City Deal

Lekan Ali

2024-07-04

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# read in demographic dataset
df_demographic <- read_excel("Demographic.xlsx")

# read in activities dataset
df_activities <- read_excel("Activities.xlsx")

# read in outcome dataset
df_outcomes <- read_excel("Outcomes.xlsx")
```

Including Plots

You can also embed plots, for example:

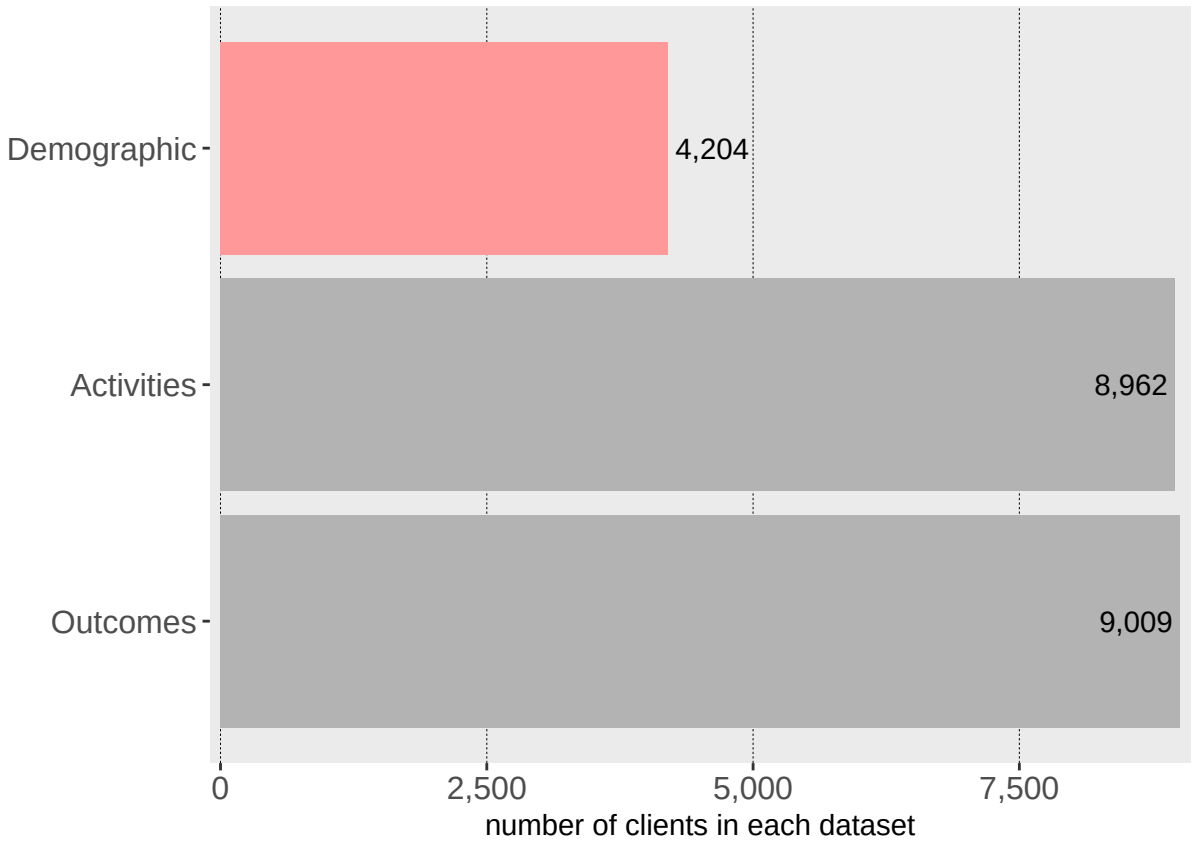
```
options(width = 300)

# 1.1 All three datasets have a common feature name (which is the ID of the client) but apparently this

names(df_activities)[1] <- "Unique ID"

# 1.2 create a function that picks only unique client and then add the name of the dataset to the final
number_of_clients <- function(df, group, dataset) {
  df %>% group_by({{group}}) %>%
    mutate(instances = row_number()) %>% # this code assigns a number to each instance or if a client e
    filter(instances == 1) %>% # this picks only the first instance such that duplicates are excluded
    #dplyr::select({{group}}) %>%
    mutate(dataset = {{dataset}}) %>%
    dplyr::select({{group}}, dataset) # create a new column, the new column should have the name of the
}

# 1.3 Demographic clients (unique clients)
df_demographic_clients <- number_of_clients(df_demographic,`Unique ID`,"Demographic")

# 1.4 Activities clients (unique clients)
```

```r
df_activities_clients <- number_of_clients(df_activities,`Unique ID`, "Activities")


# 1.5 Outcomes clients (unique clients)
df_outcomes_clients <- number_of_clients(df_outcomes,`Unique ID`, "Outcomes")
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

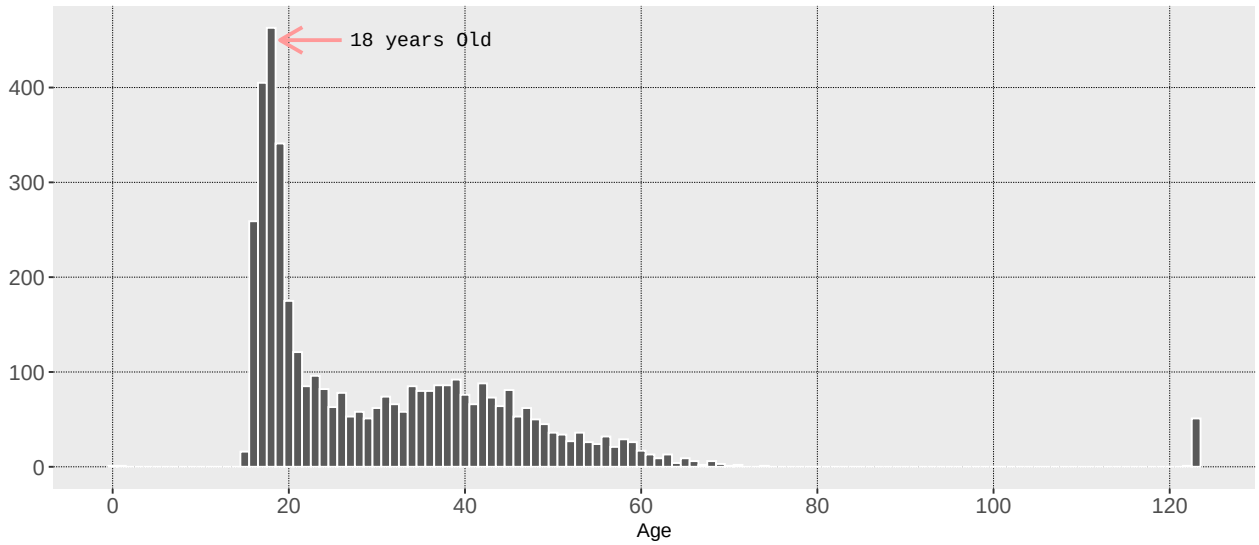
```r
# 2.1 combine the three tables such that they are ontop of each other
client_id_and_dataset <- rbind(df_demographic_clients, df_activities_clients, df_outcomes_clients)

# 2.2
client_id_and_dataset$dataset <- factor(client_id_and_dataset$dataset, levels = c("Demographic", "Activi

# 3 count the number of clients in each dataset
clients_per_dataset <- client_id_and_dataset %>%
    group_by(dataset) %>%
    summarise(count = n())
```

```r
# 4 plot the number of client in each dataset
clients_per_dataset %>%
  ggplot(aes(fct_rev(dataset), count)) +
  geom_col(fill = "grey70") +
  geom_col(data = clients_per_dataset %>% filter(dataset == "Demographic")
           , fill = "#FF9999") +
  coord_flip() +
  geom_text(data = clients_per_dataset %>%
              filter(count > 5000), aes(label = scales::comma(count), hjust = 1.1)) +
  geom_text(data = clients_per_dataset %>%
              filter(count < 5000), aes(label = scales::comma(count), hjust = -0.1)) +
  theme(panel.grid.major.y = element_blank()
        , panel.grid.minor.x = element_blank()
        , panel.grid.major.x = element_line(linetype = 2, color = "black", linewidth = 0.1)
        , axis.text = element_text(size = 12)) +
  scale_y_continuous(expand = c(0,100)
                     , labels = scales::comma) + # add a thousand separator to the values in the x axis
  labs(y = "number of clients in each dataset"
       , x = "") # label only the y axis (since the plot is rotated, y axis is now x axis)
```



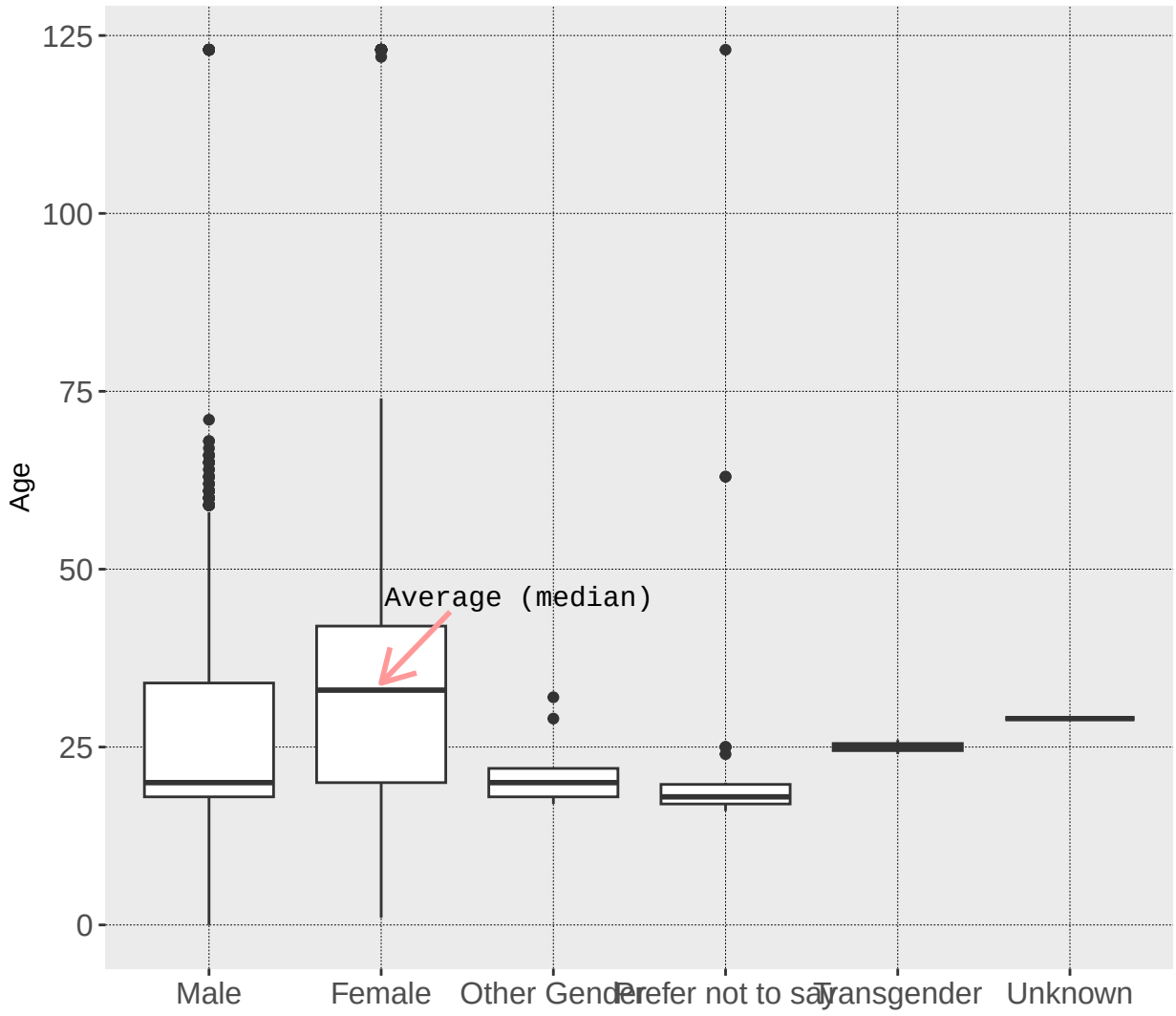

```
## Demographic Data Exploration ###
df_demographic_unique <- df_demographic %>%
    group_by(`Unique ID`) %>%
    mutate(instances = row_number()) %>%
    filter(instances == 1)


# explore age distribution
ggplot(df_demographic_unique, aes(Age)) + geom_histogram(binwidth = 1, color = "white") + scale_x_contir
    panel.grid.major = element_line(linetype = 2, color = "black", linewidth = 0.08), axis.text = elemer
    labs(y = "", caption = "Source: Demographic dataset") + annotate("text", x = 35, y = 450, label = "
    size = 1, arrow = arrow(length = unit(0.2, "inches")))
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.
```



Source: Demographic dataset

```
# explore age distribution by sex change sex from character column to a factor column
df_demographic_unique$Sex <- factor(df_demographic_unique$Sex, levels = c("Male", "Female", "Other Gend

# plot age distribution using box plot
ggplot(df_demographic_unique, aes(Age, Sex)) + geom_boxplot() + coord_flip() + theme(panel.grid.minor.x
    color = "black", linewidth = 0.08), axis.text = element_text(size = 12), plot.caption = element_tex
    family = "mono") + annotate("segment", x = 44, xend = 34, y = 2.4, yend = 2, colour = "#FF9999", si
```



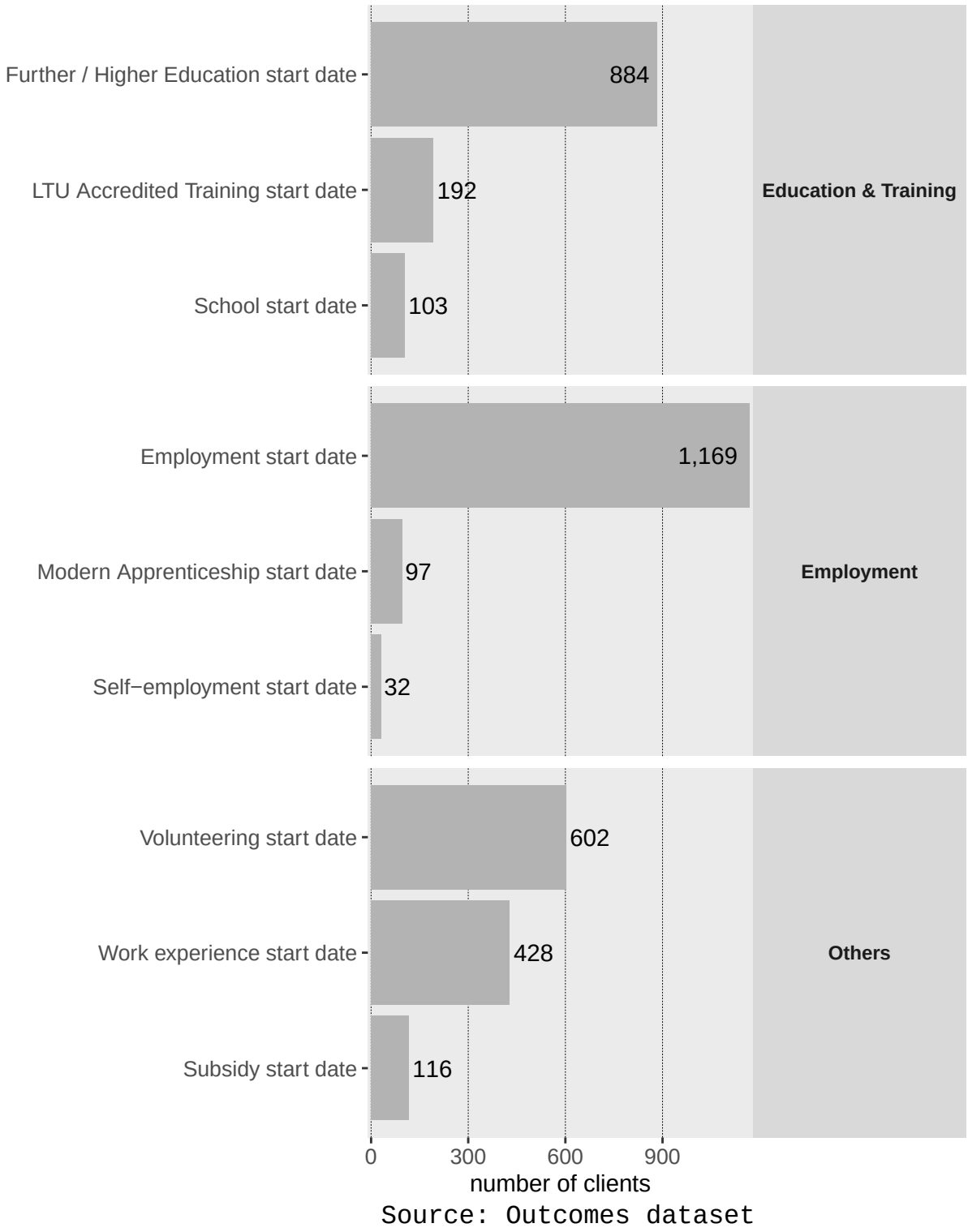
Source: Demographic dataset

```
## Exploring Outcomes

df_outcomes_unique <- df_outcomes %>%
    group_by(`Unique ID`) %>%
    mutate(instances = row_number()) %>%
    filter(instances == 1)


df3 <- df_outcomes_unique %>%
    mutate(`Subsidy start date` = case_when(`Subsidy start date` == "NULL" ~ 0, TRUE ~ 1), `Subsidy end
        "NULL" ~ 0, TRUE ~ 1), `Self-employment start date` = case_when(`Self-employment start date` ==
        "NULL" ~ 0, TRUE ~ 1), `Work experience start date` = case_when(`Work experience start date` ==
        "NULL" ~ 0, TRUE ~ 1), `Volunteering start date` = case_when(`Volunteering start date` == "NULL
        0, TRUE ~ 1), `LTU Accredited Training start date` = case_when(`LTU Accredited Training start da
        "NULL" ~ 0, TRUE ~ 1), `Further / Higher Education start date` = case_when(`Further / Higher Edu
        "NULL" ~ 0, TRUE ~ 1), `Date FE/HE qualification achieved` = case_when(`Date FE/HE qualificatio
        0, TRUE ~ 1))
```

```r
# select specific columns (that can be regarded as positive outcomes, mostly features with start date)
df4 <- df3 %>%
    dplyr::select(1, 3, 5, 6, 7, 40, 42, 44, 48, 56)

# using pivot longer, transpose df4 such that we have lesser columns
df5 <- df4 %>%
    pivot_longer(cols = -1, names_to = "outcomes")  # this means pivot/transpose all columns except the

# create a new column, add group the outcomes into three different groups of Education/training, employ
df6 <- df5 %>%
    mutate(Outcome_type = case_when(outcomes == "Subsidy start date" | outcomes == "Subsidy end date" |
        "Volunteering start date" | outcomes == "Volunteering completion date" ~ "Others", outcomes ==
        "Employment", TRUE ~ "Education & Training"))  # group the outcomes to either employment, educa


# sum each outcomes
df7 <- df6 %>%
    group_by(outcomes, Outcome_type) %>%
    summarise(total_outcome = sum(value))
```

```
## `summarise()` has grouped output by 'outcomes'. You can override using the `.groups` argument.
```

```r
# create a facet plot
df7 %>%
    ggplot(aes(reorder(outcomes, total_outcome), total_outcome)) + geom_col(fill = "grey70") + geom_tex
    filter(total_outcome < 650), aes(label = total_outcome), hjust = -0.1) + geom_text(data = df7 %>%
    filter(total_outcome > 650), aes(label = scales::comma(total_outcome), hjust = 1.2)) + coord_flip()
    panel.grid.major.y = element_blank(), panel.grid.major = element_line(linetype = 2, color = "black"
        color = "black", family = "mono")) + facet_grid(Outcome_type ~ ., scales = "free_y") + labs(x =
    10))
```



Source: Outcomes dataset