PPI_Stats

Zain Arifin

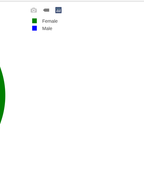
2025-07-20

```
data_summary <- data %>%
  count(Age_Range)

plot_ly(data = data_summary,
        x = ~Age_Range,
        y = ~n,
        type = 'bar',
        marker = list(
          color = ~n,
          colorscale = "Plasma"
        ),
        text = ~paste(n),
        hoverinfo = 'text') %>%
  layout(title = "Age Range",
         xaxis = list(title = "Age Range", tickangle = -45),
         yaxis = list(title = ""),
         bargap = 0.2,
         plot_bgcolor = "#ffffff",
         paper_bgcolor = "#ffffff")
```

```
gender_counts <- data %>%
  count(Gender) %>%
  mutate(
    percent = round(100 * n / sum(n), 1),
    text_label = paste0(Gender, "<br>", percent, "%<br>Count: ", n)
  )

plot_ly(gender_counts, labels = ~Gender, values = ~n, type = 'pie',
        hole = 0.4,  # this makes it a ring/donut chart
        text = ~text_label,
        textinfo = "text",
        insidetextorientation = 'radial',
        marker = list(colors = c("green", "blue", #00CC96', 'A4B63FA', '#FFA15A')) %>%
  layout(title = 'Gender Distribution',
         showlegend = TRUE)
```

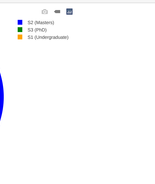



```
degree_counts <- data %>%
  count(Degree) %>%
  mutate(
    percent = round(100 * n / sum(n), 1),
    text_label = paste0(Degree, "<br>", percent, "%<br>Count: ", n)
  )

# Define desired order
desired_order <- c("S1 (Undergraduate)", "S2 (Masters)", "S3 (PhD)")
degree_counts$Degree <- factor(degree_counts$Degree, levels = desired_order)
degree_counts$text_label <- factor(degree_counts$text_label, levels = c("S1 (Undergraduate)<br>10.9%<br>Count: 1
0",
                                                                        "S2 (Masters)<br>65.2%<br>Count: 60",
                                                                        "S3 (PhD)<br>23.9%<br>Count: 22"))

degree_counts <- degree_counts %>% arrange(Degree)

plot_ly(degree_counts,
        labels = ~as.character(Degree),
        hole = 0.4,
        values = ~n,
        type = 'pie',
        text = ~text_label,
        textinfo = 'text',
        insidetextorientation = 'horizontal',
        marker = list(colors = c('orange', 'blue', 'green')) %>%
  layout(title = 'Degree Distribution',
         showlegend = TRUE,
         legend = list(traceyorder = "array",
                       categoryarray = desired_order))
```




```
city_counts <- data.frame(table(Data$City))
colnames(city_counts) <- c("City", "Freq")

library(tmap)
library(sf)
```

```
## Linking to GEOS 3.11.2, GDAL 3.6.2, PROJ 9.2.0; sf_use_s2() is TRUE
```

```
Sys.setenv("SHAPE_RESTORE_SHX" = "YES")
ROI_counties = st_read("S:/Counties/16 Counties/Census2011_Admin_Counties_generalised20m.shp")
```

```
## Reading layer `Census2011_Admin_Counties_generalised20m' from data source
##   `C:\Users\ritsuer\Documents\PPI_Ireland\Pequova_2025\PPI\Ireland\Data_Students\16_Counties\16 Counties\Cen
sus2011_Admin_Counties_generalised20m.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 26 features and 19 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 17491.14 ymin: 19589.93 xmax: 334536.4 ymax: 466919.3
## Projected CRS: TM65 / Irish Grid
```

```
ROI_counties <- select(ROI_counties, c(Nation = NUTS3NAME, County = COUNTYNAME, pop = Total2011, ID = COUNTY))
ROI_counties$City <- gsub(" County", "", ROI_counties$County)

Ireland <- ROI_counties
Ireland <- left_join(Ireland, city_counts)
```

```
## Joining with `by = join_by(City)`
```

```
tmap_mode("view")
```

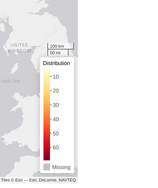
```
## ℹ tmap mode set to "view".
```

```
Ireland %>%
  tm_shape() +
  tm_fill(col = "Freq",
          palette = "YlOrRd",          # Yellow to Red palette
          title = "Distribution",
          text.size = 0.7,
          style = "cont",
          position = c("top", "left")) +
  tm_scale_bar(breaks = c(0, 50, 100), text.size = 0.5,
               position = c("right", "bottom")) +
  tm_layout(frame = FALSE)
```

```
##
```

```
## == tmap v3 code detected ==============================================
```

```
## [v3->v4] `tm_polygons()`: instead of `style = "cont"`, use fill.scale =
##  `tm_scale_continuous()`.
## ℹ Migrate the argument(s) 'palette' (rename to "values") to
##  `tm_scale_continuous(<HERE>)`
## [v3->v4] `tm_polygons()`: use "fill" for the fill color of polygons/symbols
##  (instead of "col"), and "col" for the outlines (instead of "border.col").
## [v3->v4] `tm_polygons()`: migrate the argument(s) related to the legend of the
##  visual variable `fill` namely "title" to "fill.legend = tm_legend(<HERE>)"
## [tm_polygons()] Arguments `text.size` and `position` unknown.
## ℹ `tm_scale_bar()` is deprecated. Please use `tm_scalebar()` instead.
## [tm_scalebar() color palettes: see palettes from the R package cols4all. Run
##  `cols4all::c4a_gui()` to explore them. The old palette name "YlOrRd" is named
##  "brewer.yl_or_rd"
## Multiple palettes called "yl_or_rd" found: "brewer.yl_or_rd", "matplotlib.yl_or_rd". The first one, "brewer.yl
_or_rd", is returned.
```




```
### University ###
data_summary <- data %>%
  count(University) %>%
  arrange(n) %>%
  mutate(University = factor(University, levels = unique(University)))

plot_ly(data = data_summary,
        x = ~n,
        y = ~University,
        type = 'bar',
        orientation = 'h',              # horizontal bars to show universities on y-axis
        marker = list(
          color = ~n,
          colorscale = "Plasma"
        ),
        text = ~paste(n),
        hoverinfo = 'text') %>%
  layout(title = "Students Distribution in Irish Universities",
         xaxis = list(title = "Count"),
         yaxis = list(title = "", categoryorder = "array", categoryarray = levels(data_summary$University)),
         bargap = 0.2,
         plot_bgcolor = "#ffffff",
         paper_bgcolor = "#ffffff")
```

```
data_summary <- data %>%
  count(Major_general) %>%
  arrange(n) %>%
  mutate(Major = factor(Major_general, levels = unique(Major_general)))

plot_ly(data = data_summary,
        x = ~n,
        y = ~Major,
        type = 'bar',
        orientation = 'h',              # horizontal bars to show universities on y-axis
        marker = list(
          color = ~n,
          colorscale = "Plasma"
        ),
        text = ~paste(n),
        hoverinfo = 'text') %>%
  layout(title = "Students Majors in Irish Universities",
         xaxis = list(title = "Count"),
         yaxis = list(title = "", categoryorder = "array", categoryarray = levels(data_summary$University)),
         bargap = 0.2,
         plot_bgcolor = "#ffffff",
         paper_bgcolor = "#ffffff")
```