2324-CT5100 Data Visualisation
Learning Journal

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Vikrant Singh Jamwal
(23104534)
M.Sc. Artificial Intelligence

University of Galway, Ireland
Professor:
Dr. Conor Hayes

Table of Content

Section 1: “Introduction”

There are many learning outcomes of the Data Visualisation module that will be very useful and insightful. This course not only teaches the just the path to create a data visualisation but also helps to develop a mindset for developing a clear as well as insightful data visualisation keeping various factors in mind.

Few major learning outcomes are:

- **Significance of the knowledge about the audience, space, time and medium.**

  There is a misconception that visualizing a data relies only on the message and insights it provides, but it is more about who is going to read the visualisation, how much time does we have to present the visualisation, the space given for the data visualisation and who will be presenting it. All these aspects also play a major role in creating a good data visualisation for that scenario.

- **Importance of colour and other visual elements**

  Colour is inseparable part of data visualisation, and this module provides detailed explanation on the use colours in various situations. Distinguishing the use of hue, chroma, luminance, saturation, and gradients for different requirements including colour vision deficiency. Various visual and graphical elements are also important based on the requirements. Overall, the outcome of the data visualisation should be aesthetically good as well as clearly understandable even for the people with CVD.

- **Storytelling**

  A good data visualisation should tell an engaging story. This course explains the usage and need of each graphical element with examples from various real-life occurrences which helps in understanding how a story can be told from the data. This story does not need to be verbally told but can also be understandable visually. Creating such a visualisation or a set of visualizations which tells a story can be misleading, which is also tackled precisely in the module.

Data visualisation module covered every important aspect which leads to the development of a clear and insightful visualisation.

Section 2: "An alternative visualization to a 3D visualisation"

2.1. Exploring dataset:

- Dataset is small with 9 instances and 5 features.
- Feature "Field" is an Object type and other features are Numerical.
- Scale of each numerical feature is comparable, hence can be drawn on one common scale.

2.2. Flaws in the 3D Design:

- It is not easy to interpret the values of datapoints on each axis in 3D representation. The dotted lines and shadows drawn are also confusing and chaotic.
- Colours are used for the "Field" feature having 9 values, hence colours with similar shades are also taken which created a visual confusion. Also, colours of the spheres are 3D represented with shades and shines making it converge with similar shaded datapoints.
- Axis is not labelled, hence making it difficult to interpret the values of the datapoints.

2.3. Approach to overcome Flaws:

- As the Dataset is small, a simple 2D plot is used where X-axis has all the 9 Fields and Y axis is the common scale for the remaining 4 features.
- Each feature will be of visually distinct colour (As features are less than Fields, hence less colours are used making it easy to distinguish).
- Grid Lines is drawn to trace the value of the datapoints on the axis and Legends are used to determine the colours associated with the features.
- Visual elements used are:
  - 2D plot (X-axis: "Fields", Y-axis: Scale (0 to 9))
  - Line + Marker for Datapoints
  - Colour for each Feature
  - Grid lines

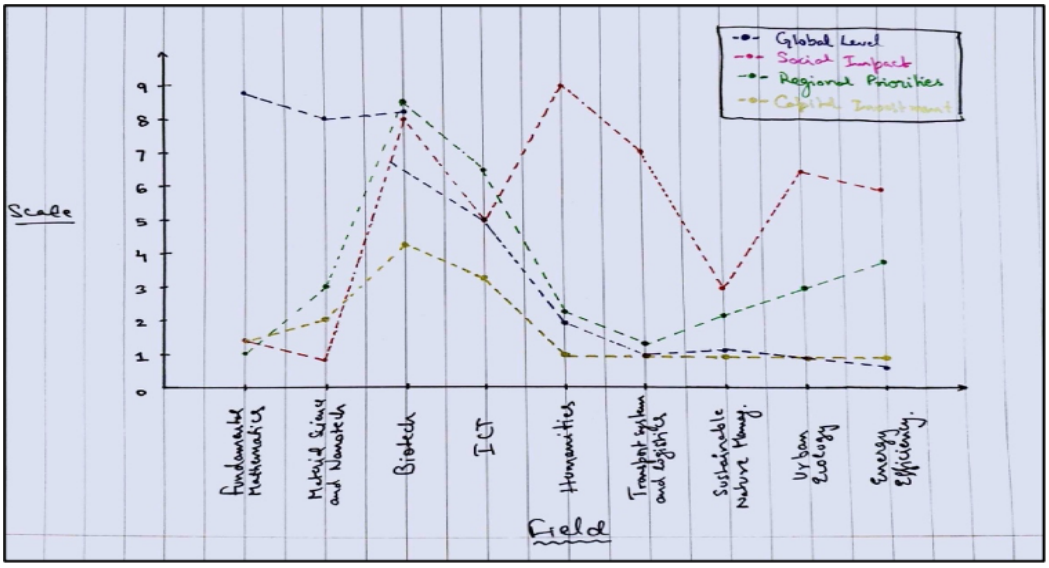


Figure 1

2.4. Created using Python's Matplotlib

- Used Python's Pandas and Matplotlib libraries to **sort** the data by capital investment and plot the features according to the above approach.

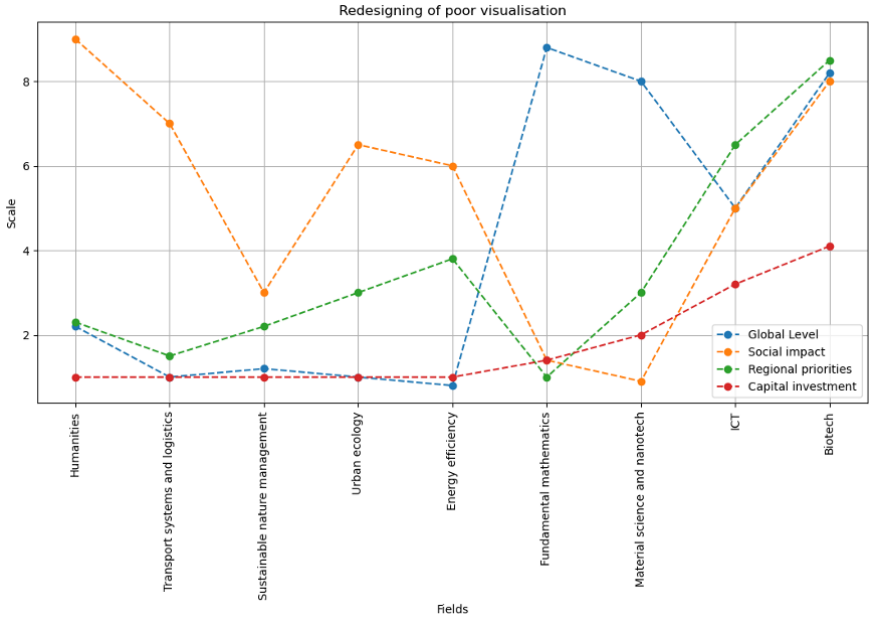


Figure 2

Here, Global level, Social Impact and Regional priorities are visualised with the increasing capital investment for the fields.

Advantages and Visual perception

- For a small dataset, a complex visualization takes more time to understand the data than the data itself. This approach tackles the complexity with a simple 2D representation of data in a single plot with features of different colours making it easy to interpret the different features in a single plot.
- As we have given colours to features than Fields unlike the poor visualization, as there are less features (four) making it easy to choose distinct colour for each one.
- Lines are dotted, so that overlapping can be distinguishable. Also, grid and markers are used which can be used to trace the values of each datapoint.

Reviewing

Message

- We can visualize how Global Level, Social Impact and Regional priorities are correlated with the capital investment of each sector. As the capital investment increases, we can clearly see the rise in the other three attributes.

Audience and Presentation

- Investors and CEOs of companies underlying in these sectors. This can be presented through a slideshow as to understand the return on investment and opportunity to grow by expanding the network and capital in these sectors. Eyes are drawn to finding patterns and trends in the visualisation to see the correlation between each line.

Section 3: "Why this visualisation/infographic succeeds"

3.1. Task

- Aim for this week's exercise was to find a visualisation and to explain why this visualisation succeeds using the qualities by Cairo and Bertini. I have found a data visualisation of an NBA club The Spurs analyzing their defense under different managers within a time period.

3.2 Visualisation

A good Visualization by Cairo standards

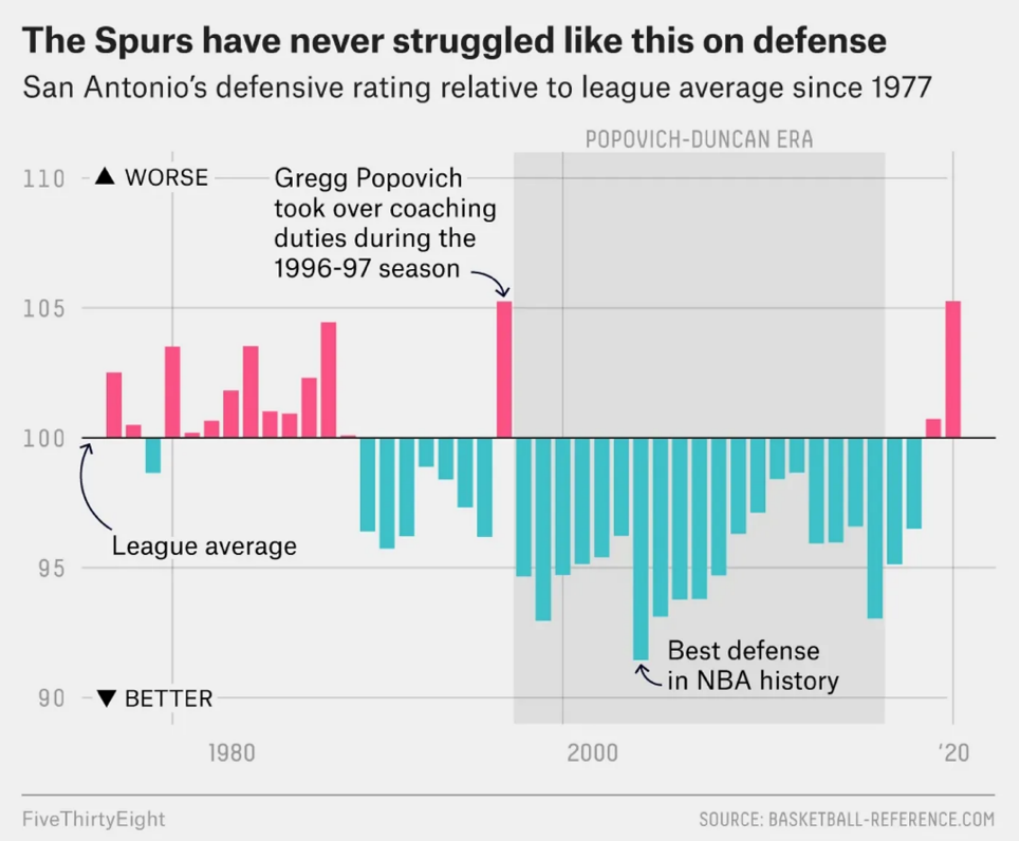


Figure 3 – Source: https://www.basketball-reference.com/

3.3. Qualities of Cairo standards

3.3.1. Truthful

Clear picture with correct information without any misleading or missing information. Timespan is large enough to compare the POPVICH-DUNCAN ERA with the rest of the time. No modulation in scale or information to support a propaganda.

3.3.2. Functional

Correct use of all the visual encoding elements to bring out insights from the data. Highlighted the area to focus on the era to form a story. Added labels of facts and key information like "Best defense in NBA history", "League average"

and the directions for WORSE and BETTER to give the idea that higher means worse and lower ratings signify a better defense.

3.3.3. Insightful

From the visual piece we get a clear understanding of how San Antonio Spurs team performed at defense with or without the new coach. Each key place is well labelled and no information between years are hidden. Hence, this visualization provides good insights.

3.3.4. Enlightening

From the Visuals, it can be believed that the influence of the new defense coach improved the Spurs defense and took them to break the record. The whole era of the coach, spurs significantly showed improvement in their defense and reached the best of all time. Hence, the visual changes the mind for better if the information given is grasped.

3.3.5. Aesthetically Pleasing

Some of the aesthetic features includes keeping league average in the middle, use of grid lines to trace the year and ratings both, two contrasting colors are used to show the bad and good ratings compared to the league average. Also, the axis labels, grids and era are put in background (in Gray) and important lines and texts are in foreground (in Black).

Hence, it passes all the requirements for a good visualization by Cairo.

Review

3.5. Analytical Relationship

- Relationship will be “difference from a reference”. As the visualisation is the diverging bar chart which shows the difference of defense score off the club from mean 100.

3.6. Message

- The message is clear and concise showing that the club was struggling in defense and how coach Gregg Popovich took over and helped the team to reach the best defense of history with consistent results.

3.7. Visual Perception

- Eyes were drawn to the highlighted area in the visualisation which shows the message of this visualisation very clearly. Colours used are CVD friendly as well as the visualization involves use of labels which tells a precise story with good understandings for any type of audience (with or without being an NBA fan).
- The visual elements used minimal ink. As the labels allows the understanding of the time-period, hence time on the x axis is not binned densely. Also, as the message does not depend on exact value of defense or time, hence grids are used efficiently to trace the values.

Section 4: "Successful/unsuccessful visualisation decisions"

4.1. Task

- Aim for this exercise is to examine the visualisation decisions took by the analyst while making these six visualisations and justify how each decision is successful or unsuccessful.

4.2. Audience and Presenter

- Goal of these visualisation was to provide comparison between the performance of a company on various aspects with its competitors. Hence the audience will be the shareholders of the company and presenter will be an analyst presenting through a slideshow.
- Time is less as it will be explained in a single slide time, hence it should not be detailed but clear and concise.

Visualization 1

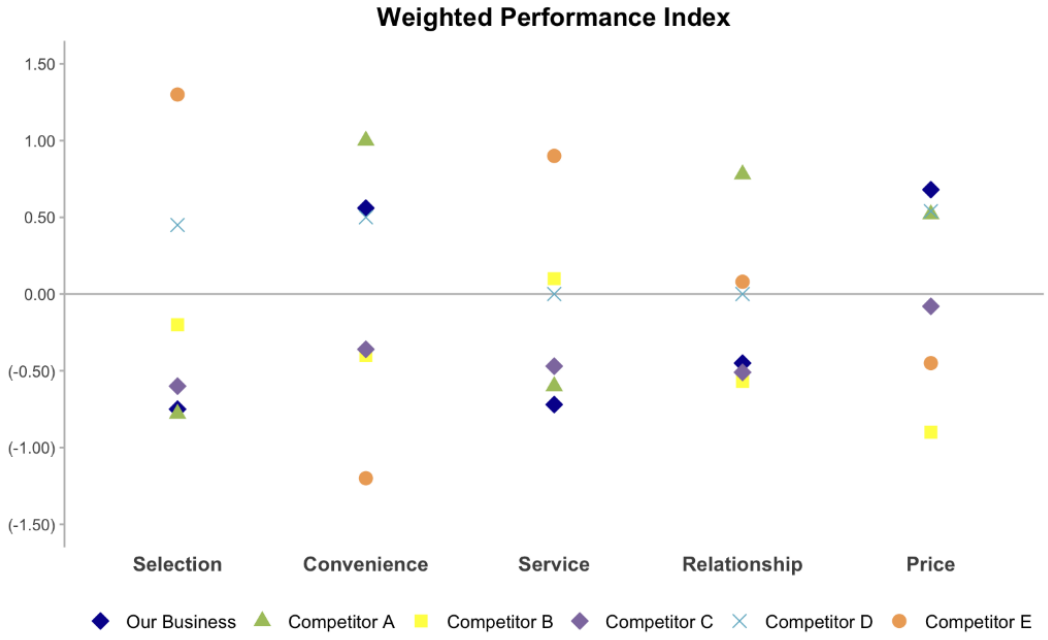


Figure 4

Analysis:

1. pre-attentive processing stage will not be able to figure out any significant pattern due to excessive use of colors and shapes. As for each company the color and shape are different, it is difficult to trace.
2. Also, the points are small and are in proximity of each other, making it tough to distinguish between each point.
3. The column wise separation is quite significant, hence only the categories (Selection, Service, Price, etc.) are visibly distinguishable.
4. Some of the shapes used are similar and can cause confusion between two companies.

Visualization 2

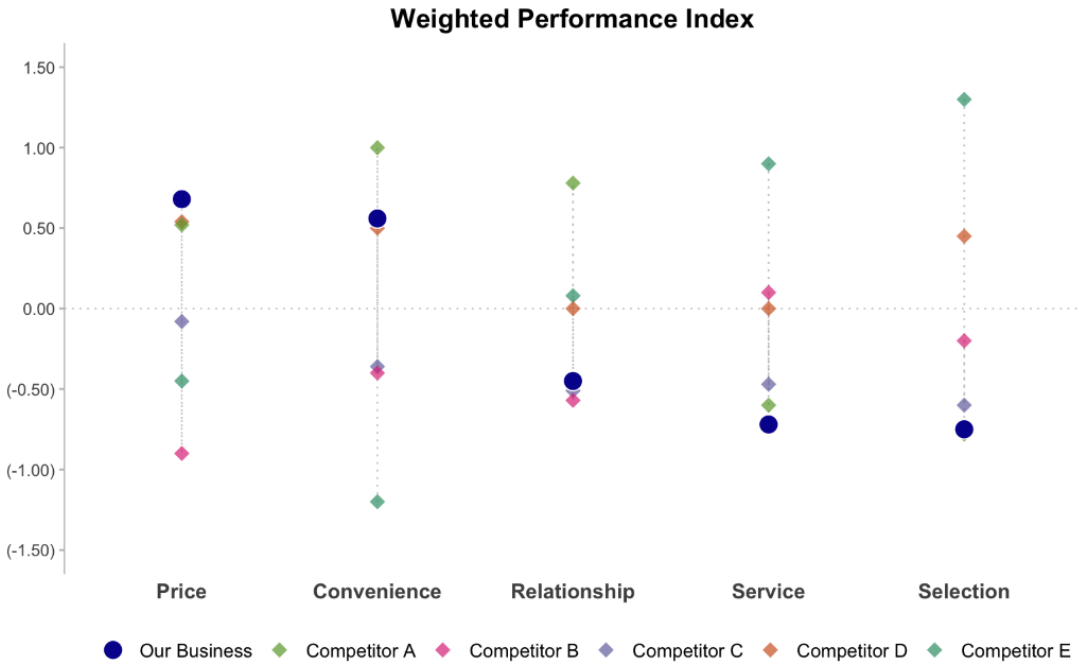


Figure 5

Analysis:

1. The designer updated the visualization by representing their business by circle and all other companies by a similar shape. This change can help the pre-attentive stage of processing a visualization to clearly spot their company over other companies.
2. Also, the color used for their business is darker and for all the other companies, light colors are used. This change brings the significant information (their business) to foreground and other information (other companies) to background.
3. The tracing vertical line for each category seems not quite useful as the significant proximity gap already distinguish them clearly.
4. Also, there is no way of tracing the exact value of each point. Can't get the exact numerical weighted performance Index of any point.

Visualization 3

Company	Price	Convenience	Relationship	Service	Selection
Our Business	10.0	8.2	1.8	1.0	1.1
Competitor A	9.1	10.0	10.0	1.7	1.0
Competitor B	1.0	4.3	1.0	5.6	3.5
Competitor C	5.7	4.4	1.4	2.4	1.8
Competitor D	9.2	8.0	4.8	5.0	6.3
Competitor E	3.6	1.0	5.3	10.0	10.0

Figure 6

Analysis:

1. Designer has changed the approach from plotting the points to tabular visualization with colors and bars. This update overcomes some of the issues from the previous visualizations.
2. For highlighting their business, a different color is used than for all the other companies with the bars at each cell representing the value of the cell. Numerical value is given, hence from this visualization, exact value of each cell can be observed unlike the previous visualizations.
3. Still, the visualization needs active attention to understand and analyze the data, which defies the sole purpose of data visualization.

Visualization 4

Company	Price	Convenience	Relationship	Service	Selection
Our Business	0.68	0.56	-0.45	-0.72	-0.75
Competitor A	0.52	1.00	0.78	-0.60	-0.78
Competitor B	-0.90	-0.40	-0.57	0.10	-0.20
Competitor C	-0.08	-0.36	-0.51	-0.47	-0.60
Competitor D	0.54	0.50	0.00	0.00	0.45
Competitor E	-0.45	-1.20	0.08	0.90	1.30

Figure 7

Analysis:

1. Heat map is applied to the visualization, where red signifies the negative index values and blue signifies the positive index. This change helps visually comparing the values rather than comparing each cell with numerical values.
2. Because of colors, it is tough to see the numerical values as the color gets darker. Active attention is still required to understand the visualization.
3. Still the issue of aesthetically pleasing persists as tabular form of visualization is not considered as a good way of presenting data.
4. A good visualization which provides important details but still needs a better way to present.

Visualization 5

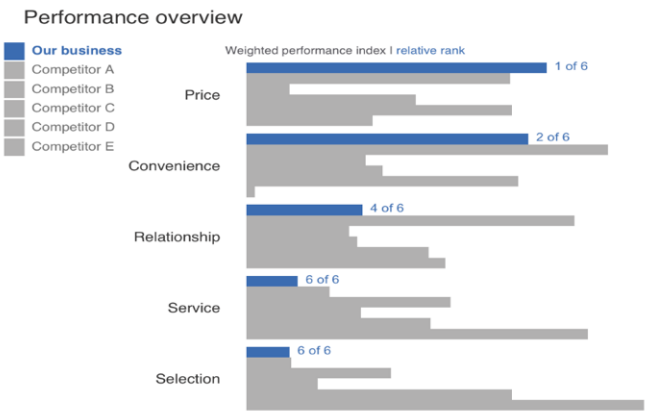


Figure 8

Analysis:

1. A changed approach of presenting the data in a horizontal bar chart is a significant improvement from the previous visualization. Proximity gap between each group distinguishes the categories and legend identifies the bars.
2. Designer highlighted their business with a darker color which brings it to the foreground and specified the rank of their business compared to other companies for each section.
3. Length of each bar, color and rank helped the visualization to get easily understandable and interpretable without any active attention requirement.
4. Better than previous approaches as it satisfies the purpose of data visualization.
5. As the color of other companies are similar, hence this chart can be confusing to analyze the performance between companies. But this is the best approach to compare their business with other companies as it satisfies all the points of the theory of perception.

Visualization 6

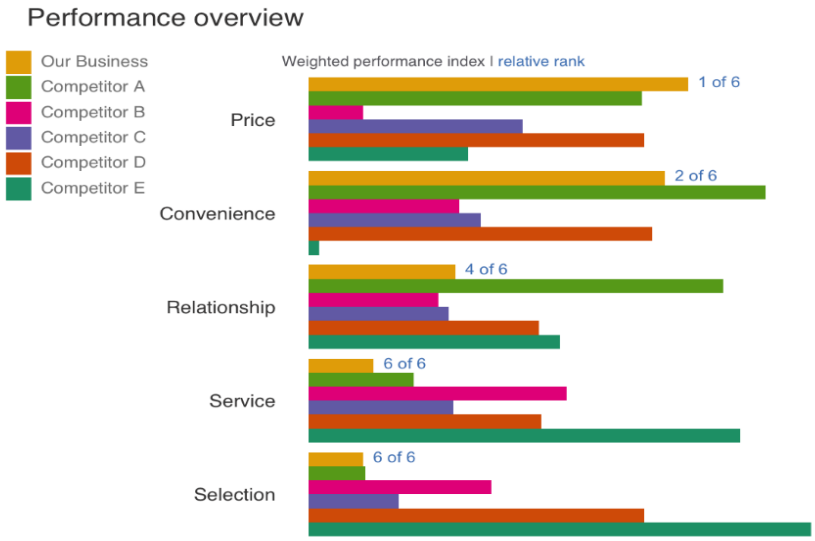


Figure 9

Analysis:

1. A slight update of giving colors to each company is an improvement to the overall visualization as now each company can individually be compared and visualized.
2. Although, the final visualization does not provide the exact Index, but for comparative analysis it satisfies the theory of perception efficiently and is aesthetically pleasing as well.

Section 5: "Population growth vs commuting distance from Dublin"

5.1. Database

Two databases are merged,

DB1 had data of population growth and population in different counties and provinces of Ireland.

DB2 had data of distance and commute time from the largest city of each province to Dublin. Data is collected using Google maps. Distance and compute time is of Bus Services at 9:00 AM on weekdays, where time is affected by the population movements.

5.2. Data File

	County	Province	Year_2006	Year_2016	pop_growth	letter	KM	commute_time.min.
1	Laois	Leinster	67059	84697	0.263022115	z	89.1	132
2	Meath	Leinster	162831	195044	0.19783088	y	132	85
3	Kildare	Leinster	186335	222504	0.194107387	x	37.2	63
4	Cavan	Ulster	64003	76176	0.19019421	w	117	135
5	Longford	Leinster	34391	40873	0.188479544	v	119	128
6	Louth	Leinster	111267	128884	0.158330862	u	84.4	75
7	Wexford	Leinster	131749	149722	0.136418493	t	138	127
8	Dublin	Leinster	1187176	1347359	0.134927761	s	0	0
9	Kilkenny	Leinster	87558	99232	0.133328765	r	128	108
10	Carlow	Leinster	50349	56932	0.130747383	q	97.3	90
11	Wicklow	Leinster	126194	142425	0.128619427	p	49.3	55
12	Cork	Munster	481295	542868	0.127931934	o	258	199
13	Westmeath	Leinster	79346	88770	0.118770953	n	79.2	110
14	Galway	Connacht	231670	258058	0.113903397	m	208	190
15	Leitrim	Connacht	28950	32044	0.106873921	l	153	251
16	Offaly	Leinster	70868	77961	0.100087487	k	124	125
17	Roscommon	Connacht	58768	64544	0.098284781	j	168	252
18	Monaghan	Ulster	55997	61386	0.096237298	i	103	129
19	Donegal	Ulster	147264	159192	0.080997392	h	240	240
20	Sligo	Connacht	60894	65535	0.076214405	g	191	258
21	Waterford	Munster	107961	116176	0.076092293	f	209	200
22	Clare	Munster	110950	118817	0.070905813	e	228	228
23	Tipperary	Munster	149244	159553	0.069074804	d	188	167
24	Limerick	Munster	184055	194899	0.058917171	c	240	223
25	Kerry	Munster	139835	147707	0.056294919	b	299	306
26	Mayo	Connacht	123839	130507	0.053844104	a	234	190

5.3. Custom palette created and tested for CVD friendly

Masataka Okabe and Kei Ito developed a set of colors that is unambiguous for both people with a CVD and to people with normal vision. We can use these palettes for creating a CVD friendly palette.

Creating a custom palette using HCL wizard as mentioned in the exercise. Used “hclwizard()” to create a qualitative custom palette with distinguishable and CVD friendly colours.

These colours are chosen for the purpose of qualitative analysis where colours should be easily distinguishable from each other and hence can be used to establish counties in different provinces.

Custom palette

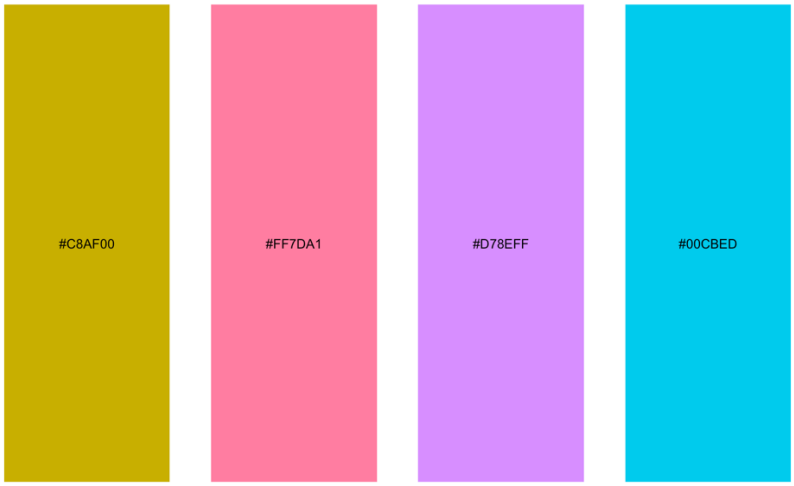


Figure 10a

CVD test using “cvd_grid()”

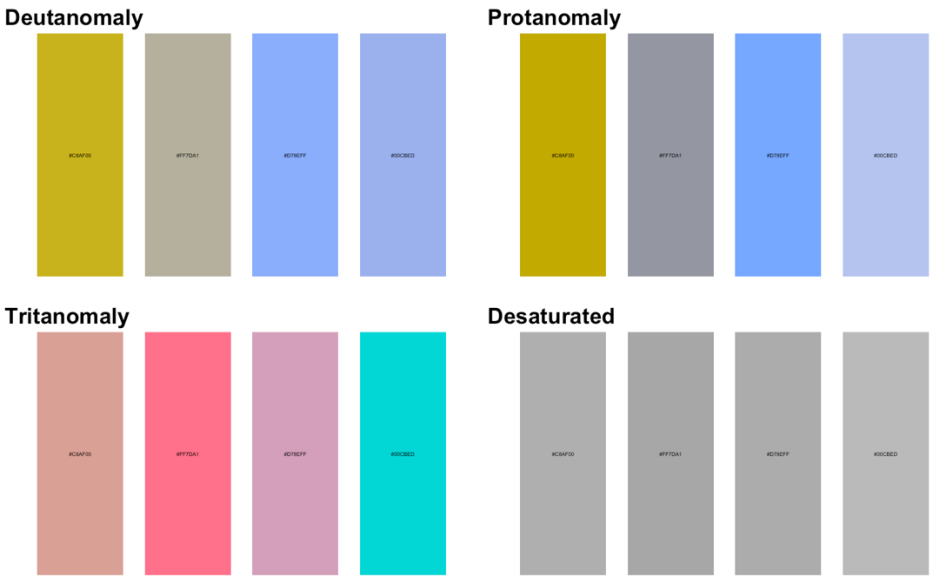


Figure 10b

5.4. Exploratory Analysis and Trends

- Finding meaningful trends by exploring the database through visualisations and providing an explanation of the pattern recognized.

5.4.1. Visualising population growth with compute distance from Dublin by Bus service at 9:00 AM on a weekday.

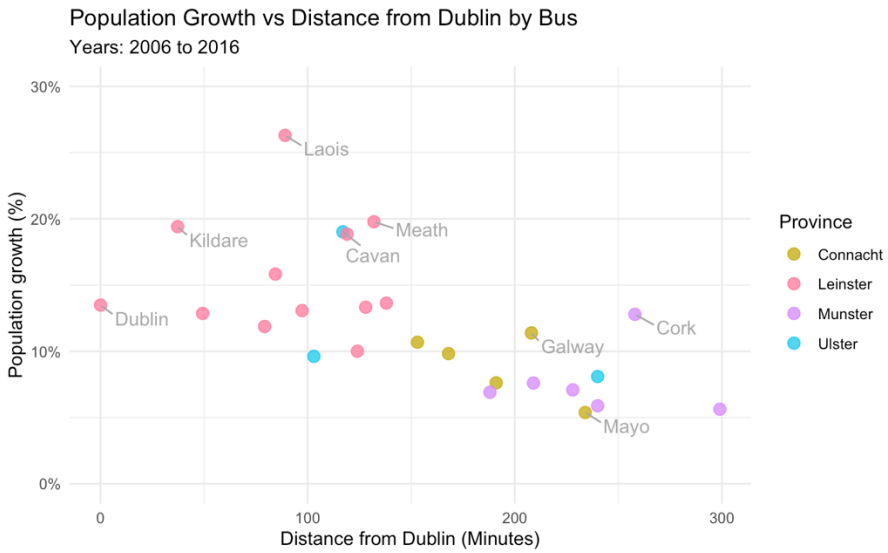


Figure 11

Trends

- As the Distance increases, we can see a decline in the population growth, it suggests that far-away places from Dublin have less population growth from 2006-2016.

5.4.2. Visualising population growth with computing time from Dublin by Bus service at 9:00 AM on a weekday.

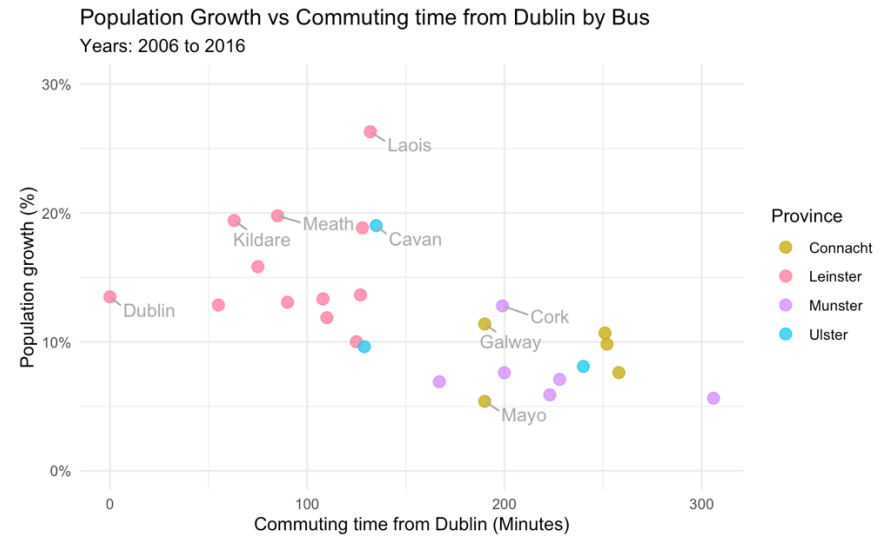


Figure 12

Trends

- We can see similar results with commuting time, as the commute time increased, the population growth declined.

5.4.3. Visualizing Population growth with Commute time from Dublin for each province.

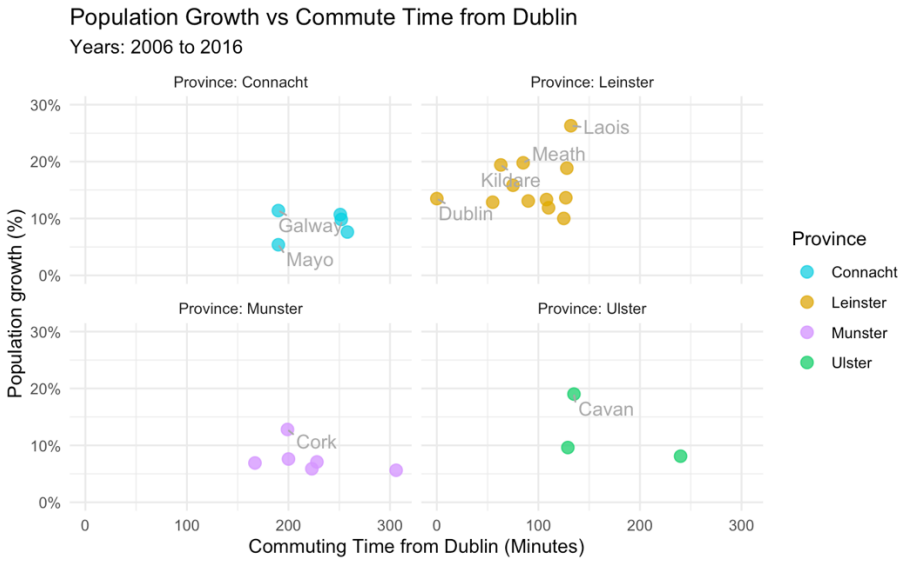


Figure 13

Trends

- Leinster has less average commute time from Dublin and high population growth compared to the other provinces.

5.4.4. Visualizing Distance and Commute Time together with Population growth.

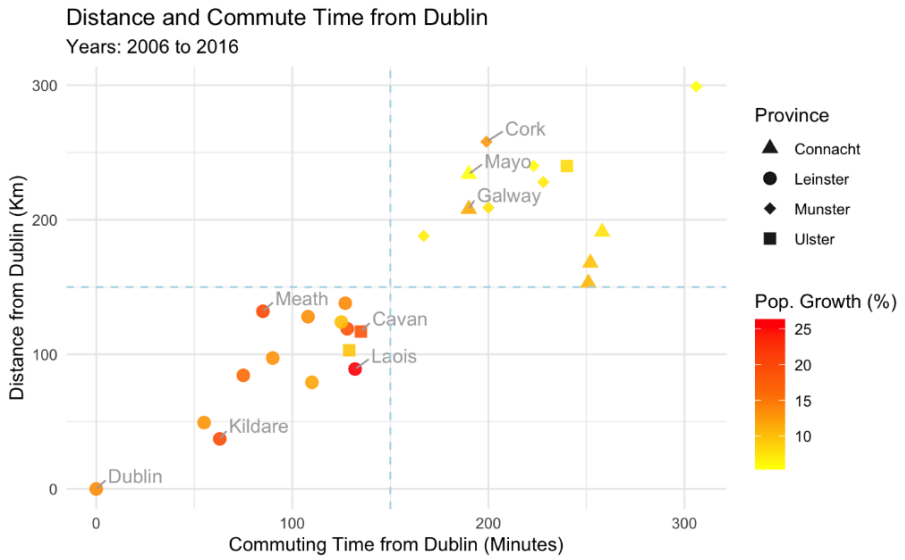


Figure 14

Trends

- Counties within the range of 150km and 150min of commute time from Dublin has higher population growth than the far-away counties.

5.4.5. Visualizing Distance and Commute Time together with Population growth.

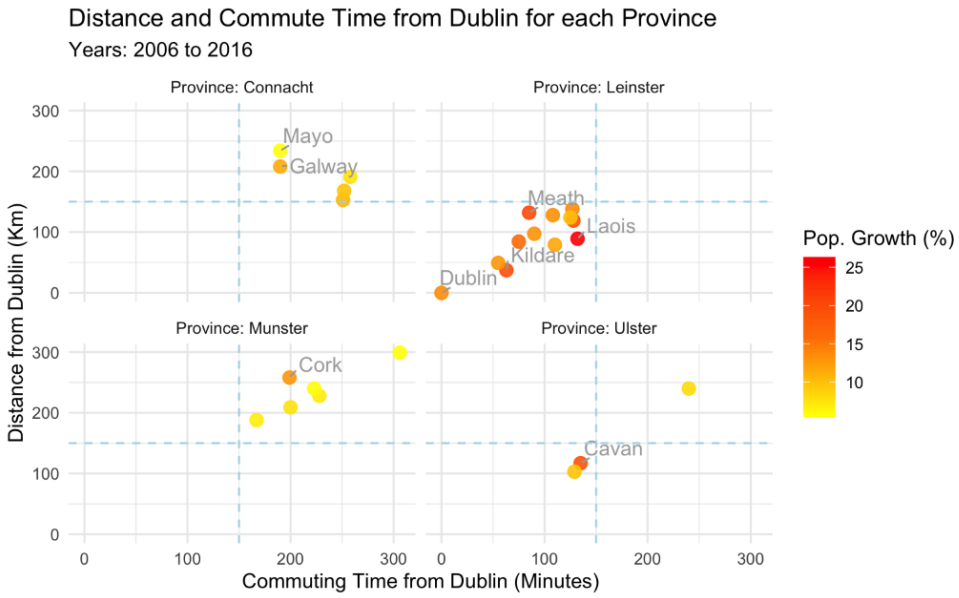


Figure 15

Observations and Trends

- Leinster province has the highest population growth as it is closest to the highest populated county Dublin. Laois has the highest growth out of all the counties.
- Province Ulster has two counties including Cavan within the range of 150 km from Dublin by bus, and we can see the incline in their population, on the other hand the third county has less increment as it is not in the range.
- Other provinces which are distant from Dublin has less population growth, still in the major cities like Cork, Galway, and Cavan we can see the significant increase.

Section 6: "Redesign of a Road Safety Authority slide"

6.1. Task

- To create a visualisation or multiple visualisations that be placed in a box of a slideshow and supports the points made in the module slide.

6.2. Few aspects to be considered

- Audience: Minister of transport (No necessary tech knowledge)
- Space: Slide show (One slide with various highlighting visualizations)

6.3. Exploratory Analysis Visualizations

- First Visualization offers the information of fatalities over 5 years from 2018 to 2022.

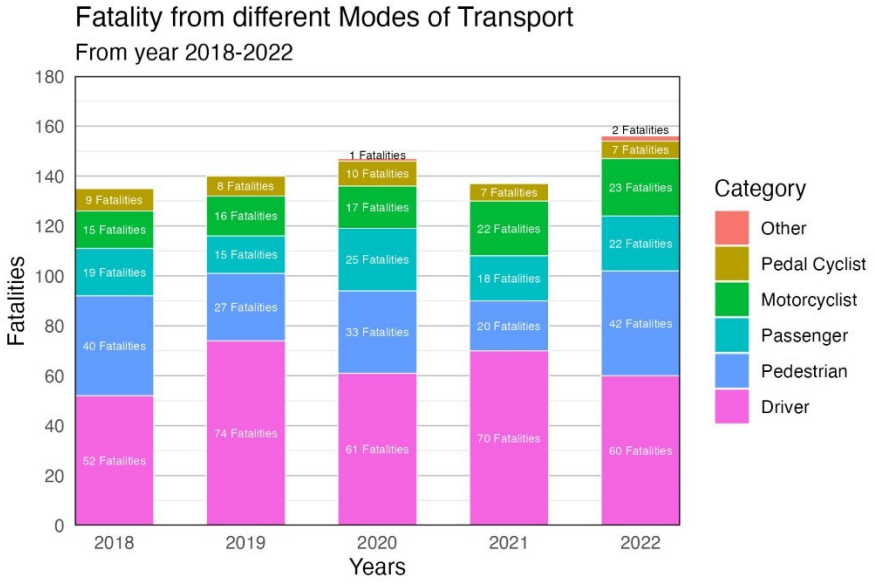


Figure 16a

- Second Visualization offers the information of fatalities from 2021 and 2022.

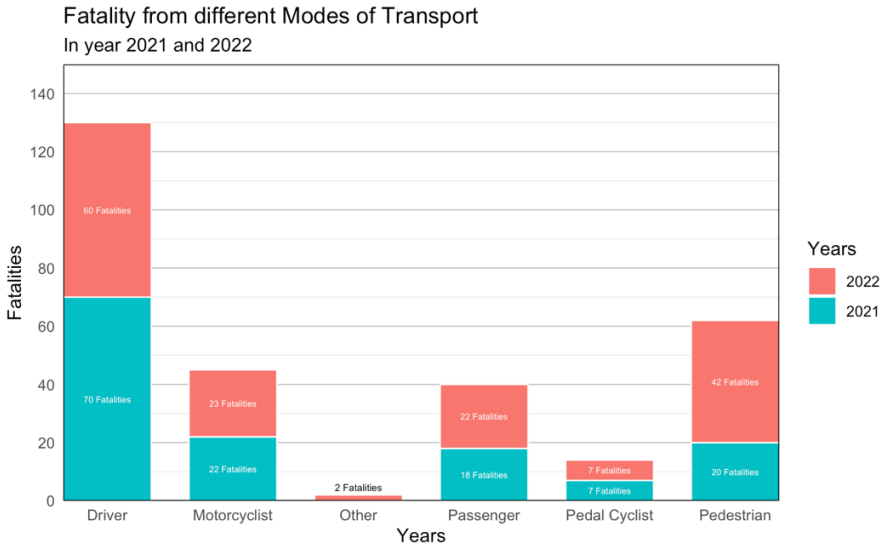
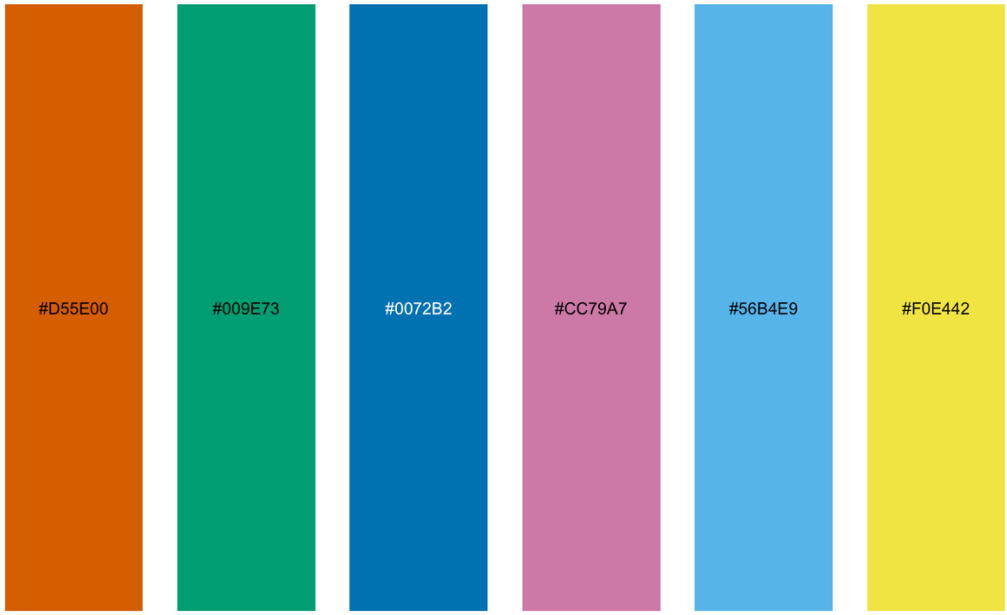


Figure 16b

- But according to our audience, space, and time limit there is better way of creating visualizations which covers all the bullet points.

6.4 CVD friendly colour palette

- Used CVD colour palettes for both the visualisations. Masataka Okabe and Kei Ito developed a set of CVD friendly colors and few of those colours are used to create these two visualisations below.



6.5. Creating multiple visualisations

- Created Bar chart and Pie chart together to understand the divergence as well as the proportion of the dataset. All together they create a story covering all the required points.

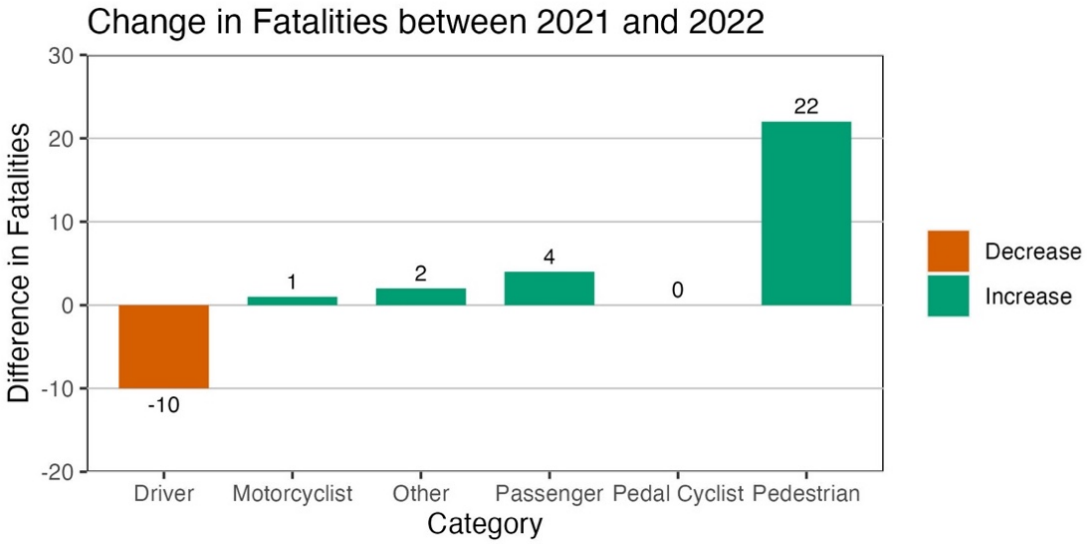


Figure 17: Bar chart

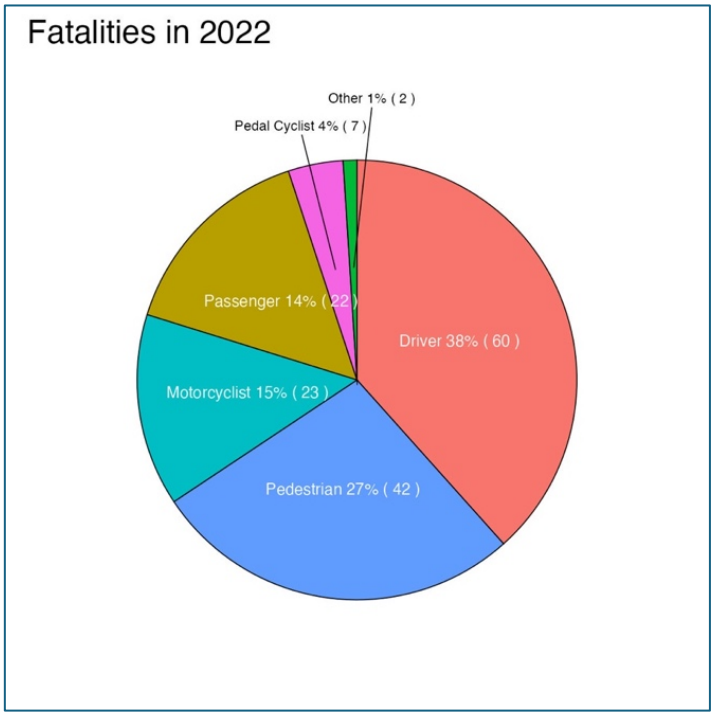


Figure 18: Pie chart

- Combination of both the pie chart and bar-graph covers all the bullet points and takes care of the audience knowledge and time limitations of a slideshow.

6.6. Observations

- From the pie chart we can visualize that **38% are Driver fatalities** and **pedestrian fatalities are at 27%.**
- From the **bar graph** we can clearly see the **increase or decrease in the fatalities** for each category from 2021 to 2022.
- Fatalities **increased** for **pedestrians (+22), passengers (+4), motorcyclists (+1)** and **other road users (+2).**
- **Decline** has been seen in **driver fatalities (-10).**
- Number of **pedal cyclists were 7** in 2021 and **remained same** with no change in 2022.

Section 7: "Visualising Irish COVID-19 data"

7.1. Task

- Creating a Heatmap, where each tile represents each county's monthly cases per 100K diverge from the mean number of cases (per 100K) in that month.

7.2. Preprocessing data

- Used reference from the given codes in the module. (**code: Ireland-Covid-Monthly-HeatMap.html**)
- From these codes available, taken mutated databases to extract months and years. Also created columns for cases/100000 per month of each county and mean cases per month.
- Used these cases to mutate another column of the difference of cases/100K of each county with the mean of that month. This column will be used to fill the gradient in the heat map.

7.3. Colour Palette

- Custom diverging colours from HCL wizard and tested for CVD friendly.

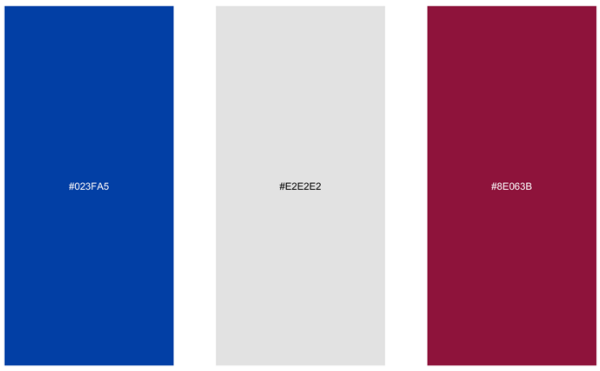


Figure 19a

- Tested using 'colorblindr' library for CVD friendly.

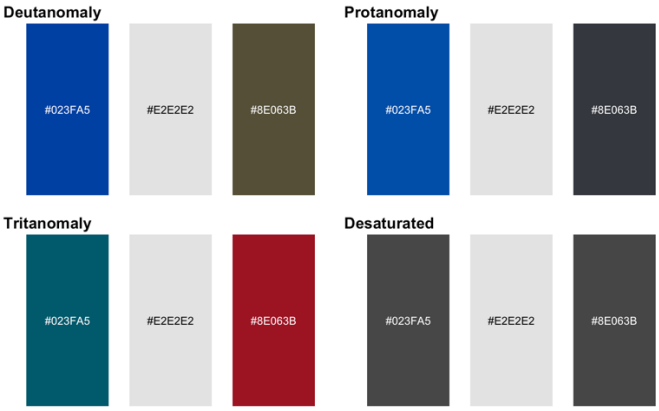


Figure 19b

7.4. Heat Map

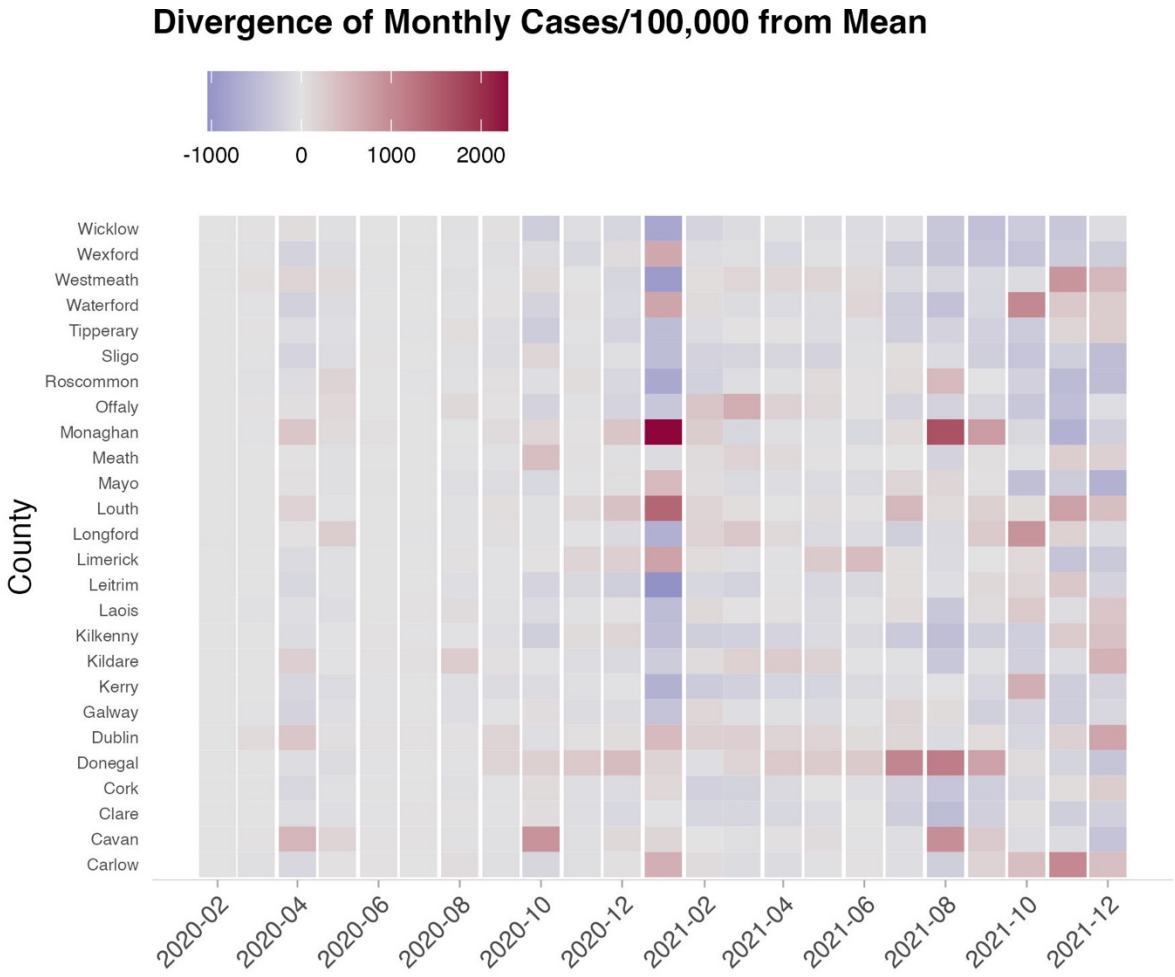


Figure 19

7.5. Observations

- We can visualize as the first months there were no significant sign of covid, hence each county is grey as the mean and the cases are both almost 0.
- From April, we can see the change in the mean hence counties with divergence from mean can be visualized in both positive and negative direction (Blue being lesser cases and red means cases increased).
- January 2021 can be seen as the most diverged month where we can see which counties were affected and which are still stable. Monaghan can be seen as highly affected in January 2021 out of all the counties.
- A decline in mean and divergence from mean can be seen from Feb 2021 to June 2021, and then second wave hits the counties.

Appendices

Includes section codes, references, and description of the visualisation.

Appendix (Section 2)

Used Python notebook, pandas and matplotlib to handle data and create visualisation.

Code for Figure 2

```
# Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Creating a dataframe using pandas
df = pd.read_csv("5D_chart_data.csv")
# Sorting the database by "Capital investment"
df = df.sort_values(by="Capital investment")

# Dropping the field column
df_a = df.drop("Field", axis=1)
df_a = df_a[["Global Level ","Social impact","Regional priorities","Capital
investment"]]

# visualising top 5 rows in the dataframe
df_a.head()

# Plotting the visualisation
plt.figure(figsize=(13, 6))
# Specifying the line and marker style
plt.plot(df["Field"], df_a, linestyle="--", marker="o",label=df_a.columns)
# Labelling axis
plt.xlabel('Fields')
plt.ylabel('Scale')
# Title of the plot
plt.title('Redesigning of poor visualisation')

# Rotating the x-ticks by 90 degrees
plt.xticks(rotation=90)
# Keeping the grids
plt.grid(True)
# Showing the legend for the columns
plt.legend();
```

Appendix (Section 3)

Aim of the weekly exercise was to find a good visualisation. I have taken the above analyzed visualisation (Figure 3) from a basketball reference website.
Reference: https://www.basketball-reference.com/

Appendix (Section 4)

Aim of this weekly assignment was to analyze six visualisations. I have taken these six figures (4, 5, 6, 7, 8, 9) from the module file.
Reference:https://universityofgalway.instructure.com/courses/13767/assignments/62457?module_item_id=483918

Appendix (Section 5)

Aim of this weekly exercise was to create a given database with commuting distance and time to analyze the relationship with population.

Reference of CVD pallet code

- https://universityofgalway.instructure.com/courses/13767/files/1577228?module_item_id=489850

Reference of other codes used and modified from the module for this exercise

- https://universityofgalway.instructure.com/courses/13767/files/1570554?module_item_id=487601&fd_cookie_set=1

- https://universityofgalway.instructure.com/courses/13767/files/1570243?module_item_id=487529&fd_cookie_set=1

Code for Figure 10a and 10b (CVD Pallet)

```
# Reading the database
df_pop <-read.csv("pop.csv")
df_dist <- read.csv("dataset4-5.csv")

# Joining the both databases by primary key County
df_data <- inner_join(df_pop, df_dist, by = "County")

# Labelling the counties
counties_to_label<-c("Dublin", "Laois", "Galway", "Mayo", "Cork","Meath",
"Kildare","Cavan")
data_for_labelled_counties<-filter(df_data, County %in% counties_to_label)

# Choosing CVD pallet colours using HCL wizard
hclwizard()
custom_palette <- qualitative_hcl(n = 4, h = c(75, -150), c = 100, l = 71)

# Testing CVD friendly using CVD grid of colorblindr library
cvd_grid(palette_plot(custom_palette, label_size=1))

write.csv(df_data, "final_data.csv")
```

Code for Figure 11

```
# Scatter plot of Distance and Population growth, colour by Provinces
ggplot(df_data,
       aes(x=KM,
           y=100* pop_growth,
           colour = Province))+

# Point size and transparency
  geom_point(size=3, alpha=0.8)+

# Y-axis scaling with percent format
  scale_y_continuous(
    limits = c(0, 30),
    labels = scales::percent_format(scale=1)
  ) +

# Labelling some of the major counties
  geom_text_repel(data=data_for_labelled_counties,
                  aes(label=County), colour="darkgrey",
                  min.segment.length = 0,
                  nudge_x=20,
                  nudge_y=-1) +

# Filling with custom palette (CVD friendly)
scale_colour_manual(values= custom_palette)+

# Titles, axis labels and theme
labs(title="Population Growth vs Distance from Dublin",
       subtitle="Years: 2006 to 2016",
       y="Population growth (%)",
       x="Distance from Dublin (Km)")+
theme_minimal()
```

Code for Figure 12

```
# Scatter plot for Commute time and Population growth, colour by provinces
ggplot(df_data,
       aes(x=commute_time.min.,
           y=100* pop_growth,
           colour = Province))+
# Point size and transparency
  geom_point(size=3, alpha=0.8)+

# Y-axis scaling with percent format
  scale_y_continuous(
    limits = c(0, 30),
    labels = scales::percent_format(scale=1) ) +
```

```
  # Labelling some of the major counties
  geom_text_repel(data=data_for_labelled_counties,
                  aes(label=County), colour="darkgrey",
                  min.segment.length = 0,
                  nudge_x=20,
                  nudge_y=-1) +

# Filling with custom palette (CVD friendly)
scale_colour_manual(values= custom_palette)+

# Titles, axis labels and theme
labs(title="Population Growth vs Commuting time from Dublin",
       subtitle="Years: 2006 to 2016",
       y="Population growth (%)",
       x="Commuting time from Dublin (Minutes)")+
theme_minimal()
```

Code for Figure 13

```
# Scatter plot with plotting each province differently
ggplot(df_data,
       aes(x=commute_time.min.,
           y=100* pop_growth,
           colour = Province))+

# Point size and transparency
  geom_point(size=3, alpha=0.8)+

# Y-axis scaling with percent format
  scale_y_continuous(
    limits = c(0, 30),
    labels = scales::percent_format(scale=1)
  ) +

# Filling with custom palette (CVD friendly)
  scale_colour_manual(values= custom_palette)+

# Labelling some of the major counties
  geom_text_repel(data=data_for_labelled_counties,
                  aes(label=County), colour="darkgrey",
                  min.segment.length = 0,
                  nudge_x=20,
                  nudge_y=-1) +

# Titles, axis labels and theme
labs(title="Population Growth vs Commute Time from Dublin",
       subtitle="Years: 2006 to 2016",
       y="Population growth (%)",
       x="Commuting Time from Dublin (Minutes)")+
```

```
theme_minimal()+

# Creates multiple plots each representing one Province
facet_wrap(~Province, labeller = label_both)
```

Code for Figure 14

```
# Scatter plots for commute time with Distance shape by provinces
ggplot(df_data,
       aes(x=commute_time.min.,
           y= KM,
           colour=100*pop_growth,
           shape = Province))+

# Selecting point size, transparency, and shape of each province
  geom_point(size=3, alpha=0.9)+
  scale_shape_manual(values = c( 17,19, 18,15))+

# Heat colours, from yellow as low and red as high pop growth
  scale_color_gradientn(colours = rev(heat.colors(3)), "Pop. Growth (%)")+

# Labelling some major counties
  geom_text_repel(data=data_for_labelled_counties,
                  aes(label=County), colour="darkgrey",
                  min.segment.length = 0,
                  nudge_x=20,
                  nudge_y=10) +

# Creating light blue dashed line passing by.
  geom_vline(xintercept = 150, color = "lightblue", linetype = "dashed") +
  geom_hline(yintercept = 150, color = "lightblue", linetype = "dashed") +

# Titles, axes labels and theme
  labs(title="Distance and Commute Time from Dublin",
       subtitle="Years: 2006 to 2016",
       y="Distance from Dublin (Km)",
       x="Commuting Time from Dublin (Minutes)")+

theme_minimal()
```

Code for Figure 15

```
# Scatter plot of time with distance, colour pop growth and divided by provinces
ggplot(df_data,
       aes(x=commute_time.min.,
           y= KM,
           colour=100*pop_growth,
          ))+
```

```
# Selecting point size, transparency
  geom_point(size=3, alpha=0.9)+

# Colour gradient for increasing population growth
  scale_color_gradientn(colours = rev(heat.colors(3)), "Pop. Growth (%)")+

# Labelling important counties
  geom_text_repel(data=data_for_labelled_counties,
                  aes(label=County), colour="darkgrey",
                  min.segment.length = 0,
                  nudge_x=20,
                  nudge_y=10) +
  geom_vline(xintercept = 150, color = "lightblue", linetype = "dashed") +
  geom_hline(yintercept = 150, color = "lightblue", linetype = "dashed") +


# Theme, titles and axes labels
  labs(title="Distance and Commute Time from Dublin for each Province",
       subtitle="Years: 2006 to 2016",
       y="Distance from Dublin (Km)",
       x="Commuting Time from Dublin (Minutes)")+

theme_minimal()+

# Dividing the scatter plot into multiple plots by province
facet_wrap(~Province,labeller = label_both)
```

Appendix (Section 6)

Aim for this weekly assignment was to redesign a road safety authority slide. There were some bullet points which should be satisfied through the visualisation or set of visualisations.

Reference for the stacked bar chart was taken from the code given in the module

- https://universityofgalway.instructure.com/courses/13767/files/1637923?module_item_id=509236&fd_cookie_set=1

Reference for the pie chart is again taken from the code given in the module

- https://universityofgalway.instructure.com/courses/13767/files/1637919?module_item_id=509235&fd_cookie_set=1

Code for Figure 16a

```
## Importing libraries

library(ggplot2)
library(dplyr)
library(tidyr)
library(colorspace)
library(forcats)

# Reading databases
df_fatal_db <- read.csv("RSA_Traffic_Fatalities_18_22_v2.csv")
head(df_fatal_db)

# Pivoting table from wide to long format
dail_long<-df_fatal_db%>% pivot_longer(cols=starts_with("X"), names_to = "year",
values_to = "fatalities")

# Converted year column to actual year values by removing prefix X
dail_long['year'][dail_long['year'] == 'X2018'] <- '2018'
dail_long['year'][dail_long['year'] == 'X2019'] <- '2019'
dail_long['year'][dail_long['year'] == 'X2020'] <- '2020'
dail_long['year'][dail_long['year'] == 'X2021'] <- '2021'
dail_long['year'][dail_long['year'] == 'X2021'] <- '2021'
dail_long['year'][dail_long['year'] == 'X2022'] <- '2022'

# Filtering each years data
dail_2018<-dail_long%>%filter(year=='2018')
dail_2019<-dail_long%>%filter(year=='2019')
dail_2020<-dail_long%>%filter(year=='2020')
dail_2021<-dail_long%>%filter(year=='2021')
dail_2022<-dail_long%>%filter(year=='2022')

# Defining orders for plots
bar_order = c("Driver",  "Pedestrian", "Passenger", "Motorcyclist", "Pedal Cyclist",
"Other")
```

```
# ggplot will create this plot in descending order
# however, we want to view it in ascending order
# That's why reversed the order
bar_order = rev(bar_order)

# combining dataframes in df_fatal
df_fatal<-rbind(dail_2018, dail_2019, dail_2020,dail_2021, dail_2022)

# Creating short variable with specified levels
df_fatal<-df_fatal%>%mutate(short = factor(df_fatal$Category,levels=bar_order))

# Negation operator defination
`%!in%` <- Negate(`%in%`)


bar_width<-0.6

# Creating stacked bar charts
plot1<-ggplot(df_fatal,
       aes(x = year , y = fatalities, fill = short))+
  geom_col(width = bar_width, size = 0.2, colour="white") +

# Condition for label and colour if fatalities > 5 or not to avoid overlapping
  geom_text(aes(label =  ifelse(fatalities>5, paste(fatalities, "Fatalities"), "")),
            position = position_stack(vjust = 0.5), size=2,
            colour = ifelse(df_fatal$short!='Other',"White","Black")) +
  geom_text(aes(label = ifelse(fatalities>0 & fatalities<5, paste(fatalities,
"Fatalities"), "")),
            position = position_stack(vjust =2.3), size = 2,
            color = "Black")+

# Y-axis scalling
  scale_y_continuous(limits= c(0,180),breaks = seq(0,180, by = 20),
expand=c(0,0))+

# X-axis scalling
  scale_x_discrete( expand = c(0, 0))+

# Titles and theme
  ggtitle("Fatality from different Modes of Transport ", subtitle = "From year 2018-
2022") +

# Margins and text size
  theme(axis.title.y = element_text(margin = margin(r = 10)),
        plot.margin = margin(10, 10, 3, 3),
        plot.title = element_text(size = 11),) +

  theme_minimal()+
```

```
#  Grids and legend
  theme(panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line(color = "gray", size = 0.3))+
  theme(panel.border = element_rect(color = "black", fill = NA, size = 0.5))+

  ylab("Fatalities") +
  xlab("Years")+
  labs(fill = "Category")

# Saving the plot
ggsave("plot1.jpeg", plot1, width = 6, height = 4)
```

Code for Figure 16b

```
# Creating a stacked bar chart for 2021 and 2022
# Specifying order and reversing
bar_order = c("2021", "2022")

bar_order = rev(bar_order)

# Binding to one data frame and creating factor variable short
df_fatal<-rbind(dail_2021, dail_2022)

df_fatal<-df_fatal%>%mutate(short = factor(year,levels=bar_order))

# Negation operator
`%!in%` <- Negate(`%in%`)

bar_width<-0.7

# Creating stacked bar chart with category vs fatalities and filled by year
ggplot(df_fatal,
       aes(x = Category , y = fatalities, fill = short))+
  geom_col(width = bar_width, size = 0.4, colour="white") +

# Condition for label and colour if fatalities > 5 or not to avoid overlapping
  geom_text(aes(label =  ifelse(fatalities>2, paste(fatalities, "Fatalities"), "")),
            position = position_stack(vjust = 0.5), size=1.8,
            colour = ifelse(df_fatal$short!='Other',"White","Black")) +
  geom_text(aes(label = ifelse(fatalities>0 & fatalities<5, paste(fatalities,
"Fatalities"), "")),
            position = position_stack(vjust =2.3), size = 2,
            color = "Black")+

# Y-axis scaling
  scale_y_continuous(limits= c(0,150),breaks = seq(0,150, by = 20),
expand=c(0,0))+
```

```
# X-axis scaling
scale_x_discrete( expand = c(0, 0))+

# Title, themes, grids and legend
  ggtitle("Fatality from different Modes of Transport ", subtitle = "In year 2021 and
2022") +

  theme(axis.title.y = element_text(margin = margin(r = 10)),
        plot.margin = margin(15, 15, 8, 8),
        plot.title = element_text(size = 11),) +

  theme_minimal()+
  theme(panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line(color = "gray", size = 0.3))+
  theme(panel.border = element_rect(color = "black", fill = NA, size = 0.5))+

  ylab("Fatalities") +
  xlab("Years")+
  labs(fill = “Years”)
```

Code for Figure 17

```
# Creating diverging barchart from a reference of 2021 fatalities
df_diff <- df_fatal %>%
  group_by(Category) %>%
  summarise(diff = fatalities[year == "2022"] - fatalities[year == "2021"])

# Plot the differences
plot <- ggplot(df_diff, aes(x = Category, y = diff, fill = diff >= 0)) +
  scale_y_continuous(limits= c(-20,30),breaks = seq(-20,30, by = 10),
expand=c(0,0))+
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +

# Colour used are CVD friendly, text for each bar and theme selection
  scale_fill_manual(values = c("#D55E00", "#009E73"), labels = c("Decrease",
"Increase")) +
  geom_text(aes(label = diff), vjust = ifelse(df_diff$diff >= 0, -0.5, 1.5), size = 3) +
  ggtitle("Change in Fatalities between 2021 and 2022") +
  ylab("Difference in Fatalities") +
  xlab("Category") +
  labs(fill = element_blank())+
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.y = element_line(color = "grey", size = 0.3),
        panel.background = element_rect(fill = "white",colour = "black") )

# Saving the plot
 ggsave("plot.jpeg", plot, width = 6, height = 3)
```

Code for Figure 18

```
# For labelling pie chart
library(ggrepel)

# Colour palette (CVD friendly)
mycols <- c("#D55E00",  "#009E73", "#0072B2", "#CC79A7", "#56B4E9",
"#F0E442")

# Plotting the palette
palette_plot(mycols, label_size = 3)

# Converting to percentage as used by the Pie chart in decreasing order of %
year="2022"
dail_2022$percent <- round(dail_2022$fatalities*100/sum(dail_2022$fatalities),0)
dail_2022<-dail_2022%>%arrange(-percent)

dail_2021$percent <- round(dail_2021$fatalities*100/sum(dail_2021$fatalities),0)
dail_2021<-dail_2021%>%arrange(-percent)

dail_2019$percent <- round(dail_2019$fatalities*100/sum(dail_2019$fatalities),0)
dail_2019<-dail_2019%>%arrange(-percent)

dail_2018$percent <- round(dail_2018$fatalities*100/sum(dail_2018$fatalities),0)
dail_2018<-dail_2018%>%arrange(-percent)

dail_2020$percent <- round(dail_2020$fatalities*100/sum(dail_2020$fatalities),0)
dail_2020<-dail_2020%>%arrange(-percent)

# List of dataframes
dail_year=list("2018"=dail_2018,
               "2019"= dail_2019,
               "2020"=dail_2020,
               "2021"=dail_2021,
               "2022"=dail_2022)

year<-"2022" # Creating the plot for 2022

# Creating short column for reordering the dataframe
dail_year[[year]]<-dail_year[[year]]%>%mutate(short =
factor(Category,levels=bar_order))

# Label position for each section of piechart
dail_2022_pie<- dail_year[[year]] %>%
  arrange(desc(percent)) %>%
  mutate(lab.ypos = cumsum(percent) - 0.5*percent)

# Creating the piechart for 2022
plot3<-ggplot(dail_2022_pie, aes(x = "",
                   y = percent, hue = short, fill = mycols)) +
```

```
# Divided section by black line
  geom_col( colour = "black", size=0.2, width = 1)+

# Starting at 0 degree
  coord_polar("y", start = 0) +

# Text and label with condition of percent>10 or not
  geom_text(aes(y = lab.ypos, label =
                ifelse((percent > 10), paste(Category, sprintf("%1.1i%%", percent),"(",
fatalities, ")", sep = " "), "")),
            size=2.3, colour = "white") +

  geom_text_repel(aes(y = lab.ypos  ,label =
                      ifelse((percent <= 10), paste(Category, sprintf("%1.1i%%",
percent),"(", fatalities, ")", sep = " "), "")),
                size=2, segment.color = "black", segment.size=0.2, nudge_x=0.7) +

# Title and legends
  ggtitle("Fatalities in 2022") +

  theme_void() +
  theme(legend.position = "none")

# Saving the pie chart
ggsave("plot3.jpeg", plot3, width = 4, height = 4)
```

Appendix (Section 7)

Aim for this weekly exercise is to create a Heatmap, where each tile represents each county's monthly cases per 100K diverge from the mean number of cases (per 100K) in that month.

References for creating a heatmap is taken from module code

- https://universityofgalway.instructure.com/courses/13767/files/1666520?module_item_id=517758&fd_cookie_set=1

Code for Figure 19a, 19b and 20

```
##Libraries imported

library(ggplot2)
library(dplyr)
library(lubridate)
library(colorspace)
library(colorblindr)
library(shinyjs)

## Custom Palette with divergence hcl wizard

hclwizard()
custom_palette <- diverging_hcl(n = 3, h = c(260, 0), c = 80, l = c(30, 90), power = 1.5)
palette_plot(custom_palette, label_size = 3)

## Checking for CVD friendly
cvd_grid(palette_plot(custom_palette, label_size = 3))

## Used reference of the present Heatmap available in the module to preprocess the data to create months, years, cases/100000 and other features.

IRL_counties_Covid19_df <- read.csv("IRL_counties_Covid19_df.csv")

IRL_counties_Covid19_df$DailyCCase[is.na(IRL_counties_Covid19_df$DailyCCase)] <-0

# extract year value from TimeStamp and stored in the dataframe
IRL_counties_Covid19_df$year <-
year(ymd(IRL_counties_Covid19_df$TimeStamp))
# extract month value from TimeStamp and stored in the dataframe
IRL_counties_Covid19_df$month <-
  month(ymd(IRL_counties_Covid19_df$TimeStamp))

# calculate monthly figures per county
IRL_covid_per_month <- IRL_counties_Covid19_df %>%
  group_by(month, year, CountyName) %>%
  summarise(total_cases_per_month = sum(DailyCCase), .groups = "drop")
```

```
# We need to normalise by population of each county.
# Get the population per county from the original data frame
county_pop <- IRL_counties_Covid19_df %>%
  filter(TimeStamp == "2021-03-01") %>% select(CountyName, Population)

# add population figure for each county using a left outer join
IRL_covid_per_month <-
  merge(x = IRL_covid_per_month,
        y = county_pop,
        by = "CountyName",
        all.x = TRUE)


# calculate total number cases per county normalized by 100,000 of population
IRL_covid_per_month <- IRL_covid_per_month %>%
  mutate(total_cases_per_month_per_100k =  round((
    100000  * total_cases_per_month / Population
  ), 0))

# Create a date object to represent each month - year combination
# As a date object has to have a day value I have chosen the 1st of the month
# Using a Date format will keep the dates in order on the x-axis
IRL_covid_per_month <- IRL_covid_per_month %>%
  mutate(date = as.Date(paste0(year, "-", month, "-01", sep = "")))

# select the subset of columns  needed to plot the heatmap
Plot_data_IRL_covid_per_month <- IRL_covid_per_month[, c(7, 1, 6)]

## Plotting the heatmap

monthly_mean_cases <- IRL_covid_per_month %>%
  group_by(date) %>%
  summarise(mean_cases_per_100k = mean(total_cases_per_month_per_100k))

# Merging the data of mean cases that we created above with our main dataframe
by date
heatmap_data <- merge(IRL_covid_per_month, monthly_mean_cases, by =
"date", all.x = TRUE)

# Calculating the difference between cases and mean per month per 100k cases.
heatmap_data <- heatmap_data %>%
  mutate(diff_from_mean = total_cases_per_month_per_100k -
mean_cases_per_100k)

# Plotting heatmap
heat_map <- ggplot(heatmap_data, aes(x = date, y = CountyName, fill =
diff_from_mean)) +
  geom_tile() +

# Using Gradient2 to fill low, mid and high as custom divergence value by HCL
```

```
  scale_fill_gradient2(low = custom_palette[1], mid = custom_palette[2], high =
custom_palette[3], midpoint = 0) +
# Giving title name and y axis name
  labs(title = "Divergence of Monthly Cases/100,000 from Mean",
       y = "County") +

# Theme settings for ticks, lines, background and margins
  theme(
    axis.title.x = element_blank(),
    axis.text.y = element_text(size = 6),
    axis.ticks.x = element_line(linewidth = 0.3, colour = "darkgrey"),
    axis.line.x = element_line(linewidth = 0.1, colour = "darkgrey"),
    axis.ticks.y = element_blank(),
    axis.line.y = element_blank(),
    panel.background = element_blank(),
    panel.grid.major = element_blank(),
    plot.margin = unit(c(
      t = 0.0,
      r = 0.0,
      b = 0.5,
      l = 0.5
    ), "cm"),
    plot.title = element_text(size = 12, face = "bold")
  ) +

# Theme settings for legend adjustment
  theme(
    legend.text = element_text(size = 8),
    legend.position = "top",
    # move to the bottom
    legend.justification = "left",
    legend.title = element_blank(),
    legend.key.size = unit(1.5, "line"),
    legend.spacing.x = unit(0.5, 'cm'),
    legend.background = element_rect(
      fill = "white",
      linewidth = 0.5,
      colour = "white"
    )
  )+
# Theme for x axis bins/breaks and angle adjustments.
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  scale_x_date(date_breaks = "2 month", date_labels = "%Y-%m")

# Visualising and Saving the JPEG of the plot.
heat_map
ggsave("heat_map.jpeg", heat_map, width = 6, height = 5)
```