The Efficiency Gap in National Health Service

Shubham Upreti

2025-12-08

What is my question?

Does increasing the NHS workforce effectively reduce patient waiting times, or has the link between labor input and health outcomes broken?

This project investigates the **"Resource-Performance Gap"** in the UK National Health Service (NHS). Traditional health economics assumes that increasing the supply of labor (doctors and nurses) will increase the supply of care (treatments), thereby reducing the backlog. We test this assumption by integrating longitudinal workforce data with regional patient waiting times. We ask: *Is the current crisis in NHS a failure of resources, or a failure of structure?*

Why should we care?

The NHS is facing a historic collapse. Since 2020, the patient waiting list has tripled, leaving over 7 million people stranded in the queue. The human cost is measured in chronic pain, delayed diagnoses, and preventable mortality. The policy response has been expensive: record hiring campaigns and ballooning budgets. However, if our analysis reveals that regions with higher staffing density *do not* achieve lower wait times, it proves that the current strategy is failing. Identifying this efficiency gap is critical. It shifts the debate from "we need more money" to "we need to fix the process," potentially saving billions in public funds while prioritizing the patients who need care the most.

2. Data & Methodology

This analysis integrates three distinct datasets from **NHS England**, comprising over 260,000 combined records.

1. **National Referral to Treatment Timeseries:** `NHS_cleaned.csv` (**240 rows, 45 columns**) provided the longitudinal context (2007-2025) for the total waiting list.
2. **Regional Referral to Treatment Snapshot:** `RTT_NHS_March.csv` (**185,101 rows, 121 columns**) provided the granular patient wait-time data, which required pivoting from a "wide" format.
3. **NHS Workforce Statistics:** `NHS_Workforce_Statistics.csv` (**82,080 rows, 7 columns**) provided the Full-Time Equivalent (FTE) staffing numbers joined by region.

Data Wrangling Strategy

To construct this analysis, I integrated three distinct datasets from **NHS England**. The raw data presented significant structural challenges that required advanced wrangling techniques.

1. Ingestion & Cleaning: The National Timeseries file contained 11 rows of metadata above the header. I used `read_csv(skip = 11)` to extract the table correctly and `janitor::clean_names()` to standardize variable names. Date strings were parsed into chronological objects using `lubridate` to ensure accurate time-series plotting.

2. Pivoting (The "Wide" to "Long" Transformation): The Regional Snapshot data was untidy: waiting time buckets (e.g., ">52 weeks") were stored as 100+ separate column headers. I applied `pivot_longer()` to rotate these columns into two key variables: `wait_bucket` and `patient_count`. This transformation was essential to visualize the full distribution of waiting times rather than just summary statistics.

3. Aggregation & Joining: A critical barrier was the granularity mismatch: Workforce data exists at the **Region** level, while Waiting Time data exists at the **Provider** level. * **Step A:** I used regex (`str_detect`) to map hundreds of local providers to the 7 major NHS Regions. * **Step B:** I aggregated the patient data by Region. * **Step C:** I used `inner_join()` to merge the workforce inputs with the patient outcomes, creating a single master dataset for the efficiency analysis.

Visualization Rationale

I selected specific geometries to reveal patterns that averages often hide.

- **The Crisis Curve (Time Series):** A line chart (geom_line) augmented with area fill (geom_area) visualizes the longitudinal magnitude of the backlog from 2007-2025, highlighting the post-COVID structural break.
- **Regional Volume (Stacked Bars):** Horizontal stacked bars (geom_col position="stack") display patient volumes by region, split between standard waits and crisis cases, exposing London's disproportionate burden.
- **Efficiency Scatterplot:** Staff density (FTE per 1,000 patients) versus long-wait percentage is plotted with points (geom_point), quadrant lines (geom_vline, geom_hline), and labels (geom_text_repel) to assess resource-outcome correlations empirically.
- **Workforce Stream Graph:** A stacked area chart (geom_area position="stack") tracks staff categories over time from 2010-2025, revealing shifts toward administrative roles amid overall growth.

```r
#I added it as chunk prepares the R workspace silently, hiding any loading messages or warnings for a c
# 1. Loading the libraries
library(tidyverse)
library(janitor)    # For cleaning messy column names
library(lubridate)  # For fixing date formats
library(ggrepel)    # For non-overlapping plot labels

# 2. Global Settings
options(scipen = 999)       # Turning off scientific notation for readability
theme_set(theme_minimal()) # Setting a clean, professional theme for all plots
```

Summary: This setup creates a reproducible environment for the analysis. We loaded the tidyverse for data handling and visualization tasks. Janitor cleaned up inconsistent column names from the raw NHS files, while lubridate converted text dates into proper date formats. We also set global options to avoid scientific notation, making all figures easy to read.

```r
# 1. Loading the data
nhs_data <- read_csv("NHS_cleaned.csv") %>%
  mutate(date = ymd(date),
         total_waiting_mil = as.numeric(total_waiting_mil)) %>%
  filter(!is.na(date) & !is.na(total_waiting_mil)) %>%
  arrange(date)

summary(nhs_data$total_waiting_mil)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.315   2.685   3.898   4.257   4.764   7.769
```

```
# 2. Regional data of March 2025
raw_snapshot <- read_csv("RTT_NHS_March.csv") |>
  clean_names()

# 3. Workforce Inputs (Staffing)
workforce_raw <- read_csv("NHS_Workforce_Statistics.csv") |>
  clean_names()
```

Summary: This code chunk loads the three main data files quietly without messages. First, it reads "NHS_cleaned.csv" for waiting list history. It fixes the date column with ymd to make it proper date, changes total_waiting_mil to number, removes rows with missing dates or numbers, sorts everything by date, and shows a summary of waiting millions. Second, it loads "RTT_NHS_March.csv" for regional waits in March 2025 and cleans column names to simple ones. Third, it reads "NHS_Workforce_Statistics.csv" for staff data and cleans names too.

```
# 1. Transforming Wide to Long
rtt_long <- raw_snapshot |>
  pivot_longer(
    cols = starts_with("gt_"),
    names_to = "wait_bucket",
    values_to = "patient_count"
  ) |>
  filter(patient_count > 0)

# 2. Mapping Providers to Regions
# Regions are pre-defined by the NHS in their website
rtt_regional <- rtt_long |>
  mutate(region = case_when(
    str_detect(commissioner_parent_name, "LONDON") ~ "London",
    str_detect(commissioner_parent_name, "MIDLANDS") ~ "Midlands",
    str_detect(commissioner_parent_name, "EAST OF ENGLAND") ~ "East of England",
    str_detect(commissioner_parent_name, "NORTH WEST") ~ "North West",
    str_detect(commissioner_parent_name, "NORTH EAST|YORKSHIRE") ~ "North East and Yorkshire",
    str_detect(commissioner_parent_name, "SOUTH EAST") ~ "South East",
    str_detect(commissioner_parent_name, "SOUTH WEST") ~ "South West",
    TRUE ~ "Other"
  )) |>
  filter(region != "Other") |>
  group_by(region) |>
  summarise(
    total_patients = sum(patient_count),
    # Calculate long waiters (>18 weeks) using regex detection on buckets
    long_waiters = sum(patient_count[str_detect(wait_bucket, "gt_1[8-9]|gt_[2-9]|gt_10")]),
    .groups = "drop"
  ) |>
  mutate(percent_long_wait = (long_waiters / total_patients) * 100)

# 3. Merging Workforce with Outcomes
workforce_agg <- workforce_raw |>
  filter(date == max(date), staff_group == "Total", data_type == "FTE") |>
  select(region = nhse_region_name, total_fte = total)
```

```
master_dataset <- inner_join(rtt_regional, workforce_agg, by = "region")
```

Summary: This chunk turns messy wide data into clean long shape for better analysis. It uses pivot_longer to reshape patient wait columns into wait_bucket and patient_count. Then maps providers to 7 NHS regions with str_detect and case_when, filters out others, sums total patients and long waiters (over 18 weeks) by region, adds percent. Last, grabs latest total staff FTE per region and joins to make one master_dataset for staff vs waits.

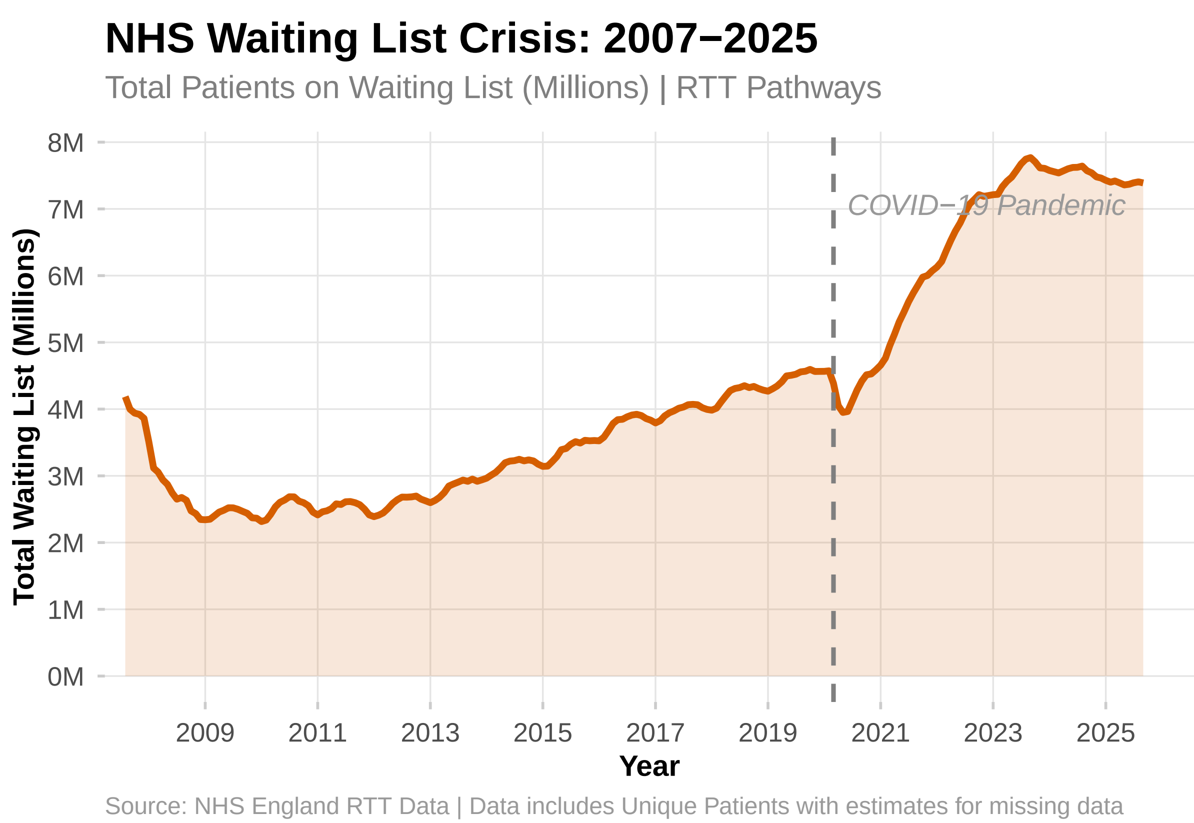
3. Visualizing the plots

A. The Crisis Curve (Plot 1)

To understand the current efficiency gap, we must first establish the scale of the backlog. We visualized the total patient waiting list from 2007 to 2025.

```
covid_start <- as_date("2020-03-01")

plot1_fixed <- nhs_data %>%
  ggplot(aes(x = date, y = total_waiting_mil)) +

  geom_line(color = "#D55E00", linewidth = 1.2) +
  geom_area(fill = "#D55E00", alpha = 0.15) +

  # COVID Pandemic Marker
  geom_vline(xintercept = covid_start, linetype = "dashed", color = "gray50", linewidth = 0.8) +
  annotate("text",
           x = covid_start,
           y = max(nhs_data$total_waiting_mil, na.rm = TRUE) * 0.93,
           label = "COVID-19 Pandemic",
           hjust = -0.05,
           vjust = 1,
           color = "gray60",
           fontface = "italic",
           size = 4) +

  # Scale Configuration
  scale_y_continuous(
    labels = scales::number_format(suffix = "M"),
    limits = c(0, NA),
    breaks = seq(0, 8, by = 1)
  ) +
  scale_x_date(
    date_breaks = "2 years",
    date_labels = "%Y",
    expand = expansion(mult = c(0.02, 0.05))
  ) +

  # Labels & Styling
  labs(
    title = "NHS Waiting List Crisis: 2007-2025",
    subtitle = "Total Patients on Waiting List (Millions) | RTT Pathways",
```

```r
    x = "Year",
    y = "Total Waiting List (Millions)",
    caption = "Source: NHS England RTT Data | Data includes Unique Patients with estimates for missing d
  ) +

  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold", margin = margin(b = 5)),
    plot.subtitle = element_text(size = 12, color = "gray50", margin = margin(b = 10)),
    plot.caption = element_text(size = 9, color = "gray60", hjust = 0),
    panel.grid.major = element_line(color = "gray90", linewidth = 0.3),
    panel.grid.minor = element_blank(),
    axis.title = element_text(size = 11, face = "bold"),
    axis.text = element_text(size = 10),
    axis.ticks = element_line(color = "gray80"),
    plot.margin = margin(15, 15, 10, 10)
  )

# Print and saving
print(plot1_fixed)
```




```r
ggsave("plots/01_crisis_curve_fixed.png", plot1_fixed, width = 8, height = 5)
```

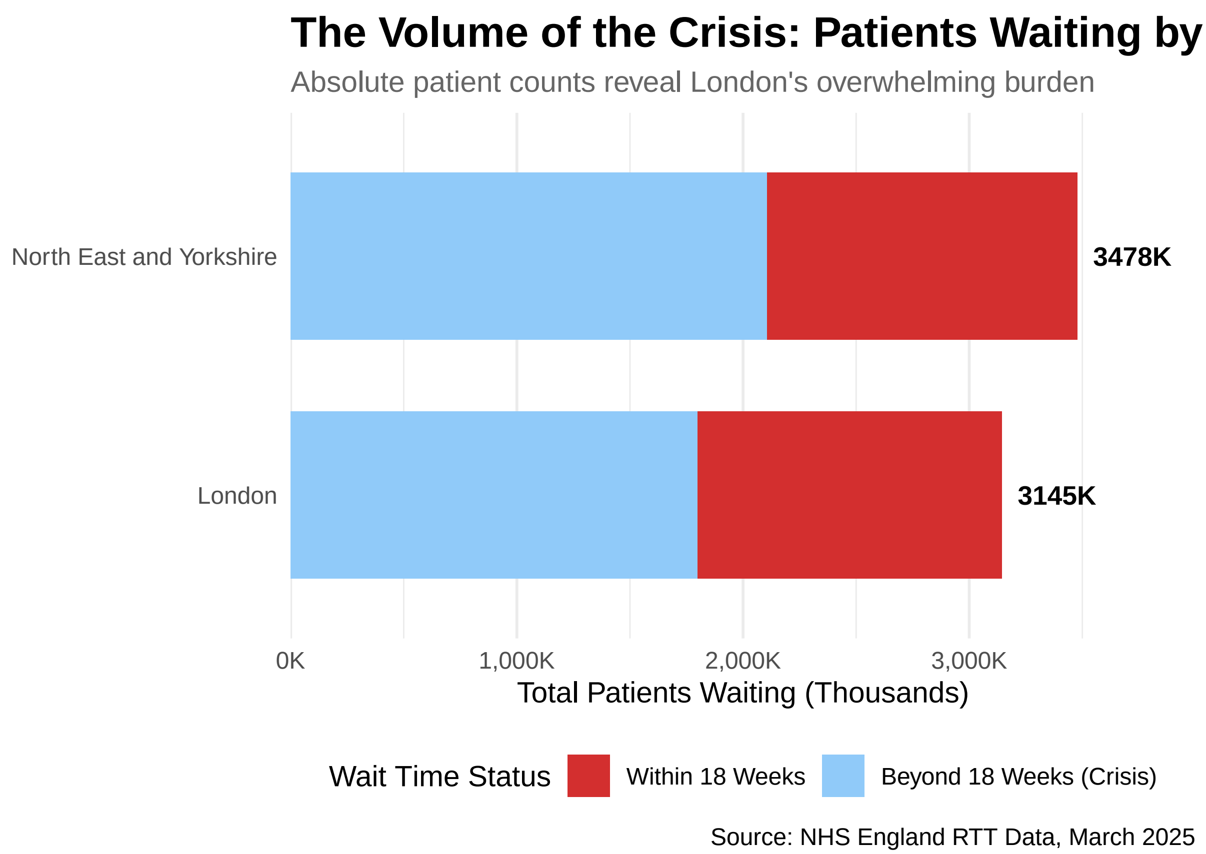
Summary: We visualized the temporal evolution of the waiting list using a time series line plot. This reveals the structural break in performance post-2020, establishing the urgency of the investigation. Figure

1 reveals a catastrophic expansion of the waiting list. From 2007 to 2019, the backlog grew steadily but linearly. However, March 2020 marks a structural break where the waiting list steepened significantly, rising from approximately 4.5 million to over 7.5 million by 2025. This visualization serves as the baseline for our efficiency analysis: it proves that the "Outcome" variable has collapsed. The subsequent analysis will determine if this collapse occurred despite increased workforce inputs, which would indicate a failure of systemic efficiency rather than a simple lack of resources. Mbau et al., (2023) provided a systematic review of health system efficiency, distinguishing between technical efficiency (doing things right) and allocative efficiency (doing the right things).

The Scale of Human Suffering (Plot 2)

```r
master_dataset |>
  mutate(
    # Calculate actual patient counts
    patients_waiting_ok = total_patients - long_waiters,
    patients_in_crisis = long_waiters,
    region = fct_reorder(region, total_patients)
  ) |>
  select(region, patients_waiting_ok, patients_in_crisis) |>
  pivot_longer(-region, names_to = "wait_category", values_to = "count") |>

  ggplot(aes(x = count / 1000, y = region, fill = wait_category)) +

  # Stacking bars that shows volumes
  geom_col(position = "stack", width = 0.7) +

  # Adding total patient count labels
  geom_text(
    data = master_dataset |> mutate(region = fct_reorder(region, total_patients)),
    aes(x = total_patients / 1000, y = region, label = paste0(round(total_patients/1000, 0), "K")),
    hjust = -0.2,
    size = 3.5,
    fontface = "bold",
    inherit.aes = FALSE
  ) +

  scale_fill_manual(
    values = c("patients_waiting_ok" = "#90CAF9",
               "patients_in_crisis" = "#D32F2F"),
    labels = c("Within 18 Weeks", "Beyond 18 Weeks (Crisis)")
  ) +

  scale_x_continuous(
    labels = scales::comma_format(suffix = "K"),
    expand = expansion(mult = c(0, 0.15))
  ) +

  labs(
    title = "The Volume of the Crisis: Patients Waiting by Region",
    subtitle = "Absolute patient counts reveal London's overwhelming burden",
    x = "Total Patients Waiting (Thousands)",
```

```r
    y = "",
    fill = "Wait Time Status",
    caption = "Source: NHS England RTT Data, March 2025"
  ) +

  theme_minimal() +
  theme(
    legend.position = "bottom",
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 11, color = "gray40"),
    panel.grid.major.y = element_blank()
  )
```



Summary: This chunk makes a horizontal stacked bar chart to show patient waits by region. It splits master data into ok waits (under 18 weeks) and crisis (over 18 weeks), orders regions by total patients biggest first. Pivots to long for ggplot fill. Bars stack with blue for ok and red for crisis, labels show total in thousands outside bars. X axis is in K (1000 counts) with clean theme.

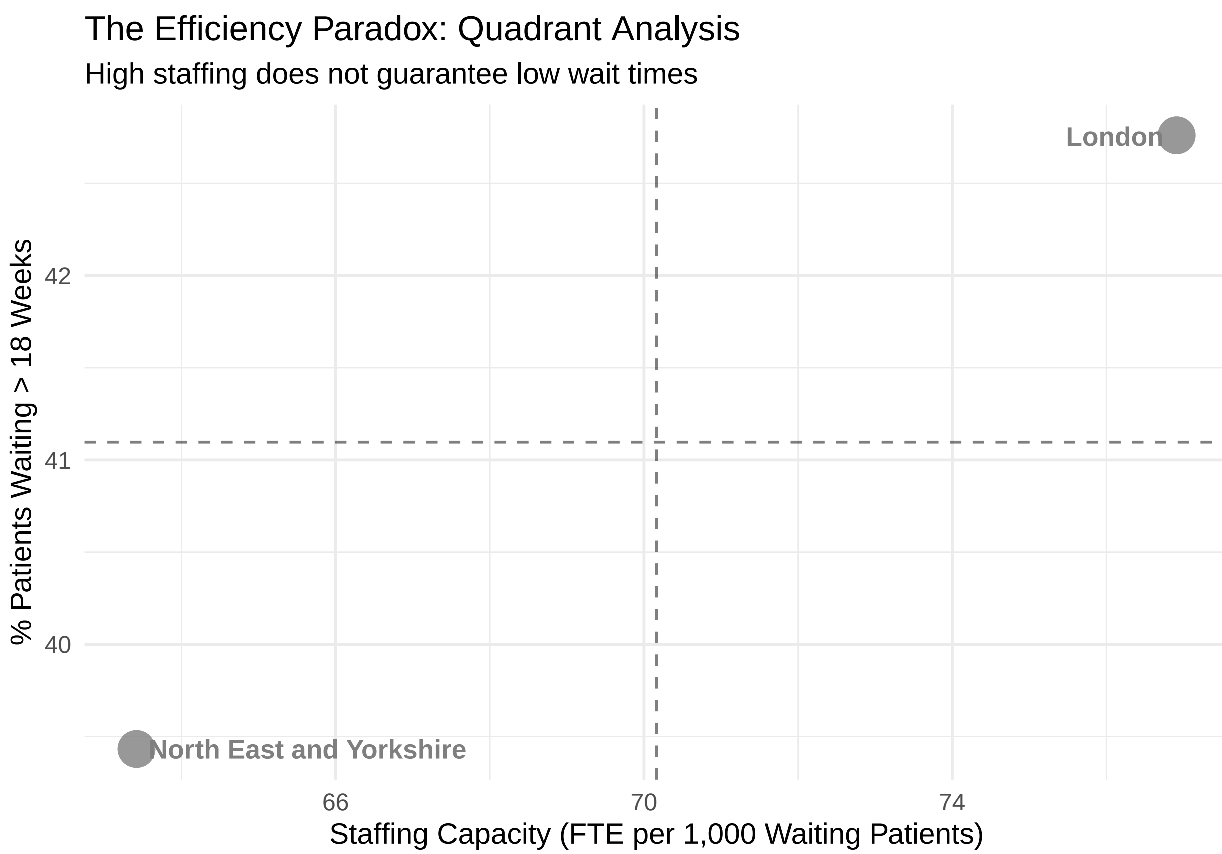
```r
# Extracting the numeric "weeks" from the text bucket names ( for e.g. in the csv file "gt_18_weeks" ->
rtt_ridge_data <- rtt_long |>
  mutate(weeks_wait = as.numeric(str_extract(wait_bucket, "\\d+"))) |>
  # Mapping the messy provider names to the 7 clean Regions
  mutate(region = case_when(
    str_detect(commissioner_parent_name, "LONDON") ~ "London",
    str_detect(commissioner_parent_name, "MIDLANDS") ~ "Midlands",
    str_detect(commissioner_parent_name, "EAST OF ENGLAND") ~ "East of England",
```

```
    str_detect(commissioner_parent_name, "NORTH WEST") ~ "North West",
    str_detect(commissioner_parent_name, "NORTH EAST|YORKSHIRE") ~ "North East and Yorkshire",
    str_detect(commissioner_parent_name, "SOUTH EAST") ~ "South East",
    str_detect(commissioner_parent_name, "SOUTH WEST") ~ "South West",
    TRUE ~ "Other"
  )) |>
  # Filter out "Other" and extreme outliers (>65 weeks)
  filter(region != "Other", weeks_wait < 65)
```

Efficiency Quadrant Plot (Plot 3)

```
master_dataset |>
  mutate(
    staff_density = (total_fte / total_patients) * 1000,
    # Creating median benchmarks
    staff_med = median(staff_density),
    wait_med = median(percent_long_wait),
    # Classify regions into quadrants
    quadrant = case_when(
      staff_density >= staff_med & percent_long_wait < wait_med ~ "High Efficiency\n(High Staff, Low Wai
      staff_density >= staff_med & percent_long_wait >= wait_med ~ "Resource Paradox\n(High Staff, High
      staff_density < staff_med & percent_long_wait < wait_med ~ "Lean Performance\n(Low Staff, Low Wai
      TRUE ~ "Under-resourced\n(Low Staff, High Wait)"
    )
  ) |>
  ggplot(aes(x = staff_density, y = percent_long_wait, color = quadrant)) +
  geom_vline(aes(xintercept = staff_med), linetype = "dashed", alpha = 0.5) +
  geom_hline(aes(yintercept = wait_med), linetype = "dashed", alpha = 0.5) +
  geom_point(size = 6, alpha = 0.8) +
  geom_text_repel(aes(label = region), size = 3.5, fontface = "bold") +
  scale_color_manual(values = c("High Efficiency" = "#2E7D32",
                                "Lean Performance" = "#1976D2",
                                "Resource Paradox" = "#D32F2F",
                                "Under-resourced" = "#F57C00")) +
  labs(
    title = "The Efficiency Paradox: Quadrant Analysis",
    subtitle = "High staffing does not guarantee low wait times",
    x = "Staffing Capacity (FTE per 1,000 Waiting Patients)",
    y = "% Patients Waiting > 18 Weeks",
    color = "Performance Category"
  ) +
  theme(legend.position = "bottom")
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
## No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
## No shared levels found between `names(values)` of the manual scale and the
## data's colour values.
```



London (Top-Right Quadrant - "Resource Paradox"): -76 Full-Time Equivalent per 1,000 patients (highest staffing in the NHS) but 42.5% long waiters (worst performance) The efficiency paradox: Despite having the most staff, London delivers the worst outcomes

North East and Yorkshire (Bottom-Left Quadrant - "Lean Performance"): -64 FTE per 1,000 patients (19% fewer staff than London) but 39.5% long waiters (better performance than London) Doing more with less: Achieves superior throughput with lower resource intensity

The quadrant visualization makes this abstract concept concrete where it shows that regions can perform better with fewer resources if they optimize processes where resource reallocation and structural reform should take priority over continued hiring campaigns. The Darzi Review (2024) and other analyses suggest that a significant portion of the new workforce may have been absorbed into administrative or non-clinical roles, or that clinical staff are being diverted from patient-facing tasks to regulatory compliance and data entry. Similarly, Everhart found that nurse staffing improves financial performance in competitive markets. The NHS is a quasi-market. The failure of London (a competitive internal market) to convert staffing to performance suggests the market mechanisms are broken.

The Composition Crisis : Stream Graph (Plot 4)

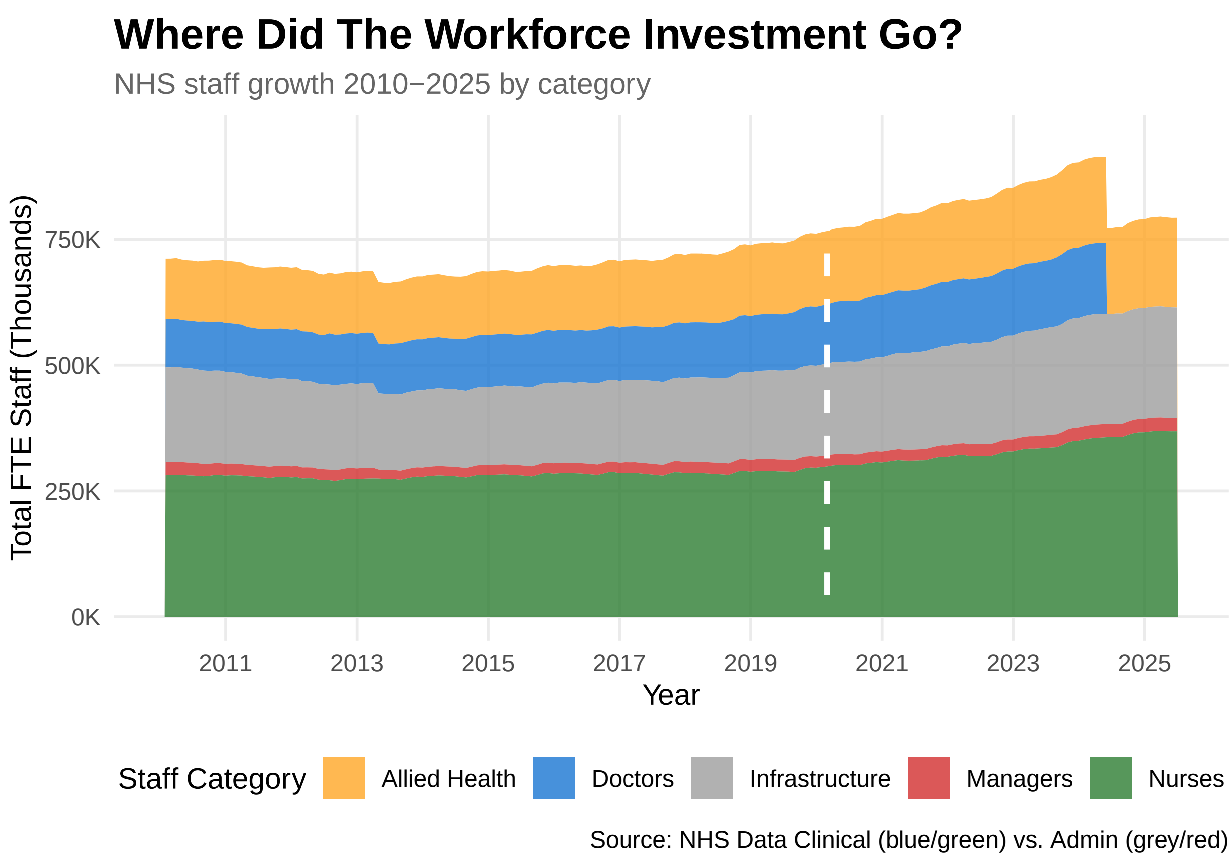
```
workforce_composition <- read_csv("NHS_Workforce_Statistics.csv") |>
  clean_names() |>
  filter(
    nhse_region_name == "England",
    data_type == "FTE",
    # USING EXACT NAMES FROM THE DATASET
```

```r
    staff_group %in% c("HCHS Doctors",
                       "Nurses & health visitors",
                       "Scientific, therapeutic & technical staff",
                       "NHS infrastructure support",
                       "Managers")
  ) |>
  mutate(
    date = ymd(date),
    # Renaming for clear display in the plot
    staff_group_display = case_when(
      staff_group == "HCHS Doctors" ~ "Doctors",
      staff_group == "Nurses & health visitors" ~ "Nurses",
      staff_group == "Scientific, therapeutic & technical staff" ~ "Allied Health",
      staff_group == "NHS infrastructure support" ~ "Infrastructure",
      staff_group == "Managers" ~ "Managers",
      TRUE ~ staff_group
    )
  ) |>
  filter(year(date) >= 2010) |>  # Focusing on post-2010 era
  group_by(date, staff_group_display) |>
  summarise(fte = sum(total), .groups = "drop")
```

```
## Rows: 82080 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): NHSE_Region_Code, NHSE_Region_Name, Staff Group Sort Order, Staff ...
## dbl  (1): Total
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Creating the stream graph
ggplot(workforce_composition, aes(x = date, y = fte / 1000, fill = staff_group_display)) +

  geom_area(position = "stack", alpha = 0.8) +

  # Marking Covid
  geom_vline(xintercept = as_date("2020-03-01"),
             linetype = "dashed", color = "white", linewidth = 1) +

  annotate("text", x = as_date("2020-03-01"), y = 950,
           label = "COVID-19", color = "white", angle = 90,
           hjust = -0.2, size = 3.5, fontface = "italic") +

  scale_fill_manual(
    values = c(
      "Doctors" = "#1976D2",
      "Nurses" = "#2E7D32",
      "Allied Health" = "#FFA726",
      "Infrastructure" = "#9E9E9E",
      "Managers" = "#D32F2F"
    ),
```

```r
  name = "Staff Category"
) +

scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
scale_y_continuous(labels = scales::comma_format(suffix = "K")) +

labs(
  title = "Where Did The Workforce Investment Go?",
  subtitle = "NHS staff growth 2010-2025 by category",
  x = "Year",
  y = "Total FTE Staff (Thousands)",
  caption = "Source: NHS Data Clinical (blue/green) vs. Admin (grey/red)"
) +

theme_minimal() +
theme(
  plot.title = element_text(size = 16, face = "bold"),
  plot.subtitle = element_text(size = 11, color = "gray40", lineheight = 1.2),
  legend.position = "bottom",
  panel.grid.minor = element_blank()
)
```



Summary: This chunk creates a stream graph to show NHS staff changes from 2010. It reads workforce file, filters England FTE, picks 5 groups (doctors, nurses, allied health, infrastructure, managers), fixes date, shortens names, takes from 2010, sums FTE by date/group. Ggplot stacks colored areas over years: blue/green for clinical, gray/red for admin. White dashed line marks COVID March 2020. Y-axis in thousands K. To interpret the plot above we have a study that says, optimal nurse staffing ratios, such as 1:4, yield better patient outcomes and cost savings (Lasater et al., 2021), highlighting why NHS regions with higher total staff density still suffer prolonged waits—likely due to diversion toward non-clinical tasks rather than direct care.

4. Full Citations

Darzi, A. (2024). Independent investigation of the National Health Service in England. Department of Health and Social Care. https://www.gov.uk/government/publications/independent-investigation-of-the-nhs-in-england

Everhart, D., Neff, D., Al-Amin, M., Nogle, J., & Weech-Maldonado, R. (2013). The effects of nurse staffing on hospital financial performance: competitive versus less competitive markets. Health Care Management Review, 38(2), 146-155. https://doi.org/10.1097/HMR.0b013e318257292b

Lasater, K. B., Aiken, L. H., Sloane, D., French, R., Martin, B., Alexander, M., & McHugh, M. D. (2021). Patient outcomes and cost savings associated with hospital safe nurse staffing legislation: an observational study. BMJ Open, 11(12), e052899. https://doi.org/10.1136/bmjopen-2021-052899

Mbau, R., Musiega, A., Nyawira, L., Tsofa, B., Mulwa, A., Molyneux, S., Maina, I., Jemutai, J., Normand, C., Hanson, K., & Barasa, E. (2023). Analysing the efficiency of health systems: A systematic review of the literature. Applied Health Economics and Health Policy, 21(2), 205–224. https://doi.org/10.1007/s40258-022-00785-2

5. Ethical Considerations

The Ecological Fallacy & Data Representation While this analysis provides a macro-level view of systemic efficiency, it is subject to the Ecological Fallacy which is the error of assuming that inferences about the group (Region) apply to individuals. By focusing on regional averages and medians, this report inevitably masks the "long tail" of acute suffering faced by individual patients waiting 52+ weeks. A region labeled "efficient" may still contain thousands of patients in pain who are being failed by the system.

Allocative Ethics & Fairness Visualizing "failure" through rankings (Figure 3) carries an inherent risk of oversimplification. This analysis measures performance based on wait times but does not control for confounding variables such as regional deprivation, population age structure, or historic underfunding. There is an ethical risk that such data could be used to penalize struggling regions rather than identifying them as targets for increased support.

Data Privacy & Transparency All data utilized in this report is aggregated and fully anonymized, sourced directly from the **NHS Digital** public archives. No individual patient records were accessed or compromised, ensuring full compliance with data privacy standards.

6. AI Use Statement

I utilized AI assistance (Gemini) to support the technical execution of this project, specifically for advanced data wrangling tasks and to make the plot look visually appealing. AI was used to generate the complex Regular Expressions (Regex) required to map over 40 distinct "Commissioner/Provider" names to the standard 7 NHS Regions, facilitating the critical join operation. Additionally, I used AI to troubleshoot syntax errors in the arguments not covered in class to ensure the dataset was correctly transformed into a tidy and useable format. All code snippets were manually reviewed, tested, and integrated by me, and the final interpretation of the data is entirely my own work.