEP3" ~ Valeur_ajuste*(-1),
    Variable == "DEP4" ~ Valeur_ajuste*(-1),
    Variable == "STIM1" ~ Valeur_ajuste*(-1),

```

```

Variable == "STIM4" ~ Valeur_ajuste*(-1),
Variable == "NOV1" ~ Valeur_ajuste*(-1),
Variable == "NOV2" ~ Valeur_ajuste*(-1),
Variable == "ATT2" ~ Valeur_ajuste*(-1),
Variable == "ATT5" ~ Valeur_ajuste*(-1),
Variable == "ATT6" ~ Valeur_ajuste*(-1),
TRUE ~ Valeur_ajuste
)
)

### Approach 2 : Définir une vecteur avec les colonnes à inverser ----
toInvert <-
  c("EFF1", "EFF4",
    "PERS2", "PERS4",
    "DEP3", "DEP4",
    "STIM1", "STIM4",
    "NOV1", "NOV2",
    "ATT2", "ATT5", "ATT6")

UEQ.step.3 <-
  UEQ.step.2 %>%
  mutate(
    Valeur_inverse =
      case_when(
        Variable %in% toInvert ~ Valeur_ajuste*(-1),
        TRUE ~ Valeur_ajuste
      )
  )

rm(toInvert)

# Step 4: Grouper les dimensions du Modèle correspondant ----
## AttrakDiff Step 4 ----
Attrakdiff.step.4 <-
  Attrakdiff.step.3 %>%
  mutate(
    Facteurs =
      case_when(
        str_detect(Variable, "QP") ~ "Qualité Pragmatique (QP)",
        str_detect(Variable, "QHS") ~ "Qualité Hédonique - Stimulation (QH-S)",
        str_detect(Variable, "QHI") ~ "Qualité Hédonique - Identité (QH-I)",
        str_detect(Variable, "ATT") ~ "Attractivité Globale (ATT)",
        TRUE ~ "ATTENTION, Erreur dans la bases de données"
      )
  )

## UEQ Step 4 ----
UEQ.step.4 <-
  UEQ.step.3 %>%
  mutate(
    Facteurs =
      case_when(
        str_detect(Variable, "ATT") ~ "Attraction",
        str_detect(Variable, "PERS") ~ "Compréhensibilité",
        str_detect(Variable, "DEP") ~ "Contrôlabilité",
        str_detect(Variable, "EFF") ~ "Efficacité",
        str_detect(Variable, "NOV") ~ "Originalité",
        str_detect(Variable, "STIM") ~ "Stimulation",
        TRUE ~ "ATTENTION, Erreur dans la bases de données"
      )
  )

# Step 5: Calculer les valeurs moyennes et leur déviation standards ----
## fonction `group_by()` : Regroupe les observations par rapport à une ou plusieurs
variables

```

```

# More Info: # https://www.r-graph-gallery.com/4-barplot-with-error-bar.html

## AttrakDiff Step 5 ----
Attrakdiff.step.5 <-
  Attrakdiff.step.4 %>%
  group_by(Facteurs) %>%
  summarise(Mean = mean(Valeur_inverse),
            Stand_dev = sd(Valeur_inverse),
            Se = Stand_dev / sqrt(length(Valeur_inverse))
            )

## UEQ Step 5 ----
UEQ.step.5 <-
  UEQ.step.4 %>%
  group_by(Facteurs) %>%
  summarise(Mean = mean(Valeur_inverse),
            Stand_dev = sd(Valeur_inverse),
            Se = Stand_dev / sqrt(length(Valeur_inverse))
            )

# Global Results ----
Attrakdiff.Results <- list()
UEQ.Results <- list()

### AttrakDiff Graphique 1 ----
Attrakdiff.Results$Graphique.1 <-
  ggplot() +
  aes(x= Facteurs, y = Mean, group = 1) +
  geom_point() +
  geom_line() +
  #geom_bar(stat = "identity") +
  geom_errorbar( aes(x=Facteurs, ymin = Mean - Se,
                    ymax = Mean + Se ),
                width=0.1, colour="orange", alpha=0.9, size=0.5) +
  scale_x_discrete( name = "AttrakDiff Resutls") +
  scale_y_continuous(limits=c(-3,3)) +
  geom_hline(yintercept=0, linetype="dashed", color = "blue") +
  labs(x = "",
        y = "Moyenne ",
        title = "AtrakDiff Profile pour les XXX",
        subtitle = paste("Total of answers:" , Attrakdiff_donnes_final %>% nrow()),
        caption = paste0("Dernière mise à jour: ", Sys.time() %>% format( '%d/%m/%Y'))
        ) +
  theme_minimal(base_size = 12, base_family = "Palatino")

### Saving the global Results
Attrakdiff.Results$Graphique.1
Attrakdiff.Results$Tableau_1 <- Attrakdiff.step.5

# Saving the File
#ggsave("Figures/AttrakDiff-1.jpg", width = 11, height = 5, dpi="print" )
#write_csv(Results$Tableau_I, "tables/tableu_1.csv")

## Exportez les donnes sous Excel
#library("writexl")
#write_xlsx(Results$Tableau_I, "tables/tableu_1.xlsx")

### UEQ Graphique 1 ----
UEQ.Results$Graphique.1 <-
  UEQ.step.5 %>%
  ggplot() +

```

```

aes( x=Facteurs, y=Mean) +
geom_point() +
geom_bar(stat = "identity") +
geom_errorbar( aes(x=Facteurs,
                    ymin = Mean - Se,
                    ymax = Mean + Se ),
                width=0.1, colour="orange", alpha=0.9, size=0.5) +
scale_y_continuous(limits = c(-3,3), breaks = c(-3:3)) +
labs(x = "",
      y = "Level ",
      title = "UEQ Profile for XXX",
      subtitle = paste("Quantité de Participants:" , UEQ_donnees_final %>% nrow()),
      caption = paste("Dernière mise à jour: ", Sys.time() %>% format( '%d/%m/%Y'))
) +
theme_minimal(base_size = 12, base_family = "Palatino")

# Saving the File
#ggsave("figures/UEQ-1.jpg", width = 10, height = 5, dpi="print" )

# Graphique 2: ----
## function `left_join()`: Fusionne des tableaux

## AttrakDiff Graphique 2 ----
### Charger les données des paramètres d'Attrakdiff du Google Docs
Attrakdiff.parameters <- read_csv("data/Data Experimentation - Parameters
Attrakdiff.csv")
Attrakdiff.parameters <-
  Attrakdiff.parameters %>% arrange(Variable)

Attrakdiff.step.6 <-
  Attrakdiff.step.4 %>% group_by(Facteurs,Variable) %>%
  summarise(Mean = mean(Valeur_inverse)) %>%
  left_join(Attrakdiff.parameters, by="Variable")

# Faire que certaines variables deviennent des variables catégoriques.
Attrakdiff.step.7 <-
  Attrakdiff.step.6 %>%
  mutate(
    Variable = factor(Variable),
    Scale = factor(Scale),
    Left = factor(Left),
    Right = factor(Right)
  )

Attrakdiff.Results$Graphique.2 <-
  Attrakdiff.step.7 %>%
  ggplot() +
  aes(x = Variable, y=Mean, group=1) +
  geom_line( color="grey" ) +
  geom_point() +
  coord_flip() +
  annotate("text", x = 1:28, y = -4, label = Attrakdiff.parameters$Right ) +
  annotate("text", x = 1:28, y = 4, label = Attrakdiff.parameters$Left ) +
  scale_y_continuous(name="Moyenne", breaks=seq(-3,3,1), limits=c(-5, 5)) +
  theme_minimal(base_size = 12, base_family = "Palatino") +
  annotate("rect", xmin=c(1,8,15,22), xmax=c(7,14,21,28),
            ymin=rep(-3,4), ymax=rep(3, 4),
            alpha = .1 , fill = c("blue", "red", "grey","green")) +
  annotate("text",
            y = c(2, 2, 2, 2),
            x = c(4, 11, 19, 26),

```

```

        label = c("Attractivité \n globale",
                  "Qualité \n hédonique - identification",
                  "Qualité \n hédonique - stimulation",
                  "Qualité \n pragmatique"),
        family = "Palatino", fontface = 3, size=3) +
labs(x = "",
     y = "Level ",
     title = "AtrakDiff Profile",
     subtitle = paste("Total of answers:" , Attrakdiff_donnes_final %>% nrow() )
     ) +
theme(
  legend.position = "right",
  panel.border = element_blank(),
  panel.spacing = unit(0.1, "lines"),
  strip.text.x = element_text(size = 18, family = "Palatino")
)

# Saving the File
#ggsave("figures/AttrakDiff-2.jpg", width = 7, height = 7, dpi="print" )

## UEQ Graphique 2 ----
## Charger les donnes des parametres d'Attrakdiff du Google Docs
UEQ_parameters <- read_csv("data/Data Experimentaion - Parameters UEQ.csv")

UEQ.step.6 <-
  UEQ.step.4 %>%
  group_by( Variable ) %>%
  summarise( Mean = mean(Valeur_inverse)) %>%
  left_join(UEQ_parameters, by="Variable")

# Faire que certain variables deviennent des variables catégoriques.
UEQ.step.7 <-
  UEQ.step.6 %>%
  mutate(
    Variable = factor(Variable),
    Scale = factor(Scale)
  )

UEQ.Results$Graphique.2 <-
  UEQ.step.7 %>%
  ggplot() +
  aes(x = Variable, y=Mean, group =1) +
  geom_line( color="grey" ) +
  geom_point() +
  coord_flip() +
  annotate("text", x = 1:26, y = -4, label = UEQ.step.7$Left) +
  annotate("text", x = 1:26, y = 4, label = UEQ.step.7$Right) +
  scale_y_continuous(name="Moyenne", breaks=seq(-3,3,1), limits=c(-5, 5)) +
  annotate("rect", xmin=c(1,7,11,15, 19, 23), xmax=c(6,10,14,18, 22, 26),
           ymin=rep(-3,6), ymax=rep(3, 6),
           alpha = .1 , fill = c("blue", "red", "grey","green", "orange", "yellow")) +
  annotate("text",
           y = c(2, 2, 2, 2, 2, 2),
           x = c(4, 8, 12, 16, 20, 23),
           label = c("Attraction",
                     "Contrôlabilité",
                     "Efficacité",
                     "Originalité",
                     "Compréhensibilité",
                     "Stimulation"
                     ),
           family = "Palatino", fontface = 3, size=3) +
  theme_minimal(base_size = 12, base_family = "Palatino") +
  labs(x = "",
       y = "Level ",
       title = "UEQ Profile",

```

```

      subtitle = paste("Total of answers:" , UEQ_donnees_final %>% nrow()),
      caption = "Group X") +
theme(
  legend.position = "right",
  panel.border = element_blank(),
  panel.spacing = unit(0.1, "lines"),
  strip.text.x = element_text(size = 18, family = "Palatino")
)

# Saving the File
#ggsave("figures/UEQ-2.jpg", width = 8, height = 8, dpi="print" )

# Graphique 3 ----
## Attrakdiff Graphique 3 ----
Attrakdiff.step.8 <-
  Attrakdiff.step.4 %>%
  filter(Facteurs == "Qualité Hédonique - Stimulation (QH-S)" |
         Facteurs == "Qualité Hédonique - Identité (QH-I)") %>%
  summarise(QH = mean(Valeur_inverse),
            QH_sd = sd(Valeur_inverse),
            QH_IC_min = t.test(Valeur_inverse)$conf.int[1], # see
https://larmarange.github.io/analyse-R/intervalles-de-confiance.html
            QH_IC_max = t.test(Valeur_inverse)$conf.int[2]
  )

Attrakdiff.step.9 <-
  Attrakdiff.step.4 %>%
  filter(Facteurs == "Qualité Pragmatique (QP)") %>%
  summarise(QP = mean(Valeur_inverse),
            QP_sd = sd(Valeur_inverse),
            QP_IC_min = t.test(Valeur_inverse)$conf.int[1], # see
https://larmarange.github.io/analyse-R/intervalles-de-confiance.html
            QP_IC_max = t.test(Valeur_inverse)$conf.int[2]
  )

Attrakdiff.Table <- tibble(Attrakdiff.step.8, Attrakdiff.step.9)
names(Attrakdiff.Table)

Attrakdiff.Results$Graphique.3 <-
  Attrakdiff.Table %>%
  ggplot() +
  aes(x=QP, y=QH) +
  geom_point()+
  ylim(-3,3)+ xlim(-3,3) +
  geom_hline(yintercept=c(-1,1))+
  geom_vline(xintercept=c(-1,1)) +
  annotate("rect",
          xmin = Attrakdiff.Table$QP_IC_min, xmax = Attrakdiff.Table$QP_IC_max,
          ymin = Attrakdiff.Table$QH_IC_min, ymax = Attrakdiff.Table$QH_IC_max,
          alpha = .5 , fill = c("blue")) +
  annotate("rect", xmin=c(-1), xmax=c(1),
          ymin=c(-1), ymax=c(1),
          alpha = .1 , fill = c("#009999")) +
  annotate("text",
          y = c(0.5),
          x = c(0),
          label = c("Neutre"),
          family = "Palatino", fontface = 3, size=4) +
  labs(title = "Global AttrakDiff ",
       #subtitle = paste("Total of answers:" , total),
       x = "Qualité Pragmatique",
       y = "Qualité Hedonique ") +
  theme_minimal(base_size = 10, base_family = "Palatino")
#ggsave("figures/AttrakDiff-3.jpg", width = 5, height = 7, dpi="print" )

rm(Attrakdiff.Table )

```



```

## UEQ Graphique 3 ----
UEQ.step.8 <-
  UEQ.step.4 %>%
  mutate(
    Global_scale =
      case_when(
        str_detect(Facteurs, "Compréhensibilité") ~ "Qualité Pragmatique (QP)",
        str_detect(Facteurs, "Efficacité") ~ "Qualité Pragmatique (QP)",
        str_detect(Facteurs, "Contrôlabilité") ~ "Qualité Pragmatique (QP)",
        str_detect(Facteurs, "Originalité") ~ "Qualité Hédonique",
        str_detect(Facteurs, "Stimulation") ~ "Qualité Hédonique",
        str_detect(Facteurs, "Attraction") ~ "Attraction",
        TRUE ~ "ATTENTION"
      )
  )

## Identification des Valeurs pour chaque composant du modèle UEQ
UEQ.step.9 <-
  UEQ.step.8 %>%
  group_by(Global_scale) %>%
  summarise(Moyenne = mean(Valeur_inverse),
            Std = sd(Valeur_inverse),
            Se = Std / sqrt(length(Valeur_inverse)))

UEQ.Results$Graphique.3 <-
  UEQ.step.9 %>%
  ggplot() +
  aes(x= Global_scale, y=Moyenne) +
  geom_bar(stat = "identity") +
  geom_errorbar( aes(x=Global_scale,
                    ymin = Moyenne - Se,
                    ymax = Moyenne + Se ),
                width=0.1, colour="orange", alpha=0.9, size=0.5
  ) +
  scale_y_continuous(limits = c(-3,3), breaks = c(-3:3)) +
  labs(x = "UEQ Results",
       y = "Level ",
       title = "UEQ Profile for XXX",
       subtitle = paste("Total of answers:" , UEQ_donnees_final %>% nrow()),
       caption = paste("Dernière mise à jour: ", Sys.time() %>% format( '%d/%m/%Y'))
  ) +
  theme_minimal(base_size = 12, base_family = "Palatino")

# Saving the File
#ggsave("Figures/UEQ-3.jpg", width = 10, height = 5, dpi="print" )

```