CSCI4210U
Web Application Development
Winter, 2024
Faculty: Dr Livingstone

(TIME ALLOWED: 120 minutes)

This examination paper comprises 7 pages

Questions are coded as follows:

- Questions are worth various marks which will be shown as thus: (5 pts)
- The total number of points available for this examination is **40**.

Exam format

- Open book

Permitted materials:

- Your notes, lecture notes
- Your existing code, including your worksheet code.
- Any documentation online (tidyverse, sf, ggally, ggplot2 books, etc.).

Permitted software and services, and their permitted use:

- Your code editor of choice (RStudio is strongly recommended)
- Web browser (Firefox, Chrome, etc)
- Searching on websites (Google, Bing, stackoverflow, ChatGPT)

Prohibited services and behaviours:

- Sharing or showing the exam with any person(s), website, app, or similar service.
- Asking for help on Discord, Stack Overflow, homework help websites, GitHub, direct messaging, texting, visual, and any other form of communication. Communicating with anyone other than the exam invigilators during the exam period.
- Sharing your code with others.

The standard Ontario Tech academic integrity policy applies to this exam.

Instructions

This exam is to be completed and submitted to Canvas before the exam period has finished. You are permitted to use any files (including PDFs, e-books, source code) that you have stored on your laptop. In addition, you will also be permitted to access any R related documentation:

- https://tidyverse.tidyverse.org/
- https://cran.r-project.org/web/packages
- https://r-spatial.github.io/sf/
- https://ggplot2-book.org/index.html

Overview

This exam involves the creation of five distinct data visualisations. These include:

1. Base R timeseries graphic using UK lung disease data (5 pts)
2. Timeseries with European financial data (10 pts)
3. Relationships with NBA sports data (10 pts)
4. Data manipulation and a choropleth map of flight traffic (15 pts)

Question 1 must use base graphics, while questions 2-4 must use ggplot2. Example figures accompany all questions. Use these as guides in answering the questions.

Exam files

1. Download the exam file accompanying this exam – 4210_final.zip.
2. Use the provided R script file for this exam.
3. Insert your name and Student ID into the file header.

How to submit

Once you have finished, or once the exam period is ending, submit your .R script file. Only filetypes of ".R" will be accepted. Do not compress (zip) your file. Insert your name and Student ID into the file header.

You should begin this process at least 5 minutes before the test completes to ensure you upload your files on time.

Note: A grace period of 2 minutes will be given to allow for any potential technical issues (Canvas access is slow etc).

Submissions beyond this grace time will not be accepted and will be marked as 0.

Tip: Abuse the grace period and you will do so at your own peril. The physical equivalent to violating this rule is not handing in your physical exam paper when they are collected.

Questions

Question 1. Timeseries plot with Base R graphics (5pts)

The variable d_q1 contains timeseries data on monthly lung disease deaths in the UK. Using Base R graphics, generate a stacked bar plot of the data. Using RColorBrewer, create a palette of the necessary colour codes using the qualitative palette “Pastel1”. Label your axes. Give your graphic a title. Your graphic should match Figure 1.

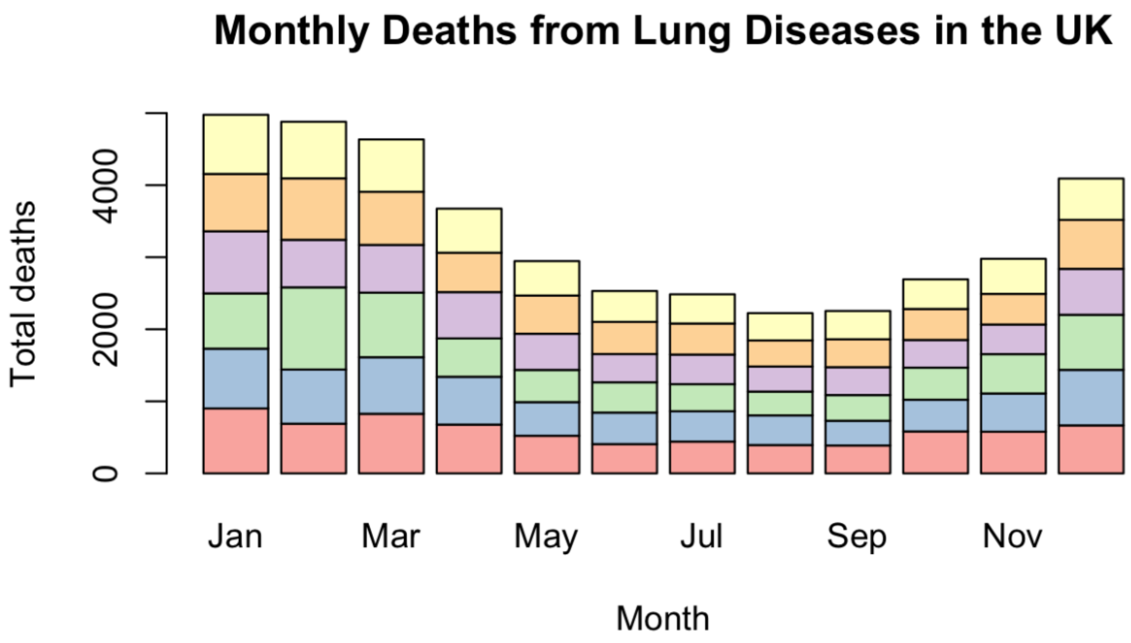

Figure 1. Monthly death from lung disease in the UK, coloured by year.

Question 2. Time series with financial data (10 pts)

The variable `d_q2` contains financial data from the EU Stock Exchange, consisting of daily closing prices of Germany DAX, Switzerland SMI, France CAC, and UK FTSE. The data are sampled in business time. Create a graphic that visualizes x=Dax and y=FTSE. Use appropriate `geom_` layers to show how these values change over time, as indicated in the `index` column. Use a line width=1.5 and point size=1.5. Add a straight line using `geom_abline()`, with an `intercept` and `slope` of 1, colour `firebrick`. Set both axis limits to [1000, 6500]. Label your legend "Time". Colour your geoms with the `viridis` colour scale. Your graphic should match Figure 2.

Hint: *if your graphic has many unwanted connecting lines, you may need to arrange your data first!*

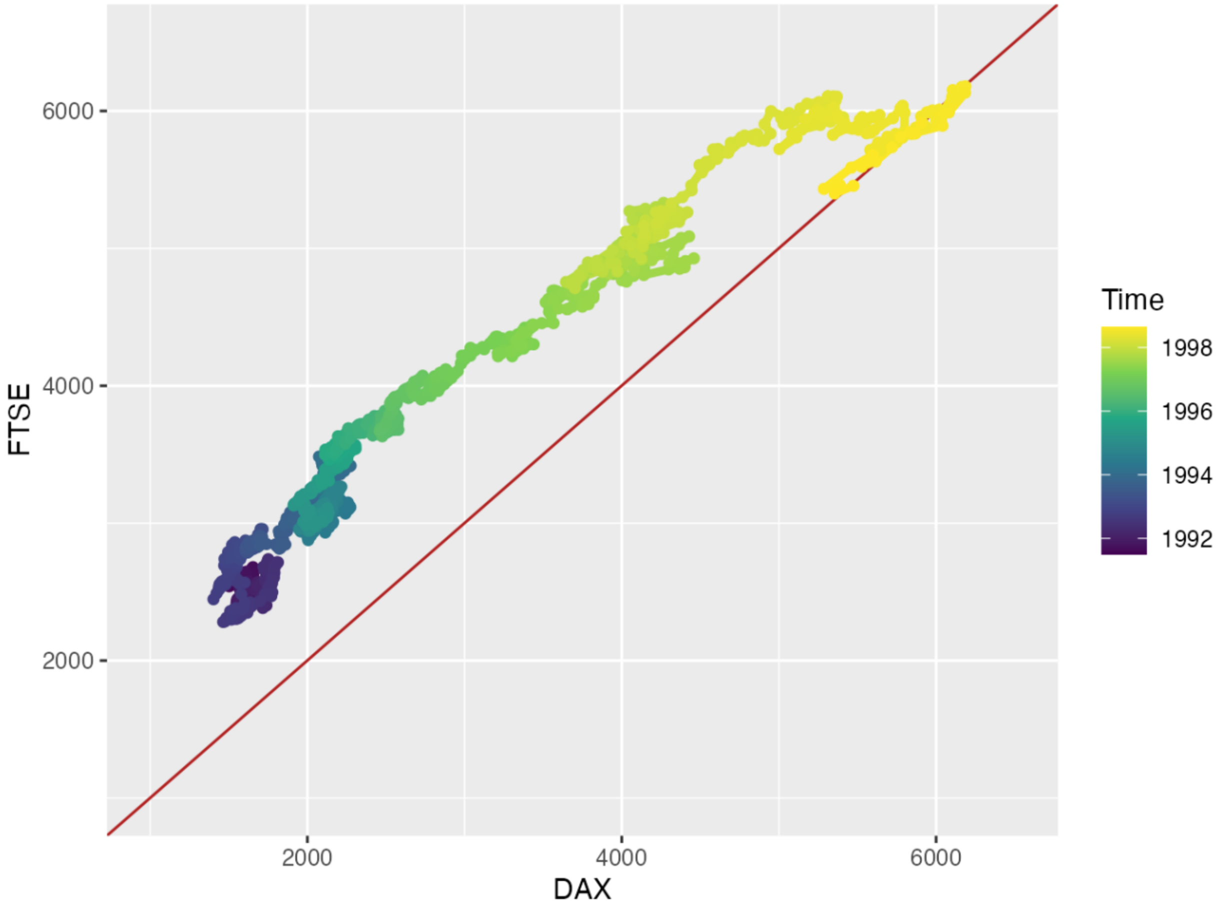

Figure 2. Trends in the European financial markets in the 1990s.

Question 3. Relationships with NBA data (10 pts)

The variable `d_q3` contains performance data on a range of NBA players. The data show a breakdown of players by `name` (character), type of performance `statistic` (character), and `scaled_score` (numeric, -2:2). Create a heatmap of the data, where tile shading reflects a color gradient of `scaled_score`, ranging from: "`white`" to "`steelblue`". Give the tiles a white border. Update the names of the x and y axis labels to "Player" and "Performance statistic" respectively, and Legend to "Performance %". The default legend has 5 ticks, at locations -2, -1, 0, 1, 2. Give these breaks the labels 0, 25, 50, 75, 100. Apply a minimal theme. Rotate x-axis labels by -45^0. Give your graphic a title. Your graphic should match Figure 3.

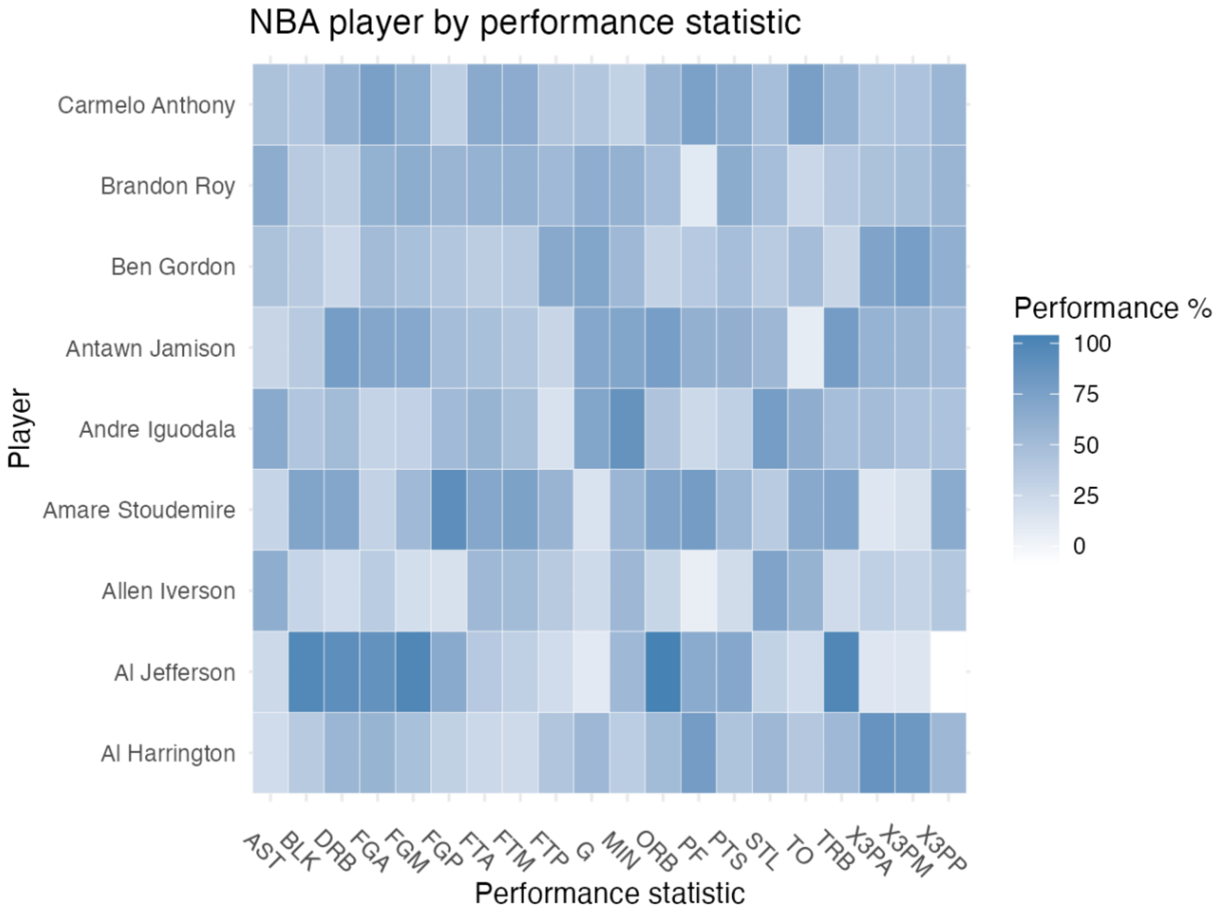

Figure 3. Heatmap of NBA player statistics

Question 4. Choropleth map (15 pts)

Here you will produce a map of the USA, shading airports by the number of yearly inbound flights they receive. Variables for this question include:

- `d_q4_shp` - geographic data of the USA in a 'simple feature' `sf` object.
- `d_q4_flights` - information including flight date, origin, and destination airport codes.
- `d_q4_airports` – airport information including airport code, name, and geographic latitude and longitude coordinates.
- `d_q4_wrangeled` – a pre-wrangled tibble for mapping. Use this only if you are unable to complete Part A (wrangling) and unable produce the requested d_q4.

Part A: Wrangling (5 pts) - Begin by joining the `flights` and `airports` tibbles. Use the destination airport column in flights, as we want inbound flight airport information. Drop na's from the data with na.omit (or use a more restrictive join). Your tibble should be of size [48894, 8]. Next, filter airports to keep those where `longitude` > -130, and `latitude` > +25; this will remove Hawaii and other islands. Finally, convert your tibble to an sf object with the following line of code:

```
st_as_sf(coords = c("lon", "lat"), crs = 4167, remove = FALSE)
```

After this, your tibble should be of size [48788, 9]. Next, count the number of flights at each destination airport, summarising the count in a new column '`n`'.
***Hint:** try `n()`. Save your result in `d_q4`. Your tibble should now be of size [96, 3].*

***Note:** If you are unable to complete Part A, use `d_q4_wrangeled` instead of d_q4 for Part B. You will get zero marks for Part A if you use the pre-wrangled data.*

Part B: Mapping (10 pts) - Using the geographic data in `d_q4_shp`, create a map of the USA, with fill = "`gray97`", colour = "`black`". Overlay airports using `d_q4` or *`d_q4_wrangeled`*, colouring by `n`, and adjusting point size by `n`. Colour the points using a ColorBrewer scale with the "`PuRd`" palette. Set 3 break points, ranging from the min to the max values of `n`. ***Hint:** try using `seq()`*. Give your legend the labels "Quiet", "Moderate", and "Busy". Disable the legend for the size aesthetic. ***Hint:** `guides()`*. Set the CRS coordinate to `st_crs(4326)`. Give your plot a title and subtitle. Apply a light theme. Your graphic should match Figure 4.

***Note:** if you used d_q4_wrangled, your graphic should instead match Figure 4b. Yes, they intentionally look different.*

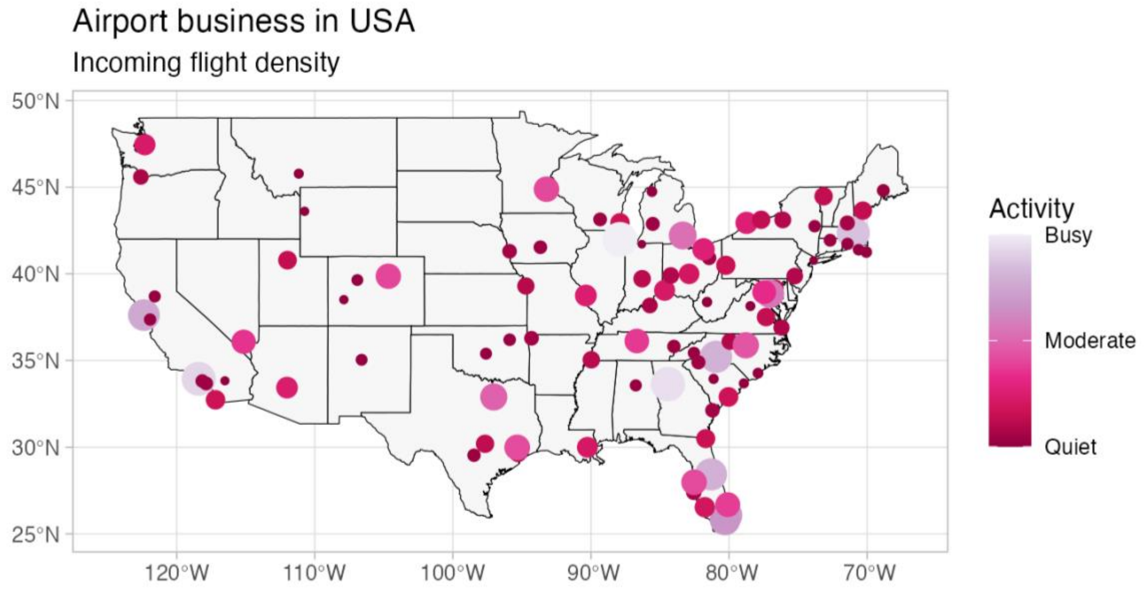

Figure 4. Map of airports in USA showing inbound flight activity.

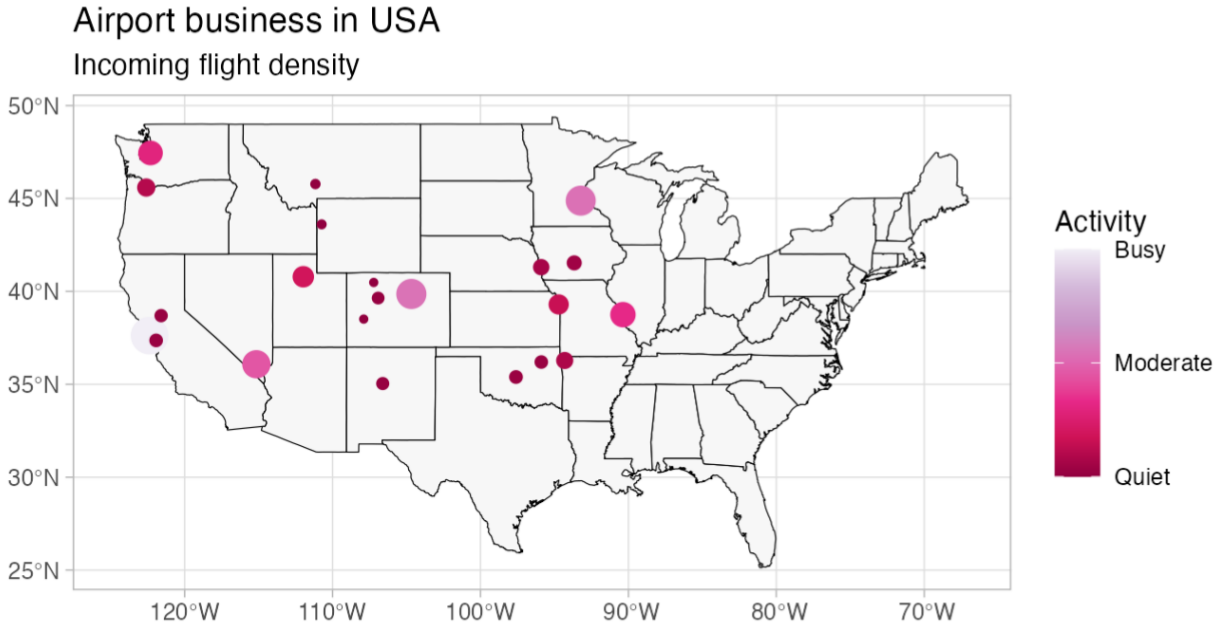

Figure 4b. Map of airports in USA using pre-wrangled data.