Introduction to Quantitative Research Methods Seminar

Week 3

James Rice

James.rice@ucl.ac.uk

Thursday, January 29, 2026

Week 3 Focus

- This week we move from summary tables to **visualizing distributions** using `ggplot2`.
- We will:
  - Filter the assault dataset by borough code (e.g. Ealing, Croydon)
  - Create histograms with different bin widths
  - Create a boxplot for a borough
  - Arrange multiple plots on one slide (side-by-side) using `gridExtra`
  - Save plots to image files (for reports/assignments)

Dataset Recap

- We continue using: `Ambulance and Assault Incidents data.csv`.
- Variable of interest: `Assault_09_11` (assault incidents where an ambulance was called, 2009–2011).
- We subset by borough code:
  - **00AJ** = Ealing (in the provided code)
  - **00AH** = Croydon (in the provided code)

```
> head(data)
# A tibble: 6 × 5
  BorCode WardName     WardCode WardType                Assault_09_11
  <chr>   <chr>        <chr>    <chr>                           <dbl>
1 00AA    Aldersgate   00AAFA   Prospering Metropolitan            10
2 00AA    Aldgate      00AAFB   Prospering Metropolitan             0
3 00AA    Bassishaw    00AAFC   Prospering Metropolitan             0
4 00AA    Billingsgate 00AAFD   Prospering Metropolitan             0
5 00AA    Bishopsgate  00AAFE   Prospering Metropolitan           188
6 00AA    Bread Street 00AAFF   Prospering Metropolitan             0
>
```

Why Histograms and Boxplots?

Histograms: a picture of how values are spread out

- Splits the number line into **equal-width ranges** (bins).
- Counts how many observations fall into each range.
- The resulting bars show the **shape of the distribution**.
- **Key idea:** changing the bin width changes the level of detail.

Boxplots: a compact summary of "typical" values and extremes

- The **box** covers the middle 50% of the data (from the 25th to the 75th percentile).
- The **line inside** is the median (the middle value).
- The **whiskers** show how far the data extend beyond the box (to the typical extremes).
- Points beyond the whiskers (outliers) flag **unusually large/small values**.

Rule of thumb: Use a histogram to see overall shape; use a boxplot to compare groups quickly.

Bin Width Intuition

- The **bin width** controls how we group values on the x-axis.
- Smaller bin width:
  - more bars, more detail, potentially noisier
- Larger bin width:
  - fewer bars, smoother shape, can hide structure

In `ggplot2`, you set it with:

`geom_histogram(binwidth =` h`)`

Seminar task: ggplot2 fundamentals + exporting plots

Goal: Learn how to use `ggplot()` to create graphs and apply the most common customisations.

By the end of today you should be able to:

- Create plots with `ggplot()` + a geometry (e.g. `geom_histogram()`, `geom_boxplot()`).
- Add clear titles and axis labels:
  - `ggtitle()`, `xlab()`, `ylab()`
- Apply a theme (e.g. `theme_classic()`) and make cosmetic edits (fill/outline colours, text size).
- Export your plots as **.png** using `ggsave()` (recommended settings: 9cm × 9cm, `dpi=300`).

First Task: Ealing (00AJ) histogram + customisation

Create a histogram of `Assault_09_11` for **Ealing** (Borough code `00AJ`).

Minimum requirements:

- Use `ggplot2` and `geom_histogram()`
- Use **bin width = 75**
- Add title + x/y labels: `ggtitle()`, `xlab()`, `ylab()`

Then apply these customisations:

- Switch to **bin width = 100**
- Apply `theme_classic()` (conservative look)
- Set bar fill to **white** and outline colour to **black**
- Set title size to **8** and make titles/axis labels **bold**
- Set axis text + axis title size to **8**
- Save as **PNG** with `ggsave()` using: 9cm × 9cm, `dpi=300`

Second Task: Boxplot of Ealing and Croydon comparison

Create a boxplot of `Assault_09_11` for **Ealing (00AJ)**.

- Use `geom_boxplot()` (vertical)
- Add `ggtitle()` and appropriate labels
- Apply customisation (experiment)
- Save as a **PNG**

Seminar questions

1. Explain why each visualisation is useful and what type of data are required:
   - Scatter plot
   - Histogram
   - Box & whisker plot
   - Line chart
2. Croydon (Borough code `00AH`): recreate and compare histograms using **binwidth = 50** versus **binwidth = 200**. Place the two plots side-by-side using `gridExtra::grid.arrange(ncol=2, nrow=1)`.
   - Why do the plots look so different?
   - What does this tell you about choosing bin widths?

Step 0: Packages and Loading the Data

The provided solution code uses `ggplot2` and `gridExtra`.

```
library("ggplot2")
library("gridExtra")

# use setwd() and load dataset with read.csv()
data <- read.csv("Ambulance and Assault Incidents data.csv")
```

Note: If `gridExtra` is missing, the code installs it:

```
install.packages("gridExtra")
```

Task 1 (Ealing): Filter Borough 00AJ

We filter rows to keep only Ealing (`BorCode == "00AJ"`).

```
# filter out 00AJ (Ealing)
borCode00AJ <- data[data$BorCode=="00AJ",]
```

```
> head(borCode00AJ)
# A tibble: 6 × 5
  BorCode WardName        WardCode WardType                    Assault_09_11
  <chr>   <chr>           <chr>    <chr>                               <dbl>
1 00AJ    Acton Central   00AJGC   Prospering Metropolitan               229
2 00AJ    Cleveland       00AJGD   Prospering Metropolitan               113
3 00AJ    Dormers Wells   00AJGE   Multicultural Metropolitan            260
4 00AJ    Ealing Broadway 00AJGF   Prospering Metropolitan               258
5 00AJ    Ealing Common   00AJGG   Prospering Metropolitan               125
6 00AJ    East Acton      00AJGH   Multicultural Metropolitan            253
>
```

Task 1 (Ealing): Histogram with Binwidth = 75

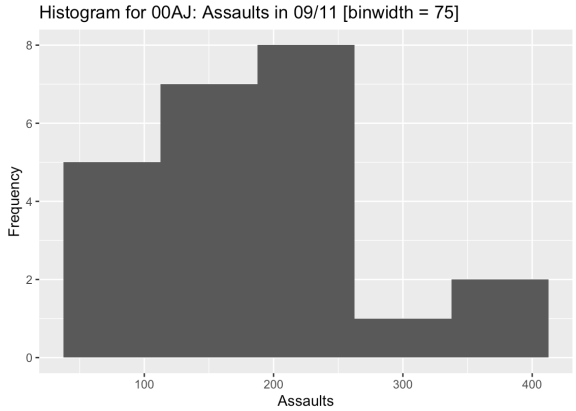
This is the first histogram specification for Ealing.

```
image.plot1 <- ggplot(borCode00AJ, aes(x=Assault_09_11)) +
  geom_histogram(binwidth = 75) +
  ggtitle("Histogram for 00AJ: Assaults in 09/11 [binwidth = 75]") +
  xlab("Assaults") +
  ylab("Frequency")

image.plot1
```

This produces a histogram of `Assault_09_11` for Ealing using bins of width 75.

Ealing Histogram (Binwidth = 75)



Identify the typical range (where bars are tallest) of assault prevalence and any long tails (rare extreme values).

Saving the Plot (Ealing Histogram 75)

The solution code saves the histogram to disk.

```
ggsave("histogram_1.png", image.plot1,
       width=9, height=9, units="cm", dpi=300)
```

This creates `histogram_1.png` at 9cm×9cm, 300 dpi.

Task 2 (Ealing): Histogram with Binwidth = 100 + Styling

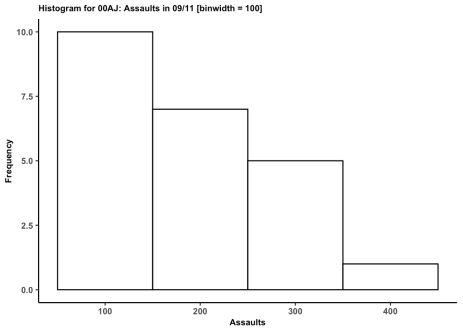
A second histogram for Ealing using a larger bin width and classic styling.

```
image.plot2 <- ggplot(borCodeOOAJ, aes(x=Assault_09_11)) +
  geom_histogram(binwidth = 100, fill="white", colour="black") +
  ggtitle("Histogram for OOAJ: Assaults in 09/11 [binwidth = 100]") +
  xlab("Assaults") +
  ylab("Frequency") +
  theme_classic() +
  theme(plot.title = element_text(size=8, face = "bold"),
        axis.text=element_text(size=8, face = "bold"),
        axis.title=element_text(size=8,face="bold"))

image.plot2
```

Conceptually, this produces the same histogram but with **binwidth 100** and bolder labeling.

Ealing Histogram (Binwidth = 100)



Compare to binwidth 75):

- With **binwidth = 100**, bars are wider and the shape looks smoother.
- With **binwidth = 75**, you see more local variation (more "detail").
- Neither is "correct" universally; the goal is a readable summary without hiding important structure.

Saving the Plot (Ealing Histogram 100)

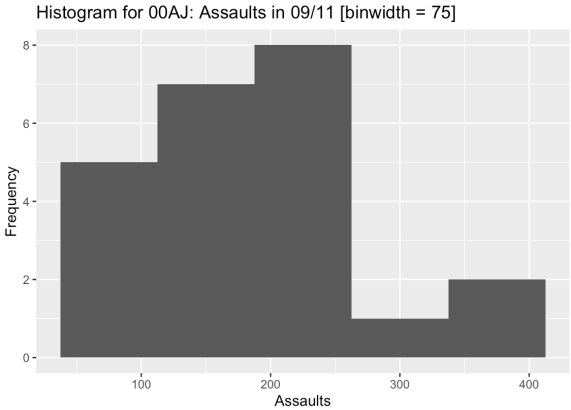
The solution code saves the second histogram.

```
ggsave("histogram_2.png", image.plot2,
       width=9, height=9, units="cm", dpi=300)
```

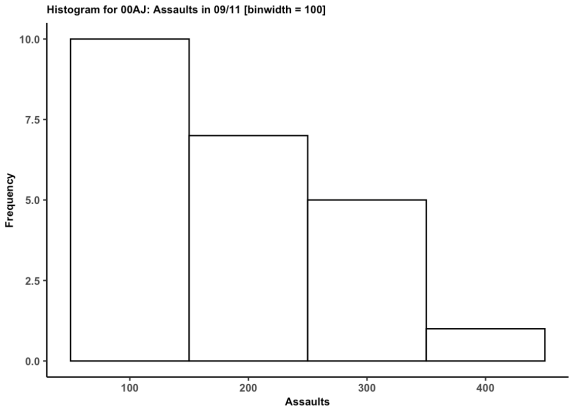
This creates `histogram_2.png` at 9cm×9cm, 300 dpi.

Ealing Histograms (Binwidth = 75 vs 100)

Binwidth = 75



Binwidth = 100



Identify the typical range (where bars are tallest) of assault prevalence and any long tails (rare extreme values). How does changing the binwidth alter what you notice first?

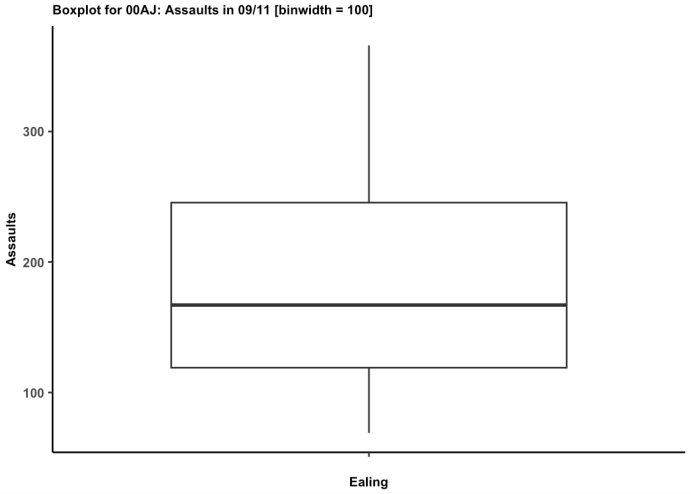
Task 3 (Ealing): Boxplot

We now summarize Ealing's distribution with a boxplot.

```
image.plot3 <- ggplot(borCodeOOAJ, aes(x="", y=Assault_09_11)) +
  geom_boxplot() +
  theme_classic() +
  ggtitle("Boxplot for OOAJ: Assaults in 09/11 [binwidth = 100]") +
  xlab("Ealing") +
  ylab("Assaults") +
  theme(plot.title = element_text(size=8, face = "bold"),
        axis.text=element_text(size=8, face = "bold"),
        axis.title=element_text(size=8,face="bold"))

image.plot3
```

Ealing Boxplot



Saving the Plot (Ealing Boxplot)

```
ggsave("boxplot_3.png", image.plot3,
       width=9, height=9, units="cm", dpi=300)
```

This creates `boxplot_3.png`.

Seminar questions (1): What does each plot *do*?

Task: For each visualisation below, explain:

- **Why it is useful** (what question it helps you answer).
- **What data you need** to produce it (variable type(s), structure, and any time/order requirement).

Visualisation	Useful for	Data required
Scatter plot		
Histogram		
Box & whisker plot		
Line chart		

In one sentence, say what you would *look for* in the plot (trend, skew, outliers, seasonality, clusters, etc.).

Seminar questions (2): Croydon histograms (binwidth sensitivity)

Task: For Croydon (`BorCode == "00AH"`), recreate and compare histograms of `Assault_09_11` using:

- **binwidth = 50** vs **binwidth = 200**
- Same labels/theme across both plots (so the comparison is fair)
- Arrange the two plots **side-by-side**

Answer in writing:

1. Why do the two histograms look so different?
2. What does this tell you about how binwidth selection can shape interpretation?
3. What would be a sensible binwidth choice here, and why?

Seminar Question (Croydon): Filter Borough 00AH

The code repeats the workflow for Croydon (`BorCode == "00AH"`).

```
data <- read.csv("Ambulance and Assault Incidents data.csv")

# filter out 00AH (Croydon)
borCode00AH <- data[data$BorCode=="00AH",]
```

Goal: compare how bin width changes the appearance of Croydon's histogram, and place two plots side-by-side.

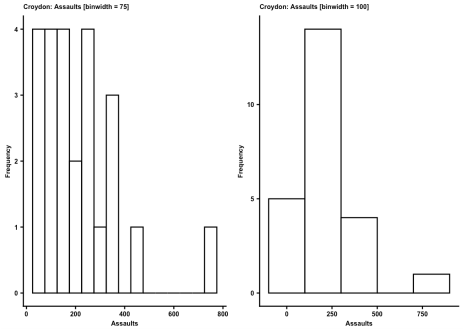
Croydon: Two Histograms + Arrange Side-by-Side

The solution code builds two histograms and arranges them in a 2-column layout.

```
plot.hist_75 <- ggplot(borCodeOOAH, aes(x=Assault_09_11)) +
  geom_histogram(binwidth = 50, fill="white", colour="black") +
  ggtitle("Croydon: Assaults [binwidth = 75]") +
  xlab("Assaults") +
  ylab("Frequency") +
  theme_classic() +
  theme(plot.title = element_text(size=6, face = "bold"),
        axis.text=element_text(size=6, face = "bold"),
        axis.title=element_text(size=6,face="bold"))

plot.hist_100 <- ggplot(borCodeOOAH, aes(x=Assault_09_11)) +
  geom_histogram(binwidth = 200, fill="white", colour="black") +
  ggtitle("Croydon: Assaults [binwidth = 100]") +
  xlab("Assaults") +
  ylab("Frequency") +
  theme_classic() +
```

Croydon Side-by-Side Histogram



What you observe:

- The **smaller binwidth** plot (50) will show more jagged detail and local peaks.
- The **larger binwidth** plot (200) will look much smoother and may conceal sub-structure.
- Use the comparison to justify a binwidth choice that best communicates distribution shape.

Saving the Croydon combined figure

The combined two-panel figure is saved as:

```
ggsave("histogram plot 4.png", plot.hist_75_100,
       width=15, height=9, units="cm", dpi=300)
```

Answer: This creates `histogram plot 4.png` (15cm×9cm, 300 dpi) containing **two Croydon histograms side-by-side**.