

Application of R for Data Analysis in Social Science Research

Dr. Kapil Kumar

Associate Professor, Department of Statistics
Chaudhary Charan Singh University, Meerut, India

kapilstats@gmail.com

ICSSR-sponsored Research Methodology Course: A Comprehensive Research
Methodology Course for Research Scholars of Social Science (18–27 February
2025)

Centre for International Trade and Development, School of International
Studies, Jawaharlal Nehru University, New Delhi

26 February 2025 · Session 2 of 2

Where we left off

Session 1 gave us

- RStudio, scripts and a project folder with `data/`
- vectors, data frames and factors
- reading CSV, text and Excel files
- group summaries with `tapply()`
- base graphics: scatter, histogram, box plot

And a first look at `ito.csv` 659 days of air-quality readings at ITO, Delhi, in 11 columns. Date was stored as text, and some readings were missing.

Today We turn that file into a small research study: question, data, cleaning, exploration, tests, a model, and an honest list of limitations.

Plan of the session

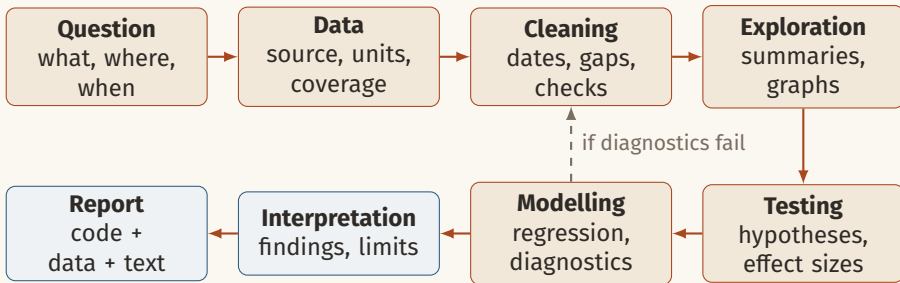
1. Question and workflow
2. Data and cleaning
3. Exploring the data
4. Testing hypotheses
5. Modelling
6. Findings and limitations

By the end you can frame a research question for secondary data, document cleaning decisions, explore with `dplyr` and `ggplot2`, choose and read a t , paired or chi-square test, fit and diagnose a regression, and state what the data cannot tell you.

PART 1

Question and workflow

A research workflow



Every box leaves a trace in the script, so a reader can check each decision. That, more than any single test, makes secondary-data research credible.

The case: air quality at ITO, Delhi

- ITO is a busy traffic junction in central Delhi with a continuous monitoring station
- Air quality is a social-science topic: health burden, unequal exposure, commuting, festivals, crop-residue burning and public policy
- Secondary data: we did not design the measurement, so we must check it

Research questions **RQ1** How does $PM_{2.5}$ vary over months and seasons, and how often does it exceed the national standard?

RQ2 Do the pollutants move together?

RQ3 Was 2024 different from 2023 once the season is allowed for?

Fix the questions *before* looking at results.

Packages and colours used in this session

- **dplyr**: data wrangling with verbs and the pipe %>%
- **ggplot2**: layered graphics
- **lubridate**: reading and splitting dates
- **forecast**: moving averages, autocorrelation, ARIMA errors
- **lmtest**: regression diagnostic tests

```
# install once
# install.packages(c("dplyr", "ggplot2",
#   "lubridate", "forecast", "lmtest"))
library(dplyr)
library(ggplot2)
library(lubridate)
library(forecast)
library(lmtest)
col_main <- "#A9471F"   # rust
col_alt  <- "#2F5D80"   # slate blue
```

Two new tools in one slide

dplyr: verbs for data frames

`mutate()` add or change columns;
`filter()` keep rows; `group_by()` +
`summarise()` one row per group;
`arrange()` sort.

The pipe `%>%` reads as “and then”:
`ito %>% group_by(Season) %>%
 summarise(mean(PM25))`

ggplot2: a grammar of graphics A

plot = data + aesthetic mapping +
geometric layers:

```
ggplot(ito, aes(Date, PM25)) +  
  geom_line() +  
  labs(y = "PM2.5")
```

Add layers with `+`: lines, points, boxes,
reference lines, labels.

Base R from Session 1 does all of this too; these packages make longer pipelines easier to read.

PART 2

Data and cleaning

The data

- Daily averages for the ITO station from the Central Control Room for Air Quality Management, Delhi NCR (Central Pollution Control Board), airquality.cpcb.gov.in/ccr
- One row per day: `ito.csv`

Units Concentrations in $\mu\text{g}/\text{m}^3$; CO in mg/m^3 .

Column	Meaning
PM25, PM10	particles below 2.5 / 10 micrometres
NO, NO2, Nox	nitrogen oxides (exhaust)
NH3	ammonia
SO2	sulphur dioxide (coal, oil)
CO	carbon monoxide (combustion)
Ozone	ground-level ozone (sunlight)
Benzene	benzene (fuel vapour)

Reading the file and fixing the dates

```
ito <- read.csv("data/ito.csv")
ito$Date <- dmy(ito$Date)      # "01-01-2023" -> Date
class(ito$Date); range(ito$Date)

## [1] "Date"
## [1] "2023-01-01" "2024-10-20"

all_days <- seq(min(ito$Date), max(ito$Date),
                by = "day")
nrow(ito); sum(duplicated(ito$Date)); length(all_days)

## [1] 659
## [1] 0
## [1] 659
```

- `dmy()` reads day-month-year text as a real date, so R can sort it and extract months
- Checks: date range, duplicated days, and whether every calendar day is present

Reading it 659 consecutive days, 1 January 2023 to 20 October 2024, none duplicated: not two full years.

Missing values

```
colSums(is.na(ito[, -1]))
```

```
##      PM25      PM10      NO      NO2      Nox      NH3      SO2  
##       20       20      17      17      17      17      17  
##       CO      Ozone Benzene  
##       17       18       21
```

```
miss <- ito$Date[is.na(ito$PM25)]  
table(format(miss, "%Y-%m")) # when?
```

```
##  
## 2023-05 2023-06 2023-12 2024-08  
##      14       1       2       3
```

Decisions Most gaps come from a station outage, 18 May to 1 June 2023. Missing is not zero, and we do not invent values for tests or models. `na.omit()` on all ten pollutants would drop 25 days; an analysis of PM_{2.5} needs to drop only the 20 days without PM_{2.5}.

Plausibility checks

```
mins <- round(sapply(ito[, -1], min, na.rm = TRUE), 2) # any negative readings?
mins; sum(ito$PM25 > ito$PM10, na.rm = TRUE)          # PM2.5 is part of PM10

##      PM25      PM10       NO       NO2      Nox      NH3      SO2       CO      Ozone Benzene
##    10.02    11.48     7.93     2.30    14.87     7.16     3.05     0.25     5.21     0.56
## [1] 4

sd_month <- round(tapply(ito$Benzene, format(ito$Date, "%Y-%m"), sd, na.rm = TRUE), 2)
sd_month[c("2023-08", "2023-09", "2024-08", "2024-09", "2024-10")] # day-to-day variation

## 2023-08 2023-09 2024-08 2024-09 2024-10
##      1.10      0.55      0.30      0.05      0.03
```

No negatives; 4 impossible days ($PM_{2.5} > PM_{10}$); benzene flat from July 2024.

Cleaning in code: every decision visible

```
ito$Benzene[ito$Date >= as.Date("2024-07-03")] <- NA # flat-lined sensor
seasons <- c("Winter", "Pre-monsoon", "Monsoon", "Post-monsoon")
ito <- ito %>%
  mutate(Year = year(Date),
         Month = factor(month.abb[month(Date)], levels = month.abb),
         Season = factor(case_when(month(Date) %in% c(12, 1, 2) ~ seasons[1],
                                   month(Date) %in% 3:5 ~ seasons[2],
                                   month(Date) %in% 6:9 ~ seasons[3],
                                   TRUE ~ seasons[4]),
                        levels = seasons),
         Exceed = PM25 > 60) # above the 24-hour national standard
summary(ito$Season)
```

```
##      Winter  Pre-monsoon    Monsoon Post-monsoon
##         150         184         244          81
```

Data note

What was checked and decided `ito.csv`
is the raw download.

`ITO_Polution_Data.csv`, made in the live session, is the same data after `na.omit()` (634 rows); `Polution_Data.csv` and `Data_Set1.csv` hold simulated values and are not used. Dates: 1 Jan 2023 to 20 Oct 2024, complete, no duplicates. 17–21 missing values per pollutant, mostly one outage. No negative values.

- 4 days with $PM_{2.5} > PM_{10}$ are kept but flagged
- Benzene set to missing from 3 July 2024 (flat readings)
- NH_3 in early 2023 is several times its 2024 level: possibly an instrument change, so it is not interpreted
- Seasons: winter Dec–Feb, pre-monsoon Mar–May, monsoon Jun–Sep, post-monsoon Oct–Nov
- Standard: $60 \mu g/m^3$ (24-hour, India); WHO guideline $15 \mu g/m^3$

PART 3

Exploring the data

Summary statistics

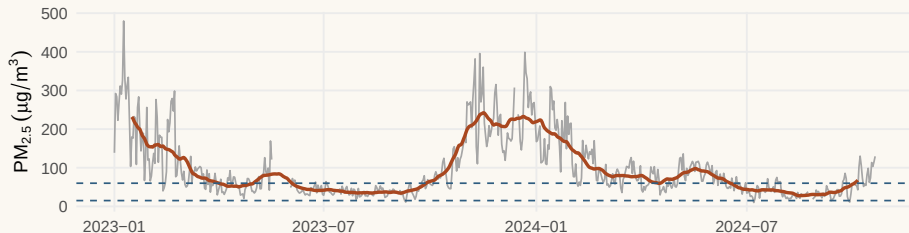
```
vars <- c("PM25", "PM10", "NO2", "SO2", "CO", "Ozone")
round(sapply(ito[, vars], function(v) c(n = sum(!is.na(v)), mean = mean(v, na.rm = TRUE),
  median = median(v, na.rm = TRUE), max = max(v, na.rm = TRUE))), 1)
```

##	PM25	PM10	NO2	SO2	CO	Ozone
## n	639.0	639.0	642.0	642.0	642.0	641.0
## mean	95.2	150.7	72.6	12.4	1.4	27.9
## median	65.3	123.5	59.1	11.9	1.2	26.0
## max	479.8	618.5	322.9	50.4	4.4	84.6

Reading it PM_{2.5}: mean 95, median 65 $\mu\text{g}/\text{m}^3$: a few very bad days pull the mean up. The median day is above the standard of 60, and 99% of days exceed the WHO guideline (15).

PM_{2.5} day by day

```
pm_ma <- ma(na.interp(ts(ito$PM25)), order = 30) # 30-day moving average
ggplot(ito, aes(Date, PM25)) +
  geom_line(colour = "grey65") +
  geom_line(aes(y = as.numeric(pm_ma)), colour = col_main, linewidth = 1) +
  geom_hline(yintercept = c(15, 60), linetype = 2, colour = col_alt) +
  labs(x = NULL, y = expression(PM[2.5]~(mu*g/m^3)))
```



Reading the time plot

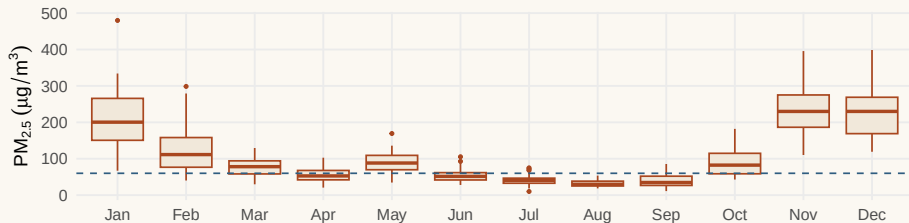
What we see A strong yearly cycle: peaks in November–January, troughs in July–September. The smooth curve (30-day average) looks much the same in 2023 and 2024. $PM_{2.5}$ was above 60 on 54% of days in 2023 and 54% in 2024 (to 20 October).

- Dashed lines: WHO guideline (15) and the Indian 24-hour standard (60)
- Social reading: exposure is seasonal, so winter is when policy (and outdoor workers) feel it most

Method note The 20 missing days were interpolated (`na.interp()`) *only* to draw the moving average.

Monthly distribution

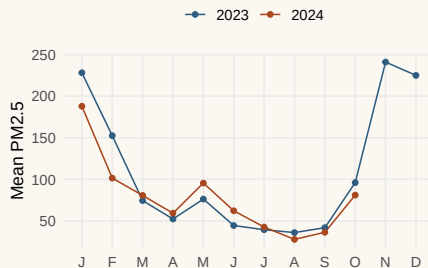
```
ggplot(ito, aes(Month, PM25)) +  
  geom_boxplot(fill = "#F1E8DA", colour = col_main, outlier.size = 0.8) +  
  geom_hline(yintercept = 60, linetype = 2, colour = col_alt) +  
  labs(x = NULL, y = expression(PM[2.5]~(mu*g/m^3)))
```



Monthly medians are below the standard of 60 only in April and from June to September.

Monthly means: 2023 against 2024

```
monthly <- ito %>%  
  group_by(Year, Month) %>%  
  summarise(pm25 = mean(PM25, na.rm = TRUE),  
            .groups = "drop")  
ggplot(monthly, aes(Month, pm25, group = Year,  
                   colour = factor(Year))) +  
  geom_line() + geom_point() +  
  scale_colour_manual(values = c(col_alt, col_main)) +  
  scale_x_discrete(labels = substr(month.abb, 1, 1)) +  
  labs(x = NULL, y = "Mean PM2.5", colour = NULL) +  
  theme(legend.position = "top")
```



Both years follow the same seasonal path; 2024 was lower in January–February and higher in May–June.

How often is the standard exceeded?

```
by_season <- ito %>%  
  filter(!is.na(PM25)) %>%  
  group_by(Season) %>%  
  summarise(days = n(),  
            pct_above_60 = round(100 * mean(Exceed)))  
by_season
```

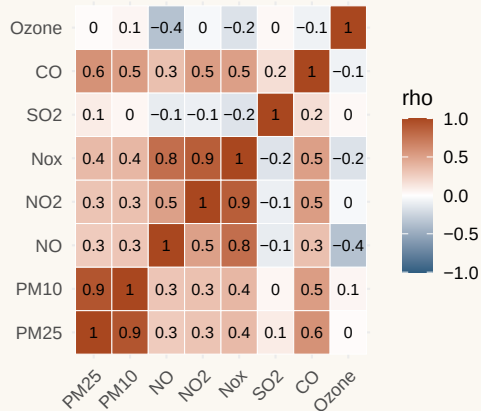
A tibble: 4 x 3

##	Season	days	pct_above_60
##	<fct>	<int>	<dbl>
## 1	Winter	148	97
## 2	Pre-monsoon	170	62
## 3	Monsoon	240	14
## 4	Post-monsoon	81	80

Reading it In winter the standard is breached on 97% of days, in the monsoon on 14%. A percentage of days is easier to communicate to policy makers and the public than a mean concentration.

Do the pollutants move together?

```
polls <- c("PM25", "PM10", "NO", "NO2", "Nox",  
           "SO2", "CO", "Ozone")  
cm <- cor(ito[, polls], method = "spearman",  
          use = "pairwise.complete.obs")  
cm_long <- as.data.frame(as.table(cm))  
ggplot(cm_long, aes(Var1, Var2, fill = Freq)) +  
  geom_tile(colour = "white") +  
  geom_text(aes(label = round(Freq, 1)),  
            size = 3.2) +  
  scale_fill_gradient2(low = col_alt,  
                       high = col_main, limits = c(-1, 1)) +  
  labs(x = NULL, y = NULL, fill = "rho") +  
  theme(axis.text.x = element_text(angle = 45,  
                                    hjust = 1))
```



Reading the correlations

- Spearman's ρ : rank based, so robust to the extreme winter days; NH_3 and benzene left out (doubtful readings)
- $\text{PM}_{2.5}$ and PM_{10} : $\rho = 0.93$, as expected for nested particle sizes with shared sources
- NO , NO_2 , NO_x and CO form a traffic cluster
- Ozone is negatively related to NO ($\rho = -0.35$): near traffic, fresh NO consumes ozone

Correlation is not a common source

Many pollutants are high in winter for the same reason (cold, still air). The season is a third variable behind part of every correlation here, just as in social data where a trend over time links unrelated series.

PART 4

Testing hypotheses

The logic of a test

- H_0 : no difference or no association; H_1 : what we suspect
- Fix the level α (usually 0.05) *before* testing
- **p-value**: the probability, if H_0 were true, of a result at least as extreme as the one observed
- It is *not* the probability that H_0 is true

	H_0 true	H_0 false
Reject H_0	Type I error (α)	correct
Keep H_0	correct	Type II error (β)

Which test? Two independent groups: Welch t (or Wilcoxon rank-sum). Same units measured twice: paired t (or signed-rank). Two categorical variables: chi-square test.

Report effect sizes with confidence intervals, not just “ $p < 0.05$ ”.

Is winter worse than the monsoon?

```
winter <- log(ito$PM25[ito$Season == "Winter"])
monsoon <- log(ito$PM25[ito$Season == "Monsoon"])
tt <- t.test(winter, monsoon) # Welch t-test
ratio <- exp(tt$estimate[[1]] - tt$estimate[[2]])
wx <- wilcox.test(exp(winter), exp(monsoon))
round(c(ratio = ratio, lo = exp(tt$conf.int[1]),
        hi = exp(tt$conf.int[2])), 2)

## ratio    lo    hi
##  4.17  3.79  4.60

c(t_p = tt$p.value, wilcoxon_p = wx$p.value)

##          t_p wilcoxon_p
## 1.972e-83  9.954e-58
```

H_0 : equal mean of $\log \text{PM}_{2.5}$
 $e^{\bar{x}_1 - \bar{x}_2}$ is the ratio of the two
geometric means

Reading it A typical winter day has 4.2 times the $\text{PM}_{2.5}$ of a typical monsoon day (95% CI 3.8 to 4.6); $t = 29.1$ on 259 df, $p < 0.001$; the Wilcoxon test agrees.

Was 2024 cleaner than 2023?

```
js <- monthly %>% filter(as.integer(Month) <= 9)
m23 <- js$pm25[js$Year == 2023]
m24 <- js$pm25[js$Year == 2024]
pt <- t.test(m24, m23, paired = TRUE)
wx2 <- wilcox.test(m24, m23, paired = TRUE)
round(c(diff = pt$estimate[[1]], lo = pt$conf.int[1],
        hi = pt$conf.int[2], p = pt$p.value,
        wilcoxon_p = wx2$p.value), 2)
```

##	diff	lo	hi	p	wilcoxon_p
##	-5.73	-24.59	13.12	0.50	0.91

- Pairs: the same calendar month (January to September) in both years, so the season cancels out

Reading it 2024 was on average $5.7 \mu\text{g}/\text{m}^3$ lower (95% CI -24.6 to 13.1), $p = 0.5$: no evidence of a change. With 9 pairs the test has little power, so this is not proof of “no change”.

Season and exceedance: a chi-square test

```
tab <- table(ito$Season, ito$Exceed); tab

##
##           FALSE TRUE
## Winter           4  144
## Pre-monsoon      65  105
## Monsoon          207   33
## Post-monsoon     16   65

ct <- chisq.test(tab)
round(c(ct$statistic, ct$parameter, p = ct$p.value,
        V = sqrt(ct$statistic[[1]] / sum(tab))), 3)

## X-squared      df          p          V
##    295.09      3.00      0.00      0.68
```

$$X^2 = \sum \frac{(O - E)^2}{E}$$

$$\text{Cramér's } V = \sqrt{X^2 / (n(k - 1))}$$

$$k = \min(\text{rows}, \text{columns}) = 2$$

Reading it TRUE: above 60.
Exceedance depends strongly on the season ($p < 0.001$; $V = 0.68$, a large association). All expected counts exceed 5.

A weekend effect on traffic pollution?

```
ito$Weekend <- wday(ito$Date, week_start = 1) >= 6
tw <- t.test(NO2 ~ Weekend, data = ito)
round(c(tw$estimate, p = tw$p.value), 3)

## mean in group FALSE mean in group TRUE
##           71.406           75.637
##           p
##           0.335

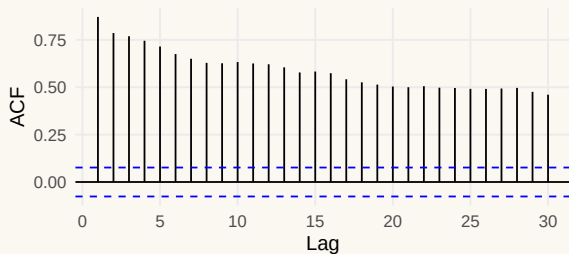
round(tw$conf.int[1:2], 1)           # 95% CI
## [1] -12.9    4.4
```

- Hypothesis: less office traffic on Saturdays and Sundays means less NO₂

Reading it Mean NO₂ is 71.4 on weekdays and 75.6 at weekends; $p = 0.34$. No evidence of a weekend effect at ITO, perhaps because traffic there stays heavy all week. Absence of evidence is not evidence of absence.

The hidden assumption: independent days

```
ggAcf(ito$PM25, lag.max = 30,  
      na.action = na.pass) +  
labs(title = NULL)
```



Why it matters Today's $PM_{2.5}$ is strongly related to yesterday's (lag-1 autocorrelation 0.87). The daily tests treat days as independent, so their p-values are too small.

The winter–monsoon gap survives this; a borderline result would not. Remedy: monthly summaries or a time-series model.

PART 5

Modelling

A first model, and why it misleads

```
# time in years since 1 January 2023
ito$Time <- as.numeric(ito$Date - min(ito$Date)) / 365.25
ito$MonthNum <- month(ito$Date)      # 1, 2, ..., 12
m0 <- lm(PM25 ~ Time + MonthNum, data = ito)
round(coef(summary(m0)), 4)

##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  125.243      7.2682  17.232  0.0000
## Time         -34.261      6.1495  -5.571  0.0000
## MonthNum      0.187      0.9846   0.190  0.8494

summary(m0)$r.squared

## [1] 0.05115
```

The problem Month entered as the number 1–12 forces a straight line from January to December, but $\text{PM}_{2.5}$ is high at *both* ends of the year. The model explains only 5% of the variation, and its “significant” downward trend partly reflects a series that starts in January and ends in October.

A better model: months as categories, log scale

```
m1 <- lm(log(PM25) ~ Time + Month, data = ito)
round(coef(summary(m1))[1:6, ], 4)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	5.2755	0.0500	105.4811	0.0000
## Time	-0.0142	0.0309	-0.4588	0.6465
## MonthFeb	-0.5494	0.0682	-8.0600	0.0000
## MonthMar	-0.9626	0.0669	-14.3987	0.0000
## MonthApr	-1.2973	0.0676	-19.1778	0.0000
## MonthMay	-0.8371	0.0728	-11.4960	0.0000

```
c(R2 = summary(m1)$r.squared, n = nobs(m1))
```

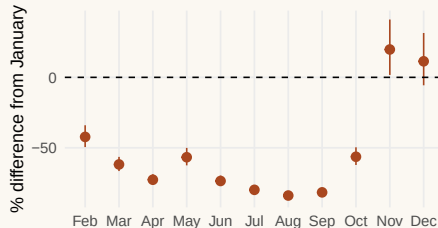
##	R2	n
##	0.7499	639.0000

$\log \text{PM}_{2.5,t} = \beta_0 + \beta_1 t + \gamma_{m(t)} + \varepsilon_t$
 t : years since 1 Jan 2023; $\gamma_{\text{Jan}} = 0$
(reference)

- Month is a factor: one coefficient per month, no shape imposed
- The log scale turns effects into percentages and tames the skew
- R^2 rises to 0.75

Month effects and the trend

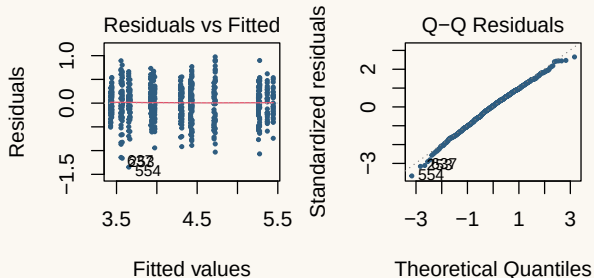
```
ci <- confint(m1)[-1:2, ]
eff <- data.frame(Month = factor(month.abb[-1],
                                levels = month.abb),
                  est = coef(m1)[-1:2],
                  lo = ci[, 1], hi = ci[, 2])
ggplot(eff, aes(Month, 100 * (exp(est) - 1))) +
  geom_pointrange(aes(ymin = 100 * (exp(lo) - 1),
                     ymax = 100 * (exp(hi) - 1)),
                colour = col_main) +
  geom_hline(yintercept = 0, linetype = 2) +
  labs(x = NULL, y = "% difference from January")
```



Reading it August is 84% below January. Trend: -1.4% per year (95% CI -7.2 to 4.8%): no clear change.

Checking the model

```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))  
plot(m1, which = 1:2, pch = 20, cex = 0.6,  
     col = col_alt)
```



```
bp <- bptest(m1)      # Breusch-Pagan  
dw <- dwtest(m1)      # Durbin-Watson  
round(c(BP_p = bp$p.value,  
        DW = dw$statistic[[1]]), 3)
```

```
## BP_p.BP      DW  
##    0.000    0.876
```

Reading it Unequal spread ($p < 0.001$) and strong autocorrelation (DW 0.88, ideal 2): standard errors too small.

Allowing for day-to-day dependence

```
X <- model.matrix(~ Time + Month, data = ito)[, -1]
m2 <- Arima(log(ito$PM25), order = c(1, 0, 0),
            xreg = X) # AR(1) errors
se <- cbind(ols = coef(summary(m1))[2:4, 2],
            ar1 = sqrt(diag(m2$var.coef))[3:5])
round(se, 3); round(coef(m2)[["ar1"]], 2)

##           ols    ar1
## Time      0.031 0.059
## MonthFeb  0.068 0.115
## MonthMar  0.067 0.118
## [1] 0.57
```

$$\varepsilon_t = \phi \varepsilon_{t-1} + u_t \quad (\text{AR}(1) \text{ errors})$$

Reading it $\hat{\phi} = 0.57$: strong day-to-day carry-over. The honest standard errors are about 1.8 times larger. The seasonal pattern stays clear; the trend stays undetectable.

Why not a seasonal decomposition?

- `decompose()` and `stl()` estimate a yearly pattern from repeated years: they need at least two complete cycles
- We have 22 months; a yearly STL is not possible yet
- Setting `frequency = 200` to make `stl()` run imposes a 200-day cycle that does not exist

What we used instead Month effects act as the seasonal component, Time as the trend, residuals as the noise; the 30-day moving average showed trend and season together.

With three or more years use `stl(ts(y, frequency = 365), s.window = "periodic")` or `mstl()`.

PART 6

Findings and limitations

What the data say

RQ1: season and exceedance $\text{PM}_{2.5}$ exceeded the national standard on 54% of days, ranging from 14% in the monsoon to 97% in winter; winter levels are about 4.2 times monsoon levels.

RQ2: co-movement Particles move together, and so do the traffic gases; ozone moves against NO.

RQ3: 2024 vs 2023 No detectable change: paired test $p = 0.5$; model trend -1.4% per year with a wide interval around zero.

Policy reading, with care
Measures concentrated in October–February target the period of highest exposure; these data alone cannot show whether any measure worked.

Limitations

- **One station:** ITO is a traffic hotspot, not “Delhi”
- **22 months:** too short to separate a trend from year-to-year variation
- **No weather data:** wind, inversions and rain drive the season
- **Measurement:** outages, impossible PM values, a stuck benzene sensor
- **Dependence:** daily tests overstate precision
- **Ecological data:** a station reading is not a person's exposure; people who commute, work outdoors or live near roads differ
- **Observational:** associations, not causal effects of policies

Next steps more stations, weather, more years.

Key take-aways

1. **Start from a question**, then check that the data can answer it (coverage, units, gaps).
2. **Clean in code and write a data note**: missing is not zero, impossible values are flagged, every decision is visible.
3. **Explore before testing**: summaries, time plots and group comparisons showed the main story.
4. **Report effects with intervals**, read p-values correctly, and remember “no evidence” is not “no effect”.
5. **Code categories as factors**, check residuals, and respect dependence in time.
6. **State limitations**: one station, short series, no weather, observational data.

References

- Wickham, H., Çetinkaya-Rundel, M., and Grolemund, G. (2023). *R for Data Science*, 2nd Ed., O'Reilly, Sebastopol.
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*, 2nd Ed., Springer, New York.
- Fox, J., and Weisberg, S. (2019). *An R Companion to Applied Regression*, 3rd Ed., Sage, Thousand Oaks.
- Hyndman, R. J., and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*, 3rd Ed., OTexts, Melbourne.
- Breusch, T. S., and Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica*, 47(5), 1287–1294.
- Central Pollution Control Board (2009). *National Ambient Air Quality Standards*, Gazette of India, New Delhi.
- World Health Organization (2021). *WHO Global Air Quality Guidelines: Particulate Matter (PM_{2.5} and PM₁₀), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide*, WHO, Geneva.
- Data: Central Control Room for Air Quality Management, Delhi NCR, Central Pollution Control Board. <https://airquality.cpcb.gov.in/ccr>
- R Core Team. *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna. <https://www.R-project.org>

Thank you

Questions and discussion