Uber Analytics Infographic Code

Jasmine Kwok

Loading in Libraries

```r
# Load libraries
library(dplyr)
library(scales)
library(ggplot2)
library(patchwork)
library(gghighlight)
library(tidyquant)
library(ggseas)
library(slider)
library(ggmap)
library(ggtext)
library(tidygeocoder)
library(sf)
```

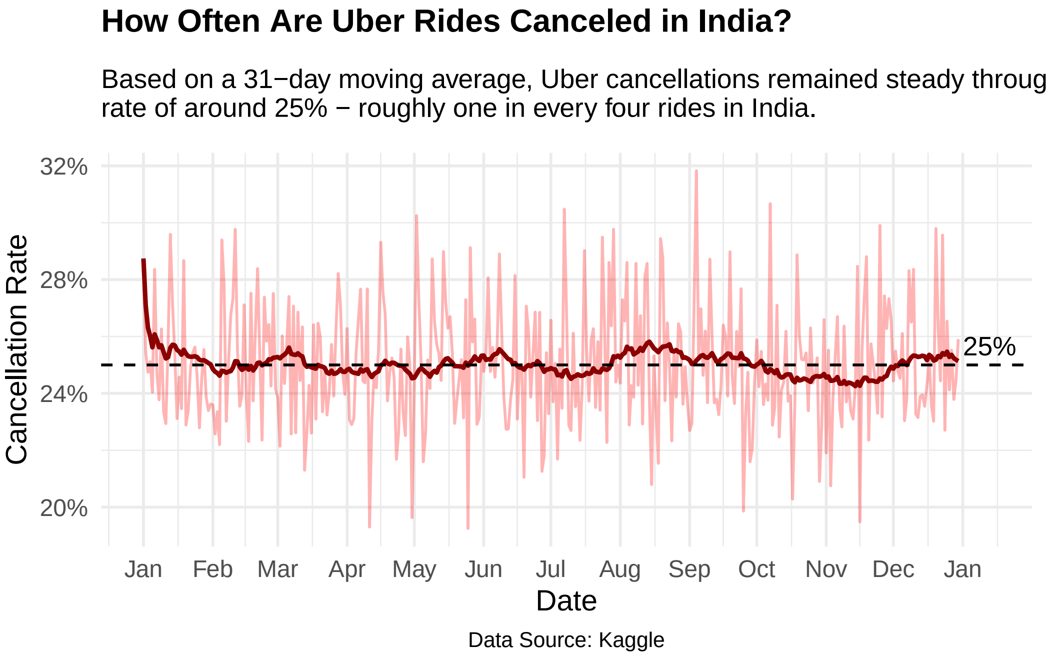
Data Preprocessing

```r
# Load in the data
uber <- read.csv('../data/uber_ride_bookings.csv')
# head(uber)

# add day of the week from date
uber$Date <-as.Date(uber$Date)
uber$DayOfWeek <- weekdays(uber$Date)

# add hour from time
uber$Time <- as.POSIXct(uber$Time, format = "%H:%M:%S")
uber$Hour <- as.numeric(format(uber$Time, "%H"))
```

```r
# order day of weeek
uber <- uber |>
           mutate(DayOfWeek = factor(DayOfWeek,
                        levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturda
                        ordered = TRUE))

# filter to only cancellations
uber_cancellations <- uber |>
  filter(Cancelled.Rides.by.Customer == 1 | Cancelled.Rides.by.Driver == 1)
```

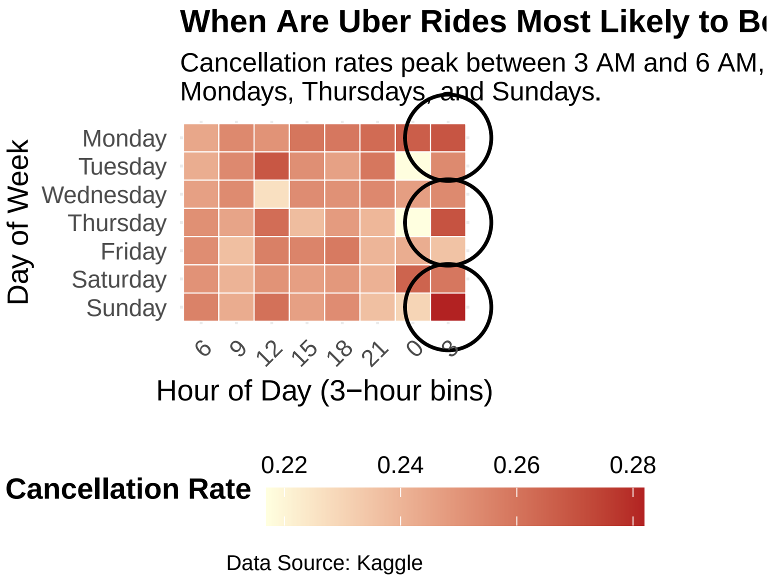
Time Series

```r
# Step 1: Preprocess data - combine cancellation types
timeseries_cancellations <- uber %>%
  mutate(Date = as.Date(Date)) %>%
  group_by(Date) %>%
  summarise(
    total_bookings = n(),
    cancellations = sum(Booking.Status %in% c("Cancelled by Driver", "Cancelled by Customer"
    .groups = "drop"
  ) %>%
  mutate(
    cancellation_rate = cancellations / total_bookings
  ) %>%
  arrange(Date)

# Step 2: Compute 31-day moving average (partial windows allowed)
timeseries_cancellations <- timeseries_cancellations %>%
  mutate(
    ma_31 = slide_dbl(cancellation_rate, mean, .before = 30, .complete = FALSE)
  )

# Step 3: Plot
timeseries <- ggplot(timeseries_cancellations, aes(x = Date)) +
  geom_line(aes(y = cancellation_rate), alpha = 0.3, color = "red") +       # raw rate
  geom_line(aes(y = ma_31), size = 0.8, color = "darkred") +                # smoothed rate
  scale_y_continuous(labels = scales::percent) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  labs(
    x = "Date",
    y = "Cancellation Rate",
    title = "How Often Are Uber Rides Canceled in India?",
    subtitle = "
Based on a 31-day moving average, Uber cancellations remained steady throughout 2024 at an al
rate of around 25% - roughly one in every four rides in India.
",
    caption = "Data Source: Kaggle"
  ) +
  theme_minimal() +
  # 3. LEGEND AND LAYOUT: Move legend to the bottom for a horizontal feel.
  theme(
    plot.title = element_text(size = 12, face = "bold", margin = margin(b = 0.5)),
    plot.subtitle = element_text(size = 10, margin = margin(b = 0.5)),
    plot.caption = element_text(hjust = 0.5, size = 8),
    axis.text.y = element_text(hjust = 1)
  ) +
  geom_hline(yintercept = 0.25, linetype = "dashed", color = "black") +
  annotate("text", x = as.Date("2025-01-13"), y = 0.25, label = "25%", color = "black", vjus

timeseries
```



Heat Map

```
# Create 3-hour bins
uber <- uber %>%
  mutate(
    cancelled = (Cancelled.Rides.by.Customer == 1 | Cancelled.Rides.by.Driver == 1),
    # Create 3-hour bins: floor divide hour by 3 and multiply back by 3 (labels 0,3,6,...)
    Hour3 = (Hour %/% 3) * 3
  )

# Define reordered levels so 6 AM comes first, then 9 AM, 12 PM
new_levels <- c(6, 9, 12, 15, 18, 21, 0, 3)

# Define reordered date levels so Monday comes first
date_levels <- c("Sunday","Saturday","Friday","Thursday","Wednesday", "Tuesday","Monday")

# Aggregate cancellation rate by DayOfWeek and 3-hour bins
heat_data_3hr <- uber %>%
  group_by(DayOfWeek, Hour3) %>%
  summarise(rate = mean(cancelled), .groups = "drop")

# Calculate overall average cancellation rate (for reference line)
overall_avg_rate <- mean(heat_data_3hr$rate)

# Create a new column to highlight above-average cancellation rates (optional)
heat_data_3hr <- heat_data_3hr %>%
  mutate(above_avg = rate > overall_avg_rate)

# Points to highlight
highlight_points <- data.frame(
  DayOfWeek = c("Monday", "Thursday", "Sunday"),
  Hour3 = c(3, 3, 3)
)

# Plot normalized heatmap with 3-hour bins, darker color for above-average cancellations
heatmap <-
  heat_data_3hr |>
  ggplot(aes(y = factor(DayOfWeek, level = date_levels),
             x = factor(Hour3, levels = new_levels), fill = rate)) +
  geom_tile(color = "white") +
  # Highlight circles
  geom_point(
    data = highlight_points,
    aes(x = factor(Hour3, levels = new_levels), y = DayOfWeek),
    shape = 21,           # circle with fill and border
    size = 14.5,             # adjust circle size
    color = "black",      # border color
    fill = "transparent", # hollow circle
    stroke = 1          # border thickness
  ) +
  scale_fill_gradient(
    low = "lightyellow",
    high = "firebrick",
    limits = c(min(heat_data_3hr$rate), max(heat_data_3hr$rate)),
    name = "Cancellation Rate"
  ) +
  guides(
  fill = guide_colorbar(
    title = "Cancellation Rate",
    title.position = "left",
    title.vjust = 0.5,         # centers title vertically
    barwidth = unit(5, "cm"),  # makes legend as long as plot (adjust as needed)
    barheight = unit(0.5, "cm"),
    label.position = "top",
    ticks = FALSE
  )) +
  labs(
    title = "When Are Uber Rides Most Likely to Be Canceled?",
    subtitle = "Cancellation rates peak between 3 AM and 6 AM, especially on
Mondays, Thursdays, and Sundays.",
    x = "Hour of Day (3-hour bins)",
    y = "Day of Week",
    caption = "Data Source: Kaggle"
  ) +
  theme_minimal(base_size =) +
  theme(
    plot.title = element_text(size = 12, face = "bold", margin = margin(b = 5)),
    plot.subtitle = element_text(size = 10),
    plot.caption = element_text(hjust = 0.5, size = 8),
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.margin = margin(t = 10, b = 30),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.title = element_text(face = "bold", vjust = 0.5, hjust = 0.5),
    legend.justification = "center",
    legend.box.just = "center",
    legend.key.width = unit(3, "cm")
  ) + coord_fixed(ratio = 0.8, clip = "off")

heatmap
```



When Are Uber Rides Most Likely to Be

Cancellation rates peak between 3 AM and 6 AM,
Mondays, Thursdays, and Sundays.

Data Source: Kaggle

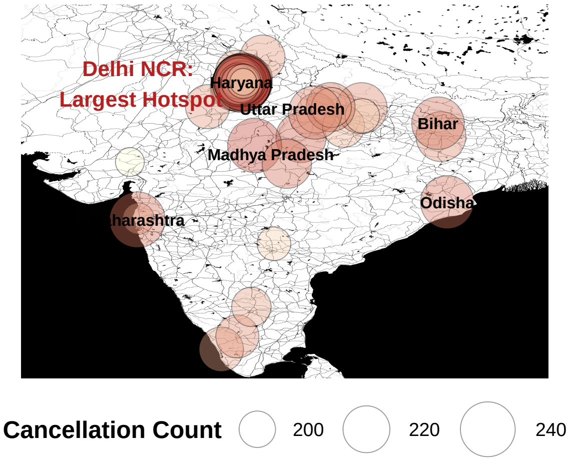
Spatial Data Map + Inset

```
# get latitude and longitude of pickup locations
uber_cancellations <- uber |>
  mutate(full_address = paste(Pickup.Location, "India")) |>
  group_by(Pickup.Location) |>
  summarize(count = n(),
            cancel_count = sum(Cancelled.Rides.by.Customer == 1 | Cancelled.Rides.by.Driver =
            cancel_rate = cancel_count/count) |>
  geocode(address = Pickup.Location, method = 'osm', lat = latitude,
          long = longitude,
          custom_query = list(countrycodes = "in"))
```

```r
# clean uber cancellations
uber_cancellations_clean <- uber_cancellations %>%
  filter(
    !is.na(longitude),
    !is.na(latitude),
    is.finite(longitude),
    is.finite(latitude)
  )

# (in terms of latitude and longitude)
india <- c(left = 68, bottom = 8.0, right = 93.5, top = 34)
map <- get_stadiamap(india, zoom = 7, maptype = "stamen_toner_background")

# subset data for labels
hotspot_labels <- uber_cancellations_clean |> arrange(desc(cancel_rate)) |> head(30)
hotspot_labels <- hotspot_labels |>
  mutate(state = case_when(
      Pickup.Location %in% c("Vinobapuri","Akshardham","Munirka", "Qutub Minar","GTB Nagar","
      Pickup.Location %in% c("Kadarpur","Narsinghpur") ~ "Madhya Pradesh",
      Pickup.Location %in% c("Chhatarpur","Indirapuram", "Vatika Chowk") ~ "Uttar Pradesh",
      Pickup.Location %in% c("Shivaji Park") ~ "Maharashtra",
      Pickup.Location %in% c("Vaishali") ~ "Bihar",
      Pickup.Location %in% c("Rajiv Nagar") ~ "Odisha",
      TRUE ~ "Unknown"))
# maybe mean is not good for latitude
hotspot_labels <- hotspot_labels |>
  group_by(state) |>
  summarize(latitude = mean(latitude),
            longitude = mean(longitude))

# Plot the map
spatial_data_map <-
  ggmap(map) +
  geom_point(
    data = uber_cancellations_clean,
    aes(x = longitude, y = latitude, fill = cancel_rate, size = cancel_count),
    shape = 21,
    color = "black",
    stroke = 0.25,
    alpha = 0.4
  ) +
  # annotate and add text labels for the main hotspots.
  annotate(
    "text", x = 73, y = 28.5, label = "Delhi NCR: \nLargest Hotspot",
    color = "firebrick", size = 2.8, fontface = "bold"
  ) +
  geom_text(data = hotspot_labels ,
  aes(x = longitude, y = latitude, label = state),
  size = 2.2, color = "black", fontface = "bold"
  ) +
  scale_fill_gradient(
    low = "lightyellow",
    high = "firebrick",
    name = "Cancellation Rate"
  ) +
  scale_size_continuous(name = "Cancellation Count", range = c(2, 10)) + # Increased max size
  labs(
    title = "Where Do Ubers Cancel Most in India?",
    subtitle = "Uber cancellations are most concentrated in Delhi-NCR, reflecting
heavy demand and congestion. Across different states and cities,
cancellation rates remain broadly similar with no major regional spikes.",
    caption = "Data Source: Kaggle" # Good practice for infographics
  ) +
  theme_void(base_size = 9) +
  theme(
    plot.title = element_text(size = 12, face = "bold", margin = margin(b = 5, t=10)),
    plot.subtitle = element_text(size = 10, margin = margin(b = 5)),
    plot.caption = element_text(hjust = 0.5, size = 8),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.title = element_text(face = "bold"),
    legend.key.width = unit(0.5, 'cm') # Make the color bar wider
  ) +
  guides(fill = "none") +
  coord_fixed(ratio = 0.6, xlim = c(68, 90), ylim = c(8, 35))

# To view the plot
spatial_data_map
```

Where Do Ubers Cancel Most in India?

Uber cancellations are most concentrated in Delhi−NCR, reflecting
heavy demand and congestion. Across different states and cities,
cancellation rates remain broadly similar with no major regional spik

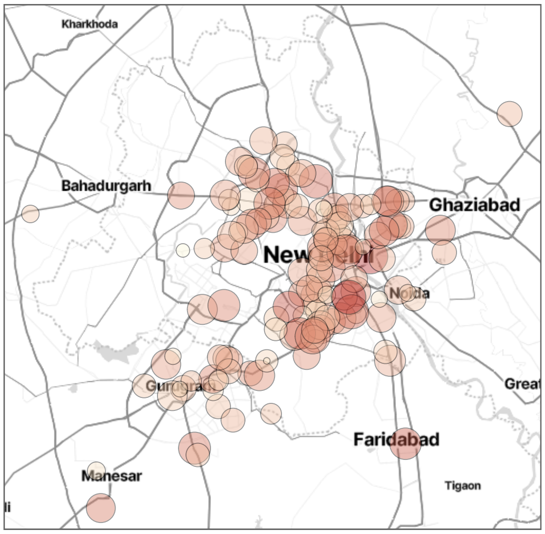


Data Source: Kaggle

```r
# Define Delhi NCR bounding box
delhi_bbox <- c(left = 76.8, bottom = 28.3, right = 77.5, top = 28.9)

# stadia map for inset
delhi_map <- get_stadiamap(bbox = delhi_bbox, zoom = 10, maptype = "stamen_toner_lite")

# create inset plot
inset_map <- ggmap(delhi_map) +
  geom_point(
    data = subset(uber_cancellations_clean, between(latitude, 28.3, 28.9) & between(longitude
    aes(x = longitude, y = latitude, fill = cancel_rate, size = cancel_count),
    shape = 21, color = "black", stroke = 0.25, alpha = 0.4
  ) +
  scale_fill_gradient(low = "lightyellow", high = "firebrick", guide = "none") +
  scale_size_continuous(guide = "none") +
  theme_void() +
  labs(title = "Delhi NCR (Zoomed In)",
       subtitle = "While Central New Delhi forms the dense core of cancellations,
the issue extends to other major hubs, with significant and distinct
clusters in both Gurugram and Noida.") +
  theme(
    plot.title = element_text(
      size = 12,
      face = "bold",
      hjust = 0.5,
      margin = margin(b = 2)
    ),
    panel.border = element_rect(
      fill = NA,
      color = "grey40",
      linewidth = 0.4
    ),
    plot.margin = margin(3, 3, 3, 3),
    plot.subtitle = element_text(size = 10, margin = margin(b = 5)),
  )

inset_map
```

Delhi NCR (Zoomed In)

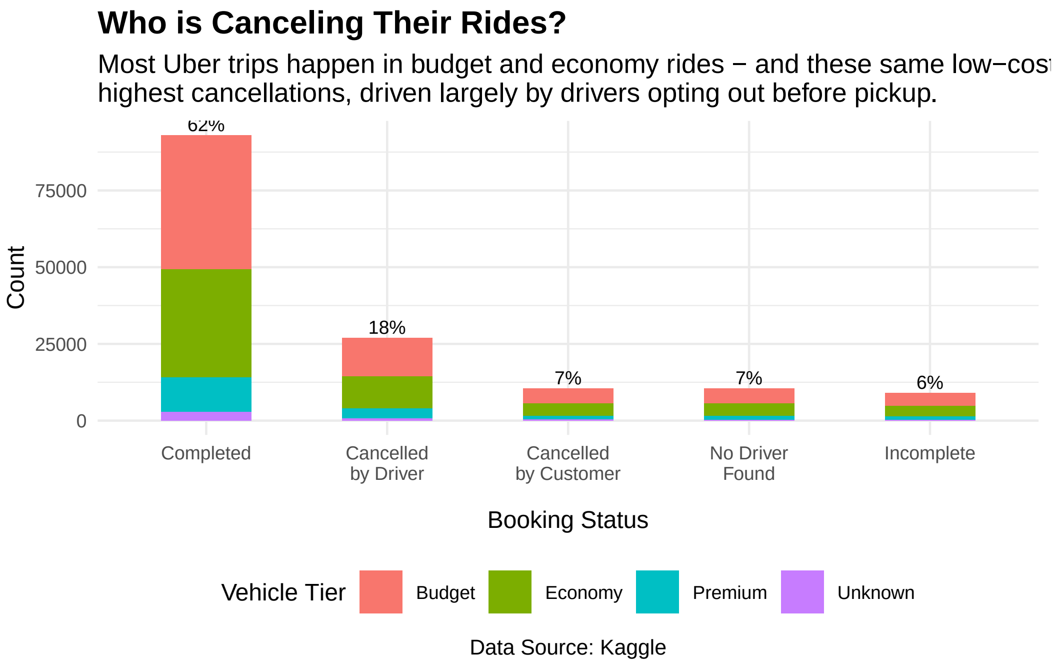
While Central New Delhi forms the dense core of cancellations, the issue extends to other major hubs, with significant and distinct clusters in both Gurugram and Noida.



Stacked Bar Chart

```
# grouped data by booking status and vehicle tier
uber_booking_status_vehicle_tier_grouped_df <- uber |>
  mutate(Tier = case_when(
      Vehicle.Type %in% c("eBike","Bike","Auto") ~ "Budget",
      Vehicle.Type %in% c("Go Mini", "Go Sedan") ~ "Economy",
      Vehicle.Type %in% c("Premier Sedan", "UberXL") ~ "Premium",
      TRUE ~ "Unknown")) |>
  group_by(Booking.Status,Tier) |>
  summarise(count = n()) |>
  mutate(total = sum(count),
         prop = count/total)

# grouped data by booking status
uber_booking_status_grouped_df <- uber  |>
  group_by(Booking.Status) |>
  summarise(count = n()) |>
  mutate(total = sum(count),
         prop = count/total)

stacked_barchart <- uber_booking_status_vehicle_tier_grouped_df |>
  ggplot(aes(x = reorder(Booking.Status,-count), y = count, fill = Tier)) +
    geom_bar(aes(y=count), stat = "identity", width = 0.5) +
    scale_x_discrete(labels = label_wrap(width = 12)) +
    geom_text(data = uber_booking_status_grouped_df,  # Add text labels for the proportions
    aes(x = Booking.Status, y = count, label = scales::percent(prop, accuracy = 1)),  # label
    vjust = -0.3,
    inherit.aes = FALSE,
    size = 2.5) +
    labs(x = "Booking Status",
         y = "Count",
         fill = "Vehicle Tier",
         title = "Who is Canceling Their Rides?",
         subtitle = "Most Uber trips happen in budget and economy rides - and these same low-
highest cancellations, driven largely by drivers opting out before pickup.",
         caption = "Data Source: Kaggle") +
  theme_minimal(base_size = 9) +
  # 3. LEGEND AND LAYOUT: Move legend to the bottom for a horizontal feel.
  theme(
    plot.title = element_text(size = 12, face = "bold", margin = margin(b = 5)),
    plot.subtitle = element_text(size = 10, margin = margin(b = 5)),
    plot.caption = element_text(hjust = 0.5, size = 8),
    axis.title.x = element_text(margin = margin(t = 10)),
    legend.position = "bottom"
  )
```

```
stacked_barchart
```


Who is Canceling Their Rides?
Most Uber trips happen in budget and economy rides − and these same low−cos
highest cancellations, driven largely by drivers opting out before pickup.
Count
Completed
Cancelled by Driver
Cancelled by Customer
No Driver Found
Incomplete
Booking Status
Vehicle Tier
Budget
Economy
Premium
Unknown
Data Source: Kaggle