

Testing Typst - Tables, graphs, animations?

The purpose of this document is to just test out rendering the `typst` format using Quarto. In particular I want to check that one can produce coloured tables, faceted graphs, and perhaps animations.

At the same time, I am thinking about how best to display the evolution of available seats over time. In essence, each single performance (i.e. on a given date and time) would draw a path/curve in the plane [days till performance, number of available seats], which is why I would like to try an animation visualization. Afterwards, I'll be thinking how to "fit" a curve that describes the data, i.e. an "optimal" estimate of all the paths drawn out by the performances.

A table created using the `gt` package, with some colors

First, let's use the `gt` package to nicely display the list of shows in my dataset.

Shows in the dataset

Show name	Opening date	Closing date	Number of performances	Number of records
A Christmas Carol (ish)	Nov 16, 2024	Dec 31, 2024	54	1845
Oedipus	Oct 4, 2024	Jan 4, 2025	69	2148
Unicorn	Feb 4, 2025	Apr 26, 2025	93	558

Data downloaded on 2024-11-21

A Plotly animation

Next, I want to look at a single show - Oedipus - and visualize the evolution of available seats as the performance data approaches. Let's try an animation using Plotly.

Actually, Plotly animations cannot be rendered into Typst. I was simply confused by some unclear documentation and error messaging. The below code block won't get evaluated and is included just for reference.

```

dat |>
  filter(show_name == "Oedipus") |>
  # Sort by request date so that the frames are ordered correctly,
  # and by DESCENDING performance date so that in the animation
  # the correct dot disappears when days till performance becomes 0 for the
  performance
  arrange(request_date, desc(performance_date)) |>
  mutate(frame_var = as.character(request_date)) |>
  plot_ly(
    x = ~days_till_performance,
    y = ~available_seat_count,
    color=~performance_type,
    text=~paste0("Performance time: ", performance_timestamp),
    frame = ~frame_var,
    type = "scatter",
    mode = "markers",
    marker = list(size=10, opacity=0.7, line=list(width=1, color="gray")),
    colors=c("deeppink", "navy")) |>
  animation_slider(
    currentvalue = list(prefix="As seen on ")
  ) |>
  layout(
    title = "Oedipus - Number of available seats as the performance date
    approaches",
    xaxis = list(title = "Number of days till performance"),
    yaxis = list(title = "Number of available seats"),
    legend = list(title=list(text="Evening or matinee performance"), x=0.1,
    y=0.9)
  )

```

A static visualisation

Instead of the animation, let's display a series of plots, each one displaying the number of available tickets vs the number of days till the performance, based on information obtained (requested from the API) on the given date.

I am not particularly happy with this visualization. But the point was try out the Typst format rendering anyway.

Oedipus - Number of available seats as the performance date approaches

Each subgraph shows information as obtained on the given date.
A datapoint corresponds to a single performance date.

